AWS Certified Developer - Associate Cheat Sheet

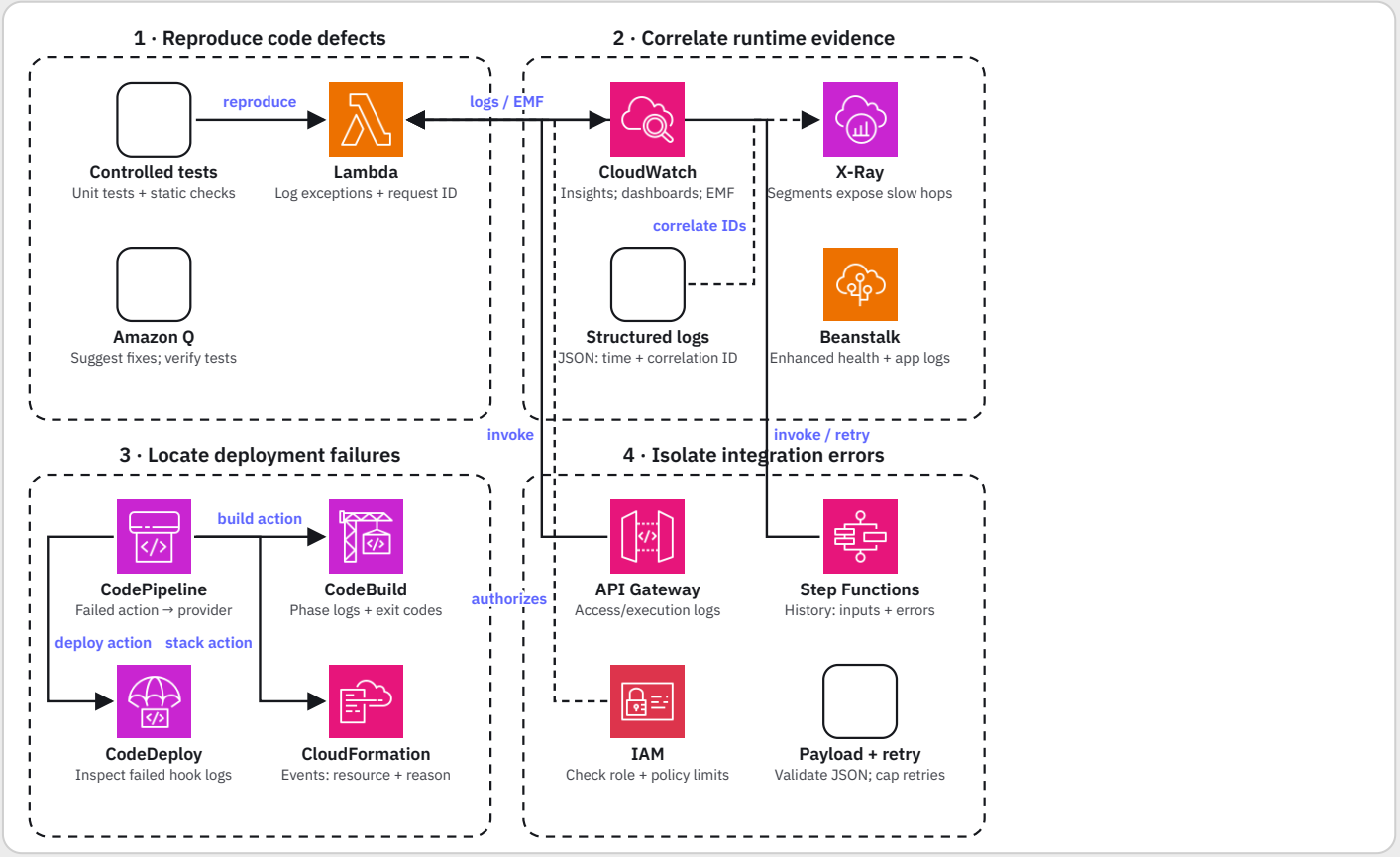
DVA-C02 · Domain 4 of 4

BlowStack

DOMAIN 4 · TROUBLESHOOTING AND OPTIMIZATION · 18% OF THE EXAM · TASK STATEMENT 4.1

Assist in a root cause analysis

Reproduce, correlate, inspect, isolate



1 Reproduce code defects

Use **unit tests** to reproduce defects with controlled inputs; static analysis checks code without executing it. Catch expected exceptions and log context; Amazon Q suggestions still require verification.

2 Correlate runtime evidence

Use **Logs Insights** for queries and regex filters; correlate timestamps and IDs with AWS X-Ray traces. Dashboards and anomaly detection reveal trends; enhanced health exposes unhealthy instances.

3 Locate deployment failures

Follow the **failed action** to provider output: build phase logs, deployment lifecycle hooks, or AWS CloudFormation events. Exit code 0 means success; a nonzero code normally fails the command or hook.

4 Isolate integration errors

For **integration errors**, inspect serialization and workflow history. Retry throttling or transient failures with exponential backoff and jitter; retries do not fix malformed payloads or denied access.

Exam traps

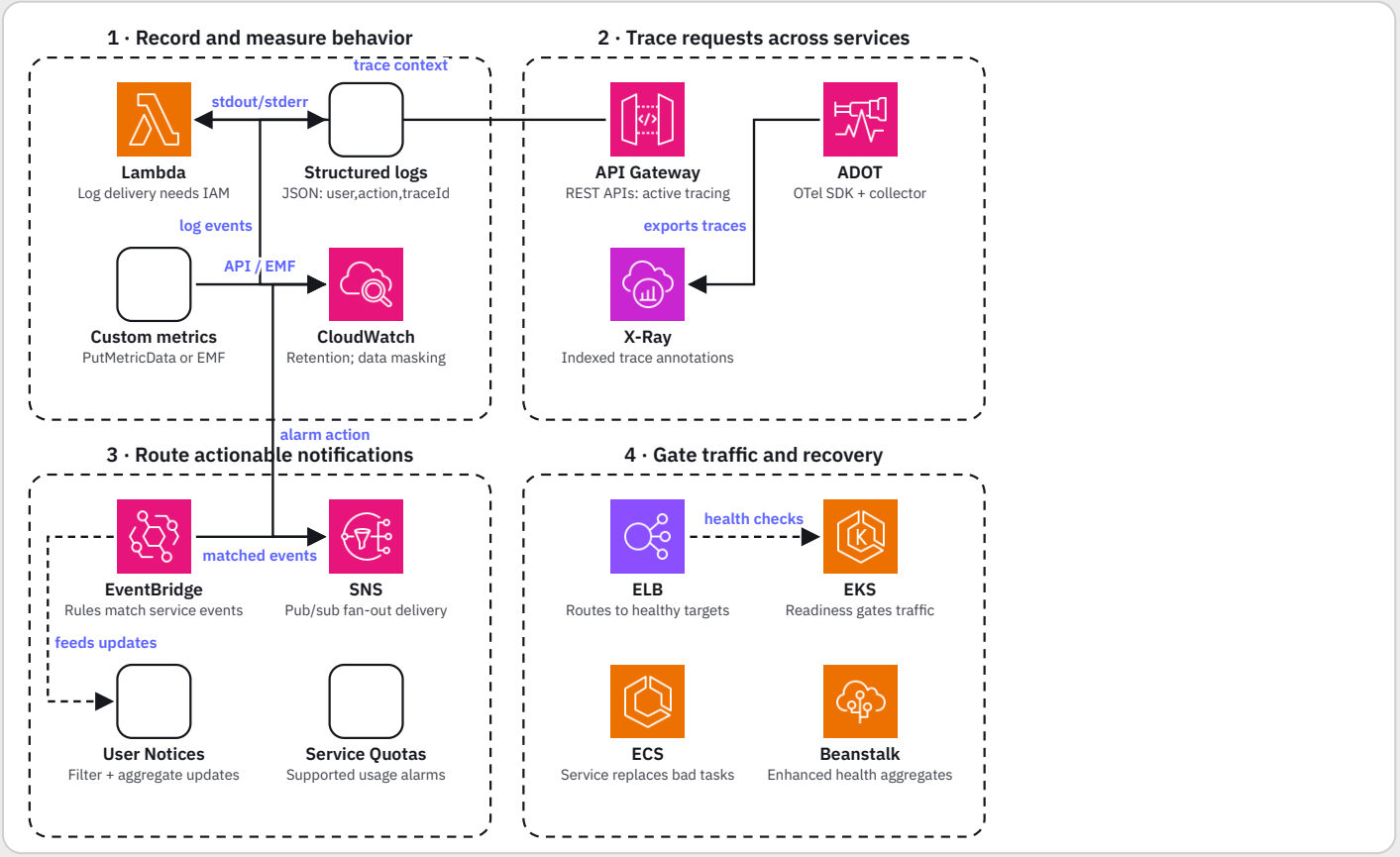
- **EMF requires its schema**, not arbitrary JSON; CloudWatch extracts metrics from valid EMF logs. Request IDs as dimensions create high cardinality and cost; keep them in log fields.
- **An Allow is not sufficient**: explicit denies override it, and boundaries or organizational controls can limit access. Check the execution role and resource policy, not only the developer's permissions.
- **Choose the log store**: Logs Insights queries CloudWatch logs; Amazon Athena queries logs in S3; Amazon OpenSearch Service searches indexed logs. Logs are not automatically present in all three.

Get practice exams and lessons for AWS Certified Developer - Associate

DOMAIN 4 · TROUBLESHOOTING AND OPTIMIZATION · TASK STATEMENT 4.2

Instrument code for observability

Record, trace, alert, check



1 Record and measure behavior

Logs record events; monitoring tracks SLIs such as latency and success rate; observability investigates causes across telemetry. Use **JSON** and correlation IDs; configure levels and retention. EMF extracts metrics from logs.

2 Trace requests across services

Use **annotations** for indexed X-Ray filters; metadata is not indexed. Propagate trace context across service calls. Enable active tracing for API Gateway REST APIs and Lambda. Sampling limits captured requests.

3 Route actionable notifications

Use **EventBridge** for deployment/state-change events; CloudWatch alarms for metric thresholds. SNS fans out to subscribers; AWS User Notifications filters and aggregates AWS events for people.

4 Gate traffic and recovery

Use **readiness** to stop traffic without restarting; liveness failure restarts a container. ELB checks configured targets. ECS container checks require a task-definition healthCheck; image-only checks are not monitored.

Exam traps

- **High-cardinality dimensions** such as userId or requestId create many metric time series, including with EMF. Keep these identifiers in logs or traces; use bounded dimensions for metrics.
- Service Quotas cannot alarm on every quota: **utilization alarms** require supported CloudWatch usage metrics. SNS delivers notifications; it does not detect quota exhaustion.
- **Active tracing** is not full code instrumentation: Lambda runtime tracing does not automatically create downstream-call subsegments. Instrument calls with an SDK or ADOT; sampling omits requests.

Get practice exams and lessons for AWS Certified Developer - Associate

AWS Certified Developer - Associate Cheat Sheet

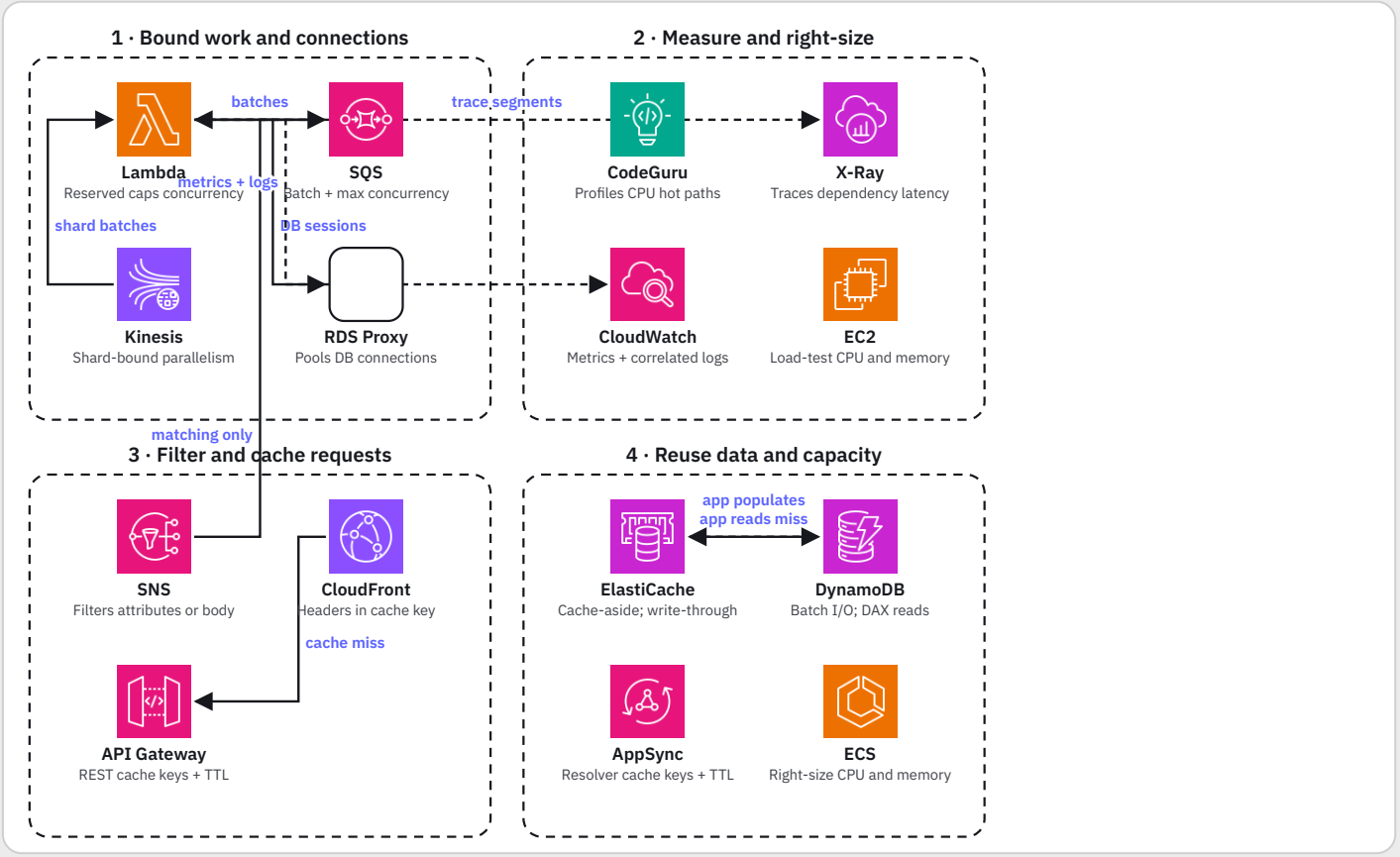
DVA-C02 · Domain 4 of 4

BlowStack

DOMAIN 4 · TROUBLESHOOTING AND OPTIMIZATION · TASK STATEMENT 4.3

Optimize applications by using AWS services and features

Bound, measure, filter, cache



1 Bound work and connections

Use **reserved concurrency** to reserve and cap Lambda capacity; provisioned concurrency reduces cold starts. SQS buffers bursts; cap consumers to protect downstreams. RDS Proxy reduces connection overhead.

2 Measure and right-size

Use **load tests** to pick the smallest EC2 size or Lambda memory meeting latency/throughput targets; Lambda CPU scales with memory. Correlate structured logs by time/request ID; profile hot code, trace slow hops.

3 Filter and cache requests

Use **filter policies** for SNS content routing, not consumer-side discards. Cache varying headers in CloudFront/API Gateway keys; TTL trades freshness for hits. Forwarding a header alone does not vary the cache.

4 Reuse data and capacity

Use **ElastiCache** for application-managed caching; DAX for DynamoDB eventual reads; AppSync for resolver results. Cache-aside fills on misses; write-through adds write work. Right-size ECS CPU/memory under load.

Exam traps

- **Concurrency is not request rate**: Lambda concurrency ≈ requests/second × duration in seconds. Reserved concurrency caps simultaneous executions, not a fixed requests/second rate.
- **Cache-aside is not automatic**: the application loads and populates on a miss; write-through updates the cache on writes. TTL limits staleness but does not guarantee immediate invalidation.
- **DAX is not strongly consistent**: DAX caches eventually consistent DynamoDB reads; strongly consistent reads pass through to DynamoDB and do not populate the cache.

Get practice exams and lessons for AWS Certified Developer - Associate

DOMAIN 4 · TROUBLESHOOTING AND OPTIMIZATION · WHAT TO REMEMBER

Concepts and services of Troubleshooting and Optimization

Concepts to know

Distributed Tracing

Traces connect operations across services for one request; **use traces to locate slow downstream calls**, metrics to detect trends, and logs for event details. Sampling controls trace volume but can leave requests unrecorded.

Structured Logging

Structured logs store named fields such as severity, request ID, and error code; **correlate events with request IDs** rather than timestamps alone. Keep high-cardinality identifiers in logs instead of custom metric dimensions.

Concurrency

Concurrency is the number of in-flight operations, not requests per second. For steady traffic, concurrency approximates request rate multiplied by average duration; **limit concurrency to protect downstream capacity** and buffer excess work when appropriate.

Core services



Amazon CloudWatch

Use **Logs Insights to query application logs**, embedded metric format to extract custom metrics from logs, and metric alarms for threshold notifications; dashboards summarize operational health.



AWS X-Ray

Use **traces and subsegments to isolate downstream latency** and service maps to locate failures. Annotations are indexed for filtering; metadata stores additional context without indexing.



AWS Lambda

More memory also increases CPU allocation; benchmark duration and cost. **Reserved concurrency caps execution**; provisioned concurrency preinitializes environments to reduce cold-start latency.

Get practice exams and lessons for AWS Certified Developer - Associate