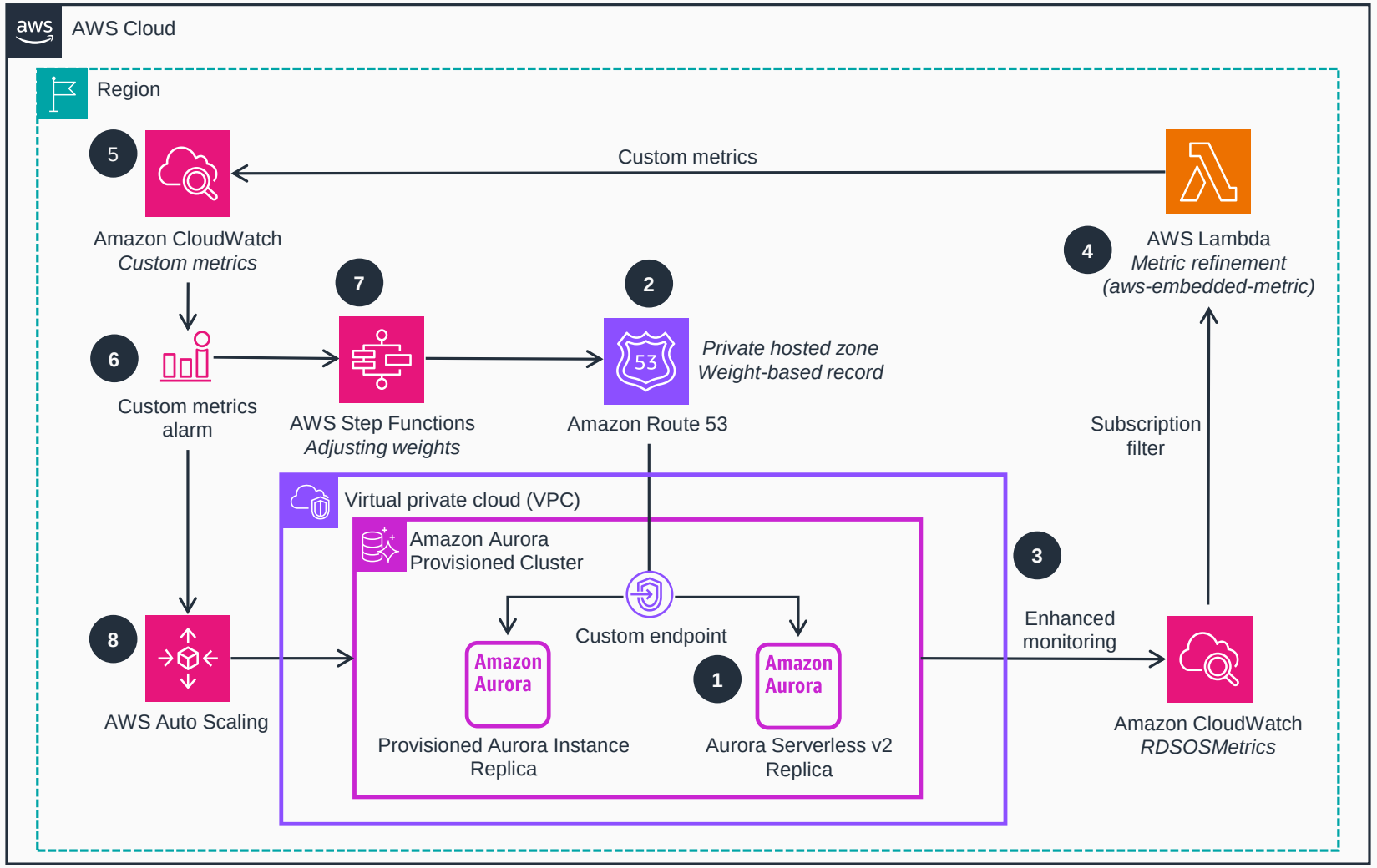
Guidance for Handling Data during Traffic Spikes on AWS

This architecture diagram shows how to prevent service failures due to instance creation time when scale-out occurs in a Amazon Aurora provisioned cluster. By integrating AWS services, it provides the flexibility to respond to sudden traffic spikes in Aurora at minimal cost with fully automated configuration.



1. Add **Amazon Aurora Serverless v2** (replica) to the **Aurora** Provisioned Cluster, and create each custom endpoint pointing to the replica.
2. Configure a private hosted zone on **Amazon Route 53**, and create a weight-based record with the same record name for each endpoint. Set the weight of the record pointing to the **Aurora Serverless v2** to zero.
3. Enable the Enhanced Monitoring feature on all instances to automatically collect RDSOSMetrics information in **Amazon CloudWatch**.
4. Create an **AWS Lambda** function that refines the information collected in the RDSOSMetrics log group to create custom metrics for the target instances for monitoring.
5. Use aws-embedded-metric to push the refined custom metrics to **CloudWatch** in near-real time.
6. Based on your stored custom metrics, configure **CloudWatch** alarms for weight adjustment.
7. When the alarm for weight adjustment occurs, it calls **AWS Step Functions**. **Step Functions** adjusts the weight of the **Route 53** record to distribute and recover traffic to **Aurora Serverless v2**.
8. Set up an autoscaling policy based on the custom metric alarms collected using **AWS Auto Scaling**. When an autoscaling alarm occurs, it invokes autoscaling for the provisioned replica instance.

Reviewed for technical accuracy May 7, 2024

AWS Reference Architecture