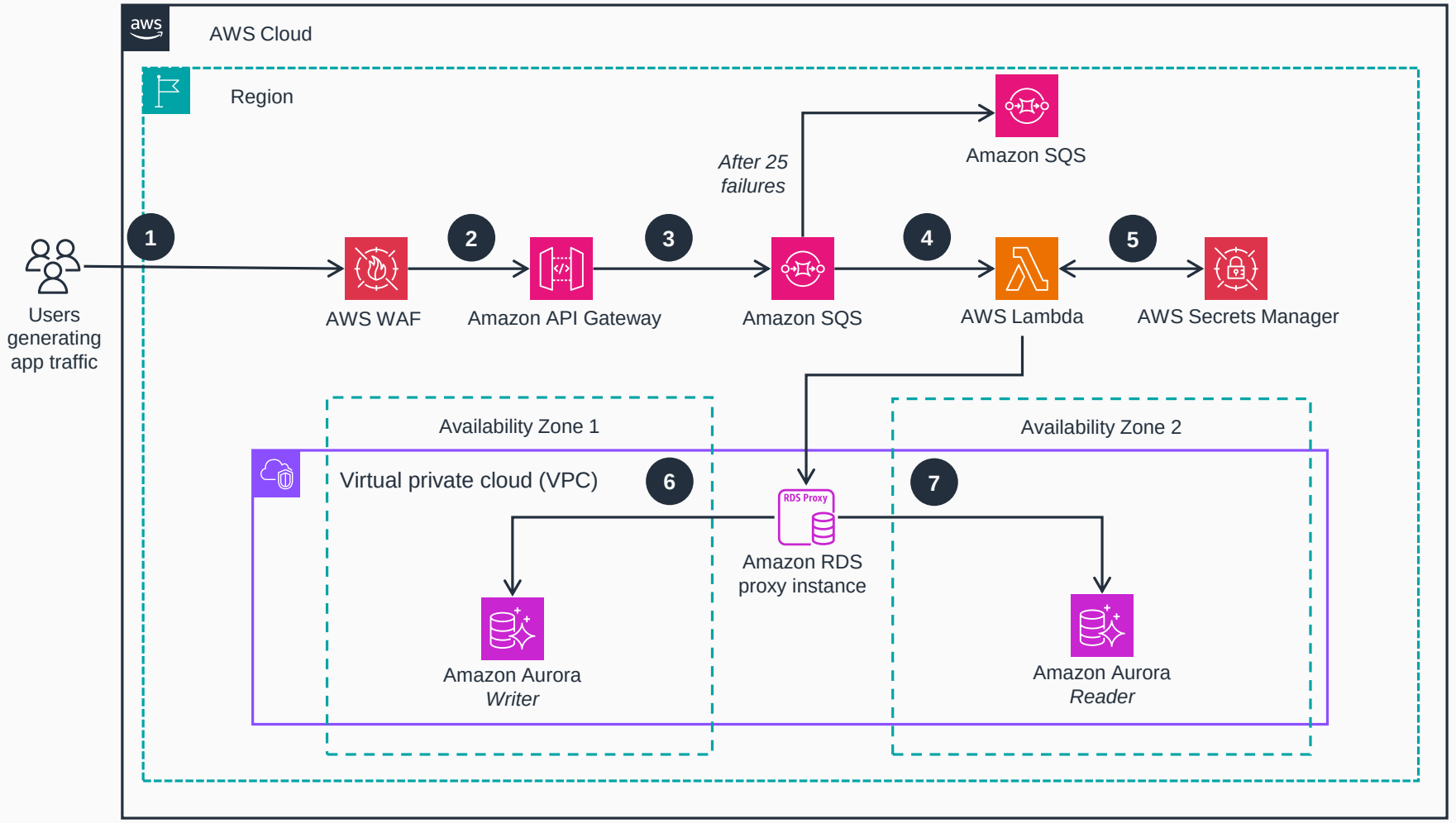
Guidance for Designing Resilient Applications with Amazon Aurora and Amazon RDS Proxy

This architecture diagram shows how to achieve near-zero RPO using Amazon Aurora and Relational Database Service (Amazon RDS) Proxy.



1. A user generates a request to write to the **Amazon Aurora** database. This request is evaluated by the **AWS WAF** configured with standard rules to protect against common web exploits.
2. If the request complies with the enacted **AWS WAF** policies, the request is routed to an **Amazon API Gateway**.
3. **API Gateway** forwards HTTPS requests to an **Amazon Simple Queue Service (Amazon SQS)** queue.
4. In the background, an event source mapping in **AWS Lambda** continuously polls the **Amazon SQS** queue for new messages. **Amazon SQS** is set to attempt message processing 25 times. If a message fails all attempts, it's sent to an **Amazon SQS** dead-letter queue (DLQ).
5. Upon receiving a new message, **Lambda** retrieves the database credentials stored in **AWS Secrets Manager** to connect to the **Aurora** database.
6. The message retrieved from **Amazon SQS** is written by **Lambda** to the primary **Aurora** instance in Availability Zone 1. This instance serves as the writer instance, with a reader instance deployed to Availability Zone 2.
7. In the event of a primary instance failure, **Aurora** automatically promotes a secondary instance to become the new primary, a process known as failover. Throughout this failover process, **Lambda** continues writing data to the **Aurora** cluster.

Reviewed for technical accuracy April 1, 2024

AWS Reference Architecture