Question #1 *Topic 1*

A data engineer is configuring an AWS Glue job to read data from an Amazon S3 bucket. The data engineer has set up the necessary AWS Glue connection details and an associated IAM role. However, when the data engineer attempts to run the AWS Glue job, the data engineer receives an error message that indicates that there are problems with the Amazon S3 VPC gateway endpoint.
The data engineer must resolve the error and connect the AWS Glue job to the S3 bucket.
Which solution will meet this requirement?

A. Update the AWS Glue security group to allow inbound traffic from the Amazon S3 VPC gateway endpoint.

B. Configure an S3 bucket policy to explicitly grant the AWS Glue job permissions to access the S3 bucket.

C. Review the AWS Glue job code to ensure that the AWS Glue connection details include a fully qualified domain name.

D. Verify that the VPC's route table includes inbound and outbound routes for the Amazon S3 VPC gateway endpoint.

Suggested Answer: *D*

Community vote distribution

D (93%) 7%

HunkyBunky Highly Voted 9 months, 1 week ago

Selected Answer: D

A - wrong - AWS glue - are serverless service, so it don't have any security groups
B - wrong - Because we have error with VPC, not with S3 itself
C - wrong - Becuase with S3 - we always have only FQDN for buckets

upvoted 7 times

alexbg88 11 months, 3 weeks ago

they most certainly can have SGs.

upvoted 1 times

Robertwilliamm Most Recent 4 days, 4 hours ago

Selected Answer: D

D. is Correct Option
Thanks to SkillCertExams I successfully cleared my DEA-C01 exam today.

upvoted 1 times

Dreamer78692 1 month, 2 weeks ago

Selected Answer: B

THought needed an explicit security setting

upvoted 1 times

ninomfr64 2 months, 3 weeks ago

Selected Answer: D

A- NO: on SG we just need to allow outbound traffic, as SG i statefull reurn traffic is allowed
B - NO: since we configured IAM permission for Glue Job, there is no need to configure a resource-policy (cross account is not mentioned)
C- NO: in bucket connection configuration you just need to provide s3://bucket-name/prefix
D - YES: although there is no inbound and outbound routes in route table, we need to ensure a route is in place to reach a the VPC Gateway Policy

upvoted 1 times

MephiboshethGumani 4 months ago

Selected Answer: D

D. Verify that the VPC's route table includes inbound and outbound routes for the Amazon S3 VPC gateway endpoint.

Explanation:

AWS Glue jobs need to connect to the S3 bucket through the Amazon S3 VPC gateway endpoint when they are in a VPC. If the route table does not have proper inbound and outbound routes to the S3 VPC gateway endpoint, the AWS Glue job will not be able to access S3, which results in an error.

upvoted 1 times

wilsonfromnyc9 9 months, 1 week ago

D is valid

upvoted 1 times

GiorgioGss 9 months, 1 week ago

Selected Answer: D

Although there is no such thing as "inbound and outbound routes" when we talk about VPC route table, when we define a S3 gateway endpoint we must have proper routes in place. I will go with D.

upvoted 4 times

ampersandor 9 months, 1 week ago

Selected Answer: D

Be sure that the subnet configured for your AWS Glue connection has an Amazon S3 VPC gateway endpoint or a route to a NAT gateway in the subnet's route table.

upvoted 2 times

GZMartinelli 9 months, 3 weeks ago

Selected Answer: D

D is correct

upvoted 1 times

lunachi4 11 months, 2 weeks ago

Selected Answer: D

I think D. We check "VPC's route table"

upvoted 1 times

teo2157 12 months ago

Selected Answer: C

A - wrong - AWS glue doesn't have any security groups

B - wrong - You can´t give permissions in the S3 to the AWS glue job but to the role

D. wrong because there has to be a definend route for the S3 gateway endpoint in the subnet assigned to the glue job but not in the VPC's route table and also route tables doesn´t have inbound and outbound routes.

upvoted 1 times

shammous 9 months, 3 weeks ago

"route tables don´t have inbound and outbound routes."? It does. You need to check how the VPC works in AWS.

upvoted 2 times

nanaw770 1 year ago

Selected Answer: D

D is correct answer.

upvoted 2 times

tgv 1 year, 1 month ago

I will go with D, the other options don't seem to be related.

upvoted 1 times

VerRi 1 year, 1 month ago

Selected Answer: D

"problems with the Amazon S3 VPC gateway endpoint"

upvoted 2 times

damaldon 1 year, 4 months ago

Go with A:

If you receive an error, check the following:

The correct privileges are provided to the role selected.

The correct Amazon S3 bucket is provided.

The security groups and Network ACL allow the required incoming and outgoing traffic.

The VPC you specified is connected to an Amazon S3 VPC endpoint.

upvoted 1 times

Aesthet 1 year, 4 months ago

some relevant info:

main: https://docs.aws.amazon.com/glue/latest/dg/connection-VPC-disable-proxy.html additional (glue crawler instead of glue job here, but I think this is relevant for both): https://docs.aws.amazon.com/glue/latest/dg/connection-S3-VPC.html

upvoted 2 times

⊟ **Aesthet** 1 year, 4 months ago

Both ChatGPT and I agree with D

upvoted 4 times

⊟ **DevoteamAnalytix** 1 year, 2 months ago

:-)) nice

upvoted 1 times

Question #2 *Topic 1*

A retail company has a customer data hub in an Amazon S3 bucket. Employees from many countries use the data hub to support company-wide analytics. A governance team must ensure that the company's data analysts can access data only for customers who are within the same country as the analysts.
Which solution will meet these requirements with the LEAST operational effort?

A. Create a separate table for each country's customer data. Provide access to each analyst based on the country that the analyst serves.

B. Register the S3 bucket as a data lake location in AWS Lake Formation. Use the Lake Formation row-level security features to enforce the company's access policies.

C. Move the data to AWS Regions that are close to the countries where the customers are. Provide access to each analyst based on the country that the analyst serves.

D. Load the data into Amazon Redshift. Create a view for each country. Create separate IAM roles for each country to provide access to data from each country. Assign the appropriate roles to the analysts.

Suggested Answer: *B*

Community vote distribution

B (95%) 5%

k350Secops Highly Voted 1 year, 1 month ago

Selected Answer: B

AWS Lake Formation: It's specifically designed for managing data lakes on AWS, providing capabilities for securing and controlling access to data.
Row-Level Security: With Lake Formation, you can define fine-grained access control policies, including row-level security. This means you can enforce policies to restrict access to data based on specific conditions, such as the country associated with each customer.
Least Operational Effort: Once the policies are defined within Lake Formation, they can be centrally managed and applied to the data in the S3 bucket without the need for creating separate tables or views for each country, as in options A, C, and D. This reduces operational overhead and complexity.

upvoted 12 times

Mike_27 Most Recent 2 weeks, 6 days ago

Selected Answer: B

Strongly agree with B

upvoted 1 times

dried0extents 3 months, 3 weeks ago

Selected Answer: A

I agree that it is A

upvoted 1 times

gray2205 10 months, 1 week ago

if the situation is not about least operational effort, D makes sense

upvoted 1 times

lunachi4 11 months, 2 weeks ago

Selected Answer: B

Select B. It means "with the LEAST operational effort".

upvoted 1 times

nanaw770 1 year ago

Selected Answer: B

B is correct answer.

upvoted 2 times

mattia_besharp 1 year, 2 months ago

Selected Answer: B

AWS really likes Lakeformation, plus creating separate tables might require some refactoring, and the requirements is about the LEAST operational effor

upvoted 1 times

rishadhb 1 year, 3 months ago

Selected Answer: A

Agreed with Bartosz. I think setup DataLake, then integrate it with LakeFormation take a lot of effort than just separate the table

upvoted 1 times

GiorgioGss 1 year, 3 months ago

Selected Answer: B

Keyword "LEAST operational effort" - I will go with B

upvoted 1 times

BartoszGolebiowski24 1 year, 4 months ago

Creating DataLake takes at least few days to set up and the solution should be LEAST operational. I think B is not correct.

upvoted 2 times

[Removed] 1 year, 5 months ago

Selected Answer: B

https://docs.aws.amazon.com/lake-formation/latest/dg/register-data-lake.html

https://docs.aws.amazon.com/lake-formation/latest/dg/registration-role.html

upvoted 3 times

Question #3 *Topic 1*

A media company wants to improve a system that recommends media content to customer based on user behavior and preferences. To improve the recommendation system, the company needs to incorporate insights from third-party datasets into the company's existing analytics platform.
The company wants to minimize the effort and time required to incorporate third-party datasets.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use API calls to access and integrate third-party datasets from AWS Data Exchange.

B. Use API calls to access and integrate third-party datasets from AWS DataSync.

C. Use Amazon Kinesis Data Streams to access and integrate third-party datasets from AWS CodeCommit repositories.

D. Use Amazon Kinesis Data Streams to access and integrate third-party datasets from Amazon Elastic Container Registry (Amazon ECR).

Suggested Answer: *A*

Community vote distribution

A (92%) 8%

KelvinPun Highly Voted 9 months, 1 week ago

Selected Answer: A

AWS DataSync is primarily used for data transfer services designed to simplify, automate, and accelerate moving data between on-premises storage systems and AWS storage services, as well as between different AWS storage services. Its primary role is not for accessing third-party datasets but for efficiently transferring large volumes of data.
In contrast, AWS Data Exchange is designed specifically for discovering and subscribing to third-party data in the cloud, providing direct API access to these datasets, which aligns perfectly with the company's need to integrate this data into their recommendation systems with minimal overhead.

upvoted 14 times

hsnin Highly Voted 1 year, 3 months ago

Selected Answer: A

AWS Data Exchange is a service that makes it easy to share and manage data permissions from other organizations

upvoted 5 times

ttpro1995 Most Recent 6 months, 1 week ago

Selected Answer: A

Yeah, AWS want people to buy data from their marketplace. So, ... you know.

upvoted 3 times

Shubham1989 9 months, 1 week ago

Should be AWS Data Exchange.

upvoted 1 times

lunachi4 11 months, 2 weeks ago

Selected Answer: A

I will go with A. Kinesis Data Stram is more operational overhead.

upvoted 1 times

Manohar24 11 months, 3 weeks ago

Selected Answer: A

A is correct

upvoted 1 times

sudohogan 1 year ago

A is correct, DataSync doesn't really rely on API calls.

upvoted 2 times

nanaw770 1 year ago

Selected Answer: A

A is correct answer.

upvoted 1 times

0060594 1 year, 1 month ago

Selected Answer: A

AWS DataExchange

upvoted 1 times

k350Secops 1 year, 1 month ago

Selected Answer: A

options B, C, and D involve using Amazon Kinesis Data Streams or other services that may not be directly suited for integrating third-party datasets from external sources like AWS Data Exchange. These options might require additional configurations, data processing steps, or infrastructure management, resulting in higher operational overhead compared to directly leveraging AWS Data Exchange's capabilities through API calls (Option A).

upvoted 1 times

kj07 1 year, 3 months ago

I will go with A.

upvoted 1 times

Josa2 1 year, 3 months ago

Selected Answer: B

There is no info or guarantee this third-party dataset is available in AWS to be part of a data-share, hence the more assertive answer is B

upvoted 2 times

GiorgioGss 1 year, 3 months ago

Selected Answer: A

A for me. "You can also discover and subscribe to new third-party data sets available through AWS Data Exchange"
https://docs.aws.amazon.com/data-exchange/latest/userguide/what-is.html

upvoted 3 times

ceramem 1 year, 4 months ago

A

Data exchange is primarily designed for this purpose.

upvoted 2 times

TonyStark0122 1 year, 4 months ago

A

Data exchange is primarily designed for this purpose.

upvoted 3 times

lalitjhawar 1 year, 5 months ago

A

Data Exchange is the AWS official third-party datasets repository: https://aws.amazon.com/data-exchange

upvoted 3 times

Question #4 *Topic 1*

A financial company wants to implement a data mesh. The data mesh must support centralized data governance, data analysis, and data access control. The company has decided to use AWS Glue for data catalogs and extract, transform, and load (ETL) operations.
Which combination of AWS services will implement a data mesh? (Choose two.)

A. Use Amazon Aurora for data storage. Use an Amazon Redshift provisioned cluster for data analysis.

B. Use Amazon S3 for data storage. Use Amazon Athena for data analysis.

C. Use AWS Glue DataBrew for centralized data governance and access control.

D. Use Amazon RDS for data storage. Use Amazon EMR for data analysis.

E. Use AWS Lake Formation for centralized data governance and access control.

Suggested Answer: *BE*

Community vote distribution

BE (100%)

hsnin Highly Voted 1 year, 3 months ago

Selected Answer: BE

The answer is B and E.
The data mesh implementation uses Amazon S3 and Athena for data storage and analysis, and AWS Lake Formation for centralized data governance and access control. When combined with AWS Glue, you can efficiently manage your data.

upvoted 7 times

ninomfr64 Most Recent 2 months, 3 weeks ago

Selected Answer: BE

S3 (storage) LakeFormation (governance) Athena (analytics) can be used to implement data mesh. In real life you would use DataZone or nowadays SageMaker Unified Studio

upvoted 1 times

Shubham1989 9 months, 1 week ago

Selected Answer: BE

S3 is best storage for data lake, and AWS lake formation is best for management.

upvoted 1 times

nanaw770 9 months, 1 week ago

Selected Answer: BE

BE are correct answer.

upvoted 2 times

Josa2 9 months, 1 week ago

Selected Answer: BE

Sometimes I think examtopics uses us to calibrate the right answers hehehe, by the goal statement and the services outlines and objectives there are no way the answer be different then B,E

upvoted 3 times

pypelyncar 9 months, 1 week ago

Selected Answer: BE

A: Cost-effective storage: Amazon S3 is a highly scalable and cost-effective object storage service perfect for storing large datasets commonly found in financial institutions.
Centralized data lake: S3 acts as the central data lake where all data from different domains can reside in its raw or processed form.
Easy data access: Athena provides a serverless interactive query service that allows data analysts to directly query data stored in S3 using standard SQL. This simplifies data exploration and analysis without managing servers.
B: Data governance: Lake Formation helps establish data ownership, access control, and lineage for data products within the data mesh. It ensures data quality, security, and compliance with regulations.
Fine-grained access control: Lake Formation allows you to define granular access policies for each data domain, ensuring only authorized users can access specific data sets. This aligns with the need for centralized control in a data mesh.

upvoted 1 times

Fredrik1 11 months, 1 week ago

Must be B and E

upvoted 1 times

minhtien1707 1 year, 3 months ago

Selected Answer: BE

i thing so

upvoted 1 times

alexua 1 year, 4 months ago

B and E .

C - is not correct "AWS Glue DataBrew is a visual data preparation tool that makes it easier for data analysts and data scientists to clean and normalize data to prepare it for analytics and machine learning (ML)"

upvoted 3 times

Alcee 1 year, 4 months ago

B and E

upvoted 1 times

TonyStark0122 1 year, 4 months ago

BE

Given the requirements for implementing a data mesh architecture with centralized data governance, data analysis, and data access control, the two better choices from the options provided would be:

B. Use Amazon S3 for data storage. Use Amazon Athena for data analysis.

E. Use AWS Lake Formation for centralized data governance and access control.

upvoted 2 times

milofficial 1 year, 5 months ago

Selected Answer: BE

Textbook question, the keyword data mesh means S3, the keyword data governance means LakeFormation

upvoted 4 times

Question #5 *Topic 1*

A data engineer maintains custom Python scripts that perform a data formatting process that many AWS Lambda functions use. When the data engineer needs to modify the Python scripts, the data engineer must manually update all the Lambda functions.
The data engineer requires a less manual way to update the Lambda functions.
Which solution will meet this requirement?

A. Store a pointer to the custom Python scripts in the execution context object in a shared Amazon S3 bucket.

B. Package the custom Python scripts into Lambda layers. Apply the Lambda layers to the Lambda functions.

C. Store a pointer to the custom Python scripts in environment variables in a shared Amazon S3 bucket.

D. Assign the same alias to each Lambda function. Call reach Lambda function by specifying the function's alias.

Suggested Answer: *B*

Community vote distribution

B (100%)

TonyStark0122 Highly Voted 9 months, 1 week ago

B. Package the custom Python scripts into Lambda layers. Apply the Lambda layers to the Lambda functions.
Explanation:
Lambda layers allow you to centrally manage shared code and dependencies across multiple Lambda functions. By packaging the custom Python scripts into a Lambda layer, you can simply update the layer whenever changes are made to the scripts, and all the Lambda functions that use the layer will automatically inherit the updates. This approach reduces manual effort and ensures consistency across the functions.

upvoted 20 times

pypelyncar Most Recent 9 months, 1 week ago

Selected Answer: B

Centralized Code Management: Lambda layers allow you to store and manage the custom Python scripts in a central location outside the individual Lambda function code. This eliminates the need to update the script in each Lambda function manually.
Reusable Code: Layers provide a way to share code across multiple Lambda functions. Any changes made to the layer code are automatically reflected in all the functions using that layer, streamlining updates.
Reduced Deployment Size: By separating core functionality into layers, you can keep the individual Lambda function code focused and smaller. This reduces deployment package size and potentially improves Lambda execution times.

upvoted 4 times

JavierEF 10 months ago

Selected Answer: B

Lambda Layers is a feature created with this literal objective in mind.

upvoted 2 times

John2025 1 year ago

B is right

upvoted 2 times

4c78df0 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 1 times

4c78df0 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 1 times

FunkyFresco 1 year, 1 month ago

Selected Answer: B

Lamba layers

upvoted 1 times

ba72eb9 1 year, 3 months ago

Option B

upvoted 2 times

⊟ **kj07** 1 year, 3 months ago

Typical use case for Lambda Layers.

Option B.

upvoted 2 times

Question #6 *Topic 1*

A company created an extract, transform, and load (ETL) data pipeline in AWS Glue. A data engineer must crawl a table that is in Microsoft SQL Server. The data engineer needs to extract, transform, and load the output of the crawl to an Amazon S3 bucket. The data engineer also must orchestrate the data pipeline.

Which AWS service or feature will meet these requirements MOST cost-effectively?

A. AWS Step Functions

B. AWS Glue workflows

C. AWS Glue Studio

D. Amazon Managed Workflows for Apache Airflow (Amazon MWAA)

Suggested Answer: *B*

Community vote distribution

B (100%)

⊟ **milofficial** Highly Voted 👍 1 year, 5 months ago

Selected Answer: B

Glue workflows are the easiest solution here:

https://aws.amazon.com/blogs/big-data/orchestrate-an-etl-pipeline-using-aws-glue-workflows-triggers-and-crawlers-with-custom-classifiers/

https://aws.amazon.com/blogs/big-data/extracting-multidimensional-data-from-microsoft-sql-server-analysis-services-using-aws-glue/

upvoted 9 times

⊟ **dev_vicente** Highly Voted 👍 9 months, 1 week ago

Selected Answer: B

I asked an AI.

Analysis of the answers:

A. AWS Step Functions:

It is a good option for orchestrating workflows with steps from different AWS services, but requires additional development to connect to Microsoft SQL Server.

B. AWS Glue Workflows:

This is the best and most profitable option. AWS Glue is designed specifically for ETL on AWS and integrates directly with data sources such as Microsoft SQL Server through connectors. This allows for easier configuration and avoids the need for additional development.

C. AWS Glue Studio:

It is a visual interface for AWS Glue that makes it easy to create and manage ETL jobs. However, the underlying functionality comes from AWS Glue (B) workflows.

D. Amazon Managed Workflows for Apache Airflow (Amazon MWAA):

It's a viable option, but it's generally more expensive than native AWS services like AWS Glue Workflows. Additionally, it requires some Airflow experience for setup and maintenance.

upvoted 8 times

⊟ **Adrifersilva** Most Recent ⏲ 9 months ago

Selected Answer: B

https://community.aws/content/2iBQiAGS4RvEolgSQKu4iF8InTV/choose-the-right-data-orchestration-service-for-your-data-pipeline?lang=en

upvoted 1 times

⊟ **Shubham1989** 9 months, 1 week ago

Selected Answer: B

Glue is easiest here to choose from.

upvoted 1 times

⊟ **DevoteamAnalytix** 1 year, 1 month ago

Selected Answer: B

Agree with B. CRAWLING and ETL are the main functions of a Glue workflow and MS SQL is supported:

https://docs.aws.amazon.com/glue/latest/dg/crawler-data-stores.html

upvoted 3 times

⊟ **Alcee** 1 year, 4 months ago

Is B !

upvoted 1 times

Question #7 *Topic 1*

A financial services company stores financial data in Amazon Redshift. A data engineer wants to run real-time queries on the financial data to support a web-based trading application. The data engineer wants to run the queries from within the trading application.
Which solution will meet these requirements with the LEAST operational overhead?

A. Establish WebSocket connections to Amazon Redshift.

B. Use the Amazon Redshift Data API.

C. Set up Java Database Connectivity (JDBC) connections to Amazon Redshift.

D. Store frequently accessed data in Amazon S3. Use Amazon S3 Select to run the queries.

Suggested Answer: *B*

Community vote distribution

B (100%)

ninomfr64 2 months, 3 weeks ago

Selected Answer: B

You can query a Redshift cluster with either JDBC/ODBC or Data API. The latter just requires you to maintain AWS SDK and IAM role, while the former needs network plumbing to access VPC, JDBC/ODBC driver, database credentials possibly stored in Secret Manager

upvoted 1 times

Scotty_Nguyen 3 months, 1 week ago

Selected Answer: B

B is correct

upvoted 1 times

Palee 3 months, 2 weeks ago

Selected Answer: B

Most efficient solution

upvoted 1 times

bhawna901 7 months, 1 week ago

Amazon Redshift Data API:
Provides a serverless and simple HTTP-based API to interact with Redshift.
Ideal for web-based applications since it eliminates the need to manage persistent database connections (like JDBC or ODBC).
Allows the trading application to send queries directly to Redshift using HTTPS requests, making it easy to integrate with modern applications.
Removes the complexity of managing database connection pooling in real-time, which reduces operational overhead.
Securely integrates with IAM roles and policies for authentication and access control.

upvoted 2 times

markill123 9 months, 3 weeks ago

Selected Answer: B

A) Redshift doesn't support WebSockets;
C) It is way harder to manage DB connections than using Redshift Data API which will offer you the possibility to run SQL queries directly.
D)

upvoted 1 times

04e06cb 11 months, 4 weeks ago

Selected Answer: B

B is correct

upvoted 1 times

k350Secops 1 year, 1 month ago

Selected Answer: B

Inside application with minimal effort then using API would be correct

upvoted 2 times

DevoteamAnalytix 1 year, 1 month ago

Selected Answer: B

"The Amazon Redshift Data API enables you to painlessly access data from Amazon Redshift with all types of traditional, cloud-native, and containerized, serverless web service-based applications and event-driven applications."
https://aws.amazon.com/de/blogs/big-data/using-the-amazon-redshift-data-api-to-interact-with-amazon-redshift-clusters/#:~:text=The%20Amazon%20Redshift%20Data%20API%20is%20not%20a%20replacement%20for,supported%20by%20the%20AWS%20SDK.

upvoted 3 times

GiorgioGss 1 year, 3 months ago

Selected Answer: B

Even if you don't know nothing about them, you will still choose B because it seems the "LEAST operational overhead" :)

upvoted 3 times

Alcee 1 year, 4 months ago

B. DATA API

upvoted 1 times

TonyStark0122 1 year, 4 months ago

B. Use the Amazon Redshift Data API.

Explanation:
The Amazon Redshift Data API is a lightweight, HTTPS-based API that provides an alternative to using JDBC or ODBC drivers for running queries against Amazon Redshift. It allows you to execute SQL queries directly from within your application without the need for managing connections or drivers. This reduces operational overhead as there's no need to manage and maintain WebSocket or JDBC connections.

upvoted 4 times

milofficial 1 year, 5 months ago

Selected Answer: B

Real time queries with S3 are obviously BS. B it is:

https://docs.aws.amazon.com/redshift/latest/mgmt/data-api.html

upvoted 4 times

Question #8 *Topic 1*

A company uses Amazon Athena for one-time queries against data that is in Amazon S3. The company has several use cases. The company must implement permission controls to separate query processes and access to query history among users, teams, and applications that are in the same AWS account.
Which solution will meet these requirements?

A. Create an S3 bucket for each use case. Create an S3 bucket policy that grants permissions to appropriate individual IAM users. Apply the S3 bucket policy to the S3 bucket.

B. Create an Athena workgroup for each use case. Apply tags to the workgroup. Create an IAM policy that uses the tags to apply appropriate permissions to the workgroup.

C. Create an IAM role for each use case. Assign appropriate permissions to the role for each use case. Associate the role with Athena.

D. Create an AWS Glue Data Catalog resource policy that grants permissions to appropriate individual IAM users for each use case. Apply the resource policy to the specific tables that Athena uses.

Suggested Answer: *B*

Community vote distribution

B (100%)

milofficial Highly Voted 1 year, 5 months ago

Selected Answer: B

Haha they copied this from the old DA Specialty. It's B

https://docs.aws.amazon.com/athena/latest/ug/user-created-workgroups.html

upvoted 17 times

TonyStark0122 Highly Voted 1 year, 4 months ago

B. Create an Athena workgroup for each use case. Apply tags to the workgroup. Create an IAM policy that uses the tags to apply appropriate permissions to the workgroup.

Explanation:
Athena workgroups allow you to isolate and manage different workloads, users, and permissions. By creating a separate workgroup for each use case, you can control access to query history, manage permissions, and enforce resource usage limits independently for each workload. Applying tags to workgroups allows you to categorize and organize them based on the use case, which simplifies policy management.

upvoted 15 times

Scotty_Nguyen Most Recent 3 months, 1 week ago

Selected Answer: B

B is correct

upvoted 1 times

Manohar24 11 months, 3 weeks ago

Selected Answer: B

B is correct.

upvoted 2 times

k350Secops 1 year, 1 month ago

Selected Answer: B

The only other answer that's confusing is C But its not the one. Creating separate IAM roles for each use case and associating them with Athena would not provide the necessary isolation and access control for query processes and query history.

upvoted 2 times

dev_vicente 1 year, 3 months ago

Selected Answer: B

B is more granular

upvoted 1 times

Question #9 *Topic 1*

A data engineer needs to schedule a workflow that runs a set of AWS Glue jobs every day. The data engineer does not require the Glue jobs to run or finish at a specific time.
Which solution will run the Glue jobs in the MOST cost-effective way?

A. Choose the FLEX execution class in the Glue job properties.

B. Use the Spot Instance type in Glue job properties.

C. Choose the STANDARD execution class in the Glue job properties.

D. Choose the latest version in the GlueVersion field in the Glue job properties.

Suggested Answer: *A*

Community vote distribution

A (100%)

pypelyncar Highly Voted 1 year ago

Selected Answer: A

The FLEX execution class leverages spare capacity within the AWS infrastructure to run Glue jobs at a discounted price compared to the standard execution class. Since the data engineer doesn't have specific time constraints, utilizing spare capacity is ideal for cost savings.
Today's date its a checkbox in order to spare capacity and will mean we dont know when is going to finish, which is recommended to increase a timeout

upvoted 8 times

TonyStark0122 Highly Voted 1 year, 4 months ago

A. Choose the FLEX execution class in the Glue job properties.

Explanation:
The FLEX execution class in AWS Glue allows jobs to use idle resources within the Glue service, which can significantly reduce costs compared to the STANDARD execution class. With FLEX, Glue jobs run when resources are available, which is a cost-effective approach for jobs that don't need to be completed within a specific timeframe.

upvoted 6 times

GabrielSGoncalves Most Recent 11 months, 1 week ago

Selected Answer: A

FLEX is how you lower Glue cost when you dont have urgency to run ETLs.

upvoted 1 times

k350Secops 1 year, 1 month ago

Selected Answer: A

As its said the FLEX job comes cheaper that hiring a spot instance

upvoted 3 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: A

I'd go with A

upvoted 1 times

lalitjhawar 1 year, 5 months ago

A
Flex allows you to optimize your costs on your non-urgent or non-time sensitive data integration workloads such as testing, and one-time data loads. With Flex, AWS Glue jobs run on spare compute capacity instead of dedicated hardware. The start and runtimes of jobs using Flex can vary because spare compute resources aren’t readily available and can be reclaimed during the run of a job

https://aws.amazon.com/blogs/big-data/introducing-aws-glue-flex-jobs-cost-savings-on-etl-workloads/

upvoted 5 times

Question #10 *Topic 1*

A data engineer needs to create an AWS Lambda function that converts the format of data from .csv to Apache Parquet. The Lambda function must run only if a user uploads a .csv file to an Amazon S3 bucket.
Which solution will meet these requirements with the LEAST operational overhead?

A. Create an S3 event notification that has an event type of s3:ObjectCreated:*. Use a filter rule to generate notifications only when the suffix includes .csv. Set the Amazon Resource Name (ARN) of the Lambda function as the destination for the event notification.

B. Create an S3 event notification that has an event type of s3:ObjectTagging:* for objects that have a tag set to .csv. Set the Amazon Resource Name (ARN) of the Lambda function as the destination for the event notification.

C. Create an S3 event notification that has an event type of s3:*. Use a filter rule to generate notifications only when the suffix includes .csv. Set the Amazon Resource Name (ARN) of the Lambda function as the destination for the event notification.

D. Create an S3 event notification that has an event type of s3:ObjectCreated:*. Use a filter rule to generate notifications only when the suffix includes .csv. Set an Amazon Simple Notification Service (Amazon SNS) topic as the destination for the event notification. Subscribe the Lambda function to the SNS topic.

Suggested Answer: *A*

Community vote distribution

A (100%)

milofficial Highly Voted 1 year, 5 months ago

Selected Answer: A

"only if a user uploads data to an Amazon S3 bucket" that excludes B & C because we need s3:ObjectCreated:*

You don't need SNS for S3 event notifications so A is easier.

upvoted 13 times

TonyStark0122 Highly Voted 1 year, 4 months ago

A. Create an S3 event notification that has an event type of s3:ObjectCreated:*. Use a filter rule to generate notifications only when the suffix includes .csv. Set the Amazon Resource Name (ARN) of the Lambda function as the destination for the event notification.

Explanation:
This solution directly triggers the Lambda function only when a .csv file is uploaded to the S3 bucket, minimizing unnecessary invocations of the Lambda function. It uses a specific event type (s3:ObjectCreated:*) and a filter rule to ensure that the Lambda function is invoked only for relevant events. Additionally, it directly invokes the Lambda function without the need for additional services like Amazon SNS, reducing operational overhead.

upvoted 8 times

Adrifersilva Most Recent 9 months ago

Selected Answer: A

s3:ObjectCreated:* instead of s3:*: triggers the Lambda function only when objects are created in the bucket.

upvoted 1 times

theloseralreadytaken 9 months, 1 week ago

Selected Answer: A

A is the answer for least operational. C also correct!

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: A

since is the least operational, the D its a candidate, however add a SNS operation, which in this case is not needed. so A includes S3 and triggering towards the lambda function. 2 services.

upvoted 1 times

k350Secops 1 year, 1 month ago

Selected Answer: A

S3 event notification to lamba for file prefix with.csv is the least overhead way

upvoted 1 times

DevoteamAnalytix 1 year, 1 month ago

Selected Answer: A

"You can use Lambda to process event notifications from Amazon Simple Storage Service. Amazon S3 can send an event to a Lambda function when an object is created or deleted"

https://docs.aws.amazon.com/lambda/latest/dg/with-s3.html

upvoted 2 times

Question #11 *Topic 1*

A data engineer needs Amazon Athena queries to finish faster. The data engineer notices that all the files the Athena queries use are currently stored in uncompressed .csv format. The data engineer also notices that users perform most queries by selecting a specific column.
Which solution will MOST speed up the Athena query performance?

A. Change the data format from .csv to JSON format. Apply Snappy compression.

B. Compress the .csv files by using Snappy compression.

C. Change the data format from .csv to Apache Parquet. Apply Snappy compression.

D. Compress the .csv files by using gzip compression.

Suggested Answer: *C*

Community vote distribution

C (100%)

⊟ **milofficial** Highly Voted 1 year, 5 months ago

Selected Answer: C

If the exam would only have these kinds of questions everyone would be blessed

upvoted 11 times

⊟ **[Removed]** 1 year, 5 months ago

Hahahaha! I believe that this kind of question is only for the beta calibration purpose. They won't be in the final exam version.

upvoted 1 times

⊟ **TonyStark0122** Highly Voted 1 year, 4 months ago

C. Change the data format from .csv to Apache Parquet. Apply Snappy compression.

Explanation:
Apache Parquet is a columnar storage format optimized for analytical queries. It is highly efficient for query performance, especially when queries involve selecting specific columns, as it allows for column pruning and predicate pushdown optimizations.

upvoted 6 times

⊟ **Scotty_Nguyen** Most Recent 3 months, 1 week ago

Selected Answer: C

C is correct

upvoted 1 times

⊟ **GabrielSGoncalves** 11 months, 1 week ago

Selected Answer: C

C is the way to do It based on best practices recommended by AWS (https://aws.amazon.com/pt/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/)

upvoted 1 times

⊟ **hnk** 1 year, 1 month ago

Selected Answer: C

C is correct

upvoted 1 times

⊟ **k350Secops** 1 year, 1 month ago

Selected Answer: C

switching to Apache Parquet format with Snappy compression offers the most significant improvement in Athena query performance, especially for queries that select specific columns

upvoted 1 times

⊟ **d8945a1** 1 year, 1 month ago

Selected Answer: C

Parquet is columnar storage and the question specifies that users performs most queries by selecting a specific column.

upvoted 1 times

⊟ **wa212** 1 year, 2 months ago

Selected Answer: C

https://aws.amazon.com/jp/blogs/news/top-10-performance-tuning-tips-for-amazon-athena/

upvoted 2 times

Alcee 1 year, 4 months ago

C easy

upvoted 1 times

Question #12 *Topic 1*

A manufacturing company collects sensor data from its factory floor to monitor and enhance operational efficiency. The company uses Amazon Kinesis Data Streams to publish the data that the sensors collect to a data stream. Then Amazon Kinesis Data Firehose writes the data to an Amazon S3 bucket.

The company needs to display a real-time view of operational efficiency on a large screen in the manufacturing facility.

Which solution will meet these requirements with the LOWEST latency?

A. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to process the sensor data. Use a connector for Apache Flink to write data to an Amazon Timestream database. Use the Timestream database as a source to create a Grafana dashboard.

B. Configure the S3 bucket to send a notification to an AWS Lambda function when any new object is created. Use the Lambda function to publish the data to Amazon Aurora. Use Aurora as a source to create an Amazon QuickSight dashboard.

C. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to process the sensor data. Create a new Data Firehose delivery stream to publish data directly to an Amazon Timestream database. Use the Timestream database as a source to create an Amazon QuickSight dashboard.

D. Use AWS Glue bookmarks to read sensor data from the S3 bucket in real time. Publish the data to an Amazon Timestream database. Use the Timestream database as a source to create a Grafana dashboard.

Suggested Answer: *A*

Community vote distribution

A (96%) 4%

fceb2c1 Highly Voted 1 year, 3 months ago

Selected Answer: A

https://aws.amazon.com/blogs/database/near-real-time-processing-with-amazon-kinesis-amazon-timestream-and-grafana/

Look at the architecture diagram

upvoted 13 times

V0811 10 months, 4 weeks ago

No! :-)

Because: The company needs to display a REAL-TIME view of operational efficiency on a large screen in the manufacturing facility.

So it's for sure C

upvoted 1 times

Udyan 5 months, 2 weeks ago

V0811 - and new firehose, ofc decrease latency, khomon man

upvoted 1 times

milofficial Highly Voted 1 year, 5 months ago

Selected Answer: A

real time -> no Quicksight. And bookmarks to read sensor data real time is just as stupid as the flat earth theory. A it is.

upvoted 7 times

Scotty_Nguyen Most Recent 3 months ago

Selected Answer: A

A is correct

upvoted 1 times

Adrifersilva 9 months ago

Selected Answer: A

Grafana:

Real-time Performance:

Grafana is known for its excellent real-time data visualization capabilities.

It's often used for operational dashboards that require frequent updates.

Integration:

Works well with time-series databases and streaming data sources. [2]

upvoted 3 times

deepcloud 10 months, 2 weeks ago

Selected Answer: A

Firehose cannot use Timestream as destination. Answer is A

upvoted 7 times

samadal 10 months, 2 weeks ago

Option A is for processing data in Flink and then sending it to Timestream. This is advantageous when complex data processing is required in Flink, but the processing step where complex analytics are processed can handle additional latency.

Option C performs data processing in Flink, sends the data directly to Timestream without any additional steps, and provides dashboards via QuickSight. Since data can be started immediately after arriving in Timestream, latency is likely to be higher.

Therefore, option C is preferable because it can handle latency by performing data processing, publishing data directly to Timestream, and provides fast dashboards using QuickSight.

upvoted 1 times

teo2157 10 months, 2 weeks ago

Selected Answer: A

Amazon QuickSight is primarily designed for business intelligence and data visualization, and it can provide near real-time views depending on the data refresh rate. However, it is not typically used for real-time streaming data visualization with very low latency. For real-time dashboards with very low latency, services like Grafana are more suitable.

You can use Amazon Managed Grafana to setup the dashboard so you're using an AWS service which is always preferible on these exams.

upvoted 4 times

V0811 10 months, 4 weeks ago

Selected Answer: C

Because: The company needs to display a real-time view of operational efficiency on a large screen in the manufacturing facility.

upvoted 1 times

jyrajan69 11 months ago

Based on this it should be C, why use an open source app when you can an AWS Service

https://community.amazonquicksight.com/t/real-time-data-visualization-capabilities-of-amazon-quicksight/24007

upvoted 2 times

Just_Ninja 1 year, 1 month ago

The Question is: Which solution will meet these requirements with the LOWEST latency?

So just A can be the right answer "lowest latency!!!!"

upvoted 1 times

LanoraMoe 1 year, 2 months ago

I go with Option A. Kinesis Data Firehose can connect to 3 AWS destinations so far S3, Redshift and OpenSearch.

upvoted 1 times

certplan 1 year, 3 months ago

Option A:

- Involves additional steps: Option A requires writing data to Amazon Timestream after processing with Apache Flink, potentially introducing additional latency compared to a more direct approach like Option C.
- Grafana integration: While Grafana is a powerful visualization tool, setting up and configuring Grafana dashboards might require additional effort compared to using Amazon QuickSight, which offers more straightforward integration with AWS services like Amazon Timestream.

upvoted 1 times

certplan 1 year, 3 months ago

C. - **Processing Sensor Data with Amazon Flink**: Similar to option A, this approach uses Amazon Managed Service for Apache Flink to process sensor data, providing real-time analytics or transformation capabilities.

- **Data Firehose Delivery Stream to Timestream**: Sets up a new Amazon Data Firehose delivery stream to publish processed data directly to Amazon Timestream. Data Firehose is a fully managed service for delivering real-time streaming data to destinations such as data lakes, databases, and analytics services.

- **Timestream Database as a Source for QuickSight Dashboard**: Similar to option B, the data stored in Amazon Timestream serves as the data source for creating an Amazon QuickSight dashboard.

upvoted 2 times

⊟ **certplan** 1 year, 3 months ago

A. - **Processing Sensor Data**: Utilizes Amazon Managed Service for Apache Flink, a fully managed service for real-time data processing. This service is used to process the sensor data, which likely involves real-time analysis or transformation of incoming data streams.

- **Connector for Apache Flink to Amazon Timestream**: Integrates a connector for Apache Flink to write processed data into Amazon Timestream, a fully managed time-series database. Timestream is optimized for IoT and time-series data.

- **Timestream Database as a Source for Grafana Dashboard**: The data stored in Timestream serves as the data source for creating a Grafana dashboard. Grafana is a popular open-source analytics and monitoring platform that visualizes time-series data.

upvoted 1 times

⊟ **certplan** 1 year, 3 months ago

Considerations:

Option A utilizes Amazon Managed Service for Apache Flink to process sensor data and then writes the processed data to Amazon Timestream. From there, the Timestream database serves as a source to create a Grafana dashboard.

Thus the data goes through Apache Flink for processing, then to Timestream, and finally to Grafana. "Each additional step introduces potential latency".

Option C processes sensor data using Amazon Managed Service for Apache Flink and then publishes data directly to Amazon Timestream via a Data Firehose delivery stream. Finally, it uses Timestream as a source to create an Amazon QuickSight dashboard.

So, in terms of latency, both options involve processing data in real-time using Apache Flink. However, Option C has a more direct data flow by publishing data directly to Timestream, potentially reducing latency compared to Option A, where the data has to go through an additional step of writing to Timestream.

upvoted 1 times

⊟ **TonyStark0122** 1 year, 4 months ago

A. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to process the sensor data. Use a connector for Apache Flink to write data to an Amazon Timestream database. Use the Timestream database as a source to create a Grafana dashboard.

Explanation:

Amazon Managed Service for Apache Flink provides real-time stream processing capabilities, which can process sensor data with low latency. By using Apache Flink connectors, the processed data can be efficiently written to Amazon Timestream, which is optimized for time-series data storage and querying.

upvoted 2 times

Question #13 *Topic 1*

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. Create a daily schedule to run the crawler. Specify a database name for the output.

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. Allocate data processing units (DPUs) to run the crawler every day. Configure the output destination to a new path in the existing S3 bucket.

Suggested Answer: *B*

Community vote distribution

B (100%)

TonyStark0122 Highly Voted 9 months, 1 week ago

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. Create a daily schedule to run the crawler. Specify a database name for the output.

Explanation:
Option B correctly sets up the IAM role with the necessary permissions using the AWSGlueServiceRole policy, which is designed for use with AWS Glue. It specifies the S3 bucket path of the source data as the crawler's data store and creates a daily schedule to run the crawler. Additionally, it specifies a database name for the output, ensuring that the crawled data is properly cataloged in the AWS Glue Data Catalog.

upvoted 10 times

GiorgioGss Highly Voted 1 year, 3 months ago

Selected Answer: B

A,C are wrong because you use don't need full S3 access. D is wrong because you don't need to provision DPU and the destination should be a database, not an s3 bucket. so it's B

upvoted 6 times

plutonash Most Recent 5 months, 2 weeks ago

Selected Answer: B

answer B is incomplete. Even we include AWSGlueServiceRole policy on IAM role, S3 access is not garantee

upvoted 2 times

LrdKanien 8 months ago

How does Glue get access to S3 if you don't do B?

upvoted 1 times

LrdKanien 8 months ago

I meant A

upvoted 1 times

Asmunk 8 months ago

S3 access is part of the AWSGlueServiceRole Policy
https://docs.aws.amazon.com/aws-managed-policy/latest/reference/AWSGlueServiceRole.html

upvoted 1 times

⊟ **sam_pre** 3 months, 1 week ago

It adds only for the glue related buckets, but it doesnt grant permissions for S3 that we need to read in order to fetch data, isnt it ?

upvoted 1 times

⊟ **k350Secops** 1 year, 1 month ago

Selected Answer: B

Glue Crawlers are serverless. Assigning DPUs is the point where i decided it option B

upvoted 4 times

Question #14 *Topic 1*

A company loads transaction data for each day into Amazon Redshift tables at the end of each day. The company wants to have the ability to track which tables have been loaded and which tables still need to be loaded.
A data engineer wants to store the load statuses of Redshift tables in an Amazon DynamoDB table. The data engineer creates an AWS Lambda function to publish the details of the load statuses to DynamoDB.
How should the data engineer invoke the Lambda function to write load statuses to the DynamoDB table?

A. Use a second Lambda function to invoke the first Lambda function based on Amazon CloudWatch events.

B. Use the Amazon Redshift Data API to publish an event to Amazon EventBridge. Configure an EventBridge rule to invoke the Lambda function.

C. Use the Amazon Redshift Data API to publish a message to an Amazon Simple Queue Service (Amazon SQS) queue. Configure the SQS queue to invoke the Lambda function.

D. Use a second Lambda function to invoke the first Lambda function based on AWS CloudTrail events.

Suggested Answer: *B*

Community vote distribution

B (92%) 8%

milofficial Highly Voted 1 year, 5 months ago

Selected Answer: B

https://docs.aws.amazon.com/redshift/latest/mgmt/data-api-monitoring-events.html

upvoted 14 times

TonyStark0122 Highly Voted 1 year, 4 months ago

The most appropriate way for the data engineer to invoke the Lambda function to write load statuses to the DynamoDB table is:

B. Use the Amazon Redshift Data API to publish an event to Amazon EventBridge. Configure an EventBridge rule to invoke the Lambda function.

Explanation:
Option B leverages the Amazon Redshift Data API to publish events to Amazon EventBridge, which provides a serverless event bus service for handling events across AWS services. By configuring an EventBridge rule to invoke the Lambda function in response to events published by the Redshift Data API, the data engineer can ensure that the Lambda function is triggered whenever there is a new transaction data load in Amazon Redshift. This approach offers a straightforward and scalable solution for tracking table load statuses without relying on additional Lambda functions or services.

upvoted 10 times

MephiboshethGumani Most Recent 1 month, 2 weeks ago

Selected Answer: B

Here's why Option B is best:

Amazon Redshift Data API can be used by applications and scripts to interact with Redshift (e.g., run SQL queries, check load status).

Amazon EventBridge can receive custom events or service-generated events and route them to targets like Lambda.

This approach decouples the data load process from status logging, using EventBridge as a clean integration point.

EventBridge rule can filter and trigger the Lambda function when Redshift (or your data pipeline) signals a successful data load.

upvoted 1 times

MephiboshethGumani 3 months, 2 weeks ago

Selected Answer: B

the data engineer should use Amazon EventBridge (formerly CloudWatch Events) to trigger the Lambda function based on a schedule or events that correspond to the completion of the data load process in Amazon Redshift.

upvoted 1 times

altonh 6 months, 4 weeks ago

Selected Answer: D

The statement in B is inaccurate.

You don't 'use Amazon Redshift Data API to publish' event to EventBridge. Redshift Data API has no function to write to EventBridge. Instead, the statement should be "Use EventBridge to monitor Data API events..." Perhaps this is a typo.

But if I assume there are no typos in all the statements, then I would go for D. Although not a perfect solution, the cloud trail events have more info than the Redshift Data API events.

upvoted 2 times

taxo 10 months, 2 weeks ago

This job doesn’t need a real time check

upvoted 1 times

John2025 1 year ago

Why not used SQS to keep API change in the Queue ?

upvoted 2 times

pypelyncar 1 year ago

Im not 100% sure of B or C, this is a tricky question. The reason is due to either SQS or EventBridge has not direct connection natively speaking to Redshift Data API. There is no way to publish events by itself. So, this means either SQS / EventBridge eventually need a "proxy" (e.g lambda function) in order to publish events or process events to this 2 sources. In both services we need something to publish those events from Redshift. so Yes, we need a lamda function between Redshift Data API and (SQS|EB). so either B,C doesnt seem to be 100% right. I think this question its a good candidate to be "Choose two options" but none has 100% right. Both are valid considering that there is an adapter function between 2 solutions.

upvoted 1 times

San_Juan 10 months, 2 weeks ago

It seems that Redshift Data API could directly publishing events in EventBridge (see first comment). For monitoring the Redshift Data API, you could use both EventBridge (near-real-time) or CloudTrail (stored in S3): https://docs.aws.amazon.com/redshift/latest/mgmt/data-api-monitoring.html

But both services are related to "Data API" not Redshift database itself. So it is really tricky.

upvoted 1 times

San_Juan 10 months, 2 weeks ago

So, you could use the Redshift table STV_LOAD_STATE,
https://docs.aws.amazon.com/redshift/latest/dg/r_STV_LOAD_STATE.html
and running a "select" query on that table for getting status of tables (filtering by timestamp) and add the result to EventBridge, applying a rule on those events to invoke the lambda function. I guess that B is the most appropiate answer.

upvoted 1 times

Question #15 *Topic 1*

A data engineer needs to securely transfer 5 TB of data from an on-premises data center to an Amazon S3 bucket. Approximately 5% of the data changes every day. Updates to the data need to be regularly proliferated to the S3 bucket. The data includes files that are in multiple formats. The data engineer needs to automate the transfer process and must schedule the process to run periodically.
Which AWS service should the data engineer use to transfer the data in the MOST operationally efficient way?

A. AWS DataSync

B. AWS Glue

C. AWS Direct Connect

D. Amazon S3 Transfer Acceleration

Suggested Answer: *A*

Community vote distribution

A (100%)

TonyStark0122 Highly Voted 1 year, 4 months ago

A. AWS DataSync

Explanation:
AWS DataSync is a managed data transfer service that simplifies and accelerates moving large amounts of data online between on-premises storage and Amazon S3, EFS, or FSx for Windows File Server. DataSync is optimized for efficient, incremental, and reliable transfers of large datasets, making it suitable for transferring 5 TB of data with daily updates.

upvoted 12 times

deepbro Most Recent 3 weeks, 5 days ago

Selected Answer: A

Data Sync all the way

upvoted 1 times

Scotty_Nguyen 2 months ago

Selected Answer: A

AWS DataSync:
Purpose: AWS DataSync is designed for automated, efficient, and secure data transfer between on-premises storage and AWS storage services like Amazon S3.
Key Features:
Automates and schedules data transfers, supporting periodic syncs.
Handles incremental transfers, only copying changed or new files (ideal for the 5% daily changes).
Supports multiple file formats and preserves metadata.
Provides encryption for secure transfers.
Scales to handle large datasets like 5 TB and optimizes transfer performance.
Why it fits: DataSync meets all requirements by automating periodic transfers, efficiently handling incremental updates, and supporting diverse file formats with minimal operational overhead.

upvoted 2 times

sam_pre 3 months, 1 week ago

Selected Answer: A

DataSync perfectly fit for this requirement

upvoted 2 times

San_Juan 10 months, 2 weeks ago

Aseems correct.

AWS Direct Connect is a networking service, nothing to be realted to sync data between on-premises and cloud storage services, as DataSync does (" online service that automates and accelerates moving data between on premises and AWS Storage services.").

upvoted 2 times

pypelyncar 1 year ago

Selected Answer: A

DataSync, locations, tasks, is all what you need.

upvoted 2 times

FunkyFresco 1 year, 1 month ago

Selected Answer: A

is datasync

upvoted 1 times

augustino0890 1 year, 2 months ago

A. AWS DataSync

AWS DataSync is a data transfer service specifically designed to simplify and accelerate moving large volumes of data between on-premises storage systems and AWS storage services like S3.

upvoted 2 times

KelvinPun 1 year, 2 months ago

Selected Answer: A

That's the job of DataSync

upvoted 1 times

Rafaaws 1 year, 2 months ago

A - DataSync is build for this use case

upvoted 1 times

milofficial 1 year, 5 months ago

Selected Answer: A

Typical DataSync use case

upvoted 2 times

Question #16 *Topic 1*

A company uses an on-premises Microsoft SQL Server database to store financial transaction data. The company migrates the transaction data from the on-premises database to AWS at the end of each month. The company has noticed that the cost to migrate data from the on-premises database to an Amazon RDS for SQL Server database has increased recently.
The company requires a cost-effective solution to migrate the data to AWS. The solution must cause minimal downtown for the applications that access the database.
Which AWS service should the company use to meet these requirements?

A. AWS Lambda

B. AWS Database Migration Service (AWS DMS)

C. AWS Direct Connect

D. AWS DataSync

Suggested Answer: *B*

Community vote distribution

B (100%)

milofficial Highly Voted 1 year, 5 months ago

Selected Answer: B

Whoever is the admin that pre-marks the answers, it's time to go

upvoted 22 times

TonyStark0122 Highly Voted 1 year, 4 months ago

B. AWS Database Migration Service (AWS DMS)

Explanation:
AWS Database Migration Service (DMS) is specifically designed for migrating data from various sources, including on-premises databases, to AWS with minimal downtime and disruption to applications. It supports homogeneous migrations (e.g., SQL Server to SQL Server) as well as heterogeneous migrations (e.g., SQL Server to Amazon RDS for SQL Server).

upvoted 13 times

shammous Most Recent 9 months, 3 weeks ago

Hahaha, I loved the "downtown" typo in the question. I always say the same instead of "downtime"..

upvoted 1 times

San_Juan 10 months, 2 weeks ago

D could be OK.
I mean, it is talking about migrating to a cloud database and switching-off the current on-premise database. So you could use AWS Snowball Edge Storage to move the backup of the on-premises database, and when it is in the edge storage, copy the data to a new cloud-based SQL server instance using AWS DataSync

https://aws.amazon.com/es/blogs/storage/seamlessly-migrate-large-sql-databases-using-aws-snowball-and-aws-datasync/

upvoted 2 times

shammous 9 months, 3 weeks ago

Right, but Snowball isn't mentioned in Option D. Thus, you can't consider it as OK.

upvoted 3 times

pypelyncar 1 year ago

Selected Answer: B

AWS DMS offers a cost-effective solution for database migrations compared to replicating data to a fully managed RDS instance.
You only pay for the resources used during the migration, making it ideal for infrequent, monthly transfers

upvoted 2 times

xaocho 1 year, 2 months ago

AWS Database Migration Service (DMS)

upvoted 1 times

lucas_rfsb 1 year, 3 months ago

Selected Answer: B

B Since it's for Migration porpouse, typical for DMS

upvoted 2 times

Question #17 *Topic 1*

A data engineer is building a data pipeline on AWS by using AWS Glue extract, transform, and load (ETL) jobs. The data engineer needs to process data from Amazon RDS and MongoDB, perform transformations, and load the transformed data into Amazon Redshift for analytics. The data updates must occur every hour.
Which combination of tasks will meet these requirements with the LEAST operational overhead? (Choose two.)

A. Configure AWS Glue triggers to run the ETL jobs every hour.

B. Use AWS Glue DataBrew to clean and prepare the data for analytics.

C. Use AWS Lambda functions to schedule and run the ETL jobs every hour.

D. Use AWS Glue connections to establish connectivity between the data sources and Amazon Redshift.

E. Use the Redshift Data API to load transformed data into Amazon Redshift.

Suggested Answer: *AD*

Community vote distribution

AD (82%) Other

rralucard_ Highly Voted 1 year, 5 months ago

Selected Answer: AD

AWS Glue triggers provide a simple and integrated way to schedule ETL jobs. By configuring these triggers to run hourly, the data engineer can ensure that the data processing and updates occur as required without the need for external scheduling tools or custom scripts. This approach is directly integrated with AWS Glue, reducing the complexity and operational overhead.
AWS Glue supports connections to various data sources, including Amazon RDS and MongoDB. By using AWS Glue connections, the data engineer can easily configure and manage the connectivity between these data sources and Amazon Redshift. This method leverages AWS Glue's built-in capabilities for data source integration, thus minimizing operational complexity and ensuring a seamless data flow from the sources to the destination (Amazon Redshift).

upvoted 7 times

pypelyncar Highly Voted 1 year ago

Selected Answer: AD

A. Configure AWS Glue triggers to run the ETL jobs every hour.
Reduced Code Complexity: Glue triggers eliminate the need to write custom code for scheduling ETL jobs. This simplifies the pipeline and reduces maintenance overhead.
Scalability and Integration: Glue triggers work seamlessly with Glue ETL jobs, ensuring efficient scheduling and execution within the Glue ecosystem.
D. Use AWS Glue connections to establish connectivity between the data sources and Amazon Redshift.
Pre-Built Connectors: Glue connections offer pre-built connectors for various data sources like RDS and Redshift. This eliminates the need for manual configuration and simplifies data source access within the ETL jobs.
Centralized Management: Glue connections are centrally managed within the Glue service, streamlining connection management and reducing operational overhead.

upvoted 6 times

saransh_001 Most Recent 4 months, 2 weeks ago

Selected Answer: AD

A. AWS Glue provides a built-in mechanism to trigger ETL jobs at scheduled intervals, such as every hour. Using Glue triggers minimizes the need for additional custom code or services, reducing operational overhead.
D. AWS Glue connections simplify the process of establishing secure and reliable connections to various data sources (Amazon RDS, MongoDB) and the destination (Amazon Redshift). This approach reduces the need for manually configuring connection settings and makes the ETL pipeline easier to maintain.

upvoted 2 times

San_Juan 9 months, 1 week ago

Selected Answer: AC
A. because the question is saying that the jobs are build in Glue, and must run every hour.
C. because you can run the jobs as Lambda functions every hour.

B. discarded, because the question is saying that "DE" is using Glue, DataBrew is for cleaning data without code, but it seems that the "DE" is writing code for transforming the data.

D. Discarted, because the connections are not directly related to the question, that it is saying that you should run every hour Glue jobs, and the connections doesn't seem relevant.

E. Discarted, because is saying that the data source is RDS and MongoDB, not Redshift, so you cannot use the Redshift Data API for getting the data and transform it.

upvoted 1 times

sachin 10 months, 3 weeks ago

AE

D is not valid. as it shoyld be

Use AWS Glue connections to establish connectivity between the data sources (including Amazon Redshift) and Glue Job

upvoted 1 times

samadal 10 months, 2 weeks ago

An AWS Glue connection is a setting that allows an AWS Glue job to access a data source. This allows you to connect to databases such as RDS, MongoDB, etc. However, this opinion states that this connection is not used to load data directly into Redshift, and that Glue jobs must use the COPY command to load data into Redshift, which is inappropriate. However, since Glue jobs can process data and load it directly into Redshift, it is a bit of a stretch to consider option D as unconditionally wrong.

upvoted 1 times

DevoteamAnalytix 1 year, 1 month ago

Selected Answer: AD

I was not sure about A - But in AWS console => Glue => Triggers => Add Trigger I have found the Trigger type: "Schedule - Fire the trigger on a timer."

upvoted 3 times

lucas_rfsb 1 year, 3 months ago

Selected Answer: CD

I found this question actually confusing. In which step the transformation would be implemented itself? I can be wrong, but with Glue triggers we would only run the job, but not the transformation logic itself. In this way, I would go in C and D

upvoted 1 times

milofficial 1 year, 3 months ago

Selected Answer: AD

Not a clear question - B would kinda make sense - but AD seems to be more correct

upvoted 3 times

GiorgioGss 1 year, 3 months ago

Selected Answer: AD

A - this is obvious and D -https://docs.aws.amazon.com/glue/latest/dg/console-connections.html

upvoted 4 times

TonyStark0122 1 year, 4 months ago

A. Configure AWS Glue triggers to run the ETL jobs every hour.

D. Use AWS Glue connections to establish connectivity between the data sources and Amazon Redshift.

Explanation:

Option A: Configuring AWS Glue triggers allows the ETL jobs to be scheduled and run automatically every hour without the need for manual intervention. This reduces operational overhead by automating the data processing pipeline.

Option D: Using AWS Glue connections simplifies connectivity between the data sources (Amazon RDS and MongoDB) and Amazon Redshift. Glue connections abstract away the details of connection configuration, making it easier to manage and maintain the data pipeline.

upvoted 3 times

milofficial 1 year, 5 months ago

Selected Answer: AB

Lambda triggers for Glue jobs make me dizzy

upvoted 2 times

Question #18 *Topic 1*

A company uses an Amazon Redshift cluster that runs on RA3 nodes. The company wants to scale read and write capacity to meet demand. A data engineer needs to identify a solution that will turn on concurrency scaling.
Which solution will meet this requirement?

A. Turn on concurrency scaling in workload management (WLM) for Redshift Serverless workgroups.

B. Turn on concurrency scaling at the workload management (WLM) queue level in the Redshift cluster.

C. Turn on concurrency scaling in the settings during the creation of any new Redshift cluster.

D. Turn on concurrency scaling for the daily usage quota for the Redshift cluster.

Suggested Answer: *B*

Community vote distribution

B (100%)

TonyStark0122 Highly Voted 1 year, 4 months ago
B. Turn on concurrency scaling at the workload management (WLM) queue level in the Redshift cluster.

Explanation:
Concurrency scaling in Amazon Redshift allows the cluster to automatically add and remove compute resources in response to workload demands. Enabling concurrency scaling at the workload management (WLM) queue level allows you to specify which queues can benefit from concurrency scaling based on the query workload.
upvoted 9 times

saransh_001 Most Recent 4 months, 2 weeks ago
Selected Answer: B
Concurrency Scaling in Amazon Redshift is a feature that automatically adds temporary clusters to handle spikes in query traffic, providing additional read and write capacity.
This feature is enabled through Workload Management (WLM) at the queue level in Redshift. Each queue can be configured to use concurrency scaling for handling queries that exceed the capacity of the main cluster.
Why option A is incorrect:
Turn on concurrency scaling in workload management (WLM) for Redshift Serverless workgroups: This option is for Redshift Serverless rather than clusters on RA3 nodes. Serverless clusters handle scaling differently and don't require manual concurrency scaling settings like the RA3 clusters.
upvoted 4 times

lsj900605 7 months, 3 weeks ago
B"You can manage which queries are sent to the concurrency-scaling cluster by configuring WLM queues. You're charged for concurrency-scaling clusters only for the time they're actively running queries."
https://docs.aws.amazon.com/redshift/latest/dg/concurrency-scaling.html
upvoted 1 times

San_Juan 10 months, 1 week ago
Selected answer: B
B. According to documentation, the "concurrency scaling" is set up in workload management queue (see comment below).

A. discarted, because redshift serverless scales automatically (it doesn't need enable "concurrency scaling").
upvoted 2 times

d8945a1 1 year, 1 month ago
Selected Answer: B
B. Turn on concurrency scaling at the workload management (WLM) queue level in the Redshift cluster.
upvoted 1 times

khchan123 1 year, 2 months ago
Selected Answer: B
Answer is B.
B. Turn on concurrency scaling at the workload management (WLM) queue level in the Redshift cluster.
upvoted 1 times

milofficial 1 year, 5 months ago

Selected Answer: B

https://docs.aws.amazon.com/redshift/latest/dg/concurrency-scaling-queues.html

upvoted 4 times

Question #19 *Topic 1*

A data engineer must orchestrate a series of Amazon Athena queries that will run every day. Each query can run for more than 15 minutes.
Which combination of steps will meet these requirements MOST cost-effectively? (Choose two.)

A. Use an AWS Lambda function and the Athena Boto3 client start_query_execution API call to invoke the Athena queries programmatically.

B. Create an AWS Step Functions workflow and add two states. Add the first state before the Lambda function. Configure the second state as a Wait state to periodically check whether the Athena query has finished using the Athena Boto3 get_query_execution API call. Configure the workflow to invoke the next query when the current query has finished running.

C. Use an AWS Glue Python shell job and the Athena Boto3 client start_query_execution API call to invoke the Athena queries programmatically.

D. Use an AWS Glue Python shell script to run a sleep timer that checks every 5 minutes to determine whether the current Athena query has finished running successfully. Configure the Python shell script to invoke the next query when the current query has finished running.

E. Use Amazon Managed Workflows for Apache Airflow (Amazon MWAA) to orchestrate the Athena queries in AWS Batch.

Suggested Answer: *AB*

Community vote distribution

AB (67%) BE (20%) 7%

rralucard_ Highly Voted 1 year, 4 months ago

Selected Answer: AB

AWS Lambda can be effectively used to trigger Athena queries. By using the start_query_execution API from the Athena Boto3 client, you can programmatically start Athena queries. Lambda functions are cost-effective as they charge based on the compute time used, and there's no charge when the code is not running. However, Lambda has a maximum execution timeout of 15 minutes, which means it's not suitable for long-running operations but can be used to trigger or start queries.
AWS Step Functions can orchestrate multiple AWS services in workflows. By using a Wait state, the workflow can periodically check the status of the Athena query, and proceed to the next step once the query is complete. This approach is more scalable and reliable compared to continuously running a Lambda function, as Step Functions can handle long-running processes better and can maintain the state of each step in the workflow.

upvoted 10 times

San_Juan 10 months, 1 week ago

Lambda max timeout is 15 minutes, and the query takes more than 15 minutes. So Lambda should be ended prior the Athena query.

upvoted 4 times

GiorgioGss Highly Voted 1 year, 3 months ago

Selected Answer: BE

B - because
https://docs.aws.amazon.com/step-functions/latest/dg/sample-athena-query.html
E - because
https://aws.amazon.com/blogs/big-data/orchestrate-amazon-emr-serverless-spark-jobs-with-amazon-mwaa-and-data-validation-using-amazon-athena/

upvoted 9 times

San_Juan 10 months, 1 week ago

I discarted E because Airflow is more expensive than Glue/Step-Functions. So B (step-function) and D (glue python shell).

upvoted 2 times

DevoteamAnalytix 1 year, 1 month ago

The question is about a "combination of steps" - MWAA and Step Functions are different options, so I would prefer AB

upvoted 3 times

sachin 10 months, 3 weeks ago

BE is right. A is only giving option to envoke the athena query. how about the response. if the execution is beyong 15 mins

upvoted 1 times

Evan_Lin Most Recent 4 months, 3 weeks ago

Selected Answer: AB

After real-world testing, A is a valid answer. This is because the Lambda only sends the API request to Athena, which runs the query. Even if the Lambda times out, the query result is still stored in the designated S3 bucket.

upvoted 2 times

Udyan 5 months, 2 weeks ago

Selected Answer: AB

Why?

B (Step Functions): Step Functions are ideal for orchestrating long-running workflows, including polling the Athena query status and invoking the next query when ready.

A (Lambda): Lambda is used to programmatically trigger Athena queries within Step Functions, despite its 15-minute limitation, because Step Functions can manage the long runtime using Wait states.

Why Not C, D, or E?

C and D involve Glue, which is better suited for ETL jobs than orchestration, making them less efficient and cost-effective.

E (Amazon MWAA) introduces unnecessary cost and complexity for a straightforward workflow.

upvoted 2 times

haby 6 months, 2 weeks ago

Selected Answer: BC

BC for me

A - lambda function will stop at 900s, so it will stop before query finishes(more than 15mins)

E - Airflow is way more complex and expensive than step function

upvoted 4 times

altonh 6 months, 4 weeks ago

Selected Answer: CE

AB - Because of the Lambda timeout

CE—is correct. The query will be executed by a glue job, which will be orchestrated by Airflow. The job will be scheduled using AWS Batch.

upvoted 1 times

Eleftheriia 7 months ago

Selected Answer: AB

Not E because "You should use Step Functions if you prioritize cost and performance"

https://aws.amazon.com/managed-workflows-for-apache-airflow/faqs/

And also the fact that the queries take longer than 15 min can be handled with step functions, therefore AB

upvoted 2 times

truongnguyen86 7 months, 2 weeks ago

A.Why it's correct: AWS Lambda is a cost-effective, serverless option for invoking Athena queries using the Boto3 API. Lambda charges are based on execution time and memory usage, making it an efficient solution for periodic query execution.

B. Why it's correct: Step Functions provide a serverless orchestration option with a pay-per-use pricing model. Adding a Wait state prevents excessive API calls and ensures queries are executed in sequence, making it a cost-effective and scalable solution.

Why the other options are less optimal:

--

E. Use Amazon Managed Workflows for Apache Airflow (MWAA): MWAA is powerful for complex workflows, but its pricing includes environment uptime costs, which can be higher than Lambda and Step Functions for simple tasks like orchestrating Athena queries.

By choosing A and B, you balance cost-effectiveness and simplicity for orchestrating daily Athena queries.

upvoted 1 times

San_Juan 10 months, 1 week ago

Selected Answer: BD

Lambda maximum timeout is 15 minutes. So the query takes more than Lambda could manage. So you cannot use lambda. Use Step-Function (answer B) or glue python (answer D)

Airflow is more expensive than Glue/Step-Functions, so E is discarted also.

upvoted 2 times

V0811 10 months, 4 weeks ago

Selected Answer: AB

It should be AB

upvoted 2 times

alex1991 1 year ago

Selected Answer: AB

Since the Athena API supports async/await, users are able to separate the steps into trigger queries and get results after 15 minutes.

upvoted 2 times

pypelyncar 1 year ago

Selected Answer: BE

tricky, A is valid. Still, cost effective:

B no one doubt on it. then why E?

MWAA offers a managed Apache Airflow environment for orchestrating complex workflows.

It can handle long-running tasks like Athena queries efficiently.

Batch Processing: Leveraging AWS Batch within the Airflow workflow allows for distributed and scalable execution of the Athena queries, improving overall processing efficiency.

upvoted 1 times

> **San_Juan** 10 months, 1 week ago
>
> A could be not valid, as queries takes more than 15 minutes, and Lambda maximum timeout is 15 minutes. Lambda would be ended prior than the query is finished.
>
> upvoted 1 times
>
> > **JoeAWSOCM** 6 months, 4 weeks ago
> >
> > Lambda is just for triggering the query. Its not waiting for the query to finish. The status of the query will be checked using Step functions.
> >
> > upvoted 1 times

valuedate 1 year, 1 month ago

Selected Answer: AB

my opinian

upvoted 2 times

valuedate 1 year, 1 month ago

Selected Answer: AB

I would prefer AB

upvoted 2 times

VerRi 1 year, 1 month ago

Selected Answer: AB

Lambda for kick start Athena

Step Functions for orchestration

upvoted 3 times

sdas1 1 year, 1 month ago

Option C and D involve using an AWS Glue Python shell script to run a sleep timer and periodically check whether the current Athena query has finished running. While this approach might seem cost-effective in terms of using AWS Glue, it's not the most efficient way to manage the execution of Athena queries. AWS Glue is primarily designed for ETL (Extract, Transform, Load) tasks rather than orchestrating long-running query execution.

Therefore, while both options B, C and D could technically work, they might not be the most cost-effective or efficient solutions for orchestrating long-running Athena queries. Instead, options A and E would likely be more cost-effective and suitable for this scenario.

upvoted 1 times

sdas1 1 year, 1 month ago

Option B, utilizing AWS Step Functions, can be a cost-effective solution for orchestrating the execution of Athena queries, but it might not be the most cost-effective in this scenario because Step Functions are billed based on state transitions and the duration of state execution. Since each query can run for more than 15 minutes, using Step Functions to wait and periodically check the status of the queries could potentially result in higher costs, especially if the queries frequently take a long time to complete.

upvoted 1 times

Question #20 *Topic 1*

A company is migrating on-premises workloads to AWS. The company wants to reduce overall operational overhead. The company also wants to explore serverless options.

The company's current workloads use Apache Pig, Apache Oozie, Apache Spark, Apache Hbase, and Apache Flink. The on-premises workloads process petabytes of data in seconds. The company must maintain similar or better performance after the migration to AWS.

Which extract, transform, and load (ETL) service will meet these requirements?

A. AWS Glue

B. Amazon EMR

C. AWS Lambda

D. Amazon Redshift

Suggested Answer: *B*

Community vote distribution

B (79%) A (21%)

milofficial Highly Voted 1 year, 5 months ago

Selected Answer: B

Glue is like the more good-looking one, but weaker brother of EMR. So when it's about petabyte scales, let EMR do the work and have Glue stay away from the action.

upvoted 18 times

Ell89 Most Recent 4 months ago

Selected Answer: B

Glue doesnt natively support Pig, HBase and Flink.

upvoted 1 times

Udyan 5 months, 2 weeks ago

Selected Answer: B

Apache = EMR

upvoted 1 times

heavenlypearl 7 months, 3 weeks ago

Selected Answer: B

Amazon EMR Serverless is a deployment option for Amazon EMR that provides a serverless runtime environment. This simplifies the operation of analytics applications that use the latest open-source frameworks, such as Apache Spark and Apache Hive. With EMR Serverless, you don't have to configure, optimize, secure, or operate clusters to run applications with these frameworks.

https://docs.aws.amazon.com/emr/latest/EMR-Serverless-UserGuide/emr-serverless.html

upvoted 2 times

87ebc7d 8 months ago

Discarded, not 'discarted'. 'Discarted' isn't a word.

upvoted 2 times

leotoras 9 months, 3 weeks ago

B.

Amazon EMR Serverless is a deployment option for Amazon EMR that provides a serverless runtime environment. This simplifies the operation of analytics applications that use the latest open-source frameworks, such as Apache Spark and Apache Hive. With EMR Serverless, you don't have to configure, optimize, secure, or operate clusters to run applications with these frameworks.

upvoted 1 times

Eleftheriia 10 months ago

Selected Answer: A

I think it is A, Glue

• Amazon EMR is used for petabyte-scale data collection and data processing.

• AWS Glue is used as a serverless and managed ETL service, and also used for managing data quality with AWS Glue Data Quality.

upvoted 2 times

San_Juan 10 months, 1 week ago

Selected Answer: A

Glue.

It talks about "serverless" so EMR is discarted. The mention of Spark, Hbase, etc is for confusing you, because it doesn't say that they wanted to keep using them. Glue can run Spark using "glueContext" (similar a SparkContext) for reading tables, files and create frames.

upvoted 1 times

sachin 10 months, 3 weeks ago

The company also wants to explore serverless options. ? Glue (A). or EMR Serverless

upvoted 1 times

V0811 10 months, 4 weeks ago

Selected Answer: A

Serverless: AWS Glue is a fully managed, serverless ETL service that automates the process of data discovery, preparation, and transformation, helping minimize operational overhead.Integration with Big Data Tools: It integrates well with various AWS services and supports Spark jobs for ETL purposes, which aligns well with Apache Spark workloads.Performance: AWS Glue can handle large-scale ETL workloads, and it is designed to manage petabytes of data efficiently, comparable to the performance of on-premises solutions.While B. Amazon EMR could also be considered for its flexibility in handling big data workloads using tools like Apache Spark, it requires more management and doesn't fit the serverless requirement as closely as AWS Glue. Therefore, AWS Glue is the most suitable choice given the constraints and requirements.

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: B

EMR provides a managed Hadoop framework that natively supports Apache Pig,
Oozie, Spark, and Flink. This allows the company to migrate their existing workloads with minimal code changes, reducing development effort

upvoted 3 times

tgv 1 year ago

Selected Answer: B

That's exactly the purpose of EMR.

"Amazon EMR is the industry-leading cloud big data solution for petabyte-scale data processing, interactive analytics, and machine learning using open-source frameworks such as Apache Spark, Apache Hive, and Presto."

https://aws.amazon.com/emr/

upvoted 2 times

Just_Ninja 1 year, 1 month ago

Selected Answer: A

Glue is Serverless :)

upvoted 3 times

wa212 1 year, 2 months ago

Selected Answer: B

https://docs.aws.amazon.com/ja_jp/emr/latest/ManagementGuide/emr-what-is-emr.html

upvoted 2 times

certplan 1 year, 3 months ago

- While AWS Glue is a fully managed ETL service and offers serverless capabilities, it might not provide the same level of performance and flexibility as Amazon EMR for handling petabyte-scale workloads with complex processing requirements.
- AWS Glue is optimized for data integration, cataloging, and ETL jobs but may not be as well-suited for heavy-duty processing tasks that require frameworks like Apache Spark, Apache Flink, etc., which are commonly used for large-scale data processing.
- Documentation on AWS Glue can be found in the AWS Glue Developer Guide https://docs.aws.amazon.com/glue/index.html.

upvoted 2 times

certplan 1 year, 3 months ago

A. AWS Glue:

AWS Glue is a fully managed extract, transform, and load (ETL) service provided by Amazon Web Services (AWS). It allows users to prepare and load data for analytics purposes

B. Amazon EMR:

Amazon Elastic MapReduce (EMR) is a cloud-based big data platform provided by AWS. It allows users to process and analyze large amounts of data using popular frameworks such as Apache Hadoop, Apache Spark, Apache Hive, Apache HBase, and more.

https://docs.aws.amazon.com/emr/index.html
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-best-practices.html
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-manage.html
https://docs.aws.amazon.com/emr/latest/DeveloperGuide/emr-developer-guide.html

As per the AWS/Amazon docs, option B specifically calls out it out with the specific features/options that the question asked directly about.

upvoted 2 times

⊟ **GiorgioGss** 1 year, 3 months ago

Selected Answer: B

https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-release-components.html

upvoted 1 times

Question #21 *Topic 1*

A data engineer must use AWS services to ingest a dataset into an Amazon S3 data lake. The data engineer profiles the dataset and discovers that the dataset contains personally identifiable information (PII). The data engineer must implement a solution to profile the dataset and obfuscate the PII.
Which solution will meet this requirement with the LEAST operational effort?

A. Use an Amazon Kinesis Data Firehose delivery stream to process the dataset. Create an AWS Lambda transform function to identify the PII. Use an AWS SDK to obfuscate the PII. Set the S3 data lake as the target for the delivery stream.

B. Use the Detect PII transform in AWS Glue Studio to identify the PII. Obfuscate the PII. Use an AWS Step Functions state machine to orchestrate a data pipeline to ingest the data into the S3 data lake.

C. Use the Detect PII transform in AWS Glue Studio to identify the PII. Create a rule in AWS Glue Data Quality to obfuscate the PII. Use an AWS Step Functions state machine to orchestrate a data pipeline to ingest the data into the S3 data lake.

D. Ingest the dataset into Amazon DynamoDB. Create an AWS Lambda function to identify and obfuscate the PII in the DynamoDB table and to transform the data. Use the same Lambda function to ingest the data into the S3 data lake.

Suggested Answer: *B*

Community vote distribution

B (65%) C (32%)

⊟ **milofficial** Highly Voted 1 year, 3 months ago

Selected Answer: B

How does Data Quality obfuscate PII? You can do this directly in Glue Studio: https://docs.aws.amazon.com/glue/latest/dg/detect-PII.html

upvoted 12 times

⊟ **Eleftheriia** 6 months, 4 weeks ago

Yes, and regarding the "Create a rule in AWS Glue Data Quality to obfuscate the PII. " which is included in answer C, it cannot be done like this because in the aws glue console there is a section, "detect sensitive data" and then "types of sensitive information to detect". Therefore through this console you can obfuscate PII.
Relevant tutorial: https://www.youtube.com/watch?v=-TZZBfcnxBw

upvoted 1 times

⊟ **Khooks** Highly Voted 1 year ago

Selected Answer: B

Option C involves additional steps and complexity with creating rules in AWS Glue Data Quality, which adds more operational effort compared to directly using AWS Glue Studio's capabilities.

upvoted 5 times

⊟ **Kalyso** Most Recent 3 months ago

Selected Answer: B

Actually it is B. No need to create a rule in AWS Glue.

upvoted 1 times

⊟ **plutonash** 5 months, 2 weeks ago

Selected Answer: C

B. Use the Detect PII transform in AWS Glue Studio to identify the PII. Obfuscate the PII. Use an AWS Step Functions state machine to orchestrate a data pipeline to ingest the data into the S3 data lake. Detect PII transform only detects. Obfuscate the PII ok but how ? Answer C explain how

upvoted 1 times

⊟ **Udyan** 5 months, 2 weeks ago

Selected Answer: C

Why C is better than B:
Obfuscation clarity: Option C explicitly mentions using a Glue Data Quality rule to obfuscate PII, while option B does not specify how obfuscation is implemented.
Accuracy: Glue Data Quality provides a more structured way to handle obfuscation compared to relying solely on Glue Studio's PII detection.
Thus, C is the most accurate and operationally efficient solution.

upvoted 1 times

markill123 9 months, 3 weeks ago

The keyt

upvoted 1 times

antun3ra 10 months, 3 weeks ago

Selected Answer: B

B provides a streamlined, mostly visual approach using purpose-built tools for data processing and PII handling, making it the solution with the least operational effort.

upvoted 2 times

portland 11 months, 1 week ago

Selected Answer: C

https://aws.amazon.com/blogs/big-data/automated-data-governance-with-aws-glue-data-quality-sensitive-data-detection-and-aws-lake-formation/

upvoted 1 times

portland 11 months, 1 week ago

Actually it is B

upvoted 1 times

qwertyuio 11 months, 3 weeks ago

Selected Answer: B

https://docs.aws.amazon.com/glue/latest/dg/detect-PII.html

upvoted 2 times

bakarys 12 months ago

Selected Answer: C

anwser is C

upvoted 1 times

bigfoot1501 1 year ago

I don't think we need to use much more services to fulfill these requirements. Just AWS Glue is enough, it can detect and obfuscate PII data already. Source: https://docs.aws.amazon.com/glue/latest/dg/detect-PII.html#choose-action-pii

upvoted 3 times

VerRi 1 year, 1 month ago

Selected Answer: C

We cannot directly handle PII with Glue Studio, and Glue Data Quality can be used to handle PII.

upvoted 3 times

Just_Ninja 1 year, 1 month ago

Selected Answer: A

A very easy was is to use the SDK to identify PII.

https://docs.aws.amazon.com/code-library/latest/ug/comprehend_example_comprehend_DetectPiiEntities_section.html

upvoted 1 times

kairosfc 1 year, 1 month ago

Selected Answer: C

The transform Detect PII in AWS Glue Studio is specifically used to identify personally identifiable information (PII) within the data. It can detect and flag this information, but on its own, it does not perform the obfuscation or removal of these details.

To effectively obfuscate or alter the identified PII, an additional transformation would be necessary. This could be accomplished in several ways, such as:

Writing a custom script within the same AWS Glue job using Python or Scala to modify the PII data as needed.
Using AWS Glue Data Quality, if available, to create rules that automatically obfuscate or modify the data identified as PII. AWS Glue Data Quality is a newer tool that helps improve data quality through rules and transformations, but whether it's needed will depend on the functionality's availability and the specificity of the obfuscation requirements

upvoted 3 times

okechi 1 year, 2 months ago

Answer is option C. Period

upvoted 2 times

arvehisa 1 year, 3 months ago

Selected Answer: B

B is correct.

C: glue data quality cannot obfuscate the PII

D: need to write code but the question is the "LEAST operational effort"

upvoted 4 times

certplan 1 year, 3 months ago

In python ---

```
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from pyspark.sql import SparkSession

# Initialize Spark session
spark = SparkSession.builder \
.appName("Example Glue Job") \
.getOrCreate()

# Initialize Glue context
glueContext = GlueContext(SparkContext.getOrCreate())

# Retrieve Glue job arguments
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

# Define your EMR step
emr_step = [
{
"Name": "My EMR Step",
"ActionOnFailure": "CONTINUE",
"HadoopJarStep": {
"Jar": "s3://your-bucket/emr-scripts/your_script.jar",
"Args": [
"arg1",
"arg2"
]
}
}
]

# Execute the EMR step
response = glueContext.start_job_run(args['JOB_NAME'], job_run_args={'--extra-py-files': 'your_script.py'})
print(response)
```

upvoted 2 times

Question #22 *Topic 1*

A company maintains multiple extract, transform, and load (ETL) workflows that ingest data from the company's operational databases into an Amazon S3 based data lake. The ETL workflows use AWS Glue and Amazon EMR to process data.
The company wants to improve the existing architecture to provide automated orchestration and to require minimal manual effort.
Which solution will meet these requirements with the LEAST operational overhead?

A. AWS Glue workflows

B. AWS Step Functions tasks

C. AWS Lambda functions

D. Amazon Managed Workflows for Apache Airflow (Amazon MWAA) workflows

Suggested Answer: *B*

Community vote distribution

B (70%) A (30%)

valuedate Highly Voted 1 year, 1 month ago

Selected Answer: B

Glue Workflow only orchestrate crawlers and glue jobs

upvoted 15 times

DevoteamAnalytix Highly Voted 1 year, 1 month ago

Selected Answer: B

For me it's B because I did not found a possibility how Glue can trigger/orchestrate EMR processes OOTB.
But with StepFunction there is a way: https://aws.amazon.com/blogs/big-data/orchestrate-amazon-emr-serverless-jobs-with-aws-step-functions/

upvoted 7 times

Rpathak4 Most Recent 3 months, 1 week ago

Selected Answer: A

Why Not the Other Options?

B. AWS Step Functions More flexible but requires manual setup of states and transitions for Glue & EMR. Higher operational overhead than Glue Workflows.
C. AWS Lambda Lambda is not ideal for long-running ETL workflows. Best suited for lightweight data transformations or event-driven tasks.
D. Amazon MWAA (Apache Airflow) More control but requires cluster management and custom DAGs. Higher maintenance than Glue Workflows.

upvoted 1 times

Palee 3 months, 2 weeks ago

Selected Answer: B

The company wants to improve the existing architecture so A cannot be the right choice

upvoted 1 times

plutonash 5 months, 2 weeks ago

Selected Answer: B

it is interesting to choose A for minimum effort but only step functions can trigger the work both on EMR and on GLUE jobs

upvoted 1 times

ttpro1995 6 months, 1 week ago

Selected Answer: B

We have both Glue job and EMR job, so we need Step Functions to connect those.
Airflow can do it, but required more dev work.

upvoted 2 times

Adrifersilva 9 months ago

Selected Answer: A

glue workflows is part of the glue ecosystem so its provides seamless integration with minimal changes

upvoted 1 times

Shatheesh 9 months ago

Answer A, Glue workflows

upvoted 1 times

Shanmahi 10 months, 1 week ago

Selected Answer: A

Glue workflows are managed services and best for considering least operational overhead.

upvoted 1 times

V0811 10 months, 4 weeks ago

Selected Answer: A

AWS Glue Workflows are specifically designed for orchestrating ETL jobs in AWS Glue. They allow you to define and manage complex workflows that include multiple jobs and triggers, all within the AWS Glue environment.Integration: AWS Glue workflows seamlessly integrate with other AWS Glue components, making it easier to manage ETL processes without the need for external orchestration tools.Minimal Operational Overhead: Since AWS Glue is a fully managed service, using Glue workflows will reduce the operational overhead compared to managing separate orchestrators or building custom solutions.While D. Amazon Managed Workflows for Apache Airflow (Amazon MWAA) is also a good choice for more complex orchestration, it may involve more management overhead compared to the more straightforward AWS Glue workflows. Thus, AWS Glue workflows provide the least operational overhead given the context of this scenario.

upvoted 1 times

HunkyBunky 12 months ago

Selected Answer: B

B - because AWS Glue can't trigger EMR

upvoted 1 times

FunkyFresco 1 year, 1 month ago

Selected Answer: B

EMR in workflows , i dont think so

upvoted 3 times

VerRi 1 year, 1 month ago

Selected Answer: B

There is no way for Glue Workflow to trigger EMR

upvoted 4 times

acoshi 1 year, 2 months ago

Selected Answer: A

https://aws.amazon.com/blogs/big-data/orchestrate-an-etl-pipeline-using-aws-glue-workflows-triggers-and-crawlers-with-custom-classifiers/

upvoted 2 times

lucas_rfsb 1 year, 3 months ago

Selected Answer: A

Since it seems to me that this pipeline is complex, with multiple workflows, I would go for Glue workflows.

upvoted 6 times

jasango 1 year, 3 months ago

Yo me voy por la D) Amazon MWAA porque Glue Workflows solo admite Jobs de Glue y Step Function puede fucionar pero no son workflows de datos. Amazon MWAA son workflows de datos y esta integrado tanto con Glue como EMR: https://aws.amazon.com/blogs/big-data/simplify-aws-glue-job-orchestration-and-monitoring-with-amazon-mwaa/

upvoted 3 times

certplan 1 year, 3 months ago

Here's an example of how you can use AWS Glue to initiate an EMR (Elastic MapReduce) job:

Let's assume you have an AWS Glue job that performs ETL tasks on data stored in Amazon S3. You want to leverage EMR for a specific task within this job, such as running a complex Spark job.

1. Define a Glue Job: Create an AWS Glue job using the AWS Glue console, SDK, or CLI. Define the input and output data sources, as well as the transformations you want to apply.

2. Incorporate EMR Step: Within the Glue job script, include a section where you define an EMR step. An EMR step is a unit of work that performs a specific task on an EMR cluster.

Code follows in the next entry...

upvoted 2 times

Question #23 *Topic 1*

A company currently stores all of its data in Amazon S3 by using the S3 Standard storage class.

A data engineer examined data access patterns to identify trends. During the first 6 months, most data files are accessed several times each day. Between 6 months and 2 years, most data files are accessed once or twice each month. After 2 years, data files are accessed only once or twice each year.

The data engineer needs to use an S3 Lifecycle policy to develop new data storage rules. The new storage solution must continue to provide high availability.

Which solution will meet these requirements in the MOST cost-effective way?

A. Transition objects to S3 One Zone-Infrequent Access (S3 One Zone-IA) after 6 months. Transfer objects to S3 Glacier Flexible Retrieval after 2 years.

B. Transition objects to S3 Standard-Infrequent Access (S3 Standard-IA) after 6 months. Transfer objects to S3 Glacier Flexible Retrieval after 2 years.

C. Transition objects to S3 Standard-Infrequent Access (S3 Standard-IA) after 6 months. Transfer objects to S3 Glacier Deep Archive after 2 years.

D. Transition objects to S3 One Zone-Infrequent Access (S3 One Zone-IA) after 6 months. Transfer objects to S3 Glacier Deep Archive after 2 years.

Suggested Answer: *B*

Community vote distribution

B (52%) C (48%)

⊟ **helpaws** Highly Voted 9 months, 1 week ago

Selected Answer: B

"S3 Glacier Flexible Retrieval delivers low-cost storage, up to 10% lower cost (than S3 Glacier Instant Retrieval), for archive data that is accessed 1-2 times per year and is retrieved asynchronously"

Source: https://aws.amazon.com/s3/storage-classes/glacier/

upvoted 21 times

⊟ **ttpro1995** 6 months, 1 week ago

The requirement does not state how fast data need to be retrieval. So, pick Glacier deep for even more cost saving.

upvoted 2 times

⊟ **JimOGrady** 3 months, 1 week ago

question states: "must continue to provide high availability." So NOT Deep Archive

upvoted 1 times

⊟ **sam_pre** 3 months ago

High availability and fast retrieval are two different things

upvoted 1 times

⊟ **WarPig666** Highly Voted 6 months, 2 weeks ago

Selected Answer: C

Flexible retrieval will be higher cost than deep archive. If records only need to be retrieved once or twice a year, this doesn't mean they need to be instantly available.

upvoted 8 times

⊟ **Tani0908** Most Recent 6 days, 22 hours ago

Selected Answer: B

As they mention high availability it will be B

upvoted 1 times

⊟ **AM027** 2 months, 1 week ago

Selected Answer: B

"S3 Glacier Flexible Retrieval delivers low-cost storage, up to 10% lower cost (than S3 Glacier Instant Retrieval), for archive data that is accessed 1-2 times per year and is retrieved asynchronously"

upvoted 1 times

Rpathak4 3 months, 1 week ago

Selected Answer: C

Why Not the Other Options?

A. S3 One Zone-IA → Glacier Flexible Retrieval ✖ One Zone-IA is risky (data loss if the AZ fails). Glacier Flexible Retrieval is more expensive than Deep Archive.

B. S3 Standard-IA → Glacier Flexible Retrieval ✖ Glacier Flexible Retrieval is not the cheapest long-term storage. Deep Archive costs much less.

D. S3 One Zone-IA → Glacier Deep Archive ✖ One Zone-IA lacks high availability (single AZ failure = data loss). S3 Standard-IA is safer.

upvoted 1 times

anonymous_learner_2 4 months, 1 week ago

Selected Answer: C

Glacier deep archive has the same availability as flexible retrieval and there's no retrieval time requirement so C is the most cost effective that meets the requirements.

upvoted 2 times

luigiDDD 5 months, 1 week ago

Selected Answer: C

C is the most cost effective

upvoted 2 times

plutonash 5 months, 2 weeks ago

Selected Answer: B

"data files are accessed only once or twice each year", this is "S3 Glacier Flexible Retrieval" definition

upvoted 1 times

Udyan 5 months, 2 weeks ago

Selected Answer: C

Is it mentioned in question that Retrieval time is constraint, no, so, if any engineer need to access data, say May and November, so he/she can wait for 2-3 days to get data, as in the long run, they have an year to analyze the data so, deep archive will save costs only.

upvoted 2 times

Udyan 5 months, 2 weeks ago

Selected Answer: C

This question was in Stephen Maarek Udemy practice questions too, here concern not given for extraction time so, just see cost friendlyness, thus, C over B

upvoted 2 times

HagarTheHorrible 6 months, 1 week ago

Selected Answer: B

deep archive doesn't make sense

upvoted 1 times

Eleftheriia 6 months, 3 weeks ago

Selected Answer: B

For once or twice a year it is flexible retrieval.

upvoted 1 times

jk15997 6 months, 3 weeks ago

Selected Answer: C

There is no requirement for the retrieval time.

upvoted 2 times

altonh 6 months, 4 weeks ago

Selected Answer: C

There is no requirement for the retrieval time. So this is more cost-effective.

upvoted 4 times

iamwatchingyoualways 7 months ago

Selected Answer: C

No instant access is mentioned. Most Cost effective.

upvoted 3 times

truongnguyen86 7 months, 2 weeks ago

Selected Answer: B

Option B is the correct answer because it balances cost-effectiveness and availability:

S3 Standard-IA offers cost savings for infrequently accessed data while maintaining high availability across multiple zones.

S3 Glacier Flexible Retrieval is a good balance for archiving with occasional access needs.

upvoted 2 times

lsj900605 7 months, 3 weeks ago

Selected Answer: B

B

High availability means the need for readily available service.

S3 Standard-IA deliver 99.9% availability vs S3 One Zone-IA deliver 99.5% availability

S3 Glacier Flexible Retrieval has configurable retrieval times, from minutes to hours, with free bulk retrievals.

But with S3 Glacier Deep Archive it's retrieval time is within 12 hours

https://aws.amazon.com/s3/storage-classes/

upvoted 1 times

Question #24 *Topic 1*

A company maintains an Amazon Redshift provisioned cluster that the company uses for extract, transform, and load (ETL) operations to support critical analysis tasks. A sales team within the company maintains a Redshift cluster that the sales team uses for business intelligence (BI) tasks. The sales team recently requested access to the data that is in the ETL Redshift cluster so the team can perform weekly summary analysis tasks. The sales team needs to join data from the ETL cluster with data that is in the sales team's BI cluster.

The company needs a solution that will share the ETL cluster data with the sales team without interrupting the critical analysis tasks. The solution must minimize usage of the computing resources of the ETL cluster.

Which solution will meet these requirements?

A. Set up the sales team BI cluster as a consumer of the ETL cluster by using Redshift data sharing.

B. Create materialized views based on the sales team's requirements. Grant the sales team direct access to the ETL cluster.

C. Create database views based on the sales team's requirements. Grant the sales team direct access to the ETL cluster.

D. Unload a copy of the data from the ETL cluster to an Amazon S3 bucket every week. Create an Amazon Redshift Spectrum table based on the content of the ETL cluster.

Suggested Answer: *A*

Community vote distribution

A (67%) D (33%)

arvehisa Highly Voted 1 year, 3 months ago

Selected Answer: A

A: redshift data sharing:

https://docs.aws.amazon.com/redshift/latest/dg/data_sharing_intro.html

With data sharing, you can securely and easily share live data across Amazon Redshift clusters.

B: materialized view is only within 1 redshift cluster, across different tables

upvoted 5 times

lucas_rfsb Highly Voted 1 year, 3 months ago

Selected Answer: D

In my opinion using Redshift Data Sharing will consume less resources. 'D' envolves using a S3 bucket.

upvoted 5 times

lucas_rfsb 1 year, 3 months ago

Sorry I wanted to select A but did D

upvoted 7 times

motk123 Most Recent 9 months, 1 week ago

Seems that the performance of the critical ETL cluster should not be affected when using data sharing, so the answer is likely A:

https://docs.aws.amazon.com/redshift/latest/dg/data_sharing_intro.html

Supporting different kinds of business-critical workloads – Use a central extract, transform, and load (ETL) cluster that shares data with multiple business intelligence (BI) or analytic clusters. This approach provides read workload isolation and chargeback for individual workloads. You can size and scale your individual workload compute according to the workload-specific requirements of price and performance.

https://docs.aws.amazon.com/redshift/latest/dg/considerations.html

The performance of the queries on shared data depends on the compute capacity of the consumer clusters.

upvoted 2 times

wimalik 9 months, 4 weeks ago

A as Redshift data sharing allows you to share live data across Redshift clusters without having to duplicate the data. This feature enables the sales team to access the data from the ETL cluster directly without interrupting the critical analysis tasks or overloading the ETL cluster's resources. The sales team can join this shared data with their own data in the BI cluster efficiently.

upvoted 1 times

San_Juan 10 months ago

Selected Answer: D

"The solution must minimize usage of the computing resources of the ETL cluster." That is key. You shouldn't use ETL cluster, so unload data to S3 and run queries in a separate Redshift Spectrum database. ETL cluster do nothing meanwhile.

upvoted 1 times

VerRi 1 year, 1 month ago

Selected Answer: A

Typetical Redshift data sharing use case

upvoted 3 times

valuedate 1 year, 1 month ago

key words: "weekly"
"The solution must minimize usage of the computing resources of the ETL cluster."

Answer:D

upvoted 2 times

d8945a1 1 year, 1 month ago

Selected Answer: A

Typical usecase of datasharing in Redshift.

The question mentions that - 'team needs to join data from the ETL cluster with data that is in the sales team's BI cluster.' This is possible with datashare.

upvoted 4 times

jasango 1 year, 3 months ago

Selected Answer: D

The spectrum table is accessed from the sales cluster with zero impact on the ETL cluster.

upvoted 3 times

certplan 1 year, 3 months ago

Options A, B, and C involve granting the sales team direct access to the ETL cluster, which could potentially impact the performance of the ETL cluster and interfere with its critical analysis tasks. Option D provides a more isolated and scalable approach by leveraging Amazon S3 and Redshift Spectrum for data sharing while minimizing the usage of the ETL cluster's computing resources.

https://docs.aws.amazon.com/redshift/latest/dg/c-using-spectrum-sharing-data.html
https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-design-tables.html

upvoted 1 times

certplan 1 year, 3 months ago

Overall, while both options offer ways to share data between the ETL and BI clusters, Option D offers a more robust and scalable solution that minimizes the impact on the ETL cluster's resources and provides greater flexibility and independence for the sales team's analysis tasks.

By unloading a copy of the data from the ETL cluster to Amazon S3 and leveraging Redshift Spectrum for querying, the solution aligns with AWS best practices for managing data and resource usage in Amazon Redshift clusters. It ensures that critical analysis tasks are not interrupted while providing the sales team with the necessary access to perform their analysis tasks efficiently.

upvoted 2 times

GiorgioGss 1 year, 3 months ago

Selected Answer: A

Initially I would go with B but that definitely will use more resource.

upvoted 5 times

[Removed] 1 year, 5 months ago

Selected Answer: A

To share data between Redshift clusters and meet the requirements of sharing ETL cluster data with the sales team without interrupting critical analysis tasks and minimizing the usage of the ETL cluster's computing resources, Redshift Data Sharing is the way to go

https://docs.aws.amazon.com/redshift/latest/dg/data_sharing_intro.html

"Supporting different kinds of business-critical workloads – Use a central extract, transform, and load (ETL) cluster that shares data with multiple business intelligence (BI) or analytic clusters. This approach provides read workload isolation and chargeback for individual workloads. You can size and scale your individual workload compute according to the workload-specific requirements of price and performance"

upvoted 4 times

Question #25 *Topic 1*

A data engineer needs to join data from multiple sources to perform a one-time analysis job. The data is stored in Amazon DynamoDB, Amazon RDS, Amazon Redshift, and Amazon S3.
Which solution will meet this requirement MOST cost-effectively?

A. Use an Amazon EMR provisioned cluster to read from all sources. Use Apache Spark to join the data and perform the analysis.

B. Copy the data from DynamoDB, Amazon RDS, and Amazon Redshift into Amazon S3. Run Amazon Athena queries directly on the S3 files.

C. Use Amazon Athena Federated Query to join the data from all data sources.

D. Use Redshift Spectrum to query data from DynamoDB, Amazon RDS, and Amazon S3 directly from Redshift.

Suggested Answer: *C*

Community vote distribution

C (100%)

⊟ **lucas_rfsb** Highly Voted 👍 9 months ago

Selected Answer: C

I would go for C because Federated Query is typical for this porpouse. Besides, we don't need to add/duplicate resources in S3. But I see that, becasuse Athena is more optimized for S3, it can be considered a tricky question, since there can be more trade-offs to consider, such as data governance that are easier if data is centralized in S3 in my opinion.

upvoted 7 times

⊟ **pypelyncar** Most Recent 🕘 6 months, 3 weeks ago

Selected Answer: C

Serverless Processing: Athena is a serverless query service, meaning you only pay for the queries you run. This eliminates the need to provision and manage compute resources like in EMR clusters,
making it ideal for one-time jobs.
Federated Query Capability: Athena Federated Query allows you to directly query data from various sources like DynamoDB, RDS, Redshift, and S3 without physically moving the data. This eliminates data movement costs and simplifies the analysis process.
Reduced Cost for Large Datasets: Compared to copying data to S3, which can be expensive for large datasets, Athena Federated Query avoids unnecessary data movement, reducing overall costs.

upvoted 4 times

⊟ **certplan** 9 months, 1 week ago

Amazon Athena Federated Query allows you to query data from multiple federated data sources including relational databases, NoSQL databases, and object stores directly from Athena. While this might seem like an efficient way to join data from different sources without the need for copying data into Amazon S3, it's essential to consider the cost implications.

AWS documentation on Amazon Athena Federated Query [1] explains that while Federated Query enables you to query data from external data sources without data movement, it does not eliminate data transfer costs. Depending on the data sources involved (such as Amazon RDS, DynamoDB, etc.), there might be data transfer costs associated with querying data directly from these sources.

[1] Amazon Athena Federated Query Documentation: https://docs.aws.amazon.com/athena/latest/ug/federated-data-sources.html

upvoted 2 times

⊟ **certplan** 9 months, 1 week ago

1. Data Storage Costs: Storing data in Amazon S3 is generally cheaper compared to the other AWS storage options like Amazon Redshift or Amazon RDS.

2. Compute Costs: Amazon: Athena is a serverless query service that allows you to query data directly from S3 without the need for provisioning or managing infrastructure. You only pay for the queries you run, which can be more cost-effective compared to provisioning an EMR cluster (option A) or using Redshift Spectrum (option D), both of which involve compute resources that you might not fully utilize.

3. Data Transfer Costs: Option B involves copying the data once into S3, and then there are no additional data transfer costs for querying the data using Athena. In contrast, options A and D would involve data transfer costs as data is moved between different services.

Amazon Athena Pricing: https://aws.amazon.com/athena/pricing/

Amazon S3 Pricing: https://aws.amazon.com/s3/pricing/

upvoted 1 times

⊟ **certplan** 9 months, 1 week ago

Point:

"perform a one-time analysis job"

Option C (Amazon Athena Federated Query) might seem appealing, but it's generally more suited for querying data from external sources without copying the data into S3. However, since the data is already within AWS services, copying it to S3 and using Athena directly would likely be more cost-effective.

upvoted 1 times

⊟ **[Removed]** 11 months, 2 weeks ago

Selected Answer: C

You can query these sources by using Federated Queries, which is a native feature of Athena. The other options may increase costs and operational overhead, as they use more than one service to achieve the same result

https://docs.aws.amazon.com/athena/latest/ug/connectors-available.html

upvoted 4 times

⊟ **GiorgioGss** 9 months, 3 weeks ago

Agree. C

upvoted 2 times

Question #26 *Topic 1*

A company is planning to use a provisioned Amazon EMR cluster that runs Apache Spark jobs to perform big data analysis. The company requires high reliability. A big data team must follow best practices for running cost-optimized and long-running workloads on Amazon EMR. The team must find a solution that will maintain the company's current level of performance.
Which combination of resources will meet these requirements MOST cost-effectively? (Choose two.)

A. Use Hadoop Distributed File System (HDFS) as a persistent data store.

B. Use Amazon S3 as a persistent data store.

C. Use x86-based instances for core nodes and task nodes.

D. Use Graviton instances for core nodes and task nodes.

E. Use Spot Instances for all primary nodes.

Suggested Answer: *BD*

Community vote distribution

BD (100%)

⊟ **[Removed]** Highly Voted 11 months, 2 weeks ago

Selected Answer: BD

HDFS is not recommended for persistent storage because once a cluster is terminated, all HDFS data is lost. Also, long-running workloads can fill the disk space quickly. Thus, S3 is the best option since it's highly available, durable, and scalable.

AWS Graviton-based instances cost up to 20% less than comparable x86-based Amazon
EC2 instances: https://aws.amazon.com/ec2/graviton/

upvoted 9 times

⊟ **BartoszGolebiowski24** 10 months, 3 weeks ago

If you are using instance storage this is true, but you can use EBS instead of instance storage.
EBS has better performance than s3 for HDFS. This is the keyword from question, so EBS > S3

I would rather select AD.

upvoted 1 times

⊟ **sam_pre** Most Recent 3 months ago

Selected Answer: BD

Cost effective + high reliability > S3
Gravitation > Low cost

upvoted 1 times

⊟ **ttpro1995** 6 months, 1 week ago

Selected Answer: BD

Rule of thumb: pick the AWS in-house solution provided for that service.
Graviton is aws processor, and also EMRFS on S3.

upvoted 1 times

⊟ **pypelyncar** 6 months, 3 weeks ago

Selected Answer: BD

s3 no question.
Graviton=> Cost-Effectiveness: Graviton instances are ARM-based instances specifically designed for cloud workloads.
They offer significant cost savings compared to x86-based instances while delivering comparable or better performance for many Apache Spark workloads.
Performance: Graviton instances are optimized for Spark workloads and can deliver the same level of performance as x86-based instances in many cases. Additionally, EMR offers performance-optimized versions of Spark built for Graviton instances.

upvoted 3 times

⊟ **okechi** 8 months, 2 weeks ago

My answer is BE

upvoted 1 times

⊟ **chris_spencer** 8 months, 2 weeks ago

E is incorrect, Spot instances does not provide high reliability as required by the company.

upvoted 4 times

⊟ **certplan** 9 months, 1 week ago

A. - AWS recommends using Amazon S3 as a persistent data store for Amazon EMR due to its scalability, durability, and cost-effectiveness. Storing data in HDFS would require managing and maintaining additional infrastructure, which may incur higher costs in terms of storage, management, and scalability compared to using Amazon S3. AWS documentation emphasizes the benefits of integrating Amazon EMR with Amazon S3 for cost optimization and efficiency.

D. - While Graviton instances may offer cost savings in certain scenarios, they might not always be the most cost-effective option depending on the specific workload requirements and availability of compatible software. x86-based instances are more commonly supported by a broader range of software and frameworks, which could result in better performance and compatibility in some cases. Additionally, AWS documentation on instance types and pricing can provide insights into the cost-effectiveness of Graviton instances compared to x86-based instances.

upvoted 2 times

⊟ **GiorgioGss** 9 months, 3 weeks ago

Selected Answer: BD

B and D.

upvoted 3 times

⊟ **nyaopoko** 8 months, 4 weeks ago

yes BD is answer

upvoted 1 times

Question #27 *Topic 1*

A company wants to implement real-time analytics capabilities. The company wants to use Amazon Kinesis Data Streams and Amazon Redshift to ingest and process streaming data at the rate of several gigabytes per second. The company wants to derive near real-time insights by using existing business intelligence (BI) and analytics tools.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Kinesis Data Streams to stage data in Amazon S3. Use the COPY command to load data from Amazon S3 directly into Amazon Redshift to make the data immediately available for real-time analysis.

B. Access the data from Kinesis Data Streams by using SQL queries. Create materialized views directly on top of the stream. Refresh the materialized views regularly to query the most recent stream data.

C. Create an external schema in Amazon Redshift to map the data from Kinesis Data Streams to an Amazon Redshift object. Create a materialized view to read data from the stream. Set the materialized view to auto refresh.

D. Connect Kinesis Data Streams to Amazon Kinesis Data Firehose. Use Kinesis Data Firehose to stage the data in Amazon S3. Use the COPY command to load the data from Amazon S3 to a table in Amazon Redshift.

Suggested Answer: *C*

Community vote distribution

C (63%) D (34%)

blackgamer Highly Voted 1 year, 3 months ago

Selected Answer: C

The answer is C. It can provide near real-time insight analysis. Refer the article from AWS - https://aws.amazon.com/blogs/big-data/real-time-analytics-with-amazon-redshift-streaming-ingestion/

upvoted 8 times

helpaws Highly Voted 1 year, 3 months ago

Selected Answer: C

Key word here is near real-time. If it's involve S3 and COPY, it's not gonna be near real-time

upvoted 7 times

markill123 9 months, 2 weeks ago

Redshift cannot create external schemas that map directly to Kinesis Data Streams. You would still need an intermediary step, such as Firehose or S3, to handle data ingestion. Additionally, maintaining auto-refreshing materialized views directly from a stream isn't feasible with Redshift.

upvoted 6 times

melligeri Most Recent 3 months ago

Selected Answer: C

https://aws.amazon.com/blogs/big-data/real-time-analytics-with-amazon-redshift-streaming-ingestion/#:~:text=Before%20the%20launch,the%20data%20stream.

upvoted 1 times

Rpathak4 3 months, 1 week ago

Selected Answer: D

✅ Use Kinesis Data Firehose to load data into Redshift via S3 for the simplest and most scalable solution.
✅ Firehose automatically batches, transforms, and loads data with no manual intervention required.
✅ Achieves near real-time analytics with minimal operational effort.

upvoted 2 times

MephiboshethGumani 3 months, 2 weeks ago

Selected Answer: D

Creating an external schema and using materialized views directly on top of Kinesis Data Streams is also not an ideal choice because this approach can add complexity and doesn't leverage fully managed solutions like Kinesis Data Firehose. The manual management of data refresh rates adds operational overhead.

upvoted 1 times

Eltanany 3 months, 2 weeks ago

Selected Answer: C

Refer to the article from AWS - https://aws.amazon.com/blogs/big-data/real-time-analytics-with-amazon-redshift-streaming-ingestion/

upvoted 1 times

jesusmoh 3 months, 4 weeks ago

Selected Answer: D

option D provides a streamlined, efficient, and low-overhead approach to achieving real-time analytics with the specified technologies.

upvoted 1 times

plutonash 5 months, 2 weeks ago

Selected Answer: D

A: Kinesis Data Streams to stage data in Amazon S3. not really easy,

B: sql directly to Kinesis Data Streams : functionality not exist

C : external schema from redshift to Kinesis Data Streams : functionality not exist

D : near real-time = Kinesis Data Firehose

upvoted 2 times

subbie 5 months, 3 weeks ago

Selected Answer: C

https://aws.amazon.com/blogs/big-data/real-time-analytics-with-amazon-redshift-streaming-ingestion/

upvoted 1 times

subbie 5 months, 3 weeks ago

Selected Answer: B

https://aws.amazon.com/blogs/big-data/real-time-analytics-with-amazon-redshift-streaming-ingestion/

upvoted 1 times

haby 6 months, 1 week ago

Selected Answer: A

A for me

C - Redshift does not natively support direct mapping to Kinesis Data Streams. Some extra configs are needed.

D - There will be a 60s latency when using Firehose, so it's "Near" real time not real time.

upvoted 1 times

HagarTheHorrible 6 months, 1 week ago

Selected Answer: D

Redshift does not natively support direct mapping to Kinesis Data Streams. Materialized views cannot directly query streaming data from Kinesis.

upvoted 1 times

altonh 6 months, 4 weeks ago

Selected Answer: C

See https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-streaming-ingestion-getting-started.html

upvoted 1 times

Asen_Cat 7 months, 3 weeks ago

Selected Answer: D

D could be the most standard way to handle this case. How to use C to implement it is questionable for me.

upvoted 3 times

> **Asmunk** 7 months, 2 weeks ago
>
> https://docs.aws.amazon.com/streams/latest/dev/using-other-services-redshift.html
>
> upvoted 1 times

heavenlypearl 7 months, 3 weeks ago

Selected Answer: C

Amazon Redshift can automatically refresh materialized views with up-to-date data from its base tables when materialized views are created with or altered to have the autorefresh option. Amazon Redshift autorefreshes materialized views as soon as possible after base tables changes.

https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-refresh.html

upvoted 1 times

royalrum 8 months ago

Firehose is Near-Real time, you can set your buffer size and stream to either Redshift or S3 directly. Since Redshift is not in the option, use s3...

upvoted 1 times

Shatheesh 8 months, 1 week ago

Selected Answer: D

Kinesis Data Streams , option D using Kinesis Data Firehose is a fully managed service that automatically handles the ingestion of data

upvoted 1 times

Question #28 *Topic 1*

A company uses an Amazon QuickSight dashboard to monitor usage of one of the company's applications. The company uses AWS Glue jobs to process data for the dashboard. The company stores the data in a single Amazon S3 bucket. The company adds new data every day.
A data engineer discovers that dashboard queries are becoming slower over time. The data engineer determines that the root cause of the slowing queries is long-running AWS Glue jobs.
Which actions should the data engineer take to improve the performance of the AWS Glue jobs? (Choose two.)

A. Partition the data that is in the S3 bucket. Organize the data by year, month, and day.

B. Increase the AWS Glue instance size by scaling up the worker type.

C. Convert the AWS Glue schema to the DynamicFrame schema class.

D. Adjust AWS Glue job scheduling frequency so the jobs run half as many times each day.

E. Modify the IAM role that grants access to AWS glue to grant access to all S3 features.

Suggested Answer: *AB*

Community vote distribution

AB (100%)

rralucard_ Highly Voted 11 months ago

Selected Answer: AB

A. Partition the data that is in the S3 bucket. Organize the data by year, month, and day.

• Partitioning data in Amazon S3 can significantly improve query performance. By organizing the data by year, month, and day, AWS Glue and Amazon QuickSight can scan only the relevant partitions of data, which reduces the amount of data read and processed. This approach is particularly effective for time-series data, where queries often target specific time ranges.

B. Increase the AWS Glue instance size by scaling up the worker type.

• Scaling up the worker type can provide more computational resources to the AWS Glue jobs, enabling them to process data faster. This can be especially beneficial when dealing with large datasets or complex transformations. It's important to monitor the performance improvements and cost implications of scaling up.

upvoted 10 times

MLOPS_eng 6 months, 1 week ago

How does partitioning data in S3 improve the performance of AWS Glue jobs? Partitioning data s3 improve the query performance, but the question was the action should the DE take to improve the performance of AWS Glue jobs !

upvoted 1 times

Leo87656789 8 months, 3 weeks ago

I would also go for A, B.
But there are no worker types in AWS Glue. You can only increase the DPU.

upvoted 1 times

DevoteamAnalytix 7 months, 4 weeks ago

Here you can find 5 different Worker types:
https://docs.aws.amazon.com/glue/latest/dg/add-job.html

upvoted 2 times

tgv 7 months ago

It looks like there are various worker types in AWS Glue actually. I'll go with AB as well.

"With AWS Glue, you only pay for the time your ETL job takes to run. There are no resources to manage, no upfront costs, and you are not charged for startup or shutdown time. You are charged an hourly rate based on the number of Data Processing Units (or DPUs) used to run your ETL job. A single Data Processing Unit (DPU) is also referred to as a worker. AWS Glue comes with three worker types to help you select the configuration that meets your job latency and cost requirements. Workers come in Standard, G.1X, G.2X, and G.025X configurations."

https://docs.aws.amazon.com/glue/latest/dg/components-key-concepts.html

upvoted 2 times

⊟ certplan Most Recent 9 months, 1 week ago

1. **Partition the Data in Amazon S3**:

- AWS documentation on optimizing Amazon S3 performance: https://docs.aws.amazon.com/AmazonS3/latest/userguide/optimizing-performance.html
- AWS Glue documentation on partitioning data for AWS Glue jobs: https://docs.aws.amazon.com/glue/latest/dg/how-it-works.html#how-partitioning-works
- Best practices for partitioning in Amazon S3: https://docs.aws.amazon.com/AmazonS3/latest/userguide/best-practices-partitioning.html

2. **Optimizing AWS Glue Job Settings**:

- AWS Glue documentation on optimizing job performance: https://docs.aws.amazon.com/glue/latest/dg/best-practices.html
- AWS Glue documentation on scaling AWS Glue job resources: https://docs.aws.amazon.com/glue/latest/dg/monitor-profile-glue-job-cloudwatch-metrics.html

By referring to these documentation resources, the data engineer can gain insights into best practices and recommendations provided by AWS for optimizing AWS Glue jobs, thereby justifying the suggested actions to address the issue of slowing job performance.

upvoted 2 times

Question #29 *Topic 1*

A data engineer needs to use AWS Step Functions to design an orchestration workflow. The workflow must parallel process a large collection of data files and apply a specific transformation to each file.
Which Step Functions state should the data engineer use to meet these requirements?

A. Parallel state

B. Choice state

C. Map state

D. Wait state

Suggested Answer: *C*

Community vote distribution

C (100%)

GabrielSGoncalves 10 months ago

Selected Answer: C

Clearly is mapping state

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: C

The Map state allows you to define a single execution path for processing a collection of data items in parallel.
This aligns perfectly with the data engineer's requirement of parallel processing a large collection of data files

upvoted 3 times

FunkyFresco 1 year ago

Selected Answer: C

to execute in parallel

upvoted 1 times

sveni1502 1 year, 1 month ago

Selected Answer: C

C is Correct
To meet the requirement of parallel processing a large collection of data files and applying a specific transformation to each file, the data engineer should use the Map state in AWS Step Functions.
The Map state is specifically designed to run a set of tasks in parallel for each element in a collection or array. Each element (in this case, each data file) is processed independently and in parallel, allowing the workflow to take advantage of parallel processing.

upvoted 3 times

lucas_rfsb 1 year, 3 months ago

Selected Answer: C

C, Map state is correct

upvoted 1 times

Aesthet 1 year, 4 months ago

With Step Functions, you can orchestrate large-scale parallel workloads to perform tasks, such as on-demand processing of semi-structured data. These parallel workloads let you concurrently process large-scale data sources stored in Amazon S3. For example, you might process a single JSON or CSV file that contains large amounts of data. Or you might process a large set of Amazon S3 objects.

To set up a large-scale parallel workload in your workflows, include a Map state in Distributed mode.

upvoted 1 times

Aesthet 1 year, 4 months ago

C is correct.
Map state is designed precisely for the requirement described. It allows you to iterate over a collection of items, processing each item individually. The Map state can automatically manage the iteration and execute the specified transformation on each item in parallel, making it the perfect choice for parallel processing of a large collection of data files.

upvoted 1 times

⊟ **rralucard_** 1 year, 4 months ago

Selected Answer: C

The Map state is specifically designed for processing a collection of items (like data files) in parallel. It allows you to apply a transformation or a set of steps to each item in the input array independently.

The Map state automatically iterates over each item in the array and performs the defined steps. This makes it ideal for scenarios where you need to process a large number of files in a similar manner, as in your requirement.

upvoted 2 times

Question #30 *Topic 1*

A company is migrating a legacy application to an Amazon S3 based data lake. A data engineer reviewed data that is associated with the legacy application. The data engineer found that the legacy data contained some duplicate information.
The data engineer must identify and remove duplicate information from the legacy application data.
Which solution will meet these requirements with the LEAST operational overhead?

A. Write a custom extract, transform, and load (ETL) job in Python. Use the DataFrame.drop_duplicates() function by importing the Pandas library to perform data deduplication.

B. Write an AWS Glue extract, transform, and load (ETL) job. Use the FindMatches machine learning (ML) transform to transform the data to perform data deduplication.

C. Write a custom extract, transform, and load (ETL) job in Python. Import the Python dedupe library. Use the dedupe library to perform data deduplication.

D. Write an AWS Glue extract, transform, and load (ETL) job. Import the Python dedupe library. Use the dedupe library to perform data deduplication.

Suggested Answer: *B*

Community vote distribution

B (100%)

rralucard_ Highly Voted 👍 1 year, 4 months ago

Selected Answer: B

Option B, writing an AWS Glue ETL job with the FindMatches ML transform, is likely to meet the requirements with the least operational overhead. This solution leverages a managed service (AWS Glue) and incorporates a built-in ML transform specifically designed for deduplication, thus minimizing the need for manual setup, maintenance, and machine learning expertise.

upvoted 6 times

_JP_ Most Recent ⊙ 6 months, 2 weeks ago

Selected Answer: A

I disagree with B. That option requires additional effort just to train the ML model with labeled data. Option A is as simple as to use the robust pandas library

upvoted 2 times

V0811 10 months, 4 weeks ago

Selected Answer: B

100 % B

upvoted 1 times

GiorgioGss 1 year, 3 months ago

Selected Answer: B

B. https://docs.aws.amazon.com/glue/latest/dg/machine-learning.html
"Find matches
Finds duplicate records in the source data. You teach this machine learning transform by labeling example datasets to indicate which rows match. The machine learning transform learns which rows should be matches the more you teach it with example labeled data."

upvoted 4 times

Aesthet 1 year, 4 months ago

Remove duplicates from already migrated data - probably D.
Remove duplicates from data before migration - A is preferable.

upvoted 1 times

Question #31 *Topic 1*

A company is building an analytics solution. The solution uses Amazon S3 for data lake storage and Amazon Redshift for a data warehouse. The company wants to use Amazon Redshift Spectrum to query the data that is in Amazon S3.
Which actions will provide the FASTEST queries? (Choose two.)

A. Use gzip compression to compress individual files to sizes that are between 1 GB and 5 GB.

B. Use a columnar storage file format.

C. Partition the data based on the most common query predicates.

D. Split the data into files that are less than 10 KB.

E. Use file formats that are not splittable.

Suggested Answer: *BC*

Community vote distribution

BC (100%)

GiorgioGss Highly Voted 👍 1 year, 3 months ago

Selected Answer: BC

https://docs.aws.amazon.com/redshift/latest/dg/c-spectrum-external-performance.html

upvoted 6 times

rralucard_ Highly Voted 👍 1 year, 4 months ago

Selected Answer: BC

B. Use a columnar storage file format: This is an excellent approach. Columnar storage formats like Parquet and ORC are highly recommended for use with Redshift Spectrum. They store data in columns, which allows Spectrum to scan only the needed columns for a query, significantly improving query performance and reducing the amount of data scanned.

C. Partition the data based on the most common query predicates: Partitioning data in S3 based on commonly used query predicates (like date, region, etc.) allows Redshift Spectrum to skip large portions of data that are irrelevant to a particular query. This can lead to substantial performance improvements, especially for large datasets.

upvoted 5 times

andrologin Most Recent ⏲ 11 months, 3 weeks ago

Selected Answer: BC

Partioning helps filter the data and columnar storage is optimised for analytical (OLAP) queries

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: BC

Redshift Spectrum is optimized for querying data stored in columnar formats like Parquet or ORC.
These formats store each data column separately, allowing Redshift Spectrum to only scan the relevant columns for a specific query, significantly improving performance compared to row-oriented formats
Partitioning organizes data files in S3 based on specific column values (e.g., date,
region). When your queries filter or join data based on these partitioning columns (common query predicates), Redshift Spectrum can quickly locate the relevant data files, minimizing the amount of data scanned and accelerating query execution

upvoted 3 times

d8945a1 1 year, 1 month ago

Selected Answer: BC

https://aws.amazon.com/blogs/big-data/10-best-practices-for-amazon-redshift-spectrum/

upvoted 1 times

certplan 1 year, 3 months ago

2. **Partitioning**:
AWS documentation for Amazon Redshift Spectrum highlights the importance of partitioning data based on commonly used query predicates to improve query performance. By partitioning data, Redshift Spectrum can prune unnecessary partitions during query execution, reducing the amount

of data scanned and improving overall query performance. This guidance can be found in the AWS documentation for Amazon Redshift Spectrum under "Using Partitioning to Improve Query Performance": https://docs.aws.amazon.com/redshift/latest/dg/c-using-spectrum-partitioning.html

upvoted 1 times

⊟ **certplan** 1 year, 3 months ago

1. **Columnar Storage File Format**:

According to AWS documentation, columnar storage file formats like Apache Parquet and Apache ORC are recommended for optimizing query performance with Amazon Redshift Spectrum. They state that these formats are highly efficient for selective column reads, which aligns with the way analytical queries typically operate. This can be found in the AWS documentation for Amazon Redshift Spectrum under "Choosing Data Formats": https://docs.aws.amazon.com/redshift/latest/dg/c-using-spectrum.html#spectrum-columnar-storage

upvoted 1 times

Question #32 *Topic 1*

A company uses Amazon RDS to store transactional data. The company runs an RDS DB instance in a private subnet. A developer wrote an AWS Lambda function with default settings to insert, update, or delete data in the DB instance.

The developer needs to give the Lambda function the ability to connect to the DB instance privately without using the public internet.

Which combination of steps will meet this requirement with the LEAST operational overhead? (Choose two.)

A. Turn on the public access setting for the DB instance.

B. Update the security group of the DB instance to allow only Lambda function invocations on the database port.

C. Configure the Lambda function to run in the same subnet that the DB instance uses.

D. Attach the same security group to the Lambda function and the DB instance. Include a self-referencing rule that allows access through the database port.

E. Update the network ACL of the private subnet to include a self-referencing rule that allows access through the database port.

Suggested Answer: *CD*

Community vote distribution

CD (63%) BD (21%) BC (16%)

Alagong Highly Voted 1 year, 3 months ago

Selected Answer: CD

This solution only modifies the inbound rules of the security group of the DB instance, but it does not modify the outbound rules of the security group of the Lambda function. Additionally, this solution does not facilitate a private connection from the Lambda function to the DB instance, hence, the Lambda function would still need to use the public internet to access the DB instance. Therefore, this option does not fulfill the requirements.

upvoted 6 times

rr01 Most Recent 5 months ago

Selected Answer: BD

I would go with B & D. As C would have operational overhead in my opinion.

upvoted 1 times

altonh 6 months, 4 weeks ago

Selected Answer: BC

D is wrong. It is a bad security practice for a DB to share SG with the client.

C is correct compared to the other opinions (A & E).

upvoted 1 times

proserv 8 months, 3 weeks ago

Selected Answer: BD

B & D

C is wrong

While you want the Lambda function to access the RDS instance privately, it does not need to run in the same subnet. As long as both are in the same VPC, the Lambda function can connect.

upvoted 1 times

tgv 1 year ago

Selected Answer: CD

I will go with C and D on this one, because in my opinion B is not correctly phrased.

The correct way to phrase it would be something like:

Update the security group of the RDS instance to allow inbound traffic on the database port (3306) only from the security group associated with the Lambda function.

upvoted 2 times

sdas1 1 year, 1 month ago

While placing the Lambda function in the same subnet as the DB instance would technically allow them to communicate privately within the same network, it introduces additional complexity and operational overhead. Lambda functions typically run in AWS-managed VPCs, and configuring them

to run in a specific subnet might require manual intervention and ongoing maintenance.
upvoted 2 times

sdas1 1 year, 1 month ago

Moreover, running a Lambda function within a subnet does not inherently ensure private connectivity to the RDS instance. Additional networking configurations would still be needed to allow the Lambda function to access the RDS instance securely, such as configuring the appropriate security groups and potentially adjusting network ACLs.
Hence C can't be the answer
upvoted 2 times

Snape 1 year, 2 months ago

Selected Answer: BD

bbb ddd
upvoted 3 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: CD

I would go with CD, since it's less operational effort, in my opinion
upvoted 1 times

arvehisa 1 year, 3 months ago

Selected Answer: CD

B: need update security group. and there there may be other application need to access db except for lambda function
D: it works and reuse security group which has less operational overhead
upvoted 4 times

harrura 1 year, 3 months ago

A is not an option as it exposes the data to public
B is not an option as we don't want the lambda to be the only entity accessing the db, there can be many other apps. doing this is not scalable
upvoted 2 times

certplan 1 year, 3 months ago

B. - While updating the security group of the DB instance to allow only Lambda function invocations on the database port may seem like a viable solution, it's not the most efficient approach. This option overlooks the need for the Lambda function to be able to communicate securely with the DB instance within the same VPC/subnet.
- Reference: [Amazon RDS documentation on security groups]
(https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithSecurityGroups.html)
upvoted 1 times

certplan 1 year, 3 months ago

- AWS Lambda supports VPC configurations, allowing you to run Lambda functions within your own VPC. This enables private connectivity between Lambda functions and resources within the VPC, such as RDS DB instances.
Reference AWS Lambda documentation on VPC configurations: [AWS Lambda VPC Settings]https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html

- AWS security groups provide a flexible and scalable way to control traffic to your instances or resources. By attaching the same security group to both the Lambda function and the RDS DB instance, you can ensure they share the same set of rules for inbound and outbound traffic.
- Self-referencing rules within security groups enable instances within the same security group to communicate with each other over specified ports.
- Reference AWS documentation on security groups and self-referencing rules: [Security Groups for Your VPC]https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html
upvoted 1 times

certplan 1 year, 3 months ago

So, there coudl be a justified argument for the following:

C. Configure the Lambda function to run in the same subnet that the DB instance uses:
By running the Lambda function in the same subnet as the RDS DB instance, you enable them to communicate privately within the same network, eliminating the need for public internet access and reducing operational overhead.

D. Attach the same security group to the Lambda function and the DB instance. Include a self-referencing rule that allows access through the database port:
By attaching the same security group to both the Lambda function and the RDS DB instance, and including a self-referencing rule that allows access

through the database port, you ensure secure communication between them within the same VPC without exposing the database to the public internet. This approach minimizes operational overhead by centralizing security management and simplifying access control.

upvoted 4 times

certplan 1 year, 3 months ago

Here's how you would implement this:

1. **Attach the same security group to both the Lambda function and the RDS DB instance**: Ensure that both resources are associated with the same security group.

2. **Create an inbound rule in the security group**: Configure the security group to allow inbound traffic on the database port (e.g., 3306 for MySQL) from the security group itself.

For example, if the security group ID is sg-1234567890 and the database port is 3306, the inbound rule would look something like this:

Type: Custom TCP Rule
Protocol: TCP
Port Range: 3306 (or the port your database uses)
Source: sg-1234567890 (the security group ID itself)

This rule allows the Lambda function, which is also part of the same security group, to communicate with the RDS DB instance through the specified port. It effectively creates a loopback or self-referencing rule within the security group, allowing internal communication between resources while maintaining security boundaries.

upvoted 1 times

certplan 1 year, 3 months ago

The phrase "Include a self-referencing rule that allows access through the database port" refers to configuring the security group associated with the resources (in this case, the Lambda function and the RDS DB instance) to allow inbound traffic from the resources themselves on a specific port, typically the port used for database communication.

In AWS security groups, a self-referencing rule means allowing traffic from the security group itself. This setup is often used to facilitate communication between resources within the same security group or VPC without needing to specify individual IP addresses.

upvoted 2 times

samadal 10 months, 2 weeks ago

Thank you. You always help me solve my problems.

upvoted 1 times

GiorgioGss 1 year, 3 months ago

Selected Answer: BC

When you want Lambda to "privately" connect to a resource (RDS in this case) that sits inside a VPC, then you deploy Lambda inside VPC. = C
Then you attach a proper IAM role to lambda.
Then, to be more secure you open the RDS security group only on the specific port:
MySQL/Aurora MySQL: 3306
SQL Server: 1433
PostgreSQL: 5432
Oracle: 1521

upvoted 1 times

BartoszGolebiowski24 1 year, 4 months ago

what does "Include a self-referencing rule that allows access through the database port." mean?

upvoted 1 times

Question #33 *Topic 1*

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. Deploy a custom Python script on an Amazon Elastic Container Service (Amazon ECS) cluster.

B. Create an AWS Lambda Python function with provisioned concurrency.

C. Deploy a custom Python script that can integrate with API Gateway on Amazon Elastic Kubernetes Service (Amazon EKS).

D. Create an AWS Lambda function. Ensure that the function is warm by scheduling an Amazon EventBridge rule to invoke the Lambda function every 5 minutes by using mock events.

Suggested Answer: *B*

Community vote distribution

B (100%)

Mperu08 2 months, 2 weeks ago

Selected Answer: D

The solution with the least operational overhead is D. Create an AWS Lambda function with an EventBridge rule to keep it warm. Lambda handles the infrastructure management automatically, and the warm-up strategy addresses potential cold start issues while maintaining minimal operational requirements compared to container-based solutions.

upvoted 1 times

MephiboshethGumani 3 months, 2 weeks ago

Selected Answer: D

D. Create an AWS Lambda function. Ensure that the function is warm by scheduling an Amazon EventBridge rule to invoke the Lambda function every 5 minutes by using mock events.

upvoted 2 times

royalrum 8 months ago

I AM THINKING B. Dont u think provisionning concurrency add additional cost even when the function is not in active use, which is unnecessary for an occasionally invoked function.

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: B

B and D are both ok. Still, since it says LEAST operational overhead, then keep it simple. B then.

upvoted 4 times

HunkyBunky 1 year, 2 months ago

Selected Answer: B

B - simple and clear

upvoted 2 times

lucas_rfsb 1 year, 3 months ago

Selected Answer: B

I would go in B

upvoted 1 times

GiorgioGss 1 year, 3 months ago

Selected Answer: B

Although D seems a good choice but the questions asks for "LEAST operational overhead" will result in B

upvoted 1 times

damaldon 1 year, 4 months ago

Answ. B

You can create a web API with an HTTP endpoint for your Lambda function by using Amazon API Gateway. API Gateway provides tools for creating and documenting web APIs that route HTTP requests to Lambda functions. You can secure access to your API with authentication and authorization

controls. Your APIs can serve traffic over the internet or can be accessible only within your VPC.
https://docs.aws.amazon.com/lambda/latest/dg/services-apigateway.html

upvoted 1 times

rralucard_ 1 year, 5 months ago

Selected Answer: B

B.
AWS Lambda functions can be easily integrated with Amazon API Gateway to create RESTful APIs. This integration allows API Gateway to directly invoke the Lambda function when the API endpoint is hit.

upvoted 2 times

Question #34 *Topic 1*

A company has a production AWS account that runs company workloads. The company's security team created a security AWS account to store and analyze security logs from the production AWS account. The security logs in the production AWS account are stored in Amazon CloudWatch Logs.
The company needs to use Amazon Kinesis Data Streams to deliver the security logs to the security AWS account.
Which solution will meet these requirements?

A. Create a destination data stream in the production AWS account. In the security AWS account, create an IAM role that has cross-account permissions to Kinesis Data Streams in the production AWS account.

B. Create a destination data stream in the security AWS account. Create an IAM role and a trust policy to grant CloudWatch Logs the permission to put data into the stream. Create a subscription filter in the security AWS account.

C. Create a destination data stream in the production AWS account. In the production AWS account, create an IAM role that has cross-account permissions to Kinesis Data Streams in the security AWS account.

D. Create a destination data stream in the security AWS account. Create an IAM role and a trust policy to grant CloudWatch Logs the permission to put data into the stream. Create a subscription filter in the production AWS account.

Suggested Answer: *D*

Community vote distribution

D (100%)

Christina666 Highly Voted 8 months, 2 weeks ago

Selected Answer: D

Cross-Account Delivery: Kinesis Data Streams in the security account ensures the logs reside in the designated security-focused environment.
CloudWatch Logs Integration: Granting CloudWatch Logs permissions to put records into the Kinesis Data Stream directly establishes a streamlined and secure data flow from the production account.
Filtering Controls: The subscription filter in the production account provides precise control over which log events are sent to the security account.

upvoted 6 times

Salam9 Most Recent 5 months ago

Selected Answer: D

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters-AccountLevel.html#DestinationKinesisExample-AccountLevel

upvoted 2 times

certplan 9 months, 1 week ago

1. **Cross-Account Access:**
- AWS Documentation: [Cross-Account Access]
https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html
- This documentation provides detailed instructions on how to set up cross-account access using IAM roles and trust policies, which is essential for allowing CloudWatch Logs in one AWS account to put data into a Kinesis Data Stream in another AWS account.

2. **Configuring CloudWatch Logs Subscription Filters:**
- AWS Documentation: [Subscription Filters for Amazon CloudWatch Logs]
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html
- This documentation explains how to create subscription filters for CloudWatch Logs, which enable you to route log data to various destinations, including Kinesis Data Streams. Placing the subscription filter in the production AWS account ensures that only the relevant security logs are sent to the Kinesis Data Stream in the security AWS account.

upvoted 2 times

GiorgioGss 9 months, 3 weeks ago

Selected Answer: D

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CrossAccountSubscriptions-Kinesis.html

upvoted 2 times

Aesthet 10 months, 4 weeks ago

Both ChatGPT and me agree with anser D

upvoted 1 times

Question #35 *Topic 1*

A company uses Amazon S3 to store semi-structured data in a transactional data lake. Some of the data files are small, but other data files are tens of terabytes.

A data engineer must perform a change data capture (CDC) operation to identify changed data from the data source. The data source sends a full snapshot as a JSON file every day and ingests the changed data into the data lake.

Which solution will capture the changed data MOST cost-effectively?

A. Create an AWS Lambda function to identify the changes between the previous data and the current data. Configure the Lambda function to ingest the changes into the data lake.

B. Ingest the data into Amazon RDS for MySQL. Use AWS Database Migration Service (AWS DMS) to write the changed data to the data lake.

C. Use an open source data lake format to merge the data source with the S3 data lake to insert the new data and update the existing data.

D. Ingest the data into an Amazon Aurora MySQL DB instance that runs Aurora Serverless. Use AWS Database Migration Service (AWS DMS) to write the changed data to the data lake.

Suggested Answer: *C*

Community vote distribution

C (100%)

GiorgioGss Highly Voted 👍 1 year, 3 months ago

Selected Answer: C

https://aws.amazon.com/blogs/big-data/implement-a-cdc-based-upsert-in-a-data-lake-using-apache-iceberg-and-aws-glue/

upvoted 7 times

plutonash Most Recent 5 months, 2 weeks ago

Selected Answer: A

Generally, AWS questions never give preference to the others solution than an AWS service so even if C could be better the answer is A

upvoted 1 times

> **Juan_pc** 2 months, 1 week ago
>
> But some files are tens of terabytes, and Lamda has a time windows of 15 minutes, that could not be enougth time to process big data
>
> upvoted 2 times

influxy 10 months, 3 weeks ago

https://aws.amazon.com/blogs/big-data/choosing-an-open-table-format-for-your-transactional-data-lake-on-aws/

upvoted 1 times

FunkyFresco 1 year, 1 month ago

Selected Answer: C

Ill go with Delta or something like that. is C

upvoted 2 times

certplan 1 year, 3 months ago

Relative to cost, here are docs for the reason for option C:

https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html

https://aws.amazon.com/blogs/big-data/

https://docs.aws.amazon.com/glue/latest/dg/welcome.html

https://docs.aws.amazon.com/emr/

Here are docs for reasons the others are not correct:

https://aws.amazon.com/lambda/pricing/

https://aws.amazon.com/rds/pricing/

https://aws.amazon.com/dms/pricing/

upvoted 2 times

damaldon 1 year, 4 months ago

Answ. D

You can migrate data from any MySQL-compatible database (MySQL, MariaDB, or Amazon Aurora MySQL) using AWS Database Migration Service.

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.MySQL.html
upvoted 1 times

⊟ **Juan_pc** 2 months, 1 week ago

D is not the best cost effectively solution

upvoted 1 times

⊟ **GiorgioGss** 1 year, 3 months ago

"other data files are tens of terabytes" - good luck with DMS on that :) I think it's C

upvoted 6 times

⊟ **[Removed]** 1 year, 5 months ago

Selected Answer: C

This is a tricky one. Although option A seems like the best choice since it uses an AWS service, I believe using Delta/Iceberg APIs would be easier than writing custom code on Lambda

upvoted 4 times

⊟ **Houyon** 1 year, 4 months ago

If all files were small I believe it would be a great idea. However, you wouldn't be able to compare heavy files with lambda due to its memory/capacity and runtime constraints

upvoted 4 times

Question #36 *Topic 1*

A data engineer runs Amazon Athena queries on data that is in an Amazon S3 bucket. The Athena queries use AWS Glue Data Catalog as a metadata table.

The data engineer notices that the Athena query plans are experiencing a performance bottleneck. The data engineer determines that the cause of the performance bottleneck is the large number of partitions that are in the S3 bucket. The data engineer must resolve the performance bottleneck and reduce Athena query planning time.

Which solutions will meet these requirements? (Choose two.)

A. Create an AWS Glue partition index. Enable partition filtering.

B. Bucket the data based on a column that the data have in common in a WHERE clause of the user query.

C. Use Athena partition projection based on the S3 bucket prefix.

D. Transform the data that is in the S3 bucket to Apache Parquet format.

E. Use the Amazon EMR S3DistCP utility to combine smaller objects in the S3 bucket into larger objects.

Suggested Answer: *AC*

Community vote distribution

AC (69%) AD (19%) 13%

rralucard_ Highly Voted 11 months ago

Selected Answer: AC

https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/

Optimizing Partition Processing using partition projection

Processing partition information can be a bottleneck for Athena queries when you have a very large number of partitions and aren't using AWS Glue partition indexing. You can use partition projection in Athena to speed up query processing of highly partitioned tables and automate partition management. Partition projection helps minimize this overhead by allowing you to query partitions by calculating partition information rather than retrieving it from a metastore. It eliminates the need to add partitions' metadata to the AWS Glue table.

upvoted 7 times

Mahidbdwh Most Recent 4 months, 2 weeks ago

Selected Answer: AC

Bucketing not address the problem of having a large number of partitions in the metadata, which is the root cause of the query planning bottleneck.

Converting to a columnar format like Apache Parquet will not directly reduce the overhead associated with managing a large number of partitions.

Combining small objects will not mitigate the planning overhead that comes from a large number of partitions in the data catalog. Hence A and C

upvoted 2 times

SMALLAM 5 months, 2 weeks ago

Selected Answer: AE

https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/

upvoted 1 times

pypelyncar 6 months, 3 weeks ago

Selected Answer: AC

Creating an AWS Glue partition index and enabling partition filtering can significantly improve query performance when dealing with large datasets with many partitions. The partition index allows Athena to quickly identify the relevant partitions for a query, reducing the time spent scanning unnecessary data. Partition filtering further optimizes the query by only scanning the partitions that match the filter conditions.

Athena partition projection based on the S3 bucket prefix is another effective technique to improve query performance. By leveraging the bucket prefix structure, Athena can prune partitions that are not relevant to the query, reducing the amount of data that needs to be scanned and processed. This approach is particularly useful when the data is organized in a hierarchical structure within the S3 bucket.

upvoted 1 times

VerRi 7 months, 1 week ago

Selected Answer: AC

D is not correct because the issue is related to partitioning.

upvoted 1 times

HunkyBunky 8 months ago

Selected Answer: AC

I guess A / C, beucase we faced with - query plans performance bottleneck, so indexing should be improved

upvoted 1 times

khchan123 8 months ago

A. Creating an AWS Glue partition index and enabling partition filtering can help improve query performance by allowing Athena to prune unnecessary partitions from the query plan. This can reduce the number of partitions that need to be scanned, resulting in faster query planning times.

C. Athena partition projection allows you to define a partition scheme based on the S3 bucket prefix. This can help reduce the number of partitions that need to be scanned, as Athena can use the prefix to determine which partitions are relevant to the query. This can also help improve query performance and reduce planning times.

upvoted 2 times

okechi 8 months, 2 weeks ago

The right answer is BD

upvoted 1 times

Christina666 8 months, 2 weeks ago

Selected Answer: AD

A. Create an AWS Glue partition index. Enable partition filtering.
Targeted Optimization: Partition indexes within the Glue Data Catalog help Athena efficiently identify the relevant partitions, significantly reducing query planning time. Partition filtering further refines the search during query execution.
D. Transform the data that is in the S3 bucket to Apache Parquet format.
Efficient Columnar Format: Parquet's columnar storage and built-in metadata often allow Athena to skip over large portions of data irrelevant to the query, leading to faster query planning and execution.

upvoted 3 times

fceb2c1 9 months, 1 week ago

Selected Answer: AC

Keyword: Athena query planning time

See explanation in the link:
https://www.myexamcollection.com/Data-Engineer-Associate-vce-questions.htm

B & D are related to analytical queries performance, not about "query planning" performance.

upvoted 4 times

ottarg 9 months, 2 weeks ago

Just finished the exam and I went with AD. I agree with GiorgioGss, but the reason why I picked A over C was becaues the table is already using Glue catalog.
If we use the indexes, there's no reason to use C as we already have the partitions indexed.
No reason to pick B if we have C selected.
Thus I picked D with this to optimize the query e.g. if I'm only selecting a subset of the columns.

upvoted 2 times

GiorgioGss 9 months, 3 weeks ago

Strange questions.... it can be ABCD

upvoted 1 times

rralucard_ 11 months ago

If your table stored in an AWS Glue Data Catalog has tens and hundreds of thousands and millions of partitions, you can enable partition indexes on the table. With partition indexes, only the metadata for the partition value in the query's filter is retrieved from the catalog instead of retrieving all the partitions' metadata. The result is faster queries for such highly partitioned tables. The following table compares query runtimes between a partitioned table with no partition indexing and with partition indexing. The table contains approximately 100,000 partitions and uncompressed text data. The orders table is partitioned by the o_custkey column.

upvoted 1 times

[Removed] 11 months, 2 weeks ago

Selected Answer: BD

https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/

upvoted 2 times

Question #37 *Topic 1*

A data engineer must manage the ingestion of real-time streaming data into AWS. The data engineer wants to perform real-time analytics on the incoming streaming data by using time-based aggregations over a window of up to 30 minutes. The data engineer needs a solution that is highly fault tolerant.

Which solution will meet these requirements with the LEAST operational overhead?

A. Use an AWS Lambda function that includes both the business and the analytics logic to perform time-based aggregations over a window of up to 30 minutes for the data in Amazon Kinesis Data Streams.

B. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to analyze the data that might occasionally contain duplicates by using multiple types of aggregations.

C. Use an AWS Lambda function that includes both the business and the analytics logic to perform aggregations for a tumbling window of up to 30 minutes, based on the event timestamp.

D. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to analyze the data by using multiple types of aggregations to perform time-based analytics over a window of up to 30 minutes.

Suggested Answer: *D*

Community vote distribution

D (100%)

rralucard_ Highly Voted 👍 1 year, 4 months ago

Selected Answer: D

D. Amazon Managed Service for Apache Flink for Time-Based Analytics over 30 Minutes: This option correctly identifies the use of Amazon Managed Service for Apache Flink for performing time-based analytics over a window of up to 30 minutes. Apache Flink is adept at handling such scenarios, providing capabilities for complex event processing, time-windowed aggregations, and maintaining state over time. This option would offer high fault tolerance and minimal operational overhead due to the managed nature of the service.

upvoted 7 times

div_div Most Recent 🕒 5 months, 1 week ago

Selected Answer: D

Lambda can not be used because it max processing limit of time is 15 min and remaining two option related to flink and using flink we can perfrom time-series and window size aggregation

upvoted 2 times

Linuslin 11 months, 1 week ago

This link is not AWS documents but I think you guys can take a look.

https://amandeep-singh-johar.medium.com/real-time-stream-processing-with-apache-flink-153992840f16

upvoted 2 times

Just_Ninja 1 year, 1 month ago

Selected Answer: D

Show the Docs

upvoted 2 times

DevoteamAnalytix 1 year, 1 month ago

Selected Answer: D

https://docs.aws.amazon.com/managed-flink/latest/java/how-operators.html#how-operators-agg

upvoted 1 times

harrura 1 year, 3 months ago

this is crazy, the answers by bot are wrong, please don't rely on them. please care to open discussions and look for reasoning

upvoted 2 times

Question #38 *Topic 1*

A company is planning to upgrade its Amazon Elastic Block Store (Amazon EBS) General Purpose SSD storage from gp2 to gp3. The company wants to prevent any interruptions in its Amazon EC2 instances that will cause data loss during the migration to the upgraded storage.
Which solution will meet these requirements with the LEAST operational overhead?

A. Create snapshots of the gp2 volumes. Create new gp3 volumes from the snapshots. Attach the new gp3 volumes to the EC2 instances.

B. Create new gp3 volumes. Gradually transfer the data to the new gp3 volumes. When the transfer is complete, mount the new gp3 volumes to the EC2 instances to replace the gp2 volumes.

C. Change the volume type of the existing gp2 volumes to gp3. Enter new values for volume size, IOPS, and throughput.

D. Use AWS DataSync to create new gp3 volumes. Transfer the data from the original gp2 volumes to the new gp3 volumes.

Suggested Answer: *C*

Community vote distribution

C (100%)

GiorgioGss Highly Voted 👍 9 months, 2 weeks ago

Selected Answer: C

https://aws.amazon.com/blogs/storage/migrate-your-amazon-ebs-volumes-from-gp2-to-gp3-and-save-up-to-20-on-costs/

upvoted 6 times

fceb2c1 Highly Voted 👍 9 months, 1 week ago

Selected Answer: C

Option C: Check section under "To modify an Amazon EBS volume using the AWS Management Console“ in GiorgioGss's link
Amazon EBS Elastic Volumes enable you to modify your volume type from gp2 to gp3 without detaching volumes or restarting instances (requirements for modification), which means that there are no interruptions to your applications during modification.

upvoted 6 times

lcsantos99 Most Recent 🕒 5 months ago

Selected Answer: C

the correct answer is C

https://aws.amazon.com/pt/blogs/storage/migrate-your-amazon-ebs-volumes-from-gp2-to-gp3-and-save-up-to-20-on-costs/

upvoted 1 times

rralucard_ 11 months ago

Selected Answer: C

Option C is the most straightforward and efficient approach to upgrading from gp2 to gp3 EBS volumes, providing an in-place upgrade path with minimal operational overhead and no interruption in service.

upvoted 2 times

Question #39 *Topic 1*

A company is migrating its database servers from Amazon EC2 instances that run Microsoft SQL Server to Amazon RDS for Microsoft SQL Server DB instances. The company's analytics team must export large data elements every day until the migration is complete. The data elements are the result of SQL joins across multiple tables. The data must be in Apache Parquet format. The analytics team must store the data in Amazon S3.
Which solution will meet these requirements in the MOST operationally efficient way?

A. Create a view in the EC2 instance-based SQL Server databases that contains the required data elements. Create an AWS Glue job that selects the data directly from the view and transfers the data in Parquet format to an S3 bucket. Schedule the AWS Glue job to run every day.

B. Schedule SQL Server Agent to run a daily SQL query that selects the desired data elements from the EC2 instance-based SQL Server databases. Configure the query to direct the output .csv objects to an S3 bucket. Create an S3 event that invokes an AWS Lambda function to transform the output format from .csv to Parquet.

C. Use a SQL query to create a view in the EC2 instance-based SQL Server databases that contains the required data elements. Create and run an AWS Glue crawler to read the view. Create an AWS Glue job that retrieves the data and transfers the data in Parquet format to an S3 bucket. Schedule the AWS Glue job to run every day.

D. Create an AWS Lambda function that queries the EC2 instance-based databases by using Java Database Connectivity (JDBC). Configure the Lambda function to retrieve the required data, transform the data into Parquet format, and transfer the data into an S3 bucket. Use Amazon EventBridge to schedule the Lambda function to run every day.

Suggested Answer: *C*

Community vote distribution

C (61%) A (39%)

⊟ **taka5094** Highly Voted 1 year, 3 months ago

Selected Answer: C

Choice A) is almost the same approach, but it doesn't use the AWS Glue crawler, so have to manage the view's metadata manually.

upvoted 7 times

⊟ **michele_scar** 7 months, 3 weeks ago

My fault: is correct A because during a migration process is more efficiently have a crawler that should catch eventually changes of a schema.

upvoted 2 times

⊟ **michele_scar** 7 months, 3 weeks ago

My fault: is correct C because during a migration process is more efficiently have a crawler that should catch eventually changes of a schema.

upvoted 2 times

⊟ **Christina666** Highly Voted 1 year, 2 months ago

Selected Answer: C

Leveraging SQL Views: Creating a view on the source database simplifies the data extraction process and keeps your SQL logic centralized.
Glue Crawler Efficiency: Using a Glue crawler to automatically discover and catalog the view's metadata reduces manual setup.
Glue Job for ETL: A dedicated Glue job is well-suited for the data transformation (to Parquet) and loading into S3. Glue jobs offer built-in scheduling capabilities.
Operational Efficiency: This approach minimizes custom code and leverages native AWS services for data movement and cataloging.

upvoted 7 times

⊟ **Dummy92yash** 10 months, 1 week ago

Glue crawler is used to catalog and find the schema. In this requirement the data was already stored in MS SQL server which a relational database. Hence I think A is correct

upvoted 4 times

⊟ **Tester_TKK** Most Recent 2 months, 1 week ago

Selected Answer: A

Option C in incorrect because it adds a Glue crawler to read the view, which is redundant if the schema is already defined in the view

upvoted 1 times

⊟ **Tester_TKK** 2 months, 1 week ago

Selected Answer: A

Crawler not needed as the schema is already in the view

upvoted 1 times

Mperu08 2 months, 2 weeks ago

Selected Answer: A

Uses AWS Glue, a serverless ETL service optimized for large-scale data processing and Parquet output. The view simplifies query logic, and scheduling is straightforward. No EC2 dependency, minimal maintenance, and distributed processing ensure efficiency.

upvoted 1 times

Eltanany 3 months, 1 week ago

Selected Answer: A

I'll go with A

upvoted 1 times

Certified101 4 months, 2 weeks ago

Selected Answer: A

A is correct - no need for crawler

upvoted 1 times

plutonash 5 months, 2 weeks ago

Selected Answer: A

the scrawler is not necessary, use GLUE job to read data from sql server and transfert to S3 with Apache Parquet format is enough.

upvoted 3 times

mtrianac 6 months, 3 weeks ago

Selected Answer: A

No, in this case, using an AWS Glue Crawler is not necessary. The schema is already defined in the SQL Server database, as the created view contains the required structure (columns and data types). AWS Glue can directly connect to the database via JDBC, extract the data, transform it into Parquet format, and store it in S3 without additional steps.

A crawler is useful if you're working with data that doesn't have a predefined schema (e.g., files in S3) or if you need the data to be cataloged for services like Amazon Athena. However, for this ETL flow, using just a Glue Job simplifies the process and reduces operational complexity.

upvoted 3 times

michele_scar 7 months, 3 weeks ago

Selected Answer: A

Glue crawler is useless because the schema is already in place with a SQL database

upvoted 1 times

michele_scar 7 months, 3 weeks ago

My fault: is correct A because during a migration process is more efficiently have a crawler that should catch eventually changes of a schema.

upvoted 1 times

michele_scar 7 months, 3 weeks ago

My fault: is correct C because during a migration process is more efficiently have a crawler that should catch eventually changes of a schema.

upvoted 1 times

leonardoFelipe 7 months, 4 weeks ago

Selected Answer: A

Usually, views aren't true objects in a SGBD, they're just a "nickname" for a specific query string, different of Materialized Views. So, my questions is: can glue crawler understand their metadata?

I'd go with A

upvoted 3 times

bakarys 12 months ago

Selected Answer: A

Option A involves creating a view in the EC2 instance-based SQL Server databases that contains the required data elements. An AWS Glue job is then created to select the data directly from the view and transfer the data in Parquet format to an S3 bucket. This job is scheduled to run every day. This approach is operationally efficient as it leverages managed services (AWS Glue) and does not require additional transformation steps.

Option D involves creating an AWS Lambda function that queries the EC2 instance-based databases using JDBC. The Lambda function is configured to retrieve the required data, transform the data into Parquet format, and transfer the data into an S3 bucket. This approach could work, but managing and scheduling Lambda functions could add operational overhead compared to using managed services like AWS Glue.

upvoted 3 times

GiorgioGss 1 year, 3 months ago

Selected Answer: C

Just beacuse it decouples the whole architecture I will go with C

upvoted 2 times

Felix_G 1 year, 4 months ago

Option C seems to be the most operationally efficient:

It leverages Glue for both schema discovery (via the crawler) and data transfer (via the Glue job).

The Glue job can directly handle the Parquet format conversion.

Scheduling the Glue job ensures regular data export without manual intervention.

upvoted 1 times

helpaws 1 year, 3 months ago

you're right: https://aws.amazon.com/blogs/big-data/extracting-multidimensional-data-from-microsoft-sql-server-analysis-services-using-aws-glue/

upvoted 1 times

taka5094 1 year, 3 months ago

Is this right?

https://aws.amazon.com/jp/blogs/big-data/extracting-multidimensional-data-from-microsoft-sql-server-analysis-services-using-aws-glue/

upvoted 1 times

rralucard_ 1 year, 4 months ago

Selected Answer: A

Option A (Creating a view in the EC2 instance-based SQL Server databases and creating an AWS Glue job that selects data from the view, transfers it in Parquet format to S3, and schedules the job to run every day) seems to be the most operationally efficient solution. It leverages AWS Glue's ETL capabilities for direct data extraction and transformation, minimizes manual steps, and effectively automates the process.

upvoted 3 times

evntdrvn76 1 year, 4 months ago

A. Create a view in the EC2 instance-based SQL Server databases that contains the required data elements. Create an AWS Glue job that selects the data directly from the view and transfers the data in Parquet format to an S3 bucket. Schedule the AWS Glue job to run every day. This solution is operationally efficient for exporting data in the required format.

upvoted 2 times

Question #40 *Topic 1*

A data engineering team is using an Amazon Redshift data warehouse for operational reporting. The team wants to prevent performance issues that might result from long- running queries. A data engineer must choose a system table in Amazon Redshift to record anomalies when a query optimizer identifies conditions that might indicate performance issues.
Which table views should the data engineer use to meet this requirement?

A. STL_USAGE_CONTROL

B. STL_ALERT_EVENT_LOG

C. STL_QUERY_METRICS

D. STL_PLAN_INFO

Suggested Answer: *B*

Community vote distribution

B (100%)

Felix_G Highly Voted 10 months ago

B

STL_ALERT_EVENT_LOG records any alerts/notifications related to queries or user-defined performance thresholds. This would capture optimizer alerts about potential performance issues.

STL_PLAN_INFO provides detailed info on execution plans. The optimizer statistics and warnings provide insight into problematic query plans.

STL_USAGE_CONTROL limits user activity but does not log anomalies.

STL_QUERY_METRICS has execution stats but no plan diagnostics.

By enabling alerts and checking STL_ALERT_EVENT_LOG and STL_PLAN_INFO, the data engineer can best detect and troubleshoot queries flagged by the optimizer as problematic before they impair performance. This meets the requirement to catch potential long running queries.

upvoted 5 times

GiorgioGss Highly Voted 9 months, 2 weeks ago

Selected Answer: B

https://docs.aws.amazon.com/redshift/latest/dg/r_STL_ALERT_EVENT_LOG.html

upvoted 5 times

HagarTheHorrible Most Recent 6 months, 1 week ago

Selected Answer: B

Control table is related to usage control metrics and doesn't focus on performance issues or anomalies related to query optimization. It's more about usage limits and controls.

upvoted 1 times

pypelyncar 6 months, 3 weeks ago

Selected Answer: B

this table records alerts that are generated by the Amazon Redshift system when it detects certain conditions that might indicate performance issues. These alerts are triggered by the query optimizer when it detects suboptimal query plans or other issues that could affect performance.

upvoted 1 times

rralucard_ 11 months ago

Selected Answer: B

https://docs.aws.amazon.com/redshift/latest/dg/cm_chap_system-tables.html

STL_ALERT_EVENT_LOG table view to meet this requirement. This system table in Amazon Redshift is designed to record anomalies when a query optimizer identifies conditions that might indicate performance issues

upvoted 1 times

Question #41 *Topic 1*

A data engineer must ingest a source of structured data that is in .csv format into an Amazon S3 data lake. The .csv files contain 15 columns. Data analysts need to run Amazon Athena queries on one or two columns of the dataset. The data analysts rarely query the entire file.
Which solution will meet these requirements MOST cost-effectively?

A. Use an AWS Glue PySpark job to ingest the source data into the data lake in .csv format.

B. Create an AWS Glue extract, transform, and load (ETL) job to read from the .csv structured data source. Configure the job to ingest the data into the data lake in JSON format.

C. Use an AWS Glue PySpark job to ingest the source data into the data lake in Apache Avro format.

D. Create an AWS Glue extract, transform, and load (ETL) job to read from the .csv structured data source. Configure the job to write the data into the data lake in Apache Parquet format.

Suggested Answer: *D*

Community vote distribution

D (100%)

imymoco 7 months, 2 weeks ago

Why not B?
I think Athena also be able to handle json.

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: D

Athena is optimized for querying data stored in Parquet format. It can efficiently scan only the necessary columns for a specific query, reducing the amount of data processed. This translates to faster query execution times and lower query costs for data analysts who primarily focus on one or two columns

upvoted 2 times

FunkyFresco 1 year, 1 month ago

Selected Answer: D

Cost effectively, and they are going to use only one or two columns, columnar.

upvoted 2 times

GiorgioGss 1 year, 3 months ago

Selected Answer: D

MOST cost-effectively = parquet

upvoted 3 times

atu1789 1 year, 5 months ago

Selected Answer: D

Glue + Parquet for cost efectiveness

upvoted 2 times

Question #42 *Topic 1*

A company has five offices in different AWS Regions. Each office has its own human resources (HR) department that uses a unique IAM role. The company stores employee records in a data lake that is based on Amazon S3 storage.
A data engineering team needs to limit access to the records. Each HR department should be able to access records for only employees who are within the HR department's Region.
Which combination of steps should the data engineering team take to meet this requirement with the LEAST operational overhead? (Choose two.)

A. Use data filters for each Region to register the S3 paths as data locations.

B. Register the S3 path as an AWS Lake Formation location.

C. Modify the IAM roles of the HR departments to add a data filter for each department's Region.

D. Enable fine-grained access control in AWS Lake Formation. Add a data filter for each Region.

E. Create a separate S3 bucket for each Region. Configure an IAM policy to allow S3 access. Restrict access based on Region.

Suggested Answer: *BD*

Community vote distribution

BD (100%)

rralucard_ Highly Voted 1 year, 4 months ago

Selected Answer: BD

https://docs.aws.amazon.com/lake-formation/latest/dg/data-filters-about.html
https://docs.aws.amazon.com/lake-formation/latest/dg/access-control-fine-grained.html

upvoted 5 times

ctndba Most Recent 7 months, 2 weeks ago

B: Since its a initial step for leverage fine grain control of Lakeformation.
D: Give granular level control and meets the requirement

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: BD

Registering the S3 path as an AWS Lake Formation location is the first step in leveraging Lake Formation's data governance and access control capabilities. This allows the data engineering team to centrally manage and govern the data stored in the S3 data lake.
Enabling fine-grained access control in AWS Lake Formation and adding a data filter for each Region is the key step to achieve the desired access control. Data filters in Lake Formation allow you to define row-level and column-level access policies based on specific conditions or attributes, such as the Region in this case

upvoted 4 times

rralucard_ 1 year, 5 months ago

If your table stored in an AWS Glue Data Catalog has tens and hundreds of thousands and millions of partitions, you can enable partition indexes on the table. With partition indexes, only the metadata for the partition value in the query's filter is retrieved from the catalog instead of retrieving all the partitions' metadata. The result is faster queries for such highly partitioned tables. The following table compares query runtimes between a partitioned table with no partition indexing and with partition indexing. The table contains approximately 100,000 partitions and uncompressed text data. The orders table is partitioned by the o_custkey column.

upvoted 1 times

atu1789 1 year, 5 months ago

Selected Answer: BD

BD makes sense

upvoted 1 times

Question #43 *Topic 1*

A company uses AWS Step Functions to orchestrate a data pipeline. The pipeline consists of Amazon EMR jobs that ingest data from data sources and store the data in an Amazon S3 bucket. The pipeline also includes EMR jobs that load the data to Amazon Redshift.

The company's cloud infrastructure team manually built a Step Functions state machine. The cloud infrastructure team launched an EMR cluster into a VPC to support the EMR jobs. However, the deployed Step Functions state machine is not able to run the EMR jobs.

Which combination of steps should the company take to identify the reason the Step Functions state machine is not able to run the EMR jobs? (Choose two.)

A. Use AWS CloudFormation to automate the Step Functions state machine deployment. Create a step to pause the state machine during the EMR jobs that fail. Configure the step to wait for a human user to send approval through an email message. Include details of the EMR task in the email message for further analysis.

B. Verify that the Step Functions state machine code has all IAM permissions that are necessary to create and run the EMR jobs. Verify that the Step Functions state machine code also includes IAM permissions to access the Amazon S3 buckets that the EMR jobs use. Use Access Analyzer for S3 to check the S3 access properties.

C. Check for entries in Amazon CloudWatch for the newly created EMR cluster. Change the AWS Step Functions state machine code to use Amazon EMR on EKS. Change the IAM access policies and the security group configuration for the Step Functions state machine code to reflect inclusion of Amazon Elastic Kubernetes Service (Amazon EKS).

D. Query the flow logs for the VPC. Determine whether the traffic that originates from the EMR cluster can successfully reach the data providers. Determine whether any security group that might be attached to the Amazon EMR cluster allows connections to the data source servers on the informed ports.

E. Check the retry scenarios that the company configured for the EMR jobs. Increase the number of seconds in the interval between each EMR task. Validate that each fallback state has the appropriate catch for each decision state. Configure an Amazon Simple Notification Service (Amazon SNS) topic to store the error messages.

Suggested Answer: *BD*

Community vote distribution

BD (87%) 13%

rralucard_ Highly Voted 11 months ago

Selected Answer: BD

https://docs.aws.amazon.com/step-functions/latest/dg/procedure-create-iam-role.html

https://docs.aws.amazon.com/step-functions/latest/dg/service-integration-iam-templates.html

upvoted 5 times

GiorgioGss Highly Voted 9 months, 2 weeks ago

Selected Answer: BD

Permissions of course and we need to see if the traffic is blocked at any hops because they mention that EMR is IN vpc so... flow-logs

upvoted 5 times

sam_pre Most Recent 2 months, 4 weeks ago

Selected Answer: DE

E> As par as I know, Step function does not require S3 access permission that EMR trying to access. so that eliminates E

D and E make sense while E is bit less likely troubleshooting, but still valid

upvoted 1 times

sam_pre 2 months, 4 weeks ago

Sorry for the typo, should be B > ... is eliminated

upvoted 1 times

lucas_rfsb 8 months, 4 weeks ago

Selected Answer: BD

I'd go in BD

upvoted 3 times

kj07 9 months, 2 weeks ago

B&D.

E is not an option to identify the failure reason.

upvoted 1 times

⊟ **atu1789** 11 months ago

Selected Answer: BE

BE. In others are are redflag keywords

upvoted 2 times

⊟ **Tester_TKK** 2 months, 1 week ago

which redflag in E ?

upvoted 1 times

Question #44 *Topic 1*

A company is developing an application that runs on Amazon EC2 instances. Currently, the data that the application generates is temporary. However, the company needs to persist the data, even if the EC2 instances are terminated.
A data engineer must launch new EC2 instances from an Amazon Machine Image (AMI) and configure the instances to preserve the data.
Which solution will meet this requirement?

A. Launch new EC2 instances by using an AMI that is backed by an EC2 instance store volume that contains the application data. Apply the default settings to the EC2 instances.

B. Launch new EC2 instances by using an AMI that is backed by a root Amazon Elastic Block Store (Amazon EBS) volume that contains the application data. Apply the default settings to the EC2 instances.

C. Launch new EC2 instances by using an AMI that is backed by an EC2 instance store volume. Attach an Amazon Elastic Block Store (Amazon EBS) volume to contain the application data. Apply the default settings to the EC2 instances.

D. Launch new EC2 instances by using an AMI that is backed by an Amazon Elastic Block Store (Amazon EBS) volume. Attach an additional EC2 instance store volume to contain the application data. Apply the default settings to the EC2 instances.

Suggested Answer: *C*

Community vote distribution

C (68%) B (33%)

khchan123 Highly Voted 1 year, 2 months ago

Selected Answer: C

CCCCCCC - you need to attach an extra EBS volume

When an instance terminates, the value of the DeleteOnTermination attribute for each attached EBS volume determines whether to preserve or delete the volume. By default, the DeleteOnTermination attribute is set to True for the root volume.
ref: https://repost.aws/knowledge-center/deleteontermination-ebs

upvoted 13 times

hnk Highly Voted 1 year, 1 month ago

Selected Answer: C

C is correct

upvoted 5 times

Chanduchanti Most Recent 4 months, 1 week ago

Selected Answer: C

When an instance terminates, the value of the DeleteOnTermination attribute for each attached EBS volume determines whether to preserve or delete the volume. By default, the DeleteOnTermination attribute is set to True for the root volume.

upvoted 2 times

saransh_001 4 months, 2 weeks ago

Selected Answer: C

Check in the option B and C the default settings are mentioned. By default an EC2 instance whenever terminates, its root volume also gets terminated. So launch new EC2 instances by using an AMI that is backed by an EC2 instance store volume. Attach an Amazon Elastic Block Store (Amazon EBS) volume to contain the application data. Apply the default settings to the EC2 instances.

upvoted 3 times

mohamedTR 8 months, 1 week ago

Selected Answer: C

B: by default, delete on termination is checked

upvoted 5 times

mohamedTR 8 months, 3 weeks ago

Selected Answer: B

By using an AMI backed by an Amazon EBS root volume, you ensure that the application data is preserved, even if the EC2 instances are terminated, because EBS volumes persist independently of the EC2 lifecycle.

upvoted 2 times

ElFaramawi 9 months ago

Selected Answer: B

This is because Amazon EBS volumes are persistent, meaning the data is preserved even if the EC2 instance is terminated, which meets the requirement to persist the data. C is incorrect because it suggests launching instances using an EC2 instance store volume, which is ephemeral. Even though it proposes attaching an Amazon EBS volume for data, the root volume remains an instance store.

upvoted 2 times

portland 10 months, 3 weeks ago

Selected Answer: C

Using default setting means B won't work.

upvoted 3 times

sdas1 11 months, 3 weeks ago

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/preserving-volumes-on-termination.html

Root volume

By default, when you launch an instance the DeleteOnTermination attribute for the root volume of an instance is set to true. Therefore, the default is to delete the root volume of the instance when the instance terminates.

Non-root volume

By default, when you attach a non-root EBS volume to an instance, its DeleteOnTermination attribute is set to false. Therefore, the default is to preserve these volumes.

Answer is C

upvoted 2 times

GustonMari 11 months, 3 weeks ago

Selected Answer: C

its C!!! B with default setting will delete the EBS volume on termination

upvoted 3 times

pypelyncar 1 year ago

Selected Answer: B

Amazon EBS volumes provide persistent block storage for EC2 instances. Data written to an EBS volume is independent of the EC2 instance lifecycle. Even if the EC2 instance is terminated, ***the data on the EBS volume remains intact***. Launching new EC2 instances from an AMI backed by an EBS volume containing the application data ensures the data persists across instance restarts or terminations

upvoted 3 times

VerRi 1 year, 1 month ago

Selected Answer: B

launch EC2 using AMI with root EBS that contains data

upvoted 1 times

ampersandor 1 year, 1 month ago

B: the root EBS volume will be deleted on termination by default.

C: the EBS is independent from EC2 Termination

upvoted 4 times

HunkyBunky 1 year, 1 month ago

Selected Answer: C

C - Looks better, because it will save data in all cases

upvoted 5 times

> **HunkyBunky** 1 year, 1 month ago
>
> And "Delete on Termination" flag by defaults sets to true, so better to use additional volume for application data
>
> upvoted 4 times

Christina666 1 year, 2 months ago

Selected Answer: C

ccccccc

upvoted 5 times

Luke97 1 year, 2 months ago

Can someone explain why C is NOT right?

upvoted 2 times

GiorgioGss 1 year, 3 months ago

Selected Answer: B

This question is more for practitioner exam :)

upvoted 3 times

Question #45 *Topic 1*

A company uses Amazon Athena to run SQL queries for extract, transform, and load (ETL) tasks by using Create Table As Select (CTAS). The company must use Apache Spark instead of SQL to generate analytics.

Which solution will give the company the ability to use Spark to access Athena?

A. Athena query settings

B. Athena workgroup

C. Athena data source

D. Athena query editor

Suggested Answer: *B*

Community vote distribution

B (73%) C (27%)

GiorgioGss Highly Voted 1 year, 3 months ago

Selected Answer: B

https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-getting-started.html

"To use Apache Spark in Amazon Athena, you create an Amazon Athena workgroup that uses a Spark engine."

upvoted 9 times

pypelyncar Highly Voted 1 year ago

Selected Answer: C

The Athena data source acts as a bridge between Athena and other analytics engines, such as Apache Spark. By using the Athena data source connector, you can access data stored in various formats (e.g., CSV, JSON, Parquet) and locations (e.g., Amazon S3, Apache Hive Metastore) through Spark applications

upvoted 5 times

Tester_TKK Most Recent 2 months, 1 week ago

Selected Answer: B

B makes sense

upvoted 1 times

lsj900605 7 months, 2 weeks ago

Selected Answer: B

It is B, not C.

The workgroup is for organizing, controlling, and monitoring queries.

The Data source is the mechanism that enables Spark to query data via Athena. It allows Spark to interact with Athena.

The question focuses on enabling Apache Spark within Athena to generate analytics instead of using SQL. Thus, you must create a Spark-enabled workgroup

upvoted 2 times

theloseralreadytaken 8 months, 1 week ago

Selected Answer: B

Athena datasource doesn't specifially enable Spark access

upvoted 2 times

andrologin 11 months, 3 weeks ago

Selected Answer: B

https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-getting-started.html

To get started with Apache Spark on Amazon Athena, you must first create a Spark enabled workgroup. After you switch to the workgroup, you can create a notebook or open an existing notebook. When you open a notebook in Athena, a new session is started for it automatically and you can work with it directly in the Athena notebook editor.

upvoted 2 times

lalitjhawar 1 year, 1 month ago

C. Athena data source

The Athena data source is a specific connector or library that allows Apache Spark to interact with data stored in Amazon Athena. This connector enables Spark to read data from Athena tables directly into Spark DataFrames or RDDs (Resilient Distributed Datasets), allowing you to perform analytics and transformations using Spark's capabilities.

upvoted 4 times

blackgamer 1 year, 3 months ago

Selected Answer: B

https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-getting-started.html

upvoted 3 times

kj07 1 year, 3 months ago

B is the correct answer.

https://aws.amazon.com/blogs/big-data/explore-your-data-lake-using-amazon-athena-for-apache-spark/

You need an Athena workgroup as a prerequisite to use Apache Spark.

upvoted 1 times

damaldon 1 year, 3 months ago

B. is the correct answer.

To use Apache Spark in Amazon Athena, you create an Amazon Athena workgroup that uses a Spark engine.

https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark-getting-started.html

upvoted 3 times

rralucard_ 1 year, 4 months ago

Selected Answer: C

https://docs.aws.amazon.com/athena/latest/ug/notebooks-spark.html

upvoted 2 times

Question #46 *Topic 1*

A company needs to partition the Amazon S3 storage that the company uses for a data lake. The partitioning will use a path of the S3 object keys in the following format: s3://bucket/prefix/year=2023/month=01/day=01.

A data engineer must ensure that the AWS Glue Data Catalog synchronizes with the S3 storage when the company adds new partitions to the bucket.

Which solution will meet these requirements with the LEAST latency?

A. Schedule an AWS Glue crawler to run every morning.

B. Manually run the AWS Glue CreatePartition API twice each day.

C. Use code that writes data to Amazon S3 to invoke the Boto3 AWS Glue create_partition API call.

D. Run the MSCK REPAIR TABLE command from the AWS Glue console.

Suggested Answer: *C*

Community vote distribution

C (93%) 7%

rralucard_ Highly Voted 11 months ago

Selected Answer: C

Use code that writes data to Amazon S3 to invoke the Boto3 AWS Glue create_partition API call. This approach ensures that the Data Catalog is updated as soon as new data is written to S3, providing the least latency in reflecting new partitions.

upvoted 8 times

Tester_TKK 2 months, 1 week ago

Hey, Did you have some of the ExamTopics questions in the exam?

upvoted 1 times

pypelyncar Most Recent 6 months, 3 weeks ago

Selected Answer: C

By embedding the Boto3 create_partition API call within the code that writes data to S3, you achieve near real-time synchronization. The Data Catalog is updated immediately after a new partition is created in S3.

upvoted 4 times

tgv 7 months ago

Selected Answer: C

The explanation could be more precise regarding the interaction with Amazon S3 and AWS Glue. The key point is that the process should be triggered immediately when new data is added to S3. This can be achieved through event-driven architecture, which indeed makes the solution intuitive and efficient.

upvoted 2 times

valuedate 7 months, 1 week ago

Selected Answer: C

add partition after writing the data in s3

upvoted 1 times

DevoteamAnalytix 7 months, 2 weeks ago

Selected Answer: D

It's about "synchronizing AWS Glue Data Catalog with S3". So for me it's D - using MSCK REPAIR TABLE for existing S3 partitions (https://docs.aws.amazon.com/athena/latest/ug/msck-repair-table.html)

upvoted 1 times

megadba 7 months, 2 weeks ago

Least latency

upvoted 1 times

okechi 8 months, 2 weeks ago

The answer is D

upvoted 2 times

GiorgioGss 9 months, 2 weeks ago

Selected Answer: C

It's pure event-driven so... C

upvoted 1 times

⊟ **atu1789** 11 months ago

Selected Answer: C

C. Least latency

upvoted 1 times

Question #47 *Topic 1*

A media company uses software as a service (SaaS) applications to gather data by using third-party tools. The company needs to store the data in an Amazon S3 bucket. The company will use Amazon Redshift to perform analytics based on the data.
Which AWS service or feature will meet these requirements with the LEAST operational overhead?

A. Amazon Managed Streaming for Apache Kafka (Amazon MSK)

B. Amazon AppFlow

C. AWS Glue Data Catalog

D. Amazon Kinesis

Suggested Answer: *B*

Community vote distribution

B (100%)

tgv Highly Voted 7 months ago

Selected Answer: B

That's exactly the purpose of AppFlow: "fully-managed integration service that enables you to securely exchange data between software as a service (SaaS) applications, such as Salesforce, and AWS services, such as Amazon Simple Storage Service (Amazon S3) and Amazon Redshift. For example, you can ingest contact records from Salesforce to Amazon Redshift or pull support tickets from Zendesk to an Amazon S3 bucket."

https://docs.aws.amazon.com/appflow/latest/userguide/what-is-appflow.html

upvoted 5 times

pypelyncar Most Recent 6 months, 3 weeks ago

Selected Answer: B

the media company can leverage a fully managed service that simplifies the process of ingesting data from their third-party SaaS applications into an Amazon S3 bucket, with minimal operational overhead. Additionally, AppFlow can integrate with Amazon Redshift, allowing the company to load the ingested data directly into their analytics environment for further processing and analysis

upvoted 4 times

GiorgioGss 9 months, 2 weeks ago

Selected Answer: B

https://docs.aws.amazon.com/appflow/latest/userguide/flow-tutorial.html

upvoted 3 times

kj07 9 months, 2 weeks ago

B seems the right choice here.

upvoted 1 times

rralucard_ 11 months ago

Selected Answer: B

https://d1.awsstatic.com/solutions/guidance/architecture-diagrams/integrating-third-party-saas-data-using-amazon-appflow.pdf
Amazon AppFlow is a fully managed integration service that enables you to securely transfer data between Software as a Service (SaaS) applications like Salesforce, Marketo, Slack, and ServiceNow, and AWS services like Amazon S3 and Amazon Redshift, in just a few clicks. It can store the raw data pulled from SaaS applications in Amazon S3, and integrates with AWS Glue Data Catalog to catalog and store metadata

upvoted 2 times

Question #48 *Topic 1*

A data engineer is using Amazon Athena to analyze sales data that is in Amazon S3. The data engineer writes a query to retrieve sales amounts for 2023 for several products from a table named sales_data. However, the query does not return results for all of the products that are in the sales_data table. The data engineer needs to troubleshoot the query to resolve the issue.
The data engineer's original query is as follows:
SELECT product_name, sum(sales_amount)

FROM sales_data -

WHERE year = 2023 -

GROUP BY product_name -
How should the data engineer modify the Athena query to meet these requirements?

A. Replace sum(sales_amount) with count(*) for the aggregation.

B. Change WHERE year = 2023 to WHERE extract(year FROM sales_data) = 2023.

C. Add HAVING sum(sales_amount) > 0 after the GROUP BY clause.

D. Remove the GROUP BY clause.

Suggested Answer: *B*

Community vote distribution

B (64%) C (36%)

GiorgioGss Highly Voted 1 year, 3 months ago

Selected Answer: B

"SELECT product_name, sum(sales_amount)
FROM sales_data
WHERE extract(year FROM sales_date) = 2023
GROUP BY product_name;"
A. This would change the query to count the number of rows instead of summing sales.
C. This would filter out products with zero sales amounts.
D. Removing the GROUP BY clause would result in a single sum of all sales amounts without grouping by product_name.
upvoted 12 times

pikuantne Highly Voted 8 months ago

None of these options make sense. I think the question is worded incorrectly. I understand that the problem is supposed to be: the products that did not have any sales in 2023 should also be visible in the report with sum of sales_amount = 0. So, the WHERE condition should be deleted and replaced with a CASE WHEN. That way all of the products in the table will be visible, but only sales for 2023 will be summed. Which is what I think this question is asking. None of the provided options do that.
upvoted 7 times

YUICH Most Recent 5 months ago

Selected Answer: B

hy Option (B) Works
If the underlying table field is a date or timestamp (rather than a numeric year column), using WHERE year = 2023 filters out all rows that do not literally match year = 2023.
By using extract(year FROM sales_data) = 2023, you are correctly filtering rows whose date (or timestamp) in the sales_data column corresponds to the year 2023.
Hence, (B) resolves the problem by filtering on the correct year value from the actual date/timestamp column, ensuring all qualifying products are included in the results.
upvoted 2 times

Udyan 5 months, 2 weeks ago

Selected Answer: C

The issue might be that some products have sales amounts of 0 or NULL, and those records are being excluded from the results because Athena may not include them in the final output when performing aggregation. By using the HAVING clause, you can filter the groups based on the aggregated sales amount (sum). This ensures that only products with a non-zero sum of sales are returned in the results. The HAVING clause is used to filter results after the aggregation.

upvoted 1 times

MLOPS_eng 6 months ago

Selected Answer: C

The HAVING clause filters the results to include only products with an aggregated sales amount greater than zero.

upvoted 1 times

Assassin27 6 months ago

Selected Answer: C

```
SELECT product_name, sum(sales_amount)
FROM sales_data
WHERE year = 2023
GROUP BY product_name
HAVING sum(sales_amount) > 0
```

Explanation:

The HAVING clause ensures that only products with a non-zero aggregated sales amount are included in the results. This will address cases where products exist in the table but have no sales data for 2023.

upvoted 1 times

kailu 6 months, 1 week ago

Selected Answer: C

There is no issue with the WHERE clause from the original query, so B is not the right option IMO.

upvoted 1 times

Shatheesh 9 months ago

C, query in the question is correct you just need to get amounts grater than Zero

upvoted 2 times

valuedate 1 year, 1 month ago

Selected Answer: B

year should be the partition in s3 so its necessary to extract. its not a column

upvoted 5 times

VerRi 1 year, 1 month ago

Selected Answer: C

No need to extract the year again

upvoted 2 times

Just_Ninja 1 year, 1 month ago

Selected Answer: C

https://docs.aws.amazon.com/kinesisanalytics/latest/sqlref/sql-reference-having-clause.html

upvoted 1 times

Snape 1 year, 2 months ago

Selected Answer: C

Wrong answers

A. Replace sum(sales_amount) with count(*) for the aggregation. This option will return the count of records for each product, not the sum of sales amounts, which is the desired result.

B. Change WHERE year = 2023 to WHERE extract(year FROM sales_data) = 2023. The year column likely stores the year value directly, so there's no need to extract it from a date or timestamp column.

D. Remove the GROUP BY clause. Removing the GROUP BY clause will cause an error because the sum(sales_amount) aggregation function requires a GROUP BY clause to specify the grouping column (product_name in this case).

upvoted 1 times

khchan123 1 year, 2 months ago

B

B. Change `WHERE year = 2023` to `WHERE extract(year FROM sales_data) = 2023`.

The issue with the original query is that it assumes there is a column named `year` in the `sales_data` table. However, it's more likely that the date or timestamp information is stored in a single column, for example, a column named `sales_date`.

To extract the year from a date or timestamp column, you need to use the `extract()` function in Athena SQL.

upvoted 3 times

⊟ **chris_spencer** 1 year, 2 months ago

None of the answer makes senses. Option C will exclude any amount that is 0. This option would be correct if it is: Add HAVING sum(sales_amount) >= 0 after the GROUP BY clause.

upvoted 2 times

⊟ **Christina666** 1 year, 2 months ago

Selected Answer: C

Gemini: C. Add HAVING sum(sales_amount) > 0 after the GROUP BY clause.

Zero Sales Products: The original query is likely missing products that had zero sales amount in 2023. This modification filters the grouped results, ensuring only products with positive sales are displayed.

Why Other Options Don't Address the Core Issue:

A. Replace sum(sales_amount) with count(*) for the aggregation. This would show how many sales transactions a product had, but not if it generated any revenue. It wouldn't solve the issue of missing products.

B. Change WHERE year = 2023 to WHERE extract(year FROM sales_data) = 2023. This is functionally equivalent to the original WHERE clause if the year column is already an integer type. It wouldn't fix missing products.

D. Remove the GROUP BY clause. This would aggregate all sales for 2023 with no product breakdown, losing the granularity needed.

upvoted 2 times

⊟ **altonh** 6 months, 3 weeks ago

Why would the query miss out on products with zero sales when the condition is based on year?

upvoted 1 times

⊟ **kj07** 1 year, 3 months ago

Not A because the engineer wants a sum not the total count.

Not C because it will filter out the data with sales_amount zero.

Not D because it will return just one result and the engineer wants the sales for multiple products.

B should be the right answer if the sales_data is a date field.

upvoted 4 times

⊟ **DevoteamAnalytix** 1 year, 1 month ago

But this is the table name...

upvoted 1 times

⊟ **Felix_G** 1 year, 3 months ago

Add HAVING sum(sales_amount) > 0 this does NOT filter out the data with sales_mount zero. it can not be any negative value.

The original query's WHERE year = 2023 condition is already appropriate for filtering data by the year 2023, so that B is unnecessary.

upvoted 1 times

⊟ **FuriouZ** 1 year, 3 months ago

>0 filters out every product which is not sold. The question was about "some products are not displayed" so using the having argument can not be the right choice

upvoted 3 times

⊟ **rralucard_** 1 year, 4 months ago

Selected Answer: C

https://docs.aws.amazon.com/athena/latest/ug/select.html

upvoted 2 times

⊟ **nyaopoko** 1 year, 2 months ago

answer is C!

upvoted 1 times

Question #49 *Topic 1*

A data engineer has a one-time task to read data from objects that are in Apache Parquet format in an Amazon S3 bucket. The data engineer needs to query only one column of the data.
Which solution will meet these requirements with the LEAST operational overhead?

A. Configure an AWS Lambda function to load data from the S3 bucket into a pandas dataframe. Write a SQL SELECT statement on the dataframe to query the required column.

B. Use S3 Select to write a SQL SELECT statement to retrieve the required column from the S3 objects.

C. Prepare an AWS Glue DataBrew project to consume the S3 objects and to query the required column.

D. Run an AWS Glue crawler on the S3 objects. Use a SQL SELECT statement in Amazon Athena to query the required column.

Suggested Answer: *B*

Community vote distribution

B (90%) 10%

XP_2600 2 weeks, 6 days ago

Selected Answer: D

B is no longer valid:
https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-select.html

Important

Amazon S3 Select is no longer available to new customers. Existing customers of Amazon S3 Select can continue to use the feature as usual.

upvoted 1 times

imymoco 6 months, 1 week ago

Selected Answer: B

only one column -> S3 select

upvoted 1 times

JoeAWSOCM 6 months, 3 weeks ago

Selected Answer: D

S3 select is for querying one object. Here the requirement is to query one column from multiple objects. Also S3 select is discontinued for new users. So answer could be D

upvoted 4 times

catoteja 10 months, 2 weeks ago

Amazon S3 Select is no longer available to new customers. Existing customers of Amazon S3 Select can continue to use the feature as usual

But with it you can only query one object xD. Glue + athena

upvoted 1 times

dungct 1 year ago

but s3 select can only select one object

upvoted 3 times

hogs 1 year ago

Selected Answer: B

omly once

upvoted 2 times

FunkyFresco 1 year, 1 month ago

Selected Answer: B

if is one-time task

upvoted 2 times

GiorgioGss 1 year, 3 months ago

Selected Answer: B

https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-select.html

upvoted 2 times

⊟ 👤 **rralucard_** 1 year, 4 months ago

Selected Answer: B

https://docs.aws.amazon.com/AmazonS3/latest/userguide/storage-inventory-athena-query.html

S3 Select allows you to retrieve a subset of data from an object stored in S3 using simple SQL expressions. It is capable of working directly with objects in Parquet format.

upvoted 3 times

Question #50 *Topic 1*

A company uses Amazon Redshift for its data warehouse. The company must automate refresh schedules for Amazon Redshift materialized views.

Which solution will meet this requirement with the LEAST effort?

A. Use Apache Airflow to refresh the materialized views.

B. Use an AWS Lambda user-defined function (UDF) within Amazon Redshift to refresh the materialized views.

C. Use the query editor v2 in Amazon Redshift to refresh the materialized views.

D. Use an AWS Glue workflow to refresh the materialized views.

Suggested Answer: *C*

Community vote distribution

C (86%) 14%

magnorm 6 months ago

Selected Answer: C

https://docs.aws.amazon.com/redshift/latest/mgmt/query-editor-v2-schedule-query.html

upvoted 2 times

pypelyncar 6 months, 3 weeks ago

Selected Answer: C

the company can automate the refresh schedules for materialized views with minimal effort. This approach leverages the built-in capabilities of Amazon Redshift, reducing the need for additional services, configurations, or custom code. It aligns with the principle of using the simplest and most straightforward solution that meets the requirements, minimizing operational overhead and complexity

upvoted 2 times

d8945a1 7 months, 3 weeks ago

Selected Answer: C

We can schedule the refresh using query scheduler from Query Editor V2.

upvoted 3 times

Christina666 8 months, 2 weeks ago

Selected Answer: C

Amazon Redshift can automatically refresh materialized views with up-to-date data from its base tables when materialized views are created with or altered to have the autorefresh option. For more details, refer to the documentation here, https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-refresh.html.

upvoted 2 times

[Removed] 9 months ago

Selected Answer: C

https://docs.aws.amazon.com/redshift/latest/mgmt/query-editor-v2-schedule-query.html

upvoted 2 times

FuriouZ 9 months, 1 week ago

Selected Answer: C

You can set autorefresh for materialized views using CREATE MATERIALIZED VIEW. You can also use the AUTO REFRESH clause to refresh materialized views automatically.

upvoted 2 times

GiorgioGss 9 months, 2 weeks ago

Selected Answer: C

https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-refresh.html

upvoted 1 times

kj07 9 months, 2 weeks ago

You can set AUTO REFRESH option on creation. So I will vote with C.

upvoted 1 times

confusedyeti69 9 months, 2 weeks ago

Selected Answer: C

Lambda requires code and configuring permissions. A and D are additional overheads as well. Vote C

upvoted 1 times

damaldon 9 months, 3 weeks ago

B.

https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-UDFs.html

upvoted 1 times

rralucard_ 11 months ago

Selected Answer: B

AWS Lambda allows running code in response to triggers without needing to provision or manage servers. However, creating a UDF within Amazon Redshift to call a Lambda function for this purpose involves writing custom code and managing permissions between Lambda and Redshift.

upvoted 2 times

Question #51 *Topic 1*

A data engineer must orchestrate a data pipeline that consists of one AWS Lambda function and one AWS Glue job. The solution must integrate with AWS services.
Which solution will meet these requirements with the LEAST management overhead?

A. Use an AWS Step Functions workflow that includes a state machine. Configure the state machine to run the Lambda function and then the AWS Glue job.

B. Use an Apache Airflow workflow that is deployed on an Amazon EC2 instance. Define a directed acyclic graph (DAG) in which the first task is to call the Lambda function and the second task is to call the AWS Glue job.

C. Use an AWS Glue workflow to run the Lambda function and then the AWS Glue job.

D. Use an Apache Airflow workflow that is deployed on Amazon Elastic Kubernetes Service (Amazon EKS). Define a directed acyclic graph (DAG) in which the first task is to call the Lambda function and the second task is to call the AWS Glue job.

Suggested Answer: *A*

Community vote distribution

A (89%) 11%

pypelyncar Highly Voted 1 year ago

Selected Answer: A

Step Functions is a managed service for building serverless workflows. You define a state machine that orchestrates the execution sequence.
This eliminates the need to manage and maintain your own workflow orchestration server like Airflow.

upvoted 6 times

hcong Most Recent 10 months, 2 weeks ago

Selected Answer: C

AWS Glue is a fully managed ETL (extract, transform, load) service that makes it easy to orchestrate data pipelines. Using the AWS Glue workflow to run Lambda functions and glue jobs is the easiest and least expensive option because it's a fully managed service that requires no additional workflow tools or infrastructure to configure and manage. Other options require additional tools or resources to configure and manage, and are therefore more expensive to manage.

upvoted 4 times

tgv 1 year ago

Selected Answer: A

Step Functions can handle both Lambda and Glue in this scenario, making it the best choice.

upvoted 2 times

hnk 1 year, 1 month ago

Selected Answer: A

B and D require additional effort
C Glue workflows do not have a direct integration with lambda
hence the best choice is A

upvoted 4 times

FuriouZ 1 year, 3 months ago

Selected Answer: A

Key word orchestrating is most likely step functions

upvoted 3 times

rralucard_ 1 year, 4 months ago

Selected Answer: A

Option A, using AWS Step Functions, is the best solution to meet the requirement with the least management overhead. Step Functions is designed for easy integration with AWS services like Lambda and Glue, providing a managed, low-code approach to orchestrate workflows. This allows for a more straightforward setup and less ongoing management compared to the other options.

upvoted 4 times

Question #52 *Topic 1*

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 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.

B. 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.

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. 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.

Suggested Answer: *B*

Community vote distribution

B (94%) 6%

pypelyncar Highly Voted 6 months, 3 weeks ago

Selected Answer: B

The AWS Glue Data Catalog is a purpose-built, fully managed service designed to serve as a central metadata repository for your data sources. It provides a unified view of your data across various sources, including structured databases (like Amazon RDS and Amazon Redshift) and semi-structured data formats (like JSON and XML files in Amazon S3).

upvoted 7 times

valuedate Most Recent 7 months, 1 week ago

Selected Answer: B

glue data catalog with crawlers

upvoted 3 times

hnk 7 months, 2 weeks ago

Selected Answer: A

B is the obvious answer

upvoted 1 times

Just_Ninja 7 months, 2 weeks ago

Sorry there is no Aurora Data Catalog :)

upvoted 1 times

tgv 7 months ago

Even though you picked A.

upvoted 3 times

GiorgioGss 9 months, 2 weeks ago

Selected Answer: B

A,C out for obvious reason

D out because it involves manual schema extract

upvoted 4 times

rralucard_ 11 months ago

Selected Answer: B

Option B, using the AWS Glue Data Catalog with AWS Glue Crawlers, is the best solution to meet the requirements with the least operational overhead. It provides a fully managed, integrated solution for cataloging both structured and semistructured data across various AWS data stores without the need for extensive manual configuration or custom coding.

upvoted 3 times

Question #53 *Topic 1*

A company stores data from an application in an Amazon DynamoDB table that operates in provisioned capacity mode. The workloads of the application have predictable throughput load on a regular schedule. Every Monday, there is an immediate increase in activity early in the morning. The application has very low usage during weekends.
The company must ensure that the application performs consistently during peak usage times.
Which solution will meet these requirements in the MOST cost-effective way?

A. Increase the provisioned capacity to the maximum capacity that is currently present during peak load times.

B. Divide the table into two tables. Provision each table with half of the provisioned capacity of the original table. Spread queries evenly across both tables.

C. Use AWS Application Auto Scaling to schedule higher provisioned capacity for peak usage times. Schedule lower capacity during off-peak times.

D. Change the capacity mode from provisioned to on-demand. Configure the table to scale up and scale down based on the load on the table.

Suggested Answer: *C*

Community vote distribution

C (88%) 6%

⊟ **rralucard_** Highly Voted 1 year, 4 months ago

Selected Answer: C

Option C, using AWS Application Auto Scaling to schedule higher provisioned capacity for peak usage times and lower capacity during off-peak times, is the most cost-effective solution for the described scenario. It allows the company to align their DynamoDB capacity costs with actual usage patterns, scaling up only when needed and scaling down during low-usage periods.

upvoted 5 times

⊟ **Rakiko** Most Recent 3 months, 4 weeks ago

Selected Answer: C

My guess is C as it stands for Cat

upvoted 1 times

⊟ **sdas1** 11 months, 3 weeks ago

C

https://docs.aws.amazon.com/wellarchitected/latest/serverless-applications-lens/capacity.html

DynamoDB auto scaling modifies provisioned throughput settings only when the actual workload stays elevated or depressed for a sustained period of several minutes. This means that provisioned capacity is probably best for you if you have relatively predictable application traffic, run applications whose traffic is consistent, and ramps up or down gradually.

upvoted 1 times

⊟ **pypelyncar** 1 year ago

Selected Answer: C

app autoscalling allows you to dynamically adjust provisioned capacity based on usage patterns. You only pay for the capacity you utilize, reducing costs compared to keeping a high, fixed capacity throughout the week

upvoted 4 times

⊟ **Christina666** 1 year, 2 months ago

Selected Answer: C

D. Change the capacity mode from provisioned to on-demand... On-demand mode is great for unpredictable workloads. In your case, with predictable patterns, you'd likely pay more with on-demand than with a well-managed, scheduled, provisioned mode.

upvoted 3 times

⊟ **tgv** 1 year, 1 month ago

I agree with you, on-demand tends to be picked when you don't know the workload. While in this scenario they know, so technically the Auto Scaling solution would be much cheaper here.

upvoted 1 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: D

As I understand, should be D

upvoted 1 times

lucas_rfsb 1 year, 2 months ago

But C is also a good choice. Maybe because it is predictable, I'm now intending to choose C

upvoted 4 times

FuriouZ 1 year, 3 months ago

Selected Answer: C

Obviously better than B because of peak scaling

upvoted 3 times

jpmadan 1 year, 3 months ago

Selected Answer: B

D

Excerpts from documentation:

This means that provisioned capacity is probably best for you if you have relatively predictable application traffic, run applications whose traffic is consistent, and ramps up or down gradually.

Whereas on-demand capacity mode is probably best when you have new tables with unknown workloads, unpredictable application traffic and also if you only want to pay exactly for what you use. The on-demand pricing model is ideal for bursty, new, or unpredictable workloads whose traffic can spike in seconds or minutes, and when under-provisioned capacity would impact the user experience.

https://docs.aws.amazon.com/wellarchitected/latest/serverless-applications-lens/capacity.html

upvoted 1 times

jpmadan 1 year, 3 months ago

selected answer should be D

upvoted 1 times

tgv 1 year, 1 month ago

Well, as your comment says:

D - on-demand capacity mode is probably best when you have new tables with unknown workloads, unpredictable application traffic and also if you only want to pay exactly for what you use.

That's not the case, they know exactly when they are expecting an increasing. So the most cost-effective solution is C - Auto Scaling.

upvoted 1 times

damaldon 1 year, 3 months ago

C.

https://docs.aws.amazon.com/autoscaling/application/userguide/services-that-can-integrate-dynamodb.html

upvoted 2 times

Question #54 *Topic 1*

A company is planning to migrate on-premises Apache Hadoop clusters to Amazon EMR. The company also needs to migrate a data catalog into a persistent storage solution.

The company currently stores the data catalog in an on-premises Apache Hive metastore on the Hadoop clusters. The company requires a serverless solution to migrate the data catalog.

Which solution will meet these requirements MOST cost-effectively?

A. Use AWS Database Migration Service (AWS DMS) to migrate the Hive metastore into Amazon S3. Configure AWS Glue Data Catalog to scan Amazon S3 to produce the data catalog.

B. Configure a Hive metastore in Amazon EMR. Migrate the existing on-premises Hive metastore into Amazon EMR. Use AWS Glue Data Catalog to store the company's data catalog as an external data catalog.

C. Configure an external Hive metastore in Amazon EMR. Migrate the existing on-premises Hive metastore into Amazon EMR. Use Amazon Aurora MySQL to store the company's data catalog.

D. Configure a new Hive metastore in Amazon EMR. Migrate the existing on-premises Hive metastore into Amazon EMR. Use the new metastore as the company's data catalog.

Suggested Answer: *B*

Community vote distribution

B (90%) 10%

Asmunk 7 months, 3 weeks ago

Selected Answer: B

A and D can be discarded because of added steps. This link provides documentation for this exact use case : https://aws.amazon.com/blogs/big-data/migrate-and-deploy-your-apache-hive-metastore-on-amazon-emr/

C is also discarded because of the serverless key word, although Aurora can be serverless it is not specified in the choice.

upvoted 2 times

Christina666 1 year, 2 months ago

Selected Answer: B

Serverless and Cost-Efficient: AWS Glue Data Catalog offers a serverless metadata repository, reducing operational overhead and making it cost-effective. Using it as an external data catalog means you don't have to manage additional database infrastructure.

Seamless Migration: Migrating your existing Hive metastore to Amazon EMR ensures compatibility with your current Hadoop setup. EMR is designed to run Hadoop workloads, facilitating this process.

Flexibility: An external data catalog in AWS Glue offers flexibility and separation of concerns. Your metastore remains managed by EMR for your Hadoop workloads, while Glue provides a centralized catalog for broader AWS data sources.

upvoted 2 times

nyaopoko 1 year, 2 months ago

B is answer!

By leveraging AWS Glue Data Catalog as an external data catalog and migrating the existing Hive metastore into Amazon EMR, the company can achieve a serverless, persistent, and cost-effective solution for storing and managing their data catalog.

upvoted 1 times

arvehisa 1 year, 2 months ago

Selected Answer: B

B. https://aws.amazon.com/jp/blogs/big-data/migrate-and-deploy-your-apache-hive-metastore-on-amazon-emr/

upvoted 2 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: A

I will go with A. Besides DMS is typical for migration, it's the only choice which explicitly concerns about how the migration itself will be made. Other choices would demand a script or GLUE ETL job if you will. But this logic of migration was never put

upvoted 2 times

LeoSantos121212121212121 1 year, 3 months ago

I will go with A

upvoted 2 times

jpmadan 1 year, 3 months ago

Selected Answer: B

serverless catalog in AWS == glue

upvoted 1 times

damaldon 1 year, 3 months ago

B.

Set up an AWS Glue ETL job which extracts metadata from your Hive metastore (MySQL) and loads it into your AWS Glue Data Catalog. This method requires an AWS Glue connection to the Hive metastore as a JDBC source. An ETL script is provided to extract metadata from the Hive metastore and write it to AWS Glue Data Catalog.

https://github.com/aws-samples/aws-glue-samples/blob/master/utilities/Hive_metastore_migration/README.md

upvoted 1 times

rralucard_ 1 year, 4 months ago

Selected Answer: B

https://aws.amazon.com/blogs/big-data/migrate-and-deploy-your-apache-hive-metastore-on-amazon-emr/ Option B is likely the most suitable. Migrating the Hive metastore into Amazon EMR and using AWS Glue Data Catalog as an external catalog provides a balance between leveraging the scalable and managed services of AWS (like EMR and Glue Data Catalog) and ensuring a smooth transition from the on-premises setup. This approach leverages the serverless nature of AWS Glue Data Catalog, minimizing operational overhead and potentially reducing costs compared to managing database servers.

upvoted 3 times

Question #55 *Topic 1*

A company uses an Amazon Redshift provisioned cluster as its database. The Redshift cluster has five reserved ra3.4xlarge nodes and uses key distribution.
A data engineer notices that one of the nodes frequently has a CPU load over 90%. SQL Queries that run on the node are queued. The other four nodes usually have a CPU load under 15% during daily operations.
The data engineer wants to maintain the current number of compute nodes. The data engineer also wants to balance the load more evenly across all five compute nodes.
Which solution will meet these requirements?

A. Change the sort key to be the data column that is most often used in a WHERE clause of the SQL SELECT statement.

B. Change the distribution key to the table column that has the largest dimension.

C. Upgrade the reserved node from ra3.4xlarge to ra3.16xlarge.

D. Change the primary key to be the data column that is most often used in a WHERE clause of the SQL SELECT statement.

Suggested Answer: *B*

Community vote distribution

B (92%) 8%

⊟ **rralucard_** Highly Voted 11 months ago

Selected Answer: B

https://docs.aws.amazon.com/redshift/latest/dg/t_Distributing_data.html
Option B, changing the distribution key, is the most effective solution to balance the load more evenly across all five compute nodes. Selecting an appropriate distribution key that aligns with the query patterns and data characteristics can result in a more uniform distribution of data and workloads, thus reducing the likelihood of one node being overutilized while others are underutilized.

upvoted 7 times

⊟ **pypelyncar** Most Recent 6 months, 3 weeks ago

Selected Answer: B

In a Redshift cluster with key distribution, data is distributed across compute nodes based on the values of the distribution key. An uneven distribution can lead to skewed workloads on specific nodes.
By choosing the table column with the largest dimension (most distinct values) as the distribution key, you ensure a more even spread of data across all nodes. This balances the processing load on each node when queries access that column.

upvoted 2 times

⊟ **khchan123** 8 months ago

Selected Answer: B

The correct solution is B. Change the distribution key to the table column that has the largest dimension. This will help to distribute the data more evenly across the nodes, reducing the load on the heavily utilized node.

upvoted 2 times

⊟ **Christina666** 8 months, 2 weeks ago

Selected Answer: A

Gemini result:
Understanding the Problem:

The scenario describes a Redshift cluster with uneven load distribution. This indicates potential issues with either the distribution style or the sort key.

Key Distribution:

The problem states that the cluster uses key distribution, meaning a specific column is designated as the distribution key. Data rows with matching distribution key values are placed on the same node.

Sort Key:

A sort key determines the order in which data is physically stored within a table's blocks on a node. A well-chosen sort key can significantly optimize query performance, especially when queries often filter by that column.

upvoted 1 times

⊟ **tgv** 7 months ago

The sort key determines the order of data storage and can improve query performance for specific queries, but it does not directly affect the distribution of data across nodes. Therefore, this will not address the uneven CPU load issue.

upvoted 1 times

⊟ **damaldon** 9 months, 3 weeks ago

B.

With "Key distribution". The rows are distributed according to the values in one column. The leader node places matching values on the same node slice. If you distribute a pair of tables on the joining keys, the leader node collocates the rows on the slices according to the values in the joining columns. This way, matching values from the common columns are physically stored together.

https://docs.aws.amazon.com/redshift/latest/dg/c_choosing_dist_sort.html

upvoted 2 times

Question #56 *Topic 1*

A security company stores IoT data that is in JSON format in an Amazon S3 bucket. The data structure can change when the company upgrades the IoT devices. The company wants to create a data catalog that includes the IoT data. The company's analytics department will use the data catalog to index the data.
Which solution will meet these requirements MOST cost-effectively?

A. Create an AWS Glue Data Catalog. Configure an AWS Glue Schema Registry. Create a new AWS Glue workload to orchestrate the ingestion of the data that the analytics department will use into Amazon Redshift Serverless.

B. Create an Amazon Redshift provisioned cluster. Create an Amazon Redshift Spectrum database for the analytics department to explore the data that is in Amazon S3. Create Redshift stored procedures to load the data into Amazon Redshift.

C. Create an Amazon Athena workgroup. Explore the data that is in Amazon S3 by using Apache Spark through Athena. Provide the Athena workgroup schema and tables to the analytics department.

D. Create an AWS Glue Data Catalog. Configure an AWS Glue Schema Registry. Create AWS Lambda user defined functions (UDFs) by using the Amazon Redshift Data API. Create an AWS Step Functions job to orchestrate the ingestion of the data that the analytics department will use into Amazon Redshift Serverless.

Suggested Answer: *A*

Community vote distribution

A (80%) C (20%)

rralucard_ Highly Voted 1 year, 4 months ago

Selected Answer: A

Option A, creating an AWS Glue Data Catalog with Glue Schema Registry and orchestrating data ingestion into Amazon Redshift Serverless using AWS Glue, appears to be the most cost-effective and suitable solution. It offers a serverless approach to manage the evolving data schema of the IoT data and efficiently supports data analytics needs without the overhead of managing a provisioned database cluster or complex orchestration setups.

upvoted 9 times

nyaopoko 1 year, 2 months ago

Selected Answer: A

Amazon Redshift Serverless is a serverless option for Amazon Redshift, which means you don't have to provision and manage clusters. This makes it a cost-effective choice for the analytics department's use case.

upvoted 2 times

VerRi Most Recent 1 year, 1 month ago

Selected Answer: A

Athena is not able to create new data catalog

upvoted 1 times

sdas1 1 year, 1 month ago

Option C

Cost-effectiveness: Amazon Athena allows you to query data directly from Amazon S3 without the need for any infrastructure setup or management. You pay only for the queries you run, making it cost-effective, especially for sporadic or exploratory analysis.
Flexibility: Since the data structure can change with IoT device upgrades, using Athena allows for flexibility in querying and analyzing the data regardless of its structure. You don't need to define a fixed schema upfront, enabling you to adapt to changes seamlessly.
Apache Spark Support: Athena supports querying data using Apache Spark, which is powerful for processing and analyzing large datasets. This capability ensures that the analytics department can leverage Spark for more advanced analytics if needed.
https://www.youtube.com/watch?v=Q93NZJBFSWw

upvoted 1 times

khchan123 1 year, 2 months ago

Selected Answer: A

The correct solution is A. Create an AWS Glue Data Catalog. Configure an AWS Glue Schema Registry. Create a new AWS Glue workload to orchestrate the ingestion of the data that the analytics department will use into Amazon Redshift Serverless.

Option C (Amazon Athena and Apache Spark) is suitable for ad-hoc querying and exploration but may not be the best choice for the analytics department's ongoing data analysis needs, as Athena is designed for interactive querying rather than complex data transformations.

upvoted 2 times

altonh 6 months, 3 weeks ago

However, combined with Notebook, Athena+Spark can be a powerful tool for analytics.

upvoted 1 times

chris_spencer 1 year, 2 months ago

Selected Answer: A

The objective is to create a data catalog that includes the IoT data and AWS Glue Data Catalog is the best option for this requirement.
https://docs.aws.amazon.com/glue/latest/dg/catalog-and-crawler.html

C is incorrect. While Athena makes it easy to read from S3 using SQL, it does not crawl the data source and create a data catalog.

upvoted 4 times

Christina666 1 year, 2 months ago

Selected Answer: C

Why Option C is the Most Cost-Effective

Serverless and Pay-as-you-go: Athena is a serverless query service, meaning you only pay for the queries the analytics department runs. No need to provision and manage always-running clusters.
Flexible Schema Handling: Athena works well with semi-structured data like JSON and can handle schema evolution on the fly. This is perfect for the scenario where IoT data structures might change.
Spark Integration: Integrating Apache Spark with Athena provides rich capabilities for data processing and transformation.
Ease of Use for Analytics: Athena's familiar SQL-like interface and ability to directly query S3 data make it convenient for the analytics department.

upvoted 2 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: C

Options A, B, and D involve setting up additional infrastructure (e.g., AWS Glue, Redshift clusters, Lambda functions) which may incur unnecessary costs and complexity for the given requirements. Option C, on the other hand, utilizes a serverless and scalable solution directly querying data in S3, making it the most cost-effective choice.

upvoted 2 times

Question #57 *Topic 1*

A company stores details about transactions in an Amazon S3 bucket. The company wants to log all writes to the S3 bucket into another S3 bucket that is in the same AWS Region.
Which solution will meet this requirement with the LEAST operational effort?

A. Configure an S3 Event Notifications rule for all activities on the transactions S3 bucket to invoke an AWS Lambda function. Program the Lambda function to write the event to Amazon Kinesis Data Firehose. Configure Kinesis Data Firehose to write the event to the logs S3 bucket.

B. Create a trail of management events in AWS CloudTraiL. Configure the trail to receive data from the transactions S3 bucket. Specify an empty prefix and write-only events. Specify the logs S3 bucket as the destination bucket.

C. Configure an S3 Event Notifications rule for all activities on the transactions S3 bucket to invoke an AWS Lambda function. Program the Lambda function to write the events to the logs S3 bucket.

D. Create a trail of data events in AWS CloudTraiL. Configure the trail to receive data from the transactions S3 bucket. Specify an empty prefix and write-only events. Specify the logs S3 bucket as the destination bucket.

Suggested Answer: *D*

Community vote distribution

D (100%)

rralucard_ Highly Voted 11 months ago

Selected Answer: D

https://docs.aws.amazon.com/AmazonS3/latest/userguide/logging-with-S3.html
Option D, creating a trail of data events in AWS CloudTrail, is the best solution to meet the requirement with the least operational effort. It directly logs the desired activities to another S3 bucket and does not involve the development and maintenance of additional resources like Lambda functions or Kinesis Data Firehose streams.

upvoted 5 times

VerRi Highly Voted 7 months, 1 week ago

Selected Answer: D

A: Don't need all activities on the S3 bucket
B: Management events include not only the data log but also the admin log
C: Don't need all activities on the S3 bucket
Option D with the LEAST operational effort

upvoted 5 times

khchan123 Most Recent 8 months ago

Selected Answer: D

Correct answer is D.

Option A or C require writing custom Lambda code to handle the events and write them to the Kinesis or S3 bucket so they are not the LEAST operational effort.

upvoted 3 times

LanoraMoe 8 months, 1 week ago

S3 object level activities such as GetObject, DeleteObject, PutObject etc are considered as Data event in cloud trail. Read and Write event be monitored separately.

upvoted 1 times

okechi 8 months, 1 week ago

The correct answer is B - CloudTrail of management events includes logging set ups like this

upvoted 1 times

GiorgioGss 9 months, 2 weeks ago

Although it might be tempting going with C, please keep in mind that if we go with C we must define lambda code, lambda permission, triggers, etc. If we go with D we just enable a trail data events and that's pretty much it.

upvoted 3 times

Felix_G 10 months ago

Other Options were Less Efficient:

A. Leverage S3 Event Notifications, Lambda function, and Kinesis Data Firehose: While this option works, it involves setting up and managing three services, increasing complexity and operational overhead. Kinesis Data Firehose introduces an unnecessary intermediary step, adding complexity for a straightforward logging task.

B. Utilize CloudTrail with Management Events: CloudTrail primarily tracks API calls and management activities related to S3 buckets, not data events like writes to objects. Consequently, it wouldn't capture the desired S3 bucket writes.

D. Employ CloudTrail with Data Events: Similar to option B, CloudTrail with data events doesn't track individual object writes within a bucket. It focuses on object-level changes like creation, deletion, or metadata modification.

upvoted 2 times

⊟ **Felix_G** 10 months ago

Option C is right , by employing S3 Event Notifications with a Lambda function directly writing to the logs S3 bucket, you achieve the desired logging functionality with minimal setup, management, and cost compared to the other options. This approach leverages the built-in integration between these services and avoids unnecessary service dependencies.

upvoted 1 times

⊟ **Luke97** 9 months ago

Check Amazon S3 object-level actions that are tracked by AWS CloudTrail logging
https://docs.aws.amazon.com/AmazonS3/latest/userguide/cloudtrail-logging-s3-info.html

You can get CloudTrail logs for object-level Amazon S3 actions. To do this, enable data events for your S3 bucket or all buckets in your account.

upvoted 1 times

⊟ **Luke97** 9 months ago

Write to S3 means PutObject, CopyObject API

upvoted 1 times

Question #58 *Topic 1*

A data engineer needs to maintain a central metadata repository that users access through Amazon EMR and Amazon Athena queries. The repository needs to provide the schema and properties of many tables. Some of the metadata is stored in Apache Hive. The data engineer needs to import the metadata from Hive into the central metadata repository.
Which solution will meet these requirements with the LEAST development effort?

A. Use Amazon EMR and Apache Ranger.

B. Use a Hive metastore on an EMR cluster.

C. Use the AWS Glue Data Catalog.

D. Use a metastore on an Amazon RDS for MySQL DB instance.

Suggested Answer: *C*

Community vote distribution

C (100%)

⊟ **rralucard_** Highly Voted 1 year, 4 months ago

Selected Answer: C

https://aws.amazon.com/blogs/big-data/metadata-classification-lineage-and-discovery-using-apache-atlas-on-amazon-emr/
Option C, using the AWS Glue Data Catalog, is the best solution to meet the requirements with the least development effort. The AWS Glue Data Catalog is designed to be a central metadata repository that can integrate with various AWS services including EMR and Athena, providing a managed and scalable solution for metadata management with built-in Hive compatibility.

upvoted 6 times

⊟ **vic614** Most Recent 1 year ago

Selected Answer: C

Data Catalog.

upvoted 1 times

⊟ **Felix_G** 1 year, 4 months ago

Option C, using the AWS Glue Data Catalog, requires the least development effort to meet the requirements for a central metadata repository accessed from EMR and Athena.

upvoted 2 times

⊟ **Felix_G** 1 year, 4 months ago

Here's an analysis of each option:

A) Amazon EMR and Apache Ranger would require significant coding to build a custom metadata repository solution

B) A Hive metastore provides metadata to EMR, but would require substantial development work to share that metadata with Athena

C) The AWS Glue Data Catalog integrates natively with EMR and Athena, providing a shared schema registry, making it the easiest solution

D) An RDS database metastore would also require building custom integration points with Athena, EMR, and other services to enable metadata sharing

Since AWS Glue provides a fully managed data catalog service purpose built for this metadata management use case across different analytics engines, Option C clearly stands out as the solution requiring the least development effort.

upvoted 2 times

Question #59 *Topic 1*

A company needs to build a data lake in AWS. The company must provide row-level data access and column-level data access to specific teams. The teams will access the data by using Amazon Athena, Amazon Redshift Spectrum, and Apache Hive from Amazon EMR.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon S3 for data lake storage. Use S3 access policies to restrict data access by rows and columns. Provide data access through Amazon S3.

B. Use Amazon S3 for data lake storage. Use Apache Ranger through Amazon EMR to restrict data access by rows and columns. Provide data access by using Apache Pig.

C. Use Amazon Redshift for data lake storage. Use Redshift security policies to restrict data access by rows and columns. Provide data access by using Apache Spark and Amazon Athena federated queries.

D. Use Amazon S3 for data lake storage. Use AWS Lake Formation to restrict data access by rows and columns. Provide data access through AWS Lake Formation.

Suggested Answer: *D*

Community vote distribution

D (100%)

Shanmahi 10 months ago

Selected Answer: D

Using Amazon S3 for storage and AWS Lake Formation for fine-grained access control like row-level or column-level access.

upvoted 1 times

cas_tori 10 months, 2 weeks ago

Selected Answer: D

this id D

upvoted 3 times

Felix_G 1 year, 4 months ago

Option D is the best solution to meet the requirements with the least operational overhead.

Using Amazon S3 for storage and AWS Lake Formation for access control and data access delivers the following advantages:

S3 provides a highly durable, available, and scalable data lake storage layer
Lake Formation enables fine-grained access control down to column and row-level
Integrates natively with Athena, Redshift Spectrum, and EMR for simplified data access
Fully managed service minimizes admin overhead vs self-managing Ranger or piecemeal solutions

upvoted 4 times

Felix_G 1 year, 4 months ago

Option A would require custom access control code development and greater ops effort
Option B still requires managing Ranger integrated with EMR
Option C does not natively support column-level security policies

upvoted 1 times

rralucard_ 1 year, 4 months ago

Selected Answer: D

https://docs.aws.amazon.com/lake-formation/latest/dg/cbac-tutorial.html
Option D, using Amazon S3 for data lake storage and AWS Lake Formation for access control, is the most suitable solution. It meets the requirements for row-level and column-level access control and integrates well with Amazon Athena, Amazon Redshift Spectrum, and Apache Hive on EMR, all with lower operational overhead compared to the other options.

upvoted 4 times

Question #60 *Topic 1*

An airline company is collecting metrics about flight activities for analytics. The company is conducting a proof of concept (POC) test to show how analytics can provide insights that the company can use to increase on-time departures.

The POC test uses objects in Amazon S3 that contain the metrics in .csv format. The POC test uses Amazon Athena to query the data. The data is partitioned in the S3 bucket by date.

As the amount of data increases, the company wants to optimize the storage solution to improve query performance.

Which combination of solutions will meet these requirements? (Choose two.)

A. Add a randomized string to the beginning of the keys in Amazon S3 to get more throughput across partitions.

B. Use an S3 bucket that is in the same account that uses Athena to query the data.

C. Use an S3 bucket that is in the same AWS Region where the company runs Athena queries.

D. Preprocess the .csv data to JSON format by fetching only the document keys that the query requires.

E. Preprocess the .csv data to Apache Parquet format by fetching only the data blocks that are needed for predicates.

Suggested Answer: *CE*

Community vote distribution

CE (100%)

rralucard_ Highly Voted 11 months ago

Selected Answer: CE

https://docs.aws.amazon.com/athena/latest/ug/performance-tuning.html

upvoted 6 times

Ramdi1 Most Recent 3 months, 3 weeks ago

Selected Answer: CE

C - Reduces latency and network costs → When Athena queries S3 data in the same AWS Region, data does not cross AWS Regions, improving performance.

Lower query execution time → No inter-region data transfer delays.

Cost-Effective → AWS charges for cross-region data transfers, but querying within the same region avoids these costs.

E - Parquet is a columnar storage format → Queries can fetch only needed columns, reducing scanning costs.

upvoted 1 times

tgv 7 months ago

Selected Answer: CE

I will go with C and E.

upvoted 1 times

matasejem 8 months, 3 weeks ago

C is not mentioned anywhere in the https://docs.aws.amazon.com/athena/latest/ug/performance-tuning.html

upvoted 1 times

damaldon 9 months, 3 weeks ago

Answer C and E

upvoted 1 times

Question #61 *Topic 1*

A company uses Amazon RDS for MySQL as the database for a critical application. The database workload is mostly writes, with a small number of reads.

A data engineer notices that the CPU utilization of the DB instance is very high. The high CPU utilization is slowing down the application. The data engineer must reduce the CPU utilization of the DB Instance.

Which actions should the data engineer take to meet this requirement? (Choose two.)

A. Use the Performance Insights feature of Amazon RDS to identify queries that have high CPU utilization. Optimize the problematic queries.

B. Modify the database schema to include additional tables and indexes.

C. Reboot the RDS DB instance once each week.

D. Upgrade to a larger instance size.

E. Implement caching to reduce the database query load.

Suggested Answer: *AD*

Community vote distribution

AD (75%) BD (25%)

⊟ **kj07** Highly Voted 1 year, 3 months ago

Here the issue is with the writes and caching will not solve them.

I will go with A and D.

upvoted 10 times

⊟ **lucas_rfsb** Highly Voted 1 year, 2 months ago

Selected Answer: AD

I will go for A and D, since other options are more likely to improve read performance issues.

upvoted 9 times

⊟ **michele_scar** Most Recent 7 months, 3 weeks ago

Selected Answer: AD

With A you should understand why the CPU is in high loading. B is mentioned in the last phrase of A (optimizing). Remain valid only D

upvoted 2 times

⊟ **sdas1** 1 year, 1 month ago

A and D

With a workload that is mostly writes and a small number of reads, caching will not be as effective in reducing CPU utilization compared to read-heavy workloads.

https://repost.aws/knowledge-center/rds-aurora-postgresql-high-cpu

upvoted 2 times

⊟ **fceb2c1** 1 year, 3 months ago

Selected Answer: AD

A and D.

For A it is mentioned here https://repost.aws/knowledge-center/rds-instance-high-cpu

upvoted 5 times

⊟ **GiorgioGss** 1 year, 3 months ago

Selected Answer: BD

Since the questions states that "the database workload is mostly writes" let's eliminate the options that improves the reads.

upvoted 5 times

⊟ **damaldon** 1 year, 3 months ago

Ans. AE

A) Use Amazon RDS Performance Insights to identify the query that's responsible for the database load. Check the SQL tab that corresponds to a particular timeframe.

E) If there's a query that's repeatedly running, use prepared statements to lower the pressure on your CPU. Repeated running of prepared statements caches the query plan. Because the plan is already in cache for further runs, the time for planning is much less.

https://repost.aws/knowledge-center/rds-aurora-postgresql-high-cpu

upvoted 4 times

Question #62 *Topic 1*

A company has used an Amazon Redshift table that is named Orders for 6 months. The company performs weekly updates and deletes on the table. The table has an interleaved sort key on a column that contains AWS Regions.

The company wants to reclaim disk space so that the company will not run out of storage space. The company also wants to analyze the sort key column.

Which Amazon Redshift command will meet these requirements?

A. VACUUM FULL Orders

B. VACUUM DELETE ONLY Orders

C. VACUUM REINDEX Orders

D. VACUUM SORT ONLY Orders

Suggested Answer: *C*

Community vote distribution

C (88%) 13%

GiorgioGss Highly Voted 👍 1 year, 3 months ago

Selected Answer: C

https://docs.aws.amazon.com/redshift/latest/dg/r_VACUUM_command.html

"A full vacuum doesn't perform a reindex for interleaved tables. To reindex interleaved tables followed by a full vacuum, use the VACUUM REINDEX option."

A - "A full vacuum doesn't perform a reindex for interleaved tables."- from the docs above

B- "A DELETE ONLY vacuum operation doesn't sort table data." - from the docs above

D - "without reclaiming space freed by deleted rows. " - from the docs above

upvoted 12 times

Shatheesh Most Recent ⊙ 9 months ago

Selected Answer: A

upvoted 1 times

d8945a1 1 year, 1 month ago

Selected Answer: C

VACUUM REINDEX makes an additional pass to analyze the interleaved sort keys.

https://docs.aws.amazon.com/redshift/latest/dg/r_VACUUM_command.html#r_VACUUM_command-parameters

upvoted 1 times

Christina666 1 year, 2 months ago

Selected Answer: C

Reclaiming Space: After updates and deletes, Redshift tables can retain deleted data blocks, taking up space. The VACUUM REINDEX command:

Reclaims the space taken up by the deleted rows.

Rebuilds indexes on the sort key columns.

Analyzing the Sort Key: Since the sort key column contains AWS Regions, rebuilding the indexes on this column will help cluster data according to region. This clustering can improve performance for queries that filter or group by region.

upvoted 1 times

arvehisa 1 year, 2 months ago

Selected Answer: C

Correct Answer: C

Requirements:

1. relcaim the disk space

2. analyze the sork key column

Document: https://docs.aws.amazon.com/redshift/latest/dg/r_VACUUM_command.html#vacuum-reindex

VACUUM FULL: A full vacuum doesn't perform a reindex for interleaved tables. To reindex interleaved tables followed by a full vacuum, use the VACUUM REINDEX option.
VACUUM REINDEX: Analyzes the distribution of the values in interleaved sort key columns, then performs a full VACUUM operation.

upvoted 1 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: A

FULL is the only one which claims space and sorts.
FULL
Sorts the specified table (or all tables in the current database) and reclaims disk space occupied by rows that were marked for deletion by previous UPDATE and DELETE operations. VACUUM FULL is the default.

https://docs.aws.amazon.com/redshift/latest/dg/r_VACUUM_command.html

upvoted 3 times

damaldon 1 year, 3 months ago

B is the answer

upvoted 1 times

kj07 1 year, 3 months ago

Option C

Analyzes the distribution of the values in interleaved sort key columns, then performs a full VACUUM operation. If REINDEX is used, a table name is required.

VACUUM REINDEX takes significantly longer than VACUUM FULL because it makes an additional pass to analyze the interleaved sort keys. The sort and merge operation can take longer for interleaved tables because the interleaved sort might need to rearrange more rows than a compound sort.

If a VACUUM REINDEX operation terminates before it completes, the next VACUUM resumes the reindex operation before performing the full vacuum operation.

upvoted 1 times

CalvinL4 1 year, 3 months ago

The answer should be B. The VACUUM DELETE ONLY command is used in Amazon Redshift to remove rows that have been marked for deletion due to updates and deletes in a table.

upvoted 1 times

Question #63

Topic 1

A manufacturing company wants to collect data from sensors. A data engineer needs to implement a solution that ingests sensor data in near real time.

The solution must store the data to a persistent data store. The solution must store the data in nested JSON format. The company must have the ability to query from the data store with a latency of less than 10 milliseconds.

Which solution will meet these requirements with the LEAST operational overhead?

A. Use a self-hosted Apache Kafka cluster to capture the sensor data. Store the data in Amazon S3 for querying.

B. Use AWS Lambda to process the sensor data. Store the data in Amazon S3 for querying.

C. Use Amazon Kinesis Data Streams to capture the sensor data. Store the data in Amazon DynamoDB for querying.

D. Use Amazon Simple Queue Service (Amazon SQS) to buffer incoming sensor data. Use AWS Glue to store the data in Amazon RDS for querying.

Suggested Answer: *C*

Community vote distribution

C (100%)

pypelyncar 1 year ago

Selected Answer: C

Amazon Kinesis Data Streams is a fully managed service that allows for seamless integration of diverse data sources, including IoT sensors. By using Kinesis Data Streams as the ingestion mechanism, the company can avoid the overhead of setting up and managing an Apache Kafka cluster or other data ingestion pipelines.

upvoted 2 times

Snape 1 year, 1 month ago

Selected Answer: C

near real time = Kinesis Data streams

upvoted 3 times

GustonMari 11 months, 3 weeks ago

to be more accurate,

Kinesis Data streams = real time

Kinesis Data Firehose = near real time

upvoted 4 times

Ousseyni 1 year, 2 months ago

Selected Answer: C

Option C is the best solution to meet the requirements

upvoted 1 times

Felix_G 1 year, 4 months ago

Option C is the best solution to meet the requirements with the least operational overhead:

Use Amazon Kinesis Data Streams to ingest real-time sensor data
Store the nested JSON data in Amazon DynamoDB for low latency queries
The key advantages of Option C are:

Kinesis Data Streams fully manages real-time data ingestion with auto-scaling and persistence
DynamoDB provides single digit millisecond latency for queries
DynamoDB natively supports nested JSON data models
Fully managed services minimize operational overhead
In contrast:

Option A requires managing Kafka clusters
Option B uses Lambda which can't provide persistent storage
Option D requires integrating SQS, Glue, and RDS leading to complexity

upvoted 3 times

⊟ **rralucard_** 1 year, 4 months ago

Selected Answer: C

Option C, using Amazon Kinesis Data Streams to capture the sensor data and storing it in Amazon DynamoDB for querying, is the best solution to meet the requirements with the least operational overhead. This solution is well-optimized for real-time data ingestion, supports the desired data format, and provides the necessary query performance.

upvoted 3 times

Question #64 *Topic 1*

A company stores data in a data lake that is in Amazon S3. Some data that the company stores in the data lake contains personally identifiable information (PII). Multiple user groups need to access the raw data. The company must ensure that user groups can access only the PII that they require.
Which solution will meet these requirements with the LEAST effort?

A. Use Amazon Athena to query the data. Set up AWS Lake Formation and create data filters to establish levels of access for the company's IAM roles. Assign each user to the IAM role that matches the user's PII access requirements.

B. Use Amazon QuickSight to access the data. Use column-level security features in QuickSight to limit the PII that users can retrieve from Amazon S3 by using Amazon Athena. Define QuickSight access levels based on the PII access requirements of the users.

C. Build a custom query builder UI that will run Athena queries in the background to access the data. Create user groups in Amazon Cognito. Assign access levels to the user groups based on the PII access requirements of the users.

D. Create IAM roles that have different levels of granular access. Assign the IAM roles to IAM user groups. Use an identity-based policy to assign access levels to user groups at the column level.

Suggested Answer: *A*

Community vote distribution

A (100%)

lucas_rfsb 9 months ago

Selected Answer: A

Amazon Athena to query the data and setting up AWS Lake Formation with data filters, the company can ensure that user groups can access only the personally identifiable information (PII) that they require. The combination of Athena for querying and Lake Formation for access control provides a comprehensive solution for managing PII access requirements effectively and securely

upvoted 3 times

Felix_G 9 months, 4 weeks ago

Selected Answer: A

The solution that will meet the requirements with the LEAST effort is:

A. Use Amazon Athena to query the data. Set up AWS Lake Formation and create data filters to establish levels of access for the company's IAM roles. Assign each user to the IAM role that matches the user's PII access requirements.

This option leverages AWS Lake Formation to create data filters and establish access levels for IAM roles, providing a straightforward approach to managing user access based on PII requirements.

upvoted 2 times

rralucard_ 11 months ago

Selected Answer: A

Option A, using Amazon Athena with AWS Lake Formation, is the most suitable solution. Lake Formation is designed to provide fine-grained access control to data lakes stored in S3 and integrates well with Athena, thereby meeting the requirements with the least effort.
https://aws.amazon.com/blogs/big-data/anonymize-and-manage-data-in-your-data-lake-with-amazon-athena-and-aws-lake-formation/

upvoted 4 times

Question #65 *Topic 1*

A data engineer must build an extract, transform, and load (ETL) pipeline to process and load data from 10 source systems into 10 tables that are in an Amazon Redshift database. All the source systems generate .csv, JSON, or Apache Parquet files every 15 minutes. The source systems all deliver files into one Amazon S3 bucket. The file sizes range from 10 MB to 20 GB. The ETL pipeline must function correctly despite changes to the data schema.
Which data pipeline solutions will meet these requirements? (Choose two.)

A. Use an Amazon EventBridge rule to run an AWS Glue job every 15 minutes. Configure the AWS Glue job to process and load the data into the Amazon Redshift tables.

B. Use an Amazon EventBridge rule to invoke an AWS Glue workflow job every 15 minutes. Configure the AWS Glue workflow to have an on-demand trigger that runs an AWS Glue crawler and then runs an AWS Glue job when the crawler finishes running successfully. Configure the AWS Glue job to process and load the data into the Amazon Redshift tables.

C. Configure an AWS Lambda function to invoke an AWS Glue crawler when a file is loaded into the S3 bucket. Configure an AWS Glue job to process and load the data into the Amazon Redshift tables. Create a second Lambda function to run the AWS Glue job. Create an Amazon EventBridge rule to invoke the second Lambda function when the AWS Glue crawler finishes running successfully.

D. Configure an AWS Lambda function to invoke an AWS Glue workflow when a file is loaded into the S3 bucket. Configure the AWS Glue workflow to have an on-demand trigger that runs an AWS Glue crawler and then runs an AWS Glue job when the crawler finishes running successfully. Configure the AWS Glue job to process and load the data into the Amazon Redshift tables.

E. Configure an AWS Lambda function to invoke an AWS Glue job when a file is loaded into the S3 bucket. Configure the AWS Glue job to read the files from the S3 bucket into an Apache Spark DataFrame. Configure the AWS Glue job to also put smaller partitions of the DataFrame into an Amazon Kinesis Data Firehose delivery stream. Configure the delivery stream to load data into the Amazon Redshift tables.

Suggested Answer: *BD*

Community vote distribution

BD (93%) 7%

rralucard_ Highly Voted 10 months, 4 weeks ago

Selected Answer: BD

Option B: Amazon EventBridge Rule with AWS Glue Workflow Job Every 15 Minutes - for its streamlined process, automated scheduling, and ability to handle schema changes.

Option D: AWS Lambda to Invoke AWS Glue Workflow When a File is Loaded - for its responsiveness to file arrival and adaptability to schema changes, though it is slightly more complex than option B.

upvoted 10 times

Felix_G 9 months, 4 weeks ago

D is incorrect! Options C, D and E have issues like unnecessary complexity, latency due to triggers, or limitations in handling large file sizes. So B and A are the best and most robust options that meet all the requirements.

upvoted 1 times

Luke97 9 months ago

A is NOT correct. The question said "The ETL pipeline must function correctly despite changes to the data schema", therefore run Glue crawler is necessary to handle schema changes.

upvoted 5 times

HagarTheHorrible Most Recent 6 months, 1 week ago

Selected Answer: BD

change od schema is the key

upvoted 1 times

valuedate 7 months ago

Selected Answer: BD

eventbridge rule or event trigger

upvoted 1 times

Ousseyni 8 months, 2 weeks ago

Selected Answer: AE

ChatCGT sid A and E

upvoted 1 times

⊟ valuedate 7 months ago

ChatCGT? ahahaha. A is NOT correct and E its too complex

upvoted 2 times

⊟ tgv 7 months ago

You should double check your information.

upvoted 2 times

⊟ Christina666 8 months, 2 weeks ago

Selected Answer: BD

eventbridge rule or event trigger

upvoted 1 times

⊟ arvehisa 8 months, 3 weeks ago

I don't think this pipeline should be triggered by an s3 file upload. However seems A cannot handle the data schema change.

if s3 trigger is good, then C and E are unnessessarily complexed. so I would go with B & D (despite the s3 trigger)

upvoted 1 times

⊟ lucas_rfsb 9 months ago

Selected Answer: BD

I will go with BD

upvoted 3 times

⊟ Felix_G 9 months, 4 weeks ago

Selected Answer: AB

The two data pipeline solutions that will meet the requirements are:

A. Use an Amazon EventBridge rule to run an AWS Glue job every 15 minutes. Configure the AWS Glue job to process and load the data into the Amazon Redshift tables.

B. Use an Amazon EventBridge rule to invoke an AWS Glue workflow job every 15 minutes. Configure the AWS Glue workflow to have an on-demand trigger that runs an AWS Glue crawler and then runs an AWS Glue job when the crawler finishes running successfully. Configure the AWS Glue job to process and load the data into the Amazon Redshift tables.

These solutions leverage AWS Glue to process and load the data from different file formats in the S3 bucket into the Amazon Redshift tables, while also handling changes to the data schema.

upvoted 2 times

⊟ chris_spencer 8 months, 2 weeks ago

A is incorret, it doesn't take care to update the data catalog.

upvoted 1 times

⊟ evntdrvn76 11 months ago

The correct answers are A. Use an Amazon EventBridge rule to run an AWS Glue job every 15 minutes. Configure the AWS Glue job to process and load the data into the Amazon Redshift tables and B. Use an Amazon EventBridge rule to invoke an AWS Glue workflow job every 15 minutes. Configure the AWS Glue workflow to have an on-demand trigger that runs an AWS Glue crawler and then runs an AWS Glue job when the crawler finishes running successfully. Configure the AWS Glue job to process and load the data into the Amazon Redshift tables. These solutions automate the ETL pipeline with minimal operational overhead.

upvoted 1 times

Question #66 *Topic 1*

A financial company wants to use Amazon Athena to run on-demand SQL queries on a petabyte-scale dataset to support a business intelligence (BI) application. An AWS Glue job that runs during non-business hours updates the dataset once every day. The BI application has a standard data refresh frequency of 1 hour to comply with company policies.
A data engineer wants to cost optimize the company's use of Amazon Athena without adding any additional infrastructure costs.
Which solution will meet these requirements with the LEAST operational overhead?

A. Configure an Amazon S3 Lifecycle policy to move data to the S3 Glacier Deep Archive storage class after 1 day.

B. Use the query result reuse feature of Amazon Athena for the SQL queries.

C. Add an Amazon ElastiCache cluster between the BI application and Athena.

D. Change the format of the files that are in the dataset to Apache Parquet.

Suggested Answer: *B*

Community vote distribution

B (100%)

⊟ **rralucard_** Highly Voted 1 year, 4 months ago

Selected Answer: B

https://docs.aws.amazon.com/athena/latest/ug/performance-tuning.html
Use the Query Result Reuse Feature of Amazon Athena. This leverages Athena's built-in feature to reduce redundant data scans and thus lowers query costs.

upvoted 6 times

⊟ **DevoteamAnalytix** 1 year, 1 month ago

Yes, seems to be B: https://aws.amazon.com/de/blogs/big-data/reduce-cost-and-improve-query-performance-with-amazon-athena-query-result-reuse/

upvoted 1 times

⊟ **Ell89** Most Recent 4 months ago

Selected Answer: D

D
query result reuse will benefit the same queries that are being re-run, it wont benefit new queries. parquet will benefit all queries.

upvoted 1 times

⊟ **rsmf** 8 months, 1 week ago

Selected Answer: B

Why not D? The question specifies the option with the least overhead, and it clearly states that the Glue job runs once a day. Since the data for that day will not change, there’s no need for additional overhead.

upvoted 1 times

⊟ **MinTheRanger** 11 months ago

D. Because "query reuse feature" is reliable only when it's identical but here hourly refresh might be on data related to that hour.

upvoted 1 times

⊟ **MinTheRanger** 11 months ago

Why not D?

upvoted 3 times

⊟ **Ousseyni** 1 year, 2 months ago

Selected Answer: B

B. Use the query result reuse feature of Amazon Athena for the SQL queries.

upvoted 2 times

⊟ **FuriouZ** 1 year, 3 months ago

Selected Answer: B

It's B: Glacier adds more retrieval time and the other options cost some money

upvoted 1 times

Question #67 *Topic 1*

A company's data engineer needs to optimize the performance of table SQL queries. The company stores data in an Amazon Redshift cluster. The data engineer cannot increase the size of the cluster because of budget constraints.

The company stores the data in multiple tables and loads the data by using the EVEN distribution style. Some tables are hundreds of gigabytes in size. Other tables are less than 10 MB in size.

Which solution will meet these requirements?

A. Keep using the EVEN distribution style for all tables. Specify primary and foreign keys for all tables.

B. Use the ALL distribution style for large tables. Specify primary and foreign keys for all tables.

C. Use the ALL distribution style for rarely updated small tables. Specify primary and foreign keys for all tables.

D. Specify a combination of distribution, sort, and partition keys for all tables.

Suggested Answer: *C*

Community vote distribution

C (100%)

rralucard_ Highly Voted 1 year, 4 months ago

Selected Answer: C

Use the ALL Distribution Style for Rarely Updated Small Tables. This approach optimizes the performance of joins involving these smaller tables and is a common best practice in Redshift data warehousing. For the larger tables, maintaining the EVEN distribution style or considering a KEY-based distribution (if there are common join columns) could be more appropriate.

upvoted 8 times

Tester_TKK Most Recent 2 months, 1 week ago

Selected Answer: C

D is wrong. There is no partition key in Redshift

upvoted 1 times

jk15997 7 months, 2 weeks ago

why not D?

upvoted 3 times

pypelyncar 1 year ago

Selected Answer: C

For small tables (less than 10 MB in size) that are rarely updated, using the ALL distribution style can provide better query performance. With the ALL distribution style, each compute node stores a copy of the entire table, eliminating the need for data redistribution or shuffling during certain queries. This can significantly improve query performance, especially for joins and aggregations involving small tables.

upvoted 3 times

DevoteamAnalytix 1 year, 1 month ago

Selected Answer: C

"ALL distribution is appropriate only for relatively slow moving tables; that is, tables that are not updated frequently or extensively." (https://docs.aws.amazon.com/redshift/latest/dg/c_choosing_dist_sort.html)

upvoted 2 times

Question #68 *Topic 1*

A company receives .csv files that contain physical address data. The data is in columns that have the following names: Door_No, Street_Name, City, and Zip_Code. The company wants to create a single column to store these values in the following format:

```
{
    "Door_No": "24",
    "Street_Name": "AAA street",
    "City": "BBB",
    "Zip_Code": "111111"
}
```

Which solution will meet this requirement with the LEAST coding effort?

A. Use AWS Glue DataBrew to read the files. Use the NEST_TO_ARRAY transformation to create the new column.

B. Use AWS Glue DataBrew to read the files. Use the NEST_TO_MAP transformation to create the new column.

C. Use AWS Glue DataBrew to read the files. Use the PIVOT transformation to create the new column.

D. Write a Lambda function in Python to read the files. Use the Python data dictionary type to create the new column.

Suggested Answer: *B*

Community vote distribution

B (95%) 5%

FuriouZ Highly Voted 1 year, 3 months ago

Selected Answer: B

NEST_TO_ARRAY would result in:
[{"key": "key1", "value": "value1"}, {"key": "key2", "value": "value2"}, {"key": "key3", "value": "value3"}]

while NEST_TO_MAP results: {
"key1": "value1",
"key2": "value2",
"key3": "value3"
}
Therefore go with B

upvoted 14 times

pypelyncar Most Recent 1 year ago

Selected Answer: B

The NEST_TO_MAP transformation is specifically designed to convert data from nested structures (like rows in a CSV) into key-value pairs, perfectly matching the requirement of creating a new column with address components as key-value pairs

upvoted 3 times

Ousseyni 1 year, 2 months ago

Selected Answer: B

AWS Glue DataBrew is a visual data preparation tool that allows for easy transformation of data without requiring extensive coding. The NEST_TO_MAP transformation in DataBrew allows you to convert columns into a JSON map, which aligns with the desired JSON format for the address data.

upvoted 4 times

GiorgioGss 1 year, 3 months ago

Selected Answer: A

Come on guys. That's and array there so...

upvoted 1 times

GiorgioGss 1 year, 3 months ago

I take that back. I will go with B because NEST_TO_ARRAY is not suitable for the desired JSON format where each attribute has its own key.

upvoted 2 times

⊟ **kj07** 1 year, 3 months ago

Option B:

NEST_TO_MAP: Converts user-selected columns into key-value pairs, each with a key representing the column name and a value representing the row value. The order of the selected column is not maintained while creating the resultant map. The different column data types are typecast to a common type that supports the data types of all columns.

https://docs.aws.amazon.com/databrew/latest/dg/recipe-actions.NEST_TO_MAP.html

PIVOT: Converts all the row values in a selected column into individual columns with values.

NEST_TO_ARRAY: Converts user-selected columns into array values. The order of the selected columns is maintained while creating the resultant array. The different column data types are typecast to a common type that supports the data types of all columns.

upvoted 2 times

⊟ **damaldon** 1 year, 3 months ago

Ans. A

NEST_TO_ARRAY Converts user-selected columns into array values. The order of the selected columns is maintained while creating the resultant array.

https://docs.aws.amazon.com/databrew/latest/dg/recipe-actions.NEST_TO_ARRAY.html

upvoted 1 times

Question #69 *Topic 1*

A company receives call logs as Amazon S3 objects that contain sensitive customer information. The company must protect the S3 objects by using encryption. The company must also use encryption keys that only specific employees can access.
Which solution will meet these requirements with the LEAST effort?

A. Use an AWS CloudHSM cluster to store the encryption keys. Configure the process that writes to Amazon S3 to make calls to CloudHSM to encrypt and decrypt the objects. Deploy an IAM policy that restricts access to the CloudHSM cluster.

B. Use server-side encryption with customer-provided keys (SSE-C) to encrypt the objects that contain customer information. Restrict access to the keys that encrypt the objects.

C. Use server-side encryption with AWS KMS keys (SSE-KMS) to encrypt the objects that contain customer information. Configure an IAM policy that restricts access to the KMS keys that encrypt the objects.

D. Use server-side encryption with Amazon S3 managed keys (SSE-S3) to encrypt the objects that contain customer information. Configure an IAM policy that restricts access to the Amazon S3 managed keys that encrypt the objects.

Suggested Answer: *C*

Community vote distribution

C (100%)

Ousseyni 8 months, 2 weeks ago

Selected Answer: C

C. Use server-side encryption with AWS KMS keys (SSE-KMS) to encrypt the objects that contain customer information. Configure an IAM policy that restricts access to the KMS keys that encrypt the objects.

Server-side encryption with AWS KMS (SSE-KMS) provides strong encryption for S3 objects while allowing fine-grained access control through AWS Key Management Service (KMS). With SSE-KMS, you can control access to encryption keys using IAM policies, ensuring that only specific employees can access them.

This solution requires minimal effort as it leverages AWS's managed encryption service (SSE-KMS) and integrates seamlessly with S3. Additionally, IAM policies can be easily configured to restrict access to the KMS keys, providing granular control over who can access the encryption keys.

upvoted 3 times

Christina666 8 months, 2 weeks ago

Selected Answer: C

Encryption at Rest: SSE-KMS provides robust encryption of the sensitive call log data while it's stored in S3.
Key Management and Access Control: AWS KMS offers centralized key management. You can easily create and manage KMS keys (Customer Master Keys - CMKs) and use fine-grained IAM policies to restrict access to specific employees.
Minimal Effort: SSE-KMS is a built-in S3 feature. Enabling it requires minimal configuration and no custom code for encryption/decryption.

upvoted 3 times

FuriouZ 9 months ago

Selected Answer: C

KMS because you can restrict access and of course for pricing ;)

upvoted 1 times

GiorgioGss 9 months, 2 weeks ago

Selected Answer: C

Least effort = C

upvoted 4 times

rralucard_ 11 months ago

Selected Answer: C

Option D does not provide the ability to restrict access to the encryption keys

upvoted 4 times

Question #70 *Topic 1*

A company stores petabytes of data in thousands of Amazon S3 buckets in the S3 Standard storage class. The data supports analytics workloads that have unpredictable and variable data access patterns.

The company does not access some data for months. However, the company must be able to retrieve all data within milliseconds. The company needs to optimize S3 storage costs.

Which solution will meet these requirements with the LEAST operational overhead?

A. Use S3 Storage Lens standard metrics to determine when to move objects to more cost-optimized storage classes. Create S3 Lifecycle policies for the S3 buckets to move objects to cost-optimized storage classes. Continue to refine the S3 Lifecycle policies in the future to optimize storage costs.

B. Use S3 Storage Lens activity metrics to identify S3 buckets that the company accesses infrequently. Configure S3 Lifecycle rules to move objects from S3 Standard to the S3 Standard-Infrequent Access (S3 Standard-IA) and S3 Glacier storage classes based on the age of the data.

C. Use S3 Intelligent-Tiering. Activate the Deep Archive Access tier.

D. Use S3 Intelligent-Tiering. Use the default access tier.

Suggested Answer: *D*

Community vote distribution

D (88%) 12%

GiorgioGss Highly Voted 1 year, 3 months ago

Selected Answer: D

Although C is more cost-effective, because of "must be able to retrieve all data within milliseconds" will go with D

upvoted 12 times

andrologin Most Recent 11 months, 3 weeks ago

Selected Answer: D

Based on this docs https://docs.aws.amazon.com/AmazonS3/latest/userguide/intelligent-tiering-overview.html

D will be appropriate as it allows for instant retrieval

upvoted 2 times

rpwags 1 year ago

Selected Answer: D

Staying with "D"... The Amazon S3 Glacier Deep Archive storage class is designed for long-term data archiving where data retrieval times are flexible. It does not offer millisecond retrieval times. Instead, data retrieval from S3 Glacier Deep Archive typically takes 12 hours or more. For millisecond retrieval times, you would use the S3 Standard, S3 Standard-IA, or S3 One Zone-IA storage classes, which are designed for frequent or infrequent access with low latency.

upvoted 3 times

raghumvj 1 year, 2 months ago

Selected Answer: D

I am confused with C or D

upvoted 2 times

chris_spencer 1 year, 2 months ago

Selected Answer: C

C is correct.

"Amazon S3 Glacier Instant Retrieval is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds."

https://aws.amazon.com/s3/storage-classes/glacier/instant-retrieval/

upvoted 1 times

tgv 1 year, 1 month ago

But C doesn't say anything about Instant Retrieval.

upvoted 3 times

Christina666 1 year, 2 months ago

Selected Answer: D

least operation overhead, D

upvoted 2 times

arvehisa 1 year, 2 months ago

The correct answer may be D. Intelligent tiering's default access tier is:

1. accessed less than 30 days: frequent access tier
2. not accessed in 30-90 days: Infrequent Access tier
3. not accessed more than 90 days: Archive Instant Access tier

Other tiers require more retrieve time need activation.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/intelligent-tiering-overview.html

upvoted 2 times

helpaws 1 year, 3 months ago

Selected Answer: C

Amazon S3 Glacier Instant Retrieval is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds

upvoted 2 times

kj07 1 year, 3 months ago

A few remarks: Data should be retrieved in ms. This means all the options with Glacier are wrong: BC

For D how you can set the S3 intelligent-Tiering if the current class is Standard?
I guess you need a lifecycle policy.

Which leaves only A as an option.

Thoughts?

upvoted 1 times

damaldon 1 year, 3 months ago

D. is correct

upvoted 1 times

Felix_G 1 year, 3 months ago

Option C. Use S3 Intelligent-Tiering. Activate the Deep Archive Access tier.

By using S3 Intelligent-Tiering and activating the Deep Archive Access tier, the company can optimize S3 storage costs with minimal operational overhead. S3 Intelligent-Tiering automatically moves objects between four access tiers, including the Deep Archive Access tier, based on changing access patterns and cost optimization. This eliminates the need for manual lifecycle policies and constant refinement, as the storage class is adjusted automatically based on data access patterns, resulting in cost savings while ensuring quick access to all data when needed.

upvoted 1 times

rralucard_ 1 year, 4 months ago

Selected Answer: D

Option D, using S3 Intelligent-Tiering with the default access tier, will meet the requirements best. It provides a hands-off approach to storage cost optimization while ensuring that data is available for analytics workloads within the required timeframe.

upvoted 3 times

Question #71 *Topic 1*

During a security review, a company identified a vulnerability in an AWS Glue job. The company discovered that credentials to access an Amazon Redshift cluster were hard coded in the job script.
A data engineer must remediate the security vulnerability in the AWS Glue job. The solution must securely store the credentials.
Which combination of steps should the data engineer take to meet these requirements? (Choose two.)

A. Store the credentials in the AWS Glue job parameters.

B. Store the credentials in a configuration file that is in an Amazon S3 bucket.

C. Access the credentials from a configuration file that is in an Amazon S3 bucket by using the AWS Glue job.

D. Store the credentials in AWS Secrets Manager.

E. Grant the AWS Glue job IAM role access to the stored credentials.

Suggested Answer: *DE*

Community vote distribution

DE (100%)

GiorgioGss Highly Voted 9 months, 2 weeks ago

Selected Answer: DE

D because it's AWS best practice for securing creds and E because after you put cred in secrets you will need permissions for accesing

upvoted 9 times

damaldon Most Recent 9 months, 3 weeks ago

Ans, DE

upvoted 1 times

rralucard_ 11 months ago

Selected Answer: DE

D. Store the credentials in AWS Secrets Manager: AWS Secrets Manager is a service that helps you protect access to your applications, services, and IT resources without the upfront investment and on-going maintenance costs of operating your own infrastructure. It's specifically designed for storing and retrieving credentials securely, and therefore, it is an appropriate choice for handling the Redshift cluster credentials.

E. Grant the AWS Glue job IAM role access to the stored credentials: IAM roles for AWS Glue will allow the job to assume a role with the necessary permissions to access the credentials in AWS Secrets Manager. This method avoids embedding credentials directly in the script or a configuration file and allows for centralized management of the credentials.

upvoted 3 times

Question #72 *Topic 1*

A data engineer uses Amazon Redshift to run resource-intensive analytics processes once every month. Every month, the data engineer creates a new Redshift provisioned cluster. The data engineer deletes the Redshift provisioned cluster after the analytics processes are complete every month. Before the data engineer deletes the cluster each month, the data engineer unloads backup data from the cluster to an Amazon S3 bucket.
The data engineer needs a solution to run the monthly analytics processes that does not require the data engineer to manage the infrastructure manually.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon Step Functions to pause the Redshift cluster when the analytics processes are complete and to resume the cluster to run new processes every month.

B. Use Amazon Redshift Serverless to automatically process the analytics workload.

C. Use the AWS CLI to automatically process the analytics workload.

D. Use AWS CloudFormation templates to automatically process the analytics workload.

Suggested Answer: *B*

Community vote distribution

B (100%)

Christina666 Highly Voted 👍 8 months, 2 weeks ago

Selected Answer: B

Fully Managed, Serverless: Redshift Serverless eliminates the need to manually create, manage, or delete clusters. It automatically scales resources based on the workload, reducing operational overhead significantly.
Cost-Effective for Infrequent Workloads: Since the analytics processes run only once a month, Redshift Serverless's pay-per-use model is ideal for minimizing costs during downtime.
Seamless S3 Integration: Redshift Serverless natively integrates with S3 for backup and restore operations, ensuring compatibility with the existing process.

upvoted 5 times

4c78df0 Most Recent 7 months ago

Selected Answer: B

B is correct

upvoted 1 times

GiorgioGss 9 months, 2 weeks ago

Selected Answer: B

"does not require to manage the infrastructure manually" = Serverless

upvoted 2 times

damaldon 9 months, 3 weeks ago

Ans. B

upvoted 1 times

Felix_G 9 months, 4 weeks ago

Selected Answer B: Options A, C and D still involve manual tasks like administering CloudFormation stacks, using AWS CLI commands, or configuring Step Function state machines.

By leveraging Redshift Serverless, the data engineer avoids all cluster and infrastructure administration effort. This has the least operational overhead to run the monthly

upvoted 3 times

rralucard_ 11 months ago

Selected Answer: B

Use Amazon Redshift Serverless. This option allows the data engineer to focus on the analytics processes themselves without worrying about cluster provisioning, scaling, or management. It provides an on-demand, serverless solution that can handle variable workloads and is cost-effective for intermittent and irregular processing needs like those described.

upvoted 2 times

Question #73 *Topic 1*

A company receives a daily file that contains customer data in .xls format. The company stores the file in Amazon S3. The daily file is approximately 2 GB in size.

A data engineer concatenates the column in the file that contains customer first names and the column that contains customer last names. The data engineer needs to determine the number of distinct customers in the file.

Which solution will meet this requirement with the LEAST operational effort?

A. Create and run an Apache Spark job in an AWS Glue notebook. Configure the job to read the S3 file and calculate the number of distinct customers.

B. Create an AWS Glue crawler to create an AWS Glue Data Catalog of the S3 file. Run SQL queries from Amazon Athena to calculate the number of distinct customers.

C. Create and run an Apache Spark job in Amazon EMR Serverless to calculate the number of distinct customers.

D. Use AWS Glue DataBrew to create a recipe that uses the COUNT_DISTINCT aggregate function to calculate the number of distinct customers.

Suggested Answer: *D*

Community vote distribution

D (100%)

rralucard_ Highly Voted 👍 1 year, 4 months ago

Selected Answer: D

AWS Glue DataBrew: AWS Glue DataBrew is a visual data preparation tool that allows data engineers and data analysts to clean and normalize data without writing code. Using DataBrew, a data engineer could create a recipe that includes the concatenation of the customer first and last names and then use the COUNT_DISTINCT function. This would not require complex code and could be performed through the DataBrew user interface, representing a lower operational effort.

upvoted 9 times

Juan_pc Most Recent 🕒 1 month, 3 weeks ago

Selected Answer: A

According to the official DataBrew documentation, it does not natively support files in .xls format (it does support .xlsx).

The correct option is A.

upvoted 1 times

pypelyncar 1 year ago

Selected Answer: D

DataBrew supports various transformations,

including the COUNT_DISTINCT function, which is ideal for calculating the number of unique values in a column (combined first and last names in this case).

upvoted 2 times

Ousseyni 1 year, 2 months ago

Selected Answer: D

go in D

upvoted 2 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: D

since it's less operational effort, I would go in D

upvoted 2 times

Question #74 *Topic 1*

A healthcare company uses Amazon Kinesis Data Streams to stream real-time health data from wearable devices, hospital equipment, and patient records.

A data engineer needs to find a solution to process the streaming data. The data engineer needs to store the data in an Amazon Redshift Serverless warehouse. The solution must support near real-time analytics of the streaming data and the previous day's data.

Which solution will meet these requirements with the LEAST operational overhead?

A. Load data into Amazon Kinesis Data Firehose. Load the data into Amazon Redshift.

B. Use the streaming ingestion feature of Amazon Redshift.

C. Load the data into Amazon S3. Use the COPY command to load the data into Amazon Redshift.

D. Use the Amazon Aurora zero-ETL integration with Amazon Redshift.

Suggested Answer: *B*

Community vote distribution

B (100%)

⊟ **rralucard_** Highly Voted 1 year, 4 months ago

Selected Answer: B

https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-streaming-ingestion.html

Use the Streaming Ingestion Feature of Amazon Redshift: Amazon Redshift recently introduced streaming data ingestion, allowing Redshift to consume data directly from Kinesis Data Streams in near real-time. This feature simplifies the architecture by eliminating the need for intermediate steps or services, and it is specifically designed to support near real-time analytics. The operational overhead is minimal since the feature is integrated within Redshift.

upvoted 7 times

⊟ **ssnei** Most Recent 8 months ago

option B

upvoted 1 times

⊟ **4c78df0** 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 1 times

⊟ **lucas_rfsb** 1 year, 2 months ago

Selected Answer: B

I'd go in B

upvoted 1 times

Question #75 *Topic 1*

A data engineer needs to use an Amazon QuickSight dashboard that is based on Amazon Athena queries on data that is stored in an Amazon S3 bucket. When the data engineer connects to the QuickSight dashboard, the data engineer receives an error message that indicates insufficient permissions.
Which factors could cause to the permissions-related errors? (Choose two.)

A. There is no connection between QuickSight and Athena.

B. The Athena tables are not cataloged.

C. QuickSight does not have access to the S3 bucket.

D. QuickSight does not have access to decrypt S3 data.

E. There is no IAM role assigned to QuickSight.

Suggested Answer: *CD*

Community vote distribution

CD (90%) 10%

⊟ **fceb2c1** Highly Voted 1 year, 3 months ago

Selected Answer: CD

C and D
https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-athena-insufficient-permissions.html

E is incorrect because it will result in authentication/authorization error, not insufficient permission error.
upvoted 8 times

⊟ **rralucard_** Highly Voted 1 year, 4 months ago

Selected Answer: CD

C. QuickSight does not have access to the S3 bucket: Amazon QuickSight needs to have the necessary permissions to access the S3 bucket where the data resides. If QuickSight lacks the permissions to read the data from the S3 bucket, it would result in an error indicating insufficient permissions.

D. QuickSight does not have access to decrypt S3 data: If the data in S3 is encrypted, QuickSight needs permissions to use the necessary keys to decrypt the data. Without access to the decryption keys, typically managed by AWS Key Management Service (KMS), QuickSight cannot read the encrypted data and would give an error.
upvoted 6 times

⊟ **bonds** Most Recent 2 months, 1 week ago

Selected Answer: AC

QuickSight requires explicit permission to connect to Athena
This connection must be established during QuickSight setup
Without this connection, QuickSight cannot execute Athena queries
Results in permissions-related errors
upvoted 1 times

⊟ **bakarys** 12 months ago

Selected Answer: CE

C. QuickSight does not have access to the S3 bucket. Amazon QuickSight needs to have the necessary permissions to access the Amazon S3 bucket where the data is stored. If these permissions are not correctly configured, QuickSight will not be able to access the data, resulting in an error.

E. There is no IAM role assigned to QuickSight. Amazon QuickSight uses AWS Identity and Access Management (IAM) roles to access AWS resources. If QuickSight is not assigned an IAM role, or if the assigned role does not have the necessary permissions, QuickSight will not be able to access the resources it needs, leading to an error.
upvoted 2 times

⊟ **Ousseyni** 1 year, 2 months ago

Selected Answer: CD

C and D

upvoted 1 times

⊟ **Christina666** 1 year, 2 months ago

Selected Answer: CD

The two most likely factors causing the permissions-related errors are:

C. QuickSight does not have access to the S3 bucket. To access data from an S3 bucket, QuickSight needs explicit S3 permissions. This is typically handled through an IAM role associated with the QuickSight service.
D. QuickSight does not have access to decrypt S3 data. If the data in S3 is encrypted (e.g., using KMS), QuickSight must have the necessary permissions to decrypt the data using the relevant KMS key.
Let's analyze why the other options are less likely the primary culprits:

E. There is no IAM role assigned to QuickSight. QuickSight needs an IAM role for overall functionality. A missing role would likely cause broader service failures, not specific data access errors.

upvoted 4 times

⊟ **taka5094** 1 year, 3 months ago

Selected Answer: CE

I think the assumptions in the problem are insufficient. If the data is encrypted, then D can be the correct answer, but if not, then E is the correct answer.

upvoted 2 times

⊟ **damaldon** 1 year, 3 months ago

Ans. CD
https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-athena-insufficient-permissions.html

upvoted 2 times

Question #76 *Topic 1*

A company stores datasets in JSON format and .csv format in an Amazon S3 bucket. The company has Amazon RDS for Microsoft SQL Server databases, Amazon DynamoDB tables that are in provisioned capacity mode, and an Amazon Redshift cluster. A data engineering team must develop a solution that will give data scientists the ability to query all data sources by using syntax similar to SQL.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use AWS Glue to crawl the data sources. Store metadata in the AWS Glue Data Catalog. Use Amazon Athena to query the data. Use SQL for structured data sources. Use PartiQL for data that is stored in JSON format.

B. Use AWS Glue to crawl the data sources. Store metadata in the AWS Glue Data Catalog. Use Redshift Spectrum to query the data. Use SQL for structured data sources. Use PartiQL for data that is stored in JSON format.

C. Use AWS Glue to crawl the data sources. Store metadata in the AWS Glue Data Catalog. Use AWS Glue jobs to transform data that is in JSON format to Apache Parquet or .csv format. Store the transformed data in an S3 bucket. Use Amazon Athena to query the original and transformed data from the S3 bucket.

D. Use AWS Lake Formation to create a data lake. Use Lake Formation jobs to transform the data from all data sources to Apache Parquet format. Store the transformed data in an S3 bucket. Use Amazon Athena or Redshift Spectrum to query the data.

Suggested Answer: *A*

Community vote distribution

A (65%) B (17%) C (17%)

GiorgioGss Highly Voted 1 year, 3 months ago

Selected Answer: A

LEAST operational overhead? query straight with Athena without any intermediate actions or services

upvoted 7 times

pypelyncar Most Recent 1 year ago

Selected Answer: A

thena natively supports querying JSON data stored in S3 using standard SQL functions.
This eliminates the need for additional data transformation steps using Glue jobs (as required in Option C or D).

upvoted 1 times

tgv 1 year, 1 month ago

As chris_spencer mentioned below, now Athena supports querying with PartiQL which technically makes the answer A correct.

upvoted 1 times

VerRi 1 year, 1 month ago

Selected Answer: A

B requires Redshift Spectrum, so A

upvoted 1 times

chris_spencer 1 year, 2 months ago

Selected Answer: C

Answer should be C.

Amazon Athena does not support querying with PartiQL until 16.04.2024, https://aws.amazon.com/about-aws/whats-new/2024/04/amazon-athena-federated-query-pass-through/

The DEA01 exam should not have include the latest feature

upvoted 2 times

Christina666 1 year, 2 months ago

Selected Answer: A

A. Unified Querying with Athena: Athena provides a SQL-like interface for querying various data sources, including JSON and CSV in S3, as well as traditional databases.
PartiQL Support: Athena's PartiQL extension allows querying semi-structured JSON data directly, eliminating the need for a separate query engine.
Serverless and Managed: Both AWS Glue and Athena are serverless, minimizing infrastructure management for the data engineers.
No Unnecessary Transformations: Avoiding transformations for JSON data simplifies the pipeline and reduces operational overhead.

B. Redshift Spectrum: While Spectrum can query external data, it's primarily intended for Redshift data warehouse extensions. It adds complexity for the RDS and DynamoDB data sources.

upvoted 4 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: B

I will go with B

upvoted 4 times

nyaopoko 1 year, 2 months ago

B is the best choice:

AWS Glue Data Catalog: AWS Glue can crawl and catalog the data sources (S3 buckets, RDS databases, DynamoDB tables) and store the metadata in the AWS Glue Data Catalog. This provides a centralized metadata repository for all data sources.

Amazon Redshift Spectrum: Redshift Spectrum is a feature of Amazon Redshift that allows you to query data directly from various data sources, including S3 buckets, without loading the data into Redshift tables. This means you can query the JSON and CSV files in S3, as well as the RDS and DynamoDB data sources, using standard SQL syntax.

SQL and PartiQL Support: Redshift Spectrum supports querying structured data sources (like RDS and CSV files) using SQL, and querying semi-structured data sources (like JSON files) using PartiQL, which is a SQL-compatible query language for JSON data.

upvoted 1 times

Luke97 1 year, 3 months ago

The answer should be B.

A is incorrect because Athena does NOT support PartiQL.

C is NOT the least operational (has the additional step to convert JSON to Parquet or csv)

D is incorrect because DynamoDB export data to S3 in DynamoDB JSON or Amzone Ion format only (https://aws.amazon.com/blogs/aws/new-export-amazon-dynamodb-table-data-to-data-lake-amazon-s3/).

upvoted 4 times

halogi 1 year, 3 months ago

Selected Answer: C

AWS Athena can only query in SQL, not PartiQL, so both A and B are incorrect. LakeFormation can not work directly with DynamoDB, so D is incorrect. The only acceptable answer is C

upvoted 2 times

andrevus 1 year, 2 months ago

similar to SQL, so A

upvoted 1 times

rralucard_ 1 year, 4 months ago

Selected Answer: A

Option A, using AWS Glue and Amazon Athena, would meet the requirements with the least operational overhead. This solution allows data scientists to directly query data in its original format without the need for additional data transformation steps, making it easier to implement and manage.

upvoted 3 times

Question #77 *Topic 1*

A data engineer is configuring Amazon SageMaker Studio to use AWS Glue interactive sessions to prepare data for machine learning (ML) models. The data engineer receives an access denied error when the data engineer tries to prepare the data by using SageMaker Studio.
Which change should the engineer make to gain access to SageMaker Studio?

A. Add the AWSGlueServiceRole managed policy to the data engineer's IAM user.

B. Add a policy to the data engineer's IAM user that includes the sts:AssumeRole action for the AWS Glue and SageMaker service principals in the trust policy.

C. Add the AmazonSageMakerFullAccess managed policy to the data engineer's IAM user.

D. Add a policy to the data engineer's IAM user that allows the sts:AddAssociation action for the AWS Glue and SageMaker service principals in the trust policy.

Suggested Answer: *B*

Community vote distribution

B (61%) C (39%)

⊟ **tgv** Highly Voted 👍 1 year, 1 month ago

Selected Answer: B

I don't believe you're supposed to assign a FullAccess policy, so I will go with B.

upvoted 6 times

⊟ **GiorgioGss** Highly Voted 👍 1 year, 3 months ago

Selected Answer: B

I will go with B since you can get access denied even with the AmazonSageMakerFullAccess.
See here: https://stackoverflow.com/questions/64709871/aws-sagemaker-studio-createdomain-access-error

upvoted 5 times

⊟ **mohamedTR** Most Recent ⊙ 8 months, 3 weeks ago

Selected Answer: B

B. the engineer needs to assume specific roles to allow interaction between these services. The sts:AssumeRole action is necessary for this purpose

upvoted 1 times

⊟ **junrun3** 10 months, 1 week ago

Selected Answer: C

B, this approach involves setting up the trust relationship for roles. It is not a typical requirement for resolving access issues with SageMaker Studio directly.

upvoted 2 times

⊟ **LR2023** 11 months, 2 weeks ago

OPtion A
https://docs.aws.amazon.com/glue/latest/dg/glue-is-security.html

upvoted 1 times

⊟ **LR2023** 11 months, 2 weeks ago

and You can attach AWSGlueServiceRole to your users, groups, and roles.

upvoted 2 times

⊟ **LR2023** 9 months ago

Sorry changed my mind option B makes most sense

upvoted 2 times

⊟ **Christina666** 1 year, 2 months ago

Selected Answer: C

SageMaker Permissions: The AmazonSageMakerFullAccess managed policy provides broad permissions for using Amazon SageMaker features, including SageMaker Studio and the ability to interact with other AWS services like AWS Glue.
Least Privilege: While this policy is quite permissive, it's the most direct solution to the immediate access issue. After resolving the error, you can refine permissions for a more granular approach.

upvoted 1 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: C

I will go with C

upvoted 3 times

> **nyaopoko** 1 year, 2 months ago
>
> Option A (AWSGlueServiceRole managed policy) is not relevant, as this policy is intended for the AWS Glue service itself, not for users accessing SageMaker Studio.
>
> Option B (adding a policy with sts:AssumeRole action) is not necessary, as SageMaker handles the role assumption process internally.
>
> Option D (sts:AddAssociation action) is not a valid action and is not required for accessing SageMaker Studio or using AWS Glue interactive sessions.
>
> upvoted 2 times

fceb2c1 1 year, 3 months ago

https://repost.aws/knowledge-center/sagemaker-featuregroup-troubleshooting

upvoted 1 times

damaldon 1 year, 3 months ago

Ans. C

https://docs.aws.amazon.com/aws-managed-policy/latest/reference/AmazonSageMakerFullAccess.html

upvoted 2 times

atu1789 1 year, 4 months ago

Selected Answer: B

B. Add a policy to the data engineer's IAM user that includes the sts:AssumeRole action for the AWS Glue and SageMaker service principals in the trust policy.

• This is the most appropriate solution. The sts:AssumeRole action allows the data engineer's IAM user to assume a role that has the necessary permissions for both AWS Glue and SageMaker. This is a common approach for granting cross-service access in AWS.

upvoted 2 times

rralucard_ 1 year, 4 months ago

Selected Answer: C

Amazon SageMaker requires permissions to perform actions on your behalf. By attaching the AmazonSageMakerFullAccess managed policy to the data engineer's IAM user, you grant the necessary permissions for SageMaker Studio to access AWS Glue and other related services.

upvoted 3 times

Question #78 *Topic 1*

A company extracts approximately 1 TB of data every day from data sources such as SAP HANA, Microsoft SQL Server, MongoDB, Apache Kafka, and Amazon DynamoDB. Some of the data sources have undefined data schemas or data schemas that change.

A data engineer must implement a solution that can detect the schema for these data sources. The solution must extract, transform, and load the data to an Amazon S3 bucket. The company has a service level agreement (SLA) to load the data into the S3 bucket within 15 minutes of data creation.

Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon EMR to detect the schema and to extract, transform, and load the data into the S3 bucket. Create a pipeline in Apache Spark.

B. Use AWS Glue to detect the schema and to extract, transform, and load the data into the S3 bucket. Create a pipeline in Apache Spark.

C. Create a PySpark program in AWS Lambda to extract, transform, and load the data into the S3 bucket.

D. Create a stored procedure in Amazon Redshift to detect the schema and to extract, transform, and load the data into a Redshift Spectrum table. Access the table from Amazon S3.

Suggested Answer: *B*

Community vote distribution

B (100%)

GiorgioGss Highly Voted 9 months, 2 weeks ago

Selected Answer: B

Least effort = B

upvoted 6 times

rralucard_ Highly Voted 11 months ago

Selected Answer: B

B. Use AWS Glue to detect the schema and to extract, transform, and load the data into the S3 bucket. Create a pipeline in Apache Spark.

upvoted 5 times

Christina666 Most Recent 8 months, 2 weeks ago

Selected Answer: B

Glue ETL

upvoted 1 times

kj07 9 months, 2 weeks ago

The option with the least operational overhead is B.

upvoted 3 times

Question #79 *Topic 1*

A company has multiple applications that use datasets that are stored in an Amazon S3 bucket. The company has an ecommerce application that generates a dataset that contains personally identifiable information (PII). The company has an internal analytics application that does not require access to the PII.

To comply with regulations, the company must not share PII unnecessarily. A data engineer needs to implement a solution that with redact PII dynamically, based on the needs of each application that accesses the dataset.

Which solution will meet the requirements with the LEAST operational overhead?

A. Create an S3 bucket policy to limit the access each application has. Create multiple copies of the dataset. Give each dataset copy the appropriate level of redaction for the needs of the application that accesses the copy.

B. Create an S3 Object Lambda endpoint. Use the S3 Object Lambda endpoint to read data from the S3 bucket. Implement redaction logic within an S3 Object Lambda function to dynamically redact PII based on the needs of each application that accesses the data.

C. Use AWS Glue to transform the data for each application. Create multiple copies of the dataset. Give each dataset copy the appropriate level of redaction for the needs of the application that accesses the copy.

D. Create an API Gateway endpoint that has custom authorizers. Use the API Gateway endpoint to read data from the S3 bucket. Initiate a REST API call to dynamically redact PII based on the needs of each application that accesses the data.

Suggested Answer: *B*

Community vote distribution

B (100%)

teo2157 10 months, 2 weeks ago

Selected Answer: B

It's B based on AWS documentation

https://docs.aws.amazon.com/AmazonS3/latest/userguide/transforming-objects.html

upvoted 2 times

pypelyncar 1 year ago

Selected Answer: B

S3 Object Lambda automatically triggers the Lambda function only when there's a request to access data in the S3 bucket. This eliminates the need for pre-processing or creating multiple data copies with varying levels of redaction (Options A and C).

upvoted 3 times

4c78df0 1 year, 1 month ago

Selected Answer: B

B is correct

upvoted 2 times

damaldon 1 year, 3 months ago

Ans. B

You can use an Amazon S3 Object Lambda Access Point to control access to documents with personally identifiable information (PII).

https://docs.aws.amazon.com/comprehend/latest/dg/using-access-points.html

upvoted 4 times

atu1789 1 year, 4 months ago

Selected Answer: B

S3 Object Lambda allows you to add custom processing, such as redaction of PII, to data retrieved from S3. This is done dynamically, meaning you don't need to store multiple copies of the data. It's a more efficient and operationally simpler approach compared to managing multiple dataset versions.

upvoted 1 times

rralucard_ 1 year, 4 months ago

Selected Answer: B

Amazon S3 Object Lambda allows you to add your own code to S3 GET requests to modify and process data as it is returned to an application. For example, you could use an S3 Object Lambda to dynamically redact personally identifiable information (PII) from data retrieved from S3. This would allow you to control access to sensitive information based on the needs of different applications, without having to create and manage multiple copies of your data.

upvoted 3 times

Question #80 *Topic 1*

A data engineer needs to build an extract, transform, and load (ETL) job. The ETL job will process daily incoming .csv files that users upload to an Amazon S3 bucket. The size of each S3 object is less than 100 MB.
Which solution will meet these requirements MOST cost-effectively?

A. Write a custom Python application. Host the application on an Amazon Elastic Kubernetes Service (Amazon EKS) cluster.

B. Write a PySpark ETL script. Host the script on an Amazon EMR cluster.

C. Write an AWS Glue PySpark job. Use Apache Spark to transform the data.

D. Write an AWS Glue Python shell job. Use pandas to transform the data.

Suggested Answer: *D*

Community vote distribution

D (57%) C (43%)

halogi Highly Voted 👍 1 year, 3 months ago

Selected Answer: C

AWS Glue Python Shell Job is billed $0.44 per DPU-Hour for each job
AWS Glue PySpark is billed $0.29 per DPU-Hour for each job with flexible execution and $0.44 per DPU-Hour for each job with standard execution
Source: https://aws.amazon.com/glue/pricing/

upvoted 10 times

GustonMari 11 months, 3 weeks ago

thats true for the 1 DPU, but thats not good because the minimum DPU for PySpark Job is 1 DPU. But for Python Job the minimum DPU is 0.0625. So the Python job is way more cheaper for small dataset and quick ETL transformation

upvoted 4 times

atu1789 Highly Voted 👍 1 year, 4 months ago

Selected Answer: D

Option D: Write an AWS Glue Python shell job and use pandas to transform the data, is the most cost-effective solution for the described scenario.

AWS Glue's Python shell jobs are a good fit for smaller-scale ETL tasks, especially when dealing with .csv files that are less than 100 MB each. The use of pandas, a powerful and efficient data manipulation library in Python, makes it an ideal tool for processing and transforming these types of files. This approach avoids the overhead and additional costs associated with more complex solutions like Amazon EKS or EMR, which are generally more suited for larger-scale, more complex data processing tasks.

Given the requirements – processing daily incoming small-sized .csv files – this solution provides the necessary functionality with minimal resources, aligning well with the goal of cost-effectiveness.

upvoted 9 times

YUICH Most Recent 🕘 5 months ago

Selected Answer: D

It is important not to compare just the "price per DPU hour," but to consider the total cost by factoring in overhead for job startup, minimum DPU count, execution time, and data volume. For a relatively lightweight workload—such as processing approximately 100 MB of CSV files on a daily basis—option (D), using an AWS Glue Python shell job, is the most cost-effective choice.

upvoted 2 times

LR2023 11 months, 2 weeks ago

Selected Answer: D

going with D https://docs.aws.amazon.com/whitepapers/latest/aws-glue-best-practices-build-performant-data-pipeline/additional-considerations.html

upvoted 3 times

pypelyncar 1 year ago

Selected Answer: D

good candidate to be (2 options) for real, either spark and py have similar approaches. I would go with Pandas, although... 50/50.. it could be Spark. I hope not to find this question in the exam

upvoted 7 times

VerRi 1 year, 1 month ago

Selected Answer: C

PySpark with Spark(Flexible Execution): $0.29/hr for 1 DPU

PySpark with Spark(Standard Execution): $0.44/hr for 1 DPU

Python Shell with Pandas: $0.44/hr for 1 DPU

upvoted 3 times

cloudata 1 year, 1 month ago

Selected Answer: D

Python Shell is cheaper and can handle small to medium tasks.

https://docs.aws.amazon.com/whitepapers/latest/aws-glue-best-practices-build-performant-data-pipeline/additional-considerations.html

upvoted 6 times

chakka90 1 year, 2 months ago

D.

Because the pyspark is still being the cheap you have to use minimum of 2 DPU. Which would increase the cost anyway so, i feel that d should be correct

upvoted 3 times

khchan123 1 year, 2 months ago

Selected Answer: D

D.

While AWS Glue PySpark jobs are scalable and suitable for large workloads, C may be overkill for processing small .csv files (less than 100 MB each). The overhead of using Apache Spark may not be cost-effective for this specific use case.

upvoted 4 times

Leo87656789 1 year, 2 months ago

Selected Answer: D

Option D:

Even though the Python Shell Job is more expensive on a DPU-Hour basis, you can select the option "1/16 DPU" in the Job details for a Python Shell Job, which is definetly cheaper than a Pyspark job.

upvoted 4 times

lucas_rfsb 1 year, 2 months ago

Selected Answer: C

AWS Glue Python Shell Job is billed $0.44 per DPU-Hour for each job

AWS Glue PySpark is billed $0.29 per DPU-Hour for each job with flexible execution and $0.44 per DPU-Hour for each job with standard execution

Source: https://aws.amazon.com/glue/pricing/

upvoted 6 times

[Removed] 1 year, 3 months ago

Selected Answer: D

https://medium.com/@navneetsamarth/reduce-aws-cost-using-glue-python-shell-jobs-70a955d4359f#:~:text=The%20cheapest%20Glue%20Spark%20ETL,1%2F16th%20of%20a%20DPU.&text=This%20can%20result%20in%20massive,just%20a%2

upvoted 5 times

GiorgioGss 1 year, 3 months ago

Selected Answer: D

D is more cheaper than C. Not so scalable but is cheaper...

upvoted 4 times

rralucard_ 1 year, 4 months ago

Selected Answer: C

AWS Glue is a fully managed ETL service, which means you don't need to manage infrastructure, and it automatically scales to handle your data processing needs. This reduces operational overhead and cost.

PySpark, as a part of AWS Glue, is a powerful and widely-used framework for distributed data processing, and it's well-suited for handling data transformations on a large scale.

upvoted 5 times

Question #81 *Topic 1*

A data engineer creates an AWS Glue Data Catalog table by using an AWS Glue crawler that is named Orders. The data engineer wants to add the following new partitions:

s3://transactions/orders/order_date=2023-01-01
s3://transactions/orders/order_date=2023-01-02

The data engineer must edit the metadata to include the new partitions in the table without scanning all the folders and files in the location of the table.

Which data definition language (DDL) statement should the data engineer use in Amazon Athena?

A. ALTER TABLE Orders ADD PARTITION(order_date=’2023-01-01’) LOCATION ‘s3://transactions/orders/order_date=2023-01-01’;
ALTER TABLE Orders ADD PARTITION(order_date=’2023-01-02’) LOCATION ‘s3://transactions/orders/order_date=2023-01-02’;

B. MSCK REPAIR TABLE Orders;

C. REPAIR TABLE Orders;

D. ALTER TABLE Orders MODIFY PARTITION(order_date=’2023-01-01’) LOCATION ‘s3://transactions/orders/2023-01-01’;
ALTER TABLE Orders MODIFY PARTITION(order_date=’2023-01-02’) LOCATION ‘s3://transactions/orders/2023-01-02’;

Suggested Answer: *A*

Community vote distribution

A (100%)

Ja13 Highly Voted 11 months, 3 weeks ago

Selected Answer: A

Why the Other Options Are Incorrect:
Option B: MSCK REPAIR TABLE Orders: This command is used to repair the partitions of a table by scanning all the files in the specified location. This is not efficient if you know the specific partitions you want to add, as it will scan the entire table location.
Option C: REPAIR TABLE Orders: This is not a valid Athena DDL command.
Option D: ALTER TABLE Orders MODIFY PARTITION: This command is used to modify the location of existing partitions, not to add new partitions. It would not work for adding new partitions.

upvoted 5 times

artworkad Most Recent 1 year ago

Selected Answer: A

A is correct as per https://docs.aws.amazon.com/athena/latest/ug/alter-table-add-partition.html

upvoted 4 times

artworkad 1 year ago

A is correct as per https://docs.aws.amazon.com/athena/latest/ug/alter-table-add-partition.html

upvoted 1 times

tgv 1 year ago

Selected Answer: A

A is correct because it uses the appropriate DDL statements to add the new partitions directly without scanning all folders and files, meeting the requirements stated in the question.
B is incorrect because while it would update the partitions, it would involve scanning all files and folders.
C is incorrect because REPAIR TABLE is not a valid command.
D is incorrect because it modifies partitions instead of adding new ones.

upvoted 4 times

Question #82 *Topic 1*

A company stores 10 to 15 TB of uncompressed .csv files in Amazon S3. The company is evaluating Amazon Athena as a one-time query engine.

The company wants to transform the data to optimize query runtime and storage costs.

Which file format and compression solution will meet these requirements for Athena queries?

A. .csv format compressed with zip

B. JSON format compressed with bzip2

C. Apache Parquet format compressed with Snappy

D. Apache Avro format compressed with LZO

Suggested Answer: *C*

Community vote distribution

C (100%)

tgv Highly Voted 👍 1 year ago

Selected Answer: C

Parquet provides efficient columnar storage, enabling Athena to read only the necessary data for queries, which reduces scan times and speeds up query performance.
Snappy compression offers a good balance between compression speed and efficiency, reducing storage costs without significantly impacting query times.

upvoted 6 times

artworkad Most Recent 🕒 1 year ago

Selected Answer: C

Parquet + Snappy

upvoted 3 times

Question #83 *Topic 1*

A company uses Apache Airflow to orchestrate the company's current on-premises data pipelines. The company runs SQL data quality check tasks as part of the pipelines. The company wants to migrate the pipelines to AWS and to use AWS managed services.

Which solution will meet these requirements with the LEAST amount of refactoring?

A. Setup AWS Outposts in the AWS Region that is nearest to the location where the company uses Airflow. Migrate the servers into Outposts hosted Amazon EC2 instances. Update the pipelines to interact with the Outposts hosted EC2 instances instead of the on-premises pipelines.

B. Create a custom Amazon Machine Image (AMI) that contains the Airflow application and the code that the company needs to migrate. Use the custom AMI to deploy Amazon EC2 instances. Update the network connections to interact with the newly deployed EC2 instances.

C. Migrate the existing Airflow orchestration configuration into Amazon Managed Workflows for Apache Airflow (Amazon MWAA). Create the data quality checks during the ingestion to validate the data quality by using SQL tasks in Airflow.

D. Convert the pipelines to AWS Step Functions workflows. Recreate the data quality checks in SQL as Python based AWS Lambda functions.

Suggested Answer: *C*

Community vote distribution

C (100%)

bakarys 12 months ago

Selected Answer: C

The solution that will meet these requirements with the least amount of refactoring is Option C: Migrate the existing Airflow orchestration configuration into Amazon Managed Workflows for Apache Airflow (Amazon MWAA). Create the data quality checks during the ingestion to validate the data quality by using SQL tasks in Airflow.

Amazon Managed Workflows for Apache Airflow (MWAA) is a fully managed service that makes it easy to run open-source versions of Apache Airflow on AWS. It allows you to build workflows to design and visualize pipelines, automate complex tasks, and monitor executions. Since the company is already using Apache Airflow for orchestration, migrating to Amazon MWAA would require minimal refactoring.

upvoted 3 times

HunkyBunky 1 year ago

Selected Answer: C

Amazon MWAA - becuase we already uses Apache Airflow

upvoted 3 times

tgv 1 year ago

Selected Answer: C

Amazon MWAA is a managed service for running Apache Airflow. It allows migrating existing Airflow configurations with minimal changes. Data quality checks can continue to be implemented as SQL tasks in Airflow, similar to the current setup.

upvoted 3 times

Question #84 *Topic 1*

A company uses Amazon EMR as an extract, transform, and load (ETL) pipeline to transform data that comes from multiple sources. A data engineer must orchestrate the pipeline to maximize performance.

Which AWS service will meet this requirement MOST cost effectively?

A. Amazon EventBridge

B. Amazon Managed Workflows for Apache Airflow (Amazon MWAA)

C. AWS Step Functions

D. AWS Glue Workflows

Suggested Answer: *C*

Community vote distribution

C (75%) 13% 13%

artworkad Highly Voted 1 year ago

Selected Answer: C

Glue Workflows is for Glue job orchestration. C is for orchestration with different AWS services.

upvoted 5 times

hcong Most Recent 10 months, 2 weeks ago

Selected Answer: B

Amazon Managed Workflows for Apache Airflow (Amazon MWAA) is the best service for orchestrating complex data pipelines, especially for workloads already using Amazon EMR. Airflow is a powerful workflow orchestration tool that can be integrated with various AWS services, including EMR, to provide flexible scheduling, task dependency management, and monitoring capabilities. Using a hosted Airflow service (MWAA) can reduce administrative overhead while maintaining a familiar workflow orchestration environment.

upvoted 1 times

chrispchrisp 11 months, 1 week ago

Selected Answer: C

B is not cost effective, D is only to orchestrate Glue Jobs and Crawlers within AWS Glue itself. Hence C is correct, Step functions is cost effective and can link together your different AWS services.

upvoted 4 times

andrologin 11 months, 2 weeks ago

Selected Answer: C

This is EMR not Glue workflows hence step functions
EventBridge is best for event driven architecture

upvoted 3 times

LR2023 11 months, 2 weeks ago

Selected Answer: B

https://aws.amazon.com/blogs/big-data/build-a-concurrent-data-orchestration-pipeline-using-amazon-emr-and-apache-livy/

upvoted 1 times

bakarys 12 months ago

Selected Answer: D

The most cost-effective AWS service for orchestrating an ETL pipeline that maximizes performance is D. AWS Glue Workflows.

AWS Glue is a fully managed ETL service that makes it easy to move data between your data stores. AWS Glue simplifies and automates the difficult and time-consuming tasks of data discovery, conversion mapping, and job scheduling. AWS Glue Workflows allows you to orchestrate complex ETL jobs involving multiple crawlers, jobs, and triggers.

While the other services mentioned (Amazon EventBridge, Amazon MWAA, and AWS Step Functions) can be used for workflow orchestration, they are not specifically designed for ETL workloads and may not be as cost-effective for this use case. AWS Glue is designed for ETL workloads, and its workflows feature is specifically designed for orchestrating ETL jobs, making it the most suitable and cost-effective choice.

upvoted 2 times

⊟ **HunkyBunky** 1 year ago

Selected Answer: C

C - becuase AWS Glue can be used only for glue based ETL jobs

upvoted 1 times

⊟ **tgv** 1 year ago

Selected Answer: C

While AWS Glue Workflows are excellent for orchestrating Glue-specific ETL tasks, AWS Step Functions is more suitable for orchestrating an Amazon EMR-based ETL pipeline due to its greater flexibility, broader integration capabilities, and effective cost management. Therefore, the correct choice remains [C]

upvoted 2 times

Question #85 *Topic 1*

An online retail company stores Application Load Balancer (ALB) access logs in an Amazon S3 bucket. The company wants to use Amazon Athena to query the logs to analyze traffic patterns.

A data engineer creates an unpartitioned table in Athena. As the amount of the data gradually increases, the response time for queries also increases. The data engineer wants to improve the query performance in Athena.

Which solution will meet these requirements with the LEAST operational effort?

A. Create an AWS Glue job that determines the schema of all ALB access logs and writes the partition metadata to AWS Glue Data Catalog.

B. Create an AWS Glue crawler that includes a classifier that determines the schema of all ALB access logs and writes the partition metadata to AWS Glue Data Catalog.

C. Create an AWS Lambda function to transform all ALB access logs. Save the results to Amazon S3 in Apache Parquet format. Partition the metadata. Use Athena to query the transformed data.

D. Use Apache Hive to create bucketed tables. Use an AWS Lambda function to transform all ALB access logs.

Suggested Answer: *B*

Community vote distribution

B (89%) 11%

PGGuy Highly Voted 1 year ago

Selected Answer: B

Creating an AWS Glue crawler (Option B) is the most straightforward and least operationally intensive approach to automatically determine the schema, partition the data, and keep the AWS Glue Data Catalog updated. This ensures Athena queries are optimized without requiring extensive manual management or additional processing steps.

upvoted 5 times

andrologin Most Recent 11 months, 2 weeks ago

Selected Answer: C

AWS Crawler with classifiers allow you to determine the schema pattern on files/data that can then be used to partition the data for Athena query optimization

upvoted 1 times

PGGuy 1 year ago

Creating an AWS Glue crawler (Option B) is the most straightforward and least operationally intensive approach to automatically determine the schema, partition the data, and keep the AWS Glue Data Catalog updated. This ensures Athena queries are optimized without requiring extensive manual management or additional processing steps.

upvoted 2 times

tgv 1 year ago

Selected Answer: B

An AWS Glue crawler can automatically determine the schema of the logs, infer partitions, and update the Glue Data Catalog. Crawlers can be scheduled to run at intervals, minimizing manual intervention.

upvoted 4 times

Question #86 *Topic 1*

A company has a business intelligence platform on AWS. The company uses an AWS Storage Gateway Amazon S3 File Gateway to transfer files from the company's on-premises environment to an Amazon S3 bucket.

A data engineer needs to setup a process that will automatically launch an AWS Glue workflow to run a series of AWS Glue jobs when each file transfer finishes successfully.

Which solution will meet these requirements with the LEAST operational overhead?

A. Determine when the file transfers usually finish based on previous successful file transfers. Set up an Amazon EventBridge scheduled event to initiate the AWS Glue jobs at that time of day.

B. Set up an Amazon EventBridge event that initiates the AWS Glue workflow after every successful S3 File Gateway file transfer event.

C. Set up an on-demand AWS Glue workflow so that the data engineer can start the AWS Glue workflow when each file transfer is complete.

D. Set up an AWS Lambda function that will invoke the AWS Glue Workflow. Set up an event for the creation of an S3 object as a trigger for the Lambda function.

Suggested Answer: *B*

Community vote distribution

B (78%) 11% 11%

⊟ **tgv** Highly Voted 1 year ago

Selected Answer: B

Using EventBridge directly to trigger the AWS Glue workflow upon S3 events is straightforward and leverages AWS's event-driven architecture, requiring minimal maintenance.

upvoted 5 times

⊟ **Tester_TKK** Most Recent 2 months, 1 week ago

Selected Answer: B

EventBridge is a "bridge" for almost all AWS services

upvoted 1 times

⊟ **andrologin** 11 months, 2 weeks ago

Selected Answer: C

Event driven architecture with S3 file creation can only be EventBridge

upvoted 1 times

⊟ **bakarys** 12 months ago

Selected Answer: B

Setting up an Amazon EventBridge event that initiates the AWS Glue workflow after every successful S3 File Gateway file transfer event would meet these requirements with the least operational overhead.

This solution is event-driven and does not require manual intervention or reliance on a schedule that might not align with the actual completion time of the file transfers. The AWS Glue workflow is triggered automatically when a new file is added to the S3 bucket, ensuring that the AWS Glue workflow starts processing the new data as soon as it's available.

upvoted 2 times

⊟ **bakarys** 12 months ago

Selected Answer: D

The solution that will meet these requirements with the least operational overhead is Option D.

Setting up an AWS Lambda function that will invoke the AWS Glue Workflow, and setting up an event for the creation of an S3 object as a trigger for the Lambda function, will ensure that the workflow is automatically initiated each time a file transfer is successfully completed. This approach requires minimal operational overhead as it automates the process and does not require manual intervention or scheduling based on estimated completion times.

Options A and C involve manual intervention or assumptions about transfer times, which could lead to inefficiencies or inaccuracies. Option B is not feasible because Amazon EventBridge does not directly support triggering events based on S3 File Gateway file transfer events. Therefore, Option D is the most suitable solution.

upvoted 2 times

PGGuy 1 year ago

Selected Answer: B

Setting up an Amazon EventBridge event (Option B) to initiate the AWS Glue workflow after every successful S3 File Gateway file transfer event is the most efficient solution. It provides real-time automation with minimal operational overhead, ensuring that the Glue workflow starts immediately after the file transfer is complete.

upvoted 1 times

Question #87 *Topic 1*

A retail company uses Amazon Aurora PostgreSQL to process and store live transactional data. The company uses an Amazon Redshift cluster for a data warehouse.

An extract, transform, and load (ETL) job runs every morning to update the Redshift cluster with new data from the PostgreSQL database. The company has grown rapidly and needs to cost optimize the Redshift cluster.

A data engineer needs to create a solution to archive historical data. The data engineer must be able to run analytics queries that effectively combine data from live transactional data in PostgreSQL, current data in Redshift, and archived historical data. The solution must keep only the most recent 15 months of data in Amazon Redshift to reduce costs.

Which combination of steps will meet these requirements? (Choose two.)

A. Configure the Amazon Redshift Federated Query feature to query live transactional data that is in the PostgreSQL database.

B. Configure Amazon Redshift Spectrum to query live transactional data that is in the PostgreSQL database.

C. Schedule a monthly job to copy data that is older than 15 months to Amazon S3 by using the UNLOAD command. Delete the old data from the Redshift cluster. Configure Amazon Redshift Spectrum to access historical data in Amazon S3.

D. Schedule a monthly job to copy data that is older than 15 months to Amazon S3 Glacier Flexible Retrieval by using the UNLOAD command. Delete the old data from the Redshift cluster. Configure Redshift Spectrum to access historical data from S3 Glacier Flexible Retrieval.

E. Create a materialized view in Amazon Redshift that combines live, current, and historical data from different sources.

Suggested Answer: *A*

Community vote distribution

A (100%)

lalitjhawar Highly Voted 1 year ago

Option A (A): Configuring Amazon Redshift Federated Query allows Redshift to directly query the live transactional data in the PostgreSQL database without needing to import it. This ensures that you can access the most recent live data efficiently.

Option C (C): Scheduling a monthly job to copy data older than 15 months to Amazon S3 and then using Amazon Redshift Spectrum to access this historical data provides a cost-effective way to manage storage. This ensures that only the most recent 15 months of data are kept in Amazon Redshift, reducing storage costs. The historical data is still accessible via Redshift Spectrum for analytics queries.

upvoted 7 times

XP_2600 Most Recent 2 weeks, 4 days ago

Selected Answer: A

Question requires two answers

A and C

upvoted 1 times

AWSMM 2 months, 1 week ago

Selected Answer: C

A&C

D is incorrect and here is why: Redshift Spectrum cannot directly query data stored in Amazon S3 Glacier Flexible Retrieval (or S3 Glacier Deep Archive).

Redshift Spectrum can only query data stored in Amazon S3 Standard, S3 Intelligent-Tiering, S3 One Zone-IA, or S3 Glacier Instant Retrieval. Glacier Flexible Retrieval (previously just "Glacier") and Deep Archive are meant for long-term archival, and objects stored in those tiers aren't immediately accessible.

When an object is in Glacier Flexible Retrieval:

It must first be restored, which can take minutes to hours depending on the retrieval tier (Expedited, Standard, or Bulk).

During that time, the object remains unavailable for queries.

upvoted 1 times

Palee 3 months, 2 weeks ago

Selected Answer: D

Option A and D.

Option C doesn't talk about archiving Historical data

upvoted 1 times

Vidhi212 6 months, 3 weeks ago

Selected Answer: A

The correct combination of steps is:

A. Configure the Amazon Redshift Federated Query feature to query live transactional data that is in the PostgreSQL database.

This feature allows Amazon Redshift to directly query live transactional data in the PostgreSQL database without moving the data, enabling seamless integration with the data warehouse.

C. Schedule a monthly job to copy data that is older than 15 months to Amazon S3 by using the UNLOAD command. Delete the old data from the Redshift cluster. Configure Amazon Redshift Spectrum to access historical data in Amazon S3.

This step archives older data to Amazon S3, which is more cost-effective than storing it in Redshift. Redshift Spectrum allows querying this archived data directly from S3, ensuring analytics queries can still access historical data.

upvoted 2 times

SambitParida 6 months, 3 weeks ago

Selected Answer: A

A & C. Redshift spectrum cant read from glacier

upvoted 1 times

rsmf 8 months, 1 week ago

Selected Answer: A

A & C is the best choice

upvoted 1 times

mohamedTR 8 months, 3 weeks ago

Selected Answer: A

A & C: allows exporting Redshift data to Amazon S3 and ability to frequent access

upvoted 1 times

HunkyBunky 1 year ago

Selected Answer: A

A / C is a best choice

upvoted 1 times

artworkad 1 year ago

Selected Answer: A

AC is correct. D is not correct, because Redshift Spectrum cannot read from S3 Glacier Flexible Retrieval.

upvoted 4 times

tgv 1 year ago

Selected Answer: A

Choice A ensures that live transactional data from PostgreSQL can be accessed directly within Redshift queries.

Choice C archives historical data in Amazon S3, reducing storage costs in Redshift while still making the data accessible via Redshift Spectrum.

(to Admin: I can't select multiple answers on the voting comment)

upvoted 4 times

GHill1982 1 year ago

Correct answer is A and C.

upvoted 2 times

Question #88 *Topic 1*

A manufacturing company has many IoT devices in facilities around the world. The company uses Amazon Kinesis Data Streams to collect data from the devices. The data includes device ID, capture date, measurement type, measurement value, and facility ID. The company uses facility ID as the partition key.

The company's operations team recently observed many WriteThroughputExceeded exceptions. The operations team found that some shards were heavily used but other shards were generally idle.

How should the company resolve the issues that the operations team observed?

A. Change the partition key from facility ID to a randomly generated key.

B. Increase the number of shards.

C. Archive the data on the producer's side.

D. Change the partition key from facility ID to capture date.

Suggested Answer: *A*

Community vote distribution

A (100%)

tgv Highly Voted 1 year ago

Selected Answer: A

The best solution to resolve the issue of uneven shard usage and WriteThroughputExceeded exceptions is to balance the load more evenly across the shards. This can be effectively achieved by changing the partition key to something that ensures a more uniform distribution of data across the shards.

upvoted 6 times

Tester_TKK Most Recent 2 months, 1 week ago

Selected Answer: A

https://aws.amazon.com/blogs/big-data/under-the-hood-scaling-your-kinesis-data-streams/

upvoted 1 times

bakarys 12 months ago

Selected Answer: A

The correct answer is **A. Change the partition key from facility ID to a randomly generated key.**

Amazon Kinesis Data Streams uses the partition key that you specify to segregate the data records in the stream into shards. If the company uses the facility ID as the partition key, and if some facilities produce more data than others, then the data will be unevenly distributed across the shards. This can lead to some shards being heavily used while others are idle, and can cause `WriteThroughputExceeded` exceptions.

By changing the partition key to a randomly generated key, the data records are more likely to be evenly distributed across all the shards, which can help to avoid the issue of some shards being heavily used and others being idle. This solution requires the least operational overhead and does not involve increasing costs (as in option B), archiving data (which might not be desirable or feasible, as in option C), or changing to a partition key that might also lead to uneven distribution (as in option D).

upvoted 2 times

didorins 12 months ago

Selected Answer: A

D is not good, because you're effectively making things worse by partitioning by date. My answer is A

upvoted 2 times

Question #89 *Topic 1*

A data engineer wants to improve the performance of SQL queries in Amazon Athena that run against a sales data table.

The data engineer wants to understand the execution plan of a specific SQL statement. The data engineer also wants to see the computational cost of each operation in a SQL query.

Which statement does the data engineer need to run to meet these requirements?

A. EXPLAIN SELECT * FROM sales;

B. EXPLAIN ANALYZE FROM sales;

C. EXPLAIN ANALYZE SELECT * FROM sales;

D. EXPLAIN FROM sales;

Suggested Answer: *C*

Community vote distribution

C (100%)

FunkyFresco Highly Voted 1 year ago

Selected Answer: C

use EXPLAIN ANALIZE

https://docs.aws.amazon.com/athena/latest/ug/athena-explain-statement.html

upvoted 5 times

HunkyBunky Most Recent 12 months ago

Selected Answer: C

explain analyze + select * from table

upvoted 1 times

tgv 1 year ago

Selected Answer: C

A - Only partially meets the requirements as it does not include computational costs.

B - Incorrect syntax and does not meet the requirements.

C - Fully meets the requirements by providing both the execution plan and the computational costs.

D - Incorrect syntax and does not meet the requirements.

upvoted 4 times

Question #90 *Topic 1*

A company plans to provision a log delivery stream within a VPC. The company configured the VPC flow logs to publish to Amazon CloudWatch Logs. The company needs to send the flow logs to Splunk in near real time for further analysis.

Which solution will meet these requirements with the LEAST operational overhead?

A. Configure an Amazon Kinesis Data Streams data stream to use Splunk as the destination. Create a CloudWatch Logs subscription filter to send log events to the data stream.

B. Create an Amazon Kinesis Data Firehose delivery stream to use Splunk as the destination. Create a CloudWatch Logs subscription filter to send log events to the delivery stream.

C. Create an Amazon Kinesis Data Firehose delivery stream to use Splunk as the destination. Create an AWS Lambda function to send the flow logs from CloudWatch Logs to the delivery stream.

D. Configure an Amazon Kinesis Data Streams data stream to use Splunk as the destination. Create an AWS Lambda function to send the flow logs from CloudWatch Logs to the data stream.

Suggested Answer: *B*

Community vote distribution

B (100%)

tgv Highly Voted 1 year ago

Selected Answer: B

Kinesis Data Firehose has built-in support for Splunk as a destination, making the integration straightforward. Using a CloudWatch Logs subscription filter directly to Firehose simplifies the data flow, eliminating the need for additional Lambda functions or custom integrations.

upvoted 6 times

bakarys Most Recent 12 months ago

Selected Answer: B

Creating an Amazon Kinesis Data Firehose delivery stream to use Splunk as the destination and creating a CloudWatch Logs subscription filter to send log events to the delivery stream would meet these requirements with the least operational overhead.

Amazon Kinesis Data Firehose is the easiest way to reliably load streaming data into data lakes, data stores, and analytics services. It can capture, transform, and deliver streaming data to Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, generic HTTP endpoints, and service providers like Splunk.

CloudWatch Logs subscription filters allow you to send real-time log events to Kinesis Data Firehose and are ideal for scenarios where you want to forward the logs to other services for further analysis.

Options A and D involve Kinesis Data Streams, which would require additional management and operational overhead. Option C involves creating a Lambda function, which also adds operational overhead. Therefore, option B is the best choice.

upvoted 4 times

Question #91 *Topic 1*

A company has a data lake on AWS. The data lake ingests sources of data from business units. The company uses Amazon Athena for queries. The storage layer is Amazon S3 with an AWS Glue Data Catalog as a metadata repository.

The company wants to make the data available to data scientists and business analysts. However, the company first needs to manage fine-grained, column-level data access for Athena based on the user roles and responsibilities.

Which solution will meet these requirements?

A. Set up AWS Lake Formation. Define security policy-based rules for the users and applications by IAM role in Lake Formation.

B. Define an IAM resource-based policy for AWS Glue tables. Attach the same policy to IAM user groups.

C. Define an IAM identity-based policy for AWS Glue tables. Attach the same policy to IAM roles. Associate the IAM roles with IAM groups that contain the users.

D. Create a resource share in AWS Resource Access Manager (AWS RAM) to grant access to IAM users.

Suggested Answer: *A*

Community vote distribution

A (100%)

Ja13 Highly Voted 11 months, 3 weeks ago

Selected Answer: A

Correct Solution:
A. Set up AWS Lake Formation. Define security policy-based rules for the users and applications by IAM role in Lake Formation.

Explanation:
AWS Lake Formation: This service simplifies and automates the process of securing and managing data lakes. It allows you to define fine-grained access control policies at the database, table, and column levels.
Security Policy-Based Rules: Lake Formation allows you to create policies that specify which users or roles have access to specific data, including column-level access controls. This makes it easier to manage access based on roles and responsibilities.

upvoted 6 times

HagarTheHorrible Most Recent 6 months, 1 week ago

Selected Answer: A

A lake formation for any fine-grained access

upvoted 1 times

HunkyBunky 1 year ago

Selected Answer: A

A - Lake formation

upvoted 1 times

tgv 1 year ago

Selected Answer: A

Lake Formation supports fine-grained access control, including column-level permissions.

upvoted 4 times

Question #92 *Topic 1*

A company has developed several AWS Glue extract, transform, and load (ETL) jobs to validate and transform data from Amazon S3. The ETL jobs load the data into Amazon RDS for MySQL in batches once every day. The ETL jobs use a DynamicFrame to read the S3 data.

The ETL jobs currently process all the data that is in the S3 bucket. However, the company wants the jobs to process only the daily incremental data.

Which solution will meet this requirement with the LEAST coding effort?

A. Create an ETL job that reads the S3 file status and logs the status in Amazon DynamoDB.

B. Enable job bookmarks for the ETL jobs to update the state after a run to keep track of previously processed data.

C. Enable job metrics for the ETL jobs to help keep track of processed objects in Amazon CloudWatch.

D. Configure the ETL jobs to delete processed objects from Amazon S3 after each run.

Suggested Answer: *B*

Community vote distribution

B (100%)

⊟ **tgv** Highly Voted 👍 1 year ago

Selected Answer: B

AWS Glue job bookmarks are designed to handle incremental data processing by automatically tracking the state.

upvoted 8 times

⊟ **andrologin** Most Recent ⊙ 11 months, 2 weeks ago

Selected Answer: B

AWS Glue Bookmarks can be used to pin where the data processing last stopped hence help with incremental processing.

upvoted 1 times

⊟ **HunkyBunky** 12 months ago

Selected Answer: B

B - bookmarks is a key

upvoted 1 times

⊟ **bakarys** 12 months ago

Selected Answer: B

The solution that will meet this requirement with the least coding effort is Option B: Enable job bookmarks for the ETL jobs to update the state after a run to keep track of previously processed data.

AWS Glue job bookmarks help ETL jobs to keep track of data that has already been processed during previous runs. By enabling job bookmarks, the ETL jobs can skip the processed data and only process the new, incremental data. This feature is designed specifically for this use case and requires minimal coding effort.

Options A, C, and D would require additional coding and operational effort. Option A would require creating a new ETL job and managing a DynamoDB table. Option C would involve setting up job metrics and CloudWatch, which doesn't directly address processing incremental data. Option D would involve deleting data from S3 after processing, which might not be desirable if the original data needs to be retained. Therefore, Option B is the most suitable solution.

upvoted 3 times

Question #93 *Topic 1*

An online retail company has an application that runs on Amazon EC2 instances that are in a VPC. The company wants to collect flow logs for the VPC and analyze network traffic.

Which solution will meet these requirements MOST cost-effectively?

A. Publish flow logs to Amazon CloudWatch Logs. Use Amazon Athena for analytics.

B. Publish flow logs to Amazon CloudWatch Logs. Use an Amazon OpenSearch Service cluster for analytics.

C. Publish flow logs to Amazon S3 in text format. Use Amazon Athena for analytics.

D. Publish flow logs to Amazon S3 in Apache Parquet format. Use Amazon Athena for analytics.

Suggested Answer: *D*

Community vote distribution

D (91%) 9%

tgv Highly Voted 1 year ago

Selected Answer: D

Flow Logs can be published to S3 in Parquet format: https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-s3.html#flow-logs-s3-path

upvoted 6 times

koki2847 11 months, 2 weeks ago

https://aws.amazon.com/about-aws/whats-new/2021/10/amazon-vpc-flow-logs-parquet-hive-prefixes-partitioned-files/

upvoted 1 times

PGGuy Highly Voted 1 year ago

Selected Answer: D

Publishing flow logs to Amazon S3 in Apache Parquet format and using Amazon Athena for analytics (D) is the most cost-effective solution. This approach minimizes storage costs due to the efficient compression of Parquet, and optimizes query performance and cost in Athena due to the reduced data size and optimized columnar storage.

upvoted 5 times

jyrajan69 Most Recent 11 months ago

The question says clearly most cost effective, so on comparison between C and D, has to be C

upvoted 2 times

Ell89 4 months ago

how would text be more cost effective than columnar?

upvoted 1 times

LR2023 11 months, 2 weeks ago

Selected Answer: B

Flow logs acn be published to S3 but then option D sas in Parquet format - it is not automatically converted into parquet....

https://aws.amazon.com/solutions/implementations/centralized-logging-with-opensearch/

upvoted 1 times

HunkyBunky 1 year ago

Selected Answer: D

Apache parquet and S3 = most cost-effective solution

upvoted 2 times

Question #94 *Topic 1*

A retail company stores transactions, store locations, and customer information tables in four reserved ra3.4xlarge Amazon Redshift cluster nodes. All three tables use even table distribution.

The company updates the store location table only once or twice every few years.

A data engineer notices that Redshift queues are slowing down because the whole store location table is constantly being broadcast to all four compute nodes for most queries. The data engineer wants to speed up the query performance by minimizing the broadcasting of the store location table.

Which solution will meet these requirements in the MOST cost-effective way?

A. Change the distribution style of the store location table from EVEN distribution to ALL distribution.

B. Change the distribution style of the store location table to KEY distribution based on the column that has the highest dimension.

C. Add a join column named store_id into the sort key for all the tables.

D. Upgrade the Redshift reserved node to a larger instance size in the same instance family.

Suggested Answer: *A*

Community vote distribution

A (100%)

⊟ **andrologin** 11 months, 2 weeks ago

Selected Answer: A

ALL distribution is optimal for slowly changing dimension tables and generally small in size to allow for optimal joins.

upvoted 2 times

⊟ **bakarys** 12 months ago

Selected Answer: A

The most cost-effective solution to speed up the query performance by minimizing the broadcasting of the store location table would be:

A. Change the distribution style of the store location table from EVEN distribution to ALL distribution.

In Amazon Redshift, the ALL distribution style replicates the entire table to all nodes in the cluster, which eliminates the need to redistribute the data when executing a query. This can significantly improve query performance. Given that the store location table is updated only once or twice every few years, the overhead of maintaining the replicated data would be minimal. This makes it a cost-effective solution for improving the query performance.

upvoted 2 times

⊟ **PGGuy** 1 year ago

Selected Answer: A

Changing the distribution style of the store location table to ALL distribution (A) is the most cost-effective solution. It directly addresses the issue of broadcasting by ensuring the entire table is available on each node, significantly improving join performance without incurring substantial additional costs.

upvoted 4 times

⊟ **tgv** 1 year ago

Selected Answer: A

Using ALL distribution means the table is replicated to all nodes, eliminating the need for broadcasting during queries. Since the store location table is updated infrequently, this will significantly speed up queries without incurring frequent update costs.

upvoted 2 times

Question #95 *Topic 1*

A company has a data warehouse that contains a table that is named Sales. The company stores the table in Amazon Redshift. The table includes a column that is named city_name. The company wants to query the table to find all rows that have a city_name that starts with "San" or "El".

Which SQL query will meet this requirement?

A. Select * from Sales where city_name ~ ‘$(San|El)*’;

B. Select * from Sales where city_name ~ ‘^(San|El)*’;

C. Select * from Sales where city_name ~’$(San&El)*’;

D. Select * from Sales where city_name ~ ‘^(San&El)*’;

Suggested Answer: *B*

Community vote distribution

B (100%)

chrispchrisp Highly Voted 11 months, 1 week ago

Selected Answer: B

Regex Patterns for everyone's reference

. : Matches any single character.
* : Matches zero or more of the preceding element.
+ : Matches one or more of the preceding element.
[abc] : Matches any of the enclosed characters.
[^abc] : Matches any character not enclosed.
^ : Matches the start of a string.
$: Matches the end of a string.
| : Logical OR operator.
(abc) : Matches 'abc' and remembers the match.

Answer is B

upvoted 7 times

andrologin Most Recent 11 months, 2 weeks ago

Selected Answer: B

Regex patterns:
^ - used to capture the start of the text/string
| - used as an OR operator

upvoted 1 times

bakarys 12 months ago

Selected Answer: B

B. Select * from Sales where city_name ~ ‘^(San|El)*’;

This query uses a regular expression pattern with the ~ operator. The caret ^ at the beginning of the pattern indicates that the match must start at the beginning of the string. (San|El) matches either “San” or “El”, and * means zero or more of the preceding element. So this query will return all rows where city_name starts with either “San” or “El”.

upvoted 1 times

HunkyBunky 1 year ago

Selected Answer: B

B - becuase of regexp

upvoted 1 times

JohnYang 1 year ago

Selected Answer: B

^ asserts the position at the start of the string.

(San|El) matches either "San" or "El".

upvoted 3 times

⊟ **tgv** 1 year ago

Selected Answer: B

~: This operator indicates the use of a regular expression.

^: This symbol signifies the start of the string.

(San|El): This pattern matches strings that start with either "San" or "El".

upvoted 2 times

Question #96 *Topic 1*

A company needs to send customer call data from its on-premises PostgreSQL database to AWS to generate near real-time insights. The solution must capture and load updates from operational data stores that run in the PostgreSQL database. The data changes continuously.

A data engineer configures an AWS Database Migration Service (AWS DMS) ongoing replication task. The task reads changes in near real time from the PostgreSQL source database transaction logs for each table. The task then sends the data to an Amazon Redshift cluster for processing.

The data engineer discovers latency issues during the change data capture (CDC) of the task. The data engineer thinks that the PostgreSQL source database is causing the high latency.

Which solution will confirm that the PostgreSQL database is the source of the high latency?

A. Use Amazon CloudWatch to monitor the DMS task. Examine the CDCIncomingChanges metric to identify delays in the CDC from the source database.

B. Verify that logical replication of the source database is configured in the postgresql.conf configuration file.

C. Enable Amazon CloudWatch Logs for the DMS endpoint of the source database. Check for error messages.

D. Use Amazon CloudWatch to monitor the DMS task. Examine the CDCLatencySource metric to identify delays in the CDC from the source database.

Suggested Answer: *D*

Community vote distribution

D (100%)

tgv Highly Voted 1 year ago

Selected Answer: D

CDCLatencySource Metric: This metric measures the latency between the source database and the DMS task. It shows how long it takes for changes to be read from the source database's transaction logs.

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Monitoring.html#CHAP_Monitoring.Metrics

upvoted 5 times

HunkyBunky Most Recent 12 months ago

Selected Answer: D

only D makes sense

upvoted 1 times

sdas1 1 year ago

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Troubleshooting_Latency.html

A high CDCLatencySource metric indicates that the process of capturing changes from the source is delayed.

Answer is D

upvoted 1 times