AWS Certified Machine Learning Engineer - Associate Cheat Sheet

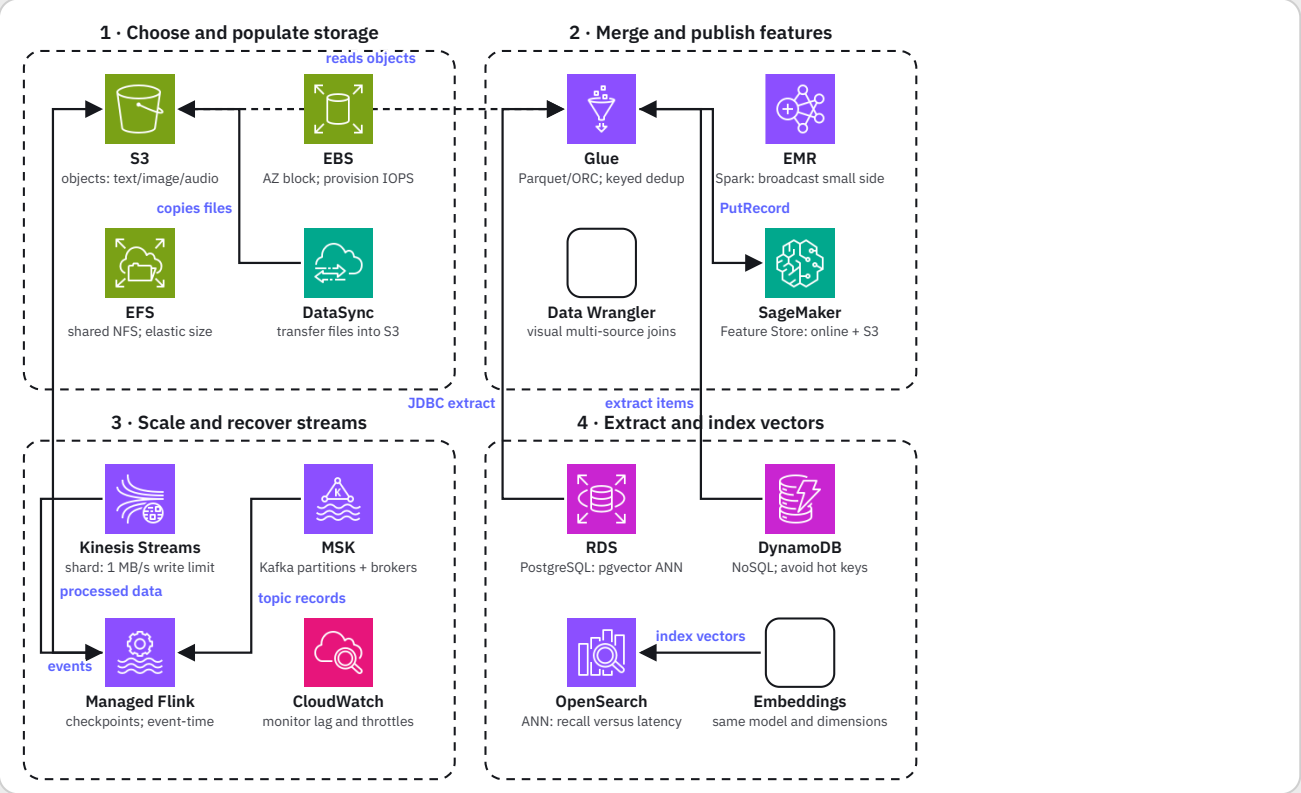
MLA-C02 (beta) · Domain 1 of 4

BlowStack

DOMAIN 1 · DATA PREPARATION FOR ML AND AI · 28% OF THE EXAM · TASK STATEMENT 1.1

Collect and store data

Extract, store, merge, stream, index



1 Choose and populate storage

Use **S3 lifecycle** for tiering; retrieval delay and fees vary by class. Amazon FSx provides specialized file systems. Price-performance includes throughput and request costs; residency covers replicas, logs, and backups.

2 Merge and publish features

Use **Feature Store** online storage for low-latency reads and its S3 offline store for historical training data; configure the required stores. Joins need compatible keys; deduplication needs explicit identity and ordering rules.

3 Scale and recover streams

Use retained streams for **replay**, Amazon Data Firehose for buffered delivery. Kinesis on-demand manages capacity; provisioned mode needs shard scaling. Flink uses checkpoints for recovery and watermarks for event-time windows.

4 Extract and index vectors

Choose **pgvector** to combine vectors with relational SQL; OpenSearch supports vector retrieval with search filters. Size indexes for vector count and dimensions; metadata schemas preserve source, timestamp, and access attributes.

Exam traps

- **JSON/CSV are not columnar**: JSON preserves nested events; CSV needs parsing/schema rules. Parquet/ORC support column pruning; S3 partition pruning additionally avoids irrelevant objects.
- **More capacity does not cure skew**: hot partition or join keys can still bottleneck. Repartition work; a broadcast join requires the small side to fit in every worker's memory.
- **ANN is not exact search**: higher speed can sacrifice recall. S3 object storage alone is not a vector index; storing embedding files does not automatically provide nearest-neighbor queries.

Get practice exams and lessons for AWS Certified Machine Learning Engineer - Associate

AWS Certified Machine Learning Engineer - Associate Cheat Sheet

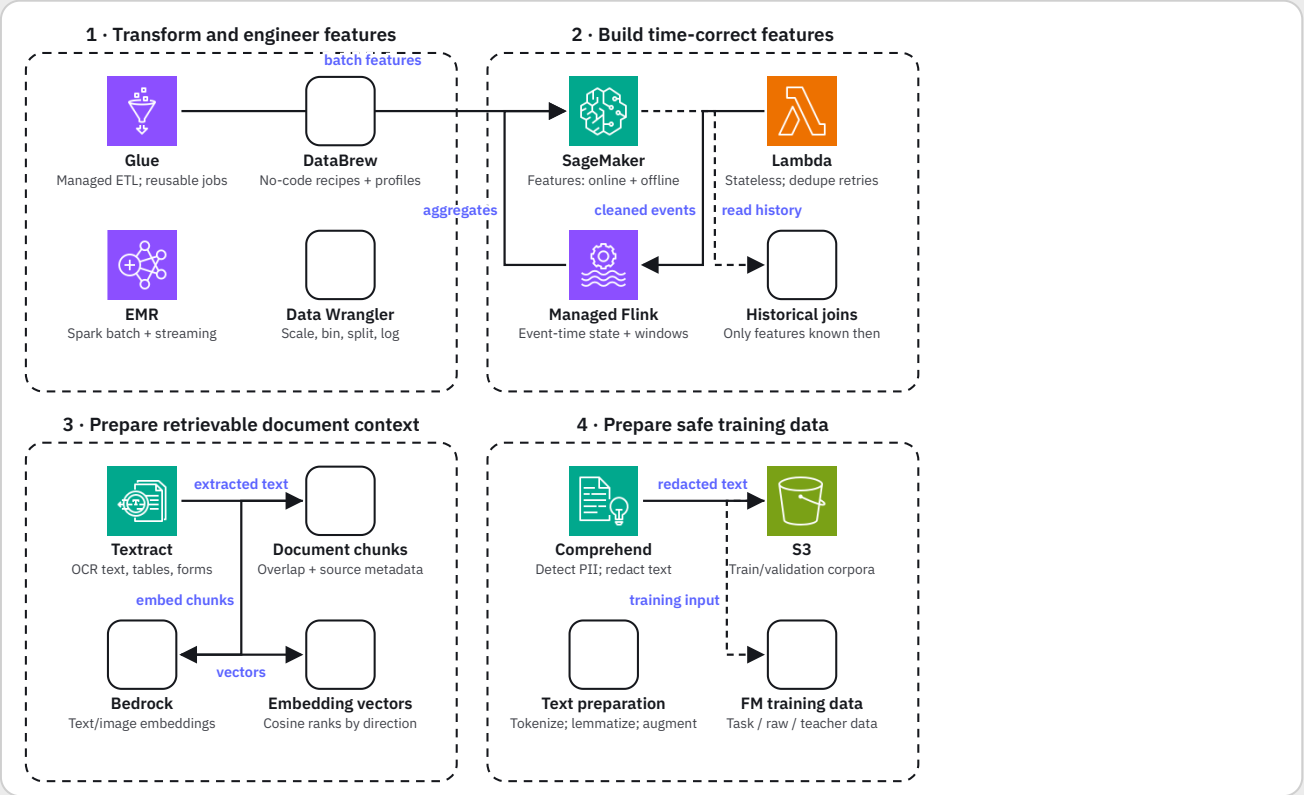
MLA-C02 (beta) · Domain 1 of 4

BlowStack

DOMAIN 1 · DATA PREPARATION FOR ML AND AI · TASK STATEMENT 1.2

Perform data transformation, feature engineering, and pre-processing

Transform, aggregate, retrieve, specialize



1 Transform and engineer features

Fit on training only; reuse transforms at serving. Min-max: (x-min)/(max-min); standardize: (x-mean)/std. L2 normalization scales each vector; binning trades detail for stability.

2 Build time-correct features

Point-in-time joins exclude future features; shared transforms limit training-serving skew. Flink uses watermarks and lateness policies for windowed aggregates; Lambda needs explicit idempotency for retry-safe effects.

3 Prepare retrievable document context

Chunk before embedding; preserve headings, source IDs and filter metadata. Model tokenization determines size limits. RAG retrieves relevant chunks as context without changing model weights.

4 Prepare safe training data

Match the objective: fine-tuning uses task examples; continued pre-training uses unlabeled domain text; distillation uses teacher outputs. Domain-specific augmentation must preserve meaning and valid labels.

Exam traps

- **Delivery is not aggregation**: Amazon Data Firehose buffers and delivers streams; its Lambda transformations do not replace Flink event-time windows.
- **Masking is not anonymization**: hidden fields can leave linkage risk. Pseudonymization retains a reidentification path; stable tokens and mapping tables remain sensitive.
- **Preprocessing is model-specific**: lemmatization can erase domain meaning. Query and document embeddings need compatible model spaces, not merely equal vector dimensions.

Get practice exams and lessons for AWS Certified Machine Learning Engineer - Associate

AWS Certified Machine Learning Engineer - Associate Cheat Sheet

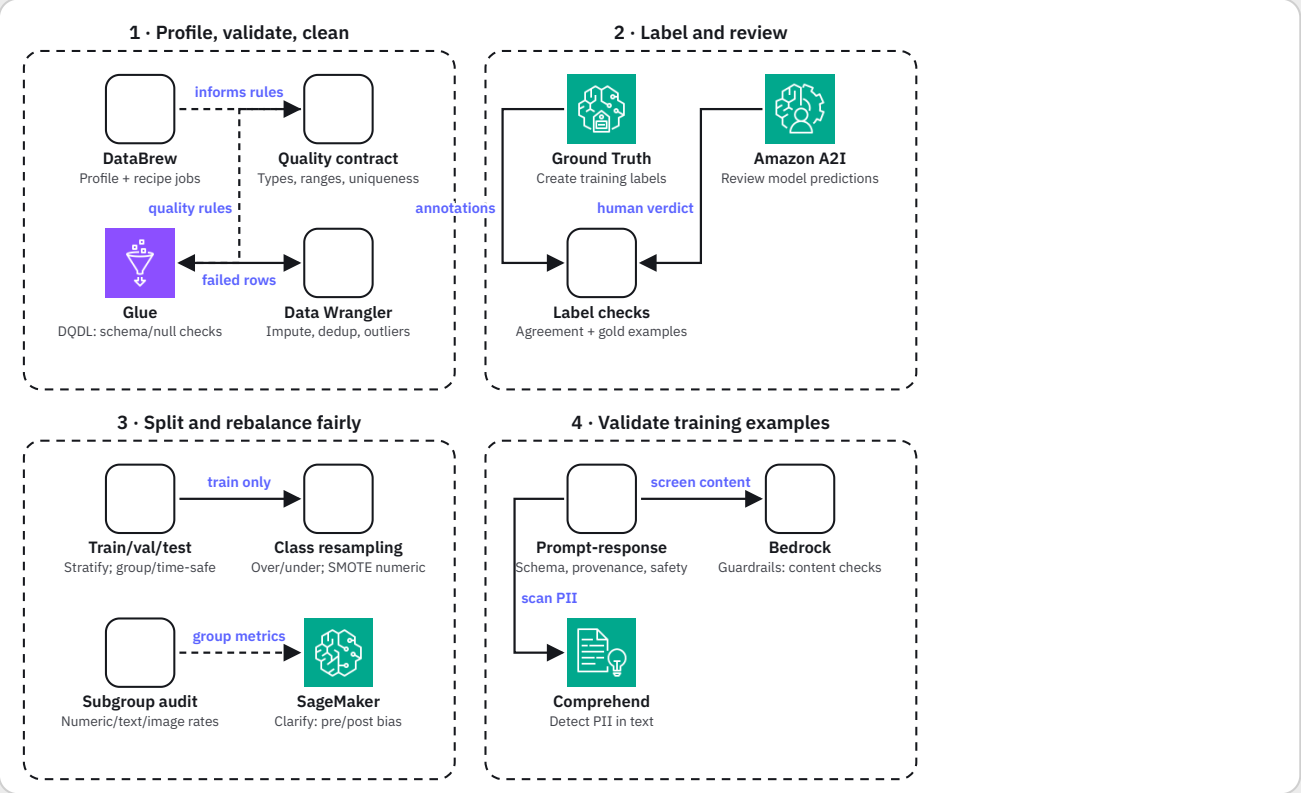
MLA-C02 (beta) · Domain 1 of 4

BlowStack

DOMAIN 1 · DATA PREPARATION FOR ML AND AI · TASK STATEMENT 1.3

Validate data quality and manage bias

Validate, label, rebalance, curate



1 Profile, validate, clean

Profile first to choose rules; Glue Data Quality evaluates DQDL in ETL. DataBrew provides no-code recipes and profile jobs; Data Wrangler cleans and transforms. Outliers need investigation, not automatic deletion.

2 Label and review

Ground Truth builds labeled datasets; A2I routes predictions to configured human-review workflows. Gold examples and annotator agreement check label quality; human review does not guarantee correctness.

3 Split and rebalance fairly

Split first: train fits, validation tunes, test evaluates; preserve groups/time. Clarify measures subgroup bias on tabular features. Shuffle reduces order effects, not imbalance; augment training only with valid labels.

4 Validate training examples

Curate before training: validate required fields, nonempty responses, source/version and integrity hashes. Explicitly screen content and sensitive data; safety screening alone does not establish data integrity.

Exam traps

- **Prevent leakage**: deduplicate/group related records before splitting; fit imputers and resample only on training data. Validation/test retain target prevalence, not artificial class balance.
- **SMOTE interpolates numeric neighbors**, not raw text or images. Use meaningful feature distances or modality-specific augmentation. Balanced classes do not guarantee representative subgroups or fairness.
- **Safety is not integrity**: Bedrock Guardrails do not automatically validate an entire training corpus or its schema/provenance. Comprehend PII detection is not a general harmful-content classifier.

Get practice exams and lessons for AWS Certified Machine Learning Engineer - Associate

AWS Certified Machine Learning Engineer - Associate Cheat Sheet

MLA-C02 (beta) · Domain 1 of 4

BlowStack

DOMAIN 1 · DATA PREPARATION FOR ML AND AI · WHAT TO REMEMBER

Concepts and services of Data Preparation for ML and AI

Concepts to know

Train-Validation-Test Split

Separate learning, tuning, and final evaluation data. **Prevent leakage**: split by time for forecasting or by entity for repeated subjects; fit preprocessing and resample only on training data.

Feature Engineering

Transform raw inputs into useful predictors. **Match transformations to the model and distribution**: scale distance-based inputs, consider logs for positive skew, and reuse fitted transformations during inference.

Data Quality

Assess completeness, validity, consistency, and uniqueness before training. **Profile first, then enforce rules**: validate schemas, investigate outliers, handle missing values, and check quality across subgroups rather than only overall.

Core services



Amazon S3

Choose **object storage for scalable training datasets** and multimodal assets. Use partitioned Parquet for selective analytics; choose storage classes by access frequency and retrieval requirements.



AWS Glue

Choose **serverless Spark ETL** for scalable joins, cleansing, and format conversion. Crawlers populate the Data Catalog; Data Quality evaluates explicit rules.



Amazon SageMaker

Use **Feature Store for reusable features**: online storage supports low-latency lookups; offline storage supports historical training datasets. Point-in-time joins prevent future-feature leakage.

Get practice exams and lessons for AWS Certified Machine Learning Engineer - Associate