AWS Data Analytics Specialty Exams

Contents

[Practice Test 1 1](#_Toc82882239)

[Practice Test 2 2](#_Toc82882240)

[Practice Test 3 10](#_Toc82882241)

[Practice Test 4 11](#_Toc82882242)

[Practice Test 5 12](#_Toc82882243)

[Practice Test 6 13](#_Toc82882244)

# Practice Test 1

1. Log analysis of auto scaling EC2 instances

AWS Kinesis Producer Library -> data stream -> Firehose -> Amazon Elasticsearch Service -> Kibana

QuickSight and Elasticsearch do not have direct integration

CloudWatch aggreagates data at 1-minute

Cloudwatch subscription only for data already in CloudWatch

1. Degradation in performance of Athena

Merge smaller files to larger files in S3

1. **Standard** worker type for AWS Glue takes 2 hours for ETL. Improve performance

Modify job properties, enable job metrics to evaluate DPUs required. Change max capacity parameters to higher number

**job bookmark** just maintains state information and prevents AWS Glue from reprocessing old data

spark.yarn.executor.memoryOverhead is for Vm allocation

1. IOT gives 1TB data to Redshift daily. Hot data used frequestly. Cost effective solution

**Delete** data older than 12 months. Older data in S3 where it can be joined with Redshift using Redshift Spectrum.

**UNLOAD** not enough, you need to delete data from Redshift

**CTAS** to move data from Redshift to S3 is not possible. UNLOAD command needed

1. High availability of HBASE data

EMR clusters in 2 different availability zones. Read replica in the other zone.

Point the two clusters to the same S3 bucket and hbase.rootdir location

1. Hybrid Cloud. Secure QuickSight access with on-prem AD

Amazon QuickSight Enterprise Edition. Set up an **AD Connector** and **SSO** in the on-premises data center.

Standard Edition doesn't support Active Directory integration

1. CSV data in S3. Glue writes to Redshift. Avoid duplicate rows in Redshift

**Staging table** in the AWS Glue job. DynamicFrameWriter class in AWS Glue to **replace** the existing rows in the Redshift table before persisting the new data.

Load data into a staging table -> join with final table to update records -> insert new records in final table.

**ResolveChoice** built-in transform in AWS Glue to avoid duplicate records is **incorrect** because ResolveChoice is just a built-in transform function that just resolves a choice type within a DynamicFrame.

Don’t use pandas dropDuplicates()

1. Click stream analytics setup. Sluggish Elasticsearch queries. **JVMMemoryPressure** errors **when trying to write index** from Firehose sometimes. (Notice emphasis on index)

**decreasing** the number of shards of the Amazon Elasticsearch **index**

don’t look at data and master nodes as the error is from index

JVMMemoryPressure error indicates that there are too many shards

1. Firehose dumps data to S3. Glue job runs every 4 hours. **Stale data**.

S3:ObjectCreated:\* **event trigger** and an AWS Lambda function that invokes the AWS Glue crawler

Change from time based scheduler to event trigger to process as soon as possible

1. Deliver streaming data to some data store. Use SQL to **modify**, aggregate, complex analytic queries, publish to dashboard. Best design.

Firehose dumps to Redshift. QuickSight queries Redshift.

Kinesis streams can’t directly write to Redshift or S3. Needs Firehose to sinking.

Modifying data in Athena CTAS a new table without modifying source data

# Practice Test 2

1. Simulation data in S3 bucket. Query using Hive on EMR. Suggish queries. HDFS at 10% utilization

Set up **instance group configurations** for both the core and task nodes of the Amazon EMR cluster. Configure an automatic scaling policy to scale-out or scale-in the instance groups depending on the threshold value for the Amazon CloudWatch **YARNMemoryAvailablePercentage** metric

**CapacityRemainingGB** is the wrong parameter. Amount of remaining disk space in HDFS which is at 10%.

**Instance fleets** does not have auto scaling feature. Only **group** has this feature.

1. AWS Direct connect to migrate on-prem 5TB warehouse to S3. Curation of S3 data that will be used by SageMaker

DMS to migrate as-is data to S3. Glue for curation.

AWS DataSync + Spark on EMR is time consuming to develop. Although answer is correct.

No need of SnowBall Edge since direct connect already exists.

1. Firehose to process, store, and load data. Kinesis stream provides real-time data to application/software. Which service can load streaming data.

Elasticsearch.

S3 select is feature of S3

Redshift Spectrum is feature of Redshift

Athena does not work on streaming data

1. Ingest streaming data from medical IOT. Strip confidential/sensitive data and store only non-confidential data.

Firehose to dump to S3. Firehose invokes lambda so that only non-confidential data gets stored and sensitive data is stripped during transformation.

Lambda functions should not be scheduled in most cases. They should be invoked on events as this is a great feature.

Kinesis streams cannot write directly to S3.

1. Athena Workgroups to isolate query executions. Query exceeding threshold should be cancelled.

Set the data limit per query under the **per-query** data usage control for each workgroup

**per-workgroup** limit: limit is the aggregation of all queries in the group. This does not cancel all queries in the workgroup but sends SNS alarm

1. Redshift as DWH with data in fact and dimension tables. Customer IP common column to both tables. Sort data on timestamp column. Join required to retrieve data. Best query performance.

DISTSTYLE KEY distribution for both dimension and fact tables. Customer IP Addr as DISTKEY for both tables. Forget about sort.

DISTSTYLE AUTO -> Redhisft chooses what it thinks is best, ALL -> table is broadcast during join (good for small tables), EVEN -> round robin, KEY -> based on one column (multiple columns not possible)

1. EMR with root device volume encryption on all nodes. Use CloudFormation template

CloudFormation template with custom AMI having encrypted root device volume under the **CustomAmild** property

**CustomAmild** -> Custom Ami ld

1. Row Level Security (RLS) on datasets aggregated by QuickSight (select 2)

User/group only see rows that match field values in the datasets (inclusion)

If no rule added for user/group. They can’t see any data (least privilege).

1. S3 with data stored in intelligent tiering has following requirement

ODBC connectivity

Metadata

ETL – pyspark and scala

(Choose 3)

ODBC – Athena, Metadata – Glue Data Catalog, ETL – Glue

1. MySQL RDS to store 3 months of data. Older data to be moved out but still required for quarterly reports. Best solution.

Schedule daily export from MySQL RDS to S3. Load 1 year data to Redshift and previous to S3. Use Redshift spectrum

Athena cannot query Redshift, only S3. So direct data in Redshift will need to be unloaded to S3.

Automated snapshot are stored in S3 owned by Amazon. To run Athena on that you will need to export data to your S3 first. Manual snapshot is stored in self owned bucket and works fine directly

**Sync a year’s worth of data on an Amazon RDS read replica** – cannot be done since you can’t subset a read replica (to only 1 year)

1. Kinesis Data Stream (web page as partition key - shard) -> **PutRecord** API is used to send data into stream -> multiple **ProvisionedThroughputExceededException** errors

Increase number of shards

Capacity limit defined by number of shards in the streams

Do not decrease number of shard as this will reduce the capacity

1. Proprietary ETL on EC2 replaced with Glue and S3. Data loaded to Redshift cluster. Save redshift query results to external storage. Best design.

Move the Redshift query results to an Amazon S3 bucket using the **UNLOAD** command.

**COPY** command can’t copy to external table

**COPY** from Redshift to S3 does not work. It is the opposite direction command. S3 to Redshift

**UNLOAD** command does not support Redshift spectrum

1. Daily CSV in S3 of 25GB. Athena on 6 months of data. Then infrequently accessed. 2+ older is not used for processing but needs to be archived. Avg query scan is 200MB under 1 minute response time. Best optimization

ETL CSV data into columnar format. 6 months after creation move to S3 standard-IA. Another policy moves to glacier after 2 years.

1. Data in N. Virginia in Redshift. QuickSight in Singapore. Performance issues.

Security group in N. Virginia with inbound rules for IP range for QuickSight in Singapore

Cross- region snapshot in Singapore increases the cost of Redshift usage

1. Gzipped csv 150MB 25K records on S3 glacier. Need only the first 10 columns for analysis.

Restore archive to S3 and use S3 select

Querying glacier is only possible with uncompressed file

Athena is not required since S3 select is cheaper

Redshift spectrum can only query uncompressed data in S3

1. Join data in S3 with Redshift table

Redshift spectrum

1. Daily 1 TB data in single csv file in S3 partitioned by date. Loading to Redshift takes time.

Split to smaller chunks and **COPY** command to Redshift. COPY command does parallel processing (MPP)

**INSERT** command only used to move data from one table to another

Streaming using Firehose is correct but costlier

1. Kinesis Data streams to ingest data. Throttling write requests and write performance is reduced on streams. (2 options)

**UpdateShardCount** API in Amazon Kinesis to increase the number of shards

random partition keys and adjust accordingly to distribute the hash key space evenly

across shards

incorrect: Enabling monitoring does not slow down write performance. It just helps in identifying hot shards in your data stream.

1. Firehose streams 1 minute data to dashboard (near real time). Implementation

Firehose -> Elasticsearch -> Kibana

Incorrect: Firehose -> redshift -> QuickSight SPICE. This is not near real time

1. 300TB Redshift DWH that gets data every 6 hours. Read queries during other time. 2 hour long running query fired every hour and causes smaller queries to be queued.

On the Amazon Redshift console, create a parameter group associated with the cluster and configure scaling for the workload management (WLM) queue.

WLM can create multiple queues and route queries to appropriate queue at runtime

Incorrect: adding more nodes will increase cost

Elastic resize makes the cluster unavailable for sometime

1. Data in S3 partitioned by customer name. separate COPY command for each customer file to Redshift takes long time.

List the data file locations in a single manifest file, then issue the COPY command with the manifest file to load all the data into Amazon Redshift.

EMR cluster will incur additional cost to do something that a simple manifest file can achive

AWS Glue Data Catalog to populate the metadata of all the files in the S3 bucket and issue a COPY command to load the data into Amazon Redshift (correct but costly)

1. 3 node denser cluster in Redshift to store 5TB of data. Additional 3TB needs to be merged. How to adjust cluster for no loss in performance

Modify the cluster to use **dense compute nodes** and scale it using **elastic resize** to increase performance.

Use elastic resize whenever possible

1. The application will collect, process, and analyze clickstream data from various websites in real-time. Which of the following is the most suitable service to use for the application?

Amazon Kinesis (only singular solution that does all above requirements)

1. Company wants ML algorithm to detect anomalies. Out of the box solution.

Use Amazon QuickSight to spot and visualize anomalies using its ML-powered anomaly detection feature.

QuickSight provides three major features 1. ML-powered anomaly detection, 2. ML-powered forecasting, 3. Autonarratives

An 'autonarrative' is just a natural-language summary widget that displays descriptive text instead of charts

1. Multiple teams having personal EMR clusters and metadata for running queries in Hive. Need to create a centralized metadata layer that exposes S3 objects as tables.

Configure an external metastore for Hive

Enable EMRFS consistent view just tracks the consistency of S3 objects

1. Multiple EC2 instances. Want centralized logging for real time monitoring and action. efficiently set up a data logging system within AWS?

Steam data to CloudWatch using CloudWatch Agent. CloudWatch logs subscription filter -> firehose -> Elasticsearch

Splunk is a proprietary logging system that you have to set up and manage

1. Large files consolidated into single compressed CSV of 100GB rested in S3. COPY to Redshift. Make processes faster.

Split large file into multiple of Redshift slices. Compress and rest in S3. COPY command on files

number of files must be a multiple of the **number of slices** in the Amazon Redshift cluster, and **not the number of its compute nodes**.

1. IOT sends 5 second interval data from location. Alert when threshold in data crossed. Fully managed solution required for when new locations are added.

IOT -> Kinesis streams -> Kinesis analytics to detect threshold breach -> lambda to poll results and send alarm to SNS. Configure autoscaling on Kinesis Data Streams.

EMR is not fully managed service

RDS is not fully managed service

Firehose has minimum batch of 60 seconds. Firehose -> Lambda is used for data transformation and not analysis and detection.

1. S3 to store data queried from Athena. Make this scalable. (3 options)

Build Lake in same region as Athena

CSV to Parquet

Compress S3 objects

1. Athena and QuickSight used to analyze data in S3. New data in S3 -> Athena as a source to QucikSight failed. Solve.

Configure the permissions for the new S3 bucket from the Amazon QuickSight console

To successfully connect Amazon QuickSight to the Amazon S3 buckets used by Athena, make sure that you authorized Amazon QuickSight to access the S3 account. It’s not enough that you, the user, are authorized. Amazon QuickSight must be authorized separately.

1. Transfer 5TB of log data from S3 to HDFS of EMR. 1000s of files of varying size. Most efficient solution

Use **S3DistCP** tool to transfer the data from the S3 bucket with the existing Amazon EMR Cluster.

DistCP is incorrect because its AWS version is **S3DistCP**

**Import/Export** is a facility between storages and not AWS services

1. Redshift dense cluster is found to be non PCI DSS compliant. Security compliance requires encryption through hardware security module (HSM) and automatic key rotation. Best design (2 options)

New redshift cluster with HSM. Migrated data to that cluster

Client and server certificate with automatic key rotation between redshift and HSM.

Incorrect: HSM can’t be enabled by modifying cluster. Only KMS can be done that way

1. Trading data from 6 sources. Notification when high volume of buy orders. Data to be analyzed and filtered using Kinesis SQL queries and archived in S3. AWS managed service solution

Kinesis Data Stream -> Kinesis Analytics -> Lambda -> Notification (SNS) -> Firehose -> S3

Incorrect: Do not use 6 data streams. 1 data stream can take data from 6 sources.

1. Upgrade online banking system with real time data collection. 25KB is the record size.

Configure Amazon Kinesis Data Streams and the banking system to use the **PutRecordAPI** to send data to the stream. Register consumers with the enhanced fan-out feature.

the enhanced fan-out feature only available in Kinesis streams.

PutRecordBatch API only available for Firehose

EC2 not required

1. Firehose stores JSON data (100 unique devices) in S3 with Athena for analysis daily. 2 options to tune storage to reduce costs.

Parquet with snappy compression

In Athena, create the external table and partition it by the device and date

Not partition by year, month, day, hour since the question mentioned 100 unique devices

1. Solution based on EC2 instances and DynamoDB. Logs pushed to CloudWatch logs. Enhance the log data with data from DynamoDB in near real-time

Lambda function to enrich data **inside** DynamoDB. Firehose subscribers to Cloudwatch logs and pushes to S3. CloudWatch Logs subscription that sends log events to delivery stream **(not logical)**

Question mentions near real time so hourly schedule is wrong.

Installing Kinesis agent to the EC2 instance is unwarranted since there is already a CloudWatch Logs integration that can deliver the logs. Creating a Kinesis Data Analytics SQL application is also unnecessary and quite costly.

1. 100s of applications on EC2 instances. Doesn’t want to manage solution. 1. Collect transform logs in JSON format 2. Handle delivery failures 3. Analyze visualize log data

Firehose to ingest log data -> lambda for format conversion -> Elasticsearch service -> failed delivers S3 bucket

QuickSight is wrong.

Kinesis Data streams is wrong because it does not directly integrated with Elasticsearch.

1. Redshift: Which of the following is NOT a best practice when choosing the best distribution style?

Select the smallest dimension based on the filtered dataset’s size

Best practices are

1. Designate a common column for the fact table and the dimension table
2. Use a DISTSTYLE ALL distribution for tables that are not frequently updated
3. Select a DISTKEY with high cardinality
4. Centralized data catalog to be populated on a scheduled basis.

Set up an AWS Glue crawler schedule to populate the data catalog

Apache Hive metastore slightly correct but is a lot of work.

1. Data in S3 in parquet format. Solution with best **visuals**. Least operational effort

Use Amazon Athena for the data source. Pair it with Amazon Quicksight and display data with geospatial charts (**best visuals**)

S3 -> QuikSight not possible

Pivot tables, wrong. Best visuals.

1. Data analyst. DWH on Redshift , Lake on S3. Move infrequently accessed data but it should be queryable with 30 seconds.

S3 Standard –IA with Athena for SQL

S3 Glacier select – access time is 1 – 5 minutes

S3 Select does not suitable for analytical workloads.

1. Data in CSV. Every user has his own access to QuickSight dashboard. Confidentiality – user should only see own data.

On the Amazon QuickSight web console, create dataset rules with row-level security to restrict access on each dashboard.

QucikSight - row-level security is configured in the **dataset, not in a manifest file**

1. 1 TB uncompressed data generated daily. 6 months data used for analysis. 365 days data SQL job job does customer trends and popular products. dc2.8xlarge redshift cluster with 40 nodes. Most cost effective solution.

6 months on redshift. Older to S3 in parquet format. Weekly job using redshift spectrum.

1. Hive batch scripts runs on S3 at 4PM and takes 1-2 hours. 3 node cluster. Cost effective solution.

Schedule an Amazon **CloudWatch Events** rule to invoke a Lambda function to run at 4:00 PM daily. Configure your AWS **Lambda function to provision an Amazon EMR cluster with a Hive** execution step. On the RunJobFlow API, set the KeepJobFlowAliveWhenNoSteps to FALSE and disable the termination protection flag.

Spot instances are not recommended as they are based on availability

Glue does not support Hive

Lambda does not have Hive runtime. It is only present in EMR and hence the above answer

1. Company putting advertising videos on social media. Continuous monitoring for sentiments and video playback issues. Latency < 30 seconds. JSON format. Best collecting and processing methods.

Kinesis Data Stream -> Kinesis Analytics -> Firehose -> S3.

KDA does not send data directly to S3. It needs Firehose

KDA doesn’t support DynamoDB. It supports Firehose, lambda, Kinesis data streams

1. Valid destinations for Firehose (choose 3)

S3, Redshift, Elasticsearch, Splunk

1. EMR cluster. S3DistCP tool used to concatenate several parquet file. But there is error. What is the workaround.

PySpark to concatenate the Parquet files

**S3DistCp does not support concatenation for Parquet files**. When you attempt to concatenate them, you get an error message

1. 10 years data in S3. Need to analyze all this data. Optimal query performance. Find out where the highest crime hour of the day occurs

Use Apache ORC. Partition by date and sort by NCIC code.

Incorrect – compressed CSV

Incorrect – nested JSON

Use Apache Parquet. Partition by NCIC code and sort by date is incorrect – partition should be by date.

1. Kinesis Data Streams -> KDA using windowing queries. Analyst wants to aggregate records falling in the same time window. **Data arrives at inconsistent intervals**. Which type of windowed query should be used?

Stagger Windows - allow for multiple overlapping windows. suited for data that arrive at inconsistent times.

Tumbling Windows: distinct time-based windows that open and close at regular intervals.

Sliding Windows: using a fixed time or rowcount interval.

Stream Joins is incorrect because it is not a type of windowed query

1. 200TB of data. Grows by 50TB every month. IT uses data of current month. Accounting of last month and previous 12 months. Management of 30 days.

2 months in Redshift. Older in S3. Redshift spectrum as data source for QuickSight.

Persistent EMR cluster is very costly

All data in S3 is incorrect as high performance required for latest data

1. ML algorithms to visualize and analyze. Analyst team has little experience with random Cut Forest (RCF) algorithm and wants a managed solution.

Use Amazon QuickSight for both forecasting and data visualization. Utilize the built-in ML-powered forecasting in QuickSight to generate the geological forecast

AWS Glue ML transform has no out-of-the-box support for the RFC algorithm.

1. Data in S3. Glue used as ETL and load to MySQL RDS. ETL uses **DynamicFrame** and analyzes complete data in S3. This is suboptimal as data size grows.

Save the state of the previous ETL using job bookmarks. Start the subsequent AWS Glue jobs based on the bookmark.

Explanation of bookmark given in an above answer.

1. High performing long term storage for historic data. 25TB uncompressed. Single row inserts every 30 min (low volume). 1000s of aggregation queries daily (high volume). Multiple complex joins. Subset of columns used most often. Most suitable storage service.

Load the historical weather data on Amazon Redshift.

Other options are Aurora MySQL, Elasticsearch, Neptune (graph database) to this question.

1. Real time data from 150 websites to KDS 40 shards -> Kinesis Consumer Library (KCL) on autoscaling EC2 instances -> DynamoDB. Avg. CPU on EC2 is 20% at peak times. DynamoDB has provisioned write capacity set to 5. Increased latency at peak times. No errors in KCL logs and CPU utilization with limits. Solve for latency.

Increase the write throughput of the DynamoDB table

Since there are no errors in KCL logs and CPU utilization is below threshold, KCL and EC2 are ruled out.

No errors means no issues with KDS shards too.

1. Food delivery company. Redshift. 1. **trips fact** table 2. **riders dimension** 3. **customer fact** table. Riders data changes slowly. Customer data changes frequently. Best design for optimal query performance.

Designate a **DISTSTYLE KEY (destination)** distribution for the **Trips** table and sort by delivery time. Use **DISTSTYLE ALL** for the **Riders** table. Use **DISTSTYLE EVEN** for the **Customers** table.

DISTSTYLE ALL for dimension table. Eliminates 2 options.

Never DISTSTYLE ALL for fact tables.

DISTSTYLE EVEN to the customers table because it is more suitable for frequently changing data

1. Athena in the **ap-southeast-1** to query S3 buckets in ap-southeast-1 and ap-northeast-2 regions at lowest cost.

Run the **AWS Glue crawler in ap-southeast-1** to catalog datasets in all Regions. Execute the Athena queries in ap-southeast-1 once the data has been crawled.

Incorrect: Modify the AWS Glue resource policies… 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.

1. Common Hive running on EMR. 3 scientists have their separate S3 buckets. On-prem AD and Kerberos used for authentication. security requirements, the access to S3 bucket must be limited to the members of the group who owned the bucket.

**Create a service role** for the EMR cluster that grants no access to Amazon S3. Create three IAM roles for each group and edit their permissions to ensure that only the group that owned the S3 bucket can access it. **Add the service role** for the EMR cluster EC2 instances to the trust policies of the three IAM roles. Configure EMRFS security configuration for the three IAM roles that will be assumed by the groups from the Active Directory.

EMR service role should not have full access to S3 – elimination

Incorrect: additional IAM roles to the cluster's EMR role…

1. Encrypted S3 bucket contains PII information processed by EMR. Data must not be publicly acceptable and EMR should not be exposed to the public internet.

**From** the Amazon EMR Console, enable the 'block public access' setting to prevent any user from creating a cluster that is accessible from the public Internet.

1. Data stored in S3 One Zone-IA. Processed using EMR using EMRFS with consistent view enabled. Performance of the cluster drops as concurrent queries and analytics jobs increases. How to increase performance of cluster in reading S3 objects.

When storing the data points on the S3 bucket, add a sequential date-based naming as a prefix for the object filename.

**Prefix increases parallelism** while reading from S3

Question is for S3 and not EMR – elimination

S3 One Zone-IA and S3 Standard has the same read performance. So there is no gain in changing the S3 bucket type.

1. Social media activity captured by KDS (partitioned by user name). posts are validated sequentially before transferring to Elasticsearch cluster. Posts take 30 minutes during peak hours to reach ElasticSearch. Reduce latency.

Use **multiple AWS Lambda** functions to process the Kinesis data stream using the **Parallelization Factor** feature.

1. Faster way to resize Redshift cluster

Elastic resize to manually add or remove nodes.

Elastic resize – 10 to 15 minutes. Classic resize 2 hours to 2 days. Only use if elastic is unavailable for the Redshift configuration.

Auto resize feature does not exist

Redshift does support online resize but it takes several hours to complete.

1. S3 -> Athena. Sensitive data. query history and execution must be separated among different users and teams for compliance purposes

Set up an **Athena workgroup** for each team and apply tags to each workgroup. Using these tags, grant appropriate permissions to the workgroup with IAM policies. Have the members use their assigned Athena workgroup.

1. present the data in near real-time for reporting purposes, for dashboards.

Firehose -> KDA -> KDS -> lambda -> DynamoDB

Buffer interval for zero is not possible.

1. Redshift cluster with PII information. Query using QlickSight. Logs must contain connections, disconnections, dB authentication attempts. Also determine who ran which query.

Using the Amazon Management Console, enable audit logging for the Amazon Redshift cluster.

AWS Artifact offers reports (e.g. ISO and PCI) that provide best practices in achieving compliance. Nothing to do with logging. Config only helps manage your configuration settings.

CloudWatch Alarm is primarily used to monitor the performance of the cluster. Not logging.

VPC flow logs just collect IP traffic going to and from network interfaces in the VPC

1. encrypt the EMR cluster’s root volumes

Enable at-rest encryption for local disks and recreate the cluster

EMRFS is a file system and not a root volume

in-transit encryption is not needed in the scenario

By default, all security configurations are disabled

# Practice Test 3

1. DynamoDB table. custom classifier to an AWS Glue crawler to extract data from the database. After running the crawler, AWS Glue returns a classification string of **UNKNOWN**. Most likely reason for the classification

Glue was unable to find a classifier with certainty greater than 0.0

Classifier generates data and generates scheme if it is sure. This is the CERTAINTY number. At 1 it means it can create correct schema. At 0 the classification string is UNKNOWN. The custom classifier did not work.

1. ALB distributes load to EC2 instances. ALB logs are in S3 bucket where they must be joined with other data. JDBC driver for the proprietary BI tool.

Athena to run queries in S3 and connect to BI tool

No need for Redshift

EMR is too huge an effort

1. JDBC connectivity. Datasets stored in Aurora PostgreSQL, Elasticsearch, and S3 (ORC format). Join data using SQL with JDBC.

EMR cluster with **Apache Presto** to query all the datasets in place. Within minutes.

DMS does not support Elasticsearch. It supports Aurora only

Glue ETL job to join data from Elasticsearch with Aurora and S3 is lot of setup.

Spark SQL job in Glue to query all datasets in place does not give interactive querying capability.

1. Persistent EMR stores data as external tables in S3. Transient EMR clusters access this same data in S3. Metdata data about S3 present in the persistent cluster. most efficient way to expose the Hive metastore

Configure Hive to use the AWS Glue Data Catalog as its metastore

DynamoDB metastore incorrect because external metastore only supports AWS Glue Data Catalog and External MySQL Database or Amazon Aurora. But both require lot of effort.

1. KDS and EMR for processing and ingesting data. Large amount of small data from several IOTs. Data rests in S3. Data in S3 queried once a day using PySpark in EMR. Redesign to reduce cost, increase efficiency and use PySpark.

Replace EMR with Glue. Glue ETL in python to merge small data and write parquet to S3.

PySpark does not work with Lambda

1. ETL process that loads data from various data sources into an Amazon Redshift cluster. COPY command used to load data into Redshift. Increase efficiency. Which additional step should be implemented as part of the new ETL process

Implement a temporary staging table to hold the data for transformation

UNLOAD moves data out of Redshift.

**S3DistCp** tool is primarily used in Amazon **EMR**, not in Redshift, to load data from Amazon S3 to HDFS

# Practice Test 4

1. Data in S3. Modify ETL to load data into Redshift 2 node, 2 slice per node. (2 options)

single COPY command with the **manifest file** to load data from several files

simultaneously

Use temporary staging tables during the loading process.

1. 1000s of small CSV in S3. Load to redshift. 6 node, 4 slice per node.

Transform the data from CSV files to **24 large files in Apache Parquet** format using AWS **Glue**. Run the **COPY** command to load the file into the Redshift cluster. To minimize costs, query the files with Amazon Athena from the Amazon S3 bucket

Single large file – eliminated. 6 \*4 = 24. Hence size is given.

Parquet better than Avro. Glue better than EMR for simpler work.

1. Athena to query S3. If threshold of data scanned by all queries is breached, email alert/notification to manager.

**Athena Workgroup**. For each workgroup used, configure the workgroup data usage control limit to the prescribed threshold and send a notification via an SNS topic.

1. Huge amounts of data loaded into Redshift. COPY took normal time but VACUUM took 3 hours longer. Analyst that schema change was made to the table. 2 options for latency increase.

The table has ten more additional columns than the previous run

The source data was not loaded in sort key order

Following are good things – 1. After the load, the table has a very low percentage of unsorted data. – 2. The VACUUM operation is run too frequently. – 3. The VACUUM operation was run with BOOST option.

1. KDS -> Java consumer app. Data written to proper shard, partitioned on userIDs and in order. But data did not arrive in order (sequence). Analyst that every time the stream was resharded, the out-of-order records from the same user ID arrived from different shards. Cause and fix.

The Kinesis consumer is not processing the parent shard completely before processing the child shards after the stream has been resharded. The parent shard must be processed completely first before processing the rest of the child shards

You should always read data from the parent shard before exhausting it and moving to the child shard.

1. Catalog the data on a data storage as **key-value pairs** that require **immediate access**. Ability to read, write, and manage **petabytes** of data using a SQL-like interface. A solution with low operational overhead is preferred

Analyze data with Apache Hive on Amazon EMR. Save the data to an Amazon DynamoDB table.

DynamoDB better than S3 for key-value pair. Hive to SQL like query. EMR for processing.

# Practice Test 5

1. Textract used on millions of JPEGs forms (S3) to extract customer data. JPEGs should be downloadable. Performance over cost.

Elasticsearch Service to index the metadata from the scanned forms and the path of the image file in Amazon S3. Search and visualize information using Kibana.

Redshift to store metadata and path of file in S3. Partially correct, but can’t do visual insights, unlike Kibana.

Parquet metadata and file in S3. Data Catalog to index information. Athena to query. Just like above, partially correct but can’t do visual insights.

1. IOT gives temperature and humility data. Visual Report every morning. Aggregate sensor data for generating reports. Cost-effective solution.

Create a transient cluster in Amazon EMR to aggregate the sensors data each night and use Amazon Quicksight to generate a report each morning.

1. WAF sends logs to Firehose to transform data rested in S3. Infrequent log analysis and **visualization**. Cost effective solution.

Run an AWS Glue crawler that connects to the Amazon S3 bucket and write tables in the AWS Glue Data Catalog. Perform ad-hoc analysis using Amazon **Athena** and save the query results in a separate S3 bucket. Use Amazon **QuickSight** to create data visualizations.

Athena is cheaper than Elasticsearch and hence above is a better option that Elasticsearch and Kibana pair.

1. Compressed files stored in S3. Every 4 hours EC2 using cron processes this data for BI. Speed up analysis workflow and refreshing the dashboard with aggregated data in minutes instead. Moreover, they want to include a **search feature** that gets refreshed with more options (2 options).

Use Kibana as a visualization tool that processes the data from the Amazon Elasticsearch

Service and set the refresh interval at the desired value.

Write an AWS Lambda function that sends that data to Amazon Elasticsearch Service

directly from Amazon S3 at the desired schedule interval.

Rule out QuickSight and custom built consumer

Firehose’s near real time requirement is not given. So don’t use and save on costs.

1. Configure QuickSight. Access to dashboard authenticated using Microsoft AD. Data encrypted in transit and rest.

Amazon QuickSight **Enterprise edition**. Set up an identity federation using SAML 2.0 and use the default encryption settings

**Enterprise edition** supports both AWS Directory Service for Microsoft Active Directory and Active Directory Connector

QuickSight Enterprise edition, the data at rest in SPICE is encrypted using block-level encryption with AWS-managed keys. You cannot use customer-provided keys that are imported into AWS KMS

1. S3 with QuickSight. The report must display all the records, including **recent data**. Suitable solution.

Select the dataset and create a daily schedule refresh in the dataset settings

Recent data hence daily.

# Practice Test 6

1. All S3 buckets encrypted with auditable access trails. EMR cluster with EMRFS to be used to process and transform data. Configuration that will allow cluster to access encrypted data.

Modify the cluster’s security configuration by delegating the appropriate CMKs for each bucket under the per bucket encryption overrides

SSE-S3 does not provide audit trails

You're not allowed to export a CMK that was created using AWS KMS

1. On-prem 80GB compressed data send daily to S3. Copy command to Redshift 4 slices. Optimal performance.

Split the file into 80 parts, with each part having a size of 1 GB.

Optimal size of compressed file is 1MB to 1GB. All files should be of same sizes. This eliminates answers with number of files are not multiple of 4, different sizes, can be cut down further.

1. most cost-effective solution for anomaly detection

Use the **RANDOM\_CUT\_FOREST** function in the Kinesis Data Analytics application to detect the anomaly and send an alert to open the valve

1. The following requirements must be enforced when defining tables in AWS Glue Data Catalog: - The catalog table should support custom naming conventions. - The catalog table must be updated with new partitions delivered into corresponding Amazon S3 bucket prefixes. Which approach fits the requirement while having the LEAST amount of configuration overhead?

Create a catalog table using the **CreateTable** API operation. Write an AWS Glue ETL script to update table partitions.

CreateCrawler API 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.

1. History of EMR events to be logged and encrypted (3 options)

Create the new EMR cluster while enabling logging and log encryption with AWS KMS

customer-managed key.

Create an Amazon S3 bucket with the proper permissions.

Configure the Amazon **EC2 instance profile and** the Amazon **EMR role** with the

correct **kms** permissions.

**EC2 instance profile** and the Amazon **S3 role** with the correct kms permissions is incorrect because it misses out on EMR permissions.

EMR logging and debugging is enabled when you launch a cluster

1. The data that passes through is stored in an Amazon DynamoDB table and is encrypted using DynamoDB's encryption client with AWS KMS managed keys before it gets written into the table. An Amazon Redshift cluster is also used as the data warehouse and is currently used by various departments. The data analyst team needs to build a loading workflow and process them without compromising the sensitive data. What should they do to achieve this task?

Enable **DynamoDB streams**. Write an AWS Lambda function to transfer the sensitive data to a secured S3 bucket. Create a table in the Amazon Redshift Cluster with access granted to

users approved to access the purchases data only. Use the COPY command to load the data from Amazon S3 to the Redshift table using the IAM role with access to the KMS key.

1. Two weather stations. A has 20 sensors and B has 10 sensors. Partition key is weather station name. KDS from weather station A has bottleneck due to more data pushed. Solution.

Assign the sensor ID as the partition key instead of the station name