AWS Certified Generative AI Developer - Professional Cheat Sheet

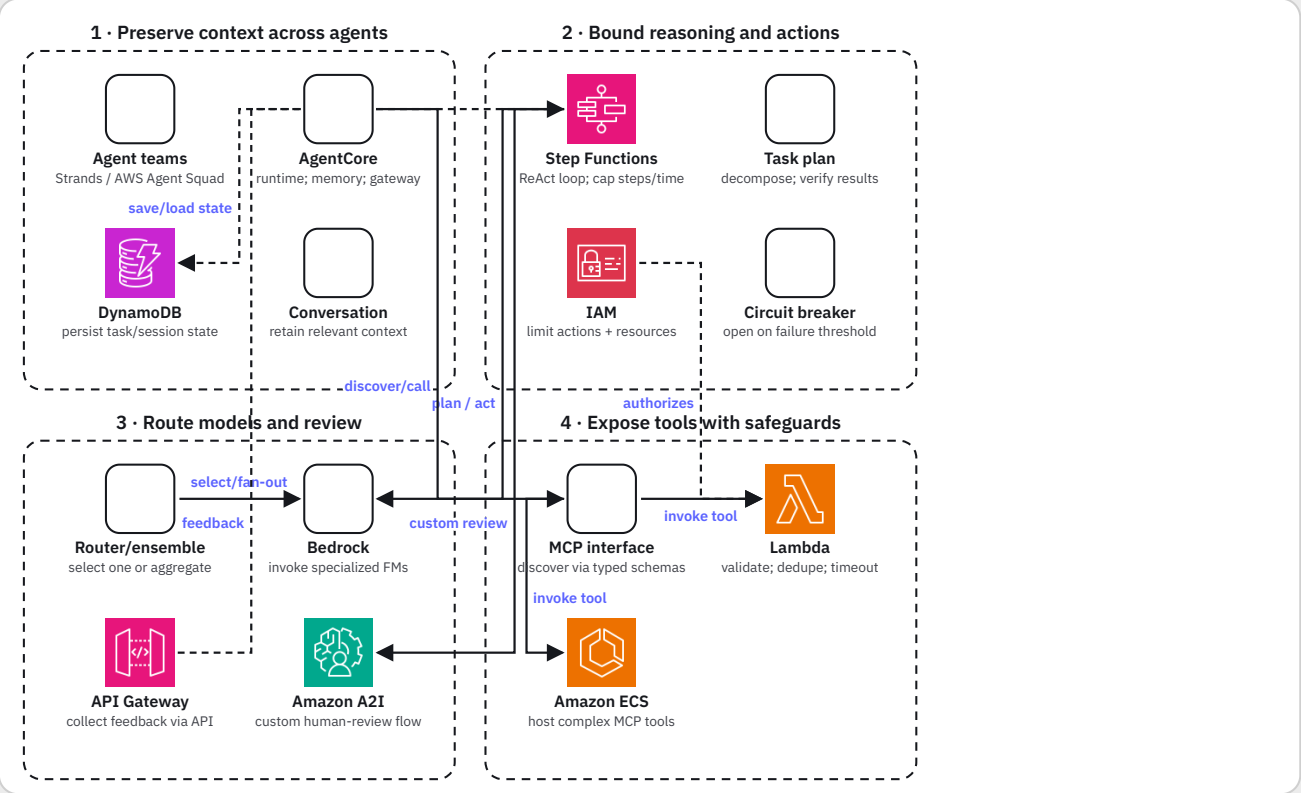
AIP-C01 · Domain 2 of 5

BlowStack

DOMAIN 2 · IMPLEMENTATION AND INTEGRATION · 26% OF THE EXAM · TASK STATEMENT 2.1

Implement agentic AI solutions and tool integrations

Remember, constrain, coordinate, extend



1 Preserve context across agents

Durable state survives agent restarts; conversation memory supplies relevant context, not necessarily every message. Strands and AWS Agent Squad coordinate agents; AgentCore components can be adopted independently.

2 Bound reasoning and actions

Bounded execution needs explicit stop conditions, timeouts and failure handling. ReAct alternates decisions, tools and observations; chain-of-thought prompts do not enforce limits. Circuit-breaker state and recovery need implementation.

3 Route models and review

Routing selects one model; ensembles invoke several and need custom aggregation. Step Functions waits for approval; Amazon A2I needs custom integration for FM review. Feedback is not automatic retraining.

4 Expose tools with safeguards

Stateless MCP: Lambda fits lightweight request/response tools; ECS fits long-running or complex servers. MCP clients standardize access; handlers validate arguments and use idempotency keys to deduplicate retries.

Exam traps

- **Guardrails are not authorization**: content filtering and prompt instructions cannot replace IAM least privilege, tool-argument validation or approval before consequential actions.
- **Tool calling is not execution**: an FM proposes a tool and arguments; application code authorizes and executes it. MCP standardizes access, not trust or permission.
- **Retry is not a circuit breaker**: retries repeat calls; a breaker blocks calls after failures. Timeouts do not prove an action failed, so idempotency must prevent duplicate side effects.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

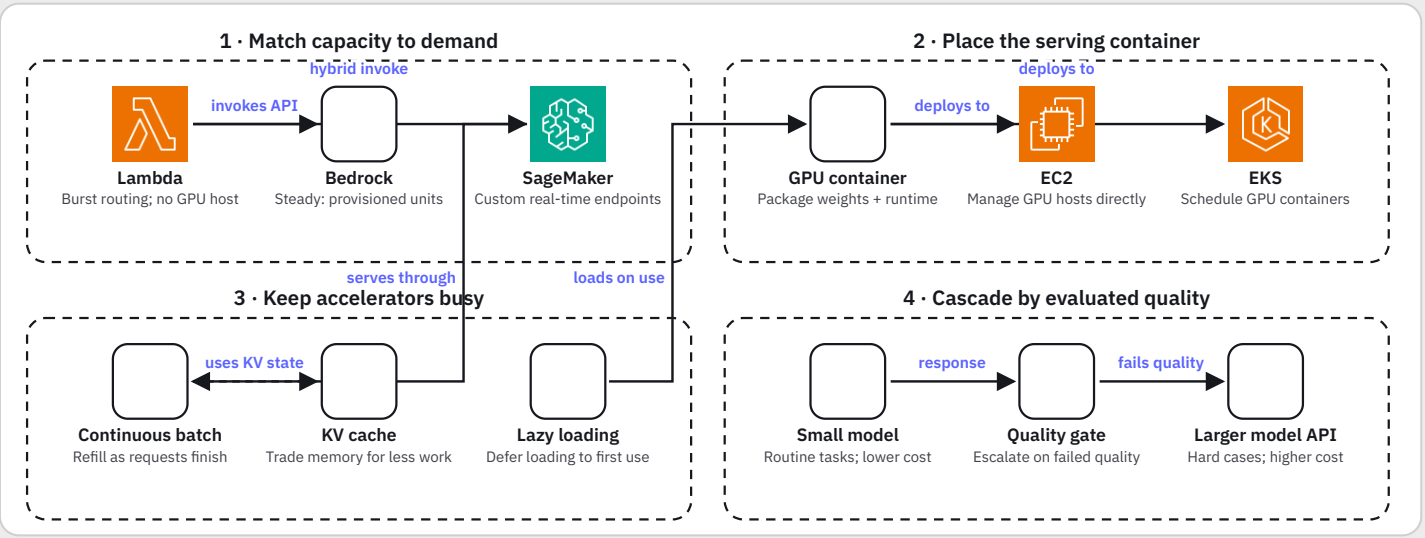
AIP-C01 · Domain 2 of 5

BlowStack

DOMAIN 2 · IMPLEMENTATION AND INTEGRATION · TASK STATEMENT 2.2

Implement model deployment strategies

Match, host, optimize, cascade



1 Match capacity to demand

Predictable load can justify Bedrock Provisioned Throughput where supported; on-demand suits variable demand. Lambda invokes APIs; SageMaker endpoints enable custom hosting alongside managed model APIs.

2 Place the serving container

Control needs favor SageMaker managed endpoints, EC2 host control, or EKS Kubernetes orchestration. Compatible GPU images, model loading, and memory sizing remain deployment responsibilities.

3 Keep accelerators busy

GPU memory must fit weights plus growing KV state. Continuous batching improves utilization; lazy loading reduces startup work but shifts loading latency and possible failures into the request path.

4 Cascade by evaluated quality

Model evaluation calibrates escalation using representative tasks. A small model handles routine requests; a larger API model handles harder cases. Savings depend on escalation rate and added routing latency.

Exam traps

- **Lambda is not an LLM GPU host**: it can route requests and invoke endpoints, but packaging model weights in a function does not provide GPU capacity or eliminate cold starts.
- **Provisioned throughput is not a latency guarantee**: it provides supported-model capacity. Model size, prompt length, output length, and concurrency still affect response time.
- **KV caching is not response caching**: it reuses attention keys and values to avoid repeated computation. Longer contexts and more concurrent sequences still consume accelerator memory.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

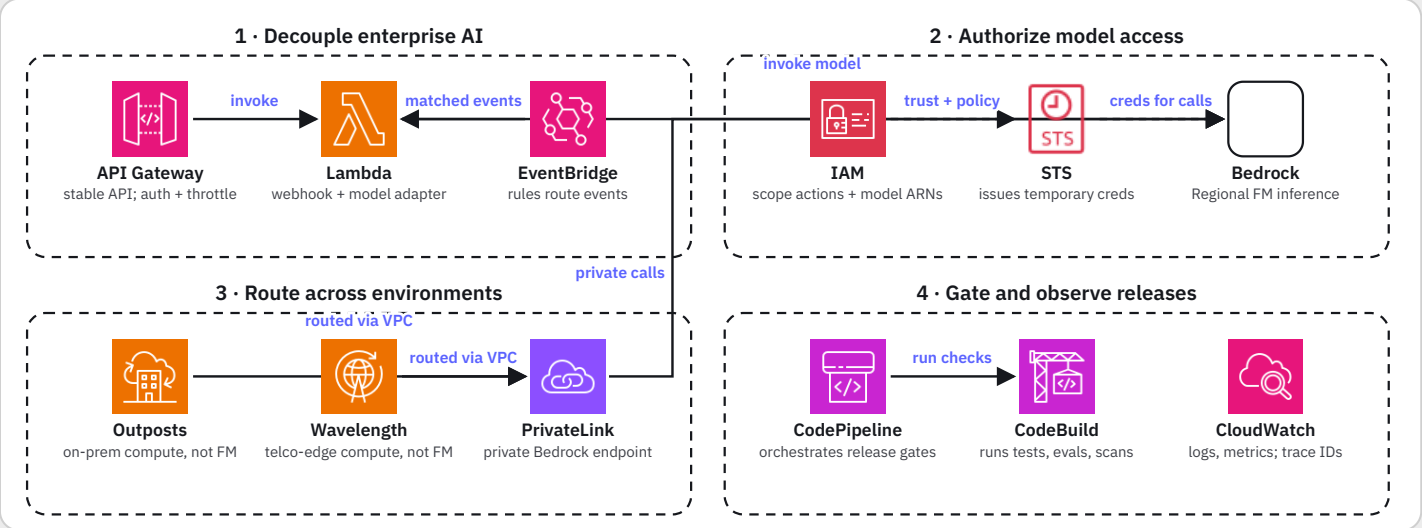
AIP-C01 · Domain 2 of 5

BlowStack

DOMAIN 2 · IMPLEMENTATION AND INTEGRATION · TASK STATEMENT 2.3

Design and implement enterprise integration architectures

Integrate, authorize, route, release



1 Decouple enterprise AI

Use APIs for immediate responses; use **events** for loosely coupled processing. Webhook handlers validate authenticity and deduplicate retries; asynchronous consumers handle failures independently.

2 Authorize model access

Use **federation** to map enterprise identities to scoped roles, then obtain temporary credentials. Role trust controls assumption; permission policies restrict model and data access. Sign-in alone does not authorize inference.

3 Route across environments

Use **private routing** through VPC connectivity to reach a Regional Bedrock endpoint. Outposts keeps supported processing on premises; Wavelength places apps at the telco edge. Neither automatically keeps FM inference local.

4 Gate and observe releases

Use **quality gates** to block releases on failed security scans or GenAI evaluations. Configure approval and rollback paths explicitly; instrument gateway and backend requests with correlation IDs for distributed troubleshooting.

Exam traps

- **Synchronization ≠ event routing**: Amazon AppFlow transfers supported SaaS data; Amazon EventBridge routes events. Consumers still need idempotency and reconciliation.
- **Private access ≠ residency**: AWS PrivateLink avoids public internet transit; compliance still depends on inference location, cross-Region routing, logs, backups and downstream data flows.
- **API gateway ≠ GenAI policy**: Amazon API Gateway provides API controls; semantic prompt checks, token budgets and model-selection policies require additional backend logic or model-service controls.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

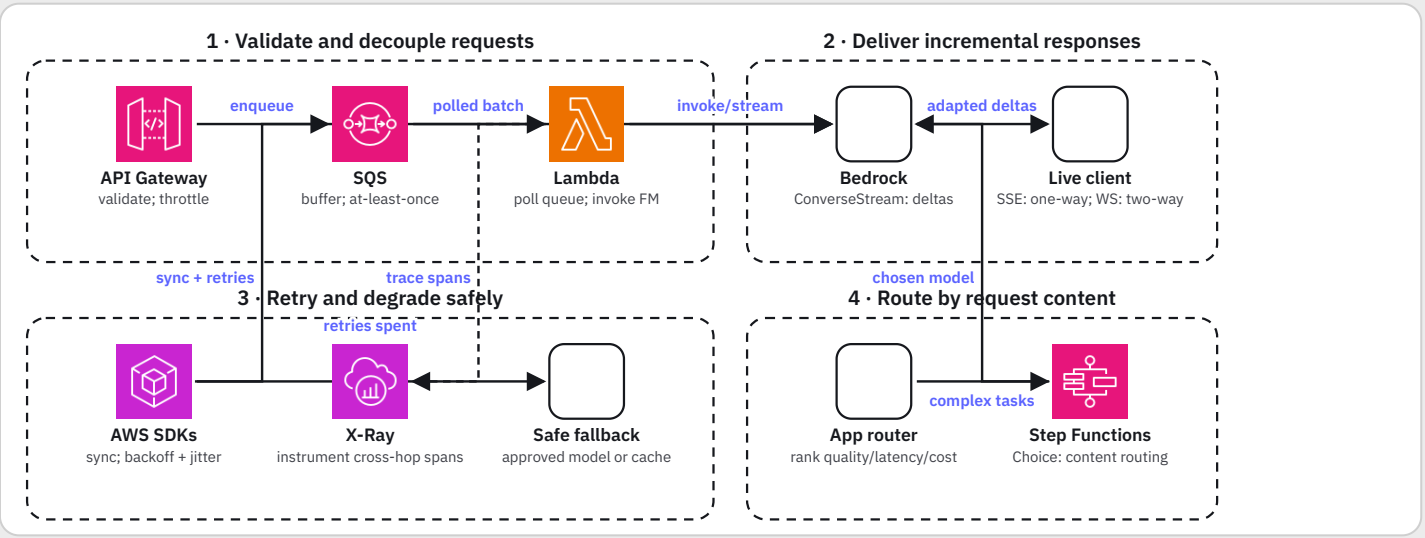
AIP-C01 · Domain 2 of 5

BlowStack

DOMAIN 2 · IMPLEMENTATION AND INTEGRATION · TASK STATEMENT 2.4

Implement FM API integrations

Decouple, stream, recover, route



1 Validate and decouple requests

Use **SQS** when work can wait; Lambda polls and scales consumers. Use SDK calls for synchronous results. Gateway validates request shape; application code checks meaning.

2 Deliver incremental responses

Use **streaming** for incremental output: ConverseStream or InvokeModelWithResponseStream. Application code translates Bedrock events into SSE events or WebSocket messages; streaming does not reduce total generation work.

3 Retry and degrade safely

Use **bounded retries** for throttling/transient faults, then a safe fallback. SDK retry policies need limits; X-Ray requires instrumentation. Gateway throttles callers, not Bedrock model quotas.

4 Route by request content

Use **content-based routing** for specialized FMs: application code handles fixed mappings; Step Functions Choice branches on request attributes. Metric-based selection needs explicit logic, not automatic orchestration.

Exam traps

- **SQS is not exactly-once execution.** Standard queues can redeliver; make workers idempotent, size visibility timeout for processing, and send repeatedly failing messages to a dead-letter queue.
- **Chunked transfer is transport, not token framing.** API Gateway needs a supported REST response-streaming integration; WebSocket callbacks are messages, not HTTP chunks.
- **Validation is not authorization.** Gateway schema checks do not establish caller permissions or business validity; fallback models and cached answers must preserve access and safety controls.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

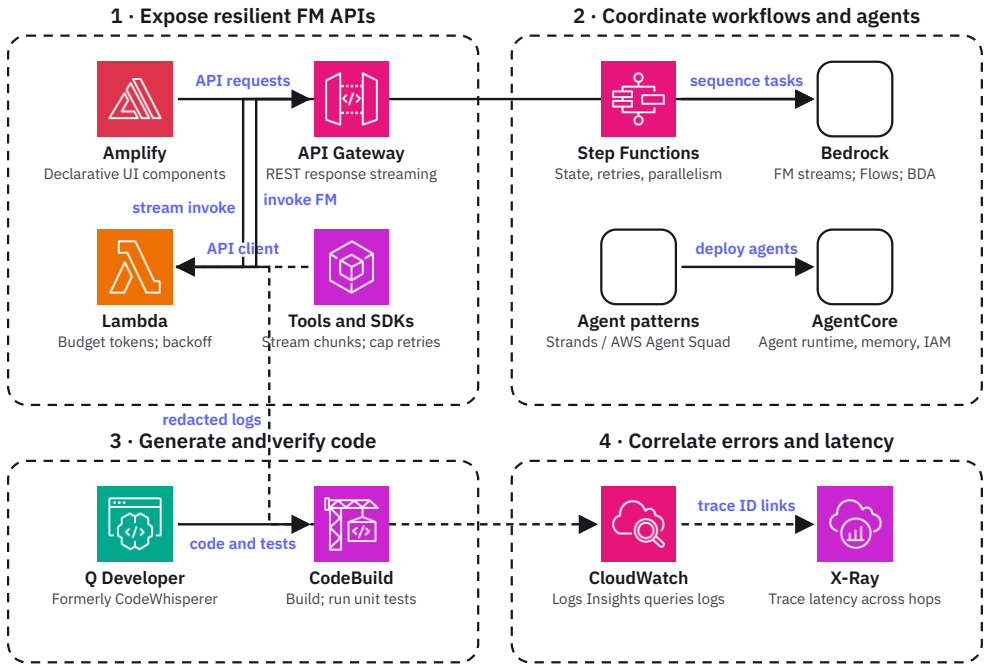
AIP-C01 · Domain 2 of 5

BlowStack

DOMAIN 2 · IMPLEMENTATION AND INTEGRATION · TASK STATEMENT 2.5

Implement application integration patterns and development tools

Expose, orchestrate, test, trace



1 Expose resilient FM APIs

API-first: agree OpenAPI contracts before UI/backend work. Amplify components still need accessibility testing. REST response streaming delivers incremental tokens; it does not remove integration timeouts.

2 Coordinate workflows and agents

Durable state: Step Functions coordinates business tasks; Bedrock Flows chains prompts, and Bedrock Data Automation extracts structured content. AgentCore hosts agents; Strands/Agent Squad define coordination.

3 Generate and verify code

Unit testing: mock FM/tool calls in CodeBuild for deterministic checks; profile live latency separately. Q Developer generates/refactors code, but generated code and tests still require review.

4 Correlate errors and latency

Event correlation: join redacted prompt/response logs and model errors by request/trace ID. Logs Insights searches logs; X-Ray locates slow spans. Q Developer can help classify error patterns.

Exam traps

- Streaming does not enlarge the **context window**. Instructions, history, retrieved content, and reserved output must fit; trim, summarize, or retrieve selectively before invocation.
- **Exponential backoff** needs jitter, bounded attempts, and retryable errors. Blindly retrying timeouts can duplicate CRM writes or tool actions; side effects need idempotency.
- **Tool calling** produces a requested action, not an authorized execution. Application code must validate arguments, enforce permissions, execute the tool, and return its result.

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

AWS Certified Generative AI Developer - Professional Cheat Sheet

AIP-C01 · Domain 2 of 5

BlowStack

DOMAIN 2 · IMPLEMENTATION AND INTEGRATION · WHAT TO REMEMBER

Concepts and services of Implementation and Integration

Concepts to know

Workflow Orchestration

Coordinates state, branching, retries, and approvals across tasks. Choose **explicit workflow control** for bounded agent execution and auditable business processes; individual functions implement tasks rather than maintain the overall workflow.

Tool Calling

Lets models request external operations through defined interfaces. Choose **application-enforced execution controls**: validate arguments, authorize access, and make side effects idempotent; a model's tool request is not permission to execute.

Model Routing

Selects models according to task needs, quality, latency, and cost. Choose **routing for model selection**, cascading for escalation from cheaper models, and ensembles when multiple model outputs must be combined.

Core services

Amazon Bedrock

Choose **managed FM access** without operating inference infrastructure. Use streaming for incremental output and supported Provisioned Throughput configurations for predictable capacity needs.

AWS Step Functions

Choose **stateful orchestration** for multistep agent workflows, conditional model routing, bounded retries, and human approval waits rather than embedding coordination in individual functions.

AWS Lambda

Choose **short-lived, stateless execution** for tool handlers, validation, webhooks, and lightweight MCP servers; externalize durable state and use containers for long-running or resource-heavy tools.

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