Collection

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

INGESTION DECISIONS

- **Streams** — Replay and consumers
- **Firehose** — Buffered delivery
- **MSK** — Kafka compatibility
- **Keys** — Balance and order

01 Collection 18%

Choose the ingestion path

- **Kinesis Data Streams** retains an ordered, partitioned event log for replay and independent consumers. Consumers perform processing and delivery; creating a stream alone does not load S3 or Redshift.
- **Amazon Data Firehose** buffers and delivers records to supported destinations with managed scaling. Its Redshift delivery path stages data in S3 and uses **COPY**; buffering adds latency.
- **Amazon MSK** provides managed Apache Kafka for Kafka-compatible producers, topics and consumers. Size partitions, brokers and message limits for the workload; managed infrastructure still needs capacity planning.
- **CloudWatch Logs subscriptions** feed incoming logs to services such as Kinesis Data Streams, Firehose or Lambda. The CloudWatch agent collects application logs; EC2 detailed monitoring supplies metrics.
- **AWS DMS** supports database migration and change data capture. **AWS Glue JDBC** supports scheduled extracts and transformations. **Amazon AppFlow** supplies managed integrations with supported SaaS sources such as Salesforce.

Producers, partition keys and throughput

- A **partition key** maps records to a shard. Spread traffic across enough distinct keys while preserving required entity ordering; sensor IDs can distribute load more evenly than a few station names.
- A **hot shard** can throttle while aggregate stream capacity remains unused. Inspect per-shard load and key skew before adding shards; increasing capacity alone cannot divide one hot key.
- **Kinesis Producer Library (KPL)** aggregates small user records and buffers requests for efficiency. Buffering trades latency for throughput; wait for acknowledgments and handle failures before shutting down a producer.
- **PutRecords** batches writes through the SDK. Inspect individual record results and retry failed records with backoff. Retrying a whole partially successful batch can create duplicates and affect ordering.
- Check **record size**, write rate and request quotas before choosing an ingestion service. For oversized payloads, consider splitting records or storing objects in S3 and streaming references.

Collection / Storage & Data Management

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

INGESTION DECISIONS

- **Streams** — Replay and consumers
- **Firehose** — Buffered delivery
- **MSK** — Kafka compatibility
- **Keys** — Balance and order

Consumers, replay and delivery guarantees

- **Kinesis Client Library (KCL)** coordinates shard leases and checkpoints using DynamoDB. Monitor its metadata table for throttling as workers and shards grow; distinguish this from stream read throttling.
- **Enhanced fan-out** gives registered consumers dedicated per-shard read throughput and push delivery over HTTP/2. It reduces contention between consumers; it does not increase producer write capacity.
- Ordering is scoped to a **shard**, with partition keys grouping related records. After resharding, finish parent shards before processing their children; KCL manages these dependencies.
- Producer retries and consumer recovery can both replay records. Use stable **event IDs**, deduplication and idempotent destination writes for end-to-end correctness; checkpoints alone do not prevent duplicate side effects.
- Monitor **IteratorAgeMilliseconds**, write/read throttling, errors and downstream processing time. Rising iterator age indicates consumer lag; scale the constrained stage and preserve ordering requirements.
- Set **retention** to cover replay and outage recovery. Keep a separate S3 archive for durable history; longer retention does not repair an overloaded producer or consumer.

02 Storage and Data Management 22%

File formats and physical layout

- **Parquet** and **ORC** are columnar formats that support column pruning, compression and predicate pushdown. **CSV**, JSON and row-oriented Avro serve different interchange needs and often scan more data for column subsets.
- **Partition** by common query filters such as date, campaign or account. Use a useful time granularity; excessive high-cardinality partitions and tiny partitions increase metadata and scheduling overhead.
- **Hive-style paths**, such as year=YYYY/month=MM/day=DD/, encode partition values in directory names. A timestamp embedded only in a file name does not create catalog partitions.
- **Compact small files** to reduce object listing and task startup overhead. Keep enough reasonably sized files for parallel readers; one enormous file can also limit parallelism.
- **Compression** reduces storage and scan I/O. Text gzip files are generally not splittable; columnar row groups support parallel access. Sorting within partitions can improve block skipping for selective filters.

Storage & Data Management

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

A QUERYABLE DATA LAKE

- **S3** — Durable objects
- → **Layout** — Partitions and columns
- → **Catalog** — Schema and locations
- → **Query** — Read relevant data

Catalogs, schemas and freshness

- **AWS Glue Data Catalog** stores database, table, schema, location and partition metadata. Athena, Redshift Spectrum and EMR can share it; cataloging data does not copy or transform the records.
- **Crawlers** discover schemas and partitions; **classifiers** identify data structure. Set schema-change behavior deliberately so discovery does not unexpectedly replace curated metadata.
- For controlled table names, create the **catalog table** explicitly and configure discovery around its location. Use APIs, crawlers or ETL catalog updates to register new partitions.
- **MSCK REPAIR TABLE** discovers compatible Hive-style partitions. **ALTER TABLE ADD PARTITION** registers explicit locations. Partition projection can avoid catalog enumeration when partition values follow predictable rules.
- Object freshness and **metadata freshness** are separate. New files in an existing partition can be visible without a recrawl; new partitions or schema changes need corresponding metadata updates.
- Point each table at the **correct S3 prefix**, separate unrelated datasets and prune partitions in queries. Catalogs are regional; align metadata access, bucket access and cross-Region transfer costs.

Lifecycle, availability and storage tiers

- **S3 Standard** suits frequent access. **Standard-IA** offers immediate retrieval with retrieval charges and minimum billing considerations. **One Zone-IA** stores data in one Availability Zone and suits recreatable data.
- **Glacier Flexible Retrieval** and **Deep Archive** require restoration before normal analytical reads. **Glacier Instant Retrieval** provides immediate access; choose the exact storage class against latency and access frequency.
- Use **S3 Lifecycle** rules for object-age transitions and expiration. Retain raw source data separately from curated data when reprocessing or compliance requires different retention periods.
- **Intelligent-Tiering** responds to changing access patterns. It differs from a lifecycle rule based on days since object creation; evaluate monitoring charges and optional archive tiers.
- **Versioning**, replication and protected backups address different failure cases. S3 read and list operations provide strong consistency; catalog refresh and downstream caches still have their own delays.

Storage & Data Management / Processing

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

RELIABLE DATA PROCESSING

Ingest Track input → **Transform** Clean and enrich → **Write** Idempotent output → **Operate** Monitor and retry

Match the data store to access patterns

- **Redshift** is a columnar, massively parallel warehouse for repeated aggregations and complex joins. Keep frequently queried data local and expose less-used S3 history through **Redshift Spectrum**.
- **Spectrum external tables** query files in S3 and can join warehouse tables. Use partitioned columnar files in a queryable storage tier; warehouse capacity does not eliminate external scan costs.
- **DynamoDB** suits predictable key-based reads and writes, including nested document attributes. Single-digit millisecond access depends on a suitable key design and sufficient throughput; relational joins need another approach.
- **OpenSearch** indexes text and structured fields for interactive search and log analytics. Store searchable document metadata and S3 object references in the index while retaining original files in S3.
- **EMRFS** accesses durable S3 data; cluster **HDFS** depends on cluster nodes. Persist data and the Hive metastore outside transient clusters when both must survive termination.

03 Processing 24%

Glue transformations and incremental ETL

- **AWS Glue** runs managed ETL with Spark-based PySpark or Scala jobs. **DynamicFrames** represent records with flexible schemas; convert to Spark DataFrames when their APIs better suit a transformation.
- **ApplyMapping** maps or renames fields and types. **DropNullFields** removes null-only fields. **ResolveChoice** handles ambiguous field types; it does not select the newest duplicate record.
- **FindMatches** learns record similarity from labeled examples to identify matching entities without a shared identifier. Review and train matches; assigning a canonical entity ID is a downstream design decision.
- **Job bookmarks** track supported source processing state. S3 bookmarks use object modification information; JDBC bookmarks need suitable ordered keys. Commit bookmark state after successful processing.
- Bookmarks do not make the **destination idempotent** or track deletions automatically. Rewinds, retries and backfills need duplicate-safe writes; in-batch dropDuplicates does not remove rows already loaded elsewhere.
- Use **enableUpdateCatalog** and **partitionKeys** where supported to update catalog partitions during ETL. A crawler after successful processing is another option when schema discovery is required.

Processing

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

RELIABLE DATA PROCESSING

Ingest Track input → **Transform** Clean and enrich → **Write** Idempotent output → **Operate** Monitor and retry

Diagnose ETL bottlenecks

- Enable **job metrics**, CloudWatch logs and Spark execution diagnostics before increasing capacity. Compare driver memory, executor memory, shuffle, skew, CPU and I/O to find the constrained resource.
- Many tiny input files can exhaust **driver memory** while executors remain idle. Glue DynamicFrame **groupFiles** with **inPartition** groups supported inputs into fewer tasks; output compaction is a separate step.
- Select **worker type** and worker count for the observed memory and parallelism needs. More workers may not fix driver pressure, skewed partitions or a serial JDBC bottleneck.
- Reduce work early with **partition predicates**, needed-column selection and filtering. Balance Spark partitions; avoid collecting large distributed datasets into the driver.
- Larger **Firehose buffers** can reduce downstream small-file overhead but increase delivery delay. Its JSON-to-Parquet/ORC conversion uses catalog schema; custom CSV conversion or enrichment needs transformation logic.

EMR execution and resilience

- **Amazon EMR** runs frameworks such as Hive, Spark, Presto and HBase. Preserve compatible Hive scripts on EMR when minimizing migration changes matters; Glue jobs do not directly execute arbitrary Hive scripts.
- **Primary nodes** coordinate the cluster, **core nodes** run tasks and store HDFS data, and **task nodes** add compute. Multiple primary nodes improve control-plane availability within the cluster's Availability Zone.
- Use **transient clusters** for periodic batches, external S3 storage and a persistent Glue or RDS Hive metastore. Submit steps, terminate after completion and clean up failed runs.
- **Bootstrap actions** install software at cluster startup. A compatible **custom AMI** can preinstall dependencies and configure root volumes; enforce encryption when volumes are created.
- Scale against the bottleneck: **YARN available memory** and pending work indicate compute pressure; HDFS free space indicates storage pressure. Use interruption-tolerant task capacity for Spot savings.
- **HBase on S3** separates durable storage from compute. Read replicas can support read availability; independent clusters must not both write to the same HBase root directory.
- For S3 **503 Slow Down** responses, use exponential backoff and appropriate EMRFS retry settings. Distribute sustained request pressure across useful prefixes and avoid a large, sudden burst of parallel requests.

Processing

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

RELIABLE DATA PROCESSING

Ingest Track input → **Transform** Clean and enrich → **Write** Idempotent output → **Operate** Monitor and retry

Stateful streaming and enrichment

- **Managed Service for Apache Flink** runs stateful stream processing with windows, joins and checkpoints. Use it for cross-event aggregation or ordered pattern detection rather than treating a Firehose transform as persistent state.
- **Tumbling windows** are fixed and nonoverlapping; **sliding windows** overlap; **session windows** group activity separated by inactivity. Select a window and aggregation key that match the business event.
- **Event time**, watermarks and allowed lateness address delayed or out-of-order records. **Processing time** follows the worker clock and can change results when events are replayed.
- Enrich streams with a **reference dataset**, keyed state or an external lookup. A left join preserves events with no match; plan refreshes so reference values do not become stale.
- **Kinesis Data Analytics SQL** terminology includes in-application streams, reference tables and continuous windowed queries. The SQL application service is discontinued; apply the processing concepts through supported runtimes such as Flink.
- **Random Cut Forest (RCF)** assigns anomaly scores to unusual observations. Persist source events and derived results for investigation; route alerts through a supported output or Lambda to SNS.

Reliable orchestration and warehouse loading

- **Glue workflows** coordinate Glue jobs, crawlers and triggers. **Step Functions** orchestrates workflows spanning services such as EMR and Glue with branching, retries and failure handling.
- **S3 event notifications** can start processing promptly; **EventBridge** supports schedules. Events may be repeated or reordered, so deduplicate triggers and avoid writing outputs into an input prefix that retriggers the workflow.
- Use **Lambda** for short, bounded transformations or orchestration calls. Long-running distributed jobs belong on an appropriate processing service; the function need not remain running while that job executes.
- **Redshift COPY** loads many S3 files in parallel. Use a single COPY across a file set, reasonably balanced file sizes and enough files to use available slices; avoid serial single-row inserts for bulk loads.
- A **COPY manifest** lists exact S3 objects, including files in different prefixes. **UNLOAD** exports query results to S3; cataloging or using Spectrum does not load rows into an internal table.
- Load updates into a **staging table**, then merge or replace matching target rows transactionally. Combine retry policies, bounded concurrency and load tracking so reruns do not duplicate data or lose batches.

Analysis & Visualization

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

FRESHNESS TO INSIGHT

Source Ingestion delay → **Process** ETL and metadata → **Serve** Query or SPICE → **Visualize** Refresh and explore

04 Analysis and Visualization 18%

Athena querying and cost controls

- **Athena** provides serverless SQL over S3 using catalog metadata and supports JDBC/ODBC clients. Query relevant partitions and columns from compressed Parquet or ORC to reduce scan cost and runtime.
- **CREATE TABLE AS SELECT (CTAS)** materializes transformed query results as a new dataset. It is useful for format conversion and reusable aggregates; the output does not refresh itself when sources change.
- **Workgroups** separate query execution, history, metrics and result settings by team or application. IAM controls workgroup access; S3, Glue, Lake Formation and KMS permissions still govern data access.
- A **per-query data scan limit** cancels a query after its threshold is exceeded. A **workgroup aggregate usage threshold** supports monitoring and notifications; it is not the same per-query cancellation control.
- Enforce **result location and encryption** through workgroup settings when needed. Track usage in CloudWatch and notify through SNS; secure result buckets because query output can expose sensitive records.
- For **query timeouts**, optimize scans and joins first and inspect service quotas for eligible increases. A JDBC client timeout and the Athena service execution timeout are different constraints.

Redshift table design and maintenance

- **DISTSTYLE KEY** collocates matching join keys across tables. Choose a frequently joined key with balanced values; skew can leave most work on a few slices despite ample cluster capacity.
- **DISTSTYLE ALL** replicates a table on every node and can help small, slowly changing dimensions. It increases storage and load work; avoid blindly replicating large or frequently updated tables.
- **DISTSTYLE EVEN** distributes rows without a join key. **AUTO** lets Redshift choose distribution behavior. A destination field is useful as a distribution key only when it also fits joins and skew constraints.
- **Sort keys** help skip blocks for selective predicates, often time ranges. They solve a different problem from distribution keys, which influence data placement and network movement.
- **ANALYZE** maintains optimizer statistics. **VACUUM** addresses deleted space and sort order as applicable; automatic maintenance may lag behind heavy changes. Inspect statistics and unsorted data after large refreshes.
- **Elastic resize** changes supported capacity configurations quickly; **classic resize** supports a broader set of changes and may take longer. Compare storage, CPU, I/O and interruption needs before choosing a resize path.

Analysis & Visualization

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

FRESHNESS TO INSIGHT

Source Ingestion delay → **Process** ETL and metadata → **Serve** Query or SPICE → **Visualize** Refresh and explore

Redshift workloads and analytical access

- **Workload management (WLM)** separates workloads into queues and allocates resources. Automatic WLM manages concurrency and memory; assign priorities to protect business-critical work from heavy ETL or ad-hoc queries.
- **Concurrency scaling** adds capacity for eligible workload bursts. It helps queue contention; it does not fix poor SQL, data skew or every slow individual query.
- **Short query acceleration (SQA)** gives selected short queries a dedicated execution path so they can avoid waiting behind long-running queries.
- **Query monitoring rules** contain a unique name, one to three predicates and an action. Use supported actions such as logging or aborting runaway work; a rule is not an arbitrary capacity-scaling function.
- **Materialized views** precompute reusable query results and require refresh management. **Spectrum** queries S3 files; supported **federated queries** access operational databases. Check connector compatibility before designing cross-source joins.

QuickSight datasets and refresh

- **QuickSight** connects to sources such as Redshift, Athena, S3 and supported databases or SaaS services. Use Athena to query columnar S3 datasets; direct S3 file import uses supported file formats and a manifest.
- **SPICE** is an in-memory analytical store that reduces repeated source queries. Schedule refreshes for the required freshness and allow for ingestion time; a successful source update does not refresh SPICE automatically.
- **Direct query** reads the backing source when a visual runs, trading source workload and latency for fresher results. End-to-end freshness also includes ingestion, ETL, catalog updates and dashboard refresh behavior.
- If an Athena-backed **SPICE import fails**, inspect QuickSight authorization for source buckets, query-result buckets, catalog access and KMS keys. Adding a table to the catalog does not grant access to its S3 objects.
- **Authors** create analyses and dashboards; **readers** consume shared dashboards. Share through groups and folders to simplify permissions, and select the edition and capacity suited to identity and refresh requirements.

Analysis & Visualization / Security

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

FRESHNESS TO INSIGHT

Source Ingestion delay → **Process** ETL and metadata → **Serve** Query or SPICE → **Visualize** Refresh and explore

Visual selection and operational dashboards

- **Line charts** show time trends. **Heat maps** compare metric intensity across two categories. **Geospatial visuals** reveal regional patterns; **pivot tables** support grouped values and summaries.
- **Drill-down** moves through a hierarchy such as country, state and city. Filter or navigation actions let readers investigate details without obscuring the overview.
- QuickSight **forecasting** projects supported time-series measures. **ML-powered anomaly detection** highlights unusual behavior and contributing dimensions; anomalies need investigation rather than automatic interpretation as fraud.
- **OpenSearch Dashboards** supports search-driven operational views. A five-second dashboard refresh cannot bypass ingestion buffers, index refresh or a delayed source pipeline.
- Excessive **OpenSearch shards** consume heap and coordination resources. Monitor JVM memory pressure, indexing rejections and shard sizes; dedicated cluster-manager nodes do not replace data-node capacity.
- Use **Index State Management (ISM)** to transition suitable older, read-only indexes to **UltraWarm** for lower-cost searchable history. A snapshot requires restore for normal searches and is a separate recovery mechanism.

05 Security 18%

Identity, authorization and data sharing

- **IAM** authorizes AWS actions; database permissions authorize SQL access. Grant Redshift users or roles column-level **SELECT** for sensitive fields, or expose restricted views and remove unintended base-table access.
- **Lake Formation** governs registered S3 data through catalog databases, tables and columns. Grant only needed SELECT permissions through integrated query engines; separately authorize service APIs and query-result storage.
- Restrict direct **S3 access** so it cannot bypass Lake Formation controls. An S3 bucket policy grants object access, not SQL column filtering; IAM roles alone do not parse Parquet fields.
- For **cross-account sharing**, grant Lake Formation permissions to the receiving account or principal, complete applicable resource sharing, and create resource links where needed. The recipient still grants its users appropriate local access.
- **AWS Data Exchange** supports governed publication and subscription of data products. A presigned S3 URL grants temporary object access; it does not by itself provide subscription lifecycle management.

Security

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

PROTECT EACH BOUNDARY

- **Identity** — Authenticate
- **Permissions** — Authorize data
- **Encryption** — Protect content
- **Audit** — Record activity

QuickSight and EMR access boundaries

- **QuickSight row-level security** uses dataset rules to restrict records by user or group, such as a salesperson's territory. Dashboard filters and separate SPICE allocations are not authorization boundaries.
- **Namespaces** isolate QuickSight users and groups for multitenant arrangements. Combine tenant identity, asset sharing and dataset restrictions; namespace separation alone does not filter rows in a shared database.
- Use supported **SAML federation** or **Active Directory integration** for enterprise sign-in. Authentication establishes identity; QuickSight sharing and dataset permissions determine what that identity can see.
- **Kerberos** authenticates EMR users. **EMRFS IAM role mappings** can associate users or groups with S3 access roles; configure role trust and AssumeRole permissions, and keep the fallback instance role least privileged.
- **Kafka ACLs** authorize topic operations for authenticated principals, including certificate distinguished names with mutual TLS. TLS protects transport; security groups control network reachability rather than per-topic writes.

Encryption, masking and key control

- Use **TLS** for data in transit and service encryption for data at rest. **KMS customer managed keys** support key policies, lifecycle control and auditing; authorize both the service role and key use as required.
- **Kinesis server-side encryption** protects stored stream records without rewriting producer payloads. **SSE-KMS** protects S3 objects with KMS controls; **SSE-S3** uses S3-managed keys. Protect query results and audit logs separately.
- **Application-encrypted fields** require compatible client decryption before plaintext arithmetic. Redshift COPY does not decrypt arbitrary application ciphertext merely because its role can use a KMS key.
- **Mask or remove PII/PHI before persistence** when policy forbids storing plaintext. Check raw backups, rejected records, logs and error prefixes too. Encryption limits access but does not remove personal information.
- **Redshift KMS encryption** and **HSM encryption** have different migration procedures. Moving an unencrypted cluster to HSM encryption requires a new encrypted cluster and data migration, not a parameter-group change.
- **HSM integration** requires trusted client/server certificates and compatible cluster support; it is not supported on RA3 nodes. Plan key availability and rotation separately from KMS automatic key rotation.

Security

AWS Certified Data Analytics - Specialty

KNOWLEDGE CHEATSHEET

PROTECT EACH BOUNDARY

- **Identity** — Authenticate
- **Permissions** — Authorize data
- **Encryption** — Protect content
- **Audit** — Record activity

Private connectivity

- **Security groups**, routes, DNS and credentials all affect database connectivity. For public QuickSight-to-Redshift connections, allow the applicable QuickSight source ranges; use a VPC connection for supported private access.
- **Interface VPC endpoints** provide private access to supported service APIs, such as Athena. Hybrid clients need private network connectivity and DNS; plan access to S3 results and other dependencies as well.
- **Direct Connect** provides dedicated connectivity; a **site-to-site VPN** provides encrypted tunnels. A private circuit alone is not application TLS. VPC-attached Flink needs routes to MSK, databases and any on-premises Kafka brokers.

Public access and audit evidence

- **EMR block public access** helps prevent cluster creation with overly permissive security-group rules. Security configurations control encryption and authentication; also select appropriate private networking and storage access.
- **Redshift audit logging** captures connection and user activity evidence; enable user activity logging when SQL text is required. **CloudTrail** records AWS API activity, and VPC Flow Logs record network flows.
- **Cross-account CloudWatch Logs subscriptions** need a destination, delivery role and destination access policy plus the sender's subscription filter. Protect delivery permissions and follow the service's regional placement rules.