AWS Certified Generative AI Developer - Professional Cheat Sheet

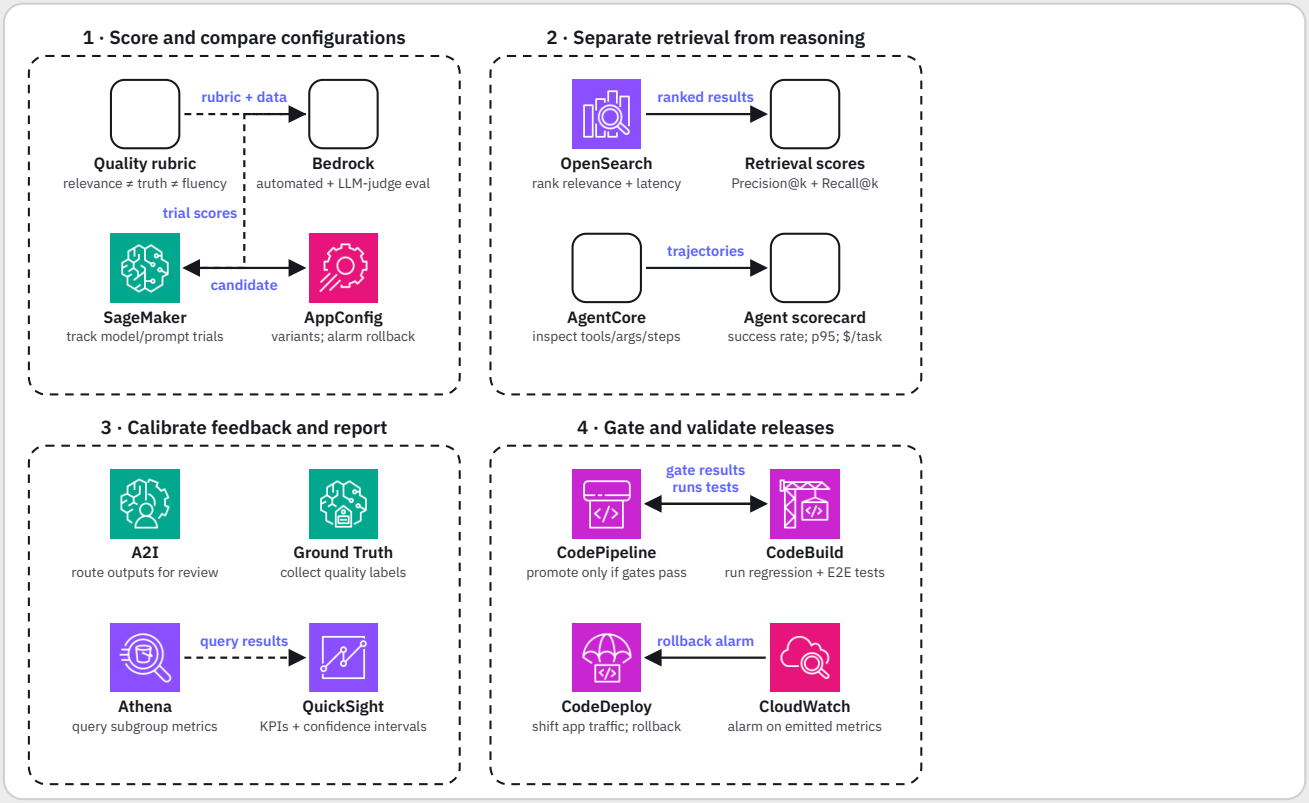
AIP-C01 · Domain 5 of 5

BlowStack

DOMAIN 5 · TESTING, VALIDATION, AND TROUBLESHOOTING · 11% OF THE EXAM · TASK STATEMENT 5.1

Implement evaluation systems for GenAI

Score, compare, review, gate



1 Score and compare configurations

A/B testing compares randomized outcomes; canaries limit exposure. Use matched datasets and rubrics, including consistency; compare token efficiency, latency, and business KPIs. App code routes AppConfig variants.

2 Separate retrieval from reasoning

Separate component scores: Precision@k = relevant results/k; Recall@k = relevant found/all relevant. Check answer grounding separately. Agent evaluation checks task completion and trajectories, not just final wording.

3 Calibrate feedback and report

Human calibration anchors judge scores to reviewed examples. Clear rubrics, reviewer agreement, and adjudication improve labels. Feed user ratings into reviewed changes; report subgroup results and uncertainty, not only averages.

4 Gate and validate releases

Quality gates enforce explicit thresholds on versioned regression sets and synthetic user workflows. Validate hallucinations, semantic drift, consistency, latency, and cost; emit custom quality metrics for alarms and rollback.

Exam traps

- **Fluent is not factual**: relevance, factual accuracy, and evidence grounding are separate checks. A persuasive answer or high user rating does not establish correctness.
- **LLM-as-a-Judge is not ground truth**: calibrate against human labels, control position bias, and version the evaluator and rubric. Aggregate gains can hide subgroup regressions.
- **Deployment tools do not evaluate FMs**: CodeDeploy shifts supported application workloads, not managed Bedrock model versions. CloudWatch needs emitted quality metrics; it does not detect hallucinations itself.

Get practice exams and lessons for AWS Certified Generative AI Developer - Professional

AWS Certified Generative AI Developer - Professional Cheat Sheet

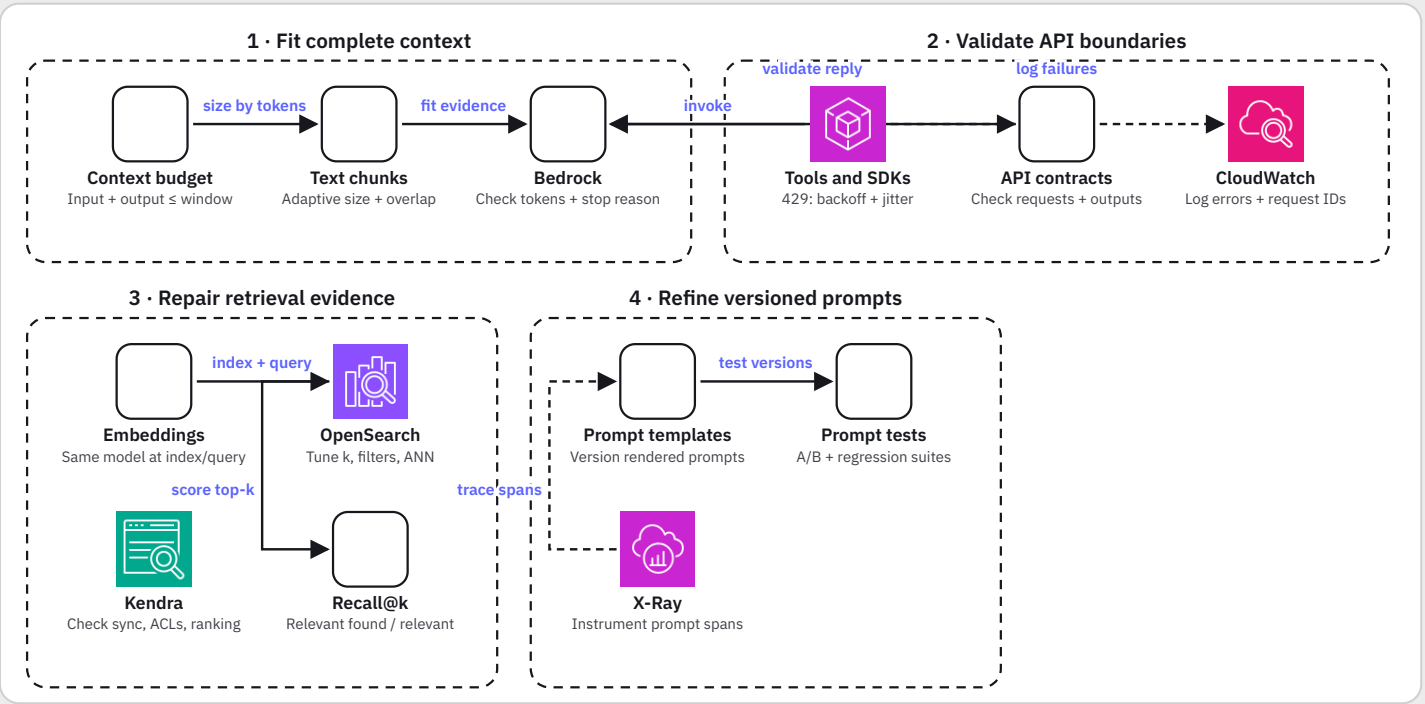
AIP-C01 · Domain 5 of 5

BlowStack

DOMAIN 5 · TESTING, VALIDATION, AND TROUBLESHOOTING · TASK STATEMENT 5.2

Troubleshoot GenAI applications

Fit, validate, retrieve, refine



1 Fit complete context

For **missing content**, count model-specific tokens across instructions, history and retrieved text; reserve output tokens. Dynamically chunk, preserve boundaries and trim redundancy; verify necessary evidence survives.

2 Validate API boundaries

For **integration errors**, validate model ID, Region and API schema first. Retry transient 429/5xx failures with capped exponential backoff and jitter, not malformed requests. Validate output schemas and safety before use.

3 Repair retrieval evidence

Low Recall@k calls for preprocessing/chunk checks, embedding model/dimension checks and filter/index diagnostics. Re-embed after changing embedding models; monitor labeled-query recall and latency over time for drift.

4 Refine versioned prompts

Use **versioned baselines**: change one prompt factor, rerun fixed regression cases, then A/B test task success. Correlate rendered-template logs in Amazon CloudWatch with AWS X-Ray spans to locate confusion and failures.

Exam traps

- A larger output limit does not fix **context overflow**: input and generated tokens share the window. A length stop indicates output truncation, not necessarily missing input.
- **Good retrieval ≠ good answers**: missing evidence requires retrieval repair; present-but-unused evidence points to prompt/context handling. Increasing k can add noise and overflow context.
- **Tracing is not quality testing**: AWS X-Ray needs instrumentation; Amazon CloudWatch needs configured capture. Neither automatically fixes prompts or replaces versioned regression tests.

Get practice exams and lessons for AWS Certified Generative AI Developer - Professional

AWS Certified Generative AI Developer - Professional Cheat Sheet

AIP-C01 · Domain 5 of 5

BlowStack

DOMAIN 5 · TESTING, VALIDATION, AND TROUBLESHOOTING · WHAT TO REMEMBER

Concepts and services of Testing, Validation, and Troubleshooting

Concepts to know

Model Evaluation

Model evaluation measures outputs against representative tasks and defined rubrics. Choose **multiple quality dimensions** over fluency alone; compare relevance, factual accuracy, task success, latency, and cost, with human review to calibrate automated judges.

Retrieval-Augmented Generation

RAG supplies retrieved evidence to a model. **Separate retrieval from generation failures**: Precision@k measures relevant results among the top k; Recall@k measures coverage of relevant evidence. Correct context with a poor answer points toward prompt or model issues.

Regression Testing

Regression testing compares changed prompts, models, or retrieval configurations against a trusted baseline on maintained test cases. Use **quality thresholds rather than exact wording** to gate releases; include semantic quality, tool behavior, latency, and cost.

Core services

Amazon Bedrock

Use **managed model evaluations** to compare foundation model outputs with automated metrics or human review; evaluate retrieval and generated responses separately when diagnosing RAG quality.

Amazon CloudWatch

Use **logs, metrics, and alarms** to diagnose request errors, throttling, token usage, and latency. Application-emitted quality metrics support regression detection and deployment monitoring.

AWS X-Ray

Use **distributed traces** to locate failures and latency across instrumented retrieval, model, and tool calls; traces show which request stage failed rather than only aggregate symptoms.

Get practice exams and lessons for AWS Certified Generative AI Developer - Professional