AWS Certified Generative AI Developer - Professional Cheat Sheet

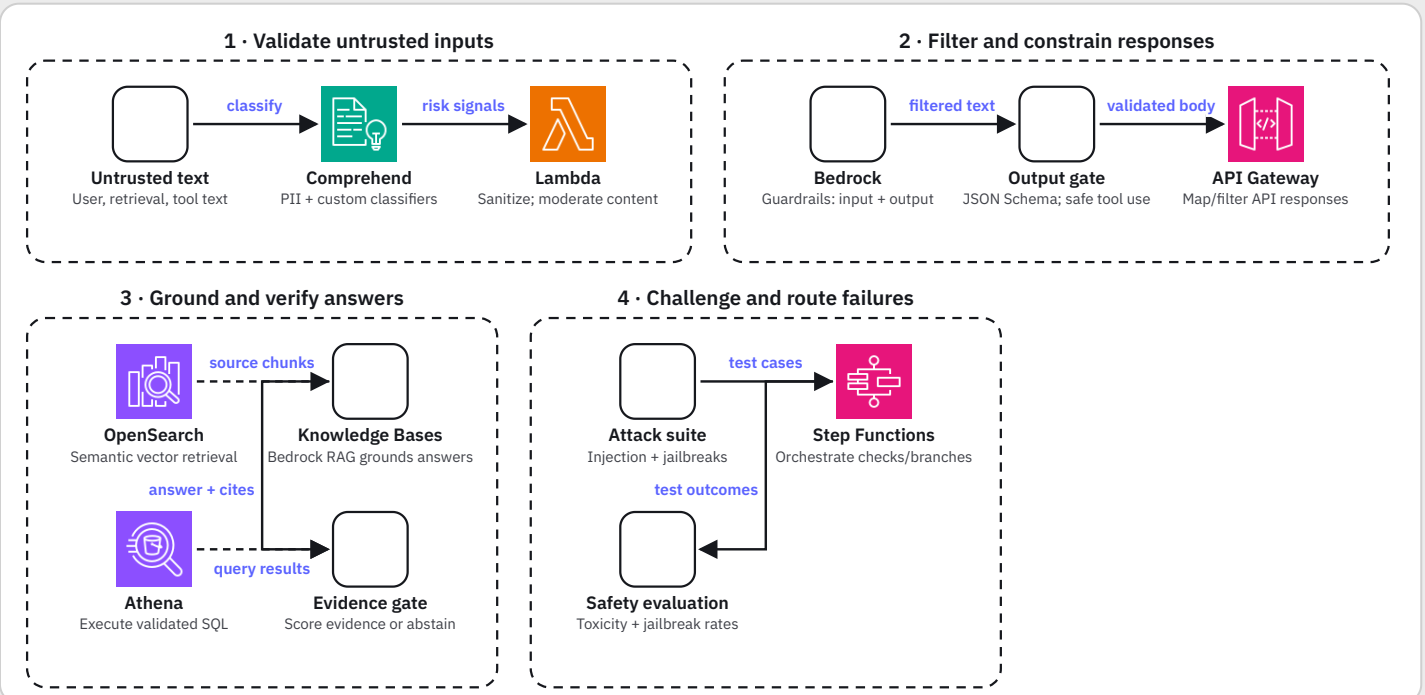
AIP-C01 · Domain 3 of 5

BlowStack

DOMAIN 3 · AI SAFETY, SECURITY, AND GOVERNANCE · 20% OF THE EXAM · TASK STATEMENT 3.1

Implement input and output safety controls

Validate, filter, verify, challenge



1 Validate untrusted inputs

Use **input validation** for format, business rules, and content policy before invocation. Comprehend supplies classification signals; Lambda implements custom checks and rejection logic.

2 Filter and constrain responses

Use **defense in depth**: configure Guardrails for harmful content, sensitive data, and prompt attacks; validate outputs before consumption. API response mappings constrain exposure, not semantic safety.

3 Ground and verify answers

Use **evidence thresholds** to accept, review, or abstain. RAG supplies grounding; similarity and confidence scores require calibration. Text-to-SQL moves calculations into Athena after authorization and query validation.

4 Challenge and route failures

Use **red teaming** against prompts, retrieval, and tools. Step Functions coordinates Lambda checks and block/review branches; representative moderation evaluations measure safety and adversarial robustness.

Exam traps

- **Guardrails are not authorization**: prompt-attack detection reduces risk but cannot guarantee jailbreak prevention or authorize tool actions. Retrieved documents can also carry injections.
- **Schema validity is not factuality**: valid JSON can contain hallucinations. RAG citations, high similarity, and model confidence do not independently prove an answer is correct.
- **Text-to-SQL is not automatically safe**: generated SQL still needs schema checks, read-only permissions, and execution limits. Database execution makes calculations deterministic, not question interpretation.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

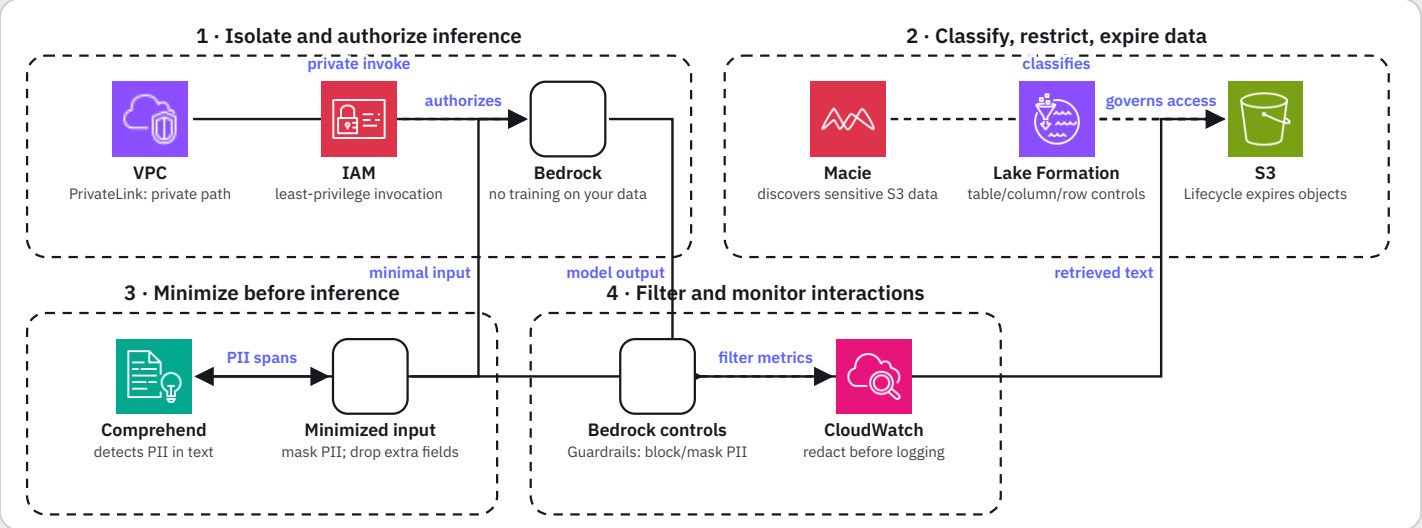
AIP-C01 · Domain 3 of 5

BlowStack

DOMAIN 3 · AI SAFETY, SECURITY, AND GOVERNANCE · TASK STATEMENT 3.2

Implement data security and privacy controls

Isolate, restrict, minimize, filter



1 Isolate and authorize inference

Private access uses VPC interface endpoints; IAM still authorizes calls. Bedrock inputs and outputs are not used to train base models; application privacy controls remain necessary.

2 Classify, restrict, expire data

Discover vs enforce: Macie classifies S3 data; Lake Formation controls governed lake tables. S3 Lifecycle implements retention; deleting noncurrent versions requires corresponding lifecycle rules.

3 Minimize before inference

Minimize first: Comprehend detects PII; application code masks or removes it before invocation. Keep task-required information to preserve utility. Anonymization must address re-identification risk.

4 Filter and monitor interactions

Filter both directions: configure Bedrock guardrails for supported PII in prompts and responses. Associate or apply them explicitly; configure CloudWatch logs, metrics, and alarms without exposing sensitive payloads.

Exam traps

- **PrivateLink is not authorization**: a private network path does not grant model access; IAM and endpoint policies still constrain requests.
- **Macie vs Comprehend**: Macie discovers sensitive data in S3; Comprehend detects PII in application text. Neither automatically masks every prompt.
- **Masking is not anonymization**: reversible tokens and linkable identifiers can still identify users. Guardrails do not sanitize copies already written to application logs.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

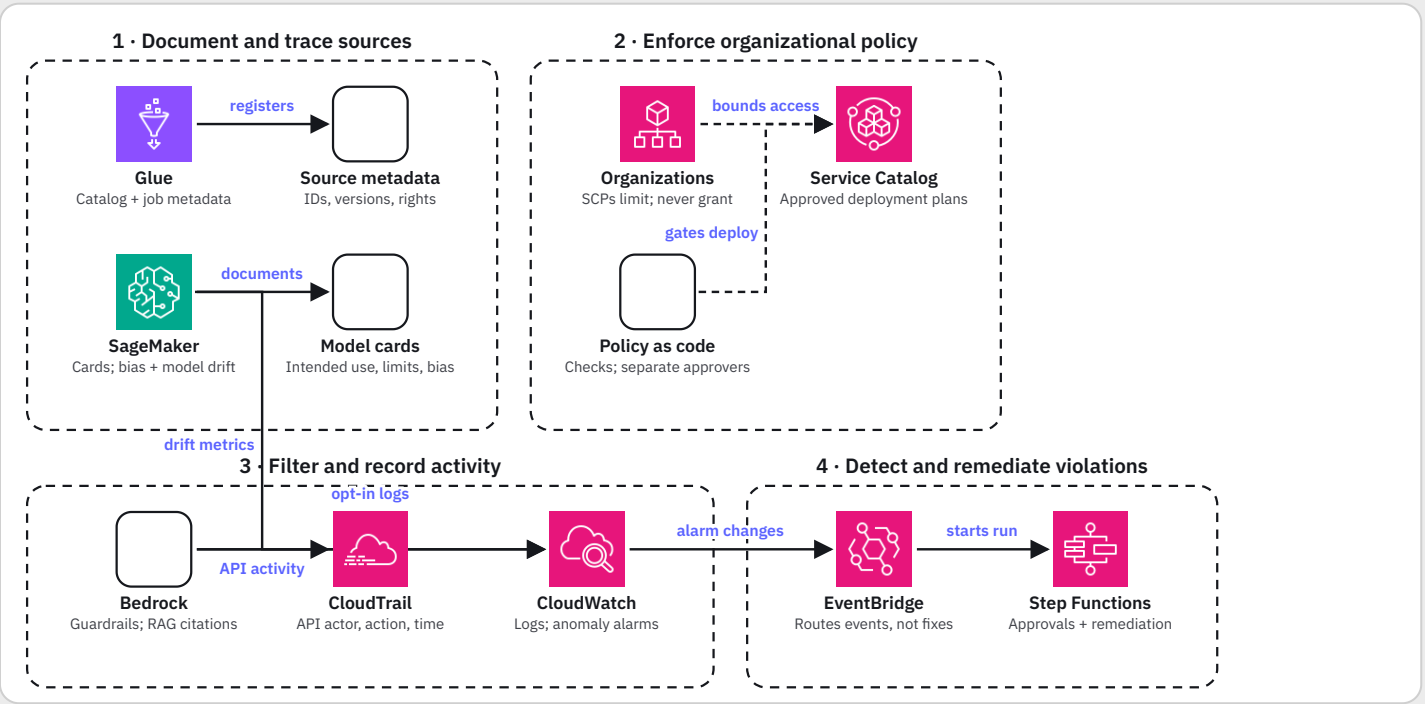
AIP-C01 · Domain 3 of 5

BlowStack

DOMAIN 3 · AI SAFETY, SECURITY, AND GOVERNANCE · TASK STATEMENT 3.3

Implement AI governance and compliance mechanisms

Trace, govern, protect, remediate



1 Document and trace sources

Use **model cards** for suitability evidence. Register sources in the catalog; capture transformation lineage and propagate source IDs. Catalog registration alone is not end-to-end provenance.

2 Enforce organizational policy

Use **policy as code** for repeatable compliance checks. SCPs cap permissions, catalog products standardize deployment, and independent approvers enforce separation of duties.

3 Filter and record activity

Enable **invocation logging** for supported model requests/responses in CloudWatch. CloudTrail audits API activity. Configure Bedrock filters and sensitive-data masking; separately protect and redact stored logs.

4 Detect and remediate violations

Route **drift or anomaly alarms** through EventBridge to Step Functions for approval and remediation. Define baselines, fairness criteria, and misuse rules; monitoring does not automatically retrain models.

Exam traps

- **CloudTrail is not a response archive**: API audit events identify callers and actions; Bedrock invocation logging must be enabled separately for supported request/response payloads.
- **Catalog registration is not complete lineage**: transformation history, source permissions, and downstream attribution require captured metadata. RAG citations do not identify FM training sources.
- **Guardrails do not establish fairness**: content filters and sensitive-data masking do not replace subgroup evaluations, bias-drift monitoring, protected audit records, or regulatory evidence.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

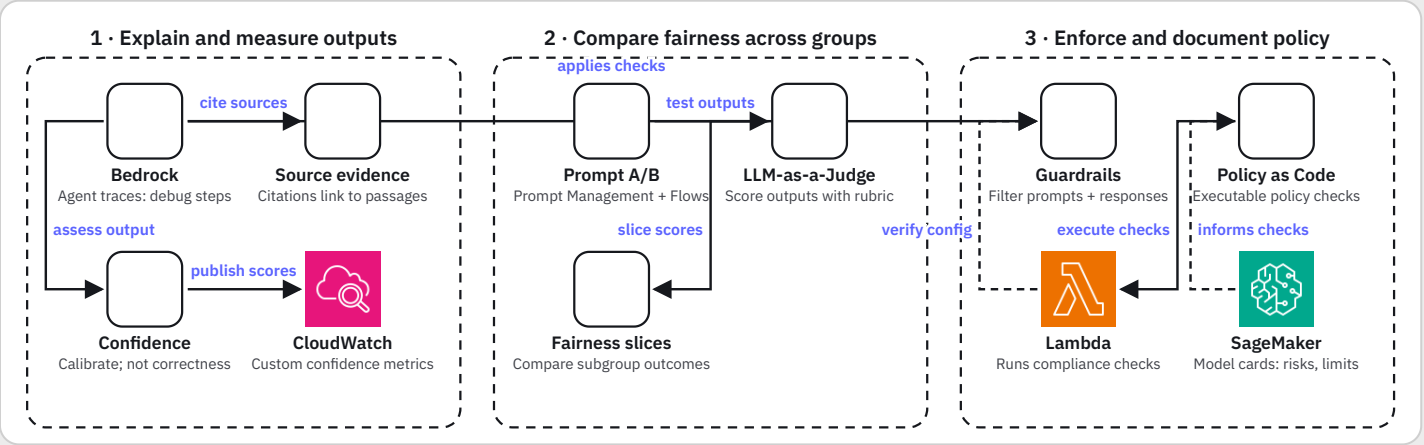
AIP-C01 · Domain 3 of 5

BlowStack

DOMAIN 3 · AI SAFETY, SECURITY, AND GOVERNANCE · TASK STATEMENT 3.4

Implement responsible AI principles

Explain, compare, enforce, document



1 Explain and measure outputs

Use **source evidence** and concise explanations for users; agent traces reveal orchestration steps for debugging. Application logic computes uncertainty metrics; CloudWatch collects and monitors them.

2 Compare fairness across groups

Use Bedrock prompt versions and flows to orchestrate comparable A/B tests. Use **subgroup metrics** to expose disparities; an LLM judge scales rubric scoring but needs validation against human judgments.

3 Enforce and document policy

Use Bedrock Guardrails for configured content controls and Lambda for custom **policy checks**. SageMaker model cards document version-specific limitations and safeguards; documentation does not enforce compliance.

Exam traps

- **High confidence is not factual correctness**. Self-reported model confidence can be miscalibrated; citations still require checking that the evidence supports the answer.
- **CloudWatch does not supply universal fairness metrics**. Applications or evaluation pipelines compute context-specific subgroup metrics and publish them for monitoring.
- **Guardrails are not complete governance**. Content filters do not prove fairness or regulatory compliance; prompt instructions and model cards are not independently enforced controls.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

AIP-C01 · Domain 3 of 5

BlowStack

DOMAIN 3 · AI SAFETY, SECURITY, AND GOVERNANCE · WHAT TO REMEMBER

Concepts and services of AI Safety, Security, and Governance

Concepts to know

AI Guardrails

Independently enforced checks constrain harmful content and sensitive-data disclosure. Choose **layered runtime controls**, not prompt instructions alone; content filtering does not replace authorization or factual verification.

Least privilege

Permissions limit identities to required resources and actions. Choose **authorization at retrieval and tool boundaries** to prevent unauthorized data access or actions; model instructions and private networking do not enforce resource permissions.

Audit Logging

Audit records establish who acted, when, on which resource, and with what outcome. Choose **complementary API and application logs** for traceability; API history alone does not capture prompts, responses, or application decisions.

Core services

Amazon Bedrock

Choose **Guardrails for runtime safety policies**, Knowledge Bases for grounding, and evaluations for safety measurement. Grounding reduces hallucination risk but does not guarantee correctness.

Amazon Comprehend

Choose **PII detection in application text** for preprocessing or redaction workflows. Amazon Macie instead discovers sensitive data stored in Amazon S3.

Amazon CloudWatch

Choose **application logs, metrics, and alarms** to monitor safety violations, uncertainty, and operational anomalies. AWS CloudTrail supplies AWS API activity records rather than complete decision logs.

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