AWS Certified Machine Learning Engineer - Associate Cheat Sheet

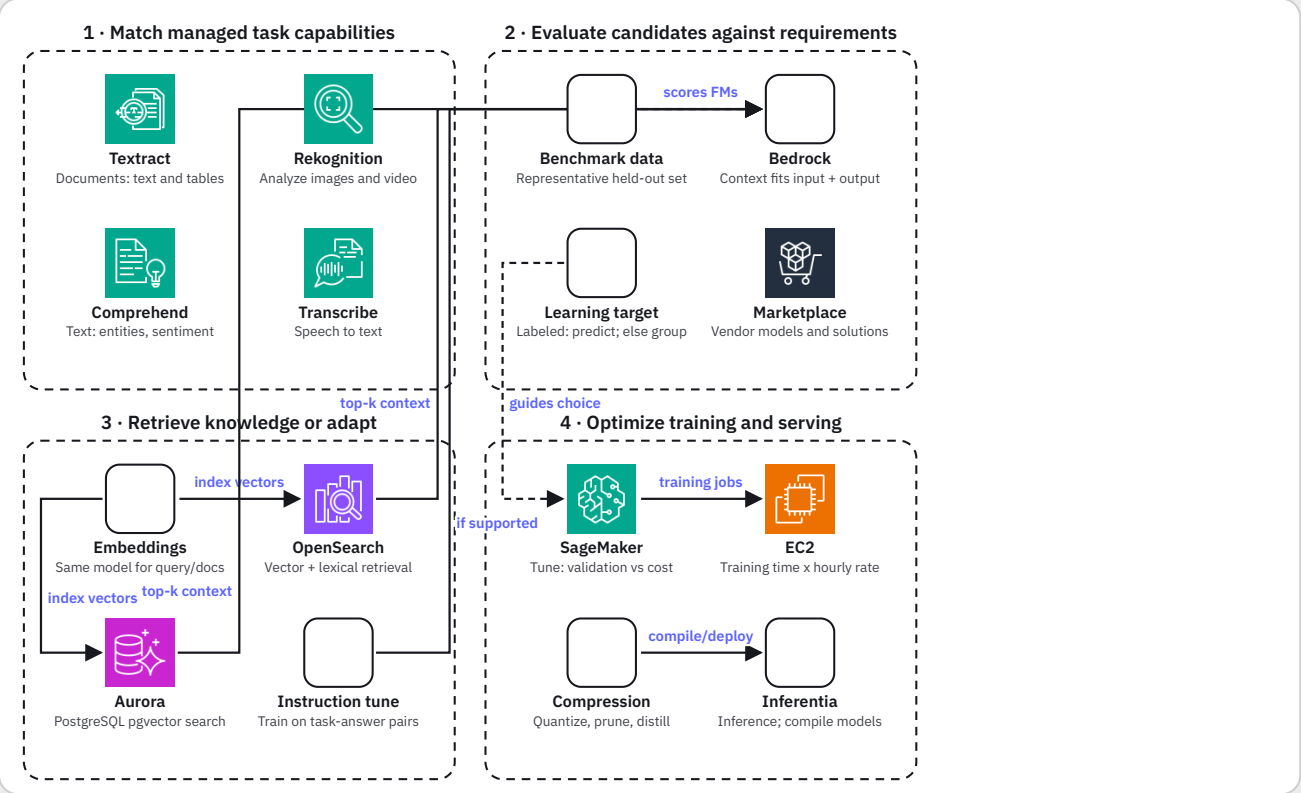
MLA-C02 (beta) · Domain 2 of 4

BlowStack

DOMAIN 2 · ML MODEL AND FOUNDATION MODEL (FM) DEVELOPMENT · 24% OF THE EXAM · TASK STATEMENT 2.1

Choose appropriate modeling approaches for ML and AI solutions

Match, evaluate, retrieve, optimize



1 Match managed task capabilities

Choose a **managed task API** when its capabilities meet requirements. Compare total cost of ownership, including integration, operations, and errors; custom training is not automatically cheaper.

2 Evaluate candidates against requirements

Use **representative evaluation** to compare domain quality, safety, interpretability, p95 latency, and cost. Labels support supervised prediction; unsupervised methods discover structure. Pretrained models reduce startup effort.

3 Retrieve knowledge or adapt

Choose **RAG** for current or private knowledge without retraining. OpenSearch supports hybrid search; Aurora PostgreSQL suits existing relational data. Supported fine-tuning adapts behavior; retrieval quality still needs evaluation.

4 Optimize training and serving

Optimize **validation gain per dollar**: tuning adds training runs; underfitting suggests excess bias, overfitting excess variance. Compression can lower serving latency and cost but requires quality checks and hardware compatibility.

Exam traps

- A larger **context window** is not better retrieval. Instructions, history, retrieved text, and generated output share the limit; irrelevant context also increases latency and token cost.
- **Instruction tuning** defines the training examples and objective; PEFT limits updated parameters, reducing memory and artifact size. Both reuse pretrained capabilities through transfer learning.
- The **largest model** is not automatically best. Simple models can meet accuracy and explanation needs; compressed models need reevaluation, and Inferentia accelerates inference, not training.

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

AWS Certified Machine Learning Engineer - Associate Cheat Sheet

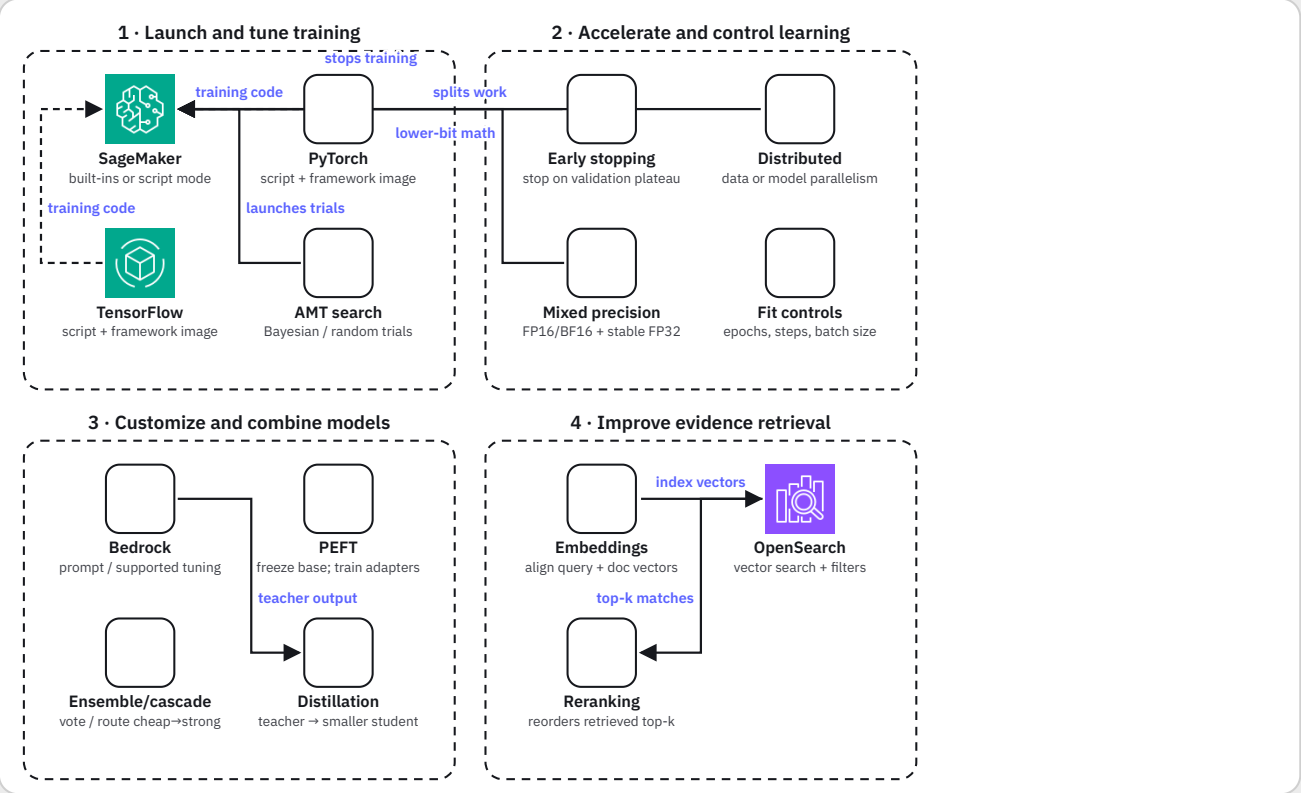
MLA-C02 (beta) · Domain 2 of 4

BlowStack

DOMAIN 2 · ML MODEL AND FOUNDATION MODEL (FM) DEVELOPMENT · TASK STATEMENT 2.2

Train, fine-tune, and customize models for ML and AI solutions

Train, tune, customize, retrieve



1 Launch and tune training

AMT needs parameter ranges and an objective metric. **Bayesian** search learns from trial results; random trials can run independently. Script mode runs custom code in a supported framework container.

2 Accelerate and control learning

Poor train and validation scores: **underfitting**—add capacity or train longer. Good train, poor validation: overfitting—regularize, add data or stop early. Distributed speedups depend on communication overhead.

3 Customize and combine models

Use **fine-tuning** for task behavior; prompts do not update weights. PEFT reduces trainable parameters and memory. Replay earlier examples and evaluate retained skills to limit catastrophic forgetting.

4 Improve evidence retrieval

Optimize **retrieval** through chunking, top-k, filters and embedding evaluation; rerank before generation. Switching embedding models generally requires re-embedding and reindexing documents.

Exam traps

- An **epoch** is one dataset pass; a step is an update. Steps per epoch ≈ examples ÷ effective batch size. Larger batches use more memory and mean fewer updates per epoch, not automatically better accuracy.
- An **ensemble** aggregates predictions and can increase inference cost. A cascade escalates selected requests; distillation trains a cheaper student, whose quality must be checked against the teacher.
- **Reranking** cannot recover documents absent from the candidate set. Fix retrieval recall first. Retrieval supplies evidence without updating model weights; fine-tuning changes model behavior through training.

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

AWS Certified Machine Learning Engineer - Associate Cheat Sheet

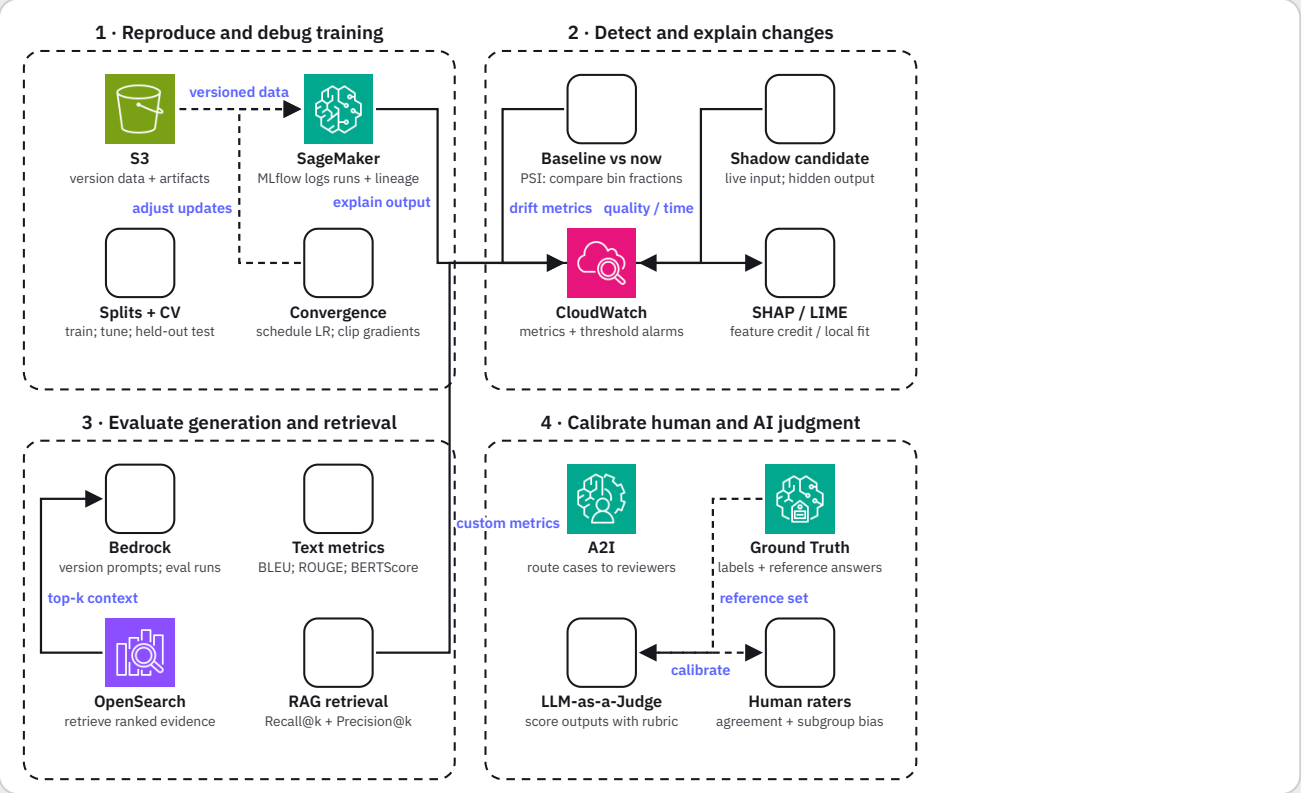
MLA-C02 (beta) · Domain 2 of 4

BlowStack

DOMAIN 2 · ML MODEL AND FOUNDATION MODEL (FM) DEVELOPMENT · TASK STATEMENT 2.3

Analyze and evaluate the performance of ML and AI systems

Reproduce, monitor, compare, evaluate



1 Reproduce and debug training

For **reproducibility**, pin code/data/environment and seeds. Fit preprocessing within each CV fold. Tune hyperparameters on validation data. Unstable loss: lower learning rate; exploding gradients: clip.

2 Detect and explain changes

Model Monitor compares captured data with a baseline; scheduled checks can send violations to CloudWatch. PSI detects label-free distribution shifts. Compare shadow quality, errors and latency before promotion.

3 Evaluate generation and retrieval

Separate retrieval from generation: Recall@k = relevant found / all relevant; Precision@k = relevant found / k. BLEU emphasizes n-gram precision; ROUGE recall; BERTScore token semantic similarity. Version prompts and eval sets.

4 Calibrate human and AI judgment

Human evaluation uses rubrics and inter-rater agreement. Calibrate LLM judges against human scores; check subgroup fairness. Validate output schema, safety and grounding before use. Human review requires configured workflows.

Exam traps

- **Shadow ≠ A/B**: shadow copies live requests, but users see production responses only. A/B sends user-visible traffic to different variants to compare outcomes.
- **Drift ≠ accuracy loss**: PSI needs no labels; accuracy evaluation needs ground truth. CloudWatch alarms do not automatically retrain models.
- **Similarity ≠ truth**: text overlap and semantic similarity cannot prove factuality or safety. SHAP attributes output differences from a baseline; LIME fits a local surrogate. Neither proves causality.

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

AWS Certified Machine Learning Engineer - Associate Cheat Sheet

MLA-C02 (beta) · Domain 2 of 4

BlowStack

DOMAIN 2 · ML MODEL AND FOUNDATION MODEL (FM) DEVELOPMENT · WHAT TO REMEMBER

Concepts and services of ML Model and Foundation Model (FM) Development

Concepts to know

Model Evaluation

Evaluation compares models against task-specific quality, safety, latency, and cost requirements. Use **representative held-out data** for final comparisons; text-overlap metrics alone do not establish factual accuracy.

Fine-Tuning

Fine-tuning adapts pretrained weights using task examples. Choose it for **behavior, style, or task specialization**, rather than frequently changing facts; weigh gains against training cost, overfitting, and catastrophic forgetting.

Retrieval-Augmented Generation

RAG supplies retrieved external evidence without retraining model weights. Choose it for **current or private knowledge**; evaluate retrieval separately from answer quality because missing evidence and poor generation require different fixes.

Core services



Amazon SageMaker

Choose **controlled model development** with built-in algorithms or framework scripts. Automatic model tuning searches hyperparameters; training jobs support custom training and distributed workloads.

Amazon Bedrock

Choose **managed foundation model access** without operating serving infrastructure. Compare models using task evaluations; use prompting, retrieval, or supported customization to address different gaps.



Amazon CloudWatch

Use **metrics, logs, and alarms** to monitor latency, errors, and published model or retrieval-quality signals. Alarms flag threshold breaches; they do not establish prediction accuracy without evaluation.

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