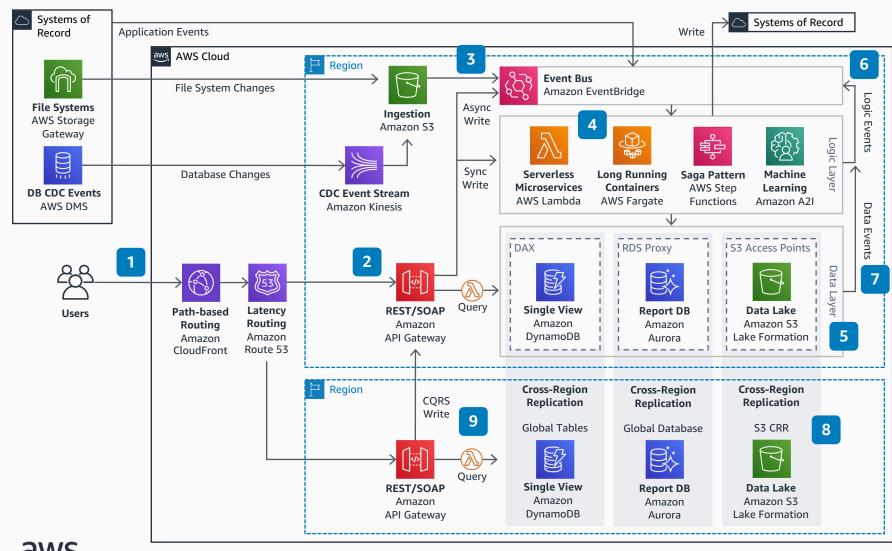
Serverless Architecture for Global Applications

Improve customer experience on your global services by deploying into multiple AWS Regions and applying event-driven architectural patterns to reduce latency and increase performance, such as event sourcing, saga orchestration, and CQRS.



- Route traffic from edge locations based on request path, allowing a gradual migration of single-CNAME legacy API operations. Then route requests to the Region with the least latency for the requester.
- Front each Region with an entry API implementing the CQRS pattern.
 For query requests, read from the data layer. For synchronous write commands, invoke the logic layer.
- Process asynchronous commands by adding them to the event bus. Source additional events from external systems.
- Process transactional logic via serverless functions and run long running tasks with containers. Use the Saga pattern to orchestrate distributed transactions with eventual consistency. Add intelligence via Machine Learning and Augmented AI.
- 5 Store for data access patterns: e.g., key-value store for Single View of Entities, relational for Reporting, and data lake for Analytics and AI/ML.
- Raise logic events from the logic layer to run event-driven workflow steps, following the Transformation pattern.
- Raise data events after changes to the canonical data model on the data layer, reducing redundance in the logic layer.
- Propagate changes across Regions with active-active data replication.
- 9 For heavy reading scenarios on other Regions, send write requests to primary Regions. For fast global writes, replicate the event and command logic on all Regions.