AWS Certified

Developer Associate

DVA-C02

QUICK REFERENCE CHEAT SHEET

65 + 20	130 min	720/1000	4 Domains	Associate Level
Scored + Unscored	Duration	Passing Score	Exam Coverage	Prerequisites

32%	26%	24%	18%
Domain 1 · Development	Domain 2 · Security	Domain 3 · Deployment	Domain 4 · Optimization

DOMAIN 1 · DEVELOPMENT WITH AWS SERVICES (32%)

AWS Lambda — Core Concepts

Execution Model	Handler(event, context) is entry point. Code outside handler runs once on init — place SDK clients, DB connections here for warm reuse. Max timeout: **15 min**. Memory: 128 MB–10,240 MB; CPU scales proportionally.
Invocation Types	**Synchronous:** API GW, ALB, Cognito — caller waits for response; errors returned directly. **Asynchronous:** S3, SNS, EventBridge — Lambda retries 2x on failure; configure DLQ (SQS/SNS) for lost events. **Poll-based:** SQS, Kinesis, DynamoDB Streams — Lambda polls via Event Source Mapping.
Concurrency	**Reserved Concurrency:** caps function max; prevents one function consuming all; guarantees capacity. Returns 429 when exceeded. **Provisioned Concurrency:** pre-initializes environments; eliminates cold starts; higher cost. Default account limit: 1,000.
Layers	Shared libraries/dependencies across functions. Max **5 layers** per function; 250 MB unzipped total. Versioned and immutable.
Versions & Aliases	**Versions:** immutable snapshots; $LATEST is mutable. **Aliases:** named pointers supporting weighted traffic splitting (canary %, linear %) for blue/green with CodeDeploy.
Cold Start Mitigation	Provisioned Concurrency (pre-warm), SnapStart (Java init snapshot), minimize package size, move heavy init outside handler, use smaller runtimes.
VPC & Internet	Lambda in VPC: add NAT Gateway in public subnet + route from private subnet -> NAT GW for internet access. Use VPC Endpoints for AWS services to avoid NAT cost.

API Gateway

Type	Cost	Key Features	Best For
REST API	Standard	Usage plans, API keys, WAF, caching, mapping templates (VTL), Lambda/Cognito/IAM auth	Enterprise APIs, monetization, transformation
HTTP API	~70% cheaper	OIDC/OAuth2 native, JWT authorizer, lower latency, no caching/usage plans	Simple Lambda or HTTP backends, cost-first
WebSocket API	Per message	Persistent connections, server-push via callback URL, $connect/$disconnect/$default routes	Real-time: chat, dashboards, collaborative apps

Integration Types	**AWS_PROXY (Lambda proxy):** passes full request; Lambda must return {statusCode, headers, body:string}. **HTTP_PROXY:** pass-through. **MOCK:** static response for testing. **AWS:** custom with VTL mapping templates.
Auth Methods	**IAM:** Sigv4 signed requests — internal/service-to-service. **Cognito Authorizer:** auto-validates User Pool JWT. **Lambda Authorizer (TOKEN):** bearer token, custom logic, cacheable. **Lambda Authorizer (REQUEST):** full request context.
Errors & Throttling	429 = throttled (10,000 req/s default, 5,000 burst). 502 = Lambda returned invalid response format. 504 = 29-second integration timeout. Enable caching (0.5–237 GB) to reduce backend calls.

DynamoDB

Concept	Detail
PK Types	Simple (Hash only) or Composite (Hash + Sort Key). PK -> partition. SK -> range Query within partition.
LSI	Same PK, different SK. Define at creation only. Strongly consistent reads supported. Max 5.
GSI	Different PK+SK. Create/delete anytime. Eventually consistent only. Own capacity. Max 20. Use for alternate access patterns.
RCU	1 strongly consistent read OR 2 eventually consistent reads of <=4 KB item/second.
WCU	1 write of <=1 KB/second. Transactional write = 2× WCU.
On-Demand	Per-request billing; instant scale; ~2.5x more expensive than provisioned. Ideal for spiky/unknown load.

Feature	Detail
Streams	Item-level change log, 24h retention. View types: KEYS_ONLY, NEW_IMAGE, OLD_IMAGE, NEW_AND_OLD_IMAGES. Triggers Lambda.
DAX	In-memory cache; microsecond reads; API-compatible. Use for read-heavy or hot-key workloads.
TTL	Mark attribute as epoch expiry. DynamoDB auto-deletes at no cost. No WCU consumed.
Transactions	TransactWriteItems / TransactGetItems: ACID across <=25 items. Costs 2× RCU/WCU.
Cond. Writes	ConditionExpression: attribute_not_exists(pk) prevents overwrite. Optimistic locking with version attribute.
Hot Partition	Symptom: ProvisionedThroughputExceededException despite adequate table capacity. Fix: redesign PK for even distribution; use write sharding.

Messaging: SQS · SNS · EventBridge · Kinesis · Step Functions

Service	Delivery	Ordering	Throughput	Key Detail
SQS Standard	At-least-once	Best-effort	Unlimited	Visibility timeout default 30s — set to >=6× Lambda timeout. Long polling: WaitTimeSeconds=20. DLQ after maxReceiveCount failures.
SQS FIFO	Exactly-once	Strict per GroupID	3,000/s batched	MessageGroupId for ordering groups. Deduplication ID or Content-Based Dedup. Name ends .fifo.
SNS	Push (all subs)	N/A	Very high	Fan-out: SNS -> multiple SQS queues. Filter policies per subscription. Targets: SQS, Lambda, HTTP, Email, SMS, Kinesis Firehose.
EventBridge	Event bus rules	N/A	High	Rules with event patterns -> targets. Scheduled rules (rate/cron). Archive & Replay. Schema Registry.
Kinesis Streams	At-least-once	Per shard	1 MB/s in per shard	Multiple consumers maintain independent positions. Enhanced Fan-Out: 2 MB/s push per consumer. 24h–365d retention.
Step Functions	Exactly-once (Std)	Workflow	High	Standard: 1yr, full audit. Express: 5min, high-throughput, cheaper. States: Task, Choice, Wait, Parallel, Map.

Exam Tip SQS + Lambda: set visibility timeout >= 6× Lambda timeout. Use ReportBatchItemFailures to retry only failed messages (partial batch). bisectBatchOnFunctionError isolates poison messages.

S3 for Developers

Presigned URLs	Temp access via caller credentials. Max 7d (user) / 12h (role). Direct browser upload or secure download.
Event Notifications	S3 -> SQS / SNS / Lambda / EventBridge (recommended: filtering, multiple targets, replay).
Multipart Upload	Recommended >100 MB; required >5 GB. Parts min 5 MB. Parallel upload. Must call CompleteMultipartUpload.
Encryption	SSE-S3 (AES256), SSE-KMS (CMK, CloudTrail), SSE-C (customer key + HTTPS), Client-Side.
S3 Select	SQL on CSV/JSON/Parquet in S3. Returns subset — up to 400% faster, 80% cheaper.
CORS	Required for browser cross-origin requests. Configure AllowedOrigins, AllowedMethods on bucket.

DOMAIN 2 · SECURITY (26%)

IAM & STS for Application Developers

App Credential Pattern	EC2: Instance Profile -> metadata endpoint 169.254.169.254. Lambda: Execution Role. ECS: Task Role (app calls) + Task Execution Role (agent: ECR pull, CloudWatch logs). **Never hardcode credentials.**
STS AssumeRole	Returns temporary credentials: AccessKeyId, SecretAccessKey, SessionToken, Expiration (15min–12h). Use for cross-account and cross-service access.

STS Variants	**AssumeRoleWithWebIdentity:** OIDC (prefer Cognito). **AssumeRoleWithSAML:** enterprise ADFS/SAML. **GetSessionToken:** MFA-based temporary creds for IAM users.
Policy Evaluation	**Explicit Deny wins always** -> Allow -> Implicit Deny. SCPs restrict maximum permissions (org-wide). Permission Boundaries set max for entity (do not grant access).
Resource-Based Policies	S3 bucket policy, SQS queue policy, Lambda resource policy, KMS key policy. Grant cross-account access by specifying external Principal. Cross-account: both identity-based AND resource-based policies must allow.

Amazon Cognito

Component	What It Does	Returns	Use Case
User Pools	User directory: sign-up, sign-in, MFA, social federation (Google/Facebook), Hosted UI, Lambda triggers	JWT: ID token, Access token, Refresh token	Authentication for web/mobile apps
Identity Pools	Exchange any identity token (User Pool JWT, OIDC, SAML) for AWS temporary credentials via STS	Temporary IAM credentials (AccessKey + SessionToken)	Grant users direct AWS API access (S3, DynamoDB)
Pre-Token Gen Lambda	Hook fires before token creation; add/override claims in JWT	Modified token with custom claims	Add subscription tier, department, org to tokens
Cognito Authorizer (API GW)	API Gateway validates JWT Access token signature and expiry automatically against User Pool	Allow/Deny to API method	Protect REST or HTTP API with Cognito auth

Exam Tip S3 per-user prefix: Identity Pool IAM role policy uses ${cognito-identity.amazonaws.com:sub} policy variable -> each user can only access their own S3 prefix automatically.

KMS · Secrets Manager · SSM Parameter Store

Service	Max Size	Auto Rotation	Cost	Best For
KMS CMK	4 KB direct encrypt; use envelope encryption for larger data	Annual (AWS Managed Keys)	Per key/month + API calls	Encryption key management; all KMS calls logged in CloudTrail
Secrets Manager	64 KB	Yes — native for RDS/Redshift; custom Lambda for others	$0.40/secret/month + $0.05/10K API	Rotating credentials: DB passwords, API keys, OAuth tokens
SSM Parameter Store (Std)	4 KB; SecureString via KMS	No native rotation	Free (standard)	Non-rotating config: feature flags, endpoints, environment values
SSM Parameter Store (Adv)	8 KB; TTL policies	No native rotation	Paid	Advanced config with expiration policies

Envelope Encryption	**Step 1:** Call KMS GenerateDataKey -> receive plaintext DEK + encrypted DEK. **Step 2:** Encrypt data locally with plaintext DEK (never send large data to KMS). **Step 3:** Store encrypted DEK with encrypted data. **Decrypt:** KMS Decrypt(encryptedDEK) -> plaintext DEK -> decrypt data locally.
CFN Dynamic References	{{resolve:secretsmanager:secret-name}} or {{resolve:ssm-secure:/path/to/param}} — value fetched at deploy time; never stored in plaintext in template. Use NoEcho:true for Parameter values.

DOMAIN 3 · DEPLOYMENT (24%)

CI/CD Developer Tools Pipeline

Service	Role	Config File	Key Detail
CodeCommit	Source: private Git repo	—	Triggers pipeline via CloudWatch Events/EventBridge on push.
CodeBuild	Build: compile, test, package	buildspec.yml	Phases: install -> pre_build -> build -> post_build. artifacts: files for next stage. cache: S3-backed for node_modules/.m2.
CodeDeploy	Deploy: EC2 / Lambda / ECS	appspec.yml	EC2: in-place or blue/green. Lambda: canary/linear traffic shifting on aliases. ECS: blue/green task sets. Auto rollback on CloudWatch Alarm.

CodePipeline	Orchestrate: stages -> actions	Pipeline JSON	Source -> Build -> Deploy. Manual Approval action: pauses + sends SNS. Supports GitHub/Bitbucket via CodeStar Connections.
CodeArtifact	Artifact repository	upstream config	Managed npm, Maven, PyPI, NuGet proxy + private storage. Integrates with CodeBuild for secure dependency management.

CodeDeploy Strategies	**EC2 In-Place:** stop -> deploy -> restart (possible downtime). **EC2/ECS Blue/Green:** new fleet/task set, shift traffic, rollback by pointing back. **Lambda Canary:** X% for N minutes then 100% (e.g., Canary10Percent5Minutes). **Lambda Linear:** X% more every N minutes. Automatic rollback: associate CloudWatch Alarm with deployment group.
appspec.yml Hooks	**EC2:** BeforeInstall, AfterInstall, ApplicationStart, ApplicationStop, **ValidateService** (run health checks here). **Lambda:** BeforeAllowTraffic, AfterAllowTraffic. **ECS:** BeforeInstallHook, AfterInstallHook, AfterAllowTestTrafficHook, AfterAllowTrafficHook.

SAM · CloudFormation · CDK

SAM Resource / CLI	Purpose
AWS::Serverless::Function	Lambda + auto execution role + event sources
AWS::Serverless::Api	API Gateway REST API inline in template
AWS::Serverless::SimpleTable	DynamoDB table (single hash key)
sam local invoke	Test Lambda locally in Docker container
sam local start-api	Local API GW emulation for REST testing
sam build --use-container	Build with matching Lambda runtime Docker image
sam deploy --guided	Interactive deploy: packages, uploads, applies stack

CFN Feature	Use
Conditions + Fn::If	Conditional resources/properties based on parameters. IsProd: !Equals [!Ref Env, prod]
Change Sets	Preview ADD/MODIFY/REMOVE before applying. Essential for production pipelines.
Drift Detection	Find manual changes made outside CloudFormation.
DeletionPolicy: Retain	Keep S3/RDS on stack deletion — resource stays, removed from CFN state.
CFN Custom Resources	Lambda-backed: handle unsupported resources; run DB migrations; call external APIs.
StackSets	Deploy across multiple accounts + regions from single template.
CDK	Infrastructure as code in TypeScript/Python. Synthesizes to CFN. L1/L2/L3 constructs.

Elastic Beanstalk Deployment Policies

Policy	Downtime	Capacity During Deploy	Rollback	Best For
All at Once	Yes (brief)	Reduced to 0 briefly	Redeploy old version	Dev/test — speed over availability
Rolling	No	Reduced (batch size out)	Redeploy	Gradual update with some capacity cost
Rolling + Additional Batch	No	Full (extra instances spun)	Redeploy	Full capacity needed during update
Immutable	No	Full (new ASG created)	Terminate new ASG instantly	Production — safest, most expensive
Blue/Green	No	Full (two environments)	Swap URL back in seconds	Zero-risk with instant DNS rollback

ECS Task Roles	**Task Role (taskRoleArn):** what the application container can DO (S3 read, DynamoDB write). **Task Execution Role (executionRoleArn):** ECS agent only — pull ECR images, write CloudWatch Logs, fetch secrets from Secrets Manager/SSM. Both are required; different purposes.
.ebextensions	YAML/JSON .config files in .ebextensions/ folder. Install packages, run shell commands, create files, set environment variables. Applied during EC2 instance provisioning.

DOMAIN 4 · TROUBLESHOOTING & OPTIMIZATION (18%)

CloudWatch

Metrics	Default: EC2 every 5 min; detailed monitoring every 1 min. Lambda: Duration, Errors, Throttles, ConcurrentExecutions, DeadLetterErrors. Custom metrics: PutMetricData API with custom Namespace.
Logs	Lambda auto-creates /aws/lambda/{name}. Execution role must have logs:CreateLogGroup + CreateLogStream + PutLogEvents (AWSLambdaBasicExecutionRole). Log Insights: SQL-like queries. Metric Filters: create metrics from log patterns. Subscriptions: stream to Lambda/Kinesis in real time.
Alarms & States	States: OK / ALARM / INSUFFICIENT_DATA (no data in evaluation period — != error). Composite Alarms: AND/OR logic across multiple alarms; reduce noise. Actions: SNS, EC2 actions, ASG scaling. Configure on Lambda error rate for CodeDeploy auto-rollback.
EventBridge Schedules	Rate: rate(5 minutes), rate(1 hour). Cron: cron(minute hour dom month dow year). Daily at 8 AM UTC: cron(0 8 * * ? *). Weekdays noon: cron(0 12 ? * MON-FRI *). ? required in dom or dow.

AWS X-Ray

Trace	End-to-end request across services. Unique Trace ID propagated via X-Amzn-Trace-Id header.
Segment	Work done by a single service or process.
Subsegment	Granular breakdown: specific DDB call, HTTP request, SQL query.
Sampling	Default: 1 req/sec + 5% additional. Configure rules for critical paths.
Daemon	Receives traces on UDP port 2000. Batches + sends to X-Ray API. Required on EC2/ECS.

Lambda	Enable Active Tracing in config. AWSXRayDaemonWriteAccess policy on execution role. SDK auto-instruments AWS SDK calls.
API Gateway	Enable X-Ray Tracing on stage — propagates Trace ID to Lambda downstream.
ECS	X-Ray daemon as sidecar container or DaemonSet. SDK in application code.
Service Map	Visual graph: services, latency percentiles, error/fault/throttle rates. Identify slow downstream dependencies.

Performance Optimization Patterns

Pattern	Service	Implementation	Key Benefit
Eliminate cold starts	Lambda	Provisioned Concurrency (pre-warms envs); SnapStart (Java post-init snapshot)	Sub-100ms consistent response time
DB connection pooling	Lambda + RDS	RDS Proxy: pools Lambda connections; presents small pool to RDS	Prevents max_connections exhaustion at high concurrency
Read caching	DynamoDB	DAX cluster (API-compatible; microsecond reads); or ElastiCache Redis/Memcached for app-level cache	Reduce DynamoDB read costs; eliminate hot-partition reads
Session externalization	EC2 / ECS	Store sessions in ElastiCache Redis instead of in-memory on instances	Auto Scaling scale-in without losing active user sessions
Cache-aside pattern	ElastiCache	Check cache -> hit: return; miss: query DB -> write to cache -> return	Reduce DB load; only cache what is requested
SQS buffer	Lambda / API GW	API GW -> SQS -> Lambda: absorbs traffic spikes; Lambda processes at its own rate	Prevent Lambda throttling on sudden write bursts
Async offloading	Lambda	Return 200 immediately; put work on SQS/SNS; process asynchronously	Reduce API response time for long-running operations
Lambda memory tuning	Lambda	Increasing memory increases CPU proportionally. Lambda Power Tuning tool finds optimal setting.	CPU-bound functions: doubling memory can halve duration at same cost

Caching Services Comparison

Service	Protocol	Persistence	Advanced Features	Use Cases
ElastiCache Redis	Redis	Yes (RDB/AOF)	Sorted Sets, pub/sub, Lua, cluster mode, multi-AZ replication, Streams	Leaderboards, session store, pub/sub, rate limiting
ElastiCache Memcached	Memcached	No	Multi-threaded, horizontal sharding only	Simple key-value cache, stateless high-throughput caching
DAX	DynamoDB API	No (read cache)	API-compatible, write-through, item + query cache	DynamoDB read acceleration (microsecond latency)
API GW Cache	HTTP	No	Per-stage; invalidate via Cache-Control: max-age=0	Reduce Lambda invocations for stable API responses

CloudFront	HTTP/HTTPS	No (edge TTL)	Cache behaviors per path, Lambda@Edge, OAC for S3	Global content delivery; reduce origin load + egress cost

MASTER QUICK REFERENCE — KNOW THESE COLD

Service / Concept	Category	Remember This
Lambda Invocation Types	Serverless	Sync: API GW/ALB (caller waits). Async: S3/SNS (retry 2x, DLQ). Poll: SQS/Kinesis (Event Source Mapping).
Lambda Concurrency	Serverless	Reserved = cap + protect. Provisioned = pre-warm, no cold start. Default: 1,000/account.
Lambda Versions & Aliases	Serverless	Versions = immutable. Aliases = named pointers. Traffic shifting for canary/blue-green deployments.
API Gateway REST vs HTTP	API	REST: usage plans, API keys, WAF, caching. HTTP: 70% cheaper, JWT auth, no usage plans.
API Gateway 502 Error	API	Lambda returned malformed response. Must return {statusCode:int, headers:{}, body:string}.
DynamoDB GSI vs LSI	Database	LSI: same PK, diff SK, at creation, strong consistency. GSI: diff PK+SK, anytime, eventually consistent only.
DynamoDB Streams	Database	Item-level change log, 24h retention. View types: KEYS_ONLY / NEW_IMAGE / OLD_IMAGE / NEW_AND_OLD_IMAGES.
DynamoDB Hot Partition	Database	Symptom: throttling despite adequate capacity. Fix: redesign PK for even distribution; write sharding.
SQS Visibility Timeout	Messaging	Set to >= 6× Lambda timeout. Prevents message reappearing while Lambda is still processing.
SQS FIFO Queue	Messaging	Exactly-once, strict ordering per MessageGroupId. Max 3,000/s batched. Ends in .fifo.
SNS Fan-Out	Messaging	SNS -> multiple SQS queues. Each consumer independent. Filter policies per subscription.
EventBridge Cron	Events	cron(minute hour dom month dow year). Daily 8 AM UTC: cron(0 8 * * ? *). ? required in dom or dow.
Cognito User Pools	Auth	Authentication: sign-up/sign-in, returns JWT (ID, Access, Refresh). Social federation, MFA, Hosted UI.
Cognito Identity Pools	Auth	Authorization: exchange JWT for AWS temp credentials. Use for direct S3/DynamoDB access.
KMS Envelope Encryption	Security	GenerateDataKey -> encrypt data with plaintext DEK locally -> store encrypted DEK. KMS max 4 KB direct.
Secrets Manager vs SSM	Security	Secrets Manager: auto-rotation, $0.40/secret. SSM: free, no native rotation. Use Secrets Manager for DB passwords.
CodeDeploy Lambda	CI/CD	Canary/Linear traffic shifting on alias. Auto rollback via CloudWatch Alarm on error rate.
SAM Transform	IaC	Transform: AWS::Serverless-2016-10-31. sam local invoke/start-api for local testing. sam deploy --guided to deploy.
CloudFormation Change Sets	IaC	Preview changes before applying. Use in CI/CD for production review gates.
Beanstalk Immutable	Deploy	New ASG, new instances. Zero downtime. Safest. Instant rollback: terminate new ASG.
ECS Task vs Exec Role	Containers	Task Role: app permissions (S3, DDB). Execution Role: ECS agent (ECR pull, CloudWatch logs).
RDS Proxy	Database	Pools Lambda->RDS connections. Prevents max_connections exhaustion at high Lambda concurrency.

X-Ray Active Tracing	Monitoring	Enable on Lambda + API GW. AWSXRayDaemonWriteAccess policy. SDK instruments AWS SDK calls automatically.
CloudWatch INSUFFICIENT_DATA	Monitoring	Not an error — means no metric data in evaluation period (function not invoked, no data points to evaluate).
SQS ReportBatchItemFailures	Messaging	Return {batchItemFailures:[{itemIdentifier:messageId}]} to retry ONLY failed messages, not entire batch.

- 720/1000 to pass | 65 scored + 20 unscored | 130 minutes | No penalty for guessing — answer every question!
- Lambda cold starts -> Provisioned Concurrency | Share dependencies -> Lambda Layers | Safe deployment -> CodeDeploy canary/linear with CloudWatch Alarm
- API GW 429 = throttled | 502 = malformed Lambda response (check statusCode/body format) | 504 = 29s timeout exceeded
- DynamoDB: Query (needs PK) vs Scan (reads everything — avoid) | Hot partition -> redesign PK | Transactions -> TransactWriteItems (2× WCU)
- SQS visibility timeout >= 6× Lambda timeout | FIFO = exactly-once ordered | Use ReportBatchItemFailures for partial retry
- Cognito User Pools = authentication (JWT) | Identity Pools = authorization (AWS temp credentials) | Use ${cognito-identity.amazonaws.com:sub} for per-user S3
- KMS 4 KB limit -> envelope encryption (GenerateDataKey) | Secrets Manager = auto-rotation | SSM = free non-rotating config
- CodeDeploy Lambda = alias traffic shifting + CloudWatch Alarm auto-rollback | ValidateService hook = run health checks after deploy
- SAM local invoke/start-api for local testing | CFN Change Sets for production review | DeletionPolicy: Retain to keep S3 on stack delete
- RDS Proxy = pools Lambda->RDS connections | ElastiCache Redis = sessions + leaderboards + pub/sub | DAX = microsecond DynamoDB reads