AWS Certified Solutions Architect - Associate Cheat Sheet

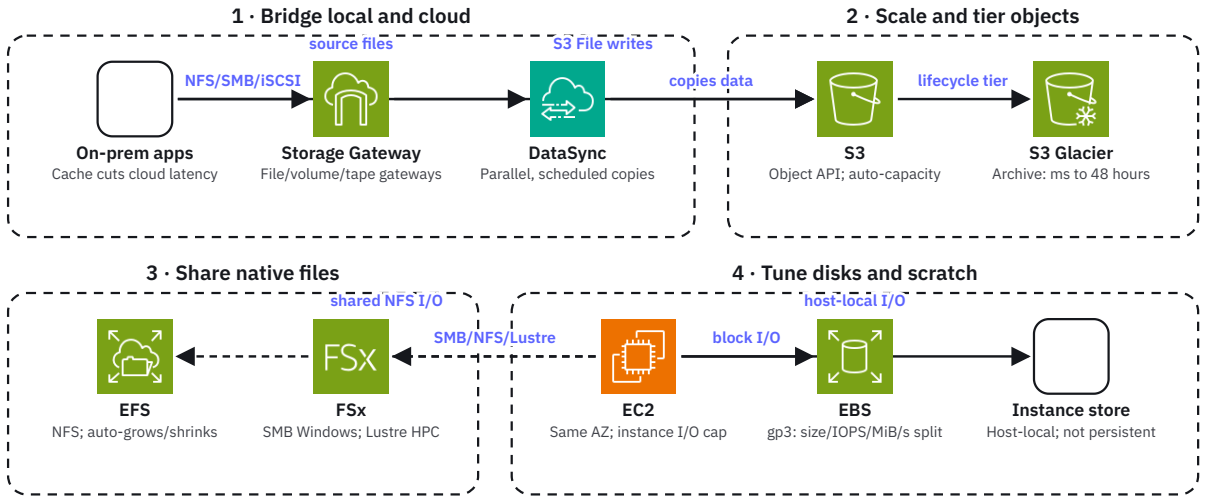
SAA-C03 · Domain 3 of 4

BlowStack

DOMAIN 3 · DESIGN HIGH-PERFORMING ARCHITECTURES · 24% OF THE EXAM · TASK STATEMENT 3.1

Determine high-performing and/or scalable storage solutions

Bridge, tier, share, tune



1 Bridge local and cloud

Ongoing hybrid access favors Storage Gateway with a local cache; DataSync migrates or synchronizes datasets rather than serving live application I/O. Size the cache for the working set.

2 Scale and tier objects

Object access fits backups, media, and data lakes. S3 capacity scales automatically; parallel requests and multipart uploads improve throughput. Glacier retrieval speed depends on the storage class and retrieval tier.

3 Share native files

File protocol drives selection: EFS supplies elastic shared NFS; FSx for Windows File Server supplies SMB; FSx for Lustre suits HPC. EFS Elastic throughput follows demand; FSx capacity and throughput require planning.

4 Tune disks and scratch

IOPS versus throughput: small random I/O stresses operations/s; large sequential I/O stresses MiB/s. gp3 tunes both independently of size; io2 suits demanding databases. EBS expansion is explicit, not automatic elasticity.

Exam traps

- **S3 is not a shared POSIX file system**. An object-storage mount does not supply full EFS file semantics; applications requiring shared native file operations need a file service.
- **EBS Multi-Attach is not multi-AZ**: supported io1/io2 volumes attach within one AZ and require coordinated access through a cluster-aware application or file system.
- **Instance store is disposable**: data survives reboot but is lost on stop, hibernation, or termination. EBS can persist after termination only when DeleteOnTermination is disabled.

Get practice exams and lessons for AWS Certified Solutions Architect - Associate

AWS Certified Solutions Architect - Associate Cheat Sheet

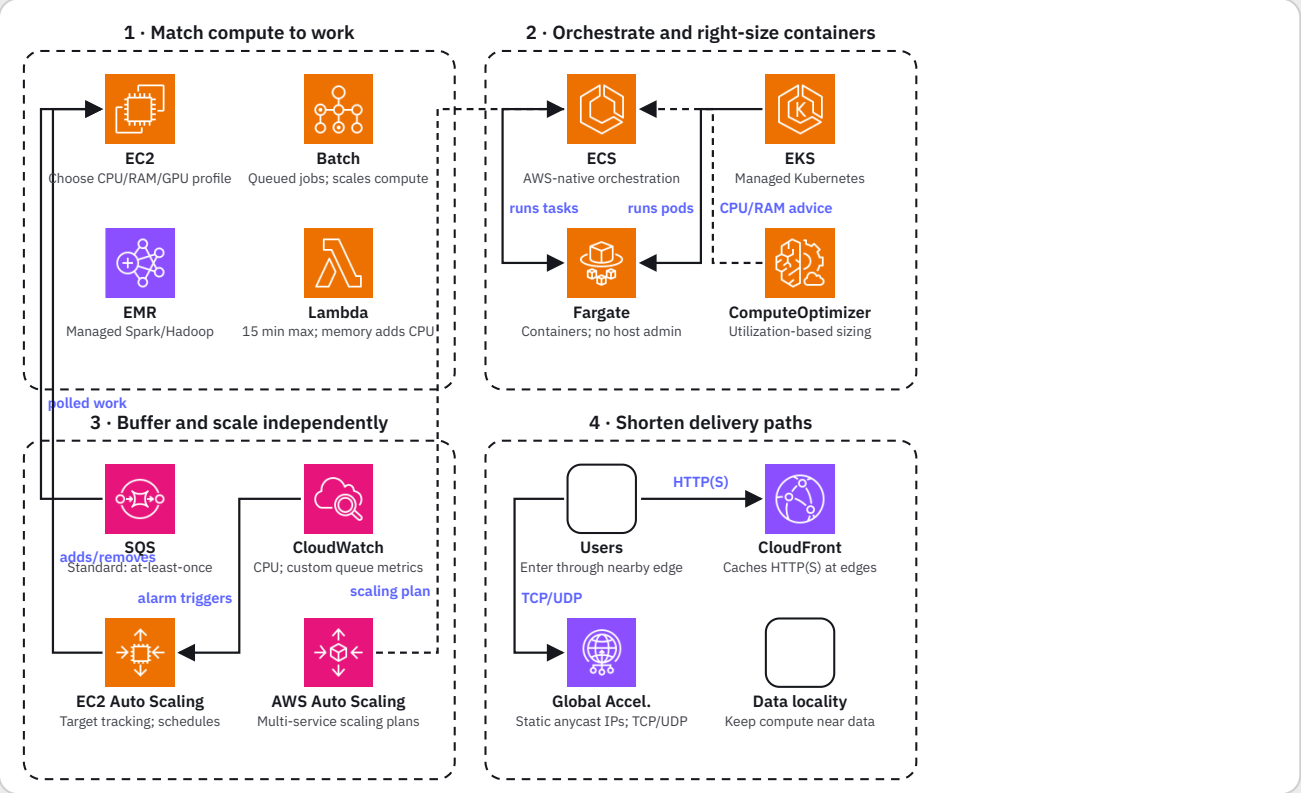
SAA-C03 · Domain 3 of 4

BlowStack

DOMAIN 3 · DESIGN HIGH-PERFORMING ARCHITECTURES · TASK STATEMENT 3.2

Design high-performing and elastic compute solutions

Match, orchestrate, scale, accelerate



1 Match compute to work

Execution model decides: EC2 for OS control, Batch for scheduled or queued batch jobs, EMR for Spark/Hadoop, Lambda for short event-driven code. Lambda scales executions automatically, subject to concurrency quotas.

2 Orchestrate and right-size containers

Orchestration ≠ compute: ECS or EKS schedules containers; Fargate supplies serverless capacity. EC2 capacity retains host management. Compute Optimizer recommends sizes from utilization; it does not apply changes.

3 Buffer and scale independently

Backlog per worker scales queue consumers independently of producers; CPU suits CPU-bound demand. Target tracking adds/removes capacity within bounds. Scheduled scaling anticipates known peaks; warmup delays usable capacity.

4 Shorten delivery paths

Network path decides: CloudFront caches content near users; Global Accelerator routes to healthy regional endpoints over the AWS network without caching. Nearby compute reduces data-transfer latency; multiple AZs improve resilience.

Exam traps

- **One queue is not fan-out**: consumers compete for messages. SNS can publish to separate SQS queues for independent subscribers. Standard SQS consumers need idempotency because delivery can repeat.
- **Horizontal ≠ vertical**: Auto Scaling adds or removes instances; it does not enlarge them. Changing an EC2 instance type is vertical scaling and can require a stop/start.
- **Serverless ≠ unlimited**: Lambda has execution-duration and concurrency limits. Fargate suits long-running containers, but ECS service scaling still needs configured policies and capacity limits.

Get practice exams and lessons for AWS Certified Solutions Architect - Associate

AWS Certified Solutions Architect - Associate Cheat Sheet

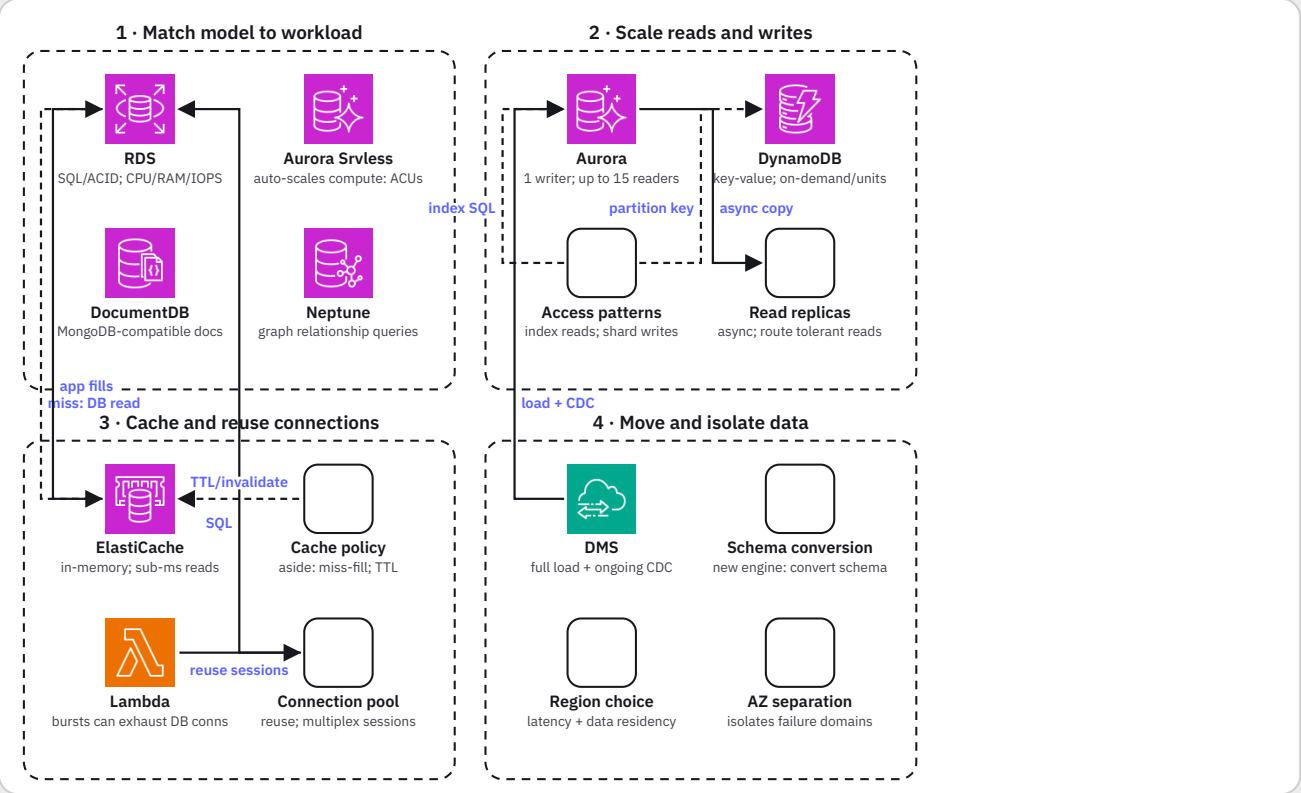
SAA-C03 · Domain 3 of 4

BlowStack

DOMAIN 3 · DESIGN HIGH-PERFORMING ARCHITECTURES · TASK STATEMENT 3.3

Determine high-performing database solutions

Choose, scale, cache, replicate



1 Match model to workload

Joins/ACID favor RDS or Aurora; engine compatibility decides MySQL vs PostgreSQL. **Serverless** Aurora scales compute; documents favor DocumentDB; relationship traversals favor Neptune.

2 Scale reads and writes

Write-heavy DynamoDB needs balanced partition keys: on-demand for unpredictable traffic, provisioned RCU/WCU for planned load. Relational replicas offload reads, not writes; lag-sensitive reads go to the writer.

3 Cache and reuse connections

Cache-aside fills on misses; write-through refreshes on writes, adding latency. TTL/invalidation controls staleness. RDS Proxy pools/multiplexes Lambda connections; session pinning reduces reuse.

4 Move and isolate data

Heterogeneous moves need schema conversion plus data migration; homogeneous moves retain the engine family. DMS CDC reduces cutover downtime. Region choice controls locality; AZ separation limits failure scope.

Exam traps

- **Multi-AZ is not read scaling** for a classic RDS DB instance: its standby provides failover, not query capacity. Read replicas serve reads; Multi-AZ DB clusters have readable instances.
- **Strong reads are not universal** in DynamoDB: tables and local secondary indexes support them; global secondary indexes are eventually consistent. Strong reads consume more read capacity.
- **Indexes and denormalization are not free**: indexes add write/storage overhead; duplicated data needs coordinated updates. Sharding scales aggregate throughput only when keys distribute load.

Get practice exams and lessons for AWS Certified Solutions Architect - Associate

AWS Certified Solutions Architect - Associate Cheat Sheet

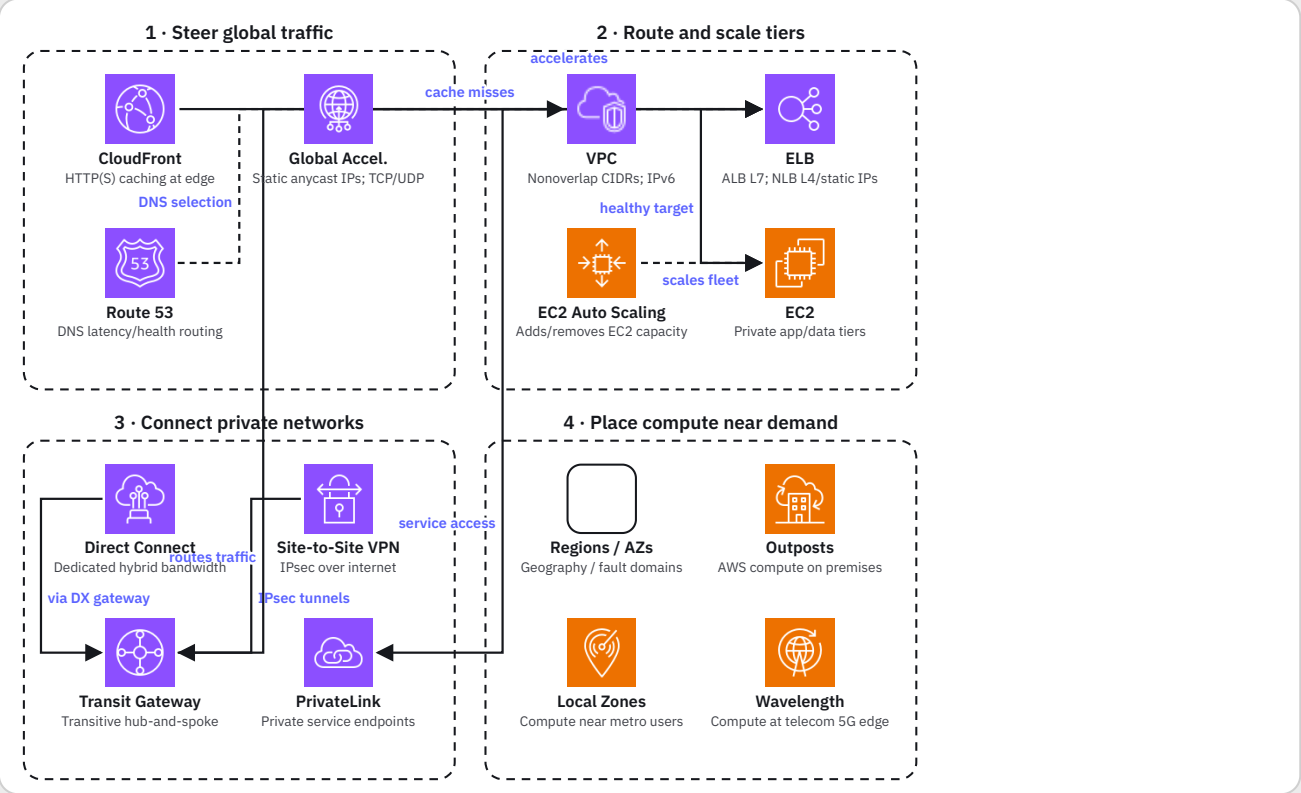
SAA-C03 · Domain 3 of 4

BlowStack

DOMAIN 3 · DESIGN HIGH-PERFORMING ARCHITECTURES · TASK STATEMENT 3.4

Determine high-performing and/or scalable network architectures

Steer, route, connect, place



1 Steer global traffic

Choose **caching versus routing**: CloudFront caches content; Global Accelerator accelerates TCP/UDP over AWS's network; Route 53 selects regional endpoints through DNS. Active-active also requires a data replication strategy.

2 Route and scale tiers

Use **multi-AZ tiers**: public ALB, private apps/data. Subnet route tables select paths; reserve IP capacity for growth. ALB routes by host/path to healthy targets; Auto Scaling replaces failed targets when ELB health checks are enabled.

3 Connect private networks

Choose **network versus service access**: Transit Gateway connects routed networks; PrivateLink exposes specific services privately. Direct Connect gives consistent bandwidth; VPN provides rapid hybrid connectivity or backup.

4 Place compute near demand

Choose **placement by latency and residency**: Outposts for local processing; Local Zones for metro proximity; Wavelength for 5G applications. Multiple AZs isolate failures. Residency also includes replicas, backups, and logs.

Exam traps

- Direct Connect is **not encrypted by default**. Dedicated connectivity does not imply encryption; use supported encryption options or IPsec VPN over Direct Connect when required.
- PrivateLink provides **service access, not transit routing**. It can expose services across overlapping VPC CIDRs; Transit Gateway does not resolve overlapping address spaces.
- ELB **does not add compute capacity**; Auto Scaling does. Sticky sessions can skew load and do not preserve session data when a target fails.

Get practice exams and lessons for AWS Certified Solutions Architect - Associate

AWS Certified Solutions Architect - Associate Cheat Sheet

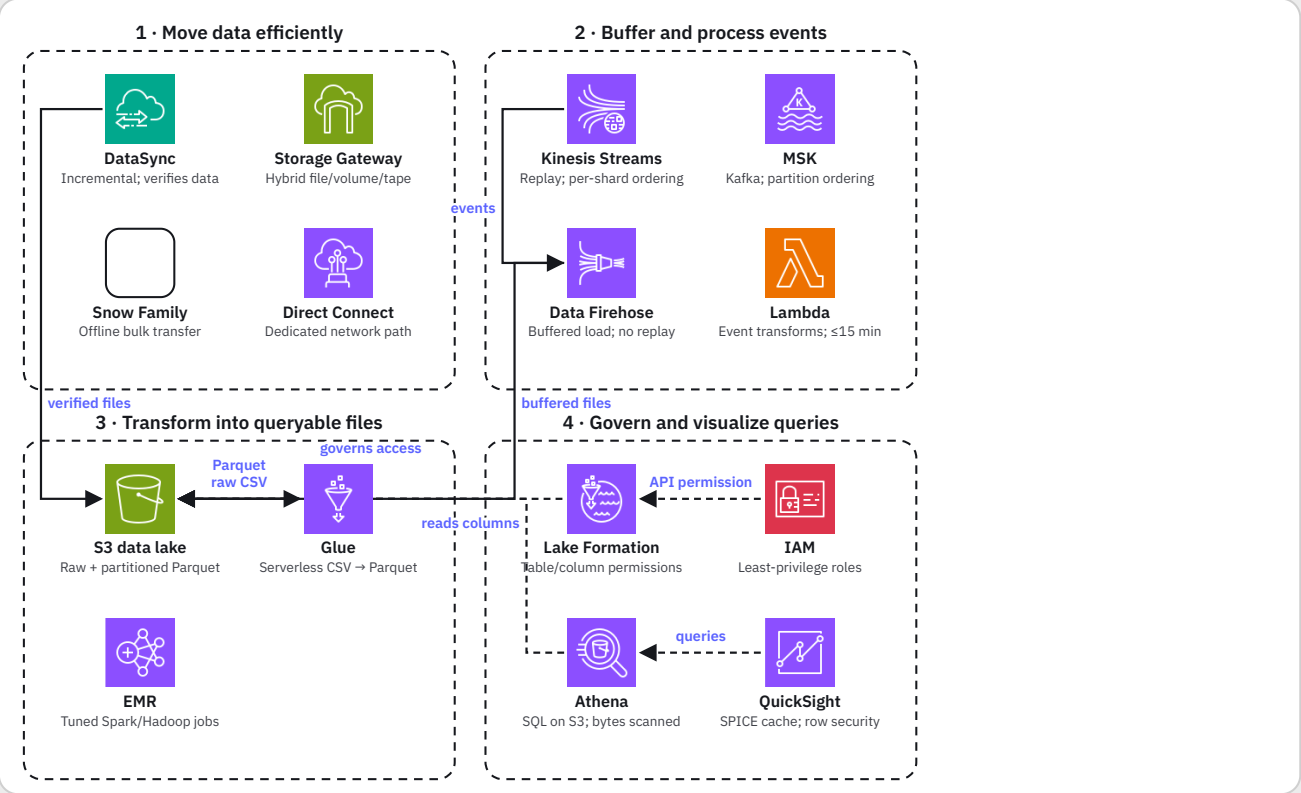
SAA-C03 · Domain 3 of 4

BlowStack

DOMAIN 3 · DESIGN HIGH-PERFORMING ARCHITECTURES · TASK STATEMENT 3.5

Determine high-performing data ingestion and transformation solutions

Transfer, stream, transform, query



1 Move data efficiently

Recurring file sync: DataSync transfers changes and verifies integrity; Storage Gateway provides hybrid storage access. Snow handles offline bulk; Direct Connect supplies connectivity, not a transfer engine.

2 Buffer and process events

Latency: Streams/MSK retain events for independent consumers; Firehose buffers destination delivery. Larger batches favor throughput over freshness. Consumer lag requires capacity, retries, and durable progress tracking.

3 Transform into queryable files

Large ETL: Glue supplies serverless Spark; EMR suits framework control and tuned distributed jobs. Convert CSV to compressed Parquet, partition for query filters, and compact small files.

4 Govern and visualize queries

SQL on S3: Athena queries without warehouse loading; QuickSight caches dashboards in SPICE. Lake Formation governs table access; IAM grants service actions. SPICE refresh schedules determine cached-data freshness.

Exam traps

- **More shards do not fix a hot key**: Distribute partition keys. Amazon Kinesis Data Streams orders within a shard, not globally; each provisioned shard supports 1 MB/s or 1,000 records/s writes.
- **Private is not authorized**: AWS PrivateLink avoids public network paths; IAM and lake permissions still apply. Encryption with AWS Key Management Service (AWS KMS) also requires key permissions.
- **Dashboard is not warehouse**: Amazon QuickSight visualizes; Amazon Athena queries S3; Amazon Redshift warehouses data. SPICE needs refresh; Parquet and partition pruning reduce Athena scans.

Get practice exams and lessons for AWS Certified Solutions Architect - Associate

DOMAIN 3 · DESIGN HIGH-PERFORMING ARCHITECTURES · WHAT TO REMEMBER

Concepts and services of Design High-Performing Architectures

Concepts to know

Horizontal Scaling

Adds workers or replicas rather than enlarging one resource. Choose **independent, stateless workers** for elastic application tiers; queues absorb bursts, while read replicas scale database reads rather than writes.

Caching

Serves reusable data closer to consumers, reducing latency and backend load. Choose **caching for repeated reads** when freshness requirements permit; expiration and invalidation control stale results.

Throughput

Measures work or bytes completed per second, unlike latency or IOPS. Choose **capacity for the actual bottleneck**: small random storage operations stress IOPS, large sequential transfers stress bandwidth, and skewed partitions limit parallelism.

Core services



Amazon EC2

Choose **instance resources that match the bottleneck**: compute-optimized for CPU-heavy processing, memory-optimized for large working sets, and accelerated computing for supported GPU workloads.



Amazon S3

Choose **object storage** for data lakes, static assets, and large datasets—not shared filesystem or block semantics. Multipart uploads and parallel requests improve large-transfer throughput.



Amazon RDS

Choose **read replicas for read scaling** and larger instances for more CPU or memory. A Multi-AZ DB instance standby provides failover, not read capacity; provision storage IOPS for I/O bottlenecks.

Get practice exams and lessons for AWS Certified Solutions Architect - Associate