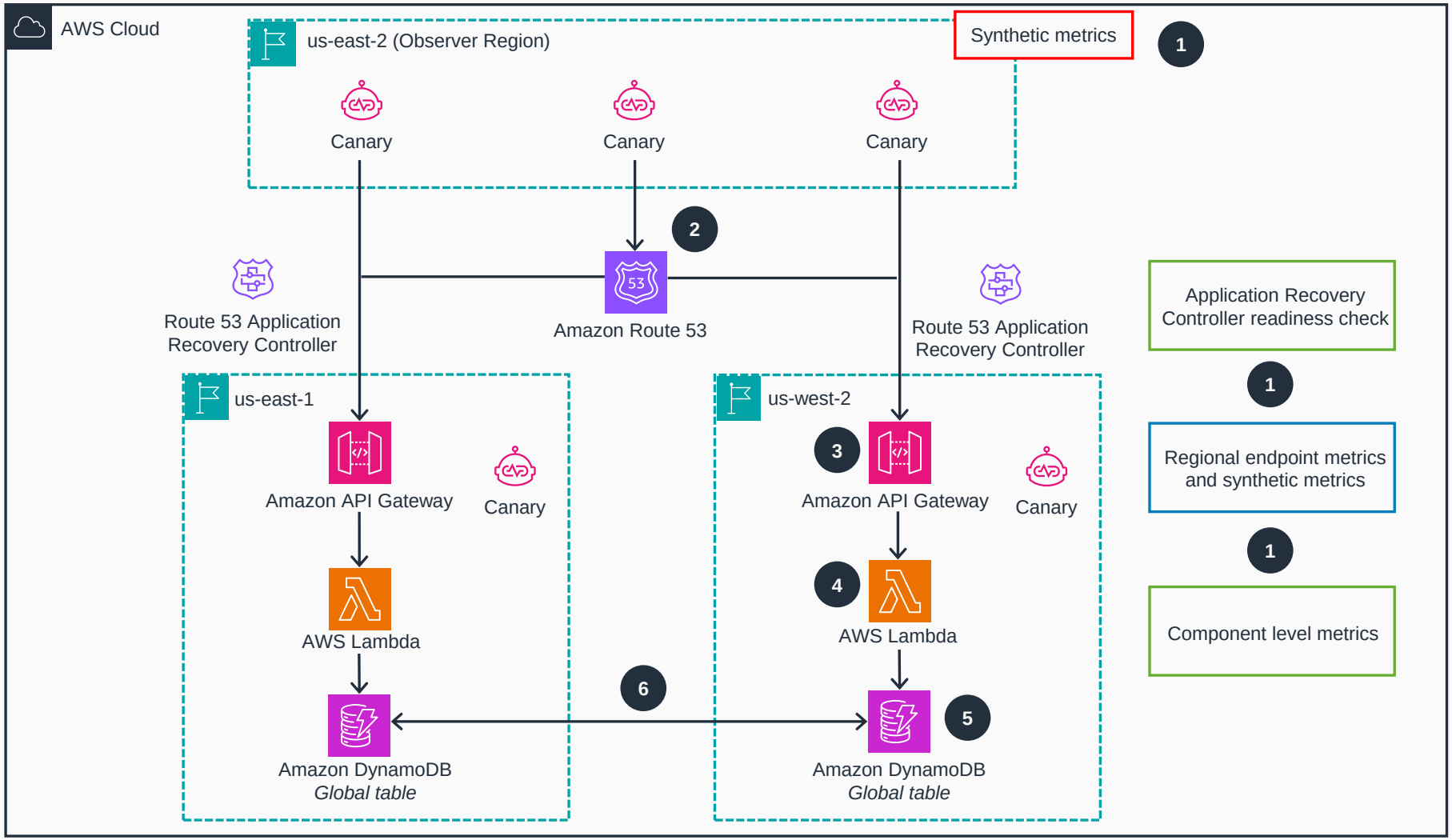
Guidance for Resilient Data Applications Using Amazon DynamoDB – Primary Region

The architecture diagram details the process, considerations, and observability setup required to build resilient multi-Region applications using DynamoDB global tables and perform failover and failback.



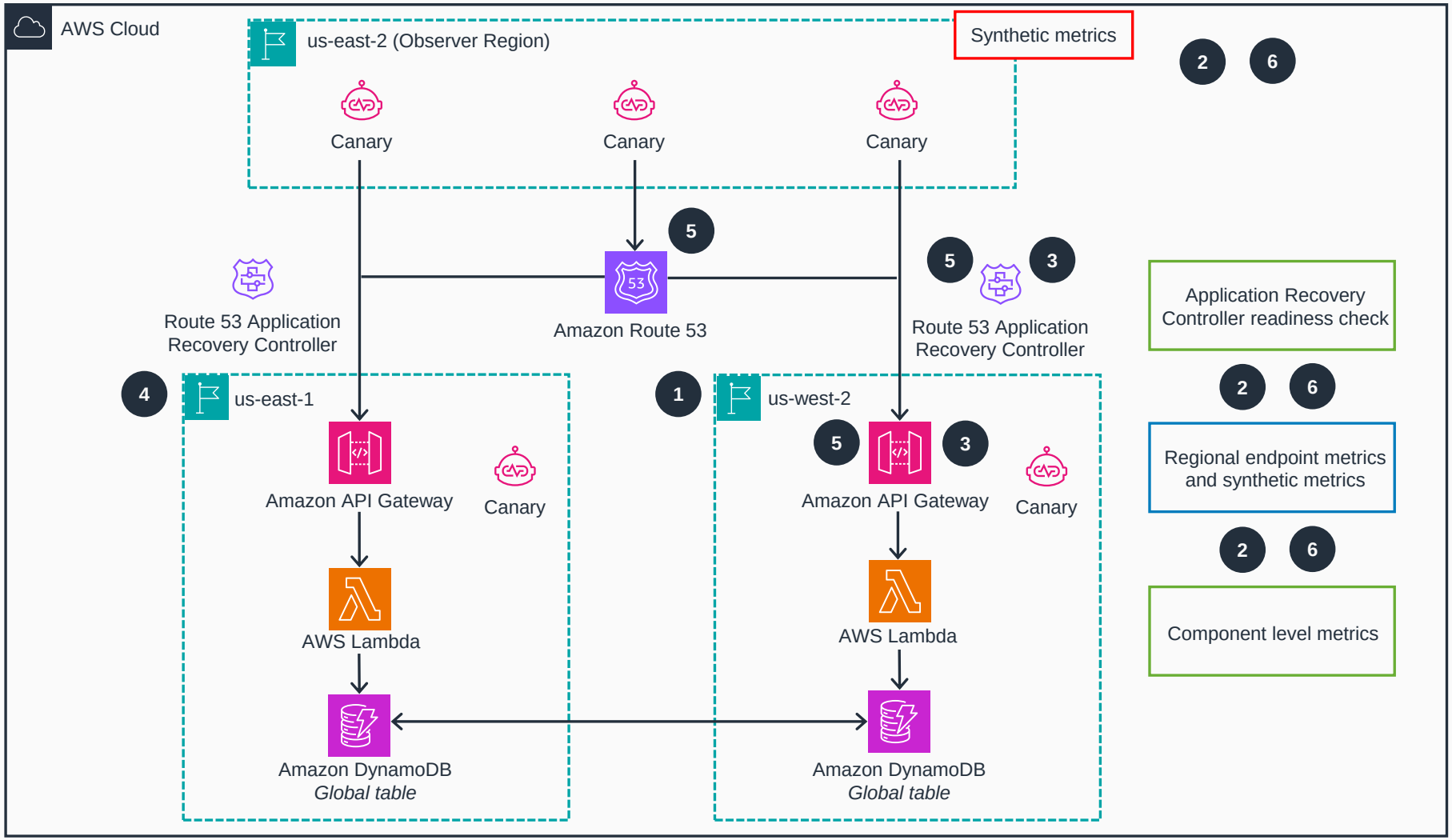
1. Check canaries, component-level metrics, regional endpoint metrics, synthetic metrics, and **AWS Health** API to identify any Region-scoped AWS service incidents (assuming they are operating in a healthy state).
2. The user's application performs a DNS lookup through **Amazon Route 53**. Based on the latency-based routing policy, **Route 53** returns the **Amazon API Gateway** endpoint from the Region with the lowest latency. In this case, the Region is us-west-2.
3. The user's application sends a request to the provided **API Gateway** endpoint in us-west-2, such as creating a new order. **API Gateway** receives the request and passes it to the configured **AWS Lambda** function in the us-west-2 Region.
4. A **Lambda** function processes the request from **API Gateway** and performs the necessary actions. In this case, it writes the new product order to the local **Amazon DynamoDB** table in us-west-2.
5. **DynamoDB** writes the data to the local table in us-west-2 and acknowledges the write success to the **Lambda** function. The **Lambda** function returns a response to the user's application through **API Gateway**, confirming the successful write.
6. The **DynamoDB** global table asynchronously replicates the data from the us-west-2 table to the corresponding table in us-east-1.

Reviewed for technical accuracy May 21, 2024

AWS Reference Architecture

Guidance for Resilient Data Applications Using Amazon DynamoDB – Cross-Region Failover and Failback

This architecture diagram details what happens if your application experiences an outage, showing how the initial setup is used to perform cross-Region failover and failback.



1. A Region-scoped outage starts in us-west-2, causing intermittent failures and elevated response times for your application in that Region.
2. Data sources such as canaries, component-level metrics, regional endpoint metrics, synthetic metrics, and **AWS Health** API help detect issues from the Region-scoped AWS service events and subsequently alert you.
3. After concluding that there is an outage in the us-west-2 Region, evacuate us-west-2 until the event is resolved. Change the **Route 53** Application Recovery Controller (ARC) to disable traffic routing to us-west-2. Consider cutting off traffic at the **API Gateway** level if you're concerned about the time it takes for DNS changes to propagate.
4. All traffic will now be routed to the us-east-1 Region. The application should be able to handle the increased traffic, provided service quotas have been adjusted previously. Monitor the us-east-1 Region closely, ensuring the application is functioning properly and handling the increased load.
5. Once the event in us-west-2 is resolved, start restoring service to that Region. Remove any **API Gateway** restrictions, change **Route 53** ARC routing control to allow traffic to us-west-2 again, and gradually restore traffic using **Route 53** weighted routing policies.
6. Continue monitoring the us-west-2 Region to confirm the application is performing at the expected level of service.

Reviewed for technical accuracy May 21, 2024

AWS Reference Architecture