Question #1 *Topic 1*

A financial services company needs to aggregate daily stock trade data from the exchanges into a data store. The company requires that data be streamed directly into the data store, but also occasionally allows data to be modified using SQL. The solution should integrate complex, analytic queries running with minimal latency. The solution must provide a business intelligence dashboard that enables viewing of the top contributors to anomalies in stock prices.
Which solution meets the company's requirements?

A. Use Amazon Kinesis Data Firehose to stream data to Amazon S3. Use Amazon Athena as a data source for Amazon QuickSight to create a business intelligence dashboard.

B. Use Amazon Kinesis Data Streams to stream data to Amazon Redshift. Use Amazon Redshift as a data source for Amazon QuickSight to create a business intelligence dashboard.

C. Use Amazon Kinesis Data Firehose to stream data to Amazon Redshift. Use Amazon Redshift as a data source for Amazon QuickSight to create a business intelligence dashboard.

D. Use Amazon Kinesis Data Streams to stream data to Amazon S3. Use Amazon Athena as a data source for Amazon QuickSight to create a business intelligence dashboard.

Suggested Answer: *D*

Community vote distribution

CHRIS12722222 Highly Voted 3 years, 2 months ago
complex, analytic queries running with minimal latency = REDSHIFT
KDF load data into redshift
Answer = C
upvoted 17 times

[Removed] Highly Voted 9 months, 3 weeks ago
the keywords "complex, analytic queries running with minimal latency"
upvoted 9 times

[Removed] Most Recent 9 months, 1 week ago
Selected Answer: C
C. Use Amazon Kinesis Data Firehose to stream data to Amazon Redshift. Use Amazon Redshift as a data source for Amazon to create a business intelligence dashboard.
C is valid

upvoted 1 times

Camille1992 1 year, 3 months ago
Selected Answer: C
the keywords "complex, analytic queries running with minimal latency"
upvoted 2 times

chinmayj213 1 year, 4 months ago
"but also occasionally allows data to be modified using SQL". This is only possible in redshift and redshift does not support kinesis stream directly , so firehouse and then redshift will work as source for quick sight
upvoted 2 times

metkillas 1 year, 5 months ago
Answer is now B as you can alter data in the materialized view when ingesting from stream. "You can now connect to and access the data from the stream using SQL and simplify your data pipelines by creating materialized views directly on top of the stream. The materialized views can also include SQL transforms as part of your ELT (extract, load and transform) pipeline."
upvoted 1 times

Krunal39 1 year, 5 months ago
Can Kinesis Data Streams directly write to S3?
upvoted 1 times

GCPereira 1 year, 7 months ago

D is a best. KFH provides near-real time data, so it's latency is little higher than that of KDS. Athena is the best data visualizer for complex and raw data. Since the data are in s3, it already appears in athena. Redshift is a data warehouse and can't reciev raw data.

upvoted 1 times

solvewithdata 1 year, 7 months ago

Selected Answer: C

can't use kds to deliver to s3 so must use kdf. use athena or redshift

upvoted 3 times

[Removed] 1 year, 8 months ago

KDS cant connect datahouse directly.

This is reason why We have to use KDF.

upvoted 1 times

gofavad926 1 year, 8 months ago

Selected Answer: C

The key is "ntegrate complex, analytic queries running with minimal latency."

upvoted 2 times

OliverF 1 year, 9 months ago

Definitely C. While you can stream data from KDS into Redshift, you cannot modify the materialised view using SQL, you can only run selects on a materialised view. S3 on the other hand doesn't support data modification using Athena.

upvoted 1 times

cd93 1 year, 10 months ago

Selected Answer: B

C is a good answer, but it is slower compared to data stream (B). Kinesis data firehose has to buffer stream data (thus near-realtime only), and then copy to s3 bucket as a staging area, then issue COPY command to Insert into Redshift table. Whereas Kinesis data stream just transfer data directly into the data warehouse via materialized view(s).

Second, the question requires the stream be transform-able via SQL, Data firehose do support transformation of data yes, but only via Lambda blueprints; Data stream transfer into Materialized View, which is written in SQL to convert Json body of the stream data, you can add more SQL here to transform it as you need - so Data Stream satisfies the requirement - albeit with lots of limitations.

Link to back my claim up: https://docs.aws.amazon.com/redshift/latest/dg/materialized-view-streaming-ingestion.html

upvoted 5 times

nishapagare97 1 year, 10 months ago

corect answer is D is wrong ?

upvoted 2 times

NikkyDicky 1 year, 11 months ago

Selected Answer: C

it's a C

upvoted 1 times

papercome 2 years ago

Selected Answer: C

There is a new feature of Redshift, supporting read KDS directly into a materialized view, which support lowest latency and make B quite promising. But it might not meet the modify by SQL requirements. So, this make C be the best choice.

upvoted 2 times

ccpmad 1 year, 11 months ago

"You don't have to send data to an Amazon Kinesis Data Firehose delivery stream, because with streaming ingestion, data can be sent directly from Kinesis Data Streams to a materialized view in an Amazon Redshift database."

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

yes, I just think what you say. The problem in this question is de modifying by SQL, and that is not possible with KDS.

upvoted 1 times

pk349 2 years, 1 month ago

C: I passed the test

upvoted 1 times

Question #2 *Topic 1*

A financial company hosts a data lake in Amazon S3 and a data warehouse on an Amazon Redshift cluster. The company uses Amazon QuickSight to build dashboards and wants to secure access from its on-premises Active Directory to Amazon QuickSight.
How should the data be secured?

A. Use an Active Directory connector and single sign-on (SSO) in a corporate network environment.

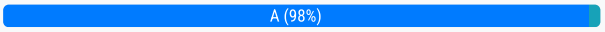
B. Use a VPC endpoint to connect to Amazon S3 from Amazon QuickSight and an IAM role to authenticate Amazon Redshift.

C. Establish a secure connection by creating an S3 endpoint to connect Amazon QuickSight and a VPC endpoint to connect to Amazon Redshift.

D. Place Amazon QuickSight and Amazon Redshift in the security group and use an Amazon S3 endpoint to connect Amazon QuickSight to Amazon S3.

Suggested Answer: *B*

Community vote distribution



JohnWick2020 Highly Voted 3 years, 8 months ago
Answer is A - "Use an AD connector and SSO in a corporate environment",
Key point of question is to “Secure access from its on-premise AD to Quicksight".
• Quicksight Enterprise edition allows for connecting to AD / using AD groups, SSO, row-level security, encryption at rest ... etc
upvoted 40 times

Paitan Highly Voted 3 years, 8 months ago
Option A. Quicksight Enterprise edition allows for connecting through AD connector.
upvoted 16 times

[Removed] Most Recent 9 months, 3 weeks ago
Option A. Quicksight Enterprise edition allows for connecting through AD connector.
upvoted 10 times

fagilom 1 year, 5 months ago
Considering the need to secure access from an on-premises Active Directory to Amazon QuickSight, Option A (Use an Active Directory Connector and Single Sign-On (SSO) in a Corporate Network Environment) is the most appropriate. It directly addresses the requirement of integrating QuickSight with the company’s Active Directory, allowing for controlled and secure access to QuickSight dashboards using corporate credentials. This approach ensures that access to QuickSight is managed in alignment with the company's existing security policies and user management practices. While it doesn't explicitly mention securing data, integrating QuickSight with SSO and Active Directory is a crucial step in overall access security, especially in a corporate environment.
upvoted 1 times

gofavad926 1 year, 8 months ago
Selected Answer: A
A for sure
upvoted 1 times

MLCL 1 year, 11 months ago
Selected Answer: A
Its a me A
upvoted 1 times

NikkyDicky 1 year, 11 months ago
Selected Answer: A
it's an A
upvoted 1 times

papercome 2 years ago
Selected Answer: D
Is this single selection? Both A and D seems correct to me.
upvoted 1 times

pk349 2 years, 1 month ago

A: I passed the test

upvoted 2 times

anjuvinayan 2 years, 2 months ago

S3 cannot be placed inside VPC. Redshift can be placed inside VPC. So here connecting Quicksight to S3 doesn't need anything else. Quicksight enterprise edition allows AD connection and answer is A

upvoted 3 times

AwsNewPeople 2 years, 3 months ago

A. Use an Active Directory connector and single sign-on (SSO) in a corporate network environment is the correct solution to secure access from an on-premises Active Directory to Amazon QuickSight. This solution allows the users to log in with their existing corporate credentials and enables IT administrators to manage access to Amazon QuickSight using their on-premises Active Directory. Amazon QuickSight supports various authentication methods, including SSO, OpenID Connect, and SAML 2.0, making it easy to integrate with existing authentication solutions.

upvoted 2 times

Vicious000 2 years, 4 months ago

Answer is A "Use an AD connector and SSO in a corporate environment"

upvoted 1 times

ran_gun 2 years, 5 months ago

The question target on How should the data be secured?

so in this case, it should be B right? option A talks only about the connecting via Active Directory connector; i doubt does it focus on the actual question

upvoted 3 times

cloudlearnerhere 2 years, 7 months ago

Correct answer is A as QuickSight access using Active Directory can be implemented using an Active Directory connector and single sign-on (SSO) in a corporate network environment.

Options B, C & D are wrong as they do not target the QuickSight authentication requirement but focus on QuickSight access to S3 and Redshift.

upvoted 2 times

Rejju 2 years, 8 months ago

The Answer seems to be A, but wondering why the answer mentioned as B. it worries me!

upvoted 3 times

Hruday 2 years, 10 months ago

Selected Answer: A

A is the answer

upvoted 2 times

Hruday 2 years, 10 months ago

A is the answer

upvoted 1 times

Question #3 *Topic 1*

A real estate company has a mission-critical application using Apache HBase in Amazon EMR. Amazon EMR is configured with a single master node. The company has over 5 TB of data stored on an Hadoop Distributed File System (HDFS). The company wants a cost-effective solution to make its HBase data highly available.
Which architectural pattern meets company's requirements?

A. Use Spot Instances for core and task nodes and a Reserved Instance for the EMR master node. Configure the EMR cluster with multiple master nodes. Schedule automated snapshots using Amazon EventBridge.

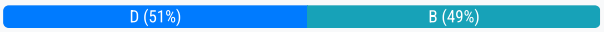
B. Store the data on an EMR File System (EMRFS) instead of HDFS. Enable EMRFS consistent view. Create an EMR HBase cluster with multiple master nodes. Point the HBase root directory to an Amazon S3 bucket.

C. Store the data on an EMR File System (EMRFS) instead of HDFS and enable EMRFS consistent view. Run two separate EMR clusters in two different Availability Zones. Point both clusters to the same HBase root directory in the same Amazon S3 bucket.

D. Store the data on an EMR File System (EMRFS) instead of HDFS and enable EMRFS consistent view. Create a primary EMR HBase cluster with multiple master nodes. Create a secondary EMR HBase read-replica cluster in a separate Availability Zone. Point both clusters to the same HBase root directory in the same Amazon S3 bucket.

Suggested Answer: *C*
Reference:
https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hbase-s3.html

Community vote distribution



cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: D

D is correct as Amazon EMR version 5.7.0 or later, you can set up a read-replica cluster, which allows you to maintain read-only copies of data in Amazon S3. In the event that the primary cluster becomes unavailable, you can access the data from the read-replica cluster to perform read operations simultaneously.

A is incorrect because using Spot EC2 instances for both of your core and task nodes could potentially cause downtime. Although this solution is the most cost-effective, it certainly doesn't provide the highest availability for Amazon EMR.

B is incorrect. While an EMR cluster with multiple master nodes can survive scenarios in which a primary master node fails, it is not, however, tolerant of Availability Zone failures.

C is wrong as It's not possible for two primary clusters to be linked to the same root directory at the same time. Take note that only one active cluster at a time can use the same HBase root directory in Amazon S3. The best way to implement this is to launch a primary EMR cluster and a secondary (read-replica) EMR cluster, since using two primary clusters is not supported.
upvoted 36 times

henom 2 years, 6 months ago
The answer is D.
Udemy course by Bonso has the same Logic.
upvoted 5 times

dushmantha Highly Voted 2 years, 12 months ago

Selected Answer: B

If we strictly want high availability then answer should be "D". But to be cost effective it only needs to go from current HDFS to S3, to make the data more available than before. Read replica is the next step if we want availability over master node crashes, etc. And it comes with additional cost. So I also suggest ans "B"
upvoted 14 times

[Removed] Most Recent 9 months, 3 weeks ago
Option D provides a robust and cost-effective solution that meets the company's requirements for making its HBase data highly available while leveraging Amazon EMR's capabilities effectively.
upvoted 10 times

NarenKA 1 year, 4 months ago

Selected Answer: D

Option D provides a robust and cost-effective solution that meets the company's requirements for making its HBase data highly available while leveraging Amazon EMR's capabilities effectively.

upvoted 3 times

kondi2309 1 year, 4 months ago

Selected Answer: D

the answer here is D.

upvoted 3 times

joselopezjm 1 year, 4 months ago

Selected Answer: D

D because it requires HA deploying two different AZ

upvoted 2 times

gofavad926 1 year, 8 months ago

Selected Answer: D

as cloudlearnerhere explains ""D is correct as Amazon EMR version 5.7.0 or later, you can set up a read-replica cluster, which allows you to maintain read-only copies of data in Amazon S3. In the event that the primary cluster becomes unavailable, you can access the data from the read-replica cluster to perform read operations simultaneously"

upvoted 3 times

NikkyDicky 1 year, 11 months ago

Selected Answer: B

B for cost

Hbase data availability is satisfied by EMRFS

upvoted 1 times

pk349 2 years, 1 month ago

D: I passed the test

upvoted 3 times

Shaggy_98 1 year, 6 months ago

Did you clear using this dumps ?

upvoted 1 times

anjuvinayan 2 years, 2 months ago

EMR is Single availability zone cluster which means we need to setup cluster in different avz for high availability. Two primary cluster is not an option. So answer is D

upvoted 2 times

kozer 2 years, 2 months ago

this recent aws documentation stateshttps://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-plan-consistent-view.html indicates consistent views are not supported and is not needed since 2020 . So yes D seems accurate or best answer but these questions are outdated and given how fast features change in AWS , this question certainly would be worded differently .

upvoted 2 times

bjmailbox 2 years, 2 months ago

Has to be option B , because it says HBASE data to be highly available which is already satisfied by EMRFS. It doesn't talk about cluster availability directly anywhere also considering the costs option D can be eliminated compared to B.

upvoted 1 times

rit25 2 years, 2 months ago

Just to tell you why not B.

Enabling EMRFS consistent view and pointing the HBase root directory to an Amazon S3 bucket are two different concepts, but they are related in this scenario.

EMRFS (EMR File System) is a file system interface that allows EMR clusters to access data stored in Amazon S3 in the same way as data stored on HDFS. By enabling EMRFS consistent view, EMR ensures that all nodes in the cluster see a consistent view of data stored in S3, which is important for applications like HBase that require strong consistency.

On the other hand, pointing the HBase root directory to an S3 bucket means that HBase tables and metadata are stored in S3, rather than on HDFS.

This allows HBase to take advantage of the durability and scalability of S3, while still providing low-latency access to data.

So, in option B, the company is using both EMRFS and S3. EMRFS is used to provide a consistent view of data stored in S3, while HBase is configured to store its tables and metadata in S3.

upvoted 1 times

AwsNewPeople 2 years, 3 months ago

D. Store the data on an EMR File System (EMRFS) instead of HDFS and enable EMRFS consistent view. Create a primary EMR HBase cluster with multiple master nodes. Create a secondary EMR HBase read-replica cluster in a separate Availability Zone. Point both clusters to the same HBase root directory in the same Amazon S3 bucket.

This solution provides a cost-effective way to make HBase data highly available by creating a primary EMR HBase cluster with multiple master nodes and a secondary EMR HBase read-replica cluster in a separate Availability Zone. By storing data on EMRFS and enabling EMRFS consistent view, both clusters can access the same data stored on an Amazon S3 bucket. This eliminates the need to store data redundantly and reduces costs. The use of multiple master nodes improves HBase availability and reliability. If the primary cluster fails, the secondary read-replica cluster can continue to serve read traffic.

upvoted 3 times

Matheus_Sampaio 2 years, 4 months ago

Selected Answer: D

D based on Bonso Udemy Course

upvoted 3 times

Vskvar 2 years, 6 months ago

Selected Answer: B

Data highly available.

D is not cost effective.

upvoted 1 times

Kako 2 years, 8 months ago

Selected Answer: D

upvoted 2 times

Question #4 *Topic 1*

A software company hosts an application on AWS, and new features are released weekly. As part of the application testing process, a solution must be developed that analyzes logs from each Amazon EC2 instance to ensure that the application is working as expected after each deployment. The collection and analysis solution should be highly available with the ability to display new information with minimal delays. Which method should the company use to collect and analyze the logs?

A. Enable detailed monitoring on Amazon EC2, use Amazon CloudWatch agent to store logs in Amazon S3, and use Amazon Athena for fast, interactive log analytics.

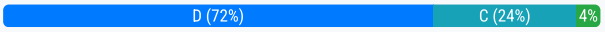
B. Use the Amazon Kinesis Producer Library (KPL) agent on Amazon EC2 to collect and send data to Kinesis Data Streams to further push the data to Amazon OpenSearch Service (Amazon Elasticsearch Service) and visualize using Amazon QuickSight.

C. Use the Amazon Kinesis Producer Library (KPL) agent on Amazon EC2 to collect and send data to Kinesis Data Firehose to further push the data to Amazon OpenSearch Service (Amazon Elasticsearch Service) and OpenSearch Dashboards (Kibana).

D. Use Amazon CloudWatch subscriptions to get access to a real-time feed of logs and have the logs delivered to Amazon Kinesis Data Streams to further push the data to Amazon OpenSearch Service (Amazon Elasticsearch Service) and OpenSearch Dashboards (Kibana).

Suggested Answer: *D*
Reference:
https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/Subscriptions.html

Community vote distribution



FHU Highly Voted 2 years, 9 months ago
I don't understand why everyone is choosing C. First of all, KPL does not send data to Kinesis Firehose, it sends data to Kinesis Data Streams, so C is very much incorrect. Second, the term KPL agent, there is no such thing. We would install Kinesis Agent on EC2 and not KPL Agent, so B and C are incorrect. In Option D, you see that it is using Cloudwatch logs which already offers what the customer wants... and implementing an Opensearch with Kibana for it would be overengineering and duplicating the same solution in another tool, duplicating data and cost. In option A it is using CloudWatch logs and Athena, which is easy to configure and works well. The answer does not say it, but the customer could use Cloudwatch Dashboards and Cloudwatch Metrics generated from the log stream. Activating EC2 detailed monitoring is not necessary though. Another thing: you guys are saying that Data Streams cannot deliver data to ES, but actually by using a Lambda you can pretty much do this... This question is very strange... But, by elimination, I would stay with option A.
upvoted 21 times

bp339 2 years, 6 months ago
KPL to Firehose is possible.
https://docs.aws.amazon.com/streams/latest/dev/kpl-with-firehose.html
upvoted 6 times

gopi_data_guy 2 years, 5 months ago
Direct KPL to Firehose is not possible. The above doc says
KPL --> Data streams --> Firehose.
Correct if I am wrong here.
upvoted 9 times

sattty 1 year, 11 months ago
Ability to display is missing in Athena
upvoted 1 times

rav009 2 years, 9 months ago
Agree, kinesis agent can write to firehose, KPL can't. So A
upvoted 3 times

JoellaLi 2 years, 8 months ago
why not D?
upvoted 2 times

rav009 2 years, 7 months ago
You are right, changed to D!

upvoted 3 times

ogerber 2 years ago

The problem with D is that kinesis data stream cannot write directly to opensearch. Only via lambda. Thats why it seams there is not correct answer... unless C is to be interpreted as kinesis agent.
https://opensearch.org/docs/1.1/opensearch/data-streams/
upvoted 2 times

JoellaLi 2 years, 8 months ago

But it 'should be highly available with the ability to display new information with minimal delays.' D is real time solution, while A is not.
upvoted 5 times

[Removed] Highly Voted 9 months, 3 weeks ago

D is correct. CloudWatch subscription is realtime. Push logs to Kinesis Firehose or Streams which can push into ElasticSearch for aggregation and dashboard.
CloudWatch-Subs -> Amazon Streams - > Firehose -> ElasticSearch -> Dashbaord
upvoted 9 times

michele_scar Most Recent 7 months, 1 week ago

Selected Answer: C

This is the only valid option. KDS doesn't have integration with Opensearch
upvoted 1 times

LeoSantos1212121212121212121 1 year, 4 months ago

C is correct, KDS cannot send data to OpenSearch as stated in answer D. Also answer D does not mention how the logs will get ingested from the EC2 instances to CloudWatch. logs.
upvoted 2 times

DigitalDanny 1 year, 6 months ago

Selected Answer: A

real-time capabilities are not a strict requirement and minimal delay is the primary consideration, then using Kinesis might indeed be considered overkill for your specific use case. In such scenarios, a simpler and cost-effective solution, such as Option A (CloudWatch + S3 + Athena), could be more suitable. This architecture allows for periodic log analysis without the need for real-time streaming.

Option A provides a straightforward setup with CloudWatch for log collection, S3 for storage, and Athena for fast and interactive log analytics. It is a serverless solution that can meet your requirements while minimizing complexity.

Consider your specific needs, the frequency of log analysis, and the trade-offs between simplicity, cost, and real-time capabilities when making your decision. If real-time insights are unnecessary, a less complex solution like Option A might be more appropriate.
upvoted 1 times

pn12345 1 year, 6 months ago

Selected Answer: C

correct answer
upvoted 2 times

roymunson 1 year, 7 months ago

Very weird one:
I'm also inbetween C & D but bot are making no sense because of:
C: KPL can't write into Firehose. The normal process flow would be KPL -> KDS -> KDF.
D: Is weird because they KDS can't write directly to OpenSearch. The normal process flow would be CloudWatch -> KDS -> Lambda -> OpenSearch. In addition to that, the question is talking about logs of the application itself and not metrics/logs of the EC2 - Instance (I'm not a native english speaker but this is how I've understood the question). I don't think that CloudWatch is the right tool for that.

In the end I go with C (hoping KPL agent is somewhat of a typo and they mean just Agent).
upvoted 2 times

TeamsDude 1 year, 8 months ago

KPL cannot send data directly to KDF.

"You can send data to your Kinesis Data Firehose Delivery stream using different types of sources: You can use a Kinesis data stream, the Kinesis Agent, or the Kinesis Data Firehose API using the AWS SDK. You can also use Amazon CloudWatch Logs, CloudWatch Events, or AWS IoT as your data source."

Source: https://docs.aws.amazon.com/firehose/latest/dev/basic-write.html#

So, C is incorrect. D would be 100% correct IF it had KDF instead of KDS. I am guessing it's a typo or who knows what !! I am gonna pretend this question won't show up in the exam :)

upvoted 1 times

gofavad926 1 year, 8 months ago

Selected Answer: D

D. I thought in a different option but:
A. "detailed monitoring on Amazon EC2" is for metrics, and report 1 minute period, so is NOT real time
BC. Do not exist KPL agent
D. Is the real time option

upvoted 1 times

Lala2020 1 year, 8 months ago

So what is the answer here?

upvoted 1 times

Hamza98 1 year, 8 months ago

Selected Answer: D

The correct answer is D, although in reality none is correct. I had this same question but instead of KDS it was Kinesis Data Firehose. Answer A B C were also among the answers and were all incorrect. Answer C will cause some delays while CloudWatch subscriptions are near-real time

upvoted 3 times

debasishg 1 year, 9 months ago

Selected Answer: C

C - i guess correct
D - says KDS --> OpenSearch does not work.

upvoted 3 times

[Removed] 1 year, 9 months ago

Selected Answer: D

https://docs.aws.amazon.com/opensearch-service/latest/developerguide/integrations.html#integrations-kinesis

upvoted 1 times

MLCL 1 year, 11 months ago

Selected Answer: C

Clearly C,
D can't be correct, Kinesis Data Streams cant send data directly to OpenSearch

upvoted 3 times

MLCL 1 year, 11 months ago

Apparently CloudWatch logs can send data to KDS, who knew https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/Subscriptions.html

upvoted 1 times

MLCL 1 year, 11 months ago

So answer is D

upvoted 1 times

MLCL 1 year, 11 months ago

Actually no, its C, since Kinesis Data Stream cannot send logs to OpenSearch haha

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: D

going w D

upvoted 1 times

developeranfc 1 year, 11 months ago

Selected Answer: D

D is the answer

upvoted 1 times

ccpmad 1 year, 11 months ago

Selected Answer: C

I think D is not correct, as Kinesis Data Streams can't push directly to OpenSearch Service: "You can still use other sources to load streaming data, such as Amazon Kinesis Data Firehose and Amazon CloudWatch Logs, which have built-in support for OpenSearch Service. Others, like Amazon S3, Amazon Kinesis Data Streams, and Amazon DynamoDB, use AWS Lambda functions as event handlers." You need Lambda to do that.
https://docs.aws.amazon.com/opensearch-service/latest/developerguide/integrations.html
So C is the correct for me.
Yes, Amazon Kinesis Producer Library (KPL) agent does not exist, but i think they are refering just to the agent
https://docs.aws.amazon.com/streams/latest/dev/writing-with-agents.html

upvoted 4 times

Question #5 *Topic 1*

A data analyst is using AWS Glue to organize, cleanse, validate, and format a 200 GB dataset. The data analyst triggered the job to run with the Standard worker type. After 3 hours, the AWS Glue job status is still RUNNING. Logs from the job run show no error codes. The data analyst wants to improve the job execution time without overprovisioning.
Which actions should the data analyst take?

A. Enable job bookmarks in AWS Glue to estimate the number of data processing units (DPUs). Based on the profiled metrics, increase the value of the executor- cores job parameter.

B. Enable job metrics in AWS Glue to estimate the number of data processing units (DPUs). Based on the profiled metrics, increase the value of the maximum capacity job parameter.

C. Enable job metrics in AWS Glue to estimate the number of data processing units (DPUs). Based on the profiled metrics, increase the value of the spark.yarn.executor.memoryOverhead job parameter.

D. Enable job bookmarks in AWS Glue to estimate the number of data processing units (DPUs). Based on the profiled metrics, increase the value of the num- executors job parameter.

Suggested Answer: *B*
Reference:
https://docs.aws.amazon.com/glue/latest/dg/monitor-debug-capacity.html

Community vote distribution
B (100%)

Donell Highly Voted 3 years, 7 months ago
Answer: B
B. Enable job metrics in AWS Glue to estimate the number of data processing units (DPUs). Based on the profiled metrics, increase the value of the maximum capacity job parameter.

Similar question is there in Jon Bonso's practice exam.
upvoted 15 times

cloudlearnerhere Highly Voted 2 years, 7 months ago
Correct answer is B as job metrics can be used to estimate the number of DPUs needed.

Options A & D are wrong as Job bookmarks help AWS Glue maintain state information and prevent the reprocessing of old data.

Options A & D are wrong as Job bookmarks help AWS Glue maintain state information and prevent the reprocessing of old data.
upvoted 6 times

gofavad926 Most Recent 1 year, 8 months ago
Selected Answer: B
B. Enable job metrics in AWS Glue to estimate the number of data processing units (DPUs). Based on the profiled metrics, increase the value of the maximum capacity job parameter.
upvoted 1 times

NikkyDicky 1 year, 11 months ago
Selected Answer: B
It's a B
upvoted 1 times

pk349 2 years, 1 month ago
B: I passed the test
upvoted 1 times

AwsNewPeople 2 years, 3 months ago
B. Enable job metrics in AWS Glue to estimate the number of data processing units (DPUs). Based on the profiled metrics, increase the value of the maximum capacity job parameter.

The data analyst should enable job metrics in AWS Glue to estimate the number of data processing units (DPUs) and profile the job to understand its resource requirements. Based on the profiled metrics, the data analyst should increase the value of the maximum capacity job parameter. This parameter controls the maximum number of DPUs that the job can use. By increasing the maximum capacity, the job can use more resources and complete faster without overprovisioning. Enabling job bookmarks can help with incremental processing but will not directly improve job execution time. Increasing the value of the executor-cores job parameter or the spark.yarn.executor.memoryOverhead job parameter may improve performance, but these parameters depend on the specific job requirements and are not directly related to the job's resource utilization. Similarly, increasing the num-executors job parameter will not directly improve job execution time.

upvoted 5 times

rocky48 2 years, 11 months ago

Selected Answer: B

Answer: B

upvoted 2 times

killohotel 3 years, 7 months ago

□답:B

북마크는 상태를 표시하는것이기때문에 A,D는 제외.

메트릭을 통해서 dpu적□ 개수를 추□하고, 최대용량을 늘리면 됩니다.

오버헤드 파라미터는 에러가 발생했을때 수□해야하지만, 문제는 3시간동안 러닝상태이기때문에 에러가 난것은 아니므로 C □외

https://docs.aws.amazon.com/ko_kr/glue/latest/dg/monitor-debug-capacity.html#monitor-debug-capacity-fix

upvoted 2 times

teo2157 1 year, 5 months ago

Clear like water

upvoted 1 times

Donell 3 years, 7 months ago

I suggest taking Jon Bonso's practice exams too.

upvoted 1 times

Huy 3 years, 7 months ago

This question is quite confused because for Glue version 2.0 jobs, you cannot instead specify a Maximum capacity. Instead, you should specify a Worker type and the Number of workers. However since A, D (no such parameters) and C (memoryOverhead not help in this case) are wrong, the best choice is B

upvoted 3 times

Shraddha 3 years, 8 months ago

Ans B

A and D = wrong, the name “bookmark” suggests persistency of a in-progress state, and so it is, used to track processed data, not for scaling. C = wrong, although you can set this parameter, the job metrics won’t help you, and this parameter won’t help long job running times because that was due to lack of computational power not memory.

upvoted 1 times

gunjan4392 3 years, 8 months ago

B is correct.

upvoted 1 times

Exia 3 years, 8 months ago

B.

A, D. Bookmark is not used for monitoring ETL job status.

upvoted 1 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 1 times

Draco31 3 years, 8 months ago

B.

B and C can make sense but C will be right only if the job returned the error spark.yarn.executor.memoryOverhead.

If no error, then the job is just taking too long so increase the max capacity

For AWS Glue version 2.0 jobs, you cannot instead specify a Maximum capacity. Instead, you should specify a Worker type and the Number of workers.

https://docs.aws.amazon.com/glue/latest/dg/add-job.html

upvoted 4 times

BillyC 3 years, 8 months ago

b is correct!

upvoted 1 times

Paitan 3 years, 8 months ago

Option B for sure. We can eliminate the two options with Bookmarks and spark.yarn.executor.memoryOverhead has nothing to do with Glue.

upvoted 1 times

Question #6 *Topic 1*

A company has a business unit uploading .csv files to an Amazon S3 bucket. The company's data platform team has set up an AWS Glue crawler to do discovery, and create tables and schemas. An AWS Glue job writes processed data from the created tables to an Amazon Redshift database. The AWS Glue job handles column mapping and creating the Amazon Redshift table appropriately. When the AWS Glue job is rerun for any reason in a day, duplicate records are introduced into the Amazon Redshift table.
Which solution will update the Redshift table without duplicates when jobs are rerun?

A. Modify the AWS Glue job to copy the rows into a staging table. Add SQL commands to replace the existing rows in the main table as postactions in the DynamicFrameWriter class.

B. Load the previously inserted data into a MySQL database in the AWS Glue job. Perform an upsert operation in MySQL, and copy the results to the Amazon Redshift table.

C. Use Apache Spark's DataFrame dropDuplicates() API to eliminate duplicates and then write the data to Amazon Redshift.

D. Use the AWS Glue ResolveChoice built-in transform to select the most recent value of the column.

Suggested Answer: *B*

Community vote distribution

A (100%)

testtaker3434 Highly Voted 3 years, 9 months ago
Answer should be A according to the link provided. Thoughts?
upvoted 19 times

lakediver 3 years, 6 months ago
Indeed A
https://aws.amazon.com/premiumsupport/knowledge-center/sql-commands-redshift-glue-job/
upvoted 6 times

Huy Highly Voted 3 years, 7 months ago
B is wrong. We don't need a staging DB here which is costly and moreover MySQL is not the right choice.
C. dropDuplicates() is used to remove duplicate records in the Spark not destination DB
D. ResolveChoice is to cast data with unidentified data type to a specified data type and also work on Spark not destination DB.

A is the answer
upvoted 15 times

NikkyDicky Most Recent 1 year, 11 months ago
Selected Answer: A
It's n A
upvoted 1 times

Espa 2 years, 1 month ago
Selected Answer: A
To me A looks correct answer, check this link

https://stackoverflow.com/questions/52397646/aws-glue-to-redshift-duplicate-data
upvoted 2 times

pk349 2 years, 1 month ago
A: I passed the test
upvoted 1 times

AwsNewPeople 2 years, 3 months ago
A. Modify the AWS Glue job to copy the rows into a staging table. Add SQL commands to replace the existing rows in the main table as postactions in the DynamicFrameWriter class.

To update the Redshift table without duplicates when AWS Glue jobs are rerun, the company should modify the AWS Glue job to copy the rows into a staging table. The job should then add SQL commands to replace the existing rows in the main table as postactions in the DynamicFrameWriter

class. This approach ensures that the data written to the Redshift table does not contain any duplicates, and the table only contains the latest data.

Loading the previously inserted data into a MySQL database and performing an upsert operation may be a feasible approach but adds complexity to the architecture. Using Spark's dropDuplicates() API to eliminate duplicates may not always work correctly when dealing with large datasets. Using the ResolveChoice built-in transform is used for handling schema changes in a column, not for removing duplicates.
upvoted 4 times

itsme1 2 years, 3 months ago

Selected Answer: A

With option B, it is to copy the RedShift data into SQL and back to RedShift.

Option A is simpler
upvoted 1 times

tpompeu 2 years, 4 months ago

Selected Answer: A

A, for sure
upvoted 1 times

henom 2 years, 7 months ago

Correct Answer - A

B is is incorrect because you can't use the COPY command to copy data directly from a MySQL database into Amazon Redshift. A workaround for this is to move the MySQL data into Amazon S3 and use AWS Glue as a staging table to perform the upsert operation. Since this method requires more effort, it is not the best approach to solve the problem.
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A as Redshift does not support merge or upsert on the single table. However, a staging table can be created and data merged with the main table.

Option B is wrong as a staging DB as MySQL is not required.

Option C is wrong as dropDuplicates() is used to remove duplicate records in the Spark and not destination DB.

Option D is wrong as ResolveChoice is to cast data with an unidentified data type to a specified data type. It does not handle duplicates.
upvoted 7 times

rocky48 2 years, 11 months ago

Selected Answer: A

Answer is A
upvoted 1 times

rocky48 2 years, 8 months ago

Got confused with C as dataframe.dropDuplicates() also will work, but as per the given question, we have to stick to AWS Glue job, thus Answer is A.
upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: A

Answer is A
upvoted 1 times

Shivanikats 3 years, 5 months ago

Answer is A
upvoted 1 times

Donell 3 years, 7 months ago

Answer: A. Modify the AWS Glue job to copy the rows into a staging table. Add SQL commands to replace the existing rows in the main table as postactions in the DynamicFrameWriter class.
upvoted 3 times

leliodesouza 3 years, 7 months ago

The answer is A.

upvoted 2 times

ariane_tateishi 3 years, 8 months ago

A should be the right answer. I found a link that helps to explain why. https://aws.amazon.com/pt/premiumsupport/knowledge-center/sql-commands-redshift-glue-job/

upvoted 1 times

lostsoul07 3 years, 8 months ago

A is the right answer

upvoted 4 times

Question #7 *Topic 1*

A streaming application is reading data from Amazon Kinesis Data Streams and immediately writing the data to an Amazon S3 bucket every 10 seconds. The application is reading data from hundreds of shards. The batch interval cannot be changed due to a separate requirement. The data is being accessed by Amazon
Athena. Users are seeing degradation in query performance as time progresses.
Which action can help improve query performance?

A. Merge the files in Amazon S3 to form larger files.

B. Increase the number of shards in Kinesis Data Streams.

C. Add more memory and CPU capacity to the streaming application.

D. Write the files to multiple S3 buckets.

Suggested Answer: *C*

Community vote distribution

A (100%)

abhineet Highly Voted 3 years, 9 months ago
It should be A, large number of small files ins3 will slow down reads
upvoted 41 times

testtaker3434 3 years, 9 months ago
Yeap, I agree its A.
upvoted 5 times

[Removed] 3 years, 6 months ago
You can speed up your queries dramatically by compressing your data, provided that files are splittable or of an optimal size (optimal S3 file size is between 200MB-1GB). Smaller data sizes mean less network traffic between Amazon S3 to Athena.
upvoted 2 times

Paitan Highly Voted 3 years, 8 months ago
Merge the files in Amazon S3 to form larger files will definitely increase read performance. So option A is the right choice.
upvoted 10 times

chinmayj213 Most Recent 1 year, 4 months ago
Everyone is saying A, which is write but why because 1000's shard and per shard capacity is 1 mb , So 1000's of files per second . which require merge to improve the query performance.
upvoted 1 times

NikkyDicky 1 year, 11 months ago
Selected Answer: A
A for sure
upvoted 1 times

pk349 2 years, 1 month ago
A: I passed the test
upvoted 2 times

Priya_angre 2 years, 1 month ago
what is right answer
upvoted 1 times

Aina 2 years, 3 months ago
A. This bit of AWS documentation: https://docs.aws.amazon.com/athena/latest/ug/performance-tuning-s3-throttling.html says "If possible, avoid having a large number of small files. Amazon S3 has a limit of 5500 requests per second, and your Athena queries share this same limit. If you scan millions of small objects in a single query, your query will likely be throttled by Amazon S3."
upvoted 1 times

AwsNewPeople 2 years, 3 months ago

A. Merge the files in Amazon S3 to form larger files.

To improve query performance when using Amazon Athena to access data from an Amazon S3 bucket, the streaming application should merge the files in S3 to form larger files. When the streaming application writes data to S3 every 10 seconds, it creates small files, which can lead to a large number of small files over time. This can lead to performance degradation in Athena queries as more small files mean more metadata needs to be scanned, and more file operations are required to read data. By merging small files into larger files, the number of files in the bucket can be reduced, which can significantly improve Athena query performance.

Increasing the number of shards in Kinesis Data Streams, adding more memory and CPU capacity to the streaming application, or writing files to multiple S3 buckets are not directly related to the issue of degraded query performance in Athena.
upvoted 4 times

itsme1 2 years, 3 months ago

Selected Answer: A

s3 has a limit of 5500 requests per second, combining reduces the requests

https://docs.aws.amazon.com/athena/latest/ug/performance-tuning.html
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A as merging files to form a bigger file can help optimize and improve query performance.
Option B is wrong as increasing shards would only increase the ingestion flow.

Options C & D are wrong as it does not improve Athena's query performance.
upvoted 7 times

MultiCloudIronMan 2 years, 8 months ago

Selected Answer: A

Merging small files into larger files will reduce the number of compute activities and speed up the process
upvoted 2 times

Abep 2 years, 9 months ago

https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/
upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: A

Answer should be A
upvoted 2 times

ru4aws 2 years, 11 months ago

Selected Answer: A

A
as merging small files into one large file will result in less meta data to maintain for the Data Catalog to maintain which results in Athena to scan data faster
upvoted 3 times

dushmantha 3 years ago

Things can be done to increase performance of Athena are use columnar formats, use small number of large files, use partitions. So the answer should be A.
upvoted 2 times

Bik000 3 years, 1 month ago

Selected Answer: A

Answer should be A
upvoted 1 times

moon2351 3 years, 3 months ago

Selected Answer: A

Answer is A
upvoted 3 times

RSSRAO 3 years, 4 months ago

Selected Answer: A

A is the correct answer. merge small files into larger files works as expected

upvoted 3 times

Question #8 *Topic 1*

A company uses Amazon OpenSearch Service (Amazon Elasticsearch Service) to store and analyze its website clickstream data. The company ingests 1 TB of data daily using Amazon Kinesis Data Firehose and stores one day's worth of data in an Amazon ES cluster.
The company has very slow query performance on the Amazon ES index and occasionally sees errors from Kinesis Data Firehose when attempting to write to the index. The Amazon ES cluster has 10 nodes running a single index and 3 dedicated master nodes. Each data node has 1.5 TB of Amazon EBS storage attached and the cluster is configured with 1,000 shards. Occasionally, JVMMemoryPressure errors are found in the cluster logs.
Which solution will improve the performance of Amazon ES?

A. Increase the memory of the Amazon ES master nodes.

B. Decrease the number of Amazon ES data nodes.

C. Decrease the number of Amazon ES shards for the index.

D. Increase the number of Amazon ES shards for the index.

Suggested Answer: *C*

Community vote distribution

C (100%)

Priyanka_01 Highly Voted 3 years, 8 months ago
IThink its C.
Refer the below link
https://aws.amazon.com/premiumsupport/knowledge-center/high-jvm-memory-pressure-elasticsearch/
upvoted 29 times

singh100 Highly Voted 3 years, 8 months ago
I agree with Option C:
Unbalanced shard allocations across nodes or too many shards in a cluster can cause JVMMemoryPressue.

Resolution - Reduce the number of shards by deleting old or unused indices.
https://aws.amazon.com/premiumsupport/knowledge-center/high-jvm-memory-pressure-elasticsearch/
upvoted 10 times

NikkyDicky Most Recent 1 year, 11 months ago
Selected Answer: C
its a C
upvoted 1 times

pk349 2 years, 1 month ago
C: I passed the test
upvoted 1 times

AwsNewPeople 2 years, 3 months ago
C. Decrease the number of Amazon ES shards for the index.

To improve the performance of Amazon ES in this scenario, the number of shards for the index should be decreased. Currently, the index has 1,000 shards, which is likely causing high overhead and slowing down query performance. In general, it's recommended to have 20-30 GB of data per shard for efficient indexing and query performance in Amazon ES. However, having too many shards can lead to inefficient resource utilization and slow query performance.

Additionally, since the cluster is configured with 3 dedicated master nodes, increasing the memory of the master nodes may not have a significant impact on performance. Decreasing the number of data nodes may also not be an effective solution, as this could reduce the capacity of the cluster to handle the 1 TB of daily data ingestion.

Increasing the number of shards for the index would further exacerbate the performance issues, as more shards would lead to more overhead and slower query performance.
upvoted 1 times

MaxwellBlackmore 2 years, 7 months ago

Selected Answer: C

According to this link
https://aws.amazon.com/premiumsupport/knowledge-center/opensearch-high-jvm-memory-pressure/
It clearly states that this issue can be caused due to "Unbalanced shard allocations across nodes or too many shards in a cluster."
upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: C

Correct answer is C as one of the causes of JVMMemoryPressure error can be Unbalanced shard allocations across nodes or too many shards in a cluster. and can be resolved by reducing the number of shards by deleting old or unused indices.

Option A is wrong because dedicated master nodes are only used to increase cluster stability. Therefore, this option won't help you improve the performance of the cluster.

Option B is incorrect because these nodes carry all the data in your indexes (storage) and do all the processing for your requests (CPU). If you decrease the number of data nodes, the performance of the cluster still won't improve.

Option D is incorrect. The JVMMemoryPressure error signifies that there is an unbalanced shard allocations across nodes. This means that there are too many shards in the Amazon ES cluster and not the other way around. To improve the performance of the cluster, you must decrease the number of shards.
upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: C

Selected Answer: C
upvoted 1 times

moon2351 3 years, 3 months ago

Selected Answer: C

Answer is C.
upvoted 2 times

Marvel_jarvis 3 years, 6 months ago

Ans - C
I got this question for my certification I gave on Dec 9th 2021.
upvoted 1 times

aws2019 3 years, 7 months ago

C is the right answer
upvoted 1 times

Donell 3 years, 7 months ago

Answer: C. Decrease the number of Amazon ES shards for the index.
Memory pressure in the JVM can result if:
You have unbalanced shard allocations across nodes
You have too many shards in a cluster

Fewer shards can yield better performance if JVMMemoryPressure errors
are encountered
Delete old or unused indices
upvoted 2 times

yogen 3 years, 7 months ago

C is correct. from documentation --
https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/sizing-domains.html
Here the shard size is 1.5*1000 (GB)/1000 (number of shards)= 1.5 GB which is much less than recommended size of shards.

The overarching goal of choosing a number of shards is to distribute an index evenly across all data nodes in the cluster. However, these shards shouldn't be too large or too numerous. A good rule of thumb is to try to keep shard size between 10–50 GiB. Large shards can make it difficult for Elasticsearch to recover from failure, but because each shard uses some amount of CPU and memory, having too many small shards can cause

performance issues and out of memory errors. In other words, shards should be small enough that the underlying Amazon ES instance can handle them, but not so small that they place needless strain on the hardware.

upvoted 6 times

lostsoul07 3 years, 7 months ago

C is the right answer

upvoted 3 times

Deep101 3 years, 8 months ago

The question says the domain is running one index, if so how can we assume there are old unused indices. shouldn't we reindex to adjust the number of shards?

upvoted 1 times

BillyC 3 years, 8 months ago

C is correct

upvoted 2 times

syu31svc 3 years, 8 months ago

From link https://aws.amazon.com/premiumsupport/knowledge-center/high-jvm-memory-pressure-elasticsearch/

You can resolve high JVM memory pressure issues by reducing traffic to the cluster. To reduce traffic to the cluster, follow these best practices:
Reduce the number of shards by deleting old or unused indices.

upvoted 3 times

Question #9 *Topic 1*

A manufacturing company has been collecting IoT sensor data from devices on its factory floor for a year and is storing the data in Amazon Redshift for daily analysis. A data analyst has determined that, at an expected ingestion rate of about 2 TB per day, the cluster will be undersized in less than 4 months. A long-term solution is needed. The data analyst has indicated that most queries only reference the most recent 13 months of data, yet there are also quarterly reports that need to query all the data generated from the past 7 years. The chief technology officer (CTO) is concerned about the costs, administrative effort, and performance of a long-term solution.
Which solution should the data analyst use to meet these requirements?

A. Create a daily job in AWS Glue to UNLOAD records older than 13 months to Amazon S3 and delete those records from Amazon Redshift. Create an external table in Amazon Redshift to point to the S3 location. Use Amazon Redshift Spectrum to join to data that is older than 13 months.

B. Take a snapshot of the Amazon Redshift cluster. Restore the cluster to a new cluster using dense storage nodes with additional storage capacity.

C. Execute a CREATE TABLE AS SELECT (CTAS) statement to move records that are older than 13 months to quarterly partitioned data in Amazon Redshift Spectrum backed by Amazon S3.

D. Unload all the tables in Amazon Redshift to an Amazon S3 bucket using S3 Intelligent-Tiering. Use AWS Glue to crawl the S3 bucket location to create external tables in an AWS Glue Data Catalog. Create an Amazon EMR cluster using Auto Scaling for any daily analytics needs, and use Amazon Athena for the quarterly reports, with both using the same AWS Glue Data Catalog.

Suggested Answer: *B*

Community vote distribution

A (100%)

Prodip Highly Voted 3 years, 9 months ago
Option A; We have implemented this to save cost .
upvoted 53 times

awssp12345 Highly Voted 3 years, 9 months ago
B is not correct because snapshotting will save costs but not solve problem of cluster being undersized
C is not correct because - CTAS is not used to move data to S3 via spectrum. CTAS Creates a new table based on a query. The owner of this table is the user that issues the command.
D is incorrect because EMR cannot be used as Data Warehouse solution And they do not need interactive query with Athena.
A is correct because that exactly specifies how to move data to Redshift spectrum and reduce cluster space:
https://docs.aws.amazon.com/redshift/latest/dg/c-getting-started-using-spectrum.html
upvoted 20 times

kondi2309 Most Recent 1 year, 4 months ago
Selected Answer: A
Def A, to save cost and less admin.
upvoted 1 times

NikkyDicky 1 year, 11 months ago
Selected Answer: A
its an A
upvoted 1 times

pk349 2 years, 1 month ago
A: I passed the test
upvoted 1 times

Aina 2 years, 3 months ago
A. The Udemy course by Stephane Maarek and Frank Kane has a really similar question in the practice exam.
upvoted 2 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: A

Correct answer is A as the AWS Glue job can be used to offload the data older than 13 months from Redshift to S3. 13 months data can be queried from Redshift, while 7 years data in S3 can be queried using Redshift Spectrum.

Option B is wrong as this would increase the cost further and would not scale far.

Option C is wrong as CTAS is not used to move data to S3 via the spectrum. CTAS creates a new table based on a query. The owner of this table is the user that issues the command.

Option D is wrong as EMR would increase the administrative effort as compared to Redshift.

upvoted 4 times

Rejju 2 years, 8 months ago

I am wondering why in the portal the correct ans is given as B. who validated and gives the right ans here?

upvoted 2 times

Abep 2 years, 9 months ago

Selected Answer: A

Answer is A

https://d1.awsstatic.com/whitepapers/amazon-redshift-cost-optimization.pdf

upvoted 3 times

rocky48 2 years, 11 months ago

Selected Answer: A

Answer-A

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: A

Answer should be A

upvoted 1 times

jrheen 3 years, 2 months ago

Answer-A

upvoted 1 times

jmensah60 3 years, 3 months ago

Selected Answer: A

A ticks all the boxes

upvoted 3 times

aws2019 3 years, 7 months ago

A is the right answer

upvoted 1 times

Shraddha 3 years, 7 months ago

B = wrong, this will not solve either cost or scale problem. C = wrong, to create table on S3 you use CREATE EXTERNAL TABLE not CTAS, also this does not remove older data. D = wrong, nonsense.

upvoted 1 times

leliodesouza 3 years, 8 months ago

The answer is A.

upvoted 2 times

AjithkumarSL 3 years, 8 months ago

When reading the Post : https://aws.amazon.com/blogs/big-data/amazon-redshift-dense-compute-dc2-nodes-deliver-twice-the-performance-as-dc1-at-the-same-price/, Option B Makes More sense.. any thoughts..

upvoted 1 times

asg76 3 years, 8 months ago

It's not cost effective..

upvoted 1 times

Question #10 *Topic 1*

An insurance company has raw data in JSON format that is sent without a predefined schedule through an Amazon Kinesis Data Firehose delivery stream to an
Amazon S3 bucket. An AWS Glue crawler is scheduled to run every 8 hours to update the schema in the data catalog of the tables stored in the S3 bucket. Data analysts analyze the data using Apache Spark SQL on Amazon EMR set up with AWS Glue Data Catalog as the metastore. Data analysts say that, occasionally, the data they receive is stale. A data engineer needs to provide access to the most up-to-date data.
Which solution meets these requirements?

A. Create an external schema based on the AWS Glue Data Catalog on the existing Amazon Redshift cluster to query new data in Amazon S3 with Amazon Redshift Spectrum.

B. Use Amazon CloudWatch Events with the rate (1 hour) expression to execute the AWS Glue crawler every hour.

C. Using the AWS CLI, modify the execution schedule of the AWS Glue crawler from 8 hours to 1 minute.

D. Run the AWS Glue crawler from an AWS Lambda function triggered by an S3:ObjectCreated:* event notification on the S3 bucket.

Suggested Answer: *A*

Community vote distribution

D (100%)

singh100 Highly Voted 3 years, 9 months ago

Answer: D
Data analysts analyze the data using Apache Spark SQL on Amazon EMR for the data stored on S3 in JSON format.
Input JSON file landing in S3 triggers a Lambda which invokes Glue Crawler.
upvoted 37 times

chinmayj213 1 year, 9 months ago

D is correct out of all these answer , but at the same time running a crawler on bucket for just landing of one object. Is it a good idea ?
upvoted 2 times

zanhsieh Highly Voted 3 years, 9 months ago

A is on demand (triggered by hand). B minimum time required is 1 hr. C is 1 minute based on the cron schedule syntax. For D, it could reach to sub-minute level since it watches S3 new data events.
https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html
https://docs.aws.amazon.com/glue/latest/dg/monitor-data-warehouse-schedule.html
https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/ScheduledEvents.html#RateExpressions
upvoted 9 times

NarenKA Most Recent 1 year, 4 months ago

Selected Answer: D

Option A is not directly related to the issue of schema updates and would not address the staleness of data in the AWS Glue Data Catalog. Option B increases the frequency of crawls but still may not provide real-time updates. Option C is not practical or cost-effective due to the excessive number of crawler runs it would trigger, and the AWS Glue crawler cannot be scheduled to run every minute. Option D provides a dynamic, event-driven solution that ensures data analysts have access to the most current data available.
upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: D

its a D
upvoted 1 times

Bdtri 2 years, 1 month ago

Why triggering glue crawler can give us the latest data? Isn't it only updating metastore?
upvoted 1 times

chinmayj213 1 year, 9 months ago

yes metastore , but every filename in s3 bucket need to registered in metastore to pick the latest data.
upvoted 2 times

pk349 2 years, 1 month ago

D: I passed the test
upvoted 1 times

kondi2309 1 year, 4 months ago
why D?
upvoted 1 times

lk23 2 years, 4 months ago
very curious to know every answer so far what it says it incorrect as discussion revleas something else, why?
upvoted 3 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: D
Correct answer is D as it is event-driven and would load the data as soon as the object-created event is triggered.

Option A is wrong as this is still manual and on-demand.

Option B is wrong as the refresh interval is still 1 hr.

Option C is wrong as the minimum precision for the schedule is 5 mins.
upvoted 3 times

Abep 2 years, 9 months ago
Selected Answer: D
Answer: D
upvoted 1 times

rocky48 2 years, 11 months ago
Selected Answer: D
Answer: D
upvoted 1 times

Bik000 3 years, 1 month ago
Selected Answer: D
Answer is D
upvoted 1 times

jrheen 3 years, 2 months ago
Answer-D
upvoted 1 times

ShilaP 3 years, 3 months ago
D is correct
upvoted 1 times

aws2019 3 years, 7 months ago
The answer is D.
upvoted 1 times

iconara 3 years, 7 months ago
D seems correct, but could potentially be an expensive solution.
upvoted 1 times

Huy 3 years, 7 months ago
Although D is correct answer, the answer should mention SQS. The crawler will not run fast enough to catch up with objects created.
upvoted 2 times

Shraddha 3 years, 7 months ago
This is a textbook question. A = wrong, won't work because schema will update every 8 hours. B = wrong, not most up-to-date. C = wrong, minimum schedule is 5 minutes.

https://aws.amazon.com/blogs/big-data/build-and-automate-a-serverless-data-lake-using-an-aws-glue-trigger-for-the-data-catalog-and-etl-jobs/
upvoted 2 times

Question #11 *Topic 1*

A company that produces network devices has millions of users. Data is collected from the devices on an hourly basis and stored in an Amazon S3 data lake.
The company runs analyses on the last 24 hours of data flow logs for abnormality detection and to troubleshoot and resolve user issues. The company also analyzes historical logs dating back 2 years to discover patterns and look for improvement opportunities.
The data flow logs contain many metrics, such as date, timestamp, source IP, and target IP. There are about 10 billion events every day.
How should this data be stored for optimal performance?

A. In Apache ORC partitioned by date and sorted by source IP

B. In compressed .csv partitioned by date and sorted by source IP

C. In Apache Parquet partitioned by source IP and sorted by date

D. In compressed nested JSON partitioned by source IP and sorted by date

Suggested Answer: *D*

Community vote distribution

A (100%)

zanhsieh Highly Voted 3 years, 9 months ago

A.
BD dropped due to row based format.
Choosing between ORC and Parquet format would be tough since their performance is very close. However, data are supposed to partitioned by date then sorted by source IP, so C dropped.
upvoted 60 times

abhineet 3 years, 9 months ago

correct
upvoted 2 times

Paitan Highly Voted 3 years, 9 months ago

ORC and Parquet are ideal here. But the data should be partitioned by Date and sorted on IP and not the other way round. So option A is the right choice.
upvoted 12 times

kondi2309 Most Recent 1 year, 4 months ago

Selected Answer: A

ideal choices will be ORC and Parquet, first choice being Apache Parquet, but here we have to consider partition, and partition by Date then sort on IP is best way to store data.
upvoted 2 times

MLCL 1 year, 11 months ago

Selected Answer: A

Between A and C, if we have more IPs than Dates.
I would go with A since analysis is performed on a daily schedule and anomalies are detected on a time interval.
upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: A

its an A
upvoted 1 times

pk349 2 years, 1 month ago

A: I passed the test
upvoted 2 times

anonymous909 2 years, 3 months ago

Option A: In Apache ORC partitioned by date and sorted by source IP

Partitioning the data by date allows for faster query performance and efficient data retrieval based on time periods.

Sorting the data by source IP enables efficient filtering and joins on that attribute.
Overall, ORC partitioned by date and sorted by source IP would provide efficient storage and querying of the data.
upvoted 1 times

srirnag 2 years, 4 months ago
Option C is the best. The analysis is done on last 24 hours of data. Hence, sorting by IP may not be ideal.
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: A
A is the right answer as the company does daily analysis, so it only needs to look at the data generated for a given date

C is wrong as partitioning by source IP is incorrect for this use case, and partitioning by date is optimal.

B & D, Both the above options are not columnar storage formats, they are row-based formats that are not optimal for big data retrievals for complex analytical queries.
upvoted 3 times

rocky48 2 years, 11 months ago
Selected Answer: A
A is the right answer
upvoted 1 times

dushmantha 2 years, 12 months ago
Selected Answer: A
Agree with "zanhsieh"
upvoted 1 times

Ayaa4 3 years ago
Columnar data is faster such as ORC and Parquet, answer is A
upvoted 1 times

Bik000 3 years, 1 month ago
Selected Answer: A
Answer is A
upvoted 2 times

jrheen 3 years, 2 months ago
Answer : A
upvoted 1 times

rav009 3 years, 5 months ago
For previous 24 hours, sorted by date from C is not helpful. Sorted by timestamp makes sense.
upvoted 2 times

Donell 3 years, 7 months ago
Answer is A. In Apache ORC partitioned by date and sorted by source IP.
Because the company analyzes historical logs dating back 2 years and also past 24 hours data.
Hence the Data should be partitioned based on Date and sorted by IP and not the other way around.
ORC is columnar hence preferred data format.
upvoted 5 times

Shraddha 3 years, 7 months ago
B and D = wrong, use columnar format. C = wrong, partition by date so historical data and be separated.
upvoted 2 times

Question #12 *Topic 1*

A banking company is currently using an Amazon Redshift cluster with dense storage (DS) nodes to store sensitive data. An audit found that the cluster is unencrypted. Compliance requirements state that a database with sensitive data must be encrypted through a hardware security module (HSM) with automated key rotation.
Which combination of steps is required to achieve compliance? (Choose two.)

A. Set up a trusted connection with HSM using a client and server certificate with automatic key rotation.

B. Modify the cluster with an HSM encryption option and automatic key rotation.

C. Create a new HSM-encrypted Amazon Redshift cluster and migrate the data to the new cluster.

D. Enable HSM with key rotation through the AWS CLI.

E. Enable Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) encryption in the HSM.

Suggested Answer: *BD*
Reference:
https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html

Community vote distribution
AC (100%)

testtaker3434 Highly Voted 3 years, 9 months ago
Answer should be A and C. Using HSM you have to create a new cluster (that eliminates B). See link below, it clearly states "You can't enable hardware security module (HSM) encryption by modifying the cluster. Instead, create a new, HSM-encrypted cluster and migrate your data to the new cluster"

https://docs.aws.amazon.com/redshift/latest/mgmt/changing-cluster-encryption.html

In the same link it says you have create certificates.

My thinking that its not D, its because it can be already configured when you are settinp up the cluster. (option C)
upvoted 48 times

GeeBeeEl 3 years, 8 months ago
I dont agree with you on c..... that site you referenced says "When you modify your cluster to enable KMS encryption, Amazon Redshift automatically migrates your data to a new encrypted cluster. " also see https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html
upvoted 2 times

GeeBeeEl 3 years, 8 months ago
I see now why C is correct --- "To migrate an unencrypted cluster to a cluster encrypted using a hardware security module (HSM), you create a new encrypted cluster and move your data to the new cluster. So I agree C is correct
upvoted 3 times

Nicki1013 Highly Voted 3 years, 9 months ago
Answer: A, C

When you use an HSM, you must use client and server certificates to configure a trusted connection between Amazon Redshift and your HSM.

Reference link:
https://docs.amazonaws.cn/en_us/redshift/latest/mgmt/security-key-management.html

To migrate an unencrypted cluster to a cluster encrypted using a hardware security module (HSM), you create a new encrypted cluster and move your data to the new cluster.

Reference link:
https://docs.aws.amazon.com/redshift/latest/mgmt/changing-cluster-encryption.html
upvoted 15 times

tsangckl Most Recent 1 year, 3 months ago

Bing is answering C and D. By this explanation

Option A suggests setting up a trusted connection with HSM using a client and server certificate with automatic key rotation. While this is a valid method for some systems, it's not directly applicable to Amazon Redshift. Redshift doesn't support this method for enabling encryption.
Option C is correct because Amazon Redshift doesn't allow you to modify an existing cluster to use HSM encryption. You would need to create a new HSM-encrypted Redshift cluster and migrate the data to it.
Option D is also correct. Once the new HSM-encrypted Redshift cluster is set up, you can enable HSM with key rotation through the AWS CLI.

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: AC

It's AC

upvoted 1 times

pk349 2 years, 1 month ago

AC: I passed the test

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: AC

Correct answer is A & C as Redshift does not allow encrypting existing cluster using HSM and there needs to be trust connection established between Redshift and HSM.

Options B & D are wrong as You can enable encryption when you launch your cluster, or you can modify an unencrypted cluster to use AWS Key Management Service (AWS KMS) encryption.

Option E is wrong as it is not valid.

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: AC

Answer-A,C

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: AC

Answer is A & C

upvoted 1 times

jrheen 3 years, 2 months ago

Answer-A,C

upvoted 1 times

aws2019 3 years, 7 months ago

A and C

upvoted 1 times

Huy 3 years, 7 months ago

A, C is correct but why Redshift with HSM is asked in 2020? Redshift only works with HSM Classic and new customer can't create HSM classic anymore.

upvoted 3 times

Donell 3 years, 7 months ago

Answer: A,C (Similar question is there in Jon Bonso's practice exam).

upvoted 1 times

Shraddha 3 years, 7 months ago

B = wrong, to use HSM you have to create new clusters. D = wrong, key rotation is not done by HSM, but Redshift. E = wrong, nonsense. This is a textbook question.

https://docs.aws.amazon.com/redshift/latest/mgmt/changing-cluster-encryption.html#migrating-to-an-encrypted-cluster

upvoted 1 times

leliodesouza 3 years, 8 months ago

the answers are A and C.

upvoted 1 times

jyrajan69 3 years, 8 months ago

Definitely A and C. First answer is from the link provided by testtaker3434, and 2nd answer from the following link https://docs.aws.amazon.com/redshift/latest/mgmt/security-key-management.html

upvoted 3 times

lostsoul07 3 years, 8 months ago

A, C is the right answer

upvoted 1 times

BillyC 3 years, 8 months ago

A and C are correct

upvoted 1 times

Question #13 *Topic 1*

A company is planning to do a proof of concept for a machine learning (ML) project using Amazon SageMaker with a subset of existing on-premises data hosted in the company's 3 TB data warehouse. For part of the project, AWS Direct Connect is established and tested. To prepare the data for ML, data analysts are performing data curation. The data analysts want to perform multiple step, including mapping, dropping null fields, resolving choice, and splitting fields. The company needs the fastest solution to curate the data for this project.
Which solution meets these requirements?

A. Ingest data into Amazon S3 using AWS DataSync and use Apache Spark scrips to curate the data in an Amazon EMR cluster. Store the curated data in Amazon S3 for ML processing.

B. Create custom ETL jobs on-premises to curate the data. Use AWS DMS to ingest data into Amazon S3 for ML processing.

C. Ingest data into Amazon S3 using AWS DMS. Use AWS Glue to perform data curation and store the data in Amazon S3 for ML processing.

D. Take a full backup of the data store and ship the backup files using AWS Snowball. Upload Snowball data into Amazon S3 and schedule data curation jobs using AWS Batch to prepare the data for ML.

Suggested Answer: *C*

Community vote distribution

abhineet Highly Voted 3 years, 9 months ago
C is correct, s3 is a valid target for DMS https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Target.S3.html
upvoted 28 times

GauravM17 3 years, 9 months ago
I guess it should be A. DMS can can not do the data preprocessing and Spark is the best option on the large datasets
upvoted 2 times

Brijeshkrishna 3 years, 7 months ago
C is correct as AWS Glue uses Spark engine
upvoted 2 times

zeronine Highly Voted 3 years, 9 months ago
C. DMS supports S3 as a target.
upvoted 6 times

Frazy Most Recent 1 year, 7 months ago
C: Option A, using AWS DataSync and Apache Spark scripts, involves maintaining an on-premises EMR cluster, which adds complexity and management overhead. Option B, creating custom ETL jobs on-premises, requires significant development effort and may not be as efficient as using AWS Glue. Option D, using AWS Snowball for data transfer and AWS Batch for data curation, is less efficient and more time-consuming compared to the direct ingestion and curation approach.
upvoted 1 times

jerkane 1 year, 7 months ago
Selected Answer: C
C is correct using glue would be faster than using EMR
upvoted 1 times

monkeydba 1 year, 7 months ago
This is the differentiator. DMS can read a database source. DataSync cannot. The question says "hosted in the company's 3 TB data warehouse.". DataSync can read NFS, SMB, HDFS, S3.
https://docs.aws.amazon.com/datasync/latest/userguide/how-datasync-transfer-works.html#onprem-aws
upvoted 2 times

monkeydba 1 year, 7 months ago
DataSync can indeed pull a subset of data. https://docs.aws.amazon.com/datasync/latest/userguide/filtering.html
upvoted 1 times

monkeydba 1 year, 7 months ago
The question mentions "subset" of data. Can DataSync do that? DMS can.
upvoted 1 times

gofavad926 1 year, 8 months ago

Selected Answer: A

A. I don't understand that all people agree on C. DMS means database migration service and here they mention data warehouse and not database, so this is not a DMS compatible source: https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Source.html.

A is the valid option because with DataSync you can migrate your DATA to the S3 and then we can process it with EMR (more efficient than Glue)

upvoted 1 times

debasishg 1 year, 9 months ago

Selected Answer: C

C.

Because, 1. Datasync is used for file migration, DMS for Data. 2. GLUE ETL required to transform data after migration.

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: C

C for sure

upvoted 1 times

pk349 2 years, 1 month ago

C: I passed the test

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Correct answer is C as DMS can be used for data migration to S3. AWS Glue can be used for preprocessing and data curation.

Option A is wrong as DataSync is usually for storage migration and using Spark might be as operationally efficient as Glue.

Option B is wrong as using on-premises custom ETL jobs might not be time-efficient.

Option D is wrong as the data migration using Snowball will take time.

upvoted 4 times

Arka_01 2 years, 9 months ago

Selected Answer: C

Glue is the answer, as all the mentioned data operations are readily available with Glue.

upvoted 1 times

rocky48 2 years, 10 months ago

Selected Answer: C

C is correct

upvoted 1 times

Thiya 3 years, 7 months ago

C is the correct answer, use DMS to ingest the data from on-prem data warehouse to S3 and use Glue DataBrew to data curation.

upvoted 1 times

Donell 3 years, 7 months ago

Answer C. Ingest data into Amazon S3 using AWS DMS. Use AWS Glue to perform data curation and store the data in Amazon 3 for ML processing.

upvoted 1 times

Shraddha 3 years, 7 months ago

A = wrong, DataSync is for storage migration not data warehouse. B = wrong, ETL job on-premise is not fast. D = wrong, too slow.

upvoted 3 times

Question #14 *Topic 1*

A US-based sneaker retail company launched its global website. All the transaction data is stored in Amazon RDS and curated historic transaction data is stored in Amazon Redshift in the us-east-1 Region. The business intelligence (BI) team wants to enhance the user experience by providing a dashboard for sneaker trends.
The BI team decides to use Amazon QuickSight to render the website dashboards. During development, a team in Japan provisioned Amazon QuickSight in ap- northeast-1. The team is having difficulty connecting Amazon QuickSight from ap-northeast-1 to Amazon Redshift in us-east-1.
Which solution will solve this issue and meet the requirements?

A. In the Amazon Redshift console, choose to configure cross-Region snapshots and set the destination Region as ap-northeast-1. Restore the Amazon Redshift Cluster from the snapshot and connect to Amazon QuickSight launched in ap-northeast-1.

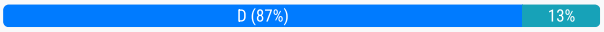
B. Create a VPC endpoint from the Amazon QuickSight VPC to the Amazon Redshift VPC so Amazon QuickSight can access data from Amazon Redshift.

C. Create an Amazon Redshift endpoint connection string with Region information in the string and use this connection string in Amazon QuickSight to connect to Amazon Redshift.

D. Create a new security group for Amazon Redshift in us-east-1 with an inbound rule authorizing access from the appropriate IP address range for the Amazon QuickSight servers in ap-northeast-1.

Suggested Answer: *B*

Community vote distribution



Priyanka_01 Highly Voted 3 years, 9 months ago
D any thoughts?
https://docs.aws.amazon.com/quicksight/latest/user/enabling-access-redshift.html
Not B: https://docs.aws.amazon.com/quicksight/latest/user/working-with-aws-vpc.html
upvoted 44 times

awssp12345 3 years, 9 months ago
Agreed
upvoted 1 times

Monika14Sharma 3 years, 8 months ago
D is the right answer!
upvoted 4 times

lakediver 3 years, 6 months ago
Agree
Remember cross region ingestion is also supported (https://aws.amazon.com/about-aws/whats-new/2014/06/29/amazon-redshift-announces-cross-region-ingestion-and-improved-query-functionality/)
As of 23 Nov 21, there's a preview capability to have cross region data sharing as well https://aws.amazon.com/about-aws/whats-new/2014/06/29/amazon-redshift-announces-cross-region-ingestion-and-improved-query-functionality/
upvoted 3 times

lakediver 3 years, 6 months ago
Changing my answer to B
https://aws.amazon.com/blogs/big-data/amazon-quicksight-deployment-models-for-cross-account-and-cross-region-access-to-amazon-redshift-and-amazon-rds/
upvoted 2 times

CHRIS12722222 3 years, 3 months ago
"Set up an Amazon Redshift-managed VPC endpoint between the Amazon Redshift cluster VPC and QuickSight VPC. For instructions, see Connecting to Amazon Redshift using an interface VPC endpoint."
https://aws.amazon.com/blogs/big-data/amazon-quicksight-deployment-models-for-cross-account-and-cross-region-access-to-amazon-redshift-and-amazon-rds/

Option B says something very different

upvoted 1 times

CHRIS12722222 3 years, 3 months ago

Quicksight has no vpc. What is mentioned in the post is connection from quicksight to a vpc using vpcid subnet id and security grp id.
Answer=D

upvoted 2 times

singh100 Highly Voted 3 years, 9 months ago

D. As mentioned in the link shared by Priyanka, B is not the answer. "QuickSight connects only to data located in the same AWS Region where you're currently using QuickSight. You can't connect QuickSight to data in another AWS Region, even if your VPC is configured to work across AWS Regions."

upvoted 11 times

teo2157 Most Recent 1 year, 6 months ago

Selected Answer: B

Hi guys, there's a new amazing IA feature in the AWS documentation called Amazon Q, if you type this question there, the answer is or "Create an Amazon Redshift cluster in the ap-northeast-1 region and copy the required data from us-east-1 Redshift" or "they can configure cross-region connectivity between QuickSight and Redshift. This involves setting up a VPC endpoint for Redshift in ap-northeast-1 region that connects to the us-east-1 Redshift cluster.", so chosing the option B as it's cheaper than the A.

upvoted 1 times

debasishg 1 year, 9 months ago

Selected Answer: D

D.
NOT A - "cross-Region snapshots .." - expensive
NOT B - "VPC endpoint.." - cross region connectivity not possible from Quicksight
NOT C - "Redshift endpoint connection string .." - needs address to be added through Security Group to allow connetivity

upvoted 1 times

nroopa 1 year, 10 months ago

Option D

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: D

Going with D

upvoted 1 times

Espa 2 years, 1 month ago

Selected Answer: D

D is the correct option

upvoted 1 times

pk349 2 years, 1 month ago

D: I passed the test

upvoted 1 times

Aina 2 years, 3 months ago

Definitely D. This scenario is explored in Stephane Maarek's and Abhishek Singh's Udemy practice exam set.

upvoted 1 times

rich_knp 2 years, 3 months ago

Selected Answer: D

D is correct

upvoted 2 times

flanfranco 2 years, 3 months ago

D: https://docs.aws.amazon.com/quicksight/latest/user/enabling-access-redshift.html
"If you activated Amazon QuickSight in multiple AWS Regions, you can create inbound rules for each Amazon QuickSight endpoint CIDR. Doing this allows Amazon QuickSight to have access to the Amazon RDS DB instance from any AWS Region defined in the inbound rules.

An Amazon QuickSight user or administrator who uses Amazon QuickSight in multiple AWS Regions is treated as a single user. In other words, even if you are using Amazon QuickSight in every AWS Region, both your Amazon QuickSight account and your users are global"

upvoted 3 times

[Removed] 2 years, 4 months ago

Selected Answer: D

Answer is D!

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: D

Correct answer is D as the Redshift security group should be updated to allow inbound access from QuickSight IP addresses.

Option B is wrong as QuickSight does not support cross-region access within the VPC.

Option A is wrong as it leads to data duplication and is not the most efficient solution.

Option C is wrong as changing the endpoint would not allow connectivity. The access needs to be enabled using security groups.

upvoted 2 times

Bansel 2 years, 8 months ago

D per AWS DA Course on Udemy!

upvoted 1 times

thirukudil 2 years, 8 months ago

Ans is D

https://docs.aws.amazon.com/quicksight/latest/user/enabling-access-redshift.html

upvoted 1 times

pgf909 2 years, 8 months ago

B is not correct as this: https://docs.aws.amazon.com/quicksight/latest/user/vpc-finding-setup-information.html

Make sure also that you use Amazon QuickSight in the same AWS Region with the VPC.

You can't use QuickSight in one AWS Region and expect to connect to a VPC in a different AWS Region.

upvoted 1 times

LukeTran3206 2 years, 8 months ago

Selected Answer: B

https://docs.aws.amazon.com/quicksight/latest/user/vpc-creating-a-connection-in-quicksight.html

AWS Region – The AWS Region where you plan to create a connection to your data source.

VPC ID – The ID of the VPC that contains the data, the subnets, and the security groups that you plan to use.

Subnet ID – The ID of the subnet that the QuickSight network interface is using.

Security group ID – The ID of the security group.

Quicksight can access using VPC ID and region so B is correct

upvoted 1 times

Question #15 *Topic 1*

An airline has .csv-formatted data stored in Amazon S3 with an AWS Glue Data Catalog. Data analysts want to join this data with call center data stored in
Amazon Redshift as part of a dally batch process. The Amazon Redshift cluster is already under a heavy load. The solution must be managed, serverless, well- functioning, and minimize the load on the existing Amazon Redshift cluster. The solution should also require minimal effort and development activity.
Which solution meets these requirements?

A. Unload the call center data from Amazon Redshift to Amazon S3 using an AWS Lambda function. Perform the join with AWS Glue ETL scripts.

B. Export the call center data from Amazon Redshift using a Python shell in AWS Glue. Perform the join with AWS Glue ETL scripts.

C. Create an external table using Amazon Redshift Spectrum for the call center data and perform the join with Amazon Redshift.

D. Export the call center data from Amazon Redshift to Amazon EMR using Apache Sqoop. Perform the join with Apache Hive.

Suggested Answer: *C*

Community vote distribution

abhineet Highly Voted 3 years, 9 months ago
I would go for C, Spectrum is serverless as well. Ques also asks for minimum development effort. For option A, you need to develop Lambda and Glue code.
upvoted 39 times

Jh2501 3 years, 9 months ago
Agree C. However, one thing I am still confused about - how can Spectrum create external table for the call centre data whereas it doesn't get stored on S3?
upvoted 7 times

Phoenyx89 3 years, 9 months ago
with a Create External Table as Select... I suppose
upvoted 1 times

DerekKey 3 years, 8 months ago
Wrong "Create External Table as Select" is Redshift command not Spectrum. Spectrum is used to query external as Read Only.
upvoted 1 times

GeeBeeEl 3 years, 9 months ago
If you are confused, check https://docs.aws.amazon.com/redshift/latest/dg/c-spectrum-external-tables.html
upvoted 3 times

Manue 3 years, 8 months ago
I think Jh2501 is not challenging Spectrum capability to create external tables, but how/why to create an external table on data which is stored in Redshift and not in S3. So, if there is not a typo in the question, C is a doubtful answer, and A could be the right one.
upvoted 5 times

JBAWA 2 years, 8 months ago
To define an external table in Amazon Redshift, use the CREATE EXTERNAL TABLE command
upvoted 1 times

Merrick 2 years, 5 months ago
https://aws.amazon.com/ko/premiumsupport/knowledge-center/redshift-spectrum-external-table/
upvoted 1 times

jove Highly Voted 3 years, 8 months ago
C doesn't make sense. Call center data is already stored in Redshift. What would be the purpose of creating an external table for the call center data? Also C suggests to perform the join with Redshift which is already under a heavy load.
upvoted 19 times

mendelthegreat 3 years, 8 months ago

1. Redshift Spectrum is a compute layer that sits between S3 and Redshift, so it will not add more load to Redshift

2. What the case is saying is that because Redshift is already under heavy load, we shouldn't load the .CSV data from S3 into redshift, so an external table would be better in Redshift Spectrum

3. The best use case for Redshift Spectrum, as described in the question, is to JOIN data in Redshift with another external data source, which in this case is S3, without having the need to bring everything into Redshift.

C is the undeniable correct answer here
upvoted 28 times

Kam006 Most Recent 1 year, 3 months ago
C is the correct answer. I do agree that table creation is part of RS and external table created to access the NON RS data sources (e.g. S3). External tables allow you to query data in S3 using the same SELECT syntax as with other Amazon Redshift tables. Here, the question says RS is already overloaded, hence we should not load the data in to RS. RS spectrum will join the S3 data along with RS data which is also serverless and minimal development efforts
upvoted 1 times

jerkane 1 year, 7 months ago
Selected Answer: C
C is correct as it is the one with minimal effort as no data is moved.
upvoted 2 times

markstudy 1 year, 7 months ago
Selected Answer: B
I would pick B:

A: Lambda is limited to 15 minutes of execution time, might not be enough to unload.
C: The call center data is already in Redshift, the missing data is the airport data.

Best possible option seems to be B: Unload redshift data and develop something to merge/join data, so redshift doesn't want to run the queries and merge.
upvoted 1 times

nroopa 1 year, 10 months ago
Selected Answer: A
Reason why C is incorrect is as it mentions Creating an external table using Amazon Redshift Spectrum for the call center data (which is already in Redshift) and performing the join with Amazon Redshift (not sure how to join on redshift as the data is already in Redshift) so i guess this incorrect until it is a type regarding the call center data. So my Option will be A
upvoted 1 times

NikkyDicky 1 year, 11 months ago
Selected Answer: C
going with C
upvoted 1 times

Hyperdanny 2 years, 1 month ago
I would pick B:

A: Lambda is limited to 15 minutes of execution time, might not be enough to unload.
C: The call center data is already in Redshift, the missing data is the airport data.

Best possible option seems to be B: Unload redshift data and develop something to merge/join data, so redshift doesn't want to run the queries and merge.
upvoted 2 times

Cloudbert 2 years, 1 month ago
Selected Answer: A
C is wrong. To use the CREATE External Table command the data has TO BE IN S3. "To define an external table in Amazon Redshift, use the CREATE EXTERNAL TABLE command. The external table statement defines the table columns, the format of your data files, and the location of your data in Amazon S3." The solution also requires to put as little burden on redshift as possible. Lambda and Glue are serverless and by choosing solution 1 we offload the burden from Redshift completely. Solution A must be correct.

upvoted 1 times

Debi_mishra 2 years, 1 month ago

I will say none of the answers are exactly correct without additional information such as call centre data structure and volume. D is not correct as its not serverless. C is not correct - you cant create external table when data is in redshift and also it will put load on redshift. A aij ok only if data is small else lambda will timeout Again B can be a problem if data is large as it will put load on redshift.

upvoted 2 times

pk349 2 years, 1 month ago

C: I passed the test

upvoted 1 times

itsme1 2 years, 3 months ago

Selected Answer: C

"Amazon Redshift Spectrum resides on dedicated Amazon Redshift servers that are independent of your cluster."
https://docs.aws.amazon.com/redshift/latest/dg/c-using-spectrum.html

C: Only caveat being that external table is not created in redshift-spectrum

A: UNLOAD is faster as oppose to Lambda, however, it also burdens redshift.

upvoted 1 times

[Removed] 2 years, 4 months ago

Selected Answer: C

Use Redshift Spectrum for it!

upvoted 1 times

silvaa360 2 years, 6 months ago

Selected Answer: C

I think that there is a typo here, instead of call center data it should be airline data. I had the same question in another paid question dump and the answer was the same as C), but will try to see if there is the same typo in there.

upvoted 2 times

nadavw 2 years, 7 months ago

Selected Answer: B

B seems to be a valid approach, taking into consideration that the load shouldn't be on RedShift, The Glue can export the data from RedShift and run (Spark serverless).
This blog explains it (ignore the daatbrew which is just a UI above the architecture):
https://aws.amazon.com/blogs/big-data/data-preparation-using-amazon-redshift-with-aws-glue-databrew/

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: C

C is the correct answer

A is wrong Although this is a possible solution, it requires a lot of development overhead to build Glue ETL scripts for joining the Redshift and S3 data. A better solution here is to use Amazon Redshift Spectrum

upvoted 2 times

rocky48 2 years, 8 months ago

Selected Answer: C

C looks simple enough.

upvoted 1 times

Question #16 *Topic 1*

A data analyst is using Amazon QuickSight for data visualization across multiple datasets generated by applications. Each application stores files within a separate Amazon S3 bucket. AWS Glue Data Catalog is used as a central catalog across all application data in Amazon S3. A new application stores its data within a separate S3 bucket. After updating the catalog to include the new application data source, the data analyst created a new Amazon QuickSight data source from an Amazon Athena table, but the import into SPICE failed.
How should the data analyst resolve the issue?

A. Edit the permissions for the AWS Glue Data Catalog from within the Amazon QuickSight console.

B. Edit the permissions for the new S3 bucket from within the Amazon QuickSight console.

C. Edit the permissions for the AWS Glue Data Catalog from within the AWS Glue console.

D. Edit the permissions for the new S3 bucket from within the S3 console.

Suggested Answer: *B*
Reference:
https://aws.amazon.com/blogs/big-data/harmonize-query-and-visualize-data-from-various-providers-using-aws-glue-amazon-athena-and-amazon- quicksight/

Community vote distribution
B (100%)

cloud4gr8 Highly Voted 3 years, 9 months ago
B.
https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-athena-insufficient-permissions.html
upvoted 29 times

lakediver 3 years, 6 months ago
For other issues please read - https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-athena.html
upvoted 1 times

gofavad926 Most Recent 1 year, 8 months ago
Selected Answer: B
B. https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-connect-S3.html
upvoted 1 times

nroopa 1 year, 10 months ago
Quick sight already has access to AWS Glue Data Catalog So A and C are not valid.
The issue is Quick sight does not have access to new S3 bucket, so we need to Edit the permissions for the new S3 bucket from within the S3 console to Give access to Quick sight
So Option D
upvoted 1 times

nroopa 1 year, 10 months ago
So its B . https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-connect-S3.html
upvoted 2 times

NikkyDicky 1 year, 11 months ago
Selected Answer: B
its a b
upvoted 1 times

pk349 2 years, 1 month ago
B: I passed the test
upvoted 2 times

Arka_01 2 years, 9 months ago
Selected Answer: B
As new S3 bucket is added. The permission to this bucket needs to be added from QuickSight console.
upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: B

Selected Answer: B

upvoted 1 times

ru4aws 2 years, 11 months ago

Selected Answer: B

https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-connect-athena.html

upvoted 1 times

ahmed_maher 3 years ago

B is right

upvoted 1 times

jrheen 3 years, 2 months ago

Answer : B

upvoted 1 times

aws2019 3 years, 7 months ago

B is right

upvoted 1 times

keitahigaki 3 years, 7 months ago

The new Quicksight dashboard doesn't have S3 access. From within the S3 console, edit the new bucket permissions.
https://docs.aws.amazon.com/ja_jp/quicksight/latest/user/create-a-data-set-athena.html

upvoted 1 times

Huy 3 years, 8 months ago

Agree with B. D is possible if the bucket policy has a Deny rule with Quicksight service role but it is rarely.
https://aws.amazon.com/premiumsupport/knowledge-center/quicksight-deny-policy-allow-bucket/

upvoted 2 times

Shraddha 3 years, 8 months ago

Answer B. A and C = wrong, Glue is updated without error so no problem there, also Glue permission is not set in QuickSight, QuickSight connects to Athena, not via Glue. D = wrong, QuickSight will create the rules for you.

Note: Athena is a very strange service as it transparently uses user's access to S3 buckets, instead of relying on service roles like most other AWS services. So, to be able to use Athena, the user itself will need to have S3 access, there is no service role creation for Athena.

https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-athena-insufficient-permissions.html

upvoted 2 times

sayed 3 years, 8 months ago

B as per the below
If you need use Amazon QuickSight with Amazon Athena or Amazon Athena Federated Query, you first need to authorize connections to Athena and the associated buckets in Amazon Simple Storage Service (Amazon S3).
https://docs.aws.amazon.com/quicksight/latest/user/athena.html

upvoted 1 times

Pruthvi 3 years, 8 months ago

B -
Athena is able to access the data source so its not S3 problem. On the quick sight end the new bucket should be added
https://docs.aws.amazon.com/quicksight/latest/user/troubleshoot-connect-S3.html

upvoted 2 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 1 times

Question #17 *Topic 1*

A team of data scientists plans to analyze market trend data for their company's new investment strategy. The trend data comes from five different data sources in large volumes. The team wants to utilize Amazon Kinesis to support their use case. The team uses SQL-like queries to analyze trends and wants to send notifications based on certain significant patterns in the trends. Additionally, the data scientists want to save the data to Amazon S3 for archival and historical re- processing, and use AWS managed services wherever possible. The team wants to implement the lowest-cost solution.
Which solution meets these requirements?

A. Publish data to one Kinesis data stream. Deploy a custom application using the Kinesis Client Library (KCL) for analyzing trends, and send notifications using Amazon SNS. Configure Kinesis Data Firehose on the Kinesis data stream to persist data to an S3 bucket.

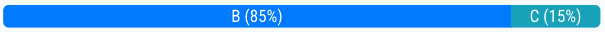
B. Publish data to one Kinesis data stream. Deploy Kinesis Data Analytic to the stream for analyzing trends, and configure an AWS Lambda function as an output to send notifications using Amazon SNS. Configure Kinesis Data Firehose on the Kinesis data stream to persist data to an S3 bucket.

C. Publish data to two Kinesis data streams. Deploy Kinesis Data Analytics to the first stream for analyzing trends, and configure an AWS Lambda function as an output to send notifications using Amazon SNS. Configure Kinesis Data Firehose on the second Kinesis data stream to persist data to an S3 bucket.

D. Publish data to two Kinesis data streams. Deploy a custom application using the Kinesis Client Library (KCL) to the first stream for analyzing trends, and send notifications using Amazon SNS. Configure Kinesis Data Firehose on the second Kinesis data stream to persist data to an S3 bucket.

Suggested Answer: *A*

Community vote distribution



Priyanka_01 Highly Voted 3 years, 9 months ago

B any thoughts ?
Multiple applications can consume from a single Kinesis Stream
Kinesis Analytics for sql like queries for analysis
Kinesis firhose can directly transfer the data into S3 from the same data stream
upvoted 50 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: B

Correct answer is B as a single Kinesis Data Streams can be configured for data ingestion. The stream can be consumed by Kinesis Data Firehose to store the data in S3 for archival. The stream can also be consumed by kinesis Data Analytics for analysis and use Lambda and SNS for notifications.

Option A is wrong as KCL solution is not ideal for executing SQL-like queries to analyze trends.

Options C & D are wrong as two Kinesis Data Streams is not needed.
upvoted 5 times

nroopa Most Recent 1 year, 10 months ago

Option B
Multiple applications can consume from a single Kinesis Stream
Kinesis Analytics for sql like queries for analysis & the Output can be sent to lambda to send SNs Trigger
https://docs.aws.amazon.com/kinesisanalytics/latest/dev/how-it-works-output-lambda.html
Kinesis firehose can directly transfer the data into S3 from the same data stream
upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: B

its a B
upvoted 1 times

pk349 2 years, 1 month ago

B: I passed the test

upvoted 1 times

[Removed] 2 years, 4 months ago

Selected Answer: B

I would also suggest B

upvoted 1 times

henom 2 years, 7 months ago

Correct Answer-B

upvoted 1 times

renfdo 2 years, 7 months ago

Selected Answer: C

I think it's C. We need Two D Data Stream. One to consume RAW data and another one to send Agregated data after Kinesis Data Analytics.
https://aws.amazon.com/blogs/big-data/building-a-real-time-notification-system-with-amazon-kinesis-data-streams-for-amazon-dynamodb-and-amazon-kinesis-data-analytics-for-apache-flink/

upvoted 2 times

thirukudil 2 years, 8 months ago

Answer is B. Using Kinesis analytics, we can analyse the trends using sql like queries. Same data stream would be a source to Firehose to put the data in s3.

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: B

KDA is best applied in this scenario. Also we do not need multiple Kinesis streams to ingest from different data sources. Any custom application will add additional development and testing overhead.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: B

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer is B

upvoted 1 times

jmensah60 3 years, 3 months ago

Selected Answer: B

Kinesis Analytics for sql like queries for analysis

upvoted 2 times

aws2019 3 years, 7 months ago

The answer is B.

upvoted 3 times

SGES 3 years, 7 months ago

B is preferable because:
Kinesis firhose can directly transfer the data into S3 from the same data stream in addition to cost effectiveness specified

upvoted 2 times

Shraddha 3 years, 7 months ago

KCL is self-service, so A and D are out. If you want multiple consumers for the same stream, just make sure you have enough shards to deal with the read-write throughput. A stream is different from a queue in that one can traverse back and forth in a stream, where in a queue one can only process one by one.

upvoted 1 times

gunjan4392 3 years, 7 months ago

B seems okay

upvoted 1 times

Question #18 *Topic 1*

A company currently uses Amazon Athena to query its global datasets. The regional data is stored in Amazon S3 in the us-east-1 and us-west-2 Regions. The data is not encrypted. To simplify the query process and manage it centrally, the company wants to use Athena in us-west-2 to query data from Amazon S3 in both
Regions. The solution should be as low-cost as possible.
What should the company do to achieve this goal?

A. Use AWS DMS to migrate the AWS Glue Data Catalog from us-east-1 to us-west-2. Run Athena queries in us-west-2.

B. Run the AWS Glue crawler in us-west-2 to catalog datasets in all Regions. Once the data is crawled, run Athena queries in us-west-2.

C. Enable cross-Region replication for the S3 buckets in us-east-1 to replicate data in us-west-2. Once the data is replicated in us-west-2, run the AWS Glue crawler there to update the AWS Glue Data Catalog in us-west-2 and run Athena queries.

D. Update AWS Glue resource policies to provide us-east-1 AWS Glue Data Catalog access to us-west-2. Once the catalog in us-west-2 has access to the catalog in us-east-1, run Athena queries in us-west-2.

Suggested Answer: *C*

Community vote distribution

zanhsieh Highly Voted 3 years, 9 months ago

B.
AWS DMS is not for this purpose, so A dropped. C would be costly since it literally replicates all data. There's no "resource policies" in AWS Glue, so D dropped.
upvoted 30 times

Huy 3 years, 7 months ago

I agree with you that D is wrong but my ideas is you shouldn't based on a property that is not available for the service. Instead, think in-depth about what is the answer actually suggest. https://docs.aws.amazon.com/glue/latest/dg/glue-resource-policies.html
Here, the answer wants AWS Glue to use Data Catalog from different region which is not supported.
upvoted 2 times

certificationJunkie 3 years, 1 month ago

glue crawler will simply generate the metadata on top of s3 files. But the Athena running in another region will still not have access to the first region files. Also, even glue crawler might not have permission to crawl in another region s3 files. Hence replication is the only option.
upvoted 2 times

certificationJunkie 3 years, 1 month ago

No, glue crawler is not restricted to a region and can catalogue data in other regions. And then Athena can use the catalogue and generate results. I have seen this happening in my project
upvoted 5 times

JoellaLi 2 years, 8 months ago

There is 'resource policies':
https://docs.aws.amazon.com/glue/latest/dg/glue-policy-examples-resource-policies.html
upvoted 3 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: B

B is correct as AWS Glue can crawl data in different AWS Regions. When you define an Amazon S3 data store to crawl, you can choose whether to crawl a path in your account or another account.

The output of the crawler is one or more metadata tables defined in the AWS Glue Data Catalog. A table is created for one or more files found in your data store. If all the Amazon S3 files in a folder have the same schema, the crawler creates one table. Also, if the Amaazon S3 object is partitioned, only one metadata table is created.

A is wrong because you can't use AWS DMS with AWS Glue Data Catalog.

C is incorrect because replicating the data in S3 means that your storage costs will also double.

D is wrong because a resource-based policy is primarily used to provide IAM users and roles granular access to metadata definitions of databases, tables, connections, and user-defined functions, and not the actual S3 data.

upvoted 14 times

GCPereira Most Recent 1 year, 6 months ago

A: DMS is not required to migrate data from one region to another. It can even be used to migrate data from an S3 bucket to another bucket in another account, but there are better and cheaper ways to do this (considering the volume of data, of course).

B: It is the correct alternative. Glue crawlers can catalog data that is in different regions. It's simple to set up and not expensive.

C: Cross-region works for data replication, but it will be duplicated unnecessarily.

D: This type of permissions is best suited for LakeFormation and would not help catalog data that is in different regions.

upvoted 1 times

nroopa 1 year, 10 months ago

Option D
https://aws.amazon.com/blogs/big-data/configure-cross-region-table-access-with-the-aws-glue-catalog-and-aws-lake-formation/

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: B

going w B

upvoted 1 times

Cloudbert 2 years, 1 month ago

Selected Answer: B

B. Source: https://docs.aws.amazon.com/glue/latest/dg/crawler-data-stores.html. You can choose to crawl a path in your account or in another account. Crawlers use an AWS Identity and Access Management (IAM) role for permission to access your data stores. The role you pass to the crawler must have permission to access Amazon S3 paths and Amazon DynamoDB tables that are crawled. Another source: https://docs.aws.amazon.com/athena/latest/ug/querying-across-regions.html. Athena can query cross-region Athena supports the ability to query Amazon S3 data in an AWS Region that is different from the Region in which you are using Athena. Querying across Regions can be an option when moving the data is not practical or permissible, or if you want to query data across multiple regions. Even if Athena is not available in a particular Region, data from that Region can be queried from another Region in which Athena is available.

upvoted 1 times

Debi_mishra 2 years, 1 month ago

B is correct for context of this question but will be a bad implementation in real life. D can be good pattern but with help of Lakeformation.

upvoted 2 times

pk349 2 years, 1 month ago

B: I passed the test

upvoted 2 times

austinoy 2 years, 4 months ago

the data is not encrypted so moving data is not "practical or permissible"?

upvoted 1 times

Ashoks 2 years, 4 months ago

D should be...

upvoted 1 times

mulder1989 2 years, 4 months ago

A, B, D simply wouldn't work because of lacking connection to the data source. The only thing that I am not sure is about the 'lowest cost'. It can be option B if the wording implies that the connectivity exits
https://aws.amazon.com/blogs/big-data/create-cross-account-and-cross-region-aws-glue-connections/

upvoted 1 times

Nicoben 2 years, 5 months ago

D.

See: https://docs.aws.amazon.com/glue/latest/dg/cross-account-access.html

upvoted 1 times

Chelseajcole 2 years, 5 months ago

That's why D is wrong? Each AWS account owns a single catalog in an AWS Region whose catalog ID is the same as the AWS account ID

https://docs.aws.amazon.com/glue/latest/dg/glue-resource-policies.html

upvoted 2 times

Haimett 2 years, 8 months ago

Selected Answer: B

Both B and D will work. The answer is B because option D is a bit more expensive.

upvoted 1 times

LukeTran3206 2 years, 8 months ago

Selected Answer: D

Must be D

upvoted 2 times

rav009 2 years, 8 months ago

Selected Answer: B

B is correct

D is wrong because there is no resource policies but only trust policy.

upvoted 1 times

VishalSingh 2 years, 8 months ago

It has https://docs.aws.amazon.com/glue/latest/dg/glue-resource-policies.html

upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: B

The lowest cost option is B. All other options are involved with greater cost, as data migration between regions costs more.

upvoted 1 times

Question #19 *Topic 1*

A large company receives files from external parties in Amazon EC2 throughout the day. At the end of the day, the files are combined into a single file, compressed into a gzip file, and uploaded to Amazon S3. The total size of all the files is close to 100 GB daily. Once the files are uploaded to Amazon S3, an
AWS Batch program executes a COPY command to load the files into an Amazon Redshift cluster.
Which program modification will accelerate the COPY process?

A. Upload the individual files to Amazon S3 and run the COPY command as soon as the files become available.

B. Split the number of files so they are equal to a multiple of the number of slices in the Amazon Redshift cluster. Gzip and upload the files to Amazon S3. Run the COPY command on the files.

C. Split the number of files so they are equal to a multiple of the number of compute nodes in the Amazon Redshift cluster. Gzip and upload the files to Amazon S3. Run the COPY command on the files.

D. Apply sharding by breaking up the files so the distkey columns with the same values go to the same file. Gzip and upload the sharded files to Amazon S3. Run the COPY command on the files.

Suggested Answer: *B*
Reference:
https://docs.aws.amazon.com/redshift/latest/dg/t_splitting-data-files.html

Community vote distribution
B (100%)

singh100 Highly Voted 3 years, 9 months ago
B. Split your data into files so that the number of files is a multiple of the number of slices in your cluster. That way Amazon Redshift can divide the data evenly among the slices.
upvoted 23 times

Shraddha Highly Voted 3 years, 7 months ago
B :
This is a textbook question. Sequential loading vs. parallel loading.

https://docs.aws.amazon.com/redshift/latest/dg/t_splitting-data-files.html
upvoted 7 times

GCPereira Most Recent 1 year, 6 months ago
files -> EC2 -> merge files at the end of the day single file compressed -> s3 (100GB daily)

A: The copy command for a large file (100GB) is slow and not effective as redshift will try to distribute the processing across the cluster and only after this division will the copy be carried out.

B: Files must be large enough to run on only one slice of the node. With this pre-processing done, the master node does not need to worry about "allocating memory" to copy this file. If each slice processes a file, the transfer speed will be optimal.

C: It's a smart option, but not the most effective. The number of slices is directly related to the number of nodes, but if the division is made thinking only about the number of nodes, it is possible to make the mistake of executing the COPY command for files that are too large.

D: The dist style must be done after loading the data.
upvoted 1 times

nroopa 1 year, 10 months ago
Option B
upvoted 1 times

NikkyDicky 1 year, 11 months ago
Selected Answer: B
I think B
upvoted 1 times

pk349 2 years, 1 month ago

B: I passed the test

upvoted 1 times

roymunson 1 year, 7 months ago

Agree I passed the test two times in a row.

upvoted 1 times

SamQiu 2 years, 6 months ago

Why can't I use Option D?

upvoted 1 times

[Removed] 2 years, 6 months ago

b

https://docs.aws.amazon.com/glue/latest/dg/cross-account-access.html

Granting access to Data Catalog resources across accounts enables your extract, transform, and load (ETL) jobs to query and join data from different accounts.

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

B is correct as the COPY command loads the data in parallel from multiple files, dividing the workload among the nodes in your cluster. When you load all the data from a single large file, Amazon Redshift is forced to perform a serialized load, which is much slower. Split your load data files so that the files are about equal size, between 1 MB and 1 GB after compression. For optimum parallelism, the ideal size is between 1 MB and 125 MB after compression. The number of files should be a multiple of the number of slices in your cluster.

upvoted 7 times

Arka_01 2 years, 9 months ago

Selected Answer: B

GZIP cannot be split. So first split the files and then gzip the slices. Also, it is recommended to have number of files equal to a number which is multiple of total slices in Redshift cluster, so that copy command can engage all worked nodes parallelly and evenly distribute the load.

upvoted 1 times

renfdo 2 years, 9 months ago

Selected Answer: B

B is the right answer

upvoted 1 times

renfdo 2 years, 9 months ago

B is the right answer

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: B

B is the right answer

upvoted 1 times

lostsoul07 3 years, 7 months ago

B is the right answer

upvoted 1 times

BillyC 3 years, 7 months ago

B is correct for me

upvoted 1 times

sanjaym 3 years, 8 months ago

B is the answer.

upvoted 1 times

Karan_Sharma 3 years, 8 months ago

Option B, By using a single full file forces copy to do a serial load. Splitting the files in multiple of number of slices in cluster and compressing them is ideal for better performance of copy

upvoted 2 times

Question #20 *Topic 1*

A large ride-sharing company has thousands of drivers globally serving millions of unique customers every day. The company has decided to migrate an existing data mart to Amazon Redshift. The existing schema includes the following tables.

➪ A trips fact table for information on completed rides.

➪ A drivers dimension table for driver profiles.

➪ A customers fact table holding customer profile information.

The company analyzes trip details by date and destination to examine profitability by region. The drivers data rarely changes. The customers data frequently changes.

What table design provides optimal query performance?

A. Use DISTSTYLE KEY (destination) for the trips table and sort by date. Use DISTSTYLE ALL for the drivers and customers tables.

B. Use DISTSTYLE EVEN for the trips table and sort by date. Use DISTSTYLE ALL for the drivers table. Use DISTSTYLE EVEN for the customers table.

C. Use DISTSTYLE KEY (destination) for the trips table and sort by date. Use DISTSTYLE ALL for the drivers table. Use DISTSTYLE EVEN for the customers table.

D. Use DISTSTYLE EVEN for the drivers table and sort by date. Use DISTSTYLE ALL for both fact tables.

Suggested Answer: *A*

Community vote distribution

C (100%)

zanhsieh Highly Voted 3 years, 9 months ago

C.

Drivers' data -> ALL, Customer's data -> EVEN, Trips table -> KEY (destination) & sort by date

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

https://slideshare.net/AmazonWebServices/deep-dive-on-amazon-redshift-80877515

upvoted 38 times

GeeBeeEl 3 years, 9 months ago

Not sure how the 2 links gave you this answer! Not saying your suggestion is wrong

upvoted 2 times

jove Highly Voted 3 years, 8 months ago

IMO, it should be B.. Reasons: Distributing the data on destination might cause a data skew which we don't want. If there is no clear dist key for a fact, it's better to dist it evenly.

upvoted 12 times

NikkyDicky Most Recent 1 year, 11 months ago

Selected Answer: C

C is right

upvoted 1 times

penguins2 1 year, 11 months ago

C is the correct answer!

upvoted 1 times

pk349 2 years, 1 month ago

C: I passed the test

upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: C

Correct answer is C as as the trip would be queries on destination and date, the trips table needs a DISTSTYLE KEY (destination) for the trips table and sort by date. As the drivers data rarely changes DISTSTYLE ALL can be applied for the drivers table, which will maintain a copy per node. Also as customers data changes frequently,

upvoted 5 times

Arka_01 2 years, 9 months ago

Selected Answer: C

Use All Distribution for rarely changing tables, as they are copied to all slices. Use Even distribution to frequently changing and large tables, as Redshift engine can randomly distribute them to different data slices. Use diststyle key for the tables where you know a join key.

upvoted 8 times

rocky48 2 years, 10 months ago

Answer is C

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: C

Answer is C

upvoted 1 times

MWL 3 years, 1 month ago

Selected Answer: C

C should be right.

The data will be analized by destination. And the requirement doesn't mention that it need to join trip and customer/driver table. So, using DISTSTYLE KEY (destination) for the trips should improve the performance of trip data.

For A, there are millons of customers, so using ALL for that will cause too much copy data.

upvoted 2 times

Shivanikats 3 years, 5 months ago

I think answer is B. Destination is not as unique a key and can cause skew with key partition. So even for the trips seems right. Drivers is updated rarely, so suitable for All. Cust is updated often so Even is good for it too.

upvoted 3 times

Bambur 3 years, 7 months ago

The answer is B. We don't know which key (with high cardinality) to use for fact table for even distribution so we should chose even diststyle and use fields that should be used to access data as sort key.. Rare updated dimension table good candidate for diststyle all, and another one frequently updated should have even diststyle.

upvoted 4 times

Huy 3 years, 8 months ago

FACT table in DW is central table and will be queried the most. Because we query trips by destination and date therefore the higher cardinality is destination -> use Destination as key. Drivers is dimension table and data is small so can be ALL to speedup the join. Customer data frequently changes and we are not sure which columns should be joined so EVEN is safe.

upvoted 4 times

Donell 3 years, 8 months ago

Answer C

upvoted 1 times

Shraddha 3 years, 8 months ago

Ans C..This is a textbook question. However, if there is an answer where both customer and driver tables are EVEN, I would go for it. ALL does not quite give benefits over EVEN.

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

upvoted 3 times

gunjan4392 3 years, 8 months ago

C seems okay

upvoted 1 times

ariane_tateishi 3 years, 8 months ago

C in my opnion is the best choice, because the Trip table is a fact table so it will join with the other tables, like customer and driver tables. The table customer is frequently updated, so in this case ALL is not recommended.

upvoted 2 times

Question #21 *Topic 1*

Three teams of data analysts use Apache Hive on an Amazon EMR cluster with the EMR File System (EMRFS) to query data stored within each teams Amazon
S3 bucket. The EMR cluster has Kerberos enabled and is configured to authenticate users from the corporate Active Directory. The data is highly sensitive, so access must be limited to the members of each team.
Which steps will satisfy the security requirements?

A. For the EMR cluster Amazon EC2 instances, create a service role that grants no access to Amazon S3. Create three additional IAM roles, each granting access to each team's specific bucket. Add the additional IAM roles to the cluster's EMR role for the EC2 trust policy. Create a security configuration mapping for the additional IAM roles to Active Directory user groups for each team.

B. For the EMR cluster Amazon EC2 instances, create a service role that grants no access to Amazon S3. Create three additional IAM roles, each granting access to each team's specific bucket. Add the service role for the EMR cluster EC2 instances to the trust policies for the additional IAM roles. Create a security configuration mapping for the additional IAM roles to Active Directory user groups for each team.

C. For the EMR cluster Amazon EC2 instances, create a service role that grants full access to Amazon S3. Create three additional IAM roles, each granting access to each team's specific bucket. Add the service role for the EMR cluster EC2 instances to the trust polices for the additional IAM roles. Create a security configuration mapping for the additional IAM roles to Active Directory user groups for each team.

D. For the EMR cluster Amazon EC2 instances, create a service role that grants full access to Amazon S3. Create three additional IAM roles, each granting access to each team's specific bucket. Add the service role for the EMR cluster EC2 instances to the trust polices for the base IAM roles. Create a security configuration mapping for the additional IAM roles to Active Directory user groups for each team.

Suggested Answer: *C*

Community vote distribution

B (100%)

jack42 Highly Voted 3 years, 8 months ago

No doubt its B. If you will have full access on Ec2 instance role and no role match then it will fall back to the default role [When a cluster application makes a request to Amazon S3 through EMRFS, EMRFS evaluates role mappings in the top-down order that they appear in the security configuration. If a request made through EMRFS doesn't match any identifier, EMRFS falls back to using the service role for cluster EC2 instances.] Also this is tested fully and its more secure then any other options.

upvoted 30 times

Shraddha Highly Voted 3 years, 7 months ago

Ans B :
This is a textbook question. Basically you:

create a new EMR service role, removing default permission from original service role which is too permissive with s3:*
create some new roles to allow access to respective s3 buckets
EMRFS by default will assume EMR service role, which means it gets all access to S3, but can be configured to assume an additional role created by user
To be able to do that, user-created roles needs to trust EMR service role (because EMRFS will assume that role first)
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-emrfs-iam-roles.html

upvoted 11 times

tsangckl Most Recent 1 year, 3 months ago

Bling choose A
for the below explanation
Option A is correct because it ensures that each team only has access to its own S3 bucket. By creating a service role that grants no access to Amazon S3 for the EMR cluster EC2 instances, you prevent unauthorized access. Then, by creating additional IAM roles that grant access to each team's specific bucket and adding these roles to the EMR role for the EC2 trust policy, you ensure that each team can access only its own data. Finally, by creating a security configuration mapping for the additional IAM roles to Active Directory user groups for each team, you ensure that only the members of each team can access their own data.
Other options are not the best solutions for this scenario. For example, Options B, C, and D involve adding the service role for the EMR cluster EC2 instances to the trust policies for the additional IAM roles or the base IAM roles, which could potentially allow unauthorized access to the S3 buckets.

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: B

B is right

upvoted 1 times

pk349 2 years, 1 month ago

B: I passed the test

upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B as the EMR service role should be provided with no access and the mapping defined for security configuration for using IAM roles mapped to groups.

Option A is wrong as the service role for the EMR cluster EC2 instances should be updated to the trust policies for the additional IAM roles.

Options C & D are wrong as the EMR service role should have no access.

upvoted 5 times

rav009 2 years, 8 months ago

Selected Answer: B

B

B is right, the service role need assume the additional roles, which means add it to the trust policy of the additional roles.

A is the opposite.

upvoted 1 times

Grimreaper69 2 years, 11 months ago

isnt b and c the same?

upvoted 1 times

rudramadhu 2 years, 11 months ago

B - create a service role that grants no access to Amazon S3.

C- create a service role that grants FULL access to Amazon S3

B is the right choice

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: B

Answer B

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer is B

upvoted 1 times

aws2019 3 years, 7 months ago

Answer B

upvoted 1 times

Donell 3 years, 7 months ago

Answer B

upvoted 2 times

Antfoot 3 years, 7 months ago

When a cluster application makes a request to Amazon S3 through EMRFS, EMRFS evaluates role mappings in the top-down order that they appear in the security configuration. If a request made through EMRFS doesn't match any identifier, EMRFS falls back to using the service role for cluster EC2 instances. For this reason, we recommend that the policies attached to this role limit permissions to Amazon S3. For more information, see Service Role for Cluster EC2 Instances (EC2 Instance Profile).

upvoted 1 times

Exia 3 years, 8 months ago

C. We need additional IAM roles.

A, B. If EC2 service role has no access to Amazon S3, no one on this EC2 can access S3 at any level. Besides, S3 deny unauthorized access by

default.

D. We need additional IAM roles.

upvoted 4 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 3 times

lvi 3 years, 8 months ago

Would go with A.

By default, no privilege given in the "default" instance profile.

Privileges are given through dedicated roles and policies for each domain.

Then allow these roles to be assumed by the EMR Service Role.

EMR will then use the appropriate IAM roles based on to the role mapping definition.

upvoted 1 times

blubb 3 years, 8 months ago

B is correct as of https://aws.amazon.com/de/blogs/big-data/build-a-multi-tenant-amazon-emr-cluster-with-kerberos-microsoft-active-directory-integration-and-emrfs-authorization/

upvoted 4 times

Question #22 *Topic 1*

A company is planning to create a data lake in Amazon S3. The company wants to create tiered storage based on access patterns and cost objectives. The solution must include support for JDBC connections from legacy clients, metadata management that allows federation for access control, and batch-based ETL using PySpark and Scala. Operational management should be limited.
Which combination of components can meet these requirements? (Choose three.)

A. AWS Glue Data Catalog for metadata management

B. Amazon EMR with Apache Spark for ETL

C. AWS Glue for Scala-based ETL

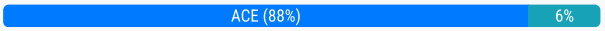
D. Amazon EMR with Apache Hive for JDBC clients

E. Amazon Athena for querying data in Amazon S3 using JDBC drivers

F. Amazon EMR with Apache Hive, using an Amazon RDS with MySQL-compatible backed metastore

Suggested Answer: *BEF*
Reference:
https://d1.awsstatic.com/whitepapers/Storage/data-lake-on-aws.pdf

Community vote distribution



Prodip Highly Voted 3 years, 9 months ago
I will go with A,C,E . Glue can do both pyspark and scala based ETL. Glue for Metadata and JDBC drivers to connect Athena from outside of AWS. Server less . so, Operational management is limited
upvoted 48 times

abhineet 3 years, 8 months ago
ya i thought so too, ACE for me
upvoted 4 times

jack42 Highly Voted 3 years, 8 months ago
Each word has meaning, So I will go with ABD, A metadata management that allows federation for access control, B- batch-based ETL using PySpark, D-JDBC connections from legacy clients. Not-C- because it mentioned only scala but questions mentioned scala operation is limited, E- you need JDBC to connect clinet not the Athena
upvoted 6 times

Mahesh22 3 years, 8 months ago
Correct. ABD is right
upvoted 1 times

vanireddy 3 years, 8 months ago
I agree with this. Correct is ABD.
upvoted 1 times

shammous 2 years, 6 months ago
EMR=Operationd overhead. ETL does it all and it is a managed service. ACE is better answer
upvoted 1 times

shammous 2 years, 6 months ago
I mean AWS Glue (not ETL) is a serverless service and you don't need to provision it.
upvoted 1 times

abgz887 2 years, 7 months ago
if we select B,D (EMR-spark,Hive-jdbc),does it not make more sense to use Emr-Hive-datastore(F),instead of glue-catalog(A),limiting operational management.
- making BDF more appropriate.
upvoted 1 times

tsangckl Most Recent 1 year, 3 months ago

Bing
Option A is correct because AWS Glue Data Catalog provides a unified metadata repository across a variety of data sources and data formats, and it integrates with Amazon S3, Amazon RDS, Amazon Athena, Amazon Redshift, and others.
Option B is correct because Amazon EMR with Apache Spark supports PySpark and Scala for batch-based ETL processing.
Option E is correct because Amazon Athena supports SQL queries and can be integrated with JDBC drivers, allowing legacy clients to execute queries.
upvoted 1 times

NarenKA 1 year, 4 months ago

Selected Answer: ABE

I will go with A. AWS Glue Data Catalog, B.Amazon EMR with Apache Spark, E. Amazon Athena aligns well with the company's requirements for a data lake architecture, offering a balance of performance, cost-efficiency, and ease of management.

While C is also a viable option for ETL processes, it's more aligned with serverless ETL jobs and might not be as flexible for Scala as Amazon EMR with Apache Spark. D and F could provide JDBC connectivity and metadata management but are more operationally intensive and less integrated with S3 tiered storage strategies compared to using Athena with the Glue Data Catalog.
upvoted 1 times

geekfrosty 1 year, 10 months ago

Why are we saying C ? C just says "Scala" ETL, even though Glue supports both pyspark and scala and AWS managed, the option specifically mentions "Scala based". Requirement is for both Scala and Pyspark that directly points to EMR. answer should be ABE.. about operational management, it says "limited", and EMR can qualify with it. using glue there is 'no' operational overhead.
upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: ACE

ACE it
upvoted 1 times

pk349 2 years, 1 month ago

ACE: I passed the test
upvoted 3 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: ACE

Correct answers are A, C & E

Option A as Glue Data Catalog provides metadata management with the federation for access control.

Option C as AWS Glue supports both serverless PySpark and Scala-based ETL with the least operational overhead.

Option E as Athena can be used for querying S3 data. Athena can be connected using JDBC drivers from the external legacy clients.

Options B, D & E is wrong as using EMR and RDS would increase the operational management and cost.
upvoted 5 times

Arka_01 2 years, 9 months ago

Selected Answer: ACE

As operation management should be less, so all EMR related options are invalid, as EMR needs management of underlying EC2 instances
upvoted 2 times

Abep 2 years, 9 months ago

Selected Answer: ACE

A. *Less* operational overhead compared to F (selected)
B. High operational overhead, when compared to "C" AWS Glue based Scala
C. *Less* operational overhead compared to "B" EMR PySpark (selected)
D. Higher operational overhead when compared to "E" Athena. https://docs.aws.amazon.com/emr/latest/ReleaseGuide/HiveJDBCDriver.html
E. *Less* operational overhead when compared to "D" EMR Hive JDBC (selected)
F. Higher operational overhead when compared to "A" Glue metadata
upvoted 2 times

rocky48 2 years, 10 months ago

Selected Answer: ACE

I will go with A,C,E

upvoted 2 times

GarfieldBin 3 years ago

Selected Answer: ABC

D and F are wrong because the question never mentions Hive. E is not right, since Athena don't need JDBC to query S3. C is right because AWS Glue can be used for Scala-based ETL.

A is right because Glue can connect on-premises DB through JDBC. https://aws.amazon.com/blogs/big-data/how-to-access-and-analyze-on-premises-data-stores-using-aws-glue/. B is right because Apache Spark can support PySpark.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: ACE

My Answer is A, C & E

upvoted 3 times

Japanese1 3 years, 4 months ago

D, F are clearly wrong.

D : JDBC connection from older clients is required, NOT from Athena.

F : It is redundant to use RDS as a metastore. And there is no requirement for MySQL-compatible backed metastore.

I'm torn between B and C. B is predominant in terms of cost constraints, but I am doubtful.

upvoted 1 times

Donell 3 years, 7 months ago

Answer: A,C,E

EMR has operational overhead.

upvoted 2 times

Donell 3 years, 7 months ago

Answer: A,C,E

EMR has operational overhead.

upvoted 3 times

Shraddha 3 years, 7 months ago

Ans - ACE

Note: This is a free score question. Anything EMR comparing to serverless Glue / Athena is operational overhead. Also remember Glue can do PySpark and Scala, and Athena can do JDBC.

upvoted 4 times

Question #23 *Topic 1*

A company wants to optimize the cost of its data and analytics platform. The company is ingesting a number of .csv and JSON files in Amazon S3 from various data sources. Incoming data is expected to be 50 GB each day. The company is using Amazon Athena to query the raw data in Amazon S3 directly. Most queries aggregate data from the past 12 months, and data that is older than 5 years is infrequently queried. The typical query scans about 500 MB of data and is expected to return results in less than 1 minute. The raw data must be retained indefinitely for compliance requirements.
Which solution meets the company's requirements?

A. Use an AWS Glue ETL job to compress, partition, and convert the data into a columnar data format. Use Athena to query the processed dataset. Configure a lifecycle policy to move the processed data into the Amazon S3 Standard-Infrequent Access (S3 Standard-IA) storage class 5 years after object creation. Configure a second lifecycle policy to move the raw data into Amazon S3 Glacier for long-term archival 7 days after object creation.

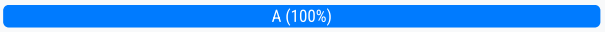
B. Use an AWS Glue ETL job to partition and convert the data into a row-based data format. Use Athena to query the processed dataset. Configure a lifecycle policy to move the data into the Amazon S3 Standard-Infrequent Access (S3 Standard-IA) storage class 5 years after object creation. Configure a second lifecycle policy to move the raw data into Amazon S3 Glacier for long-term archival 7 days after object creation.

C. Use an AWS Glue ETL job to compress, partition, and convert the data into a columnar data format. Use Athena to query the processed dataset. Configure a lifecycle policy to move the processed data into the Amazon S3 Standard-Infrequent Access (S3 Standard-IA) storage class 5 years after the object was last accessed. Configure a second lifecycle policy to move the raw data into Amazon S3 Glacier for long-term archival 7 days after the last date the object was accessed.

D. Use an AWS Glue ETL job to partition and convert the data into a row-based data format. Use Athena to query the processed dataset. Configure a lifecycle policy to move the data into the Amazon S3 Standard-Infrequent Access (S3 Standard-IA) storage class 5 years after the object was last accessed. Configure a second lifecycle policy to move the raw data into Amazon S3 Glacier for long-term archival 7 days after the last date the object was accessed.

Suggested Answer: *C*

Community vote distribution



astalavista1 Highly Voted 3 years, 2 months ago

Selected Answer: A

Agree with answer A as C&D was eliminated due to the last accessed rather than created for Lifecycle policy.
By compressing you save cost and converting to columnar data, performance is increased.

upvoted 11 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: A

Correct answer is A as columnar data format store data efficiently by employing column-wise compression and enables split and parallel processing.
Storing processed data in S3 in SA-IA and moving raw data in Glacier would help reduce costs.

Option B & D is wrong as it is recommended to use columnar data format for processing.

Options C is wrong as lifecycle rules are based on Object creation data and not last date when the object was accessed.

upvoted 6 times

GLam123 Most Recent 1 year, 8 months ago

Selected Answer: A

columnar and based on object creation time

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: A

A make sense

upvoted 1 times

pk349 2 years, 1 month ago

A: I passed the test
upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: A

It should be based on object creation, not based on object access
upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: A

Answer is A
upvoted 1 times

ru4aws 2 years, 11 months ago

Selected Answer: A

should be 5 years after object creation to Infrequent for processed data
and 7 days after object creation to glacier for raw data

There is no point of counting days from "Last accessed"
upvoted 2 times

dushmantha 2 years, 12 months ago

Selected Answer: A

Columnar data is a way of optimizing (eleminate B, D). And the lifecycle policy should be assigned after object creation (eleminate C). Ans is A
upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: A

Answer is A
upvoted 1 times

azi_2021 3 years, 2 months ago

ans should be A
upvoted 2 times

astalavista1 3 years, 2 months ago

Agree with answer A as C&D was eliminated due to the last accessed rather than created for Lifecycle policy.
By compressing you save cost and converting to columnar data, performance is increased.
upvoted 1 times

Question #24 *Topic 1*

An energy company collects voltage data in real time from sensors that are attached to buildings. The company wants to receive notifications when a sequence of two voltage drops is detected within 10 minutes of a sudden voltage increase at the same building. All notifications must be delivered as quickly as possible. The system must be highly available. The company needs a solution that will automatically scale when this monitoring feature is implemented in other cities. The notification system is subscribed to an Amazon Simple Notification Service (Amazon SNS) topic for remediation.
Which solution will meet these requirements?

A. Create an Amazon Managed Streaming for Apache Kafka cluster to ingest the data. Use an Apache Spark Streaming with Apache Kafka consumer API in an automatically scaled Amazon EMR cluster to process the incoming data. Use the Spark Streaming application to detect the known event sequence and send the SNS message.

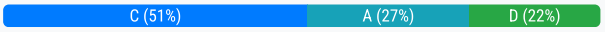
B. Create a REST-based web service by using Amazon API Gateway in front of an AWS Lambda function. Create an Amazon RDS for PostgreSQL database with sufficient Provisioned IOPS to meet current demand. Configure the Lambda function to store incoming events in the RDS for PostgreSQL database, query the latest data to detect the known event sequence, and send the SNS message.

C. Create an Amazon Kinesis Data Firehose delivery stream to capture the incoming sensor data. Use an AWS Lambda transformation function to detect the known event sequence and send the SNS message.

D. Create an Amazon Kinesis data stream to capture the incoming sensor data. Create another stream for notifications. Set up AWS Application Auto Scaling on both streams. Create an Amazon Kinesis Data Analytics for Java application to detect the known event sequence, and add a message to the message stream Configure an AWS Lambda function to poll the message stream and publish to the SNS topic.

Suggested Answer: *D*
Reference:
https://aws.amazon.com/kinesis/data-streams/faqs/

Community vote distribution



CHRIS12722222 Highly Voted 3 years, 2 months ago
Answer = D
upvoted 28 times

morpheus23 Highly Voted 3 years ago
It's C. Your confusing the "immediately" and ignoring that prior to that it's an event that happens as a sequence within 10 minutes, so firehose is a viable option.

AWS auto scaling does not support Amazon Kinesis so that rules out D
https://docs.aws.amazon.com/autoscaling/application/userguide/integrated-services-list.html
upvoted 16 times

Soumya92 1 year, 10 months ago
It seems AWS does support application auto-scaling for Kinesis Data streams since Nov 2018. https://aws.amazon.com/blogs/big-data/scaling-amazon-kinesis-data-streams-with-aws-application-auto-scaling/
upvoted 2 times

chinmayj213 1 year, 9 months ago
for Kinesis Data Stream we can use on demand option instead of provision that will able to handle
upvoted 1 times

flanfranco 2 years, 3 months ago
I think C is correct: https://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html#services-kinesis-windows
upvoted 1 times

flanfranco 2 years, 3 months ago
https://aws.amazon.com/blogs/big-data/best-practices-for-consuming-amazon-kinesis-data-streams-using-aws-lambda/
upvoted 1 times

zbyroger0902 1 year, 9 months ago

The two links seems to be evidence of why lambda is not the right choice for the question. It is stated in the two links you provided that lambda cannot deal with stateful computing except for tumbling window in the streams , and pattern recognition as the question asked for is obvious example of stateful computing that tumbling window is not applicable.
upvoted 3 times

siju13 2 years, 11 months ago
no, within the 10 minutes, the user wants to be notified immediately when the second event happens
upvoted 6 times

tsangckl Most Recent 1 year, 3 months ago
Selected Answer: D
Bing
Option D is correct because it provides a highly available, scalable, and real-time solution. Amazon Kinesis Data Streams can capture and process large streams of data records in real time. AWS Application Auto Scaling can automatically adjust capacity to maintain steady, predictable performance at the lowest possible cost. Amazon Kinesis Data Analytics can process and analyze streaming data using standard SQL, and AWS Lambda can run your code in response to events and automatically manage the compute resources for you.
Other options are not the best solutions for this scenario. For example, Options A and B involve using technologies that may not provide the real-time processing required for this use case. Option C does not provide a mechanism for detecting the known event sequence in real time.
upvoted 2 times

NarenKA 1 year, 4 months ago
Selected Answer: D
I will go with D as KDS is designed to handle massive streams of real-time data and can scale automatically to match the volume of data input and KDA processes of streaming data using SQL and analyse the incoming data stream for patterns, such as the sequence of voltage drops within 10 minutes of a voltage increase. KDA can scale based on demand, supporting the expansion to other cities. AWS Lambda to poll a notification stream and then publish to the Amazon SNS topic for immediate action upon detecting the specified event.
Option A involves managing an Apache Kafka cluster and an EMR cluster, which adds complexity and operational overhead. B uses a REST-based web service and RDS, might not scale as seamlessly and could introduce latency in detecting and notifying about the events. C serverless, does not offer the same level of real-time processing and pattern detection capabilities needed for this specific use case as KDA.
upvoted 4 times

GCPereira 1 year, 5 months ago
kinesis data stream don't support aws auto-scaling, but like everything that happens in the cloud, there exists a way to scale your kds based on lambda and aws auto-scaling... then the D question is not completely wrong...

emr is not highly available, then the A question is killed in this fact...

this problem announces real-time data streaming, but we discard kdf... not because kdf, but because lambda function... in near-real-time data processing using kdf and lambda with a consumer, just the records streamed in 1MB or 60s are processed by lambda...

b makes no sense...

this is a poor question and they should be removed from this dump...
upvoted 2 times

blackgamer 1 year, 6 months ago
The answer is D. It has a requirement of " All notifications must be delivered as quickly as possible.". That requirement rules out Firehose as it can have a delay for buffering and Kinesis data stream is real time.
upvoted 2 times

gofavad926 1 year, 8 months ago
Selected Answer: C
C. Option D mention "Set up AWS Application Auto Scaling on both streams" and this is not possible. Here is not mention that you enable on demand capacity... no, it mentions "AWS Application Auto Scaling". And what about this part? "Create another stream for notifications." It is not needed!
Option C is the correct one
upvoted 2 times

zanhsieh 1 year, 10 months ago
Selected Answer: C
Vote C.
A: No, as EMR is not HA. Single region only.
B: No, RDS is not ideal for streaming.

C: Yes. All systems are fully managed.

D: No. Although there is a solution in 2018 (https://aws.amazon.com/blogs/big-data/scaling-amazon-kinesis-data-streams-with-aws-application-auto-scaling/) but it can't scale more 10 times per 24-hrs nor scale double current shard.

upvoted 3 times

geekfrosty 1 year, 10 months ago

C can't be right, it asks for "real-time", Firehouse collects before it pushes the data out which makes it "near real-time". Answer should be D

upvoted 1 times

NikkyDicky 1 year, 11 months ago

Selected Answer: C

C, because of non-HA EMR

upvoted 2 times

EueChan 2 years ago

Selected Answer: C

EMR is an incorrect answer because it is not high-available

upvoted 2 times

theriderzone 2 years ago

Selected Answer: C

EMR clusters dy default is not highly available.

upvoted 2 times

Debi_mishra 2 years, 1 month ago

Very good question. Both A and C are correct. But A can be ruled out because it uses EMR and out of box its not a highly available service, which is a critical requirement here.

upvoted 2 times

pk349 2 years, 1 month ago

C: I passed the test

upvoted 2 times

uk_dataguy 2 years, 2 months ago

Selected Answer: C

Answer = C

According to https://aws.amazon.com/kinesis/data-firehose/faqs/

It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration.

upvoted 2 times

rich_knp 2 years, 3 months ago

Selected Answer: C

C is correct

upvoted 2 times

srirnag 2 years, 4 months ago

Its A By Elimination.

B is not ideal for streams.

C. Is not immediate. We need real time thing.

D. Way too complex. Not sure if we would need additional Stream, Application auto scalling will scale Lambda. Where did Java come from? I believe there is a hard limit for KDS too.

MSK wins on the scalability front. Spark streaming Application can call SNS SDK. Unbelievable, but it is A.

upvoted 3 times

GCPereira 1 year, 5 months ago

but emr is not high available

upvoted 1 times

Question #25 *Topic 1*

A media company has a streaming playback application. The company needs to collect and analyze data to provide near-real-time feedback on playback issues within 30 seconds. The company requires a consumer application to identify playback issues, such as decreased quality during a specified time frame. The data will be streamed in JSON format. The schema can change over time.
Which solution will meet these requirements?

A. Send the data to Amazon Kinesis Data Firehose with delivery to Amazon S3. Configure an S3 event to invoke an AWS Lambda function to process and analyze the data.

B. Send the data to Amazon Managed Streaming for Apache Kafka. Configure Amazon Kinesis Data Analytics for SQL Application as the consumer application to process and analyze the data.

C. Send the data to Amazon Kinesis Data Firehose with delivery to Amazon S3. Configure Amazon S3 to initiate an event for AWS Lambda to process and analyze the data.

D. Send the data to Amazon Kinesis Data Streams. Configure an Amazon Kinesis Data Analytics for Apache Flink application as the consumer application to process and analyze the data.

Suggested Answer: *B*

Community vote distribution

D (86%) 14%

ay12 Highly Voted 3 years, 2 months ago

Selected Answer: D

https://aws.amazon.com/kinesis/data-analytics/features/?pg=ln&sec=hs

upvoted 12 times

NarenKA Most Recent 1 year, 4 months ago

Selected Answer: D

Options A and C, which involve Kinesis Data Firehose with delivery to Amazon S3 and subsequent processing by AWS Lambda, are not optimized for near-real-time feedback due to the inherent latency in delivering data to S3 and then processing it. Option B, involving Amazon Managed Streaming for Apache Kafka and Kinesis Data Analytics for SQL, could also be a viable solution for real-time analytics, but the specific choice of Apache Flink in Option D is more directly aligned with the company's need for complex event processing and near-real-time analysis capabilities.

upvoted 1 times

gofavad926 1 year, 8 months ago

Selected Answer: D

D. Amazon Managed Service for Apache Flink

upvoted 1 times

chinmayj213 1 year, 9 months ago

A & C cannot be option as it uses firehose, which has min record buffered time as 60 second / 1 minute and here we have to do processing in 30 second. So we left with B and D

upvoted 1 times

chinmayj213 1 year, 9 months ago

D seems correct as KDA using flink for processing (sql style)

upvoted 1 times

penguins2 1 year, 11 months ago

D. AWS KDA doesn't take input from Kafka, only if KDA is provisioned using Flink, then Kafka can be a source.

upvoted 1 times

pk349 2 years, 1 month ago

D: I passed the test

upvoted 1 times

roymunson 1 year, 7 months ago

Was the "test" about typing the same comment in every single discussion to show people how you realy wasting your time with stupid actions on the internet?

upvoted 5 times

AwsNewPeople 2 years, 3 months ago

Option B is the most suitable solution as it allows the company to stream the data in real-time to Amazon Managed Streaming for Apache Kafka. Then, the data can be processed and analyzed using Amazon Kinesis Data Analytics for SQL Application, which can handle data in JSON format with dynamic schema changes. This solution allows the company to identify playback issues in near-real-time within 30 seconds.

Option A is not optimal because it requires an S3 event to trigger the Lambda function, which may introduce some latency. Option C is similar to option A, but with an additional step of writing the data to S3, which may not be necessary. Option D uses Apache Flink, which may be overkill for this use case and can be more complex to set up compared to the SQL-based Kinesis Data Analytics application in option B.

upvoted 1 times

AwsNewPeople 2 years, 3 months ago

But since the question never ask for Cost optimisation, I will go with D

upvoted 2 times

ota123 2 years, 5 months ago

Selected Answer: B

According to https://aws.amazon.com/kinesis/data-analytics/faqs/

Some keywords are "json formatted data", "schema update". These are what Kinesis Data Analytics SQL applications do. And Kinesis Data Analytics can also analytics/?nc=sn&loc=1

upvoted 1 times

nadavw 2 years, 5 months ago

The preferred service may be the Kinesis Video stream, which is intended for playback. The problem with D is the message size limit in KDS (1MB), while

https://aws.amazon.com/kinesis/video-streams/?nc=sn&loc=0&amazon-kinesis-video-streams-resources-blog.sort-by=item.additionalFields.createdDate order=desc#:~:text=Amazon%20Kinesis%20Video%20Streams%20makes%20it%20easy%20to%20securely%20stream%20video%20from%20connected%2

upvoted 1 times

rav009 2 years, 8 months ago

D

Firehose can be ruled out for it has 60 sec data latency.

KDA for SQL cannot support MSK as source.

upvoted 4 times

Arka_01 2 years, 9 months ago

Selected Answer: D

As the allowed time offset is of 30 seconds, we can eliminate options with fireshose. Kafka is third party, and not a preferred answer.

upvoted 2 times

JoellaLi 2 years, 8 months ago

This is not the reason for not selecting B, since 'Amazon Managed Streaming for Apache Kafka(MSK)' is a AWS service not third party.

upvoted 2 times

Sen5476 2 years, 12 months ago

Ans is D.

Option A & C - Firehose and its minimum buffer time is 60 sec

Option B - Will work, But JSON schema changes over time. Kinesis Data Analytics requires manual intervention to update column mapping if input changes.

upvoted 4 times

Bik000 3 years, 1 month ago

Selected Answer: D

Answer is D

upvoted 1 times

certificationJunkie 3 years, 1 month ago

How does Managed Kafka and Kinesis Analytics even interact with each other? Correct ans is D.

upvoted 1 times

MWL 3 years, 1 month ago

Selected Answer: D

Changed to D.

KDA for SQL can not set MSK as source. KDF can not process in 30 seconds.

upvoted 2 times

MWL 3 years, 1 month ago

Selected Answer: B

Vote for B.

According to https://aws.amazon.com/kinesis/data-analytics/faqs/

Some keywords are "json formatted data", "schema update". These are what Kinesis Data Analytics SQL applications do. And Kinesis Data Analytics can also get data from MSK.

upvoted 2 times

MWL 3 years, 1 month ago

Change to D, KDA for SQL can not set MSK as source. KDA for Flink can.

upvoted 3 times

shammous 2 years, 6 months ago

Also, data is in JSON format, and the schema can change. SQL is useless in this case

upvoted 1 times

jrheen 3 years, 2 months ago

D- looks good

upvoted 1 times

Question #26 *Topic 1*

An ecommerce company stores customer purchase data in Amazon RDS. The company wants a solution to store and analyze historical data. The most recent 6 months of data will be queried frequently for analytics workloads. This data is several terabytes large. Once a month, historical data for the last 5 years must be accessible and will be joined with the more recent data. The company wants to optimize performance and cost.
Which storage solution will meet these requirements?

A. Create a read replica of the RDS database to store the most recent 6 months of data. Copy the historical data into Amazon S3. Create an AWS Glue Data Catalog of the data in Amazon S3 and Amazon RDS. Run historical queries using Amazon Athena.

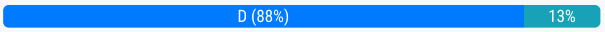
B. Use an ETL tool to incrementally load the most recent 6 months of data into an Amazon Redshift cluster. Run more frequent queries against this cluster. Create a read replica of the RDS database to run queries on the historical data.

C. Incrementally copy data from Amazon RDS to Amazon S3. Create an AWS Glue Data Catalog of the data in Amazon S3. Use Amazon Athena to query the data.

D. Incrementally copy data from Amazon RDS to Amazon S3. Load and store the most recent 6 months of data in Amazon Redshift. Configure an Amazon Redshift Spectrum table to connect to all historical data.

Suggested Answer: *D*

Community vote distribution



abhineet Highly Voted 3 years, 9 months ago
D seems correct
upvoted 20 times

Shraddha Highly Voted 3 years, 8 months ago
Ans D
Note: A and B are immediately out because RDS is not for analysis. C and D both work, but D is balanced between performance and cost. C may cost less (depending on data compression, frequency of queries) but query to recent data will be slower.
upvoted 17 times

Donell 3 years, 8 months ago
Correct, answer is D.
Redshift is suitable for running complex analytical queries. Athena is suitable for small ad-hoc queries.
upvoted 6 times

GCPereira Most Recent 1 year, 5 months ago
when we say "analytics queries", "analytics workloads" or "complex analysis", in 80% of the cases we call redshift... if we sum a short period of analysis (6 months in this case) redshift is a better option... rds will continue as a relational database, don't run analytics queries
upvoted 1 times

pk349 2 years, 1 month ago
D: I passed the test
upvoted 2 times

Espa 2 years, 1 month ago
I see you have been posting only I passed the test :)
upvoted 5 times

DipeshGandhi131 2 years ago
hahah!!
upvoted 2 times

AwsNewPeople 2 years, 3 months ago
Selected Answer: D
Option D is the most suitable solution for this scenario.

Explanation:

Incrementally copy data from Amazon RDS to Amazon S3: This allows for storing of historical data in a cost-effective manner while allowing frequent

querying of the more recent data in RDS.
Load and store the most recent 6 months of data in Amazon Redshift: This provides a performant solution for frequent queries of the most recent data.
Configure an Amazon Redshift Spectrum table to connect to all historical data: This enables joining of historical data with the more recent data in Redshift, providing the required analysis capability.
Option A does not address the requirement to optimize performance for querying the most recent data. Option B involves creating a read replica of RDS, which may not be efficient for frequently queried data. Option C also does not provide a solution for frequent querying of the most recent data.
upvoted 2 times

rags1482 2 years, 4 months ago
Option D suggests copying data from RDS to S3 incrementally, storing the most recent 6 months of data in Amazon Redshift, and configuring an Amazon Redshift Spectrum table to connect to all historical data. This approach allows the company to optimize cost and performance as Redshift is a cost-effective data warehousing solution that can handle large volumes of data. Additionally, using Redshift Spectrum enables the company to query both the recent and historical data sets together in real-time.

Option A suggests creating a read replica of the RDS database to store the most recent 6 months of data and copying the historical data into Amazon S3. This approach does not allow for real-time querying of the historical data and may result in increased query latency.
upvoted 1 times

murali12180 2 years, 4 months ago
Selected Answer: A
A. By moving the data to S3 and Glue Catalog that carries both RDS and S3 schema will enable them to use the same schema for queries. Remember the requirement says "low cost". Redshift is out of the picture.
upvoted 1 times

aws_kid 2 years, 3 months ago
I don't think read replicas for certain months can be created. Read replicas will replicate entire db. Unlikely A is the answer
upvoted 1 times

BtotheJ 2 years, 4 months ago
Selected Answer: D
D for the win
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
D is the right answer as loading and querying recent 6 months of data via Redshift gives better performance and old data can be queried via Redshift spectrum
C is wrong though it's possible to query the entire data in S3 using Athena, however, it will not be able to match the high performance offered by Redshift to query the last six months of data. So this option is not the best fit for the given use case.

Options A & B are wrong as RDS is not an ideal solution to store and query historical data. Also, 6 months data may be several terabytes large.
upvoted 3 times

aefuen1 2 years, 8 months ago
Selected Answer: D
D seems correct
upvoted 1 times

rocky48 2 years, 11 months ago
Selected Answer: D
Answer-D
upvoted 1 times

jrheen 3 years, 2 months ago
Answer-D
upvoted 1 times

simonaque 3 years, 3 months ago
Selected Answer: D
D seems correct
upvoted 1 times

ShilaP 3 years, 3 months ago
D is correct...

upvoted 1 times

Agn3001 3 years, 3 months ago

Selected Answer: D

effective way to query across S3 and RDS is using redshift spectrum

upvoted 1 times

umatrilok 3 years, 6 months ago

Historical Data points to Redshift Spectrum. Hence D

upvoted 1 times

aws2019 3 years, 7 months ago

answer is D.

upvoted 1 times

Question #27 *Topic 1*

A company leverages Amazon Athena for ad-hoc queries against data stored in Amazon S3. The company wants to implement additional controls to separate query execution and query history among users, teams, or applications running in the same AWS account to comply with internal security policies.
Which solution meets these requirements?

A. Create an S3 bucket for each given use case, create an S3 bucket policy that grants permissions to appropriate individual IAM users. and apply the S3 bucket policy to the S3 bucket.

B. Create an Athena workgroup for each given use case, apply tags to the workgroup, and create an IAM policy using the tags to apply appropriate permissions to the workgroup.

C. Create an IAM role for each given use case, assign appropriate permissions to the role for the given use case, and add the role to associate the role with Athena.

D. Create an AWS Glue Data Catalog resource policy for each given use case that grants permissions to appropriate individual IAM users, and apply the resource policy to the specific tables used by Athena.

Suggested Answer: *C*
Reference:
https://aws.amazon.com/athena/faqs/

Community vote distribution
B (100%)

Priyanka_01 Highly Voted 3 years, 9 months ago
B any thoughts?
https://docs.aws.amazon.com/athena/latest/ug/user-created-workgroups.html
upvoted 29 times

jersyl Highly Voted 3 years, 9 months ago
I think it's B. based on this link:
https://aws.amazon.com/about-aws/whats-new/2019/02/athena_workgroups/
upvoted 10 times

NarenKA Most Recent 1 year, 4 months ago
Selected Answer: B
Correct answer is B - creating Athena workgroups for each use case, tagging those workgroups, and applying IAM policies based on those tags is the most effective way to meet the company's security and compliance requirements.

Option A focuses on S3 bucket policies, which do not directly address the separation of query execution within Athena.
Option C involves creating IAM roles for use cases but does not inherently separate query execution and history within Athena itself.
Option D pertains to AWS Glue Data Catalog resource policies, which, while important for controlling access to data, do not directly manage the separation of Athena query execution and history.
upvoted 1 times

pk349 2 years, 1 month ago
B: I passed the test
upvoted 2 times

AwsNewPeople 2 years, 3 months ago
Selected Answer: B
The solution that meets the requirements to separate query execution and query history among users, teams, or applications running in the same AWS account is to create an Athena workgroup for each given use case, apply tags to the workgroup, and create an IAM policy using the tags to apply appropriate permissions to the workgroup. This allows the company to control access to specific workgroups and apply different permissions to different groups. Option B is therefore the correct answer.

Option A is not a suitable solution as creating S3 buckets for each use case would not effectively control access to Athena queries and history.

Option C is not a suitable solution as creating an IAM role for each use case would not allow for granular control over permissions and would not

effectively separate query execution and history.

Option D is not a suitable solution as creating an AWS Glue Data Catalog resource policy would not effectively separate query execution and history within Athena.

upvoted 3 times

aws_kid 2 years, 3 months ago

Does this site deliberately provide wrong answer choice?

upvoted 4 times

cloudlearnerhere 2 years, 7 months ago

Correct answer is B as Athena Workgroup can help comply with the internal security policies by separating execution for users, teams, applications and applying access control and auditing.

upvoted 4 times

lordpizzo 2 years, 11 months ago

Selected Answer: B

without a doubt, the best way to guarantee isolation and better control over history, cost and the lowest possible level of access is to create workgroups for athena. Letter B without a doubt!

upvoted 1 times

msa11a 2 years, 11 months ago

Selected Answer: B

Athena group

upvoted 1 times

awsexpert69 2 years, 11 months ago

Selected Answer: B

B is correct

upvoted 1 times

bp339 2 years, 11 months ago

Selected Answer: B

Athena Workgroups

upvoted 1 times

abdelawwal 3 years ago

B

based on that link:

https://aws.amazon.com/about-aws/whats-new/2019/02/athena_workgroups/

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer should be B

upvoted 1 times

jrheen 3 years, 2 months ago

B is correct

upvoted 1 times

jrheen 3 years, 3 months ago

Selected Answer: B

B is correct

upvoted 3 times

pidkiller 3 years, 3 months ago

I think B is the correct answer

upvoted 1 times

PravinT 3 years, 4 months ago

B is the right answer

upvoted 1 times

Question #28 *Topic 1*

A company wants to use an automatic machine learning (ML) Random Cut Forest (RCF) algorithm to visualize complex real-world scenarios, such as detecting seasonality and trends, excluding outers, and imputing missing values.
The team working on this project is non-technical and is looking for an out-of-the-box solution that will require the LEAST amount of management overhead.
Which solution will meet these requirements?

A. Use an AWS Glue ML transform to create a forecast and then use Amazon QuickSight to visualize the data.

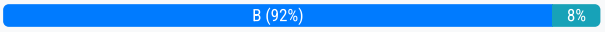
B. Use Amazon QuickSight to visualize the data and then use ML-powered forecasting to forecast the key business metrics.

C. Use a pre-build ML AMI from the AWS Marketplace to create forecasts and then use Amazon QuickSight to visualize the data.

D. Use calculated fields to create a new forecast and then use Amazon QuickSight to visualize the data.

Suggested Answer: *A*
Reference:
https://aws.amazon.com/blogs/big-data/query-visualize-and-forecast-trufactor-web-session-intelligence-with-aws-data-exchange/

Community vote distribution



jersyl Highly Voted 3 years, 9 months ago
It is B based on this link:
https://docs.aws.amazon.com/quicksight/latest/user/making-data-driven-decisions-with-ml-in-quicksight.html
upvoted 26 times

attaraya 3 years, 7 months ago
Agreed: more reference
https://docs.aws.amazon.com/quicksight/latest/user/how-does-rcf-generate-forecasts.html
upvoted 1 times

Ali_Hussein Most Recent 1 year, 10 months ago
Selected Answer: C
The correct answer is C.

Here is the explanation:

Use a pre-build ML AMI from the AWS Marketplace to create forecasts and then use Amazon QuickSight to visualize the data. This is the most out-of-the-box solution and will require the least amount of management overhead.
AWS Glue ML transforms are a great way to automate ML tasks, but they require some technical expertise to set up and use.
Amazon QuickSight is a great visualization tool, but it does not have built-in ML capabilities.
Calculated fields are a way to create new fields in a data set, but they cannot be used to create forecasts.
upvoted 1 times

MLCL 1 year, 11 months ago
Selected Answer: B
Quicksight supports RCF and can connect easily to multiple data sources (S3, JDBC ..etc)
upvoted 1 times

Espa 2 years, 1 month ago
Selected Answer: B
Quicksight uses a built-in version of RCF
upvoted 1 times

pk349 2 years, 1 month ago
B: I passed the test
upvoted 1 times

okrasheno 1 year, 1 month ago
Dude is a legend now

upvoted 1 times

yazquez 1 year, 6 months ago

It's very nice to know!! No one noticed

upvoted 1 times

uk_dataguy 2 years, 2 months ago

this scenario is shocking to see because the team is all non-technical ....

upvoted 2 times

AwsNewPeople 2 years, 3 months ago

Selected Answer: B

Both options B and C involve using Amazon QuickSight to visualize the data. However, option C involves using a pre-built machine learning Amazon Machine Image (AMI) from the AWS Marketplace to create forecasts, which may require more technical expertise to set up and manage than option B, which simply involves using ML-powered forecasting within Amazon QuickSight. Therefore, option B may be more suitable for a non-technical team looking for an out-of-the-box solution with minimal management overhead.

upvoted 1 times

rags140882 2 years, 4 months ago

Amazon QuickSight uses a built-in version of the Random Cut Forest (RCF) algorithm.

B is correct

upvoted 1 times

rags140882 2 years, 4 months ago

Option C is the best solution for this scenario. The company wants an out-of-the-box solution that requires the least amount of management overhead, and using a pre-built ML AMI from the AWS Marketplace to create forecasts and then using Amazon QuickSight to visualize the data is the most straightforward approach. The pre-built ML AMI will provide the Random Cut Forest algorithm for the team to use, and Amazon QuickSight provides an easy-to-use interface for data visualization. This solution will require minimal technical expertise and management overhead from the non-technical team.

Option B is not the best solution as using ML-powered forecasting in Amazon QuickSight does not provide the Random Cut Forest algorithm that the company wants to use.

upvoted 1 times

AwsNewPeople 2 years, 3 months ago

Both options B and C involve using Amazon QuickSight to visualize the data. However, option C involves using a pre-built machine learning Amazon Machine Image (AMI) from the AWS Marketplace to create forecasts, which may require more technical expertise to set up and manage than option B, which simply involves using ML-powered forecasting within Amazon QuickSight. Therefore, option B may be more suitable for a non-technical team looking for an out-of-the-box solution with minimal management overhead.

upvoted 1 times

renfdo 2 years, 6 months ago

Selected Answer: B

B, Quicksigth has many ML tools.

upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Correct answer is B as QuickSight ML provides an out-of-the-box ML Random Cut Forest (RCF) algorithm to help visualize complex real-world scenarios, such as detecting seasonality and trends, excluding outliers, and imputing missing values.

Options A, C & D are wrong as they do not come with the least operational overhead.

upvoted 2 times

nharaz 2 years, 8 months ago

B is correct.
Amazon QuickSight uses a built-in version of the Random Cut Forest (RCF) algorithm. The following sections explain what that means and how it is used in Amazon QuickSight.

First, let's look at some of the terminology involved:

Anomaly – Something that is characterized by its difference from the majority of the other things in the same sample. Also known as an outlier, an exception, a deviation, and so on.

Data point – A discrete unit—or simply put, a row—in a dataset. However, a row can have multiple data points if you use a measure over different dimensions.

Decision Tree – A way of visualizing the decision process of the algorithm that evaluates patterns in the data.

Forecast – A prediction of future behavior based on current and past behavior.

Model – A mathematical representation of the algorithm or what the algorithm learns.

Seasonality – The repeating patterns of behavior that occur cyclically in time series data.

Time series – An ordered set of date or time data in one field or column.

upvoted 3 times

thirukudil 2 years, 8 months ago

Selected Answer: B

B. Amazon QuickSight enables nontechnical users to confidently forecast their key business metrics. The built-in ML Random Cut Forest algorithm automatically handles complex real-world scenarios such as detecting seasonality and trends, excluding outliers, and imputing missing values. You can interact with the data with point-and-click simplicity.

upvoted 2 times

muhsin 2 years, 10 months ago

it is B

https://docs.aws.amazon.com/quicksight/latest/user/concept-of-ml-algorithms.html

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: B

B is the answer.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer should be B

upvoted 1 times

MWL 3 years, 1 month ago

Selected Answer: B

Aggree with B.

But one problem is, the question mentions about "inputing missing data". Can quicksight handle missing data for ML related visualization?

upvoted 1 times

JoellaLi 2 years, 8 months ago

Yes of course. "The built-in ML Random Cut Forest algorithm automatically handles complex real-world scenarios such as detecting seasonality and trends, excluding outliers, and imputing missing values. You can interact with the data with point-and-click simplicity."

Link: https://docs.aws.amazon.com/quicksight/latest/user/making-data-driven-decisions-with-ml-in-quicksight.html

upvoted 1 times

Question #29 *Topic 1*

A retail company's data analytics team recently created multiple product sales analysis dashboards for the average selling price per product using Amazon
QuickSight. The dashboards were created from .csv files uploaded to Amazon S3. The team is now planning to share the dashboards with the respective external product owners by creating individual users in Amazon QuickSight. For compliance and governance reasons, restricting access is a key requirement. The product owners should view only their respective product analysis in the dashboard reports.
Which approach should the data analytics team take to allow product owners to view only their products in the dashboard?

A. Separate the data by product and use S3 bucket policies for authorization.

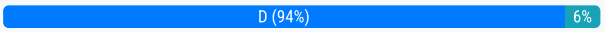
B. Separate the data by product and use IAM policies for authorization.

C. Create a manifest file with row-level security.

D. Create dataset rules with row-level security.

Suggested Answer: *B*

Community vote distribution



Priyanka_01 Highly Voted 3 years, 9 months ago
D any thoughts?
https://docs.aws.amazon.com/quicksight/latest/user/restrict-access-to-a-data-set-using-row-level-security.html
upvoted 37 times

lakediver 3 years, 6 months ago
Agree
Quick tip - row level security only available in Enterprise Edition
upvoted 3 times

GCPereira 1 year, 5 months ago
agree, this question appearedin my test
upvoted 2 times

Shraddha Highly Voted 3 years, 7 months ago
Ans D
This is a textbook question.

https://docs.aws.amazon.com/quicksight/latest/user/restrict-access-to-a-data-set-using-row-level-security.html
upvoted 7 times

NarenKA Most Recent 1 year, 4 months ago
Selected Answer: D
Options A and B involve managing access at the S3 level, which does not directly apply to controlling visibility within QuickSight dashboards.
Option C, creating a manifest file, is part of how you might structure data for QuickSight, but it does not directly address row-level security within QuickSight itself.

Option D is correct - creating dataset rules with row-level security within Amazon QuickSight is the most effective and governance-compliant method to ensure that product owners have access only to their respective product analysis in the dashboard reports.
upvoted 1 times

pk349 2 years, 1 month ago
D: I passed the test
upvoted 2 times

kondi2309 1 year, 4 months ago
agree with no doubt
upvoted 1 times

uk_dataguy 2 years, 2 months ago
Selected Answer: D

D without any doubt.
https://docs.aws.amazon.com/quicksight/latest/user/restrict-access-to-a-data-set-using-row-level-security.html
upvoted 1 times

AwsNewPeople 2 years, 3 months ago

Selected Answer: D

D. Create dataset rules with row-level security would be the best approach for this use case. Row-level security (RLS) allows you to define filters at the row level, which can be used to control access to specific data based on user attributes or permissions. This would enable the data analytics team to ensure that each product owner can only see the data for their respective products, as the filters would be applied to the dashboard data before it is displayed to the user. Option A and B would not provide the necessary granularity to restrict access to specific data based on the product owners, and option C would not be applicable in this case as it is used for securing data in Amazon Redshift clusters.
upvoted 4 times

[Removed] 2 years, 2 months ago

Hey man thanks for you answers across threads really helped me a lot, could you let us know if you have taken the exam or any tips & tricks
upvoted 1 times

dhyuk 2 years, 7 months ago

i think D
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: D

Correct answer is D as dataset rules with row-level security can be used to restrict the data the product owners would see, which is based on the product.

Options A & B are wrong as they would not provide fine-grained access control and would need extra effort.

Option C is wrong as the row-level security rules need to be defined in the dataset and not in the manifest file.
upvoted 3 times

nharaz 2 years, 8 months ago

D is correct
In the Enterprise edition of Amazon QuickSight, you can restrict access to a dataset by configuring row-level security (RLS) on it. You can do this before or after you have shared the dataset. When you share a dataset with RLS with dataset owners, they can still see all the data. When you share it with readers, however, they can only see the data restricted by the permission dataset rules. By adding row-level security, you can further control their access.
upvoted 3 times

thirukudil 2 years, 8 months ago

Selected Answer: D

In the Enterprise edition of Amazon QuickSight, you can restrict access to a dataset by configuring row-level security (RLS) on it. You can do this before or after you have shared the dataset. When you share a dataset with RLS with dataset owners, they can still see all the data. When you share it with readers, however, they can only see the data restricted by the permission dataset rules. By adding row-level security, you can further control their access.
upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: D

It should be done through dataset rules
upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: D

Answer-D
upvoted 2 times

rocky48 2 years, 9 months ago

https://docs.aws.amazon.com/quicksight/latest/user/restrict-access-to-a-data-set-using-row-level-security.html
upvoted 1 times

Ahamedkabir 3 years ago

Selected Answer: B

For sure it is right answer

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: D

Answer should be D

upvoted 1 times

certificationJunkie 3 years, 1 month ago

A is correct answer. Note that there are various dashboards for different products. And access needs to be provisioned to the product owners for respective products. Hence, ideal way is to create an IAM user for each product owner and use that user to access quickSight dashboard.

upvoted 1 times

jrheen 3 years, 2 months ago

Answer-D

upvoted 2 times

Teraxs 3 years, 2 months ago

Selected Answer: D

as mentions by others https://docs.aws.amazon.com/quicksight/latest/user/restrict-access-to-a-data-set-using-row-level-security.html

upvoted 2 times

Question #30 *Topic 1*

A company has developed an Apache Hive script to batch process data stared in Amazon S3. The script needs to run once every day and store the output in
Amazon S3. The company tested the script, and it completes within 30 minutes on a small local three-node cluster.
Which solution is the MOST cost-effective for scheduling and executing the script?

A. Create an AWS Lambda function to spin up an Amazon EMR cluster with a Hive execution step. Set KeepJobFlowAliveWhenNoSteps to false and disable the termination protection flag. Use Amazon CloudWatch Events to schedule the Lambda function to run daily.

B. Use the AWS Management Console to spin up an Amazon EMR cluster with Python Hue. Hive, and Apache Oozie. Set the termination protection flag to true and use Spot Instances for the core nodes of the cluster. Configure an Oozie workflow in the cluster to invoke the Hive script daily.

C. Create an AWS Glue job with the Hive script to perform the batch operation. Configure the job to run once a day using a time-based schedule.

D. Use AWS Lambda layers and load the Hive runtime to AWS Lambda and copy the Hive script. Schedule the Lambda function to run daily by creating a workflow using AWS Step Functions.

Suggested Answer: *C*

Community vote distribution

A (71%) C (29%)

carol1522 Highly Voted 3 years, 9 months ago

For me it is A. Not B because we are not supposed to run core nodes in spot instances, just task nodes and it is more expensive because to schedule with oozie, our cluster have to be up all the time. It is not C because glue cannot run hive script, and it is not c because lambda cannot run hive scripts also. https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/RunLambdaSchedule.html

upvoted 44 times

awssp12345 3 years, 9 months ago

Agree with A

upvoted 2 times

Prodip 3 years, 9 months ago

Perfect Explanation ; I wanted to write something but your text covers everything.

upvoted 2 times

chengxu32 3 years, 7 months ago

https://docs.aws.amazon.com/emr/latest/APIReference/API_RunJobFlow.html

With KeepJobFlowAliveWhenNoSteps parameter is set to False, the cluster will be shutdown once the steps are completed, thus the cost effective requirement is met

upvoted 9 times

jove Highly Voted 3 years, 8 months ago

+ A is the correct answer.

- B : Spot Instances are not a good option to run a 30-min-script

- C: Glue cannot run Hive scripts

- D: Lambda can run for 15 minutes maximum. Not enough time to run that script.

upvoted 8 times

Bob888 2 years, 2 months ago

C: Glue cannot run Hive scripts---> Clue can run hive scripts. But the problem is that C keep all the Glue setting and does not terminate it.

A By default, an Amazon EMR cluster will be terminated automatically when all steps have completed and there are no pending steps or other applications running on the cluster.

upvoted 4 times

tsangckl Most Recent 1 year, 3 months ago

Selected Answer: C

Bing

Option C is correct because AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy to prepare and load your data

for analytics. You can create a job in AWS Glue that incorporates your Hive script, and you can schedule this job to run once a day. This approach does not require the provisioning or management of servers, making it a cost-effective solution.
Other options involve using Amazon EMR or AWS Lambda, which could incur higher costs due to the need for server provisioning and potential for idle resources.
upvoted 1 times

gofavad926 1 year, 8 months ago

Selected Answer: A

A. As carol1522 explains in her comment
upvoted 1 times

Hamza98 1 year, 8 months ago

Selected Answer: A

A satisfies all the requirements
upvoted 1 times

rlnd2000 1 year, 9 months ago

Selected Answer: C

"Could anyone who chose "A" as the correct answer please explain how to make a Lambda function run for 30 minutes?"
upvoted 1 times

gofavad926 1 year, 8 months ago

The lambda function only create and initialise the EMR...
upvoted 2 times

petervu 2 years ago

Selected Answer: C

Since AWS Glue can run Hive script. So C will be cheaper than A.
upvoted 3 times

Venkkat 2 years ago

A for sure
upvoted 1 times

pk349 2 years, 1 month ago

A: I passed the test
upvoted 1 times

rlnd2000 1 year, 9 months ago

You passed, OK but this question was wrong in your test I think, how can you make lambda run for 30 minutes?
upvoted 1 times

uyendo123 1 year, 9 months ago

I guess that the A means Lambda function would just spin up the EMR Cluster, when Cluster has started, the Lambda function would stop.
Then the Hive script run on EMR Cluster, and terminated when script running done.
upvoted 1 times

Arjun777 2 years, 4 months ago

aws glue can run hive scripts - https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive-metastore-glue.html
upvoted 4 times

he11ow0rld 1 year, 10 months ago

I think you misunderstand this blog. hive can use the catalog generated by glue, but glue running hive script.
So, C is still wrong, A is the correct answer
upvoted 3 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A as the EMR cluster can be used to execute the Hive scripts. KeepJobFlowAliveWhenNoSteps set to false and disabling the termination protection flag would help destroy the cluster once no running jobs. CloudWatch Events with Lambda can be used to trigger the scheduled activity.

Option B is wrong as Oozie requires the EMR cluster always running, else the job cannot be scheduled and executed. Using Spot instances for core nodes is not recommended.

Option C is wrong as Glue does not support running Hive scripts.

Option D is wrong as Lambda would not be able to meet the 30 minutes job runtime requirement.

upvoted 7 times

Arka_01 2 years, 9 months ago

Selected Answer: A

This one is a classic scenario of Transient cluster. So A is the answer here.

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: A

Answer is A

upvoted 1 times

Pradhan 3 years, 7 months ago

I will go with A.

upvoted 3 times

Shraddha 3 years, 7 months ago

Ans A

B = wrong, termination flag should be off, spot instances not good for core nodes. C = Glue runs on Spark, cannot run Hive scripts. D = wrong, Lambda maximum running time 15 minutes.

upvoted 4 times

lostsoul07 3 years, 7 months ago

A is the right answer

upvoted 2 times

Sai12 3 years, 8 months ago

A based on its similarity to this article https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs-process-sample-data.html

upvoted 2 times

Question #31 *Topic 1*

A company wants to improve the data load time of a sales data dashboard. Data has been collected as .csv files and stored within an Amazon S3 bucket that is partitioned by date. The data is then loaded to an Amazon Redshift data warehouse for frequent analysis. The data volume is up to 500 GB per day.
Which solution will improve the data loading performance?

A. Compress .csv files and use an INSERT statement to ingest data into Amazon Redshift.

B. Split large .csv files, then use a COPY command to load data into Amazon Redshift.

C. Use Amazon Kinesis Data Firehose to ingest data into Amazon Redshift.

D. Load the .csv files in an unsorted key order and vacuum the table in Amazon Redshift.

Suggested Answer: *C*
Reference:
https://aws.amazon.com/blogs/big-data/using-amazon-redshift-spectrum-amazon-athena-and-aws-glue-with-node-js-in-production/

Community vote distribution
B (100%)

Paitan Highly Voted 3 years, 9 months ago
B for sure.
The COPY command loads the data in parallel from multiple files, dividing the workload among the nodes in your cluster. When you load all the data from a single large file, Amazon Redshift is forced to perform a serialized load, which is much slower. Split your load data files so that the files are about equal size, between 1 MB and 1 GB after compression. For optimum parallelism, the ideal size is between 1 MB and 125 MB after compression. The number of files should be a multiple of the number of slices in your cluster
upvoted 44 times

Shraddha Highly Voted 3 years, 7 months ago
Ans B
A = wrong, compression means download file from S3 then compress, time-consuming, also you use COPY for files not INSERT. C = wrong, will not improve performance. D = wrong, vacuum frees up storage. This is a question about parallel loading.
upvoted 11 times

Mayank7g Most Recent 1 year, 11 months ago
Selected Answer: B
B for sure
upvoted 1 times

pk349 2 years, 1 month ago
B: I passed the test
upvoted 2 times

AwsNewPeople 2 years, 3 months ago
Selected Answer: B
Option B is the most appropriate solution for improving data loading performance. Splitting large .csv files and using a COPY command can parallelize the load process and reduce the data load time. The data partitioning by date can help further optimize the load process by reducing the data scanned for each load. Compressing the .csv files may help reduce the storage cost, but it may not improve the data load time. Using an INSERT statement to ingest data into Amazon Redshift can be slow and does not take advantage of Redshift's parallel processing capability. Amazon Kinesis Data Firehose can be used to ingest streaming data in real-time, but may not be the best choice for large batch loads. Loading the .csv files in an unsorted key order and vacuuming the table can help optimize the table for query performance but may not improve the data loading performance.
upvoted 3 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: B
Correct answer is B as splitting the large file into multiple files can help improve the data loading performance using the COPY command.

Option A is wrong as the COPY command would provide the best benefit.

Option C is wrong as Kinesis Data Firehose cannot move data from S3 to Redshift. Kinesis Data Firehose delivers your data to your S3 bucket first and

then issues an Amazon Redshift COPY command to load the data into your Amazon Redshift cluster. So it doesn't still improve the load performance.

Option D is wrong as vacuuming will free up space but does not improve the load performance.

upvoted 5 times

Arka_01 2 years, 9 months ago

Selected Answer: B

S3 to Redshift upload can be done through copy command. To utilize parallelism, large files are recommended to split into small chunk of files.

upvoted 1 times

Binh12 2 years, 11 months ago

Don't know why B? here is uncompressed csv file, so no need to split file (Redshift will do automatically?
In contrast, when you load delimited data from a large, uncompressed file, Amazon Redshift makes use of multiple slices. These slices work in parallel, automatically. This provides fast load performance.
In https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-use-multiple-files.html

upvoted 2 times

Ryo0w0o 2 years, 7 months ago

Agreed. It seems like no correct answer among the choices.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: B

Answer = B

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer is B

upvoted 1 times

moon2351 3 years, 3 months ago

Selected Answer: B

Answer is B

upvoted 2 times

awsmani 3 years, 7 months ago

Ans:B split large files will help loading in performance. Having one large file will load in serialized manner which lowers performance

upvoted 1 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 3 times

BillyC 3 years, 8 months ago

B is correct for me

upvoted 2 times

sanjaym 3 years, 8 months ago

B for sure.

upvoted 2 times

syu31svc 3 years, 9 months ago

It's already in S3 so answer is B 100%

upvoted 1 times

ali_baba_acs 3 years, 9 months ago

Answer is B, A compress is a good practice but then copy command not insert, Kinesis Firehose will not improve performance, the vacuum will help freeing space not improve perf too.

upvoted 3 times

Question #32 *Topic 1*

A company has a data warehouse in Amazon Redshift that is approximately 500 TB in size. New data is imported every few hours and read-only queries are run throughout the day and evening. There is a particularly heavy load with no writes for several hours each morning on business days. During those hours, some queries are queued and take a long time to execute. The company needs to optimize query execution and avoid any downtime.
What is the MOST cost-effective solution?

A. Enable concurrency scaling in the workload management (WLM) queue.

B. Add more nodes using the AWS Management Console during peak hours. Set the distribution style to ALL.

C. Use elastic resize to quickly add nodes during peak times. Remove the nodes when they are not needed.

D. Use a snapshot, restore, and resize operation. Switch to the new target cluster.

Suggested Answer: *A*

Community vote distribution

A (100%)

awssp12345 Highly Voted 3 years, 9 months ago

Answer A is correct- https://docs.aws.amazon.com/redshift/latest/dg/cm-c-implementing-workload-management.html

upvoted 28 times

Thiya Highly Voted 3 years, 7 months ago

Answer: A

WLM Concurrency scaling feature automatically adds additional capacity for both read and write queries and charged only for the duration the queries are actively running. Hence, it is the cost-effective approach. https://docs.aws.amazon.com/redshift/latest/dg/concurrency-scaling.html

upvoted 10 times

GCPereira Most Recent 1 year, 5 months ago

Easy question, A any througs?

Through workload management, you can prioritize queries and define execution patterns across your cluster...

resizing, snapshot, or other solutions in these answers is a bit expansive and ineffective.

upvoted 2 times

pk349 2 years, 1 month ago

A: I passed the test

upvoted 4 times

AwsNewPeople 2 years, 3 months ago

Selected Answer: A

A. Enable concurrency scaling in the workload management (WLM) queue is the most cost-effective solution.

Enabling concurrency scaling in the workload management (WLM) queue allows Amazon Redshift to add more cluster capacity to handle the increased query load during peak hours. This is done automatically and can be configured based on the number of users or the number of queries. Concurrency scaling can be turned off during off-peak hours to save costs. This is a more cost-effective solution compared to adding more nodes or using elastic resize, which can be more expensive and take longer to configure. Snapshot, restore, and resize operations can also be time-consuming and may result in downtime.

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A as Redshift Concurrency Scaling can help scale the cluster to support virtually unlimited concurrent users and queries.
Options B, C & D are wrong as they scale the cluster by adding resources that would not be cost-effective.

upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: A

Wherever query is stuck for other long-running queries, we can use WLM.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: A

Answer A

upvoted 1 times

AWSRanger 3 years, 2 months ago

Selected Answer: A

A is right

upvoted 2 times

aws2019 3 years, 7 months ago

A is correct

upvoted 1 times

rosnl 3 years, 7 months ago

Answer A is correct. B, C, D = wrong, because they all talk about resizing or scaling which will not be cost-effective.

upvoted 3 times

lostsoul07 3 years, 7 months ago

A is the right answer

upvoted 2 times

BillyC 3 years, 8 months ago

A is correct for me

upvoted 2 times

syu31svc 3 years, 8 months ago

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

Answer is A 100%

upvoted 2 times

Paitan 3 years, 9 months ago

A for sure.

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

upvoted 4 times

Question #33 *Topic 1*

A company analyzes its data in an Amazon Redshift data warehouse, which currently has a cluster of three dense storage nodes. Due to a recent business acquisition, the company needs to load an additional 4 TB of user data into Amazon Redshift. The engineering team will combine all the user data and apply complex calculations that require I/O intensive resources. The company needs to adjust the cluster's capacity to support the change in analytical and storage requirements.
Which solution meets these requirements?

A. Resize the cluster using elastic resize with dense compute nodes.

B. Resize the cluster using classic resize with dense compute nodes.

C. Resize the cluster using elastic resize with dense storage nodes.

D. Resize the cluster using classic resize with dense storage nodes.

Suggested Answer: *C*
Reference:
https://aws.amazon.com/redshift/pricing/

Community vote distribution

lui Highly Voted 3 years, 9 months ago
Vote A.
"currently has a cluster of three dense storage nodes." means it is not single-node cluster, both resizing work, but classic resize take 2 hours–2 days or longer, depending on your data's size.
Dense Compute (DC) allow creation of very high performance data warehouses using fast CPUs, large amounts of RAM and solid-state disks (SSDs)
upvoted 34 times

skar21 3 years, 9 months ago
A & C should be avoided. Elastic resize is for temporary size adjustment. When you do the Elastic resize, the "SLICE" count "will not" change and not good for long term data load and computation.
upvoted 3 times

jove 3 years, 8 months ago
I don't think this is correct . When you elastic upsize you'll get more slices but when you elastic downsize you won't loose any slices. (I tested it myself)
upvoted 5 times

DonaldCMLIN Highly Voted 3 years, 9 months ago
The answer is C.

by https://aws.amazon.com/redshift/features/?nc1=h_ls
accoranding to the question "...combine all the user data and apply complex calculations that require I/O intensive resources",

drop A, B sinice "Dense Compute (DC) the best choice for less than 500GB of data".

DS2 (Dense Storage) nodes enable you to create large data warehouses using hard disk drives (HDDs).Most customers who run on DS2 clusters can migrate their workloads to RA3 clusters and get up to 2x performance and more storage for the same cost as DS2.

by https://aws.amazon.com/redshift/pricing/
"What to expect" section
Once you make your selection, you may wish to use elastic resize to easily adjust the amount of provisioned compute capacity within minutes for steady-state processing.
upvoted 26 times

GauravM17 3 years, 9 months ago
how DS would manage complex calculations and IO intensive aspects?
upvoted 8 times

certificationJunkie 3 years, 1 month ago

is 500 GB limit for each node? Since we are doing resize, we are adding new nodes as well. So we should take care of compute i think. Hence A.

upvoted 2 times

rag_mat_80 Most Recent 1 year, 2 months ago

classic resize does not let you change node type (note that original config for infra is nodetype = DS2) meaning option B is out . If you only want to increase the number of nodes of same type , then you can do that with classic resize (offcource it will create a new cluster) so in my opinion C is useless as i can use classic storage itself . Between A and D i vote for A since the use case talks about compute more than storage

upvoted 1 times

rag_mat_80 1 year, 2 months ago

correction - read the use case again and the company already has data in the warehouse so one thing for sure that classic resize will not retain system tables and data . So option B and D are out . Between A and C , i still vote for A since as per redshift documentation dc2 is a good choice for data < 10 TB . From the use case we don't know what's existing size of the warehouse but we are 4 TB to X . We don't know X so dc2 seems logical

upvoted 1 times

patou 1 year, 3 months ago

Selected Answer: C

definitively C

upvoted 1 times

tsangckl 1 year, 3 months ago

Selected Answer: B

Bing answer B

B. Resize the cluster using classic resize with dense compute nodes.

Explanation:

Option B is correct because classic resize allows you to change both the node type and number of nodes. In this case, switching to dense compute nodes would provide the I/O intensive resources needed for the complex calculations. The classic resize operation also redistributes the data and reclaims the space, which would be beneficial given the additional 4 TB of user data.
Other options are not the best solutions for this scenario. For example, Options A and C involve using elastic resize, which allows you to quickly add or remove nodes, but it doesn't allow you to change the node type. Option D involves using classic resize with dense storage nodes, but this might not provide the I/O intensive resources needed for the complex calculations.

upvoted 1 times

NarenKA 1 year, 4 months ago

Selected Answer: A

It's important to note that while classic resize (Option B and D) allows for a change in node types (from DS to DC or vice versa), it involves a longer downtime as it copies data from the old cluster to a new one.
Elastic resize is faster than classic resize. it allows you to quickly add or remove nodes to the cluster while keeping the cluster online. This minimizes downtime and can be completed in minutes, as opposed to classic resize, which can take several hours or more, depending on the size of the dataset.
Dense compute (DC) nodes are optimized for performance-intensive workloads. They offer faster CPUs, increased I/O performance, and higher storage throughput compared to dense storage (DS) nodes. Given the engineering team's need to apply complex calculations that require I/O intensive resources, switching to dense compute nodes will provide the necessary computational power and I/O performance.

upvoted 2 times

metkillas 1 year, 5 months ago

Vote A.

The questions specifies I/O as being important which goes right to compute nodes. The reccomendation from AWS is to start with Elastic Resize first. The DC node types have enough to store this amount of data and the number of resized nodes won't exceed the limitations of an elastic resize; primarily the either 2x increase or decrease max of a cluster. If you need to 4x or 8x you node size, then you need to use classic.

This was also a (similar) question in Jon Bosno's practice tests in tutorialdojos

upvoted 1 times

GCPereira 1 year, 5 months ago

You need to store 4TB of user data in Redshift (data that changes little) so the size of these datasets would vary little...

They will also execute extremely complex queries...

They need to adjust to support these requirements. The first point implies continuing with dense storage nodes, but they also need compute nodes. Given that they have storage nodes and need more computation, I would change the node type, increasing this quantity... Then I would opt for the classic resize.

C agreed?

upvoted 1 times

LocalHero 1 year, 7 months ago

I cant understand this question what to ask me?
very abstract question.
so I choose C . ChatCPT answerd hahaha.

upvoted 1 times

michalf84 1 year, 7 months ago

Selected Answer: C

Elastic as faster and dense storage due to size limit6

upvoted 1 times

markstudy 1 year, 7 months ago

Selected Answer: C

I believe option C is the most suitable choice. The reason is that dc2 nodes have a relatively limited SSD capacity of 160GB, while your data size is 4TB. Therefore, you'll need to opt for a dense storage node type to handle the increased storage requirements effectively.

upvoted 2 times

leotoras 1 year, 9 months ago

A is correct: Elastic resize across node type automates the steps of taking a snapshot, creating a new cluster, deleting the old cluster, and renaming the new cluster into a simple, quick, and familiar operation. Elastic resize operation can be run at any time or can be scheduled to run at a future time. Customers can quickly upgrade their existing DS2 or DC2 node type-based cluster to the new RA3 node type with elastic resize.

upvoted 1 times

zanhsieh 1 year, 10 months ago

Selected Answer: B

Vote for B. Following is the experience I tried on Sept 1st 2023
- Redshift currently allows me to create a dc2 or ra3 cluster. Although it shows the ds2 option on Resize option, the console will display "NumberOfNodesQuotaExceeded You do not have access to node type ds2.xlarge. Choose another node type" even the quota in the service is enough. In short, I don't think the candidate can simulate this at home unless he/she currently works for some big companies with existing ds2 Redshift clusters.

upvoted 3 times

zanhsieh 1 year, 10 months ago

- The way how Redshift increases its computation power and storage is either adding more nodes (elastic) with uniform node type and storage, or snapshot whole cluster then upgrade (classic). The snapshot way can change node type.
- Since AWS deprecated dense storage(ds2) cluster creation since 2021-08-01 and deprecated ds2 node type since 2021-12-31, no dense storage node can be added, which means we should drop C and D.
- Since converting from ds to dc, there is no way to use elastic, so we drop A.
- Can dc2 handle extra 4TB without exceeding its maximum number of nodes? Yes. dc2.l max nodes 32, total capacity 5.12TB; dc2.8xl max nodes 128, total capacity 326TB

upvoted 1 times

zanhsieh 1 year, 10 months ago

https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html#working-with-clusters-overview

upvoted 1 times

r3mo 1 year, 10 months ago

The Answer is C:
Because... Dense storage nodes provide a balance between "storage capacity and computational power", making them suitable for analytical workloads that involve heavy I/O operations.

upvoted 2 times

MLCL 1 year, 11 months ago

Selected Answer: A

Elsatic resize allows to change node types.

upvoted 2 times

developeranfc 1 year, 11 months ago

Selected Answer: B

Classic resize is for change the node types, we already got storage and we need compute

upvoted 3 times

rookiee1111 1 year, 11 months ago

Selected Answer: B

Classic Resize can allow changing node type and no of nodes, + there is no time constraints added in the question.

upvoted 2 times

Question #34 *Topic 1*

A company stores its sales and marketing data that includes personally identifiable information (PII) in Amazon S3. The company allows its analysts to launch their own Amazon EMR cluster and run analytics reports with the data. To meet compliance requirements, the company must ensure the data is not publicly accessible throughout this process. A data engineer has secured Amazon S3 but must ensure the individual EMR clusters created by the analysts are not exposed to the public internet.
Which solution should the data engineer to meet this compliance requirement with LEAST amount of effort?

A. Create an EMR security configuration and ensure the security configuration is associated with the EMR clusters when they are created.

B. Check the security group of the EMR clusters regularly to ensure it does not allow inbound traffic from IPv4 0.0.0.0/0 or IPv6 ::/0.

C. Enable the block public access setting for Amazon EMR at the account level before any EMR cluster is created.

D. Use AWS WAF to block public internet access to the EMR clusters across the board.

Suggested Answer: *B*
Reference:
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-security-groups.html

Community vote distribution
C (91%) 9%

Priyanka_01 Highly Voted 3 years, 9 months ago
C??
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html
upvoted 25 times

awssp12345 3 years, 9 months ago
Agreed
upvoted 1 times

bigollo 3 years, 9 months ago
the cluster is already created, and you can not recreate it because
is much effort
upvoted 1 times

bigollo 3 years, 9 months ago
my bad. I read again and is c
upvoted 2 times

kondi2309 Most Recent 1 year, 4 months ago
Selected Answer: C
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html
upvoted 1 times

GCPereira 1 year, 5 months ago
--- workflow ---

data with PII -> s3

analyst 1 -> EMR 1
analyst 2 -> EMR 2
...
analyst n -> EMR n

(In fact, what company allows its analysts to create an individual EMR for each person?! o.O)

--- objective ---

ensure the EMR is not accessible by public internet

--- way to make this with the least effort and least cost ---

block all account emr public access

--- have another way to make this? ---

yes, if a data analyst specialist designs a AMI for all EMR clusters and schedules a daily job to create an EMR for all analysts... buuuuuuuuut, have a lot of effort rsrsrs
upvoted 2 times

monkeydba 1 year, 7 months ago
https://aws.amazon.com/about-aws/whats-new/2019/08/amazon-emr-introduces-block-public-access-configuration-to-secure-emr-clusters-from-unintentional-network-exposure/
upvoted 1 times

pk349 2 years, 1 month ago
C: I passed the test
upvoted 4 times

Ashoks 2 years, 4 months ago
Answer is C
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: C
Correct answer is C as the EMR clusters can be configured with a block public access setting which is applied to all regions within an account.

Amazon EMR block public access prevents a cluster in a public subnet from launching when any security group associated with the cluster has a rule that allows inbound traffic from IPv4 0.0.0.0/0 or IPv6 ::/0 (public access) on a port, unless the port has been specified as an exception. Port 22 is an exception by default. You can configure exceptions to allow public access on a port or range of ports. Block public access does not take effect in private subnets.

A is wrong as security configurations can be used to configure data encryption, Kerberos authentication, and Amazon S3 authorization for EMRFS.

B is wrong Although this approach is possible, it entails a management overhead of regularly updating the security groups of the EMR cluster.

Option D is wrong as WAF does not work with EMR clusters.
upvoted 4 times

pgf909 2 years, 8 months ago
Selected Answer: B
the company must ensure the data is not publicly accessible throughout this process. How to ensure SG not be modified during the whole process if you choose C?
upvoted 1 times

pgf909 2 years, 8 months ago
B ---- as Block public access does not block IAM principals with appropriate permissions from updating security group configurations to allow public access on running clusters.... https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html
I would suggest customer to use config to trigger auto mitigation if any port is opened to public access.
upvoted 1 times

Arka_01 2 years, 9 months ago
Selected Answer: C
"with LEAST amount of effort" - this is the key statement here.
upvoted 1 times

rocky48 2 years, 10 months ago
Selected Answer: C
Selected Answer: C
upvoted 1 times

Ramshizzle 3 years ago

Selected Answer: C

B is obviously wrong. AWS Exams would never allow a compliance solution to manually check if the settings are correct every now and then.
C is better

upvoted 2 times

Bik000 3 years, 1 month ago

Selected Answer: C

My Answer is C

upvoted 1 times

CHRIS12722222 3 years, 3 months ago

Option C does not make sense since this is already enabled by default.
Option B is better. I think the best solution is to use a custom config rule with SSM remediation

https://asecure.cloud/a/cfgrule_c_emr_security_groups_restricted/

upvoted 1 times

CHRIS12722222 3 years, 3 months ago

Also it does not prevent authorised persons from overriding the default EMR block public access settings when the cluster is running.
"Block public access is only applicable during cluster creation. Block public access does not block IAM principals with appropriate permissions from updating security group configurations to allow public access on running clusters."
Ref: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html

upvoted 1 times

Shraddha 3 years, 7 months ago

Ans C
This is a textbook question.

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html

upvoted 3 times

Shraddha 3 years, 7 months ago

Ans C
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html

upvoted 4 times

AjithkumarSL 3 years, 8 months ago

I think C is Default. The question is what we need to do to ensure that, and we have to make sure the ports are not open as public.. Do you think the correct answer is B?

upvoted 4 times

Question #35 *Topic 1*

A financial company uses Amazon S3 as its data lake and has set up a data warehouse using a multi-node Amazon Redshift cluster. The data files in the data lake are organized in folders based on the data source of each data file. All the data files are loaded to one table in the Amazon Redshift cluster using a separate
COPY command for each data file location. With this approach, loading all the data files into Amazon Redshift takes a long time to complete. Users want a faster solution with little or no increase in cost while maintaining the segregation of the data files in the S3 data lake.
Which solution meets these requirements?

A. Use Amazon EMR to copy all the data files into one folder and issue a COPY command to load the data into Amazon Redshift.

B. Load all the data files in parallel to Amazon Aurora, and run an AWS Glue job to load the data into Amazon Redshift.

C. Use an AWS Glue job to copy all the data files into one folder and issue a COPY command to load the data into Amazon Redshift.

D. Create a manifest file that contains the data file locations and issue a COPY command to load the data into Amazon Redshift.

Suggested Answer: *A*
Reference:
https://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html

Community vote distribution
D (100%)

carol1522 Highly Voted 3 years, 9 months ago
D? https://docs.aws.amazon.com/redshift/latest/dg/loading-data-files-using-manifest.html
upvoted 25 times

cloudlearnerhere Highly Voted 2 years, 7 months ago
Selected Answer: D
Correct answer is D as a manifest file can be used to load the data. Also, its recommended to have a single COPY command instead of multiple concurrent COPY commands for performance.

Use the COPY command to load a table in parallel from data files on Amazon S3. You can specify the files to be loaded by using an Amazon S3 object prefix or by using a manifest file.

Amazon Redshift can automatically load in parallel from multiple compressed data files.

However, if you use multiple concurrent COPY commands to load one table from multiple files, Amazon Redshift is forced to perform a serialized load. This type of load is much slower and requires a VACUUM process at the end if the table has a sort column defined.

Options A, B & C are wrong as they add unnecessary work and cost.
upvoted 9 times

crs1234 2 years, 1 month ago
Can you share a link that gives more insight into "However, if you use multiple concurrent COPY commands to load one table from multiple files, Amazon Redshift is forced to perform a serialized load. This type of load is much slower and requires a VACUUM process at the end if the table has a sort column defined"?
upvoted 1 times

kondi2309 Most Recent 1 year, 4 months ago
Selected Answer: D
Ans D, single COPY command for performance and manifest file for loading data
upvoted 1 times

GCPereira 1 year, 5 months ago
datalake s3 -> dw redshift

problems copying files using copy-by-prefix

need a solution without increasing costs

A) emr is expansive
b) aurora needs effort configuration and glue needs development effort
c) glue job needs a development effort and copying all files to the same prefix will create a problem... which file goes to which table?
d) manifest file is the best option because you can specify exactly the prefix/key to your copy command
upvoted 1 times

tsk9921 2 years ago
D: manifest file is valid option
upvoted 1 times

pk349 2 years, 1 month ago
D: I passed the test
upvoted 2 times

Arka_01 2 years, 9 months ago
Selected Answer: D
You can use a single copy command with manifest file, containing different S3 locations. This will speed up the COPY process.
upvoted 1 times

rocky48 2 years, 11 months ago
Selected Answer: D
Selected Answer: D
upvoted 1 times

Bik000 3 years, 1 month ago
Selected Answer: D
My Answer is D
upvoted 1 times

AWSRanger 3 years, 2 months ago
Selected Answer: D
D is correct
upvoted 2 times

Shraddha 3 years, 7 months ago
Ans D
A = wrong, no segregation, increased cost. B = wrong, no segregation, unnecessary work, increased cost. C = wrong, no segregation, increased cost.
This is a question on how COPY command work. In general you should use only one COPY command because Redshift will load data in parallel, if you use many COPYs Redshift will have to load data in sequential manner.

https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-single-copy-command.html

https://docs.aws.amazon.com/redshift/latest/dg/r_COPY_command_examples.html#copy-command-examples-manifest
upvoted 6 times

lostsoul07 3 years, 7 months ago
D is the right answer
upvoted 2 times

BillyC 3 years, 8 months ago
My answer is D
upvoted 2 times

sanjaym 3 years, 8 months ago
D is right answer.
upvoted 2 times

syu31svc 3 years, 8 months ago
From the link:https://docs.aws.amazon.com/redshift/latest/dg/loading-data-files-using-manifest.html
"You can use a manifest to ensure that the COPY command loads all of the required files, and only the required files, for a data load"
So answer is D
upvoted 7 times

Paitan 3 years, 8 months ago

Using manifest file is the right choice. So option D.

upvoted 3 times

Saaho 3 years, 8 months ago

Yes D is the right answer

upvoted 4 times

Question #36 *Topic 1*

A company's marketing team has asked for help in identifying a high performing long-term storage service for their data based on the following requirements:

- ✑ The data size is approximately 32 TB uncompressed.
- ✑ There is a low volume of single-row inserts each day.
- ✑ There is a high volume of aggregation queries each day.
- ✑ Multiple complex joins are performed.
- ✑ The queries typically involve a small subset of the columns in a table.

Which storage service will provide the MOST performant solution?

A. Amazon Aurora MySQL

B. Amazon Redshift

C. Amazon Neptune

D. Amazon Elasticsearch

Suggested Answer: *B*

Community vote distribution

B (100%)

Paitan Highly Voted 3 years, 9 months ago

Redhift for sure.

upvoted 28 times

jove Highly Voted 3 years, 9 months ago

The simplest question in the exam.

upvoted 9 times

kondi2309 Most Recent 1 year, 4 months ago

Selected Answer: B

Amazon Redshift meets all the requirements.

upvoted 1 times

pk349 2 years, 1 month ago

B: I passed the test

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B as Redshift as it can be used for OLAP processing and meets all the requirements.

Amazon Redshift uses SQL to analyze structured and semi-structured data across data warehouses, operational databases, and data lakes using AWS-designed hardware and machine learning to deliver the best price-performance at any scale.
Option A is wrong as Amazon Aurora MySQL is ideal for OLTP solutions and not OLAP.

Option C s wrong as Amazon Neptune is a fast, reliable, fully-managed graph database service that makes it easy to build and run applications.

Option D is wrong as Amazon Elasticsearch would not allow complex queries.

upvoted 5 times

Arka_01 2 years, 9 months ago

Selected Answer: B

The scenario here is indicating for a columnar data storage. So the answer will be Amazon Redshift.

upvoted 1 times

rocky48 2 years, 10 months ago

Selected Answer: B

B is the right answer

upvoted 1 times

AWSRanger 3 years, 2 months ago

Selected Answer: B

B is correct

upvoted 1 times

sanpak 3 years, 6 months ago

why not D ? elastic search, search in subset of column works better in ES...

upvoted 1 times

Billhardy 3 years, 7 months ago

Ans B

upvoted 1 times

Shraddha 3 years, 8 months ago

Ans B

A = wrong, Aurora for OLTP not OLAP. C = wrong, graph database not relevant. D = wrong, complex joins in ES are expensive.

upvoted 4 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 3 times

BillyC 3 years, 9 months ago

My answer is B

upvoted 3 times

Question #37 *Topic 1*

A technology company is creating a dashboard that will visualize and analyze time-sensitive data. The data will come in through Amazon Kinesis Data Firehose with the butter interval set to 60 seconds. The dashboard must support near-real-time data.
Which visualization solution will meet these requirements?

A. Select Amazon OpenSearch Service (Amazon Elasticsearch Service) as the endpoint for Kinesis Data Firehose. Set up an OpenSearch Dashboards (Kibana) using the data in Amazon OpenSearch Service (Amazon ES) with the desired analyses and visualizations.

B. Select Amazon S3 as the endpoint for Kinesis Data Firehose. Read data into an Amazon SageMaker Jupyter notebook and carry out the desired analyses and visualizations.

C. Select Amazon Redshift as the endpoint for Kinesis Data Firehose. Connect Amazon QuickSight with SPICE to Amazon Redshift to create the desired analyses and visualizations.

D. Select Amazon S3 as the endpoint for Kinesis Data Firehose. Use AWS Glue to catalog the data and Amazon Athena to query it. Connect Amazon QuickSight with SPICE to Athena to create the desired analyses and visualizations.

Suggested Answer: *A*

Community vote distribution

A (100%)

rb39 Highly Voted 3 years, 2 months ago

Selected Answer: A

near real-time dashboards -> operational => OpenSearch

upvoted 15 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: A

Correct answer is A as Kinesis Data Firehose can ingest data to ElasticSearch and with Kibana it can provide near-real-time visualization.

Option B is wrong as SageMaker Jupyter notebook does not provide near-realt0tie visualization, but it is more for data exploration.

Option C is wrong as data with SPICE is cached and needs to be refreshed, so it does not provide near-real-time data.

Option D is wrong as using SPICE and Glue does not provide near-real-time data.

upvoted 9 times

nadavw 2 years, 5 months ago

The minimum refresh frequency of QucikSight is hourly (E. edition)
https://docs.aws.amazon.com/quicksight/latest/user/refreshing-imported-data.html

upvoted 2 times

pk349 Most Recent 2 years, 1 month ago

A: I passed the test

upvoted 1 times

Gabba 2 years, 4 months ago

Selected Answer: A

Real time dashboard is Kibana, so option A.

upvoted 1 times

rav009 2 years, 8 months ago

Selected Answer: A

A
textbox question

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: A

"The dashboard must support near-real-time data" and "time-sensitive" are the keys here. OpenSearch and Kibana can only meet these requirements.

upvoted 1 times

Question #38 *Topic 1*

A financial company uses Apache Hive on Amazon EMR for ad-hoc queries. Users are complaining of sluggish performance.
A data analyst notes the following:
➩ Approximately 90% of queries are submitted 1 hour after the market opens.
Hadoop Distributed File System (HDFS) utilization never exceeds 10%.
▪
Which solution would help address the performance issues?

A. Create instance fleet configurations for core and task nodes. Create an automatic scaling policy to scale out the instance groups based on the Amazon CloudWatch CapacityRemainingGB metric. Create an automatic scaling policy to scale in the instance fleet based on the CloudWatch CapacityRemainingGB metric.

B. Create instance fleet configurations for core and task nodes. Create an automatic scaling policy to scale out the instance groups based on the Amazon CloudWatch YARNMemoryAvailablePercentage metric. Create an automatic scaling policy to scale in the instance fleet based on the CloudWatch YARNMemoryAvailablePercentage metric.

C. Create instance group configurations for core and task nodes. Create an automatic scaling policy to scale out the instance groups based on the Amazon CloudWatch CapacityRemainingGB metric. Create an automatic scaling policy to scale in the instance groups based on the CloudWatch CapacityRemainingGB metric.

D. Create instance group configurations for core and task nodes. Create an automatic scaling policy to scale out the instance groups based on the Amazon CloudWatch YARNMemoryAvailablePercentage metric. Create an automatic scaling policy to scale in the instance groups based on the CloudWatch YARNMemoryAvailablePercentage metric.

Suggested Answer: *C*

Community vote distribution

D (100%)

Shraddha Highly Voted 3 years, 7 months ago

Ans D
A and B = wrong, instance fleet does not support auto scaling. C = wrong, HDFS utilization never exceeds 10% no scaling will never happen.
upvoted 23 times

lakediver 3 years, 6 months ago

The following are two commonly used metrics for automatic scaling:
YarnMemoryAvailablePercentage: This is the percentage of remaining memory that's available for YARN.
ContainerPendingRatio: This is the ratio of pending containers to allocated containers. You can use this metric to scale a cluster based on container-allocation behavior for varied loads. This is useful for performance tuning.
upvoted 3 times

lakediver 3 years, 6 months ago

Agree
For further reference see
https://aws.amazon.com/premiumsupport/knowledge-center/auto-scaling-in-amazon-emr/
upvoted 2 times

ariane_tateishi Highly Voted 3 years, 8 months ago

D should be the right answer. Considering the following links: the first link is possible to see that the right metric to this requirement is YARNMemoryAvailablePercentage, because the HDFS never is over 10%. The second link explain that if you will use auto scaling so you should use instance group.
https://docs.aws.amazon.com/emr/latest/ManagementGuide/UsingEMR_ViewingMetrics.html
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-instances-guidelines.html
upvoted 7 times

monkeydba Most Recent 1 year, 7 months ago

"Managed scaling is available for clusters composed of either instance groups or instance fleets."
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-managed-scaling.html#:~:text=Managed%20scaling%20is%20available%20for%20clusters%20composed%20of%20either%20instance%20groups%20or%20instance%2(
upvoted 1 times

pk349 2 years, 1 month ago

D: I passed the test

upvoted 1 times

srirnag 2 years, 4 months ago

YARNMemoryAvailablePercentage-> is for CPU intensive workload,
CapacityRemainingGB-> for Capacity intensive workload,
Instance fleet is ruled out. Hence, D

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: D

Correct answer is D as instance group configurations for core and task nodes can be used to scale as per the YARNMemoryAvailablePercentage metric.

options A & B are incorrect because an Instance Fleet doesn't have an automatic scaling policy. Only an Instance Group has this feature.

Option C is incorrect as the CapacityRemainingGB metric is just the amount of remaining HDFS disk capacity and this does not exceed 10% for each run. The cluster will not scale-in or scale-out if you choose this metric.

upvoted 4 times

cloudlearnerhere 2 years, 7 months ago

CloudWatch metrics that you can use for automatic scaling in Amazon EMR, The following are two commonly used metrics for automatic scaling:

YarnMemoryAvailablePercentage: This is the percentage of remaining memory that's available for YARN.

ContainerPendingRatio: This is the ratio of pending containers to allocated containers. You can use this metric to scale a cluster based on container-allocation behavior for varied loads. This is useful for performance tuning.

For the given use case, the correct solution should support automatic scaling. You can set up automatic scaling in Amazon EMR for an instance group, adding and removing instances automatically based on the value of an Amazon CloudWatch metric that you specify. The metric YARNMemoryAvailablePercentage represents the percentage of remaining memory available to YARN (YARNMemoryAvailablePercentage = MemoryAvailableMB / MemoryTotalMB). This value is useful for scaling cluster resources based on YARN memory usage.

upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: D

Instance Fleet cannot take part in Auto-Scaling. CapacityRemainingGB is not the parameter to refer as "(HDFS) utilization never exceeds 10%". So the answer is D.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: D

Selected Answer: D

upvoted 1 times

Ramshizzle 3 years ago

Answer should be D like others have said. However, I think it would be even better to use Instance fleets and EMR Managed auto scaling, but this is not an option here.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: D

Answer is D

upvoted 1 times

jrheen 3 years, 2 months ago

Answer-D

upvoted 1 times

ShilaP 3 years, 3 months ago

D is the right answer.

upvoted 1 times

aws2019 3 years, 7 months ago

Option D is the right choice.

upvoted 1 times

Billhardy 3 years, 7 months ago

Ans D

upvoted 1 times

Naresh_Dulam 3 years, 8 months ago

Answer is D over B. Because Spot instance fleet support "managed" auto scaling and managed auto scaling can't use Cloud watch metric like YARNMemoryAvailablePercentage.

Managed auto scaling scaled depends load on the cluster.

upvoted 4 times

lostsoul07 3 years, 8 months ago

D is the right answer

upvoted 1 times

BillyC 3 years, 8 months ago

D IS Correct for my

upvoted 3 times

Question #39 *Topic 1*

A media company has been performing analytics on log data generated by its applications. There has been a recent increase in the number of concurrent analytics jobs running, and the overall performance of existing jobs is decreasing as the number of new jobs is increasing. The partitioned data is stored in
Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) and the analytic processing is performed on Amazon EMR clusters using the EMR File System
(EMRFS) with consistent view enabled. A data analyst has determined that it is taking longer for the EMR task nodes to list objects in Amazon S3.
Which action would MOST likely increase the performance of accessing log data in Amazon S3?

A. Use a hash function to create a random string and add that to the beginning of the object prefixes when storing the log data in Amazon S3.

B. Use a lifecycle policy to change the S3 storage class to S3 Standard for the log data.

C. Increase the read capacity units (RCUs) for the shared Amazon DynamoDB table.

D. Redeploy the EMR clusters that are running slowly to a different Availability Zone.

Suggested Answer: *D*

Community vote distribution

C (100%)

Paitan Highly Voted 3 years, 9 months ago
Option C.
EMRFS consistent view tracks consistency using a DynamoDB table to track objects in Amazon S3 that have been synced with or created by EMRF. So increasing RCU for the shared DynamoDB table will help here.
upvoted 28 times

Priyanka_01 Highly Voted 3 years, 9 months ago
C any thoughts??
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emrfs-metadata.html
upvoted 13 times

awssp12345 3 years, 9 months ago
Agreed https://docs.aws.amazon.com/emr/latest/ReleaseGuide/EMR_Hive_Optimizing.html
upvoted 1 times

jueueuergen 3 years, 8 months ago
Your link is about DynamoDB as a data source and therefore unrelated to the question. The correct page is
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emrfs-metadata.html
upvoted 3 times

awssp12345 3 years, 9 months ago
Since the question specifically says "Amazon EMR clusters using the EMR File System
(EMRFS) with consistent view enabled" C makes most sense.
upvoted 2 times

Debi_mishra Most Recent 1 year, 12 months ago
Popular option C is actually wrong. Increasing RCU of dynamodb wont increase S3 list performance. Looks like there are too many tiny objects and its hitting S3 API limitations. A sounds more logical.
upvoted 2 times

pk349 2 years, 1 month ago
C: I passed the test
upvoted 1 times

anjuvinayan 2 years, 2 months ago
Answer is C.
When consistent view is enabled, a dynamo db table is created and s3 object metadata is stored in this table. Whenever s3 listing is done, it reads data from dynamodb table.
upvoted 2 times

srirnag 2 years, 4 months ago

A is right. C is not right since DDB is not in picture.
upvoted 1 times

Arjun777 2 years, 4 months ago
How is DynamoDB related to this question pls ? EMR Task node not able to lsit S3 directories and if a hash function to create a random string and add that to the beginning of the object prefixes when storing the log data in Amazon S3. This approach is known as "sharding" and can be an effective way to reduce the number of S3 requests required to retrieve log data. Therefore, option A is the most likely action to increase the performance of accessing log data in Amazon S3.
upvoted 1 times

henom 2 years, 7 months ago
The answer is A. There is no limit to create prefix inside a bucket. To scale the read/write capacity from/to S3 bucket, the recommended approach is to have additional prefixes inside the bucket to enhance the read/write capacity. For example, your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket. There are no limits to the number of prefixes in a bucket.

For example, your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket. There are no limits to the number of prefixes in a bucket.

You can increase your read or write performance by parallelizing reads. For example, if you create 10 prefixes in an Amazon S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second.
upvoted 6 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: C
Correct answer is C as the current setup uses EMR and EMRFS with Consistent View enabled which is supported by DynamoDB for metadata. Increasing the DynamoDB RCUs should help increase performance.

EMRFS consistent view tracks consistency using a DynamoDB table to track objects in Amazon S3 that have been synced with or created by EMRFS. The metadata is used to track all operations (read, write, update, and copy), and no actual content is stored in it. This metadata is used to validate whether the objects or metadata received from Amazon S3 matches what is expected. This confirmation gives EMRFS the ability to check list consistency and read-after-write consistency for new objects EMRFS writes to Amazon S3 or objects synced with EMRFS. Multiple clusters can share the same metadata.
upvoted 4 times

cloudlearnerhere 2 years, 7 months ago
Option A is wrong as for S3 list operation it's recommended to store metadata externally. S3 performance has been significantly improved and optimized as well.

Option C is wrong as the S3 standard would not help increase the performance issue. It only increases availability and durability.

Option D is wrong as it would not increase the S3 querying performance issue.
upvoted 1 times

shubhary25 2 years, 6 months ago
Is there a typo in your comment?
upvoted 1 times

JHJHJHJHJ 2 years, 9 months ago

Answer : A
Confirmed by paid dumps
upvoted 3 times

JoellaLi 2 years, 8 months ago
but what is the reason of not C?
upvoted 1 times

karanbhasin 2 years, 9 months ago
my answer is A. https://aws.amazon.com/premiumsupport/knowledge-center/emr-s3-503-slow-down/
upvoted 3 times

somenath 2 years, 9 months ago

The answer is A. There is no limit to create prefix inside a bucket. To scale the read/write capacity from/to S3 bucket, the recommended approach is to have additional prefixes inside the bucket to enhance the read/write capacity. For example, your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket. There are no limits to the number of prefixes in a bucket.

For example, your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket. There are no limits to the number of prefixes in a bucket.

You can increase your read or write performance by parallelizing reads. For example, if you create 10 prefixes in an Amazon S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second.

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: C

This looks like the correct answer

upvoted 1 times

rocky48 2 years, 10 months ago

Selected Answer: C

Answer is C.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: C

My Answer is C

upvoted 1 times

Balendu 3 years, 7 months ago

Contrary to most suggestions here. The correct answer is D. As the S3 is in one availability zone, it is likely that the entire cluster is deployed in a different zone. So redeploying the cluster to a different AZ can solve the problem. And Question clearly mentions "most likely". IMO definitely D

upvoted 2 times

CHRIS12722222 3 years, 3 months ago

I dont think this is the case here. The problem is that listing the s3 bucket items is taking longer because of the increase in new jobs. If emr was in different AZ, you may not be able to list bucket items

upvoted 1 times

Marcinha 3 years, 7 months ago

I Think A. Applications running on Amazon S3 today will enjoy this performance improvement with no changes, and customers building new applications on S3 do not have to make any application customizations to achieve this performance. Amazon S3's support for parallel requests means you can scale your S3 performance by the factor of your compute cluster, without making any customizations to your application. Performance scales per prefix, so you can use as many prefixes as you need in parallel to achieve the required throughput. There are no limits to the number of prefixes.

upvoted 2 times

Question #40 *Topic 1*

A company has developed several AWS Glue jobs to validate and transform its data from Amazon S3 and load it into Amazon RDS for MySQL in batches once every day. The ETL jobs read the S3 data using a DynamicFrame. Currently, the ETL developers are experiencing challenges in processing only the incremental data on every run, as the AWS Glue job processes all the S3 input data on each run.
Which approach would allow the developers to solve the issue with minimal coding effort?

A. Have the ETL jobs read the data from Amazon S3 using a DataFrame.

B. Enable job bookmarks on the AWS Glue jobs.

C. Create custom logic on the ETL jobs to track the processed S3 objects.

D. Have the ETL jobs delete the processed objects or data from Amazon S3 after each run.

Suggested Answer: *D*

Community vote distribution

B (100%)

paul0099 Highly Voted 3 years, 9 months ago

It is B

upvoted 21 times

Shraddha Highly Voted 3 years, 8 months ago

Ans B
This is a textbook question.

https://docs.aws.amazon.com/glue/latest/dg/monitor-continuations.html

upvoted 8 times

pk349 Most Recent 2 years, 1 month ago

B: I passed the test

upvoted 2 times

AwsNewPeople 2 years, 3 months ago

Selected Answer: B

The correct approach to solve the issue with minimal coding effort would be to enable job bookmarks on the AWS Glue jobs.

Enabling job bookmarks on the AWS Glue jobs would allow the ETL job to keep track of the last processed record in the data source. This way, on the next run, the job will only process the new or updated data that was added to the source since the last successful run, thus processing only the incremental data.

Using DataFrame instead of DynamicFrame or custom logic to track processed S3 objects could require significant coding effort and may not be the most efficient approach. Deleting processed objects or data from Amazon S3 after each run may not be ideal since it may result in loss of valuable historical data.

Therefore, enabling job bookmarks is the most appropriate approach to solve the issue with minimal coding effort.

upvoted 5 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B as AWS Glue can be used to export the data incrementally using job bookmarks with coding required.

AWS Glue tracks data that has already been processed during a previous run of an ETL job by persisting state information from the job run. This persisted state information is called a job bookmark. Job bookmarks help AWS Glue maintain state information and prevent the reprocessing of old data. With job bookmarks, you can process new data when rerunning on a scheduled interval. A job bookmark is composed of the states for various elements of jobs, such as sources, transformations, and targets. For example, your ETL job might read new partitions in an Amazon S3 file. AWS Glue tracks which partitions the job has processed successfully to prevent duplicate processing and duplicate data in the job's target data store.

Job bookmarks are implemented for JDBC data sources, the Relationalize transform, and some Amazon Simple Storage Service (Amazon S3) sources.

upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: B

For incremental data, Job bookmark is the built-in feature for Glue.

upvoted 1 times

Arka_01 2 years, 9 months ago

For incremental data, Job bookmark is the built-in option to choose for Glue.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: B

B is correct

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer is B

upvoted 2 times

Mobeen_Mehdi 3 years, 7 months ago

its strongly B as book mark only take new data it stops processing preprocessed data

upvoted 4 times

rosnl 3 years, 8 months ago

The answer is B, the hint is in the wording 'only in incremental data'.

upvoted 1 times

Billhardy 3 years, 8 months ago

Ans B

upvoted 2 times

brfc 3 years, 8 months ago

although B is the obvious answer the part of the question that says minimal coding effort suggests it might be D.

upvoted 1 times

gopi_data_guy 2 years, 5 months ago

There is no code change effort you just need to enabled job bookmark.

Removing processed data from S3 is the worst option as you are simply loosing the data from your datalake

upvoted 1 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 1 times

BillyC 3 years, 8 months ago

My answer is B

upvoted 1 times

syu31svc 3 years, 9 months ago

Job bookmarks help AWS Glue maintain state information and prevent the reprocessing of old data so answer is B 100%

upvoted 2 times

Paitan 3 years, 9 months ago

Job Bookmarks should do the trick. So option B.

upvoted 1 times

Question #41 *Topic 1*

A mortgage company has a microservice for accepting payments. This microservice uses the Amazon DynamoDB encryption client with AWS KMS managed keys to encrypt the sensitive data before writing the data to DynamoDB. The finance team should be able to load this data into Amazon Redshift and aggregate the values within the sensitive fields. The Amazon Redshift cluster is shared with other data analysts from different business units.
Which steps should a data analyst take to accomplish this task efficiently and securely?

A. Create an AWS Lambda function to process the DynamoDB stream. Decrypt the sensitive data using the same KMS key. Save the output to a restricted S3 bucket for the finance team. Create a finance table in Amazon Redshift that is accessible to the finance team only. Use the COPY command to load the data from Amazon S3 to the finance table.

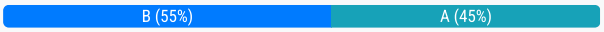
B. Create an AWS Lambda function to process the DynamoDB stream. Save the output to a restricted S3 bucket for the finance team. Create a finance table in Amazon Redshift that is accessible to the finance team only. Use the COPY command with the IAM role that has access to the KMS key to load the data from S3 to the finance table.

C. Create an Amazon EMR cluster with an EMR_EC2_DefaultRole role that has access to the KMS key. Create Apache Hive tables that reference the data stored in DynamoDB and the finance table in Amazon Redshift. In Hive, select the data from DynamoDB and then insert the output to the finance table in Amazon Redshift.

D. Create an Amazon EMR cluster. Create Apache Hive tables that reference the data stored in DynamoDB. Insert the output to the restricted Amazon S3 bucket for the finance team. Use the COPY command with the IAM role that has access to the KMS key to load the data from Amazon S3 to the finance table in Amazon Redshift.

Suggested Answer: *B*

Community vote distribution



awssp12345 Highly Voted 3 years, 9 months ago

Answer is B –
C and D are cancelled because – EMR is not needed to process DynamoDB streams. Lambda function would be good enough.

Option A is wrong because it suggests decrypting the data and storing in S3 which is not good since it contains sensitive fields.
Option B is correct because Redshift will only decrypt the data while reading it.
upvoted 53 times

freaky 3 years, 9 months ago

But why do we need to create DynamoDB streams. Streams is mentioned only in answer. Also it will be only for new data. What about the data which is already present. One of the requirement is Finane team should be able to aggregate data on sensitive field. But if they do not hae all the data in Redshift then how will aggregation provide correct result?
upvoted 3 times

blackgamer 1 year, 6 months ago

B is wrong because the data is encrypted before loading into DynamoDB which implies that it is client side encryption and Redshift doesn't support the client side encryption -

https://docs.aws.amazon.com/redshift/latest/dg/c_loading-encrypted-files.html
upvoted 1 times

JD78780 Highly Voted 3 years, 8 months ago

Correct: A
C and D can be eliminated because this is a shared Redshift cluster, so you need to create a table accessible only to the finance team. B is wrong as the application uses DynamoDB client-side encryption (not S3 client-side encryption), which means it will not automatically decrypt by AWS, and needs manual decryption before sending to S3 and then COPY'd into Redshift. Even if you want to use COPY ENCRYPTED to copy client-side encrypted S3 files, you need to specify credentials not IAM roles.
REPORT THIS AD
REPORT THIS AD
However, DynamoDB stream only does new data, so existing data won't be processed, this is not a perfect answer.
https://docs.aws.amazon.com/redshift/latest/dg/c_loading-encrypted-files.html

upvoted 13 times

dushmantha 3 years ago
I think this is the best explanation.
upvoted 1 times

ru4aws 2 years, 11 months ago
In A the issue is
though S3 is restricted the data stored still is unencrypted before loading to redshift
upvoted 2 times

JoellaLi 2 years, 8 months ago
Actually it will automatically decrypt by AWS, and no need to do manual decryption.
"After you create and configure the required components, the DynamoDB Encryption Client transparently encrypts and signs your table items when you add them to a table, and verifies and decrypts them when you retrieve them."

https://docs.aws.amazon.com/redshift/latest/dg/c_loading-encrypted-files.html
upvoted 3 times

siju13 2 years, 6 months ago
on the link you shared above, The COPY command doesn't support the following types of Amazon S3 encryption:
Client-side encryption using an AWS KMS key

if client side encruption does not support than decrypting will not work as well
upvoted 1 times

Cristian_T5 Most Recent 1 year, 2 months ago
Digital marketing isn't just about selling a product; it's about crafting an immersive brand experience. It's the fusion of compelling storytelling, cutting-edge technology, and a deep understanding of consumer psychology.
upvoted 1 times

chinmayj213 1 year, 4 months ago
When you load an encrypted file from Amazon S3 to Redshift, the encryption involved is neither purely client-side nor server-side using AWS KMS. It's a hybrid approach

Decryption during Load:

When you use the COPY command in Redshift to load the data, Redshift retrieves the data key from KMS using its IAM role or credentials.
This retrieval can be considered a server-side operation from Redshift's perspective.
upvoted 1 times

NarenKA 1 year, 4 months ago
Selected Answer: A
Lambda function processes the DynamoDB stream, the sensitive data encrypted with KMS keys can be decrypted securely using the same KMS key. Storing the decrypted data in a restricted S3 bucket accessible only to the finance team ensures that sensitive information is not exposed to unauthorised users. Creating a dedicated finance table in Redshift is accessible only to the finance team ensures that the aggregated sensitive data remains confidential and is not accessible by others. The COPY command to load data from the restricted S3 bucket into the finance table in Redshift is efficient.
B- loading encrypted data directly into Redshift and decrypting it during the COPY process is not directly supported as part of the COPY command.
C, D - involve using EMR and Apache Hive, which could add complexity and operational overhead to the data processing workflow and decryption needs to occur before the data can be processed by EMR.
upvoted 2 times

rag_mat_80 1 year, 3 months ago
COPY command does decrypt - https://docs.aws.amazon.com/redshift/latest/dg/c_loading-encrypted-files.html
upvoted 1 times

Adzz 1 year, 4 months ago
Selected Answer: B
Will go for B
upvoted 1 times

blackgamer 1 year, 6 months ago
Selected Answer: A

B is wrong because the data is encrypted before loading into DynamoDB which implies that it is client side encryption and Redshift doesn't support the client side encryption -

https://docs.aws.amazon.com/redshift/latest/dg/c_loading-encrypted-files.html

upvoted 1 times

rag_mat_80 1 year, 3 months ago

the key is this i feel - "AWS KMS managed keys to encrypt the sensitive data"

upvoted 1 times

gofavad926 1 year, 8 months ago

Selected Answer: B

B. A and B are similar but B is more secure option

upvoted 1 times

rlnd2000 1 year, 9 months ago

Selected Answer: A

B is incorrect, this option omits the step of decrypting the data before saving, I think A is the correct option.

upvoted 1 times

SMALLAM 2 years ago

The COPY command doesn't support the following types of Amazon S3 encryption: Answer A
Server-side encryption with customer-provided keys (SSE-C)
Client-side encryption using an AWS KMS key
Client-side encryption using a customer-provided asymmetric root key

upvoted 3 times

Hisayuki 2 years ago

Selected Answer: A

A is the answer

upvoted 1 times

pk349 2 years, 1 month ago

A: I passed the test

upvoted 3 times

anjuvinayan 2 years, 2 months ago

Answer is B as lambda can analyze the data in dynamo db and is save to restricted bucket which cannot be accessed without desired permission. Also to copy from s3 bucket to redshift it requires IAM role.

upvoted 2 times

akashm99101001com 2 years, 3 months ago

Selected Answer: B

https://docs.aws.amazon.com/redshift/latest/dg/c_loading-encrypted-files.html

upvoted 2 times

Arjun777 2 years, 4 months ago

Option B is not the best solution because it does not address the need to decrypt the sensitive data before loading it into Amazon Redshift. The finance team needs to be able to aggregate the values within the sensitive fields, which would not be possible if the data is not decrypted before loading it into Redshift. Option A solves this problem by creating a Lambda function that processes the DynamoDB stream and decrypts the sensitive data using the same KMS key used for encryption before loading it into Redshift. The data is also saved to a restricted S3 bucket to ensure that only the finance team has access to it.

upvoted 3 times

ota123 2 years, 5 months ago

Selected Answer: B

the decrypting is automatically done by AWS as the data is moved from Dynamo to S3. Before it's written to S3 it's reencyrpted using SSE. Writing to S3 and leaving it decrypting (as Option A suggests) would not be a secure move. Hence B should be the right answer.

upvoted 4 times

henom 2 years, 7 months ago

Ans- B

upvoted 2 times

Question #42 *Topic 1*

A company is building a data lake and needs to ingest data from a relational database that has time-series data. The company wants to use managed services to accomplish this. The process needs to be scheduled daily and bring incremental data only from the source into Amazon S3. What is the MOST cost-effective approach to meet these requirements?

A. Use AWS Glue to connect to the data source using JDBC Drivers. Ingest incremental records only using job bookmarks.

B. Use AWS Glue to connect to the data source using JDBC Drivers. Store the last updated key in an Amazon DynamoDB table and ingest the data using the updated key as a filter.

C. Use AWS Glue to connect to the data source using JDBC Drivers and ingest the entire dataset. Use appropriate Apache Spark libraries to compare the dataset, and find the delta.

D. Use AWS Glue to connect to the data source using JDBC Drivers and ingest the full data. Use AWS DataSync to ensure the delta only is written into Amazon S3.

Suggested Answer: *B*

Community vote distribution

A (100%)

paul0099 Highly Voted 3 years, 9 months ago

Seems answer is A

upvoted 27 times

zeronine Highly Voted 3 years, 9 months ago

my answer is A

upvoted 9 times

skb0071 Most Recent 1 year, 7 months ago

It's B. Glue job bookmark option is for S3 not for database.

upvoted 1 times

RollingGemini 1 year, 10 months ago

Selected Answer: A

Of course A

upvoted 1 times

juanife 2 years ago

Undoubtedly it's A.

upvoted 1 times

pk349 2 years, 1 month ago

A: I passed the test

upvoted 1 times

anjuvinayan 2 years, 2 months ago

Answer is A

Since Glue has the Job bookmark option to save the info regarding the last load, this can be used when starting next load so that duplicates are not inserted.

upvoted 1 times

renfdo 2 years, 6 months ago

Selected Answer: A

I'm sure it's A. Follow documnet explain about JDBC source.

https://docs.aws.amazon.com/glue/latest/dg/monitor-continuations.html

upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A as AWS Glue can be used to export the data from the relational database incrementally using job bookmarks in a cost-effective way.

upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: A

Incremental Data and Managed Service, these are the two key words here. So AWS Glue with Job Bookmark will do the trick.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: A

Selected Answer: A

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: A

My Answer is A

upvoted 1 times

moon2351 3 years, 3 months ago

Selected Answer: A

Answer is A

upvoted 2 times

KnightVictor 3 years, 3 months ago

Answer is A. why to use other services when Glue itself can perform the desired functions as per the question

upvoted 1 times

Shraddha 3 years, 7 months ago

Ans A

This is a textbook question.

upvoted 2 times

lostsoul07 3 years, 8 months ago

A is the right answer

upvoted 2 times

saabji 3 years, 8 months ago

A.

Job bookmarks are implemented for JDBC data sources, the Relationalize transform, and some Amazon Simple Storage Service (Amazon S3) sources

upvoted 2 times

Question #43 *Topic 1*

An Amazon Redshift database contains sensitive user data. Logging is necessary to meet compliance requirements. The logs must contain database authentication attempts, connections, and disconnections. The logs must also contain each query run against the database and record which database user ran each query.
Which steps will create the required logs?

A. Enable Amazon Redshift Enhanced VPC Routing. Enable VPC Flow Logs to monitor traffic.

B. Allow access to the Amazon Redshift database using AWS IAM only. Log access using AWS CloudTrail.

C. Enable audit logging for Amazon Redshift using the AWS Management Console or the AWS CLI.

D. Enable and download audit reports from AWS Artifact.

Suggested Answer: *C*
Reference:
https://docs.aws.amazon.com/redshift/latest/mgmt/db-auditing.html

Community vote distribution
C (100%)

Prodip Highly Voted 3 years, 9 months ago
Its C; Enhanced VPC Routing enforce COPY/UNLOAD to use VPC
upvoted 20 times

awssp12345 3 years, 9 months ago
Agreed

Amazon Redshift logs information in the following log files:
• Connection log — logs authentication attempts, and connections and disconnections.
• User log — logs information about changes to database user definitions.
• User activity log — logs each query before it is run on the database.
https://docs.aws.amazon.com/redshift/latest/mgmt/db-auditing.html
upvoted 8 times

Jh2501 3 years, 8 months ago
I am inclined to C. But does anyone know how come B is not right? CloudTrail is supposed to provide the requested service.
upvoted 2 times

jAWStest 3 years, 8 months ago
https://stackify.com/aws-redshift-monitoring-the-complete-guide/
may help with the difference between cloudtrail and db audit logging
upvoted 5 times

lakediver 3 years, 6 months ago
Agree
Further Audit logs can be analysed using Redshift Spectrum
https://aws.amazon.com/blogs/big-data/analyze-database-audit-logs-for-security-and-compliance-using-amazon-redshift-spectrum/
upvoted 2 times

Shraddha Highly Voted 3 years, 7 months ago
Ans C
A = wrong, enhanced VPC routing means data in/out within VPC. B = wrong, CloudTrail do not log data events, only configuration events. D = wrong, nonsense. This is a textbook question.

https://docs.aws.amazon.com/redshift/latest/mgmt/db-auditing.html
upvoted 8 times

pk349 Most Recent 2 years, 1 month ago
C: I passed the test
upvoted 1 times

anjuvinayan 2 years, 2 months ago

Answer is C

A-User should connect to VPN first to access Redshift in VPC, in question there is no details regarding VPN

B. Only users with AWS access will be able to connect to redshift

D. Artifacts is not a solution.

Also cloudtrail will log only access to the service and not what happened inside the service

upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: C

Correct answer is C as Redshift Audit Logging can provide the required information.

Audit logging is not enabled by default in Amazon Redshift. When you enable logging on your cluster, Amazon Redshift creates and uploads logs to Amazon S3 that capture data from the time audit logging is enabled to the present time. Each logging update is a continuation of the information that was already logged.The connection log, user log, and user activity log are enabled together by using the AWS Management Console, the Amazon Redshift API Reference, or the AWS Command Line Interface (AWS CLI).

Amazon Redshift logs information in the following log files:

Connection log — logs authentication attempts, and connections and disconnections.
User log — logs information about changes to database user definitions.
User activity log — logs each query before it is run on the database.

upvoted 3 times

cloudlearnerhere 2 years, 7 months ago

Option A is wrong as Redshift Enhanced VPC Routing supports the use of standard VPC features such as VPC Endpoints, security groups, network ACLs, managed NAT and internet gateways, enabling you to tightly manage the flow of data between your Amazon Redshift cluster and all of your data sources.

Option D is wrong as AWS Artifact is your go-to, central resource for compliance-related information that matters to you. It provides on-demand access to AWS' security and compliance reports and select online agreements.

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Option B is wrong as Amazon Redshift is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Redshift. CloudTrail captures all API calls for Amazon Redshift as events. These include calls from the Amazon Redshift console and from code calls to the Amazon Redshift API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon Redshift. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in Event history. Using the information collected by CloudTrail, you can determine certain details. These include the request that was made to Amazon Redshift, the IP address it was made from, who made it, when it was made, and other information.

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: C

It can be done by enabling Audit Logging of Redshift.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: C

Selected Answer: C

upvoted 1 times

MWL 3 years, 1 month ago

Selected Answer: C

This is what Redshift audit log do.

upvoted 1 times

jrheen 3 years, 2 months ago

Answer - A , Enhanced Routing

upvoted 1 times

Teraxs 3 years, 2 months ago

Selected Answer: C

as discussed by others

upvoted 1 times

aws2019 3 years, 7 months ago

Ans is C

upvoted 1 times

DerekKey 3 years, 8 months ago

Correct C

Amazon Redshift logs information in the following log files:

- Connection log — logs authentication attempts, and connections and disconnections.
- User log — logs information about changes to database user definitions.
- User activity log — logs each query before it is run on the database.

The connection log, user log, and user activity log are enabled together by using the AWS Management Console, the Amazon Redshift API Reference, or the AWS Command Line Interface (AWS CLI).

https://docs.aws.amazon.com/redshift/latest/mgmt/db-auditing.html

upvoted 3 times

lostsoul07 3 years, 8 months ago

C is the right answer

upvoted 2 times

mbaexam 3 years, 8 months ago

C for sure: https://aws.amazon.com/premiumsupport/knowledge-center/logs-redshift-database-cluster/

upvoted 1 times

BillyC 3 years, 8 months ago

C is correct

upvoted 1 times

syu31svc 3 years, 8 months ago

Link provided confirms C as the answer

upvoted 1 times

Woong 3 years, 8 months ago

The connection log, user log, and user activity log are enabled together by using the AWS Management Console, the Amazon Redshift API Reference, or the AWS Command Line Interface (AWS CLI). Answer is C

upvoted 2 times

Question #44 *Topic 1*

A company that monitors weather conditions from remote construction sites is setting up a solution to collect temperature data from the following two weather stations.

➩ Station A, which has 10 sensors

➩ Station B, which has five sensors

These weather stations were placed by onsite subject-matter experts.

Each sensor has a unique ID. The data collected from each sensor will be collected using Amazon Kinesis Data Streams.

Based on the total incoming and outgoing data throughput, a single Amazon Kinesis data stream with two shards is created. Two partition keys are created based on the station names. During testing, there is a bottleneck on data coming from Station A, but not from Station B. Upon review, it is confirmed that the total stream throughput is still less than the allocated Kinesis Data Streams throughput.

How can this bottleneck be resolved without increasing the overall cost and complexity of the solution, while retaining the data collection quality requirements?

A. Increase the number of shards in Kinesis Data Streams to increase the level of parallelism.

B. Create a separate Kinesis data stream for Station A with two shards, and stream Station A sensor data to the new stream.

C. Modify the partition key to use the sensor ID instead of the station name.

D. Reduce the number of sensors in Station A from 10 to 5 sensors.

Suggested Answer: *A*

Community vote distribution

C (100%)

Priyanka_01 Highly Voted 3 years, 9 months ago

C?

A and B increase the cost

upvoted 34 times

awssp12345 3 years, 8 months ago

Agreed

upvoted 2 times

lakediver 3 years, 6 months ago

Agreed

For further reading see -

https://aws.amazon.com/blogs/big-data/under-the-hood-scaling-your-kinesis-data-streams/

upvoted 2 times

sanjaym Highly Voted 3 years, 8 months ago

C is 100% correct answer.

upvoted 11 times

pk349 Most Recent 2 years, 1 month ago

C: I passed the test

upvoted 2 times

anjuvinayan 2 years, 2 months ago

Answer is C

A. No need to Increase the number of shards as in question its mentioned the throughput is less

B. More cost

C. Modifying the partition key to use the sensor ID instead of the station name is the correct answer. As of now all data from Station A which has more sensors is going to one shard and all data from Station B to another shard which has less sensors. By changing partition key to sensor ID will help to divide the data base on sensors to shard.

D. Change in infra which is not required

upvoted 4 times

rags140882 2 years, 3 months ago

Option B does involve creating a separate Kinesis data stream for Station A, which could be seen as increasing the complexity of the solution compared to modifying the partition key. However, in this scenario, the bottleneck is on data coming from Station A, and creating a separate stream with dedicated shards for that station can help to increase parallelism and improve throughput without increasing the overall cost of the solution.

On the other hand, modifying the partition key to use the sensor ID instead of the station name could result in uneven shard distribution and hot partitions if the distribution of sensors across stations is uneven. This could lead to degraded performance and require additional scaling in the future, which could increase complexity and cost over time.

So, while both options have their pros and cons, creating a separate Kinesis data stream for Station A with dedicated shards can be a more effective and scalable solution for improving throughput in this scenario.

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: C

Correct answer is C as currently the partition keys are based on station names and with two shards, Station A shard is overloaded with 10 sensors, and Station B shard with 5 sensors. Changing the partition key from station names to sensor id would distribute the data equally across shards without increasing the overall cost and complexity of the solution.

Option A is wrong as increasing shards would increase the cost.

Option B is wrong as adding Kinesis Data Stream would increase the cost.

Option D is wrong as reducing the number of sensors would reduce the data collection quality.

upvoted 5 times

cloudlearnerhere 2 years, 7 months ago

The partition key determines to which shard the record is written. The partition key is a Unicode string with a maximum length of 256 bytes. Kinesis runs the partition key value that you provide in the request through an MD5 hash function. The resulting value maps your record to a specific shard within the stream, and Kinesis writes the record to that shard. Partition keys dictate how to distribute data across the stream and use shards.

Certain use cases require you to partition data based on specific criteria for efficient processing by the consuming applications. As an example, if you use player ID pk1234 as the hash key, all scores related to that player route to shard1. The consuming application can use the fact that data stored in shard1 has an affinity with the player ID and can efficiently calculate the leaderboard. An increase in traffic related to players mapped to shard1 can lead to a hot shard. Kinesis Data Streams allows you to handle such scenarios by splitting or merging shards without disrupting your streaming pipeline.

upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

If your use cases do not require data stored in a shard to have high affinity, you can achieve high overall throughput by using a random partition key to distribute data. Random partition keys help distribute the incoming data records evenly across all the shards in the stream and reduce the likelihood of one or more shards getting hit with a disproportionate number of records. You can use a universally unique identifier (UUID) as a partition key to achieve this uniform distribution of records across shards. This strategy can increase the latency of record processing if the consumer application has to aggregate data from multiple shards.

upvoted 2 times

thirukudil 2 years, 8 months ago

Selected Answer: C

Ans is C.

A and B will increase the overall cost. D - reducing the sensors is not the good option.

C - by modifying the partition key to sensor id , input data will be evenly distributed across both the shards by avoiding the hot-sharding in the first shard

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: C

It gives you the answer here -

1. Station A is facing problem. This has 10 sensor ID and obviously more data.

2. total stream throughput is still less than the allocated Kinesis Data Streams throughput.

So we are not utilizing full stream's capability. So workloads are not evenly distributed amongst Shards.

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: C

Answer-C

upvoted 1 times

certificationJunkie 3 years, 1 month ago

C is correct answer. Increasing shards won't help there as partitioning is based on station name and there are only two stations.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: C

Answer is C

upvoted 2 times

jrheen 3 years, 2 months ago

Answer-C

upvoted 1 times

Teraxs 3 years, 2 months ago

Selected Answer: C

C- sensor id as partition key allows equal distribution of data between the two shards

upvoted 1 times

aws2019 3 years, 7 months ago

C is the right answer

upvoted 1 times

lostsoul07 3 years, 7 months ago

C is the right answer

upvoted 4 times

BillyC 3 years, 8 months ago

C is correct!

upvoted 4 times

syu31svc 3 years, 8 months ago

D is obviously wrong

From link: https://docs.aws.amazon.com/streams/latest/dev/kinesis-using-sdk-java-resharding.html

"Splitting increases the number of shards in your stream and therefore increases the data capacity of the stream. Because you are charged on a per-shard basis, splitting increases the cost of your stream"

So answer is C

upvoted 1 times

Question #45 *Topic 1*

Once a month, a company receives a 100 MB .csv file compressed with gzip. The file contains 50,000 property listing records and is stored in Amazon S3 Glacier.
The company needs its data analyst to query a subset of the data for a specific vendor.
What is the most cost-effective solution?

A. Load the data into Amazon S3 and query it with Amazon S3 Select.

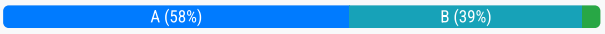
B. Query the data from Amazon S3 Glacier directly with Amazon Glacier Select.

C. Load the data to Amazon S3 and query it with Amazon Athena.

D. Load the data to Amazon S3 and query it with Amazon Redshift Spectrum.

Suggested Answer: *C*
Reference:
https://aws.amazon.com/athena/faqs/

Community vote distribution



Paitan Highly Voted 3 years, 8 months ago
Since we are talking about compressed file, Amazon Glacier Select cannot be used. So we need to transfer data to S3 and then use S3 select. So option A is the right choice.
upvoted 39 times

Merrick 2 years, 5 months ago
https://docs.aws.amazon.com/ko_kr/AmazonS3/latest/userguide/selecting-content-from-objects.html
upvoted 3 times

[Removed] 3 years, 6 months ago
Archive objects that are queried by S3 Glacier Select must be formatted as uncompressed comma-separated values (CSV).
upvoted 6 times

iris22 3 years, 3 months ago
https://docs.aws.amazon.com/amazonglacier/latest/dev/glacier-select.html
upvoted 2 times

zeronine Highly Voted 3 years, 8 months ago
my answer is A. (You may need to use Athena if data is in multiple files but this question - data is in a single compressed file)
upvoted 18 times

Marvel_jarvis 3 years, 6 months ago
Athena cant query Glacier data, so A cant be right.
upvoted 2 times

cnmc 3 years, 4 months ago
A doesn't mention Athena....
upvoted 3 times

strikeEagle 3 years, 4 months ago
please read "The file is hosted in Amazon S3 Glacier..."
upvoted 1 times

awssp12345 3 years, 8 months ago
Yes! I agree. Thank you.
upvoted 2 times

NarenKA Most Recent 1 year, 4 months ago
Selected Answer: B
Glacier Select allows to run queries directly on data stored in S3 Glacier without needing to restore and move the data to an active storage class in S3. This feature is designed for scenarios exactly like this, where you need to retrieve only a small subset of data from a large archive stored in Glacier and it is more cost-effective. You are charged for the queries you run and the data retrieved, which, for a small subset of a 100 MB file, could

be minimal. This avoids the costs associated with moving the data to Amazon S3 and storing it there for querying.
A - restoring the file to S3 and then querying it with S3 Select incurs extra costs and processing time.
C and D - Athena or Redshift Spectrum are powerful for analyzing large datasets, they introduce unnecessary complexity and costs for the given task considering the relatively small size of the dataset.
upvoted 1 times

teo2157 1 year, 6 months ago

Selected Answer: B

It's B, you can use Amazon Glacier Select to query archived data in Amazon Glacier.
https://aws.amazon.com/about-aws/whats-new/2017/11/amazon-glacier-select-makes-big-data-analytics-of-archive-data-possible/?nc1=h_ls
upvoted 2 times

GCPereira 1 year, 5 months ago

accepted, this is an old question that does not reflect current standards of the
upvoted 1 times

chinmayj213 1 year, 9 months ago

It is tricky question as now a day "Glacier-Select" support compressed g-zip csv. but the problem is file loaded a month ago and deep archival has limitation of one month to retrieve else we need to pay expedited retrieval fees
upvoted 1 times

chinmayj213 1 year, 9 months ago

https://github.com/awsdocs/amazon-glacier-developer-guide/blob/master/doc_source/glacier-select.md
upvoted 1 times

confuzz 1 year, 10 months ago

Looks like not up-to-date question. Links to AWS Docs posted in this discussion before are redirected now and don't mention Glacier Select (I don't count Blog and unofficial resources). It's only S3 Select now and it has no limit for Amazon S3 Glacier Instant Retrieval storage class.
https://docs.aws.amazon.com/AmazonS3/latest/userguide/selecting-content-from-objects.html
There is no just "S3 Glacier" now.
This limitation "The archived objects that are being queried by the select request must be formatted as uncompressed comma-separated values (CSV) files" is googled now only in Restore Object command in SDK and CLI which I guess is used to restore from Glacier.
If come back to the past, I would go with A, since I believe guys saw this limitation for Glacier Select to query from uncompressed files in the links before.
upvoted 5 times

confuzz 1 year, 10 months ago

Amazon S3 objects that are stored in the S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage classes are not immediately accessible. To access an object in these storage classes, you must restore a temporary copy of the object to its S3 bucket for a specified duration (number of days). https://docs.aws.amazon.com/AmazonS3/latest/userguide/restoring-objects.html
upvoted 2 times

Parthasarathi 1 year, 11 months ago

Selected Answer: B

The Amazon S3 Glacier Select works on objects as it supports a subset of SQL with a format like CSV, JSON, or Apache Parquet format. Objects compressed with GZIP or BZIP2 (for CSV and JSON objects only), and server-side encrypted objects can also be retrieved.
Ref : https://www.scaler.com/topics/aws/s3-glacier-select/
upvoted 7 times

pk349 2 years, 1 month ago

A: I passed the test
upvoted 1 times

flanfranco 2 years, 2 months ago

Option A:
https://aws.amazon.com/blogs/aws/s3-glacier-select/
upvoted 1 times

anjuvinayan 2 years, 2 months ago

I have searched a lot and couldn't find document stating glacier will not support compressed file. For me Glacier select is first choice and then s3 select considering cost.
upvoted 3 times

thirstylion 2 years, 3 months ago

Answer: B.
Nowhere have I read that Glacier Select cannot query compressed CSV files.
upvoted 3 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A as AWS S3 Select enables querying S3 data on selected fields. As S3 Glacier Select does not support uncompressed data, it needs to be restored to S3.

With Amazon S3 Select, you can use simple structured query language (SQL) statements to filter the contents of an Amazon S3 object and retrieve just the subset of data that you need. By using Amazon S3 Select to filter this data, you can reduce the amount of data that Amazon S3 transfers, which reduces the cost and latency to retrieve this data.

Amazon S3 Select works on objects stored in CSV, JSON, or Apache Parquet format. It also works with objects that are compressed with GZIP or BZIP2 (for CSV and JSON objects only), and server-side encrypted objects. You can specify the format of the results as either CSV or JSON, and you can determine how the records in the result are delimited.

Option B is wrong as Archive objects that are queried by S3 Glacier Select must be formatted as uncompressed comma-separated values (CSV).

Options C & D are wrong as Athena and Redshift would add additional cost.
upvoted 4 times

Arumugam_S 1 year, 8 months ago
https://www.scaler.com/topics/aws/s3-glacier-select/
but in this document they have mentioned s3 glacier select support compressed gzip format
upvoted 1 times

Rejju 2 years, 8 months ago
A and B seems to be wrong according to the below statement: Amazon S3 Select scan range requests support Parquet, CSV (without quoted delimiters), and JSON objects (in LINES mode only). CSV and JSON objects must be uncompressed. For line-based CSV and JSON objects, when a scan range is specified as part of the Amazon S3 Select request, all records that start within the scan range are processed. For Parquet objects, all of the row groups that start within the scan range requested are processed.

D is costly and hence only feasible ans I see is C.
upvoted 1 times

Rejju 2 years, 8 months ago
Amazon S3 Select scan range requests support Parquet, CSV (without quoted delimiters), and JSON objects (in LINES mode only). CSV and JSON objects must be uncompressed. For line-based CSV and JSON objects, when a scan range is specified as part of the Amazon S3 Select request, all records that start within the scan range are processed. For Parquet objects, all of the row groups that start within the scan range requested are processed.
upvoted 1 times

Haimett 2 years, 8 months ago

Selected Answer: A

GZIP or BZIP2 - CSV and JSON files can be compressed using GZIP or BZIP2. GZIP and BZIP2 are the only compression formats that Amazon S3 Select supports for CSV and JSON files. Amazon S3 Select supports columnar compression for Parquet using GZIP or Snappy. Amazon S3 Select does not support whole-object compression for Parquet objects.
upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: A

Both Athena and Redshift are viable but way more costly than the S3 select option. Glacier Select cannot query on zipped data.
upvoted 1 times

Question #46 *Topic 1*

A retail company is building its data warehouse solution using Amazon Redshift. As a part of that effort, the company is loading hundreds of files into the fact table created in its Amazon Redshift cluster. The company wants the solution to achieve the highest throughput and optimally use cluster resources when loading data into the company's fact table.
How should the company meet these requirements?

A. Use multiple COPY commands to load the data into the Amazon Redshift cluster.

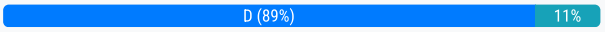
B. Use S3DistCp to load multiple files into the Hadoop Distributed File System (HDFS) and use an HDFS connector to ingest the data into the Amazon Redshift cluster.

C. Use LOAD commands equal to the number of Amazon Redshift cluster nodes and load the data in parallel into each node.

D. Use a single COPY command to load the data into the Amazon Redshift cluster.

Suggested Answer: *B*

Community vote distribution



Priyanka_01 Highly Voted 3 years, 9 months ago

D.
https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-single-copy-command.html
upvoted 35 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: D
Correct answer is D as using a single COPY command would load the data in parallel.

Amazon Redshift can automatically load in parallel from multiple compressed data files.

However, if you use multiple concurrent COPY commands to load one table from multiple files, Amazon Redshift is forced to perform a serialized load. This type of load is much slower and requires a VACUUM process at the end if the table has a sort column defined.

Option A is wrong as multiple COPY commands would force Redshift to perform a serialized load.

Option B is wrong as using EMR just makes the solution complicated.

Option C is wrong as there is no LOAD command with Redshift.
upvoted 9 times

gofavad926 Most Recent 1 year, 8 months ago

Selected Answer: D
D. https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-single-copy-command.html
upvoted 1 times

pk349 2 years, 1 month ago

D: I passed the test
upvoted 2 times

anjuvinayan 2 years, 2 months ago

Answer is D
Copy Command is used to load data to redshift. Already single copy command load data in parallel.
upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: D
Single copy command is the correct answer.
upvoted 1 times

fqc 2 years, 10 months ago

Selected Answer: D

The copy command is by default parallelized and effcient. It uses to load data from sources other than RedShift. If it is RedShift then use INSERT INTO or CREATE TABLE AS commans like in SQL.

upvoted 2 times

fqc 2 years, 10 months ago

Selected Answer: B

The copy command is by default parallelized and effcient. It uses to load data from sources other than RedShift. If it is RedShift then use INSERT INTO or CREATE TABLE AS commans like in SQL.

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: D

The copy command is by default parallelized and efficient. It uses to load data from sources other than RedShift. If it is RedShift then use INSERT INTO or CREATE TABLE AS commas like in SQL. No point of creating expensive solution as given in B.

The answer should be D

upvoted 1 times

dushmantha 3 years ago

The copy command is by default parallelized and effcient. It uses to load data from sources other than RedShift. If it is RedShift then use INSERT INTO or CREATE TABLE AS commans like in SQL. No point of creating expensive solution as given in B. The answer should be D

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: D

Answer is D

upvoted 1 times

certificationJunkie 3 years, 1 month ago

It's D. The only requirement is that all the files should lie under a common directory and in the redshift copy command you need to pass the path till directory so that it will consider all the files inside it.

upvoted 1 times

Teraxs 3 years, 2 months ago

Selected Answer: D

https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-single-copy-command.html

upvoted 1 times

RSSRAO 3 years, 4 months ago

D is correct answer

upvoted 1 times

shenshu 3 years, 5 months ago

Selected Answer: D

its D https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-single-copy-command.html

upvoted 1 times

aws2019 3 years, 7 months ago

Option D

https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-single-copy-command.html

upvoted 1 times

Billhardy 3 years, 7 months ago

Ans D

upvoted 1 times

Question #47 *Topic 1*

A data analyst is designing a solution to interactively query datasets with SQL using a JDBC connection. Users will join data stored in Amazon S3 in Apache ORC format with data stored in Amazon OpenSearch Service (Amazon Elasticsearch Service) and Amazon Aurora MySQL.
Which solution will provide the MOST up-to-date results?

A. Use AWS Glue jobs to ETL data from Amazon ES and Aurora MySQL to Amazon S3. Query the data with Amazon Athena.

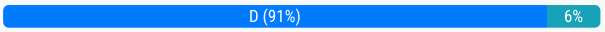
B. Use Amazon DMS to stream data from Amazon ES and Aurora MySQL to Amazon Redshift. Query the data with Amazon Redshift.

C. Query all the datasets in place with Apache Spark SQL running on an AWS Glue developer endpoint.

D. Query all the datasets in place with Apache Presto running on Amazon EMR.

Suggested Answer: *C*

Community vote distribution



cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: D

Correct answer is D as Presto is a fast SQL query engine designed for interactive analytic queries over large datasets from multiple sources.

Option A is wrong as Glue is not ideal for interactive queries but more for batch ETL jobs.

Option B is wrong as it would not provide the up-to-date results as the data needs to copied over to Redshift for querying. Also, it does not cover S3 which would need Redshift Spectrum.

Option C is wrong as Spark SQL does not allow the capability to query multiple data sources. Also, Glue Developer Endpoints help test Glue ETL jobs.
upvoted 17 times

cloudlearnerhere 2 years, 7 months ago

Presto is an open-source distributed SQL query engine optimized for low-latency, ad-hoc analysis of data. It supports the ANSI SQL standard, including complex queries, aggregations, joins, and window functions. Presto can process data from multiple data sources including the Hadoop Distributed File System (HDFS) and Amazon S3.

Presto uses a custom query execution engine with operators designed to support SQL semantics. Different from Hive/MapReduce, Presto executes queries in memory, pipelined across the network between stages, thus avoiding unnecessary I/O. The pipelined execution model runs multiple stages in parallel and streams data from one stage to the next as it becomes available.

Presto supports the ANSI SQL standard, which makes it easy for data analysts and developers to query both structured and unstructured data at scale. Currently, Presto supports a wide variety of SQL functionality, including complex queries, aggregations, joins, and window functions.
upvoted 6 times

CHRIS12722222 Highly Voted 3 years, 2 months ago

Answer = D (use presto)
upvoted 8 times

pk349 Most Recent 2 years, 1 month ago

D: I passed the test
upvoted 1 times

anjuvinayan 2 years, 2 months ago

Answer is D
A-not upto date data as its glue job
B- Up-to-date data as its DMS but DMS to ES integeration not possible
C-Not interactive
upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: D

Disparate data sources are present and OpenSearch data cannot be ingested via DMS.

upvoted 1 times

rocky48 2 years, 10 months ago

Selected Answer: D

Answer = D

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: D

Answer should be D

upvoted 2 times

Bik000 3 years, 1 month ago

Selected Answer: D

Answer should be D

upvoted 2 times

MWL 3 years, 1 month ago

Selected Answer: D

For A, I didn't find document about exporting data from open search to S3.

B: DMS doesn't support to export from ES.

C: to use spark SQL to query from ES, we also need a third-party connector, so C is not complete.

D: should work.

upvoted 3 times

chp2022 3 years, 2 months ago

Selected Answer: B

I say it should be B

upvoted 1 times

AWSRanger 3 years, 2 months ago

Selected Answer: D

D is correct

upvoted 1 times

Tsyva 3 years, 2 months ago

Selected Answer: B

IMO the answer should be B since its highlighted that it must be most up to-date results and DMS supports change data capture.

upvoted 1 times

rb39 3 years, 2 months ago

Selected Answer: D

Most up-to-date -> in-place queries if possible, so Presto. Athena solution implies moving data from RDS to S3 so adds a potential delay (S3 is not real-time)

upvoted 2 times

[Removed] 3 years, 2 months ago

Selected Answer: A

JDBC connection is a key. so Athena is the answer

upvoted 1 times

Lazy_Lord 2 years, 7 months ago

You can use JDBC on the Presto running on EMR:

https://docs.aws.amazon.com/emr/latest/ReleaseGuide/presto-adding-db-connectors.html

upvoted 1 times

Question #48 *Topic 1*

A company developed a new elections reporting website that uses Amazon Kinesis Data Firehose to deliver full logs from AWS WAF to an Amazon S3 bucket.
The company is now seeking a low-cost option to perform this infrequent data analysis with visualizations of logs in a way that requires minimal development effort.
Which solution meets these requirements?

A. Use an AWS Glue crawler to create and update a table in the Glue data catalog from the logs. Use Athena to perform ad-hoc analyses and use Amazon QuickSight to develop data visualizations.

B. Create a second Kinesis Data Firehose delivery stream to deliver the log files to Amazon OpenSearch Service (Amazon Elasticsearch Service). Use Amazon ES to perform text-based searches of the logs for ad-hoc analyses and use OpenSearch Dashboards (Kibana) for data visualizations.

C. Create an AWS Lambda function to convert the logs into .csv format. Then add the function to the Kinesis Data Firehose transformation configuration. Use Amazon Redshift to perform ad-hoc analyses of the logs using SQL queries and use Amazon QuickSight to develop data visualizations.

D. Create an Amazon EMR cluster and use Amazon S3 as the data source. Create an Apache Spark job to perform ad-hoc analyses and use Amazon QuickSight to develop data visualizations.

Suggested Answer: *D*

Community vote distribution

A (100%)

Priyanka_01 Highly Voted 3 years, 9 months ago

A ? Any thoughts
https://aws.amazon.com/blogs/big-data/analyzing-aws-waf-logs-with-amazon-es-amazon-athena-and-amazon-quicksight/
upvoted 48 times

attaraya 3 years, 7 months ago

https://docs.aws.amazon.com/athena/latest/ug/waf-logs.html
upvoted 1 times

nadavw 2 years, 7 months ago

from the link: If your use case requires the analysis of data in real time, then Amazon OpenSearch Service is more suitable for your needs. If you prefer a serverless approach that doesn't require capacity planning or cluster management, then the solution with AWS Glue, Athena, and Amazon QuickSight is more suitable.
upvoted 1 times

hans1234 Highly Voted 3 years, 8 months ago

"infrequently" is a typical keyword for athena. Same as "ad-hoc" .
upvoted 16 times

apk123457890 Most Recent 1 year, 8 months ago

Why not B? the question doesn't restrict cost parameter and this will be with least development effort
upvoted 1 times

pk349 2 years, 1 month ago

A: I passed the test
upvoted 2 times

anjuvinayan 2 years, 2 months ago

Answer is A
B. ES cost is high
C. redshift cost is High
D.EMR is less costly than Glue but nothing mentioned about queries, just job

D.EMR is less costly than Glue. But development effort is more considering cluster creation
upvoted 1 times

srirnag 2 years, 4 months ago

Why not B. Not a single line of code required. It just requires a streaming config in ES and configuration of Kibana. It is B for me.

upvoted 2 times

anjuvinayan 2 years, 2 months ago

Cost of ES is high

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A. Using Glue crawler over S3 data can be used to create a data catalog, that can be queried using Athena and visualized using QuickSight. This solution does not require additional resources, data duplication and uses serverless managed services.

Option B is wrong as Elasticsearch cluster would not provide a cost-effective solution.

Option C is wrong as Redshift cluster with a Lambda job would not provide a cost-effective solution, and would also need development effort.

Option D is wrong as EMR cluster with a Spark job would not provide a cost-effective solution, and would also need development effort.

upvoted 2 times

anjuvinayan 2 years, 2 months ago

EMR is less costly than Glue. But development effort is more considering cluster creation

upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: A

"low-cost option to perform this infrequent data analysis with visualizations of logs in a way that requires minimal development effort" - This is the key. As the solution is required for infrequent analysis, so OpenSearch will be costlier solution than a combination of Athena and QuickSight.

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: A

A is the answer as its cost effective.

upvoted 1 times

jpratik1 3 years ago

Selected Answer: A

Low cost and rare

upvoted 1 times

Bik000 3 years, 1 month ago

Either A or B

upvoted 1 times

Jayproton 3 years, 1 month ago

Selected Answer: A

Per this https://aws.amazon.com/blogs/big-data/analyzing-aws-waf-logs-with-amazon-es-amazon-athena-and-amazon-quicksight/ A should be answer

upvoted 1 times

Crypt0zknight 3 years, 6 months ago

A

https://docs.aws.amazon.com/athena/latest/ug/when-should-i-use-ate.html

Integrates easily with Glue and Quicksight

upvoted 1 times

aws2019 3 years, 7 months ago

A is prefered

upvoted 2 times

Chints01 3 years, 7 months ago

Answer is A if the ask is low-cost (asked in this question)

B - this should be the answer if the ask is to have the best solution for log analysis (this is not cost efficient)

C and D can be ruled out for obvious reasons of including so many services when the requirement can be met without them

upvoted 1 times

Donell 3 years, 8 months ago

Answer A

upvoted 1 times

Brijeshkrishna 3 years, 8 months ago

A is the answer as its cost effective. Athena and Quicksight Standard are very cost effective compared to Elasticsearch Cluster which is expensive.

upvoted 2 times

Question #49 *Topic 1*

A large company has a central data lake to run analytics across different departments. Each department uses a separate AWS account and stores its data in an
Amazon S3 bucket in that account. Each AWS account uses the AWS Glue Data Catalog as its data catalog. There are different data lake access requirements based on roles. Associate analysts should only have read access to their departmental data. Senior data analysts can have access in multiple departments including theirs, but for a subset of columns only.
Which solution achieves these required access patterns to minimize costs and administrative tasks?

A. Consolidate all AWS accounts into one account. Create different S3 buckets for each department and move all the data from every account to the central data lake account. Migrate the individual data catalogs into a central data catalog and apply fine-grained permissions to give to each user the required access to tables and databases in AWS Glue and Amazon S3.

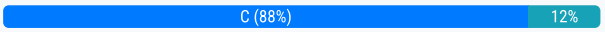
B. Keep the account structure and the individual AWS Glue catalogs on each account. Add a central data lake account and use AWS Glue to catalog data from various accounts. Configure cross-account access for AWS Glue crawlers to scan the data in each departmental S3 bucket to identify the schema and populate the catalog. Add the senior data analysts into the central account and apply highly detailed access controls in the Data Catalog and Amazon S3.

C. Set up an individual AWS account for the central data lake. Use AWS Lake Formation to catalog the cross-account locations. On each individual S3 bucket, modify the bucket policy to grant S3 permissions to the Lake Formation service-linked role. Use Lake Formation permissions to add fine-grained access controls to allow senior analysts to view specific tables and columns.

D. Set up an individual AWS account for the central data lake and configure a central S3 bucket. Use an AWS Lake Formation blueprint to move the data from the various buckets into the central S3 bucket. On each individual bucket, modify the bucket policy to grant S3 permissions to the Lake Formation service-linked role. Use Lake Formation permissions to add fine-grained access controls for both associate and senior analysts to view specific tables and columns.

Suggested Answer: *B*

Community vote distribution



Thiya Highly Voted 3 years, 7 months ago

Answer is C. I have implemented federated data lake.

upvoted 15 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: C

Correct answer is C as AWS Data Lake Formation can help provide a centralized place for maintaining data catalog to various locations, without moving the data. Also, AWS Lake Formation permissions can help provide a central access control location.

Option A is wrong as consolidating accounts would increase administrative tasks.

Option B is wrong as although it might work, it is more simpler to use AWS Lake Formation for access control.

Option D is wrong as moving all the data to central S3 would duplicate the storage cost and increase administrative tasks.

upvoted 10 times

GCPereira Most Recent 1 year, 5 months ago

accounts consolidation has a big administrative effort... then A is discarded...

B works but doesn't have permission requirements for an analyst role... then B is discarded...

if we talk about fine-grained access control and the strong power of data catalog, lake formation always is the better option... not expansive and easy to use...

a central bucket is a big administrative effort and increases storage costs due to data storage duplication... then D is discarded...

C any througs?

upvoted 1 times

pk349 2 years, 1 month ago

C: I passed the test

upvoted 1 times

uk_dataguy 2 years, 2 months ago

Selected Answer: B

Seems to be Data Lake Formation

Simplified data lake setup: Streamlines creation and configuration of a centralized data lake.

Fine-grained access control: Enables table and column-level permissions for users and groups.

Data cataloging and discovery: Facilitates a searchable, centralized data catalog using AWS Glue.

Data transformation: Supports ETL jobs to clean, enrich, and prepare data for analysis.

Integration with AWS services: Seamlessly connects with various AWS analytics and processing tools.

Security and compliance: Ensures data encryption, monitors access, and provides audit logs.

upvoted 1 times

anjuvinayan 2 years, 2 months ago

Answer is C

A-Move all data means cost and lot of effort

B-It works but Lakeformation is easy

C-Answer

D- Move all data means cost and lot of effort

upvoted 2 times

akashm99101001com 2 years, 3 months ago

Selected Answer: B

Option C is incorrect because it requires setting up an individual AWS account for the central datalake. This would be an unnecessary expense. It also requires using AWS Lake Formation to catalog the cross-account locations. This would be a time-consuming and expensive process.

AWS Lake Formation is a service that makes it easy to set up a secure data lake in days. It simplifies and automates many of the complex manual steps required to create a data lake, including collecting, cleaning, and cataloging data. You can use AWS Lake Formation to create a central data catalog that is accessible to all departments. You can use Lake Formation permissions to add fine-grained access controls to allow senior analysts to view specific tables and columns.

However, setting up an individual AWS account for the central datalake would be an unnecessary expense. It would also require additional administrative overhead to manage the different accounts.

upvoted 1 times

rags1482 2 years, 3 months ago

Option B, keeps the account structure and the individual AWS Glue catalogs on each account, but still allows for a centralized catalog using AWS Glue. It uses cross-account access for AWS Glue crawlers to scan the data in each departmental S3 bucket and identify the schema, which populates the central catalog. The senior data analysts can be added to the central account with highly detailed access controls in the Data Catalog and Amazon S3. This approach is more scalable and cost-effective in cases where there are many departments or AWS accounts involved.

Option C is a valid solution to the problem described, but it may not be the most cost-effective and efficient one. Setting up an individual AWS account for the central data lake and using AWS Lake Formation to catalog the cross-account locations with fine-grained access controls for senior analysts is a good approach, but it may involve additional administrative tasks and costs. Additionally, modifying the bucket policy for each individual S3 bucket may be cumbersome and error-prone.

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: C

Lake Formation for such fine grained access. Also, no need to use Lake Formation BluePrint as data source is S3.

upvoted 1 times

muhsin 2 years, 10 months ago

should access to the subset of columns means fine-grained access control. It can be implemented by Lake Formation, not individual S3 buckets.
So the answer is C.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: C

Selected Answer: C

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: C

Answer is C

upvoted 2 times

MWL 3 years, 1 month ago

Selected Answer: C

Use AWS Lake Formation for cross account catalog and permission.

upvoted 1 times

aws2019 3 years, 7 months ago

Answer should be C.

upvoted 1 times

yogen 3 years, 7 months ago

When did B say..to move the data.... isn't catalog of data and moving the data two different things?

upvoted 1 times

umatrilok 3 years, 7 months ago

C is the correct answer

upvoted 2 times

lostsoul07 3 years, 8 months ago

C is the right answer

upvoted 2 times

Question #50 *Topic 1*

A company wants to improve user satisfaction for its smart home system by adding more features to its recommendation engine. Each sensor asynchronously pushes its nested JSON data into Amazon Kinesis Data Streams using the Kinesis Producer Library (KPL) in Java. Statistics from a set of failed sensors showed that, when a sensor is malfunctioning, its recorded data is not always sent to the cloud.
The company needs a solution that offers near-real-time analytics on the data from the most updated sensors.
Which solution enables the company to meet these requirements?

A. Set the RecordMaxBufferedTime property of the KPL to "1'^ג" to disable the buffering on the sensor side. Use Kinesis Data Analytics to enrich the data based on a company-developed anomaly detection SQL script. Push the enriched data to a fleet of Kinesis data streams and enable the data transformation feature to flatten the JSON file. Instantiate a dense storage Amazon Redshift cluster and use it as the destination for the Kinesis Data Firehose delivery stream.

B. Update the sensors code to use the PutRecord/PutRecords call from the Kinesis Data Streams API with the AWS SDK for Java. Use Kinesis Data Analytics to enrich the data based on a company-developed anomaly detection SQL script. Direct the output of KDA application to a Kinesis Data Firehose delivery stream, enable the data transformation feature to flatten the JSON file, and set the Kinesis Data Firehose destination to an Amazon OpenSearch Service (Amazon Elasticsearch Service) cluster.

C. Set the RecordMaxBufferedTime property of the KPL to "0" to disable the buffering on the sensor side. Connect for each stream a dedicated Kinesis Data Firehose delivery stream and enable the data transformation feature to flatten the JSON file before sending it to an Amazon S3 bucket. Load the S3 data into an Amazon Redshift cluster.

D. Update the sensors code to use the PutRecord/PutRecords call from the Kinesis Data Streams API with the AWS SDK for Java. Use AWS Glue to fetch and process data from the stream using the Kinesis Client Library (KCL). Instantiate an Amazon Elasticsearch Service cluster and use AWS Lambda to directly push data into it.

Suggested Answer: *A*

Community vote distribution

Priyanka_01 Highly Voted 3 years, 9 months ago

B?
https://aws.amazon.com/blogs/big-data/perform-near-real-time-analytics-on-streaming-data-with-amazon-kinesis-and-amazon-elasticsearch-service/
upvoted 21 times

GCPereira 1 year, 5 months ago

agreed
upvoted 1 times

awssp12345 Highly Voted 3 years, 9 months ago

When Not to Use the KPL :

The KPL can incur an additional processing delay of up to RecordMaxBufferedTime within the library (user-configurable). Larger values of RecordMaxBufferedTime results in higher packing efficiencies and better performance. Applications that cannot tolerate this additional delay may need to use the AWS SDK directly. For more information about using the AWS SDK with Kinesis Data Streams, see Developing Producers Using the Amazon Kinesis Data Streams API with the AWS SDK for Java. For more information about RecordMaxBufferedTime and other user-configurable properties of the KPL, see Configuring the Kinesis Producer Library.

https://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html
upvoted 13 times

awssp12345 3 years, 9 months ago

I agree the answer is B.
upvoted 14 times

pk349 Most Recent 2 years, 1 month ago

B: I passed the test
upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B. Using PutRecord/PutRecords would send the data synchronously to Kinesis Data Streams. Kinesis Data Analytics can be detect anomalies and the data can be pushed to Kinesis Data Firehose with transformation to Elasticsearch for analysis.
Option A is wrong as Kinesis data streams does not provide data transformation feature.

Option C is wrong as copying data to S3 and loading to Redshift would not make it near-real time.

Option D is wrong as using Glue is ideal for batch jobs and not for near-real time analytics.
upvoted 7 times

dushmantha 2 years, 11 months ago

Selected Answer: B

Low latency retrival can be achieved with DynamoDB, Redis and OpenSearch. I guess that would be enough to select the answer.
upvoted 1 times

dushmantha 2 years, 11 months ago

D can be eleminated coz, KCL can't read from SDK producer
upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: B

Answer - B
upvoted 1 times

treeli 3 years ago

Selected Answer: B

near realtime should be opensearch
upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer is B
upvoted 1 times

jrheen 3 years, 2 months ago

Answer - B
upvoted 1 times

rb39 3 years, 3 months ago

B - near-realtime analytics keyword, only ES can provide that from the set of options
upvoted 2 times

aws2019 3 years, 7 months ago

B it is
upvoted 1 times

Donell 3 years, 7 months ago

Answer B
upvoted 1 times

AjithkumarSL 3 years, 7 months ago

Looks like Answer is B..

https://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html

When Not to Use the KPL
The KPL can incur an additional processing delay of up to RecordMaxBufferedTime within the library (user-configurable). Larger values of RecordMaxBufferedTime results in higher packing efficiencies and better performance. Applications that cannot tolerate this additional delay may need to use the AWS SDK directly.
upvoted 1 times

umatrilok 3 years, 8 months ago

B is the answer
upvoted 1 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 2 times

gtourkas 3 years, 8 months ago

The only thing about B is why transform to CSV since putting to Elastic Search ?

upvoted 1 times

freaky 3 years, 8 months ago

Answer B

A and C dropped because RedShift is not meant for "near-real-time" analysis. Also, it would require some kind on Visualization on top of it to do the analysis.

Dropped D because KDS in itself cannot transform data.

upvoted 2 times

Question #51 *Topic 1*

A global company has different sub-organizations, and each sub-organization sells its products and services in various countries. The company's senior leadership wants to quickly identify which sub-organization is the strongest performer in each country. All sales data is stored in Amazon S3 in Parquet format.
Which approach can provide the visuals that senior leadership requested with the least amount of effort?

A. Use Amazon QuickSight with Amazon Athena as the data source. Use heat maps as the visual type.

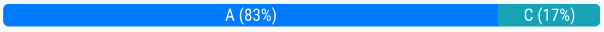
B. Use Amazon QuickSight with Amazon S3 as the data source. Use heat maps as the visual type.

C. Use Amazon QuickSight with Amazon Athena as the data source. Use pivot tables as the visual type.

D. Use Amazon QuickSight with Amazon S3 as the data source. Use pivot tables as the visual type.

Suggested Answer: *C*

Community vote distribution



ramozo Highly Voted 3 years, 9 months ago
It is A. QuickSight does not support S3 files with parquet format, Athena does it. For visualization is better a graph than a pivot table.
https://docs.aws.amazon.com/athena/latest/ug/when-should-i-use-ate.html
https://docs.aws.amazon.com/quicksight/latest/user/supported-data-sources.html
upvoted 34 times

vicks316 3 years, 8 months ago
100% agree with A
upvoted 1 times

AjNapa 3 years, 8 months ago
Why not B. Why cant S3 be the datasource for Quicksight
upvoted 2 times

AjNapa 3 years, 8 months ago
Sorry just noticed that you've mentioned Quicksight doesn't support files in S3 in parquet format
upvoted 3 times

sam202033 3 years, 8 months ago
I agree. It should be B
https://aws.amazon.com/quicksight/
https://d1.awsstatic.com/r2018/h/QuickSight%20Q/Provide%20interactive%20dashboards_QuickSight.7efb01bbe9a6b6592a821bccff04f463647afd82
upvoted 2 times

sam202033 3 years, 8 months ago
You can use any of the following relational data stores as data sources for Amazon QuickSight:

Amazon Athena

Amazon Aurora

Amazon Redshift

Amazon Redshift Spectrum

Amazon S3

Amazon S3 Analytics
upvoted 2 times

AWS_Trial 3 years, 8 months ago
Agree with you. QuickSight supports S3. the data is already on S3. I don't see need for Anthena in this scenario.

upvoted 1 times

Draco31 3 years, 8 months ago

No. https://docs.aws.amazon.com/quicksight/latest/user/supported-data-sources.html

Parquet files not supported by Quicksight, must use Athena.

upvoted 9 times

lakediver 3 years, 6 months ago

I am inclined towards C

Heat maps and pivot tables display data in a similar tabular fashion. Use a heat map if you want to identify trends and outliers, because the use of color makes these easier to spot. Use a pivot table if you want to further analyze data on the visual, for example by changing column sort order or applying aggregate functions across rows or columns.

https://docs.aws.amazon.com/quicksight/latest/user/heat-map.html

upvoted 4 times

testtaker3434 Highly Voted 3 years, 9 months ago

Agree wih C. Thoughts?

upvoted 10 times

Paitan 3 years, 8 months ago

Totally agree :-)

upvoted 2 times

gofavad926 Most Recent 1 year, 8 months ago

A or C? For me both are valid

- QuickSight does not support S3 files with parquet format.

- AWS says here https://docs.aws.amazon.com/quicksight/latest/user/pivot-table.html "Heat maps and pivot tables display data in a similar tabular fashion. Use a heat map if you want to identify trends and outliers, because the use of color makes these easier to spot. Use a pivot table if you want to analyze data on the visual."

upvoted 1 times

nroopa 1 year, 10 months ago

Ans Is C . Athena as the data is in Parquet format and Pivot table as it was to identify sub-organization is the strongest performer in each country whereas heat map will provide only correlation

upvoted 1 times

pk349 2 years, 1 month ago

B: I passed the test

upvoted 2 times

anjuvinayan 2 years, 2 months ago

Answer is A

Since Parquet is mentioned then Athena.

Visuals means heat map, Pivot table is just a table with Data

upvoted 1 times

anjuvinayan 2 years, 2 months ago

Answer is A

Since Parquet is mentioned then Athena.

Visuals means heat map, Pivot table is just a table with Data

upvoted 1 times

akashm99101001com 2 years, 3 months ago

Selected Answer: A

The correct answer is A. Here's why:

A is correct because Amazon QuickSight can be used with Amazon Athena as the data source to visualize sales data stored in Amazon S3 in Parquet format. Heat maps can be used as the visual type to quickly identify which sub-organization is the strongest performer in each country. Heat maps are a great way to visualize data that is organized in a grid format, such as sales data.

C is incorrect because pivot tables are not the best visual type to use for this scenario. Pivot tables are great for summarizing and analyzing large amounts of data, but they are not the best way to visualize data.

upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A

Correct answer is A as Quicksight can use Athena as data source to query the data in S3. Heat Maps would provide the required visual representation.

Heat maps and pivot tables display data in a similar tabular fashion. Use a heat map if you want to identify trends and outliers, because the use of color makes these easier to spot. Use a pivot table if you want to analyze data on the visual.

Option C is wrong as Heat Map would provide the required visual and would be preferred over Pivot tables as there is not need to analyze the data.

Options B & D are wrong as Quicksight supports S3 as a data source, however it does not work directly with parquet file format.

upvoted 5 times

Arka_01 2 years, 9 months ago

Selected Answer: A

Quickly identification is possible by HeatMap color coding feature. Pivot table is unnecessary here. Athena is required as data is stored in Parquet format.

upvoted 1 times

he11ow0rld 2 years, 9 months ago

Selected Answer: C

c, heat maps are for correlation (

upvoted 1 times

redwan123 2 years, 10 months ago

Selected Answer: C

heat maps are for correlation (P.S.: heat maps do not use a world map) -> Pivot tables is correct here.

quick-sight cannot directly query parquet files.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: A

Selected Answer: A

upvoted 1 times

GiveMeEz 3 years ago

Ans is B.
Heat map allows one simple chart to show the pattern for the whole thing easily.

QuickSight can use S3 as data source.
Creating a dataset using Amazon S3 files - Amazon QuickSight
https://docs.aws.amazon.com/quicksight/latest/user/create-a-data-set-s3.html

upvoted 1 times

certificationJunkie 3 years, 1 month ago

Ans is C. Heat map is to show co-relation and identify outliers. Here the requirement is very straightforward which is to identify best performer in each region. Hence Pivot table.

upvoted 3 times

Bik000 3 years, 1 month ago

Selected Answer: A

My Answer is A

upvoted 1 times

Engy2020 3 years, 3 months ago

swiftly determine which sub-organization is the best performance in each country

C: is the answer , because (in each country)you should use pivot table

upvoted 3 times

Question #52 *Topic 1*

A company has 1 million scanned documents stored as image files in Amazon S3. The documents contain typewritten application forms with information including the applicant first name, applicant last name, application date, application type, and application text. The company has developed a machine learning algorithm to extract the metadata values from the scanned documents. The company wants to allow internal data analysts to analyze and find applications using the applicant name, application date, or application text. The original images should also be downloadable. Cost control is secondary to query performance.
Which solution organizes the images and metadata to drive insights while meeting the requirements?

A. For each image, use object tags to add the metadata. Use Amazon S3 Select to retrieve the files based on the applicant name and application date.

B. Index the metadata and the Amazon S3 location of the image file in Amazon OpenSearch Service (Amazon Elasticsearch Service). Allow the data analysts to use OpenSearch Dashboards (Kibana) to submit queries to the Amazon OpenSearch Service (Amazon Elasticsearch Service) cluster.

C. Store the metadata and the Amazon S3 location of the image file in an Amazon Redshift table. Allow the data analysts to run ad-hoc queries on the table.

D. Store the metadata and the Amazon S3 location of the image files in an Apache Parquet file in Amazon S3, and define a table in the AWS Glue Data Catalog. Allow data analysts to use Amazon Athena to submit custom queries.

Suggested Answer: *A*

Community vote distribution

B (88%) 13%

rb39 Highly Voted 3 years, 2 months ago

Selected Answer: B

OpenSearch to scan all text

upvoted 12 times

skb0071 Most Recent 1 year, 7 months ago

Answer is C

Store the metadata in Redshift. Metadata is extracted using company provided ML program.

upvoted 1 times

Debi_mishra 2 years, 1 month ago

B is the correct answer. Keywords to look for in question - "Performance" and "search using text". D can be correct only if there is no text based search requirement.

upvoted 1 times

pk349 2 years, 1 month ago

B: I passed the test

upvoted 1 times

akashm99101001com 2 years, 3 months ago

Selected Answer: B

Option A is incorrect because object tags are not searchable and cannot be used to query the data. S3 Select can be used to retrieve the files based on the applicant name and application date, but object tags cannot be used to store metadata.

Option B is correct because Amazon OpenSearch Service (Amazon Elasticsearch Service) can be used to index the metadata and the Amazon S3 location of the image file. Data analysts can use OpenSearch Dashboards (Kibana) to submit queries to the Amazon OpenSearch Service (Amazon Elasticsearch Service) cluster.

Option C is incorrect because Amazon Redshift is not designed for storing large binary objects such as images. It is a data warehousing solution that is optimized for querying structured data.

Option D is incorrect because Apache Parquet files are not optimized for querying unstructured data such as images. Amazon Athena can be used to submit custom queries, but it is not optimized for querying large binary objects.

upvoted 3 times

rags1482 2 years, 3 months ago

D is the right answer

in Option B there is no direct method provided in this option to download the image file(s) associated with the search results.
upvoted 3 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B as the metadata can be indexed with the S3 file location in ElasticSearch to provide a quick search and allow the users to download the file as well.

https://aws.amazon.com/blogs/machine-learning/automatically-extract-text-and-structured-data-from-documents-with-amazon-textract/

Option A is wrong as using S3 Select would impact query performance.

Option C is wrong as it would have a huge cost impact without improving query performance much.

Option D is wrong as using Athena would impact query performance.
upvoted 4 times

JHJHJHJHJ 2 years, 9 months ago

Answer A: Validated using Jon bosco paid dumps
upvoted 2 times

JoellaLi 2 years, 8 months ago

But why A?
upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: B

Cost control is secondary to query performance - This is the key here. Though D also can do the work, but it will be slower than option B.
upvoted 2 times

rrshah83 2 years, 10 months ago

Selected Answer: D

Parquet format improves performance. None of the other options talk about performance improvement.
upvoted 3 times

Gavin_Y 2 years, 10 months ago

and 'Cost control is secondary to query performance.'
upvoted 2 times

Question #53 *Topic 1*

A mobile gaming company wants to capture data from its gaming app and make the data available for analysis immediately. The data record size will be approximately 20 KB. The company is concerned about achieving optimal throughput from each device. Additionally, the company wants to develop a data stream processing application with dedicated throughput for each consumer.
Which solution would achieve this goal?

A. Have the app call the PutRecords API to send data to Amazon Kinesis Data Streams. Use the enhanced fan-out feature while consuming the data.

B. Have the app call the PutRecordBatch API to send data to Amazon Kinesis Data Firehose. Submit a support case to enable dedicated throughput on the account.

C. Have the app use Amazon Kinesis Producer Library (KPL) to send data to Kinesis Data Firehose. Use the enhanced fan-out feature while consuming the data.

D. Have the app call the PutRecords API to send data to Amazon Kinesis Data Streams. Host the stream-processing application on Amazon EC2 with Auto Scaling.

Suggested Answer: *D*

Community vote distribution

A (100%)

zanhsieh Highly Voted 3 years, 9 months ago

A
Dedicated throughput equals enhanced fan-out. So BCD dropped.
https://docs.aws.amazon.com/streams/latest/dev/enhanced-consumers.html
upvoted 30 times

Mishra123 3 years, 8 months ago

A seems correct
upvoted 2 times

korcaptain Highly Voted 3 years, 8 months ago

A: Developing Custom Consumers with Dedicated Throughput (Enhanced Fan-Out)
upvoted 6 times

pk349 Most Recent 2 years, 1 month ago

A: I passed the test
upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: A
Correct answer is A as Kinesis Data Streams with Enhanced Fanout provides dedicated throughput for each consumer.

In Amazon Kinesis Data Streams, you can build consumers that use a feature called enhanced fan-out. This feature enables consumers to receive records from a stream with throughput of up to 2 MB of data per second per shard. This throughput is dedicated, which means that consumers that use enhanced fan-out don't have to contend with other consumers that are receiving data from the stream. Kinesis Data Streams pushes data records from the stream to consumers that use enhanced fan-out. Therefore, these consumers don't need to poll for data.

Option B is wrong as there is no option to open support case to enable dedicated throughput.

Option C is wrong as Kinesis Data Firehose does not support enhanced fan-out feature.

D is incorrect. An Auto Scaling group of EC2 instances will not provide dedicated throughput for the consumers. You have to enable the enhanced fan-out feature in Amazon Kinesis Data Streams.
upvoted 4 times

Arka_01 2 years, 9 months ago

Selected Answer: A

"dedicated throughput for each consumer" - this is the key statement here.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: A

Selected Answer: A

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: A

Answer is A

upvoted 1 times

Naresh_Dulam 3 years, 7 months ago

Have the app call the PutRecords API to send data to Amazon Kinesis Data Streams. Use the enhanced fan-out feature while consuming the data. ==> PutRecords aggregate while sending data to Kinesis Data Streams to increase producer throghuput and Enhanced fan-out increase consumer through put

B. Have the app call the PutRecordBatch API to send data to Amazon Kinesis Data Firehose. Submit a support case to enable dedicated throughput on the account. ==> Question is about stream processing using producer and consumer.

C. Have the app use Amazon Kinesis Producer Library (KPL) to send data to Kinesis Data Firehose. Use the enhanced fan-out feature while consuming the data. ==> Enhanced fanout feature is not part of Firehose

D. Have the app call the PutRecords API to send data to Amazon Kinesis Data Streams. Host the stream-processing application on Amazon EC2 with Auto Scaling. ==> We don't need EC@

upvoted 4 times

lostsoul07 3 years, 7 months ago

A is the right answer

upvoted 1 times

Draco31 3 years, 7 months ago

A.

Whatever how you consume your data (EC2 in ASG or not), the stream must have Enhanced fan out.

upvoted 1 times

sanjaym 3 years, 8 months ago

A for sure.

upvoted 1 times

sam202033 3 years, 8 months ago

Answer is A

There is no limit for the enhanced fan out . Check this link..
https://aws.amazon.com/kinesis/data-streams/faqs/

Q: Is there a limit on the number of consumers using enhanced fan-out on a given stream?

There is a default limit of 20 consumers using enhanced fan-out per data stream. If you need more than 20, please submit a limit increase request though AWS support. Keep in mind that you can have more than 20 total consumers reading from a stream by having 20 consumers using enhanced fan-out and other consumers not using enhanced fan-out at the same time.

upvoted 1 times

manish9363 3 years, 8 months ago

How A is even possible ! enhanced fan out has limitation of 20 consumers.

I will go with D

upvoted 1 times

giocal 3 years, 8 months ago

Enhanced fan out has a limit of 20 consumer per stream (not shard, stream). In a case of thousand or maybe millions devices I don't think is a valid solution.
I'll probably go with D here.

upvoted 1 times

jove 3 years, 7 months ago

I think the gaming devices are producers not consumers in the given use case.

upvoted 3 times

JoellaLi 2 years, 8 months ago

"A consumer is an application that processes all data from a Kinesis data stream. "

upvoted 1 times

Paitan 3 years, 8 months ago

Option A is the right choice.

upvoted 2 times

Nicki1013 3 years, 8 months ago

A is correct

upvoted 1 times

Question #54 *Topic 1*

A marketing company wants to improve its reporting and business intelligence capabilities. During the planning phase, the company interviewed the relevant stakeholders and discovered that:
➩ The operations team reports are run hourly for the current month's data.
➩ The sales team wants to use multiple Amazon QuickSight dashboards to show a rolling view of the last 30 days based on several categories. The sales team also wants to view the data as soon as it reaches the reporting backend.
➩ The finance team's reports are run daily for last month's data and once a month for the last 24 months of data.
Currently, there is 400 TB of data in the system with an expected additional 100 TB added every month. The company is looking for a solution that is as cost- effective as possible.
Which solution meets the company's requirements?

A. Store the last 24 months of data in Amazon Redshift. Configure Amazon QuickSight with Amazon Redshift as the data source.

B. Store the last 2 months of data in Amazon Redshift and the rest of the months in Amazon S3. Set up an external schema and table for Amazon Redshift Spectrum. Configure Amazon QuickSight with Amazon Redshift as the data source.

C. Store the last 24 months of data in Amazon S3 and query it using Amazon Redshift Spectrum. Configure Amazon QuickSight with Amazon Redshift Spectrum as the data source.

D. Store the last 2 months of data in Amazon Redshift and the rest of the months in Amazon S3. Use a long-running Amazon EMR with Apache Spark cluster to query the data as needed. Configure Amazon QuickSight with Amazon EMR as the data source.

Suggested Answer: *B*

Community vote distribution

B (100%)

ramozo Highly Voted 3 years, 9 months ago

For me is B. Redshift offers better performance for querying and analyzing latest 2 months of data and in combination with Spectrum for non-frequent queries on 24 months of data.
upvoted 41 times

Katana19 3 years, 8 months ago

they didnt require performance, the required cost-effectiveness !! 2 months of data means 200TB... a redshift cluster of 200TB is not cheap !!!
upvoted 5 times

Gavin_Y 2 years, 10 months ago

it's mentioned 'The sales team also wants to view the data as soon as it reaches the reporting backend', so I think there require perfomance
upvoted 2 times

kempstonjoystick Highly Voted 3 years, 8 months ago

https://aws.amazon.com/premiumsupport/knowledge-center/redshift-spectrum-query-charges/

"Load the data in S3 and use Redshift Spectrum if the data is infrequently accessed."
In this case, the operations data is accessed hourly; this is not infrequent. I think even with the statement of cost effective, the answer is B. IF all the monthly data of 100TB is scanned hourly as part of those reports (and there's nothing in the question to say it isn't), then the cost becomes 100TB * $5 * ~720 hours in a month, which is $360,000 per month! The storage costs for 200TB of Redshift data is $5000 a month.
upvoted 15 times

pk349 Most Recent 2 years, 1 month ago

B: I passed the test
upvoted 1 times

AwsNewPeople 2 years, 3 months ago

Selected Answer: B

Option B seems to be the best solution for this scenario. It suggests storing the last 2 months of data in Amazon Redshift and the rest of the months in Amazon S3. Setting up an external schema and table for Amazon Redshift Spectrum will allow for querying data stored in Amazon S3. Additionally, configuring Amazon QuickSight with Amazon Redshift as the data source will allow for creating reports and dashboards for the data.

This solution is cost-effective because it uses Amazon S3 to store the majority of the data, which is cheaper than storing it all in Amazon Redshift. Also, it leverages Amazon Redshift Spectrum, which allows for querying data in Amazon S3 using a standard SQL interface without needing to move

the data into Amazon Redshift. Finally, storing only two months of data in Amazon Redshift will minimize storage costs in Redshift while still allowing for fast query performance for the most recent data.

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B as the base requirements are cost and performance, keeping data in Redshift for 2 months would allow data analysis for current and previous month. Holding data for 24 months in S3 would provide a cost-effective option.

Option A is wrong as holding 24 months data in Redshift is not cost-effective.

Option C is wrong as storing 24 months of data in S3 would not provide performance.

Option D is wrong as using a long-running EMR cluster is not cost-effective.

upvoted 4 times

Arka_01 2 years, 9 months ago

Selected Answer: B

"as cost- effective as possible" - this is the key statement here. So we need fast retrieval and query performance on last 2 months data, and infrequent querying capability for last 24 months of data. So B is the correct answer.

upvoted 1 times

dushmantha 2 years, 11 months ago

Selected Answer: B

I would choose "B", although I had doubts to choose "C". The main reason for the switch is that, its not a very good use case of using Redshift Spectrum without using Redshift for any part of the job, I don't know if its possible. Ideally Redshift suppose to query hot data and Redshift Spectrum supposed to extend the querying capability to exabytes of data

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: B

B is the right answer.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

B should be Correct

upvoted 1 times

jrheen 3 years, 2 months ago

Answer - B

upvoted 1 times

Blueocean 3 years, 4 months ago

Agree B is the best and most cost effective option

upvoted 1 times

GoKhe 3 years, 5 months ago

B is the correct answer

upvoted 1 times

lixin2402 3 years, 7 months ago

Definitely, B is the right one. The cost was already being cut in half. EMR long-running instance is not cheap.

upvoted 1 times

aws2019 3 years, 7 months ago

B is the right answer

upvoted 1 times

goutes 3 years, 7 months ago

RD Spectrum can query exabytes of unstructured data in S3 without loading. It even supports gzip and snappy. So option C is correct.

upvoted 3 times

jueueuergen 3 years, 8 months ago

I think answer B is correct because of the hourly scanning costs.
However, I think we don't have enough information:
- what are the usage patterns? is only a small subset of columns required? -> less scanning
- what compression factor is possible? Spectrum seems to support compressed data, whereas data in Redshift seems to be uncompressed
[Compression factor] * [column selection] can easily decrease the amount of data that needs to be scanned by a factor of 100x, possibly even 1000x or more.

Finally, the phrasing "The sales team also wants to view the data as soon as it reaches the reporting backend." could go either way if you ask me - Spectrum doesn't introduce a lag because data is loaded lazily, however it leads to slower queries compared to Redshift.
upvoted 3 times

Huy 3 years, 8 months ago
B. One more thing that make C wrong is Spectrum only runs within a Redshift Cluster. Therefore you are both charged by the cluster and the data scanned.
upvoted 1 times

Question #55 *Topic 1*

A media company wants to perform machine learning and analytics on the data residing in its Amazon S3 data lake. There are two data transformation requirements that will enable the consumers within the company to create reports:

➪ Daily transformations of 300 GB of data with different file formats landing in Amazon S3 at a scheduled time.

➪ One-time transformations of terabytes of archived data residing in the S3 data lake.

Which combination of solutions cost-effectively meets the company's requirements for transforming the data? (Choose three.)

A. For daily incoming data, use AWS Glue crawlers to scan and identify the schema.

B. For daily incoming data, use Amazon Athena to scan and identify the schema.

C. For daily incoming data, use Amazon Redshift to perform transformations.

D. For daily incoming data, use AWS Glue workflows with AWS Glue jobs to perform transformations.

E. For archived data, use Amazon EMR to perform data transformations.

F. For archived data, use Amazon SageMaker to perform data transformations.

Suggested Answer: *BCD*

Community vote distribution

ADE (100%)

testtaker3434 Highly Voted 3 years, 9 months ago

To me, ADE.

Not B. Athena will use Glue (option A)

Not C. Its an antipattern to use Redshift to do transformations.

Not F. Would pick EMR instead of Sagemaker to do one time transformations

upvoted 46 times

awssp12345 3 years, 9 months ago

Agreed

upvoted 3 times

zeronine Highly Voted 3 years, 9 months ago

My answer is ADE.

upvoted 9 times

pk349 Most Recent 2 years, 1 month ago

ADE: I passed the test

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: ADE

Correct answers are A, D & E

Options A & D using Glue Crawler and Glue Workflows would provide ETL for daily transactions.

Option E as EMR can help perform data transformation for archived data.

Option B is wrong as Athena does not identify the schema but uses Glue Catalog.

Option C is wrong as Redshift would need to be persistent and does not provide a cost-effective solution as compared to Glue.

Option F is wrong as Amazon SageMaker is a fully managed service that provides every developer and data scientist with the ability to build, train, and deploy machine learning (ML) models quickly. It does not provide ETL capability on large data.

upvoted 6 times

sevensquare 2 years, 7 months ago

What about SageMaker Data Wrangler?

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: ADE

cost-effectively solution is required. So, A, D and E.

upvoted 1 times

rocky48 2 years, 10 months ago

Selected Answer: ADE

Selected Answer: ADE

upvoted 1 times

girish123456 2 years, 11 months ago

Selected Answer: ADE

A: For schema and new partition of data for Incremental load

D: Incremental transformation

E: Historical data migration using EMR

upvoted 2 times

GiveMeEz 3 years ago

sorry, for the 41 upvotes. Ans A can't be it. Athena doesn't scan and identify schema. Athena use the Glue Data Catalog, which is generated by Glue Crawler.

My answer: A,D,E.

upvoted 2 times

aws2019 3 years, 7 months ago

ADE is ans

upvoted 1 times

lostsoul07 3 years, 8 months ago

A, D, E is the right answer

upvoted 5 times

Draco31 3 years, 8 months ago

Yep ADE also.

I guess SageMaker will use the data more than 1 time for learning processes

upvoted 3 times

sanjaym 3 years, 8 months ago

100% ADE

upvoted 5 times

syu31svc 3 years, 8 months ago

Notice that the answers given are paired so if you were to break it down:

Identify schema --> Glue

Transformations --> Glue Jobs

Archived TBs worth of data --> EMR

So is ADE

upvoted 5 times

Paitan 3 years, 8 months ago

A, D and E.

upvoted 5 times

manish9363 3 years, 9 months ago

can glue handle 300GB data every day? It seems too much for glue.

upvoted 1 times

GiveMeEz 3 years ago

glue can. you can properly size the glue cluster for the glue job with one simple dial.

upvoted 1 times

Phoenyx89 3 years, 8 months ago

Absolutely! Glue can handle same amount of data as EMR because in the end Glue is a simplified EMR cluster with Spark, HDFS, YARN and the Glue dependencies but have the advantage of being serverless. Configuring the appropriate amount and type of DPUs you can handle 300GB of data

upvoted 8 times

omar_bahrain 3 years, 8 months ago

Thanks for describing the origion of EMR

upvoted 1 times

Question #56 *Topic 1*

A hospital uses wearable medical sensor devices to collect data from patients. The hospital is architecting a near-real-time solution that can ingest the data securely at scale. The solution should also be able to remove the patient's protected health information (PHI) from the streaming data and store the data in durable storage.
Which solution meets these requirements with the least operational overhead?

A. Ingest the data using Amazon Kinesis Data Streams, which invokes an AWS Lambda function using Kinesis Client Library (KCL) to remove all PHI. Write the data in Amazon S3.

B. Ingest the data using Amazon Kinesis Data Firehose to write the data to Amazon S3. Have Amazon S3 trigger an AWS Lambda function that parses the sensor data to remove all PHI in Amazon S3.

C. Ingest the data using Amazon Kinesis Data Streams to write the data to Amazon S3. Have the data stream launch an AWS Lambda function that parses the sensor data and removes all PHI in Amazon S3.

D. Ingest the data using Amazon Kinesis Data Firehose to write the data to Amazon S3. Implement a transformation AWS Lambda function that parses the sensor data to remove all PHI.

Suggested Answer: *C*
Reference:
https://aws.amazon.com/blogs/big-data/persist-streaming-data-to-amazon-s3-using-amazon-kinesis-firehose-and-aws-lambda/

Community vote distribution
D (100%)

Prodip Highly Voted 3 years, 9 months ago
D; transformation AWS Lambda function applied with stream data; before loading to s3
upvoted 37 times

carol1522 3 years, 9 months ago
Changed to D
upvoted 2 times

awssp12345 3 years, 9 months ago
Agreed
upvoted 2 times

certificationJunkie 3 years, 1 month ago
where does it say 'before' in option D ?
upvoted 2 times

lpc01 3 years, 5 months ago
Why is it not B?
upvoted 1 times

cnmc 3 years, 3 months ago
Because B removes the PHI *after* it is stored into S3. The question asks that PHI is removed from "streaming data", and it is also better practice to remove sensitive info before reaching storage
upvoted 5 times

pk349 Most Recent 2 years, 1 month ago
D: I passed the test
upvoted 1 times

nadavw 2 years, 7 months ago
D as IOT rules can send info to Firehose using an action https://docs.aws.amazon.com/firehose/latest/dev/writing-with-iot.html
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: D
Correct answer is D as Kinesis Data Firehose can be used for data ingestion and storage to S3 with Lambda function for data filtering and transformation. This solution involves the least operational overhead.

Option A is wrong as Kinesis Data Streams and KCLs involve operational overhead as Data Streams need to be provisioned and maintained.

Option B is wrong as the solution removes the PHI after the data is stored.

Option C is wrong as Kinesis Data Streams does not integrate with S3 directly and involves the operational overhead as Data Streams need to be provisioned and maintained.

upvoted 3 times

thirukudil 2 years, 8 months ago

Selected Answer: D

Ans is D.
Solution need least operational overhead, so kinesis data stream is out. option B is also out bcoz it is removing PHI data after putting the data in S3. So Option D is correct. Firehose will do the transformation via lambda to filter out the PHI data from stream and store the non-PHI in S3.

upvoted 2 times

Arka_01 2 years, 9 months ago

Selected Answer: D

AWS Kinesis Data Firehose is required as destination is S3. Also, Lambda function should be called as a transformation from Firehose before sending data to S3.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: D

Selected Answer: D

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: D

My Answer is D

upvoted 2 times

Donell 3 years, 7 months ago

Answer D.
Its true that the answer wordings are bad and confusing.
Kinesis Data Firehose can invoke your Lambda function to transform incoming source data and deliver the transformed data to destinations. You can enable Kinesis Data Firehose data transformation when you create your delivery stream.

upvoted 2 times

Heer 3 years, 7 months ago

ANSWER D:
EXPLAINATION:With the keyword 'near-real-time',option A and C are filtered out as KDS is real time streaming .NOw between option B & D ,We already have transformation lambda attached to 'Firehose' by default to do the necessary transformation .

upvoted 2 times

DerekKey 3 years, 7 months ago

Correct D
KDF uses transformation (Lambda) before writing to S3
Incorrect A
Implementing Lambda to process and write data to S3 with streams is crazy. You should use KDF as KDS consumer. Shuld be least operational overhead.
Incorrect B, C - security of data

upvoted 1 times

AjithkumarSL 3 years, 8 months ago

Going with D, Even the reference in the answer also pointing the same (Reference: https://aws.amazon.com/blogs/big-data/persist-streaming-data-to-amazon-s3-using-amazon-kinesis-firehose-and-aws-lambda/)

upvoted 1 times

lostsoul07 3 years, 8 months ago

D is the right answer

upvoted 2 times

kempstonjoystick 3 years, 8 months ago

Badly worded answer for D, but that's the correct answer here.

upvoted 1 times

Roontha 3 years, 8 months ago

Answer : D

Refer : https://aws.amazon.com/blogs/compute/amazon-kinesis-firehose-data-transformation-with-aws-lambda/#:~:text=Introducing%20Firehose%20Data%20Transformations&text=When%20you%20enable%20Firehose%20data,then%20delivered%20to%20the%2

upvoted 1 times

hans1234 3 years, 8 months ago

I think D means that the data is cleaned already before writing to s3, but the formulation is bad.

upvoted 2 times

Manue 3 years, 8 months ago

Agree, formulation is confusing. I guess it means lambda transformation is applied before wrting to S3, so D is right.

upvoted 1 times

sanjaym 3 years, 8 months ago

Answer is D.

It's near real-time so no need to use KDS

A - task can be handled much easier (less complicated) way by D then A.

B - PHI data is written to S3 before removal which is not acceptable.

C - KDS cannot write data to S3.

upvoted 3 times

LMax 3 years, 8 months ago

Agree, D

upvoted 1 times

Roontha 3 years, 8 months ago

Hi Sanjay

have you completed data analytic professional exam recently

upvoted 1 times

ricksun 3 years, 8 months ago

disagree, I'll go A since firehose natually integrated with lambda.

upvoted 2 times

Question #57 *Topic 1*

A company is migrating its existing on-premises ETL jobs to Amazon EMR. The code consists of a series of jobs written in Java. The company needs to reduce overhead for the system administrators without changing the underlying code. Due to the sensitivity of the data, compliance requires that the company use root device volume encryption on all nodes in the cluster. Corporate standards require that environments be provisioned though AWS CloudFormation when possible.
Which solution satisfies these requirements?

A. Install open-source Hadoop on Amazon EC2 instances with encrypted root device volumes. Configure the cluster in the CloudFormation template.

B. Use a CloudFormation template to launch an EMR cluster. In the configuration section of the cluster, define a bootstrap action to enable TLS.

C. Create a custom AMI with encrypted root device volumes. Configure Amazon EMR to use the custom AMI using the CustomAmiId property in the CloudFormation template.

D. Use a CloudFormation template to launch an EMR cluster. In the configuration section of the cluster, define a bootstrap action to encrypt the root device volume of every node.

Suggested Answer: *C*
Reference:
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-custom-ami.html

Community vote distribution
C (100%)

zeronine Highly Voted 3 years, 9 months ago
I think the answer is C
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-custom-ami.html
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-elasticmapreduce-cluster.html
upvoted 30 times

carol1522 3 years, 9 months ago
Agree with c
upvoted 2 times

LMax 3 years, 8 months ago
me too
upvoted 2 times

awssp12345 3 years, 9 months ago
Agreed
upvoted 3 times

lakediver 3 years, 6 months ago
Agree C.
If you are using an Amazon EMR version earlier than 5.24.0, an encrypted EBS root device volume is supported only when using a custom AMI. For Amazon EMR version 5.24.0 and later, you can use a security configuration option to encrypt EBS root device and storage volumes when you specify AWS KMS as your key provider.
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-data-encryption-options.html#emr-encryption-localdisk
upvoted 4 times

Huy Highly Voted 3 years, 7 months ago
Agree with C. D is a trap, it is Security Configuration section not bootstrap action in Configuration section.
upvoted 7 times

pk349 Most Recent 2 years, 1 month ago
C: I passed the test
upvoted 2 times

Ryo0w0o 2 years, 7 months ago

I will go for D.
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-data-encryption-options.html#emr-encryption-localdisk

According to the link, we can use EBS encryption from a security configuration and it says "We recommend using EBS encryption".
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: C

Correct answer is C as CloudFormation can be used to launch an EMR cluster with custom AMI with encrypted root device volumes.

Option A is wrong as open source Hadoop would not be provisioned using CloudFormation.

Option B is wrong as TLS does not provide data at rest encryption.

Option D is wrong as bootstrap actions cannot be used to encrypt root device volume.
upvoted 2 times

Naku 2 years, 6 months ago

bro, can you tell if we just do first 80 questions , can we pass?
upvoted 5 times

Arka_01 2 years, 9 months ago

Selected Answer: C

"without changing the underlying code" and "CloudFomation Template" are the keys here. So CustomAMIID for including a Custom AMI with encrypted root volume will work.
upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: C

C is the right answer.
upvoted 2 times

jrheen 3 years, 2 months ago

Answer - C
upvoted 1 times

lakediver 3 years, 6 months ago

If you are using an Amazon EMR version earlier than 5.24.0, an encrypted EBS root device volume is supported only when using a custom AMI. For more information, see Creating a custom AMI with an encrypted Amazon EBS root device volume in the Amazon EMR Management Guide
Beginning with Amazon EMR version 5.24.0, you can use a security configuration option to encrypt EBS root device and storage volumes when you specify AWS KMS as your key provider. For more information, see Local disk encryption.
upvoted 2 times

aws2019 3 years, 7 months ago

Agree with c
upvoted 1 times

lostsoul07 3 years, 7 months ago

C is the right answer
upvoted 3 times

[Removed] 3 years, 7 months ago

C sounds right but where in CF can you define a CustomAmiId? Its imageID and that's it. An AMI is an AMI. For D to work, you would have to use a 3rd party software, but it would work
upvoted 2 times

[Removed] 3 years, 7 months ago

Scratch that, you can do CustomAmiID in an EMR cluster..... C is indeed the answer.
upvoted 1 times

jove 3 years, 8 months ago

C is correct
upvoted 2 times

sanjaym 3 years, 8 months ago

Sensing answer should be C.
upvoted 1 times

jack42 3 years, 8 months ago
Its C, you cant use bootstrap action to encrypt the root volume, you need to pass it using security configurations.
upvoted 2 times

syu31svc 3 years, 8 months ago
https://aws.amazon.com/premiumsupport/knowledge-center/cloudformation-root-volume-property/
Answer is C
upvoted 3 times

Paitan 3 years, 8 months ago
Confused between C and D.
upvoted 1 times

KoMo 3 years, 8 months ago
I think the bootstrap config is only for installing additional softwares https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-elasticmapreduce-cluster-bootstrapactionconfig.html
upvoted 2 times

Question #58 *Topic 1*

A transportation company uses IoT sensors attached to trucks to collect vehicle data for its global delivery fleet. The company currently sends the sensor data in small .csv files to Amazon S3. The files are then loaded into a 10-node Amazon Redshift cluster with two slices per node and queried using both Amazon Athena and Amazon Redshift. The company wants to optimize the files to reduce the cost of querying and also improve the speed of data loading into the Amazon
Redshift cluster.
Which solution meets these requirements?

A. Use AWS Glue to convert all the files from .csv to a single large Apache Parquet file. COPY the file into Amazon Redshift and query the file with Athena from Amazon S3.

B. Use Amazon EMR to convert each .csv file to Apache Avro. COPY the files into Amazon Redshift and query the file with Athena from Amazon S3.

C. Use AWS Glue to convert the files from .csv to a single large Apache ORC file. COPY the file into Amazon Redshift and query the file with Athena from Amazon S3.

D. Use AWS Glue to convert the files from .csv to Apache Parquet to create 20 Parquet files. COPY the files into Amazon Redshift and query the files with Athena from Amazon S3.

Suggested Answer: *D*

Community vote distribution

D (100%)

ali_baba_acs Highly Voted 3 years, 8 months ago

D, is the good answer. In fact each nodes have 2 slices so ideally we can parrelize the copy process by sending a multiple of 20.
upvoted 27 times

LMax 3 years, 8 months ago

D for sure
upvoted 3 times

Paitan Highly Voted 3 years, 8 months ago

Trick question. Since we have 10 nodes with 2 slices each, ideally a multiple of 20 files should help in the parallelize the Copy process. So D is the right answer.
upvoted 8 times

roymunson 1 year, 7 months ago

And why it is a trick question? IMO it's an obv hint.
upvoted 1 times

pk349 Most Recent 2 years, 1 month ago

D: I passed the test
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: D

Correct answer is D as AWS Glue can be used to combine the .csv files to 20. parquet files. This would allow even processing across 2 slices. Also, multiple files help rapid loading of data to Redshift.

Options A & C are wrong as single large file is not efficient as it would use only single slice.

Option B is wrong as using Glue would be more cost effective as compared to EMR.
upvoted 4 times

bill1214 2 years, 1 month ago

I agree with your response - however EMR is more cost effective than Glue.. Glue is serverless while EMR is just a managed service.
upvoted 1 times

thirukudil 2 years, 8 months ago

Selected Answer: D

Option D is perfect solution to achieve both the requirements - to reduce the cost of querying (when we query on parquet files, lower the amount of data would be scanned which in turn reduce the cost) and also improve the speed of data loading into the Amazon
Redshift cluster(Split large files wherever possible to a number equal to a multiple of total number of slices. So here 20*n would be the correct splitting of the large data file.).

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: D

Best practices to Copy data from S3 to Redshift -
1) Use Columnar data format. Which is Parquet/ORC.
2) Split large files wherever possible to a number equal to a multiple of total number of slices. So here 20*n would be the correct splitting of the large data file.

upvoted 3 times

Hruday 2 years, 10 months ago

Selected Answer: D

Ans is D

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: D

Selected Answer: D

upvoted 1 times

Ramshizzle 3 years ago

Selected Answer: D

D is the right answer. 20 files = one per slice. If you use COPY on a dataset all the files will be divided over the available nodes/slices.

upvoted 1 times

somenath 3 years, 1 month ago

Per the link https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-use-multiple-files.html , the answer shall be D, for the compressed files multiple copy commands for its slices adhere parallel load whereas for the delimited file a single copy command works better. Here in all the options the file compression is taking place , so option D seems the best choice here.

upvoted 1 times

arun004 3 years, 6 months ago

Not sure D is correct or not but only reason i choose D since A and C are almost same and B won't work

upvoted 1 times

aws2019 3 years, 7 months ago

D is the right answer

upvoted 1 times

iconara 3 years, 7 months ago

D is the answer, but I doubt the total time will be shorter.The load will be quicker, shure, but it's not as if Spark reads CSV files quicker in any way, so all that you get is overhead. The Athena queries will run faster on fewer files, though, and if that wasthe focus this question would have made sense.

upvoted 1 times

afantict 3 years, 8 months ago

Is Athena query cheaper than the existing redshift query?

upvoted 1 times

lostsoul07 3 years, 8 months ago

D is the right answer

upvoted 3 times

sanjaym 3 years, 8 months ago

D is correct answer.

upvoted 2 times

apuredol 3 years, 8 months ago

is D ok? In understand you should avoid multiple concurrency copy commands
"We strongly recommend using the COPY command to load large amounts of data. Using individual INSERT statements to populate a table might be prohibitively slow. Alternatively, if your data already exists in other Amazon Redshift database tables, use INSERT INTO ... SELECT or CREATE TABLE

AS to improve performance. For information, see INSERT or CREATE TABLE AS".
https://docs.aws.amazon.com/redshift/latest/dg/t_Loading_tables_with_the_COPY_command.html
upvoted 1 times

CHRIS12722222 3 years, 3 months ago
Single COPY command loads multiple files into Redshift in parallel
upvoted 2 times

Question #59 *Topic 1*

An online retail company with millions of users around the globe wants to improve its ecommerce analytics capabilities. Currently, clickstream data is uploaded directly to Amazon S3 as compressed files. Several times each day, an application running on Amazon EC2 processes the data and makes search options and reports available for visualization by editors and marketers. The company wants to make website clicks and aggregated data available to editors and marketers in minutes to enable them to connect with users more effectively.
Which options will help meet these requirements in the MOST efficient way? (Choose two.)

A. Use Amazon Kinesis Data Firehose to upload compressed and batched clickstream records to Amazon OpenSearch Service (Amazon Elasticsearch Service).

B. Upload clickstream records to Amazon S3 as compressed files. Then use AWS Lambda to send data to Amazon OpenSearch Service (Amazon Elasticsearch Service) from Amazon S3.

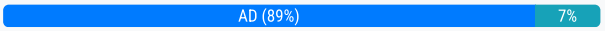
C. Use Amazon OpenSearch Service (Amazon Elasticsearch Service) deployed on Amazon EC2 to aggregate, filter, and process the data. Refresh content performance dashboards in near-real time.

D. Use OpenSearch Dashboards (Kibana) to aggregate, filter, and visualize the data stored in Amazon OpenSearch Service (Amazon Elasticsearch Service). Refresh content performance dashboards in near-real time.

E. Upload clickstream records from Amazon S3 to Amazon Kinesis Data Streams and use a Kinesis Data Streams consumer to send records to Amazon OpenSearch Service (Amazon Elasticsearch Service).

Suggested Answer: *CE*

Community vote distribution



astalavista1 Highly Voted 3 years, 2 months ago

Selected Answer: AD

OpenSearch can ingest from KDF and results not in real-time but in minutes, as such, KDF can still be used.

upvoted 10 times

penguins2 Most Recent 1 year, 11 months ago

AD for sure.

upvoted 1 times

pk349 2 years, 1 month ago

AD: I passed the test

upvoted 3 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: AD

Correct answers are A & D as Kinesis Data Firehose can be used for data ingestion with its micro batching to push the data directly to Elastic Search. Kibana can be used for visualization.

Options B, C & E are wrong as they are not the MOST efficient way.

upvoted 4 times

thirukudil 2 years, 8 months ago

Selected Answer: AD

KDF can ingest that data directly to Opensearch (Still KDF has 60 sec buffer time, they want results in minutes. so this is fine). Choose opensearch dashboards for visualization when it comes to time-sensitive.

here, some people choosing B over A. My explanation why we can ignore B is - they want result in minutes not real-time. So KDF is sufficient and will replace the whole process of streaming the data to s3 and then loading the data to OpenSearch via lambda.

upvoted 3 times

Arka_01 2 years, 9 months ago

Selected Answer: AD

"in minutes" - this is the key here. Always choose OpenSearch over Athena+Quicksight, in case of time sensitivity.

upvoted 2 times

rrshah83 2 years, 10 months ago

Selected Answer: BD

B and D.

(Reason for B as opposed to A: Firehose has minimum 1 min delay. Lambda will be "instantly". Question)

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: AD

Selected Answer: AD

upvoted 1 times

Balki 3 years ago

Selected Answer: AD

AD IS THE ANSWER

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: AD

Answer should be A & D

upvoted 3 times

certificationJunkie 3 years, 1 month ago

B and D. AWS Lambda can be triggered for s3 events generated while copying to s3. And then it loads to OpenSearch in near real time. Then use Kibana for visualization and search requirement.

upvoted 2 times

rrshah83 2 years, 10 months ago

B and D.

(Reason for B as opposed to A: Firehose has minimum 1 min delay. Lambda will be "instantly". Question)

upvoted 2 times

Bik000 3 years, 1 month ago

Selected Answer: DE

My Answer is D & E

upvoted 1 times

Bik000 3 years, 1 month ago

No A & D I meant

upvoted 1 times

jrheen 3 years, 2 months ago

Answer - A,D

upvoted 1 times

G_C_P 3 years, 2 months ago

A, D most efficient and direct to destination. B instead of A feasible but uses S3. But qn also says data stored in S3 - assume for the best approach to A, D

upvoted 1 times

G_C_P 3 years, 2 months ago

A, D - firehose direct to ES; Kibana built in

upvoted 2 times

Question #60 *Topic 1*

A company is streaming its high-volume billing data (100 MBps) to Amazon Kinesis Data Streams. A data analyst partitioned the data on account_id to ensure that all records belonging to an account go to the same Kinesis shard and order is maintained. While building a custom consumer using the Kinesis Java SDK, the data analyst notices that, sometimes, the messages arrive out of order for account_id. Upon further investigation, the data analyst discovers the messages that are out of order seem to be arriving from different shards for the same account_id and are seen when a stream resize runs.
What is an explanation for this behavior and what is the solution?

A. There are multiple shards in a stream and order needs to be maintained in the shard. The data analyst needs to make sure there is only a single shard in the stream and no stream resize runs.

B. The hash key generation process for the records is not working correctly. The data analyst should generate an explicit hash key on the producer side so the records are directed to the appropriate shard accurately.

C. The records are not being received by Kinesis Data Streams in order. The producer should use the PutRecords API call instead of the PutRecord API call with the SequenceNumberForOrdering parameter.

D. The consumer is not processing the parent shard completely before processing the child shards after a stream resize. The data analyst should process the parent shard completely first before processing the child shards.

Suggested Answer: *A*

Community vote distribution

D (100%)

Priyanka_01 Highly Voted 3 years, 8 months ago
D:https://docs.aws.amazon.com/streams/latest/dev/kinesis-using-sdk-java-after-resharding.html
the parent shards that remain after the reshard could still contain data that you haven't read yet that was added to the stream before the reshard. If you read data from the child shards before having read all data from the parent shards, you could read data for a particular hash key out of the order given by the data records' sequence numbers. Therefore, assuming that the order of the data is important, you should, after a reshard, always continue to read data from the parent shards until it is exhausted. Only then should you begin reading data from the child shards.
upvoted 80 times

pk349 Most Recent 2 years, 1 month ago
D: I passed the test
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: D
Correct answer is D as Kinesis Data Streams resharding causes the existing data to be in parent shard and new data is routed to child shards. The issue can occur if the consumers starts processing the child shard without completely processing the parent shard.
upvoted 4 times

Bik000 3 years, 1 month ago
Selected Answer: D
Answer is D for sure
upvoted 2 times

aws2019 3 years, 7 months ago
D is ans
upvoted 1 times

Kamalt 3 years, 7 months ago
Answer D.
upvoted 1 times

lostsoul07 3 years, 8 months ago
D is the right answer
upvoted 2 times

sanjaym 3 years, 8 months ago
Answer should be D.

upvoted 2 times

Paitan 3 years, 8 months ago

Nicely explained by Priyanka_01.

upvoted 3 times

Question #61 *Topic 1*

A media analytics company consumes a stream of social media posts. The posts are sent to an Amazon Kinesis data stream partitioned on user_id. An AWS
Lambda function retrieves the records and validates the content before loading the posts into an Amazon OpenSearch Service (Amazon Elasticsearch Service) cluster. The validation process needs to receive the posts for a given user in the order they were received by the Kinesis data stream.
During peak hours, the social media posts take more than an hour to appear in the Amazon OpenSearch Service (Amazon ES) cluster. A data analytics specialist must implement a solution that reduces this latency with the least possible operational overhead.
Which solution meets these requirements?

A. Migrate the validation process from Lambda to AWS Glue.

B. Migrate the Lambda consumers from standard data stream iterators to an HTTP/2 stream consumer.

C. Increase the number of shards in the Kinesis data stream.

D. Send the posts stream to Amazon Managed Streaming for Apache Kafka instead of the Kinesis data stream.

Suggested Answer: *C*
For real-time processing of streaming data, Amazon Kinesis partitions data in multiple shards that can then be consumed by multiple Amazon EC2
Reference:
https://d1.awsstatic.com/whitepapers/AWS_Cloud_Best_Practices.pdf

Community vote distribution
C (95%) 5%

Alekx42 Highly Voted 2 years, 12 months ago
Selected Answer: C
Increasing the number of shards seems to be a good idea since Lambda can process 1 batch of data from each Kinesis shard with 1 lambda invocation. This means that if you have 100 shards you can have 100 concurrent lambda invocations. If you increase the number of shards you can increase the parallelism and you could be quicker to process the data. This is assuming that the Lambda ParallelizationFactor is set to 1.
Switching to AWS Glue could increase the speed of the data processing (since Glue can use Spark, which can be way faster than a Lambda function when processing a lot of data) but this would increase the operational overhead.
upvoted 12 times

god_father Most Recent 1 year, 4 months ago
For those wondering why not go for 'A', and go for 'C' instead: in glue worker types such as G.1X, G.2X, etc must be selected which increases overhead. Hence, the one with least overhead is 'C' by using the concept of parallelism.
upvoted 1 times

GCPereira 1 year, 5 months ago
A: is wrong because glue has concurrency limit and spark is poor option to small files;
B: if lambda is a unique consumer, dont have necessity to upgrade these for enhanced-fan-out;
C: this is a tipically bottleneck problem... to solve this just insert more shards;
D: is wrong because switch SaaS have A LOT OF operational overhead
upvoted 2 times

juanife 1 year, 10 months ago
I want to contribute in this Question and telling you why B isn't correct and C yes.
Option B is useless because the question never tell that there is another consumer, so lambda is leveraging the shard throughput (there is no need to set enhanced fan-out consumer).
Incrementing shard will work, because in AWS there's 1 lambda function invocation per kinesis shard. At the same time, per shard you can increase lambda functions concurrency with the ParallelizationFactor set to a name between 1 (default value) to 10.
upvoted 2 times

MLCL 1 year, 11 months ago
Could be C or B depending on multiple factors.
To increase the prformance of KDS you have 3 options :
- More shards.

- Parallelization factors (specific to Lambda)
- HTTP/2
- Enhanced Fan-out.

upvoted 2 times

Debuggerrr 1 year, 11 months ago

B seems to be correct. As ordering has to be maintained, Lambda function will be handy only if the consumer is KCL because it has that inbuilt sorting logic for parent and child shard.

upvoted 3 times

Debi_mishra 2 years, 1 month ago

C can never be right - increasing shards cannot assure ordering and thats the catch here. B seems close.

upvoted 1 times

MLCL 1 year, 11 months ago

If you partition by user_id, it does guarantee order, since only the same user_id go to the same shard.

upvoted 3 times

MLCL 1 year, 11 months ago

The stream is partitioned by user_id, increasing the number of shards won't impact the ordering of records for a specific user because all posts from a particular user would go to the same shard.

upvoted 3 times

pk349 2 years, 1 month ago

C: I passed the test

upvoted 2 times

rags1482 2 years, 3 months ago

Answer B

based on below link

https://aws.amazon.com/about-aws/whats-new/2018/11/aws-lambda-supports-kinesis-data-streams-enhanced-fan-out-and-http2/

upvoted 2 times

Arjun777 2 years, 4 months ago

option B- Migrating the Lambda consumers to an HTTP/2 stream consumer can significantly reduce processing latency and improve the overall performance of the system. This is because HTTP/2 stream consumers allow Lambda to retrieve records from the stream more efficiently, which can help to reduce processing latency and improve the overall performance of the system.

Migrating the Lambda consumers to an HTTP/2 stream consumer can significantly reduce processing latency and improve the overall performance of the system. This is because HTTP/2 stream consumers allow Lambda to retrieve records from the stream more efficiently, which can help to reduce processing latency and improve the overall performance of the system.

Migrating to an HTTP/2 stream consumer requires minimal operational overhead, as it only involves updating the Lambda function code to use the new consumer type. This can be done easily using the AWS SDK for Lambda, and does not require any major changes to the existing architecture.

Therefore, option B is the best solution for reducing the latency with the least possible operational overhead.

upvoted 3 times

aws_kid 2 years, 2 months ago

Increasing shards is easier than enhanced fan-out and cheaper too.

upvoted 2 times

nadavw 2 years, 5 months ago

Selected Answer: B

C is a temporary solution, as there is no idea of the expected number of shards you need to increase. There is no simple auto-scaling in Kinesis, so there will be an operational overhead to continuously monitor the system and increase the number of shards. In addition, the partitioning-sharding is according to user_id - how this can be solved?

B - the enhanced fanout approach is good for it, as described here:
"The enhanced capacity enables you to achieve higher outbound throughput without provisioning more streams or shards in the same stream."
https://aws.amazon.com/blogs/compute/increasing-real-time-stream-processing-performance-with-amazon-kinesis-data-streams-enhanced-fan-out-and-aws-lambda/

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: C

"least possible operational overhead" - This is the key here. As the solution demands to reduce latency, this will be the easiest way to do so. Notice, that cost factor is not mentioned in the question.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: C

Increasing the number of shards looks ok.

upvoted 1 times

Sen5476 2 years, 12 months ago

I go with B for these two reasons.

1. Messages should be received in same order for user. Scaling out the shards during peak hours and scaling in after peak hours may change the message order. C is not the correct one.

2. HTTP/2 is enhanced fan out consumer which will reduce the latency from 200ms to 70ms. 65% latency reduction

upvoted 4 times

f4bi4n 3 years, 1 month ago

Selected Answer: C

C, but you must ensure that you use Partition Keys (In this case the User) to ensure the requested ordering per User. HTTP/2 would also decrease latency but needs more effort

upvoted 2 times

jrheen 3 years, 2 months ago

C - Increase Shards

upvoted 2 times

CHRIS12722222 3 years, 2 months ago

I think B

standard consumer latency = 200ms

Http/2 latency = 70ms

upvoted 3 times

Question #62 *Topic 1*

A company launched a service that produces millions of messages every day and uses Amazon Kinesis Data Streams as the streaming service. The company uses the Kinesis SDK to write data to Kinesis Data Streams. A few months after launch, a data analyst found that write performance is significantly reduced. The data analyst investigated the metrics and determined that Kinesis is throttling the write requests. The data analyst wants to address this issue without significant changes to the architecture.
Which actions should the data analyst take to resolve this issue? (Choose two.)

A. Increase the Kinesis Data Streams retention period to reduce throttling.

B. Replace the Kinesis API-based data ingestion mechanism with Kinesis Agent.

C. Increase the number of shards in the stream using the UpdateShardCount API.

D. Choose partition keys in a way that results in a uniform record distribution across shards.

E. Customize the application code to include retry logic to improve performance.

Suggested Answer: *AC*

Community vote distribution

CD (100%)

zanhsieh Highly Voted 3 years, 9 months ago

CD. If wrong partition keys are distributed well, then retrying would still hit the hot shards.
https://aws.amazon.com/blogs/big-data/under-the-hood-scaling-your-kinesis-data-streams/
upvoted 33 times

awssp12345 3 years, 9 months ago

agreed.
upvoted 2 times

GeeBeeEl 3 years, 8 months ago

D i agree with because of this other link https://aws.amazon.com/premiumsupport/knowledge-center/kinesis-data-stream-throttling-errors/ "use a random partition key to ingest your records. If the operations already use a random partition key, then adjust the key to correct the distribution."
Why would you want to increase the number of shards, is that in the link?
upvoted 1 times

Paitan Highly Voted 3 years, 9 months ago

C and D for me.
upvoted 7 times

MLCL Most Recent 1 year, 11 months ago

Selected Answer: CD
More shards and better partitioning is always the answer for write performance issues.
upvoted 4 times

pk349 2 years, 1 month ago

CD: I passed the test
upvoted 1 times

Chelseajcole 2 years, 5 months ago

Why no B? Kinesis should have better performance than KPL
upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: CD
Correct answers are C & D as the common causes are hitting shard limits and. increasing shard count and choosing an appropriate partition key would help solve the issue.
upvoted 3 times

thirukudil 2 years, 8 months ago

Selected Answer: CD

throttling writes mean there is no sufficient shards. so we need to increase the shard counts. Introducing partition key helps in uniform distribution of writes across the shards which in turn we can avoid hot shards

upvoted 3 times

Arka_01 2 years, 9 months ago

Selected Answer: CD

None of the other options makes sense.

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: CD

answer is CD

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: CD

Answer is C & D

upvoted 1 times

aws2019 3 years, 7 months ago

C and D

upvoted 1 times

Donell 3 years, 7 months ago

C,D is the answer.

upvoted 1 times

ariane_tateishi 3 years, 7 months ago

Considering the link bellow the option E isn't a valid option. "If there is a retry mechanic in the producer, failed records are tried again. This can also cause a delay in processing."

https://aws.amazon.com/pt/premiumsupport/knowledge-center/kinesis-data-stream-throttling/

upvoted 1 times

lostsoul07 3 years, 8 months ago

C,D is the right answer

upvoted 3 times

Par301 3 years, 8 months ago

When there are failed records that aren't able to enter the Kinesis data stream, the stream throttles. If there is a retry mechanic in the producer, failed records are tried again. This can also cause a delay in processing.

https://aws.amazon.com/premiumsupport/knowledge-center/kinesis-data-stream-throttling/

So an existing retry logic can also result in throttling. I think C&D will make more sense as they are meant for performance increase. Retry logic is more for handling errors and not handling performance.

upvoted 5 times

Draco31 3 years, 8 months ago

C and D for me.

"A few months after launch, a data analyst found that write performance is significantly reduced" means the load is increasing constantly and will not decrease, and it's not a temporary spike. So we are facing a lack of shard or bad partition key is burning some shards over time

upvoted 6 times

kamparia 3 years, 8 months ago

It's C & E for me. D is not an option because changing the partition key affects the architecture.

upvoted 1 times

MWL 3 years, 1 month ago

"changing the partition key" just change the code to use different key during putting item. No architecture changed.

upvoted 2 times

Question #63 *Topic 1*

A smart home automation company must efficiently ingest and process messages from various connected devices and sensors. The majority of these messages are comprised of a large number of small files. These messages are ingested using Amazon Kinesis Data Streams and sent to Amazon S3 using a Kinesis data stream consumer application. The Amazon S3 message data is then passed through a processing pipeline built on Amazon EMR running scheduled PySpark jobs.
The data platform team manages data processing and is concerned about the efficiency and cost of downstream data processing. They want to continue to use
PySpark.
Which solution improves the efficiency of the data processing jobs and is well architected?

A. Send the sensor and devices data directly to a Kinesis Data Firehose delivery stream to send the data to Amazon S3 with Apache Parquet record format conversion enabled. Use Amazon EMR running PySpark to process the data in Amazon S3.

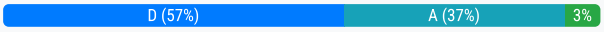
B. Set up an AWS Lambda function with a Python runtime environment. Process individual Kinesis data stream messages from the connected devices and sensors using Lambda.

C. Launch an Amazon Redshift cluster. Copy the collected data from Amazon S3 to Amazon Redshift and move the data processing jobs from Amazon EMR to Amazon Redshift.

D. Set up AWS Glue Python jobs to merge the small data files in Amazon S3 into larger files and transform them to Apache Parquet format. Migrate the downstream PySpark jobs from Amazon EMR to AWS Glue.

Suggested Answer: *A*

Community vote distribution



Phoenyx89 Highly Voted 3 years, 9 months ago
D seems the good choice because is the only answer dealing with small files but the doubt is... Glue is only batch! but there is a new article of 04/20 that says glue is now also supporting streaming process. So if we consider this article D is right. But we can?
https://aws.amazon.com/it/about-aws/whats-new/2020/04/aws-glue-now-supports-serverless-streaming-etl/
upvoted 28 times

metin 3 years, 8 months ago
In question, it is not mentioned that they definitely need a real-time solution. And Glue can be used for batch processing of stream data.
upvoted 2 times

Dr_Kiko 3 years, 7 months ago
they already use Firehouse that does data batching, so no problem with D
upvoted 1 times

lpc01 3 years, 5 months ago
Since question is searching for an answer that reflects operational efficiency, setting up individual glue jobs is definitely more time consuming so the answer is A
upvoted 2 times

lpc01 3 years, 5 months ago
Plus, AWS Kinesis Data Firehose has in-house data transformation so you don't need to add operational overhead by utilizing AWS Lambda
upvoted 2 times

juanife 2 years ago
No, the question asks for a solution that replaces ETL with Pyspark in EMR cluster indeed, so AWS GLUE ETL jobs would be a good choice here. Undoubtedly, D is the correct option, but option A is not as bad as seems, but maybe it's not as cheap as D.
upvoted 2 times

singh100 Highly Voted 3 years, 9 months ago
Anas: A

https://aws.amazon.com/blogs/big-data/optimizing-downstream-data-processing-with-amazon-kinesis-data-firehose-and-amazon-emr-running-apache-spark/
upvoted 23 times

GauravM17 3 years, 9 months ago

Should this not be D? Where are we handling the small files in A?

upvoted 1 times

GeeBeeEl 3 years, 8 months ago

You are changing them to parquet on the fly with Firehose.

upvoted 1 times

Ashish1101 2 years, 7 months ago

Firehose will batch in buffer time and reduce number of files.

upvoted 2 times

kzu19878 3 years, 9 months ago

Remember "They want to continue to use PySpark."

D is migrating PySpark into Glue

upvoted 8 times

AjNapa 3 years, 8 months ago

This part of the question is what many ppl here have missed. You're right. It's A

upvoted 2 times

Haimett 2 years, 8 months ago

There is no problem in using pyspark with Glue.

upvoted 2 times

NarenKA Most Recent 1 year, 4 months ago

Selected Answer: A

Converting data into Apache Parquet format before storing it in S3 optimizes the data for analytical processing. KDF able to automatically batch, compress, and convert incoming streaming data into Parquet format. It reduces the overhead with processing a large number of small files without the need for additional processing or intermediate steps. And it allows the team to continue using PySpark on Amazon EMR for data processing.

B - AWS Lambda to process individual messages could introduce operational overhead and not efficiently handle the conversion of a large number of small files.

C - moving data processing to Redshift would require changes to the existing PySpark-based processing pipeline and not most cost-effective solution.

D - merging small files into larger ones using Glue addresses the efficiency concern, it suggests migrating PySpark jobs from EMR to AWS Glue could involve refactoring of the existing jobs.

upvoted 2 times

GCPereira 1 year, 5 months ago

emr is very expansive and spark doesn't work well with a large number of small files... the best option is to merge small files into large files and use job glue to decrease the cost of downstream processing... D is a perfect answer

upvoted 1 times

GCPereira 1 year, 6 months ago

take a look at this sentence "...the solution needs to be well-architected"... that is, cost-efficient, secure, highly available and operational-efficient... aws emr are not highly available and need a lot of operational resources... then disagree emr... to continue pyspark job, aws glue are the best option

upvoted 1 times

MLCL 1 year, 11 months ago

Selected Answer: D

D solves the issue with small files and replaces the EMR batch job with a Glue one, which is cheaper.

If a Transient EMR cluster was in the A proposition, it would be acceptable.

upvoted 2 times

Debi_mishra 2 years, 1 month ago

Both A and D correct technically and very difficult to figure out the cost effective solution without more context. Other answers assuming EMR is a long running and is expensive but thats not mentioned here. D has upper hand considering all will be serverless.

upvoted 1 times

pk349 2 years, 1 month ago

D: I passed the test

upvoted 3 times

akashm99101001com 2 years, 3 months ago

Selected Answer: D

"cost of downstream data processing" so migrate it
upvoted 1 times

Chelseajcole 2 years, 5 months ago

Selected Answer: D

This question is testing if you know Glue can run PySpark job
upvoted 4 times

DeerSong 2 years, 5 months ago

Selected Answer: D

D for sue
upvoted 1 times

Kinlive1991 2 years, 5 months ago

Selected Answer: D

D is correct
upvoted 1 times

siju13 2 years, 6 months ago

Selected Answer: D

glue cheaper than emr
upvoted 1 times

henom 2 years, 7 months ago

The correct answer is D :- the option that says: Replace the Amazon EMR with AWS Glue. Program an AWS Glue ETL script in Python to merge the small sensor data into larger files and convert them to Apache Parquet format.

The option that says: Deploy a Kinesis Data Firehose delivery stream to collect and convert sensor data to Apache Parquet format. Deliver the transformed data into an Amazon S3 bucket. Process the data from the bucket using a PySpark Job running on an Amazon EMR cluster is incorrect. Although this option is valid, there is no significant cost reduction since Amazon EMR is still running. AWS Glue can provide lower costs while providing the same function. In addition, it is better to merge the smaller files to a large file, than just compressing them using the Apache Parquet format to improve ingestion performance.
upvoted 1 times

nadavw 2 years, 7 months ago

Selected Answer: D

The question is about the data processing and not the full pipeline including ingestion so D is the most efficient from processing perspective
upvoted 1 times

thuyeinaung 2 years, 7 months ago

Selected Answer: D

Glue can run PySpark
upvoted 1 times

alinato 2 years, 7 months ago

Selected Answer: B

Lambda can run pyspark and is cost effective and serverless meaning well architectured.
upvoted 1 times

Question #64 *Topic 1*

A large financial company is running its ETL process. Part of this process is to move data from Amazon S3 into an Amazon Redshift cluster. The company wants to use the most cost-efficient method to load the dataset into Amazon Redshift.
Which combination of steps would meet these requirements? (Choose two.)

A. Use the COPY command with the manifest file to load data into Amazon Redshift.

B. Use S3DistCp to load files into Amazon Redshift.

C. Use temporary staging tables during the loading process.

D. Use the UNLOAD command to upload data into Amazon Redshift.

E. Use Amazon Redshift Spectrum to query files from Amazon S3.

Suggested Answer: *CE*
Reference:
https://aws.amazon.com/blogs/big-data/top-8-best-practices-for-high-performance-etl-processing-using-amazon-redshift/

Community vote distribution
AC (100%)

Priyanka_01 Highly Voted 3 years, 9 months ago
A & C
Copy command and loading into temp staging tables
upvoted 31 times

carol1522 Highly Voted 3 years, 9 months ago
A and c, because the goal is move data from s3 to redshift, and in the E we are not moving.
upvoted 14 times

Debi_mishra Most Recent 2 years, 1 month ago
A & C. But If you are going to appear exam in near future - redshift auto copy is now a new no-ETL feature and may replace these options.
upvoted 2 times

pk349 2 years, 1 month ago
AC: I passed the test
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: AC
Correct answers are A & C.

Option B is wrong as S3DistCp is used to copy data between S3 and HDFS.

Option D is wrong as UNLOAD helps unloading the data from Redshift to S3.

Option E is wrong as Redshift Spectrum does not load the data into Redshift, but the requirement is to load.
upvoted 8 times

cloudlearnerhere 2 years, 7 months ago
Option A as the COPY command loads data in parallel from Amazon S3, Amazon EMR, Amazon DynamoDB, or multiple data sources on remote hosts. COPY loads large amounts of data much more efficiently than using INSERT statements, and stores the data more effectively as well. Amazon S3 provides eventual consistency for some operations. Thus, it's possible that new data won't be available immediately after the upload, which can result in an incomplete data load or loading stale data. You can manage data consistency by using a manifest file to load data

Option C as you can efficiently update and insert new data by loading your data into a staging table first. Amazon Redshift doesn't support a single merge statement (update or insert, also known as an upsert) to insert and update data from a single data source. However, you can effectively perform a merge operation. To do so, load your data into a staging table and then join the staging table with your target table for an UPDATE statement and an INSERT statement.
upvoted 4 times

dushmantha 2 years, 10 months ago

Selected Answer: AC

B is not correct because its used with EMR. D is not correct because UNLOAD is used to put data from Redshift to S3. C seems to be involve lot of work, but E does not allow to move data to Redshift but the organization requires that and A is anyway correct. So I would go with A nd C

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: AC

A, C are correct

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: AC

Answer is A & C

upvoted 1 times

jrheen 3 years, 2 months ago

Answer - A,C

upvoted 1 times

aws2019 3 years, 7 months ago

A and C

upvoted 1 times

gunjan4392 3 years, 7 months ago

A, C are correct

upvoted 1 times

lostsoul07 3 years, 8 months ago

A,C is the right answer

upvoted 2 times

Subho_in 3 years, 8 months ago

https://aws.amazon.com/blogs/big-data/top-8-best-practices-for-high-performance-etl-processing-using-amazon-redshift/

Point number 1 and 2. Option A and C must be the answer

upvoted 10 times

Ramshizzle 3 years ago

Point 5 is also important to note in the article mentioned by Subho_in.

Also look at this why to use Staging tables: https://docs.aws.amazon.com/redshift/latest/dg/merge-create-staging-table.html

upvoted 1 times

gtourkas 3 years, 8 months ago

I disagree with C. Question is about Loading data. Staging tables is about Transformation. It's A and E for me.

upvoted 1 times

APIsche 2 years, 10 months ago

"The organization want to load the dataset onto Amazon Redshift". answer E is not moving any data not does help with it

upvoted 1 times

jove 3 years, 8 months ago

It's asking a "combination of steps", so they are A and C..

upvoted 2 times

sanjaym 3 years, 8 months ago

A and C

upvoted 2 times

syu31svc 3 years, 8 months ago

A & C for sure; the rest are clearly wrong

upvoted 2 times

Question #65 *Topic 1*

A university intends to use Amazon Kinesis Data Firehose to collect JSON-formatted batches of water quality readings in Amazon S3. The readings are from 50 sensors scattered across a local lake. Students will query the stored data using Amazon Athena to observe changes in a captured metric over time, such as water temperature or acidity. Interest has grown in the study, prompting the university to reconsider how data will be stored.
Which data format and partitioning choices will MOST significantly reduce costs? (Choose two.)

A. Store the data in Apache Avro format using Snappy compression.

B. Partition the data by year, month, and day.

C. Store the data in Apache ORC format using no compression.

D. Store the data in Apache Parquet format using Snappy compression.

E. Partition the data by sensor, year, month, and day.

Suggested Answer: *CD*
Reference:
https://docs.aws.amazon.com/firehose/latest/dev/record-format-conversion.html

Community vote distribution
BD (62%) DE (38%)

Priyanka_01 Highly Voted 3 years, 9 months ago
D :can save from 30% to 90% on your per-query costs and get better performance by compressing, partitioning, and converting your data into columnar formats.
B: For partition
upvoted 44 times

jay1ram2 Highly Voted 3 years, 8 months ago
B and D are the right answers.

Some background: Snappy compresses the data to help with I/O, it roughly does the same level of compression for both parquet and AVRO. AVRO stores the data in row format and does not compresses the data. However, Parquet is a columnar store (without any additional compression algorithm like snappy applied), it natively compresses the data by 2X to 5X on average.

A) Since Parquet does a better job in compression, this option is incorrect
B) This is correct since data is partitioned with keys (year, month, day) with medium cardinality.
C) Even though ORC and Parquet are both columnar storage formats and both supported by Athena, Since no compression is used in this option, we can safely ignore this.
D) Parquet with Snappy is a better choice than ORC with no compression, so this is correct.
E) Adding sensor(ID) to the partition creates high cardinality on the partitions and may lead to multiple small files under each partition which will slow down performance. So, B is a better option as you can keep all 50 sensor data in a single file for a day.
upvoted 30 times

wally_1995 1 year, 12 months ago
I found this at this link:

Columns that are used as filters are good candidates for partitioning.
Partitioning has a cost. As the number of partitions in your table increases, the higher the overhead of retrieving and processing the partition metadata, and the smaller your files. Partitioning too finely can wipe out the initial benefit.

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

So I'd also go with B!
upvoted 1 times

Sai3596 Most Recent 1 year, 3 months ago

In the question it is mentioned that "Athena to observe changes in a captured metric over time, such as water temperature or acidity."

No signs of using a specific sensor and then observe metrics.
So if we introduce sensor in partition and not filter it in the query you are introducing an additional partition to search.
The request of the question it makes sense that we use B
upvoted 1 times

MLCL 1 year, 11 months ago

Selected Answer: BD

BD : Makes the most sense
upvoted 1 times

pk349 2 years, 1 month ago

DE: I passed the test
upvoted 1 times

GCPereira 1 year, 6 months ago

i'm also passed the test, but the sensor id increases the cardinality of the dataset... then the best option is to partition the data by year, month, and day, compress and convert JSON to a colunar file, in this case, parquet file.
upvoted 1 times

rags1482 2 years, 3 months ago

Partitioning by sensor, year, month, and day (option E) would likely increase costs as compared to partitioning by only year, month, and day (option B) because it would create a larger number of smaller partitions. Each partition would contain data from a single sensor for a given date range, resulting in more small files that would need to be scanned by Athena for each query.

So B is better answer than E
upvoted 2 times

murali12180 2 years, 4 months ago

Selected Answer: DE

partition by sensor and then by year/month/day make sense, parquet with snappy gives best compressions
upvoted 1 times

Gabba 2 years, 4 months ago

Selected Answer: BD

B partition strategy better than E.
D for sure.
upvoted 4 times

rocky48 2 years, 6 months ago

Selected Answer: DE

D is an obvious choice. E has the highest potential to save costs also for queries that filter the sensor and the task is to find the solution with the most cost savings
upvoted 2 times

rocky48 2 years, 6 months ago

Adding sensor(ID) to the partition creates high cardinality on the partitions and may lead to multiple small files under each partition which will slow down performance. But the question mentions about saving costs and not performance.
upvoted 2 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: BD

Correct answers are B & D

Option B as the data can be partitions by year, month, and day as it needs to be analyzed using captured metrics over time and not specific to any sensor.

Option D as columnar data format helps to improve query performance.

Options A & C are wrong as Avro and ORC without compression would not provide query performance similar to parquet with compression.

Option E is wrong as the data needs to be analyzed as per the metrics and not specific to a particular sensor.

upvoted 3 times

Saneeda 2 years, 7 months ago

(A. Store the data in Apache Avro format using Snappy compression) Option A includes compression but Parquet with Snappy Compression is better option because Avro stores data in row format per @jay1ram2. Correct me if I am wrong.

upvoted 1 times

aefuen1 2 years, 8 months ago

Selected Answer: BD

B and D. They are will query by time, not sensor id.

upvoted 1 times

LukeTran3206 2 years, 8 months ago

Selected Answer: BD

If possible, avoid having a large number of small files – Amazon S3 has a limit of 5500 requests per second. Athena queries share the same limit.
https://docs.aws.amazon.com/athena/latest/ug/performance-tuning.html

upvoted 1 times

Arka_01 2 years, 9 months ago

Selected Answer: DE

Because more and optimal number of partitions can be done through the option E. Snappy compression with Parquet format, allows easy integration and maximum storage saving.

upvoted 1 times

ryuhei 2 years, 10 months ago

Selected Answer: BD

Answer is B & D

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: BD

B and D are the right answers.

upvoted 2 times

rocky48 2 years, 6 months ago

E has the highest potential to save costs also for queries that filter the sensor and the task is to find the solution with the most cost savings.

upvoted 1 times

ru4aws 2 years, 11 months ago

Selected Answer: DE

The metrics of temperature and acidity may be varying between different locations of the lake, students may want to see if any issues at a particular location level based on metrics. So its advisable to partition by Sensor

upvoted 2 times

[Removed] 2 years, 12 months ago

I vote A & E. Vote for A because "gather JSON-formatted batches of water quality values in Amazon S3" is the requirement. We can't compress the Json format file using Parquet or ORC.

upvoted 1 times

Question #66 *Topic 1*

A healthcare company uses AWS data and analytics tools to collect, ingest, and store electronic health record (EHR) data about its patients. The raw EHR data is stored in Amazon S3 in JSON format partitioned by hour, day, and year and is updated every hour. The company wants to maintain the data catalog and metadata in an AWS Glue Data Catalog to be able to access the data using Amazon Athena or Amazon Redshift Spectrum for analytics.

When defining tables in the Data Catalog, the company has the following requirements:

✑ Choose the catalog table name and do not rely on the catalog table naming algorithm.

✑ Keep the table updated with new partitions loaded in the respective S3 bucket prefixes.

Which solution meets these requirements with minimal effort?

A. Run an AWS Glue crawler that connects to one or more data stores, determines the data structures, and writes tables in the Data Catalog.

B. Use the AWS Glue console to manually create a table in the Data Catalog and schedule an AWS Lambda function to update the table partitions hourly.

C. Use the AWS Glue API CreateTable operation to create a table in the Data Catalog. Create an AWS Glue crawler and specify the table as the source.

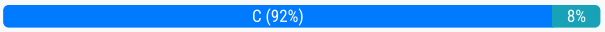
D. Create an Apache Hive catalog in Amazon EMR with the table schema definition in Amazon S3, and update the table partition with a scheduled job. Migrate the Hive catalog to the Data Catalog.

Suggested Answer: *B*

Reference:

https://docs.aws.amazon.com/glue/latest/dg/tables-described.html

Community vote distribution



Marc34 Highly Voted 3 years, 9 months ago

C.

https://docs.aws.amazon.com/glue/latest/dg/tables-described.html

In this section :

Updating Manually Created Data Catalog Tables Using Crawlers

"The following are other reasons why you might want to manually create catalog tables and specify catalog tables as the crawler source:

You want to choose the catalog table name and not rely on the catalog table naming algorithm.

"

upvoted 31 times

Phoenyx89 3 years, 9 months ago

I agree is C. B is wrong because it takes more effort than B using Lambda, instead the Glue Crawler is full automated to update and find new partitions

upvoted 1 times

awssp12345 3 years, 8 months ago

I change my answer to C

upvoted 1 times

rsn 2 years, 3 months ago

However there is no info on scheduling the crawler in C. any thoughts?

upvoted 1 times

NarenKA Most Recent 1 year, 4 months ago

Selected Answer: A

I think Option A is correct. AWS Glue crawlers automatically scan data in S3, recognize the format and schema, and create metadata tables in the AWS Glue Data Catalog. This eliminates manual schema definition, table creation and meets new partitions with minimal effort. We can specify the database in which the tables are created and control the naming of the tables through the crawler's configuration settings rather than relying on an automated naming algorithm. As new data is added to S3 in hourly, daily, and yearly partitions, running the crawler at regular intervals ensures that new partitions are discovered and added to the respective Data Catalog tables automatically.

Option C uses the AWS Glue API to create a table, which is more manual than allowing a crawler to discover and manage tables and does not automatically address the ongoing discovery of new partitions.
upvoted 1 times

pk349 2 years, 1 month ago
C: I passed the test
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: C
Correct answer is C as the AWS Glue API CreateTable operation can be used to create a table in the Data Catalog and the AWS Glue crawler can be created and the table as the source can be specified. This meets the requirement of catalog table names and keeping the table updated with new data.

B is incorrect. Although creating a new catalog table is right, the use of a Lambda function to update the table partitions entails a lot of development work. Remember that it is explicitly stated in the scenario that the solution should be implemented with the least configuration overhead.
A is incorrect because you must create a new catalog table if you do not want to rely on the catalog table naming algorithm provided by AWS Glue.
D is incorrect because this solution entails a lot of effort. A better and easier solution is to just create a new table in AWS Glue Data Catalog and set up an AWS Glue crawler.
upvoted 3 times

bp339 2 years, 8 months ago
Selected Answer: C
The following are other reasons why you might want to manually create catalog tables and specify catalog tables as the crawler source:

You want to choose the catalog table name and not rely on the catalog table naming algorithm.

You want to prevent new tables from being created in the case where files with a format that could disrupt partition detection are mistakenly saved in the data source path.
upvoted 2 times

rocky48 2 years, 10 months ago
Selected Answer: C
Answer is C
upvoted 1 times

Bik000 3 years, 1 month ago
Selected Answer: C
My Answer is C
upvoted 1 times

rb39 3 years, 2 months ago
Selected Answer: C
You need to use the API to be able to provide a custom name
upvoted 1 times

lakediver 3 years, 6 months ago
Selected Answer: C
https://docs.aws.amazon.com/glue/latest/dg/tables-described.html#update-manual-tables
upvoted 4 times

aws2019 3 years, 7 months ago
Answer should be C.
upvoted 1 times

sayed 3 years, 7 months ago
C as per the linked provided not B
read "Updating Manually Created Data Catalog Tables Using Crawlers"
https://docs.aws.amazon.com/glue/latest/dg/tables-described.html
upvoted 1 times

lostsoul07 3 years, 8 months ago
C is the right answer
upvoted 2 times

tleflond 3 years, 8 months ago

I think there might be a miss typo in C, the table needs to be defined as target and not source

upvoted 1 times

zevzek 3 years, 8 months ago

It also mentions this

The following are other reasons why you might want to manually create catalog tables and specify catalog tables as the crawler source:

-You want to choose the catalog table name and not rely on the catalog table naming algorithm

upvoted 1 times

zevzek 3 years, 8 months ago

No typo I think..

https://docs.aws.amazon.com/glue/latest/dg/tables-described.html

Updating Manually Created Data Catalog Tables Using Crawlers:

To do this, when you define a crawler, instead of specifying one or more data stores as the source of a crawl, you specify one or more existing Data Catalog tables. The crawler then crawls the data stores specified by the catalog tables. In this case, no new tables are created; instead, your manually created tables are updated.

upvoted 2 times

LMax 3 years, 8 months ago

C for me.

upvoted 2 times

sanjaym 3 years, 8 months ago

Answer should be C.

upvoted 2 times

syu31svc 3 years, 8 months ago

Answer is C

upvoted 2 times

Paitan 3 years, 8 months ago

Answer is C.

upvoted 3 times

Question #67 *Topic 1*

A large university has adopted a strategic goal of increasing diversity among enrolled students. The data analytics team is creating a dashboard with data visualizations to enable stakeholders to view historical trends. All access must be authenticated using Microsoft Active Directory. All data in transit and at rest must be encrypted.
Which solution meets these requirements?

A. Amazon QuickSight Standard edition configured to perform identity federation using SAML 2.0. and the default encryption settings.

B. Amazon QuickSight Enterprise edition configured to perform identity federation using SAML 2.0 and the default encryption settings.

C. Amazon QuckSight Standard edition using AD Connector to authenticate using Active Directory. Configure Amazon QuickSight to use customer-provided keys imported into AWS KMS.

D. Amazon QuickSight Enterprise edition using AD Connector to authenticate using Active Directory. Configure Amazon QuickSight to use customer-provided keys imported into AWS KMS.

Suggested Answer: *D*
Reference:
https://docs.aws.amazon.com/quicksight/latest/user/WhatsNew.html

Community vote distribution
B (78%) D (22%)

kikakiko Highly Voted 3 years, 8 months ago
Answer is B
Authentication: https://docs.aws.amazon.com/quicksight/latest/user/external-identity-providers-setting-up-saml.html#external-identity-providers-config-idp
Encryption: https://docs.aws.amazon.com/quicksight/latest/user/data-encryption-at-rest.html
"All keys associated with Amazon QuickSight are managed by AWS." in https://docs.aws.amazon.com/quicksight/latest/user/key-management.html - No way to use customer-provided keys in QuickSight
upvoted 31 times

chengxu32 3 years, 7 months ago
"Single Sign On with SAML or OpenID Connect" is available in both Standard and Enterprise edition, which means both A and B are correct. Since this is not a multiple choice question, then both A and B are out. Active Directory is only available in Enterprise edition, so the answer is D
upvoted 5 times

Edwars 2 years, 11 months ago
A and B aren't both correct, because encryption at rest is only available with Entreprise edition
https://docs.aws.amazon.com/quicksight/latest/user/data-encryption.html
So, answer is B
upvoted 7 times

wally_1995 1 year, 12 months ago
I don't know how much has changed after 20 months since you posted this. But
According to:
https://docs.aws.amazon.com/quicksight/latest/user/aws-directory-service.html
If you have an existing directory that you want to use for Amazon QuickSight, you can use Active Directory Connector.

Since it's only available in enterprise, then the only option here to choose is D.

Also https://docs.aws.amazon.com/quicksight/latest/user/key-management.html gives a full tutorial on how to use a customer provided key as the encryption key
upvoted 4 times

GauravM17 Highly Voted 3 years, 8 months ago
All Keys are managed by QuickSight enterprise edition and hence D can not be the answer. I would go with B
upvoted 11 times

awsmonster Most Recent 1 year, 4 months ago
Selected Answer: B

Answer is B.

An AWS Organization is required to use IAM Identity Center with AD Connector. https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ad_connector_getting_started.html, https://docs.aws.amazon.com/quicksight/latest/user/aws-directory-service.html

Since the univerasity does not has an AWS Organization and did not mention that they need one. Option B would b a more feasible option
upvoted 1 times

LeoSantos121212121212121 1 year, 4 months ago
ChatGPT chose answer D.
upvoted 1 times

NarenKA 1 year, 4 months ago
Selected Answer: B
Option D mentioned using AD Connector and configuring Amazon QuickSight to use customer-provided keys imported into AWS Key Management Service (KMS). While using customer-provided keys in AWS KMS for encryption offers additional control over encryption keys, the question does not specify a requirement that necessitates this level of key management. Additionally, the AD Connector is not a feature of Amazon QuickSight; instead, the Enterprise edition supports direct AD integration.
Therefore, option B is the correct solution, as it leverages the capabilities of Amazon QuickSight Enterprise edition to meet the university's requirements for Active Directory authentication and data encryption.
upvoted 3 times

pn12345 1 year, 6 months ago
Selected Answer: D
Correct answer
upvoted 1 times

LocalHero 1 year, 7 months ago
Probably B is correct when the problem was created in the past.
but now D looks like also correct.(AD connecter looks like very easy)
Encryption method is not defined in the sentence.
so B is more correct.It is less effort.ummm
upvoted 1 times

nroopa 1 year, 10 months ago
Ans :D
upvoted 1 times

MLCL 1 year, 11 months ago
Selected Answer: B
B : SAML 2.0 works with AD,
Enterprise Edition offers encryption at rest.
All data is encrypted in transit by default https://docs.aws.amazon.com/quicksight/latest/user/data-encryption-in-transit.html
upvoted 1 times

pk349 2 years, 1 month ago
B: I passed the test
upvoted 1 times

r3mo 1 year, 9 months ago
Sure you pass the test. But this one you got it wrong. The answer is "D"
upvoted 5 times

Mirandaali 2 years, 2 months ago
Selected Answer: D
QuickSight enables you to encrypt your SPICE datasets using the keys you have stored in AWS Key Management Service. This provides you with the tools to audit access to data and satisfy regulatory security requirements. If you need to do so, you have the option to immediately lock down access to your data by revoking access to AWS KMS keys.
upvoted 1 times

tbhtp 2 years, 2 months ago
D seems to be right. Reason: it is stated that user authentication must be via Microsoft Active Directory. This rules out options A, B and C. A and B mention SAML, C mentions an AD Connector but this is only supported in the Enterprise Edition. Source: https://docs.aws.amazon.com/quicksight/latest/user/directory-integration.html and https://docs.aws.amazon.com/quicksight/latest/user/aws-

directory-service.html. And yes it is possible to use CMKs with aws managed KMS, even for SPICE data not just meta data. Source: https://docs.aws.amazon.com/quicksight/latest/user/key-management.html - Using customer-managed keys from AWS KMS with SPICE datasets in Amazon QuickSight

upvoted 2 times

rags1482 2 years, 3 months ago

D is the right answer

https://docs.aws.amazon.com/quicksight/latest/user/aws-directory-service.html

upvoted 2 times

akashm99101001com 2 years, 3 months ago

Selected Answer: D

To create customer-managed keys (CMKs), you use AWS Key Management Service (AWS KMS) in the same AWS account and AWS Region as the Amazon QuickSight SPICE dataset. A QuickSight administrator can then use a CMK to encrypt SPICE datasets and control access.

https://docs.aws.amazon.com/quicksight/latest/user/key-management.html

upvoted 2 times

akashm99101001com 2 years, 3 months ago

Key statement - "All data in transit and at rest must be encrypted." A and B are out

upvoted 1 times

Arjun777 2 years, 4 months ago

QuickSight enables you to encrypt your SPICE datasets using the keys you have stored in AWS Key Management Service. This provides you with the tools to audit access to data and satisfy regulatory security requirements. If you need to do so, you have the option to immediately lock down access to your data by revoking access to AWS KMS keys. All data access to encrypted datasets in QuickSight SPICE is logged in AWS CloudTrail. Administrators or auditors can trace data access in CloudTrail to identify when and where data was accessed.

To create customer-managed keys (CMKs), you use AWS Key Management Service (AWS KMS) in the same AWS account and AWS Region as the Amazon QuickSight SPICE dataset. A QuickSight administrator can then use a CMK to encrypt SPICE datasets and control access.

upvoted 1 times

SorenBendixen 2 years, 4 months ago

Answer should be D. This allow the use of customer managed keys: https://docs.aws.amazon.com/quicksight/latest/user/key-management.html And easy enabling of AD

upvoted 3 times

Ody__ 2 years, 5 months ago

Selected Answer: B

Answer is B

upvoted 1 times

Question #68 *Topic 1*

An airline has been collecting metrics on flight activities for analytics. A recently completed proof of concept demonstrates how the company provides insights to data analysts to improve on-time departures. The proof of concept used objects in Amazon S3, which contained the metrics in .csv format, and used Amazon
Athena for querying the data. As the amount of data increases, the data analyst wants to optimize the storage solution to improve query performance.
Which options should the data analyst use to improve performance as the data lake grows? (Choose three.)

A. Add a randomized string to the beginning of the keys in S3 to get more throughput across partitions.

B. Use an S3 bucket in the same account as Athena.

C. Compress the objects to reduce the data transfer I/O.

D. Use an S3 bucket in the same Region as Athena.

E. Preprocess the .csv data to JSON to reduce I/O by fetching only the document keys needed by the query.

F. Preprocess the .csv data to Apache Parquet to reduce I/O by fetching only the data blocks needed for predicates.

Suggested Answer: *ACE*

Community vote distribution

CDF (100%)

carol1522 Highly Voted 3 years, 9 months ago
For me is CDF
upvoted 26 times

GauravM17 3 years, 8 months ago
Parquet file is by default compressed which is convered under F. The answer should be A,D,F
upvoted 4 times

Woong Highly Voted 3 years, 8 months ago
A is not best practice any more. [Quoted]previously Amazon S3 performance guidelines recommended randomizing prefix naming with hashed characters to optimize performance for frequent data retrievals. You no longer have to randomize prefix naming for performance, and can use sequential date-based naming for your prefixes. [Unquoted]
upvoted 14 times

Abep 2 years, 9 months ago
@Woong Thanks for quoting the excerpt. This makes option "A" incorrect.
Sharing the link to this statement for anyone who wish to verify
https://d1.awsstatic.com/whitepapers/AmazonS3BestPractices.pdf
"This guidance supersedes any previous guidance on optimizing performance for Amazon S3. For example, previously Amazon S3 performance guidelines recommended randomizing prefix naming with hashed characters to optimize performance for frequent data retrievals. You no longer have to randomize prefix naming for performance, and can use sequential date-based naming for your prefixes"
upvoted 3 times

vicks316 3 years, 8 months ago
That's absolutely right, hence should be C,D,F
upvoted 3 times

monkeydba Most Recent 1 year, 7 months ago
The comment about random strings is also in this useful link:
https://docs.aws.amazon.com/whitepapers/latest/s3-optimizing-performance-best-practices/introduction.html
upvoted 1 times

Debi_mishra 2 years, 1 month ago
C and F are not doubt easy answers. But I believe A and D both are correct. People quoting randomize prefix not required - are ignoring "sequential date-based naming" which is also not mentioned in question and Athen can run longer as S3 list operations will take time without a well distributed prefix.
upvoted 1 times

pk349 2 years, 1 month ago

CDF: I passed the test

upvoted 3 times

henom 2 years, 7 months ago

A,C,F

Some data lake applications on Amazon S3 scan millions or billions of objects for queries that run over petabytes of data. In this scenario, millions of data points are stored on Amazon S3 and it is recommended to create a random string and add that to the beginning of the object prefixes to increase the read performance for S3 objects.

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: CDF

Correct answers are C, D & F

Options C & F as using compression and columnar data format helps improve query performance and optimize storage
Option D as using Athena and S3 within the same region would help with query performance and cost.

Option A is wrong as S3 scales automatically now and is not bounded by the restriction.

Option B is wrong as using the same account does not help in optimizing the cost of query performance.

Option E is wrong as using JSON is the same as using CSV files and does help in n optimizing the cost or query performance.

upvoted 8 times

rocky48 2 years, 11 months ago

Selected Answer: CDF

Selected Answer: CDF

upvoted 1 times

GiveMeEz 3 years ago

Ans A. can't be correct at all.
Adding randomized string will make partition size too small to reap the benefits.
https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/

upvoted 1 times

f4bi4n 3 years, 1 month ago

Selected Answer: CDF

C,D,F fulfills are needs

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: CDF

My Answer is C, D & F

upvoted 1 times

aws2019 3 years, 7 months ago

For me is CDF

upvoted 4 times

mickies9 3 years, 7 months ago

As a best practice, S3 and Athena should be in the same region and account and columnar based is appropriate for the performance. My answer would be BDF

upvoted 1 times

jueueuergen 3 years, 7 months ago

CDF.
Parquet compresses by default, yes, but there is also an "uncompressed" option. So C is not redundant.

upvoted 3 times

gunjan4392 3 years, 7 months ago

I think CDF

upvoted 1 times

Donell 3 years, 8 months ago

I goes with C,D,F.

upvoted 1 times

DerekKey 3 years, 8 months ago

C -> is WRONG in my opinion

1. How do you want to compress Apache Parquet that is already compressed by default? We selected Parquet as file format in F.

"For Athena, we recommend using either Apache Parquet or Apache ORC, which compress data by default and are splittable."

2. We still need prefixes but we don't have to randomize them

You can increase your read or write performance by parallelizing reads. For example, if you create 10 prefixes in an Amazon S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second.

BUT

You no longer have to randomize prefix naming for performance and can use sequential date-based naming for your prefixes.

3. You can not reduce data transfer I/O. I/O represents an entity that sends/receives data, therefore, you can only reduce parameters of I/O e.g. data transfer bandwidth, speed, no of operations (e.g. IOPS) etc.

upvoted 1 times

Question #69 *Topic 1*

A company uses the Amazon Kinesis SDK to write data to Kinesis Data Streams. Compliance requirements state that the data must be encrypted at rest using a key that can be rotated. The company wants to meet this encryption requirement with minimal coding effort.
How can these requirements be met?

A. Create a customer master key (CMK) in AWS KMS. Assign the CMK an alias. Use the AWS Encryption SDK, providing it with the key alias to encrypt and decrypt the data.

B. Create a customer master key (CMK) in AWS KMS. Assign the CMK an alias. Enable server-side encryption on the Kinesis data stream using the CMK alias as the KMS master key.

C. Create a customer master key (CMK) in AWS KMS. Create an AWS Lambda function to encrypt and decrypt the data. Set the KMS key ID in the function's environment variables.

D. Enable server-side encryption on the Kinesis data stream using the default KMS key for Kinesis Data Streams.

Suggested Answer: *B*
Reference:
https://aws.amazon.com/kinesis/data-streams/faqs/

Community vote distribution
B (83%) D (17%)

KoMo Highly Voted 3 years, 8 months ago
B.
https://docs.aws.amazon.com/streams/latest/dev/what-is-sse.html
upvoted 20 times

cloudlearnerhere Highly Voted 2 years, 7 months ago
Selected Answer: B
Correct answer is B as Kinesis Data Streams supports data at rest encryption using Server-Side encryption. Data is encrypted before persisting and decrypted before being read by the consumers and requires no changes to producers and consumers.

Options A & C are wrong as it would require coding effort.

Option D is wrong as the default key cannot be rotated.
upvoted 6 times

Debi_mishra Most Recent 2 years, 1 month ago
B is correct. D is wrong - AWS Managed keys are rotated but as per AWS not as per customer. Here customer want rotation capability and they might want to do it number of times in a year.
upvoted 1 times

pk349 2 years, 1 month ago
B: I passed the test
upvoted 1 times

rocky48 2 years, 11 months ago
Selected Answer: B
Selected Answer: B
upvoted 1 times

Ramshizzle 3 years ago
Selected Answer: B
It should be B. B describes the method to encrypt your data at rest inside your Kinesis Data Streams.
Option D is the only valid alternative given the constraints, but there is no valid method to rotate this key yourself. So I would argue this key is not rotatable.
upvoted 1 times

MWL 3 years, 1 month ago
Selected Answer: B

Vote for B.
I think "rotatable key" means you can rotate manually, it should be CMK, not AWS managed key.
D said "using the default KMS key", it is saying to use AWS managed key. So it's not right.
upvoted 2 times

Teraxs 3 years, 1 month ago

Selected Answer: D

I'd say D:
A and C are out because they involve more coding.
B would work, but key rotation of CMK is disabled by default and the answer did not say to enable it (but mentions creation and alias, so that was likely left out on purpose)
D works, is the simplest and the key is rotated by default (no every year, used to be every 3 years)
Paragraph "Customer managaged keys"
in https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html#rotate-aws-owned-keys
upvoted 2 times

MWL 3 years, 1 month ago

I think "rotatable key" means you can rotate manually, it should be CMK, not AWS managed key.
upvoted 2 times

jrheen 3 years, 2 months ago

Answer - B
upvoted 1 times

aws2019 3 years, 7 months ago

B is the right answer
upvoted 1 times

lostsoul07 3 years, 7 months ago

B is the right answer
upvoted 2 times

jay1ram2 3 years, 8 months ago

The Answer is B. You cannot rotate "AWS Managed CMK" i.e. Default keys. It is automatically rotated every 3 years.

https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html
upvoted 3 times

Subho_in 3 years, 8 months ago

"key that can be rotated" it is talking about CMK. B must be the answer.
upvoted 2 times

gtourkas 3 years, 8 months ago

Just checked the Kinesis console. You can select either the default or the CMK for encryption at rest. Since rotation can be set for the CMK, B is the answer.
upvoted 3 times

Shivibaheti 3 years, 8 months ago

D.

the FAQ's https://aws.amazon.com/kinesis/data-streams/faqs/#kinesis-encryption
question :
"What is server-side encryption"
It mentions "Server-side encryption for Kinesis Data Streams automatically encrypts data using a user specified AWS KMS master key (CMK) "
upvoted 2 times

sly_tail 2 years, 3 months ago

No, there is no such thing as default key for KDS. It's B
upvoted 1 times

Draco31 3 years, 8 months ago

B.
it's written here: https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html
You cannot rotate key that you did not create
upvoted 3 times

Sent1 3 years, 8 months ago

It should be B.

Here is the link,

https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html

upvoted 1 times

Question #70 *Topic 1*

A company wants to enrich application logs in near-real-time and use the enriched dataset for further analysis. The application is running on Amazon EC2 instances across multiple Availability Zones and storing its logs using Amazon CloudWatch Logs. The enrichment source is stored in an Amazon DynamoDB table.
Which solution meets the requirements for the event collection and enrichment?

A. Use a CloudWatch Logs subscription to send the data to Amazon Kinesis Data Firehose. Use AWS Lambda to transform the data in the Kinesis Data Firehose delivery stream and enrich it with the data in the DynamoDB table. Configure Amazon S3 as the Kinesis Data Firehose delivery destination.

B. Export the raw logs to Amazon S3 on an hourly basis using the AWS CLI. Use AWS Glue crawlers to catalog the logs. Set up an AWS Glue connection for the DynamoDB table and set up an AWS Glue ETL job to enrich the data. Store the enriched data in Amazon S3.

C. Configure the application to write the logs locally and use Amazon Kinesis Agent to send the data to Amazon Kinesis Data Streams. Configure a Kinesis Data Analytics SQL application with the Kinesis data stream as the source. Join the SQL application input stream with DynamoDB records, and then store the enriched output stream in Amazon S3 using Amazon Kinesis Data Firehose.

D. Export the raw logs to Amazon S3 on an hourly basis using the AWS CLI. Use Apache Spark SQL on Amazon EMR to read the logs from Amazon S3 and enrich the records with the data from DynamoDB. Store the enriched data in Amazon S3.

Suggested Answer: *C*

Community vote distribution

awssp12345 Highly Voted 3 years, 9 months ago

The answer is A - Since they are already using CloudWatch Logs, it makes sense to send the CW logs to KFH which will invoke lambda and send data to S3 for further analysis.

upvoted 38 times

[Removed] 3 years, 5 months ago

KDA can refrer s3 only for data enrichment.

upvoted 3 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: A

Correct answer is A as CloudWatch logs can be integrated with Kinesis Data Firehose using subscription filters, the data can be enriched using Lambda doing a lookup on the DynamoDB tables and data storage to S3.

Options B & D are wrong as exporting the logs would not provide near-real-time data handling.

Option C is wrong as using Kinesis Data Stream for data collection with agents would increase overhead, also Kinesis Data Analytics does not support DynamoDB as a reference data source for enrichment.

upvoted 10 times

pk349 Most Recent 2 years, 1 month ago

A: I passed the test

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: A

Selected Answer: A

upvoted 1 times

GarfieldBin 3 years ago

Selected Answer: A

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html#FirehoseExample. B, D are obviously wrong. DynamoDB is not for joining.

upvoted 2 times

renfdo 2 years, 6 months ago

Agree!
upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: A

My Answer is A
upvoted 1 times

ShilaP 3 years, 3 months ago

A is the right answer
upvoted 2 times

ses13 3 years, 5 months ago

The answer is C - Since CW logs is written in gzip format to Kinesis Firehose. In the Lambda function the uncompress command should be applied first. https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html#FirehoseExample
upvoted 2 times

Ramshizzle 3 years ago

You can uncompress inside the Lambda without any issue. So answer A works.
upvoted 1 times

arun004 3 years, 6 months ago

How KDF refer data stored in DynamoDB ? i don't think it is possible
upvoted 1 times

Ramshizzle 3 years ago

Lambda does the enrichment and so the Lambda needs to read the DynamoDB table. This is perfectly fine.
upvoted 1 times

aws2019 3 years, 7 months ago

A is the right answer
upvoted 1 times

Gekko 3 years, 7 months ago

Why not D ?
https://aws.amazon.com/es/blogs/big-data/analyze-your-data-on-amazon-dynamodb-with-apache-spark/
upvoted 1 times

Dr_Kiko 3 years, 7 months ago

because hourly is not near realtime
upvoted 4 times

lostsoul07 3 years, 8 months ago

A is the right answer
upvoted 4 times

Lucas88 3 years, 8 months ago

The thing that confuses me about A is the combination of Firehose and Lambda, is it even possible to add a Lambda processing step in Firehose?
upvoted 2 times

Lucas88 3 years, 8 months ago

Never mind, it is indeed possible!
upvoted 3 times

SA_206 3 years, 8 months ago

Here Key word ' multiple Availability Zones' is crucial. We need to export the log from differnet zones.
upvoted 1 times

angadaws 3 years, 8 months ago

C - is possible too -> KDS-> KDA join Dynamo -> KDF->s3
https://docs.aws.amazon.com/firehose/latest/dev/writing-with-agents.html
https://aws.amazon.com/blogs/big-data/joining-and-enriching-streaming-data-on-amazon-kinesis/

both solution can be near real time .
A - only because CW logs is used ?? Tricky Question
upvoted 3 times

angadaws 3 years, 8 months ago

K Agent ->KDS-> KDA join Dynamo -> KDF->s3

upvoted 1 times

angadaws 3 years, 8 months ago

Ignore C <<< KDA join Dynamo not possible . (reference data on S3 would have worked)

A works well <<<<

upvoted 2 times

LMax 3 years, 9 months ago

Only A provides near-real-time. Rest would have much longer delay.

upvoted 1 times

syu31svc 3 years, 9 months ago

Answer is A

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/SubscriptionFilters.html#FirehoseExample

upvoted 2 times

Question #71 *Topic 1*

A company uses Amazon Redshift as its data warehouse. A new table has columns that contain sensitive data. The data in the table will eventually be referenced by several existing queries that run many times a day.
A data analyst needs to load 100 billion rows of data into the new table. Before doing so, the data analyst must ensure that only members of the auditing group can read the columns containing sensitive data.
How can the data analyst meet these requirements with the lowest maintenance overhead?

A. Load all the data into the new table and grant the auditing group permission to read from the table. Load all the data except for the columns containing sensitive data into a second table. Grant the appropriate users read-only permissions to the second table.

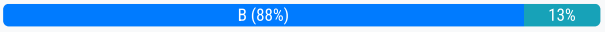
B. Load all the data into the new table and grant the auditing group permission to read from the table. Use the GRANT SQL command to allow read-only access to a subset of columns to the appropriate users.

C. Load all the data into the new table and grant all users read-only permissions to non-sensitive columns. Attach an IAM policy to the auditing group with explicit ALLOW access to the sensitive data columns.

D. Load all the data into the new table and grant the auditing group permission to read from the table. Create a view of the new table that contains all the columns, except for those considered sensitive, and grant the appropriate users read-only permissions to the table.

Suggested Answer: *D*

Community vote distribution



esuaaaa Highly Voted 3 years, 8 months ago

It's B.
https://aws.amazon.com/jp/about-aws/whats-new/2020/03/announcing-column-level-access-control-for-amazon-redshift/
upvoted 22 times

lakeswimmer 3 years, 6 months ago

B it is
For row level access I guess only option to create views, any thoughts?
upvoted 2 times

lakediver 3 years, 6 months ago

grant select(cust_name, cust_phone) on cust_profile to user1;
upvoted 2 times

rsn 2 years, 3 months ago

B talks about providing access at a user level rather than a group. It this not an operational overhead?
upvoted 1 times

cloudlearnerhere Highly Voted 2 years, 7 months ago

Selected Answer: B
Correct answer is B as Redshift supports column-level access control, which works best with the table-level access control without having to implement views.

Option A is wrong as it increases maintenance overhead.

Option C is wrong as IAM policy does not help provide column-level access control.

Option D is wrong as using Redshift column-level access control is better than views.
upvoted 6 times

pk349 Most Recent 2 years, 1 month ago

B: I passed the test
upvoted 1 times

anjuvinayan 2 years, 1 month ago

got how many questions from the dump?

upvoted 1 times

akashm99101001com 2 years, 3 months ago

Selected Answer: B

GRANT can be used to assume an IAM role as well which covers options C as well.
https://docs.aws.amazon.com/redshift/latest/dg/r_GRANT-usage-notes.html#r_GRANT-usage-notes-assumerole

upvoted 1 times

akashm99101001com 2 years, 3 months ago

"The data in the table will eventually be referenced by several existing queries that run many times a day."
If the view is based on a complex query that joins many tables or performs many calculations, it can be slow to query. If the view is based on a large amount of data, it can also be slow to query.

upvoted 2 times

nharaz 2 years, 8 months ago

B is correct According to Stephane Maarek course on Udemy
Since March 2020, Amazon Redshift supports column-level access control for data in Redshift. Customers can use column-level GRANT and REVOKE statements to help meet their security and compliance needs.

Redshift's table-level access controls for the data in Redshift are already in use by many customers, but they also want the ability to control access in more detail. You can now control access to columns without having to implement view-based access control or use another system. Column-level access control is available in all Amazon Redshift regions.

GRANT command defines access privileges for a user or user group. Privileges include access options such as being able to read data in tables and views, write data, create tables, and drop tables. Use this command to give specific privileges for a table, database, schema, function, procedure, language, or column.

upvoted 4 times

aefuen1 2 years, 8 months ago

Selected Answer: B

B. Column level access control is available in redshift.

upvoted 1 times

LukeTran3206 2 years, 8 months ago

Selected Answer: C

the key is lowest maintenance overhead!!
if you grant access permission using SQL, you will facing with endless maintenance

upvoted 1 times

dushmantha 2 years, 11 months ago

Selected Answer: C

I will choose "C". Because its easy for me to grant read only access for any user for non sensitive data. And to allow only auditers to access sensitive data. Not other way around as given in "B".

upvoted 1 times

carbita 2 years, 11 months ago

Selected Answer: B

Its B, remember that create a view is not a good practice and might have leak of data. The best practice is to GRANT

upvoted 2 times

rocky48 2 years, 11 months ago

Selected Answer: B

B is the right answer.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: B

Answer is B

upvoted 1 times

certificationJunkie 3 years, 1 month ago

B and C are very similar. The only advantage for B is that there is a single role assigned to auditors to access all the columns. While in case of C, auditors will access few columns via public role and few senstive columns via another role created specific to them.

upvoted 1 times

jrheen 3 years, 2 months ago

Answer - B

upvoted 1 times

ay12 3 years, 2 months ago

Selected Answer: B

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

upvoted 2 times

aws2019 3 years, 7 months ago

B is the ans

upvoted 1 times

Marcinha 3 years, 8 months ago

It's D. Much easier to create a view than to insert in 1 new table.

upvoted 1 times

Marcinha 3 years, 7 months ago

Changed for D

upvoted 1 times

Marcinha 3 years, 7 months ago

Changed for B

upvoted 1 times

mickies9 3 years, 8 months ago

Why not C? all other team should have access to the table except for the sensitive data columns right? Options B is providing Audit team permission to the table and then granting access to the column again. Is that second step even necessary?

upvoted 2 times

dushmantha 2 years, 11 months ago

That's what I thought too. Allowing a very limited set of users to access sensitive columns is much easier

upvoted 1 times

Question #72 *Topic 1*

A banking company wants to collect large volumes of transactional data using Amazon Kinesis Data Streams for real-time analytics. The company uses
PutRecord to send data to Amazon Kinesis, and has observed network outages during certain times of the day. The company wants to obtain exactly once semantics for the entire processing pipeline.
What should the company do to obtain these characteristics?

A. Design the application so it can remove duplicates during processing be embedding a unique ID in each record.

B. Rely on the processing semantics of Amazon Kinesis Data Analytics to avoid duplicate processing of events.

C. Design the data producer so events are not ingested into Kinesis Data Streams multiple times.

D. Rely on the exactly one processing semantics of Apache Flink and Apache Spark Streaming included in Amazon EMR.

Suggested Answer: *A*
Reference:
https://docs.aws.amazon.com/streams/latest/dev/kinesis-record-processor-duplicates.html

Community vote distribution
A (100%)

testtaker3434 Highly Voted 3 years, 9 months ago
Agree with A.
upvoted 17 times

awssp12345 3 years, 9 months ago
me too!
upvoted 2 times

vicks316 Highly Voted 3 years, 9 months ago
A.
https://docs.aws.amazon.com/streams/latest/dev/kinesis-record-processor-duplicates.html
"Applications that need strict guarantees should embed a primary key within the record to remove duplicates later when processing."
upvoted 9 times

Palee Most Recent 3 months, 2 weeks ago
Selected Answer: D
Ans D
Option A doesn't talk about "Obtain exactly once semantics for the entire processing pipeline
upvoted 1 times

daisyli 1 year, 7 months ago
D
Apache Flink provides a powerful API to transform, aggregate, and enrich events, and supports exactly-once semantics. Apache Flink is therefore a good foundation for the core of your streaming architecture.
https://aws.amazon.com/blogs/big-data/streaming-etl-with-apache-flink-and-amazon-kinesis-data-analytics/
upvoted 2 times

pk349 2 years, 1 month ago
A: I passed the test
upvoted 1 times

enoted 2 years, 4 months ago
Selected Answer: A
A - exactly what is requested in the description
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: A
Correct answer is A as producer retries can result in duplicates in Kinesis Data Streams and must be handled by the producer by using a unique key for each message.

There are two primary reasons why records may be delivered more than one time to your Amazon Kinesis Data Streams application: producer retries and consumer retries. Your application must anticipate and appropriately handle processing individual records multiple times.

Options B & C are wrong as they would not handle the exactly-once processing semantics.

Option D is wrong as although Apache Flink and Spark Streaming would work, it would need a complete change in the current application.
upvoted 3 times

rocky48 2 years, 11 months ago

Selected Answer: A

Selected Answer: A
upvoted 1 times

certificationJunkie 3 years, 1 month ago

application should be idempotent. Can be achieved by including a primary key in the record. Hence, A is correct answer
upvoted 1 times

MWL 3 years, 1 month ago

Selected Answer: A

The problem is the question is, the producer may put the records several times. Donell explained this very well.
KDA, Flink or Spark can only make sure 'Exactly once' for every record in stream. If the record is duplicated by producer, they will process exactly once for every record in stream.
So the answer should be A.
upvoted 2 times

jrheen 3 years, 2 months ago

Answer : A
upvoted 1 times

aws2019 3 years, 7 months ago

A is right
upvoted 1 times

Donell 3 years, 8 months ago

Answer A. Design the application so it can remove duplicates during processing by embedding a unique ID in each record.
Producer Retries
Consider a producer that experiences a network-related timeout after it makes a call to PutRecord, but before it can receive an acknowledgement from Amazon Kinesis Data Streams. The producer cannot be sure if the record was delivered to Kinesis Data Streams. Assuming that every record is important to the application, the producer would have been written to retry the call with the same data. If both PutRecord calls on that same data were successfully committed to Kinesis Data Streams, then there will be two Kinesis Data Streams records. Although the two records have identical data, they also have unique sequence numbers. Applications that need strict guarantees should embed a primary key within the record to remove duplicates later when processing. Note that the number of duplicates due to producer retries is usually low compared to the number of duplicates due to consumer retries.

Reference: https://docs.aws.amazon.com/streams/latest/dev/kinesis-record-processor-duplicates.html
upvoted 8 times

Heer 3 years, 8 months ago

ANSWER:D
KDS and KDF has 'exactly once' semantics .Option A is a fail safe mechanism when there is a choppy network while sending data to KDS with PutRecord.
Apache Flink and Apache Spark both guarantees 'Exactly once' semantics and which is what is the requirement as per the question .
upvoted 4 times

MWL 3 years, 1 month ago

The problem is the question is, the producer may put the records several times. Donell explained this very well. So the answer should be A.
Flink or Spark can only make sure 'Exactly once' for every record in stream. If the record is duplicated by producer, they don't work.
upvoted 1 times

lostsoul07 3 years, 8 months ago

A is the right answer
upvoted 3 times

Draco31 3 years, 8 months ago

A was a good choice until i search EMR Flink...:

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

Apache Flink is a streaming dataflow engine that you can use to run real-time stream processing on high-throughput data sources. Flink supports event time semantics for out-of-order events, exactly-once semantics, backpressure control, and APIs optimized for writing both streaming and batch applications.

Can be connected to KDS.

So i will pick up D

upvoted 2 times

liyungho 3 years, 8 months ago

According to Flink kinesis connector doc -- https://ci.apache.org/projects/flink/flink-docs-stable/dev/connectors/kinesis.html, in the Kinesis producer section,

it states "Note that the producer is not participating in Flink's checkpointing and doesn't provide exactly-once processing guarantees. Also, the Kinesis producer does not guarantee that records are written in order to the shards (See here and here for more details).

In case of a failure or a resharding, data will be written again to Kinesis, leading to duplicates. This behavior is usually called "at-least-once" semantics."

So I think the answer is A.

upvoted 2 times

omar_bahrain 3 years, 8 months ago

There are documents that link streaming (Kafka/Kinesis) and EMR/Spark/Flink to remove deduplication in realtime.

https://blog.griddynamics.com/in-stream-deduplication-with-spark-amazon-kinesis-and-s3/

in addition to the difficulty of changing running application, I would say D is a good potential candidate

upvoted 1 times

syu31svc 3 years, 8 months ago

Link provided supports A as the answer

upvoted 1 times

Question #73 *Topic 1*

A company's data analyst needs to ensure that queries run in Amazon Athena cannot scan more than a prescribed amount of data for cost control purposes.
Queries that exceed the prescribed threshold must be canceled immediately.
What should the data analyst do to achieve this?

A. Configure Athena to invoke an AWS Lambda function that terminates queries when the prescribed threshold is crossed.

B. For each workgroup, set the control limit for each query to the prescribed threshold.

C. Enforce the prescribed threshold on all Amazon S3 bucket policies

D. For each workgroup, set the workgroup-wide data usage control limit to the prescribed threshold.

Suggested Answer: *D*
Reference:
https://docs.aws.amazon.com/athena/latest/ug/workgroups-setting-control-limits-cloudwatch.html

Community vote distribution
B (100%)

lostsoul07 Highly Voted 3 years, 8 months ago
B is the right answer
upvoted 20 times

syu31svc Highly Voted 3 years, 8 months ago
From the link: https://docs.aws.amazon.com/athena/latest/ug/workgroups-setting-control-limits-cloudwatch.html
"The per-query control limit specifies the total amount of data scanned per query. If any query that runs in the workgroup exceeds the limit, it is canceled"
Answer is B
upvoted 15 times

pk349 Most Recent 2 years, 1 month ago
B: I passed the test
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago
Selected Answer: B
Correct answer is B as Athena Workgroups help set control limits and per-query control limit helps specific a limit which if exceeded by a query it would be canceled.

A is wrong as you can't configure Atehna for this purpose

C is incorrect because you can't set a threshold in Athena using S3 bucket policies.

D is incorrect because the workgroup-wide data usage control limit specifies the total amount of data scanned for all queries that run in the entire workgroup, and not on a specific query only. Remember that the requirement is to immediately cancel queries that exceed the recommended threshold.
upvoted 8 times

rocky48 2 years, 11 months ago
Selected Answer: B
Selected Answer: B
upvoted 1 times

Bik000 3 years, 1 month ago
Selected Answer: B
Answer is B
upvoted 2 times

Marvel_jarvis 3 years, 6 months ago

why not A or C? Can we set RecordMaxBufferTime to control scan?

upvoted 1 times

aws2019 3 years, 7 months ago

B is the right answer

upvoted 1 times

Donell 3 years, 7 months ago

Answer B

upvoted 1 times

Starboy 3 years, 7 months ago

B is the right answer

upvoted 3 times

AjNapa 3 years, 8 months ago

Will go with B for this scenario. I think the use case if to set a per query limit

upvoted 2 times

KoMo 3 years, 8 months ago

"per-query control limit" vs "workgroup-wide data usage control limit"

https://docs.aws.amazon.com/athena/latest/ug/workgroups-setting-control-limits-cloudwatch.html

upvoted 2 times

Paitan 3 years, 8 months ago

B and D both serve similar purpose. However in this example B is the right option.

https://docs.aws.amazon.com/athena/latest/ug/workgroups-setting-control-limits-cloudwatch.html

upvoted 3 times

jersyl 3 years, 8 months ago

It is B based on this link:

https://docs.aws.amazon.com/athena/latest/ug/manage-queries-control-costs-with-workgroups.html

upvoted 3 times

awssp12345 3 years, 9 months ago

I apologize, the Answer is B.

upvoted 3 times

awssp12345 3 years, 9 months ago

Agree with D.

upvoted 1 times

Question #74 *Topic 1*

A marketing company is using Amazon EMR clusters for its workloads. The company manually installs third-party libraries on the clusters by logging in to the master nodes. A data analyst needs to create an automated solution to replace the manual process.
Which options can fulfill these requirements? (Choose two.)

A. Place the required installation scripts in Amazon S3 and execute them using custom bootstrap actions.

B. Place the required installation scripts in Amazon S3 and execute them through Apache Spark in Amazon EMR.

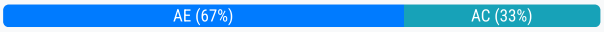
C. Install the required third-party libraries in the existing EMR master node. Create an AMI out of that master node and use that custom AMI to re-create the EMR cluster.

D. Use an Amazon DynamoDB table to store the list of required applications. Trigger an AWS Lambda function with DynamoDB Streams to install the software.

E. Launch an Amazon EC2 instance with Amazon Linux and install the required third-party libraries on the instance. Create an AMI and use that AMI to create the EMR cluster.

Suggested Answer: *AC*

Community vote distribution



ramozo Highly Voted 3 years, 9 months ago
I will choose A and E.
https://aws.amazon.com/about-aws/whats-new/2017/07/amazon-emr-now-supports-launching-clusters-with-custom-amazon-linux-amis/

https://docs.aws.amazon.com/de_de/emr/latest/ManagementGuide/emr-plan-bootstrap.html
upvoted 25 times

testtaker3434 3 years, 9 months ago
Doubt in this one... Documentation says you use E as a option to avoid A.
https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-custom-ami.html
upvoted 3 times

testtaker3434 3 years, 9 months ago
Its A and E, if you do one or the other, but you shouldn't do both (as if it was a step 1 and after that step 2)
upvoted 4 times

awssp12345 3 years, 9 months ago
Agreed!
upvoted 1 times

Paitan Highly Voted 3 years, 8 months ago
A and E.
upvoted 8 times

roymunson Most Recent 1 year, 7 months ago
AE: I'll pass the test.
upvoted 3 times

LocalHero 1 year, 7 months ago
I think existing cluster must not change.
It is danger.
Installing software for new EC2 instance is more safe.
so A and E correct.I think.
upvoted 2 times

confuzz 1 year, 11 months ago
AE
Custom AMIs created from the base EMR AMI are not supported and will lead to application provisioning errors upon cluster startup.
https://medium.com/@amberrunnels/creating-a-custom-ami-on-amazon-emr-a60ddeb7821b
upvoted 2 times

Debi_mishra 2 years, 1 month ago

A is very obvious. Between C and E - I will prefer E , as creating AMI from a master node may create a bulky AMI with lot of redundant hadoop libraries that can be done during bootstrap process.

upvoted 3 times

pk349 2 years, 1 month ago

AE: I passed the test

upvoted 1 times

VijiTu 2 years, 6 months ago

AE

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-custom-ami.html#emr-custom-ami-preconfigure

upvoted 1 times

nharaz 2 years, 8 months ago

A AND E According to Stephane Maarek Udemy course

A= You can use a bootstrap action to install additional software or customize the configuration of the EMR cluster instances. Bootstrap actions are scripts that run on the cluster after Amazon EMR launches the instance using the Amazon Linux Amazon Machine Image (AMI). Bootstrap actions run before Amazon EMR installs the applications that you specify when you create the cluster and before cluster nodes begin processing data.

E= You can create Amazon EMR clusters that have custom Amazon Machine Images (AMI) running Amazon Linux. You can create the AMI from an EC2 instance running Amazon Linux. Make sure that you have installed all the required third-party libraries on this EC2 instance. This allows you to preload additional software on your AMI and use these AMIs to launch your EMR clusters.

upvoted 1 times

JHJHJHJHJ 2 years, 9 months ago

AC Confirmed by paid dumps

upvoted 1 times

JoellaLi 2 years, 8 months ago

Could you give the reason for C?

upvoted 1 times

sly_tail 2 years, 3 months ago

C instead of E because it's absolutely redundant and inefficient to launch another instance when you already have the same master node.

upvoted 1 times

Sanmeda 2 years, 9 months ago

Answer A & E

upvoted 1 times

rocky48 2 years, 11 months ago

Selected Answer: AE

A and E.

upvoted 1 times

Bik000 3 years, 1 month ago

Selected Answer: AE

Answer is A & E

upvoted 1 times

certificationJunkie 3 years, 1 month ago

A and E are right answers. How would installing libraries on Master Nodes resolve anything? Computation happens on Data nodes (slaves) and all required packages should be installed there.

upvoted 3 times

Ryo0w0o 2 years, 7 months ago

But the quention says the company manually installs third-party libraries on the clusters "by logging in to the master nodes". Does it mean that they log in there but install the libraries in slave-nodes?

upvoted 2 times

MWL 3 years, 2 months ago

Selected Answer: AC

Although most of others choose A.E. But I think C is right instead of E.

For E: Launch EC2 instance and install these softwares are not easy for EMR. I have installed hadoop and skark one time. And it took my much time. And if I want to make a hadoop/hive/spark... environment to be used as AWS EMR, it will take much efford.

But C: I can login into master node with ssh, install the lib, and use the master node EC2 instance to cretae a custom AMI. Although it will waste the previous EMR cluster, but the if I want to establish an hadoop/spark/hive clusters using EC2, I still need several instances to prepare AMI.
So, I vote for AC.
upvoted 2 times

MWL 3 years, 2 months ago

Although most of others choose A.E. But I think C is right instead of E.
For E: Launch EC2 instance and install these softwares are not easy for EMR. I have installed hadoop and skark one time. And it took my much time. And if I want to make a hadoop/hive/spark... environment to be used as AWS EMR, it will take much efford.
But C: I can login into master node with ssh, install the lib, and use the master node EC2 instance to cretae a custom AMI. Although it will waste the previous EMR cluster, but the if I want to establish an hadoop/spark/hive clusters using EC2, I still need several instances to prepare AMI.
So, I vote for AC.
upvoted 1 times

RSSRAO 3 years, 4 months ago

Selected Answer: AE

A and E is correct
upvoted 2 times

Question #75 *Topic 1*

A data engineering team within a shared workspace company wants to build a centralized logging system for all weblogs generated by the space reservation system. The company has a fleet of Amazon EC2 instances that process requests for shared space reservations on its website. The data engineering team wants to ingest all weblogs into a service that will provide a near-real-time search engine. The team does not want to manage the maintenance and operation of the logging system.
Which solution allows the data engineering team to efficiently set up the web logging system within AWS?

A. Set up the Amazon CloudWatch agent to stream weblogs to CloudWatch logs and subscribe the Amazon Kinesis data stream to CloudWatch. Choose Amazon OpenSearch Service (Amazon Elasticsearch Service) as the end destination of the weblogs.

B. Set up the Amazon CloudWatch agent to stream weblogs to CloudWatch logs and subscribe the Amazon Kinesis Data Firehose delivery stream to CloudWatch. Choose Amazon OpenSearch Service (Amazon Elasticsearch Service) as the end destination of the weblogs.

C. Set up the Amazon CloudWatch agent to stream weblogs to CloudWatch logs and subscribe the Amazon Kinesis data stream to CloudWatch. Configure Splunk as the end destination of the weblogs.

D. Set up the Amazon CloudWatch agent to stream weblogs to CloudWatch logs and subscribe the Amazon Kinesis Firehose delivery stream to CloudWatch. Configure Amazon DynamoDB as the end destination of the weblogs.

Suggested Answer: *A*

Community vote distribution

Nicki1013 Highly Voted 3 years, 9 months ago

My answer is B

upvoted 25 times

awssp12345 3 years, 9 months ago

Agreed with B.

upvoted 3 times

GeeBeeEl 3 years, 8 months ago

Do you have a link to back this up? check https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/CWL_ES_Stream.html

upvoted 3 times

pk349 Most Recent 2 years, 1 month ago

B: I passed the test

upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B as Kinesis Data Firehose provides a managed solution to integrate with CloudWatch logs and stream data into ElasticSearch.

Option A is wrong as Kinesis Data Stream would increase the maintenance and operation of the logging system in terms of data collections and ingestion to ElasticSearch.

Option C is wrong as Kinesis Data Stream would increase the maintenance and operation of the logging system and Splunk would need a separate purchase.

Option D is wrong as Kinesis Firehose does not support DynamoDB as its destination and DynamoDB is not an ideal storage solution for logs.

upvoted 3 times

thirukudil 2 years, 8 months ago

Selected Answer: B

near real-time search engine for weblogs -> Opensearch. KDF can ingest the logs directly into OS.

upvoted 2 times

JHJHJHJHJ 2 years, 9 months ago

Answer B - KDS cannot write to open search

---

upvoted 3 times

rocky48 2 years, 10 months ago

Selected Answer: B

Answer is B

upvoted 2 times

Bik000 3 years, 1 month ago

Selected Answer: B

I think Answer should be B

upvoted 2 times

Dr_Kiko 3 years, 7 months ago

it's A or B because of Elasticsearch as destination; KDS cannot feed from CWLogs but Firehose can, hence the answer is B

upvoted 2 times

lostsoul07 3 years, 8 months ago

B is the right answer

upvoted 3 times

Lucas88 3 years, 8 months ago

It is B. They should use Firehose, not Data Streams (A). This sentence gives it away: "The team does not want to manage the maintenance and operation of the logging system."

upvoted 3 times

syu31svc 3 years, 8 months ago

I agree with A; Firehose can't stream to CloudWatch

upvoted 1 times

syu31svc 3 years, 8 months ago

Sorry I meant B

upvoted 3 times

Paitan 3 years, 8 months ago

B is the right option.

upvoted 3 times

Question #76 *Topic 1*

A company wants to research user turnover by analyzing the past 3 months of user activities. With millions of users, 1.5 TB of uncompressed data is generated each day. A 30-node Amazon Redshift cluster with 2.56 TB of solid state drive (SSD) storage for each node is required to meet the query performance goals.
The company wants to run an additional analysis on a year's worth of historical data to examine trends indicating which features are most popular. This analysis will be done once a week.
What is the MOST cost-effective solution?

A. Increase the size of the Amazon Redshift cluster to 120 nodes so it has enough storage capacity to hold 1 year of data. Then use Amazon Redshift for the additional analysis.

B. Keep the data from the last 90 days in Amazon Redshift. Move data older than 90 days to Amazon S3 and store it in Apache Parquet format partitioned by date. Then use Amazon Redshift Spectrum for the additional analysis.

C. Keep the data from the last 90 days in Amazon Redshift. Move data older than 90 days to Amazon S3 and store it in Apache Parquet format partitioned by date. Then provision a persistent Amazon EMR cluster and use Apache Presto for the additional analysis.

D. Resize the cluster node type to the dense storage node type (DS2) for an additional 16 TB storage capacity on each individual node in the Amazon Redshift cluster. Then use Amazon Redshift for the additional analysis.

Suggested Answer: *B*

Community vote distribution

B (100%)

ramozo Highly Voted 3 years, 9 months ago

B. Redshift Spectrum. "Amazon Redshift Spectrum executes queries across thousands of parallelized nodes to deliver fast results, regardless of the complexity of the query or the amount of data. "
https://aws.amazon.com/redshift/features/
upvoted 33 times

awssp12345 3 years, 9 months ago

Agree!
upvoted 1 times

lui 3 years, 8 months ago

why 30 node can save 90 days data?
upvoted 1 times

Paitan 3 years, 8 months ago

You are right, 30 nodes cannot save 90 days uncompressed data. But we can always compress the data while storing in Redshift. So that will definitely reduce the storage requirement and can be managed by the 30 nodes.
upvoted 5 times

Huy 3 years, 8 months ago

https://aws.amazon.com/blogs/aws/data-compression-improvements-in-amazon-redshift/
upvoted 2 times

pk349 Most Recent 2 years, 1 month ago

B: I passed the test
upvoted 1 times

cloudlearnerhere 2 years, 7 months ago

Selected Answer: B

Correct answer is B as the data can be stored in Redshift for 90 days for analyzing the past 3 months of user activities. Data older than 90 days can be moved to S3 and analyzed using Redshift Spectrum once a week. This provides the most cost-effective solution.

Using Amazon Redshift Spectrum, you can efficiently query and retrieve structured and semistructured data from files in Amazon S3 without having to load the data into Amazon Redshift tables. Redshift Spectrum queries employ massive parallelism to run very fast against large datasets. Much of the processing occurs in the Redshift Spectrum layer, and most of the data remains in Amazon S3. Multiple clusters can concurrently query the same