

ARC308

# Best practices for creating multi-Region architectures on AWS

**Joe Chapman**

(he/him)

Principal Solutions Architect  
AWS

**Neeraj Kumar**

(he/him)

Principal Technologist  
AWS



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## AWS re:Invent 2023 - Best practices for creating multi-Region architectures on AWS (ARC308)



AWS Events  
100K subscribers

Subscribe



37



Share



Clip



Save



1,260 views Dec 2, 2023 [#AWSreInvent](#) [#AWSreInvent2023](#)

Architecting and building a multi-Region architectures can come with a new set of challenges. These challenges include how to manage dependencies, infrastructure deployments, data replication, consistency, observability, testing, and operations. Whether you're needing to expand to multiple Regions to improve resilience, adhere to governmental data regulations, or improve end-user latency, this session highlights best practices, design principles, and sample architectures to help you meet your requirements.

Learn more about AWS re:Invent at <https://go.aws/46iuzGv>.

Subscribe:

More AWS videos: <http://bit.ly/203zS75>

More AWS events videos: <http://bit.ly/316g9t4>

### ABOUT AWS

Amazon Web Services (AWS) hosts events, both online and in-person, bringing the cloud computing community together to connect, collaborate, and learn from AWS experts.

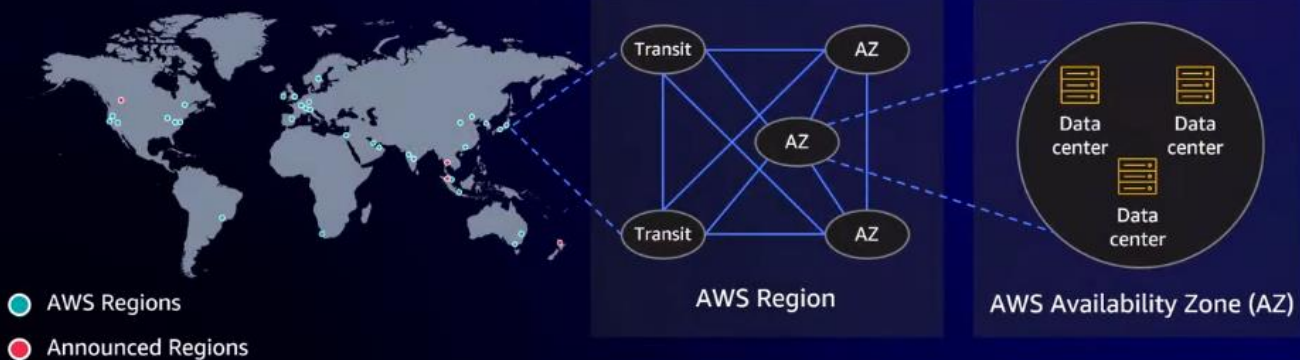
AWS is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

## Agenda

- Multi-Region – Is it necessary?
- Scenario: Retail banking
- Scenario: Authentication provider
- Recap on core learnings

# Remember: AWS Regions are built to be resilient!

 Are you Well-Architected?



## Understanding the requirements

Before getting started:

Need to understand

- Current architecture
- End state goals & success criteria
- Align business and IT stakeholders
- Can a single Region meet requirements?



Level set on cost, complexity, and benefit tradeoffs

**Multi-Region journey:**  
**Fintech retail bank**



# Company profile: Fintech retail bank

## Company profile

- Fast-growing in retail banking in the US
- Plan to expand globally
- Single Region; All-in on AWS

## Need to improve:

- DR and operational continuity
- Testing

## Success criteria:

- RPO: 15 mins
- RTO: 1 Hour

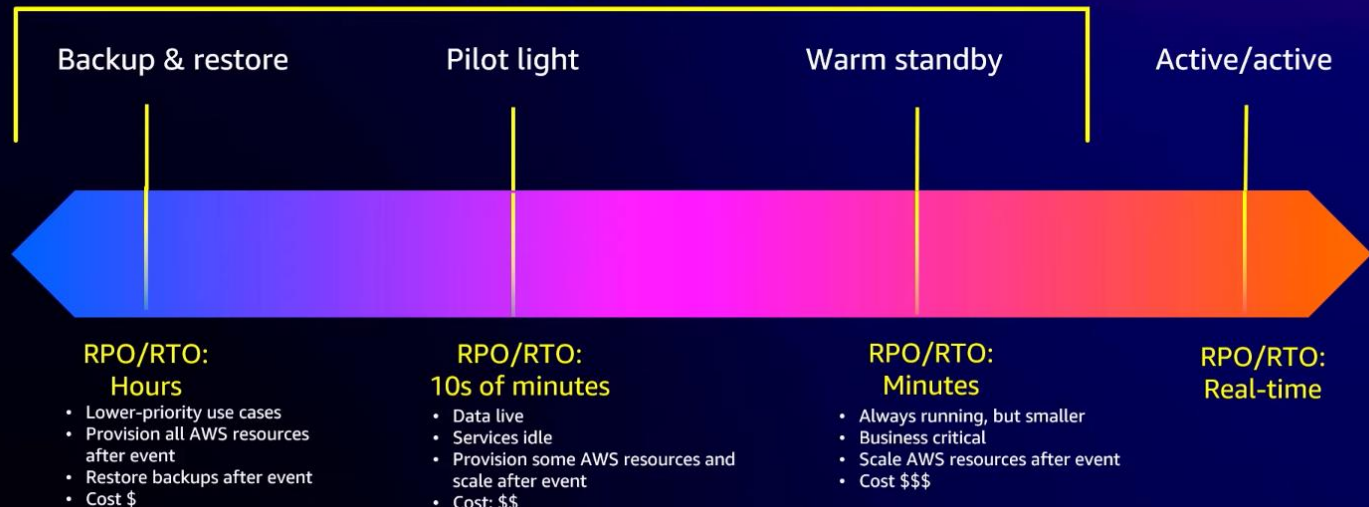


RPO is the Recovery Point Objective is about how much data period is allowed to be lost, RTO is about the speed of recovery.

# Multi-Region: Strategies for disaster recovery

## UNDERSTANDING REQUIREMENTS

### Active/passive strategies



What are the different strategies that can be adopted? The above is from our well architected framework analysis that requires the 4 different strategy levels above. The default strategy is the **backup & restore strategy** to prevent against data corruption and be able to go back to a previously good state. In **Pilot light strategy**, we take replicas of our application workloads in a different region and zone. We are focused on data replication by continuously syncing data, make the infra as a code templates available in the separate region for emergency needs. The **warm standby strategy** is that we run some baseline compute available to be used and reduce our time to recover during failover ramp-up needs. The **active-active strategy** means we have a fully available and running replica of our workloads running under an active sharding or replication implementation or splitting the transactions across the different regions.



# Warm standby

OPERATIONAL READINESS



Users



Route 53



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Warm standby

OPERATIONAL READINESS



Users



Route 53



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

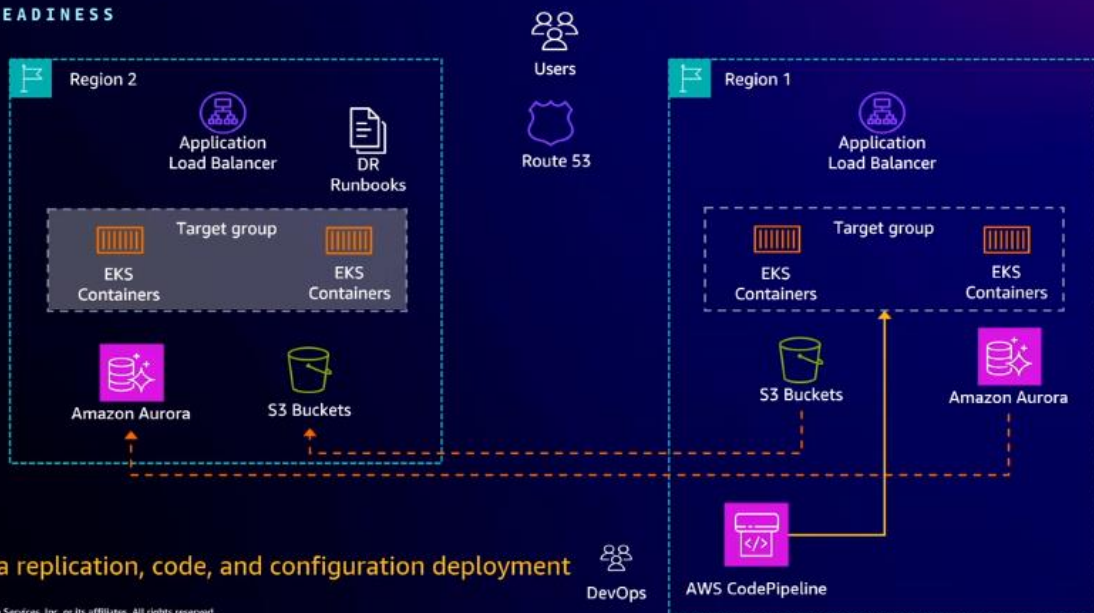
# Warm standby

OPERATIONAL READINESS



# Warm standby

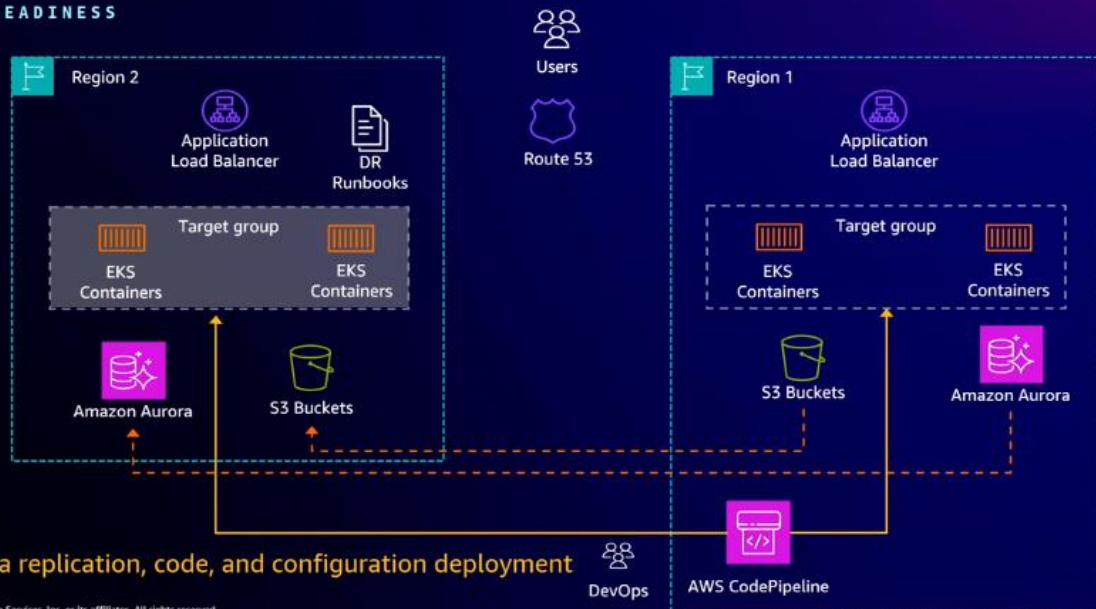
OPERATIONAL READINESS



You also need to think about how to deploy your code to both regions, deploy to the regions one at a time

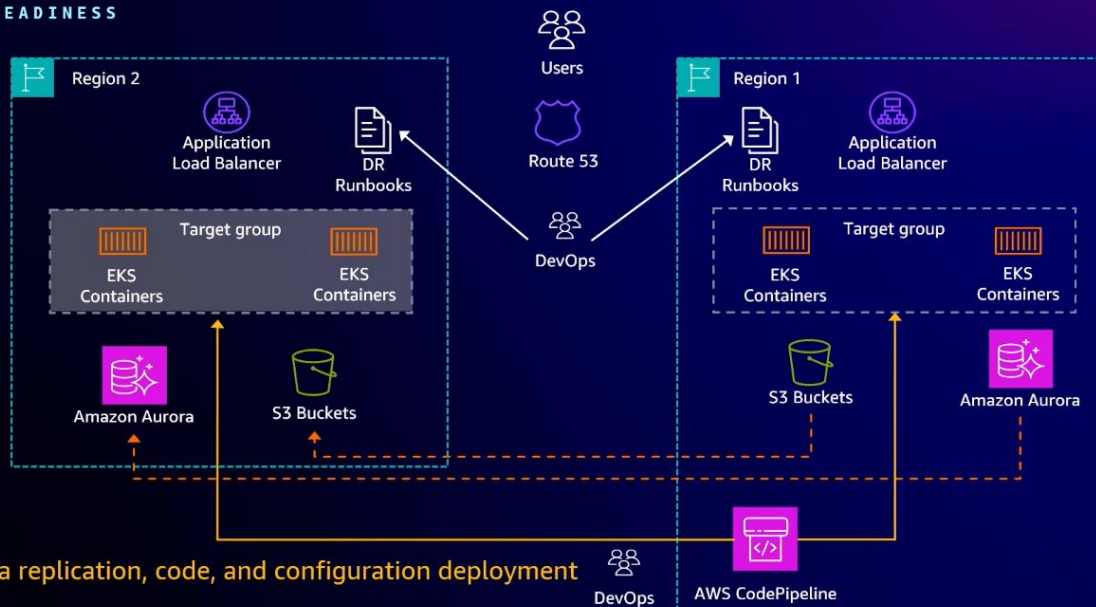
# Warm standby

OPERATIONAL READINESS



# Warm standby

OPERATIONAL READINESS



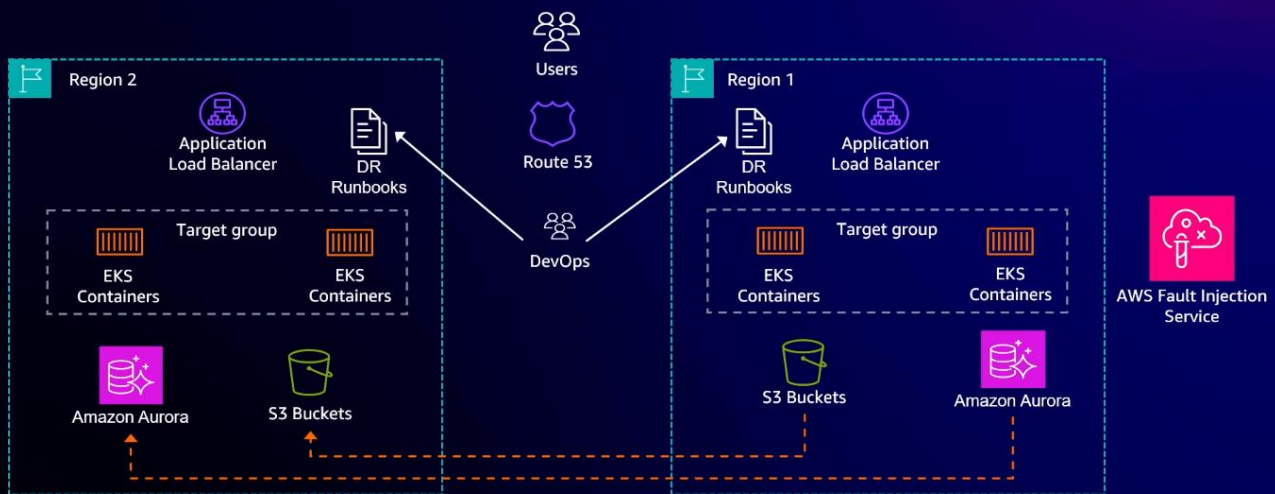
You need your runbooks to be available for both regions and can be invoked in there

# Multi-Region best practices

- DR/operational continuity strategy should be informed by business impact assessment (BIA)
- Deploy code and configuration one Region at a time
- Choose the right data replication strategy to meet RPO and data consistency requirements
- Ensure recovery procedures/runbooks can be invoked from the secondary region

## Gamedays with AWS Fault Injection Service

OPERATIONAL READINESS

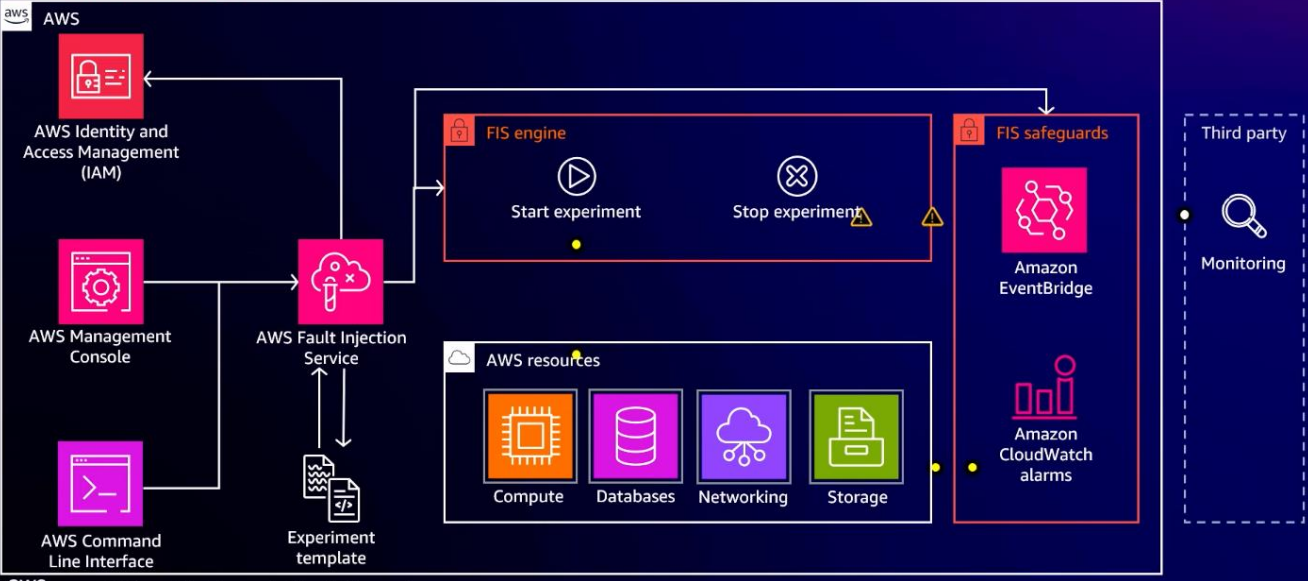


This is about testing your detection controls, recovery and mitigation controls. Are your runbooks working correctly as expected?



# Gamedays with AWS Fault Injection Service

## OPERATIONAL READINESS



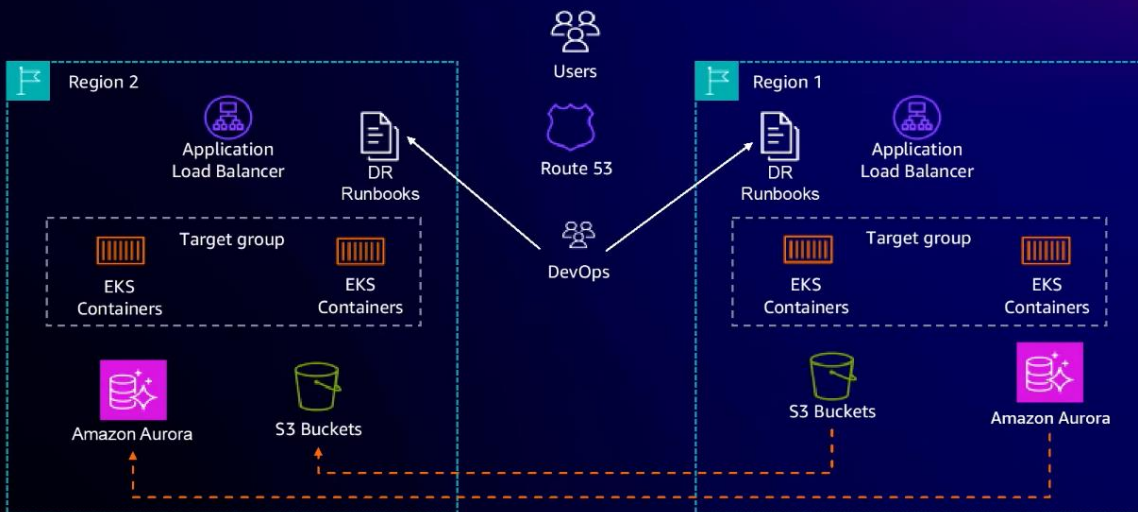
This is a fully managed service that helps you inject specific failures to test your detection, observability and recovery controls. It uses events and actions that are configured to test different simulated failure modes.

## Multi-Region best practices

- Regularly test the detection and recovery controls

## Validating the failover readiness

### OPERATIONAL READINESS

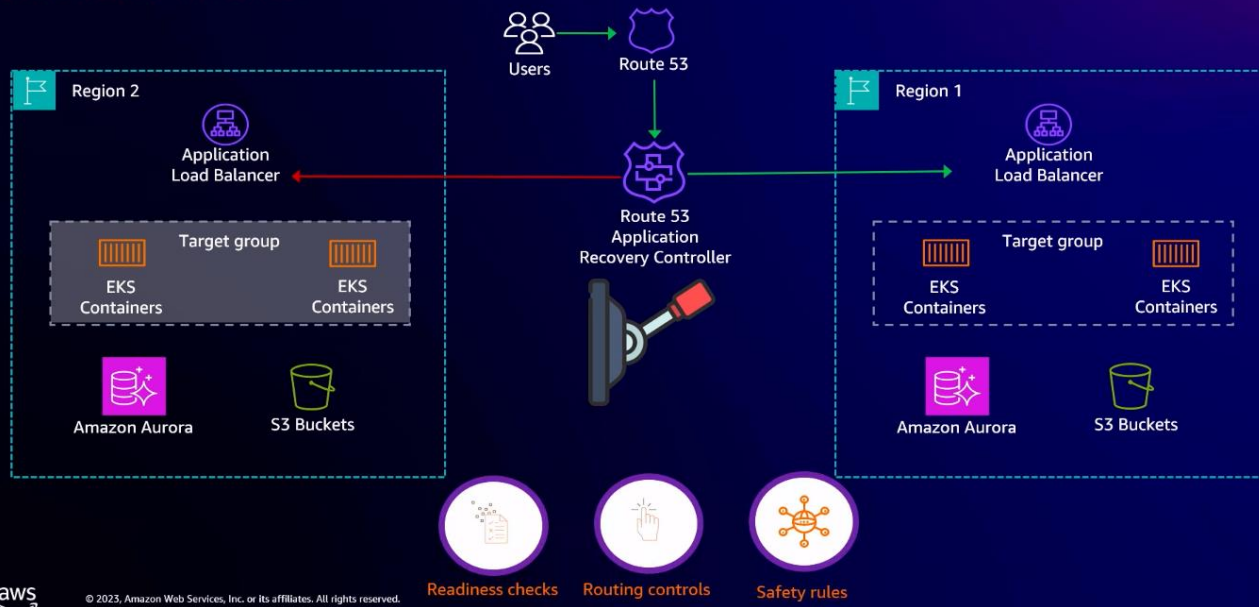


This means detecting if the secondary region has any mismatch in values/terms of service level limits, region capacity, data availability.



# Route 53 Application Recovery Controller

OPERATIONAL READINESS



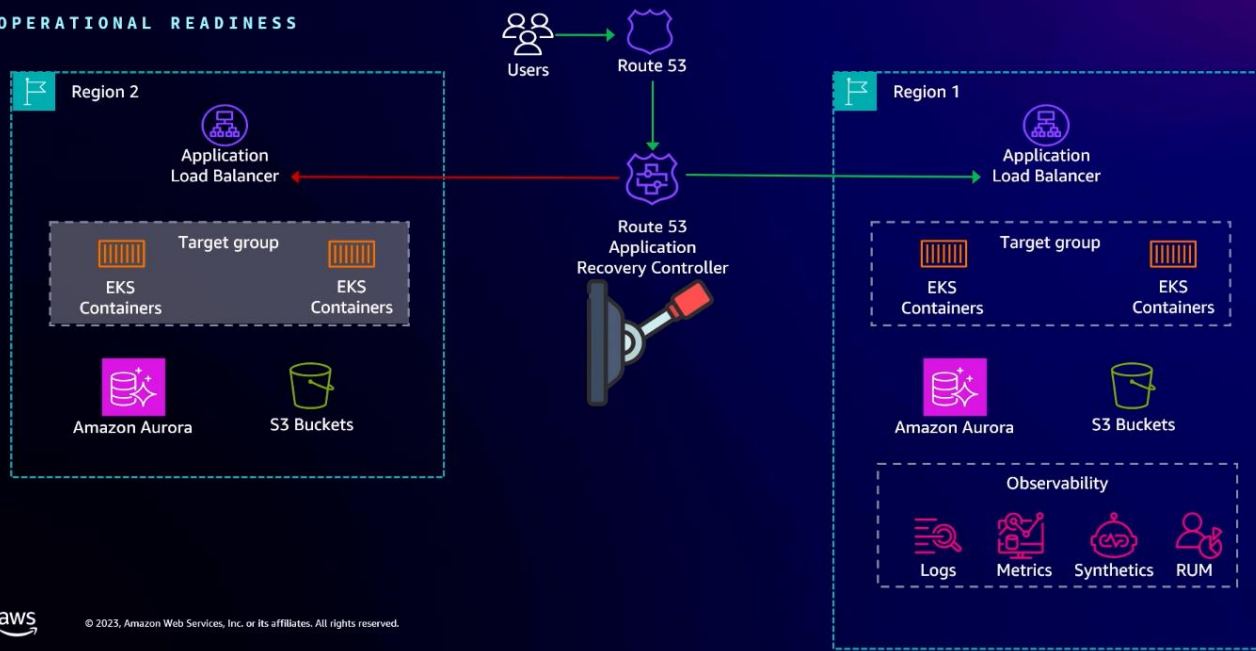
ARC is part of Route53 and can check the details provided for the secondary region infrastructure as well as the routing controller itself that manually triggers the traffic failover action.

## Multi-Region best practices

- Build failover checks as part of the recovery strategy
- Use Application Recovery Controller as part of the failover and failback strategies
- Automate the procedures, but failover decision should be manual

# Failure mode: Observability impairment

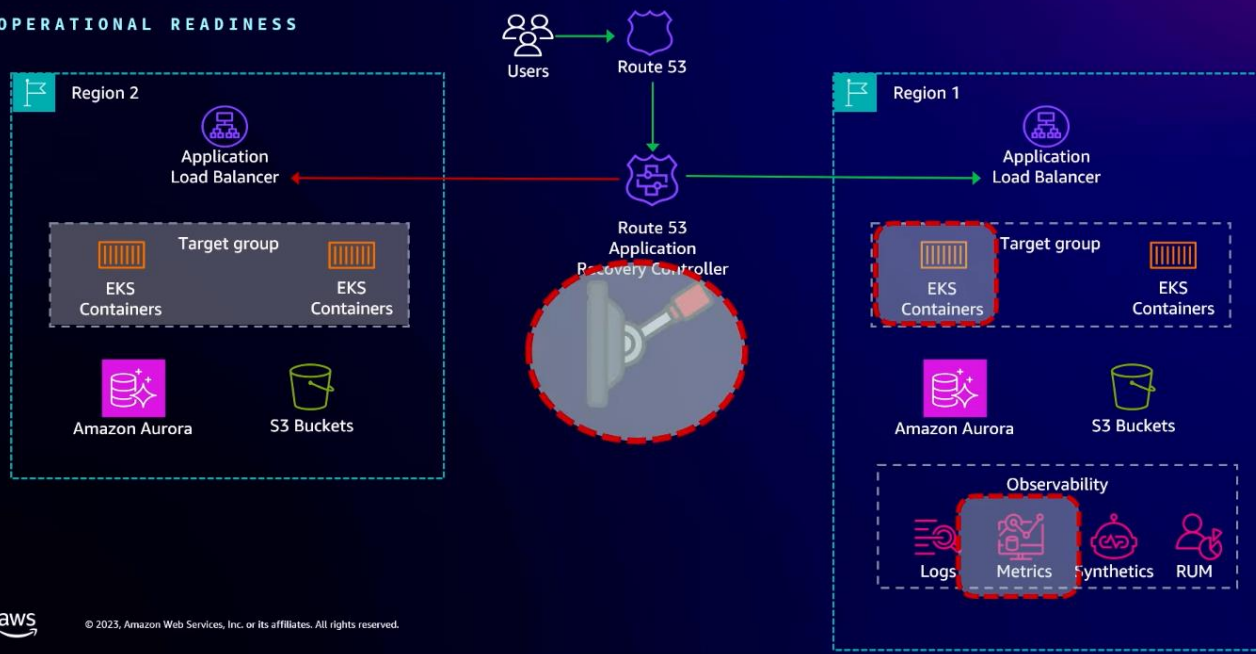
OPERATIONAL READINESS



We need signals to detect and apply the failover, this is from the observability stack. What happens if our observability/detection control is impaired? We need observability across both regions as below.

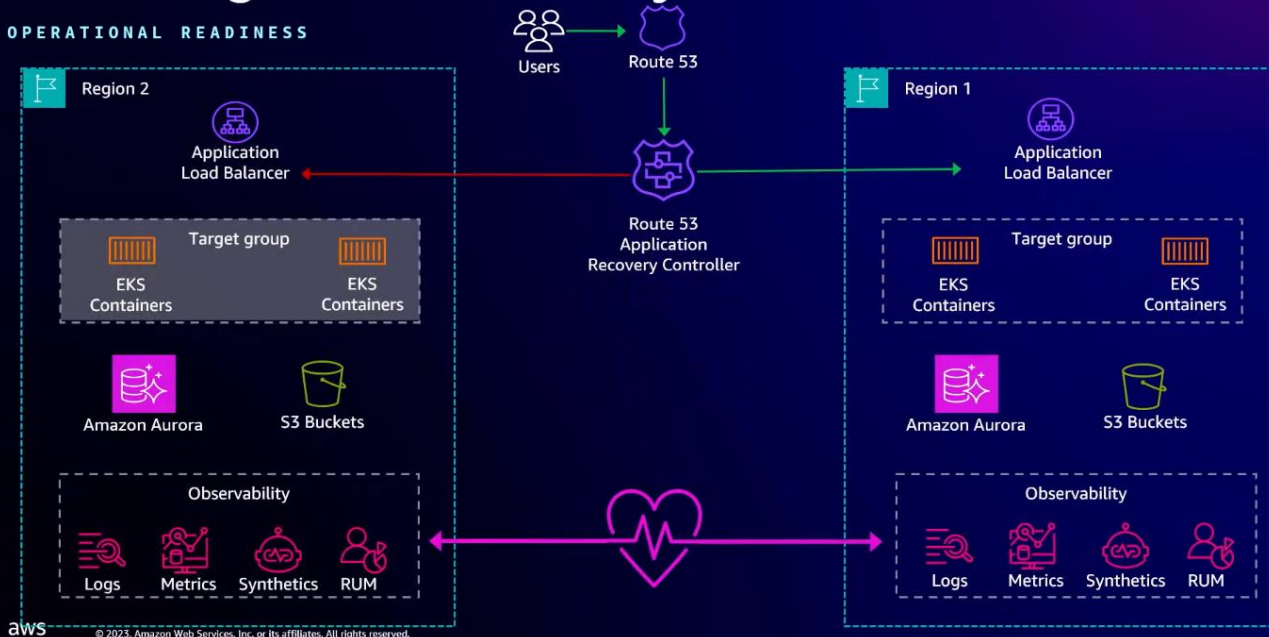
# Failure mode: Observability impairment

OPERATIONAL READINESS



# Multi-Region observability

## OPERATIONAL READINESS



We need to extend the observability stack across both regions at some level like using heartbeats from both regions and correlating both signals and make decisions.

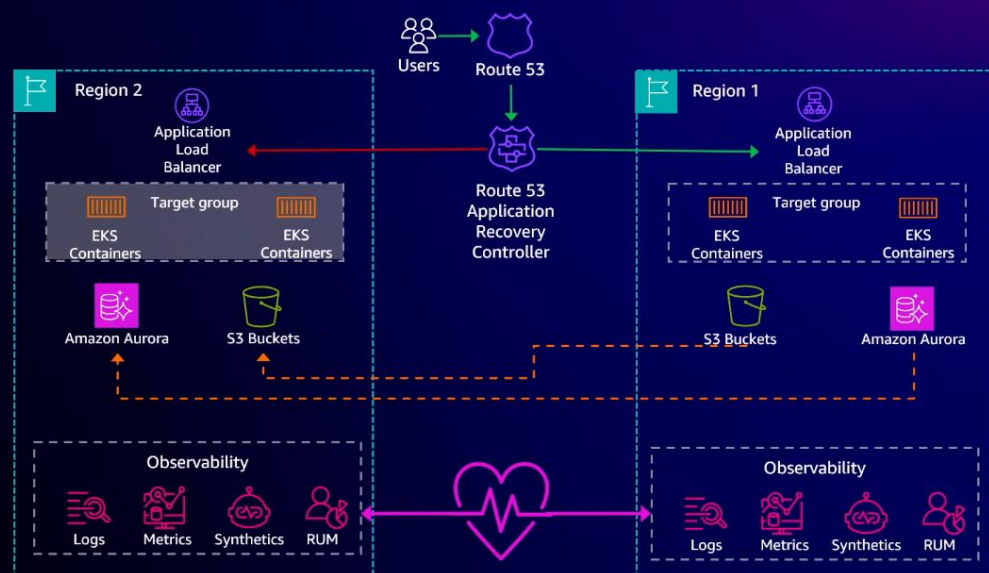
## Multi-Region best practices

- Observe Region's health from outside (another Region) to deterministically isolate the impaired Region before invoking the failover procedures

## Current (evolving) architecture & tradeoffs

### Tradeoffs:

- Operational overhead
- Additional complexity
- Additional cost





# Multi-Region journey: Authentication provider



## Company profile: Authentication provider

UNDERSTANDING THE REQUIREMENTS

### Company profile

- Fast-growing
- Single Region

### Need to improve:

- Near-zero downtime
- Latency for a growing global customer base

### Success criteria:

- 99.99% SLA for authentication API
- Reduce global authentication latency by 30%



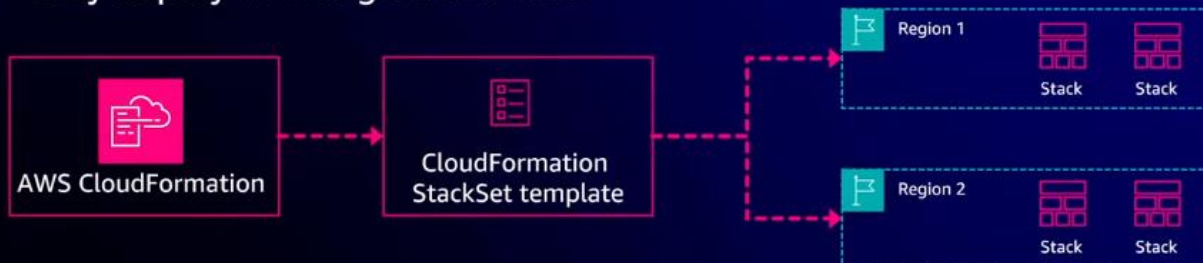
Users



## Codify everything

OPERATIONAL READINESS

- Fix environment inconsistencies
- Use smaller templates
- Only deploy to 1 Region at a time



Change the mindset – no manual changes!

# Network & routing

## UNDERSTANDING DEPENDENCIES

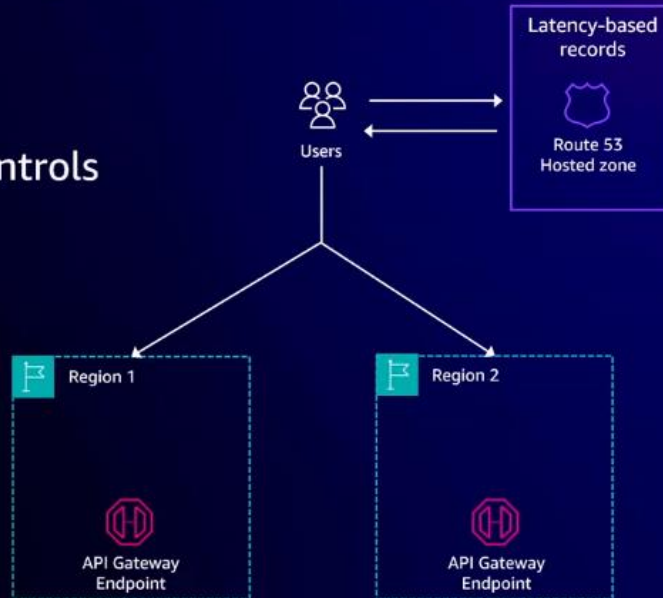
- Intelligent routing
- Resilient routing controls
- Layer 7 protection



# Network & routing

## UNDERSTANDING DEPENDENCIES

- Intelligent routing
- Resilient routing controls
- Layer 7 protection

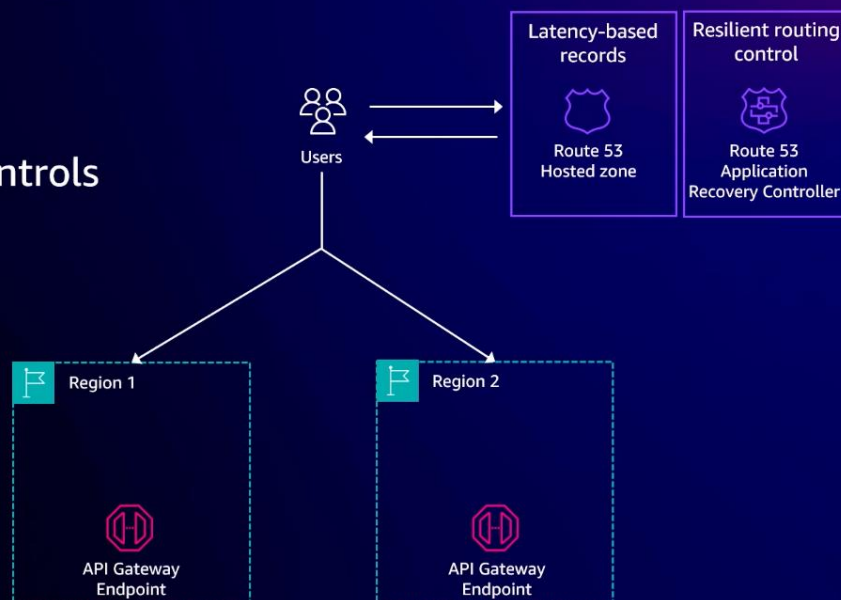


© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Network & routing

## UNDERSTANDING DEPENDENCIES

- Intelligent routing
- Resilient routing controls
- Layer 7 protection

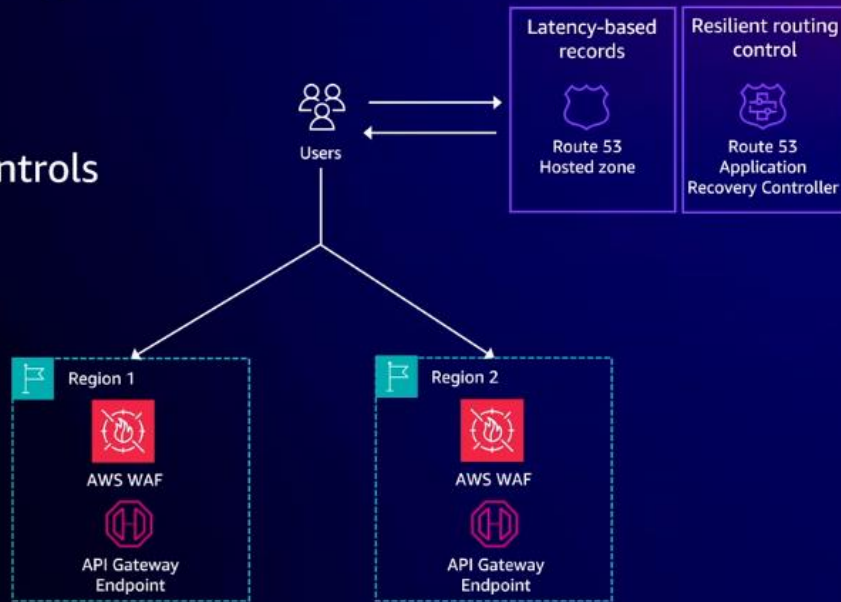


© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Network & routing

UNDERSTANDING DEPENDENCIES

- Intelligent routing
- Resilient routing controls
- Layer 7 protection



## Multi-Region best practices

- Deployment pipelines:
  - Use infrastructure as code for consistent deployments
  - All changes must be tracked in code
  - Allow “bake” time between Region deployments
- Use resilient routing controls to direct traffic flows

## Multi-Region writing

UNDERSTANDING THE DATA

- Multi-writer (active/active)
- Synchronous vs. asynchronous
- Hedging
- Idempotency

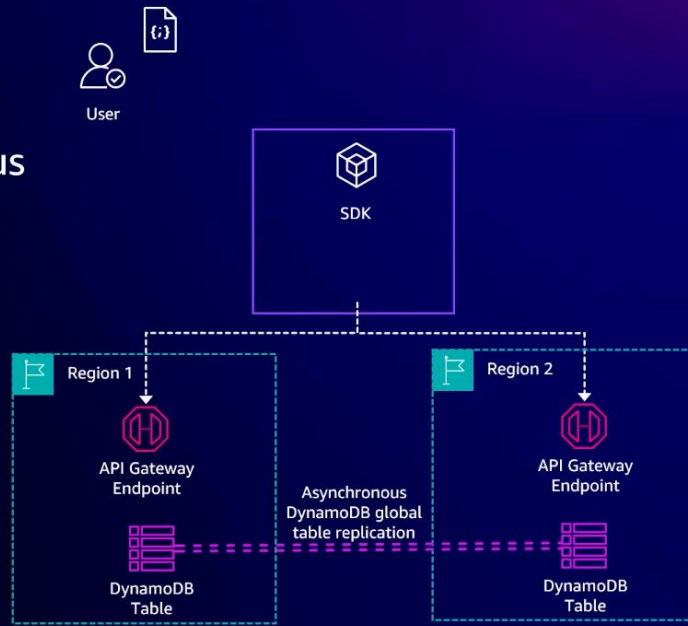




# Multi-Region writing

## UNDERSTANDING THE DATA

- Multi-writer (active/active)
- Synchronous vs. asynchronous
- Hedging
- Idempotency

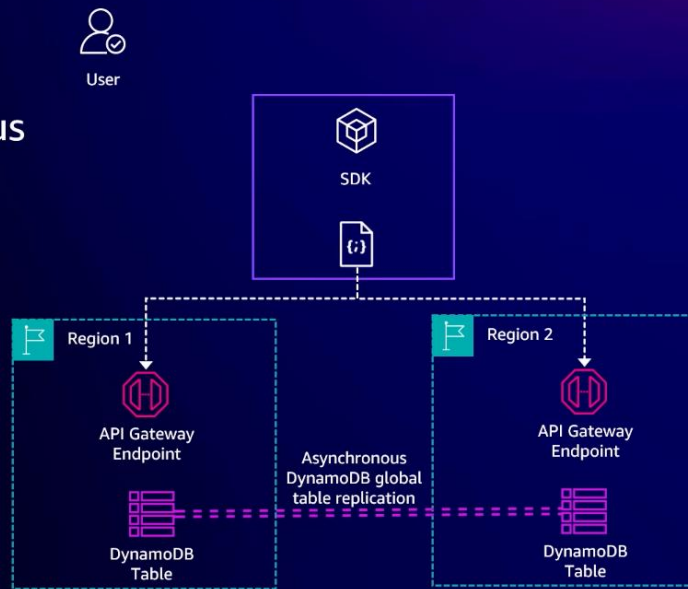


Data makes changes to the credentials; the SDK will send to both regions at the same time for updates

# Multi-Region writing

## UNDERSTANDING THE DATA

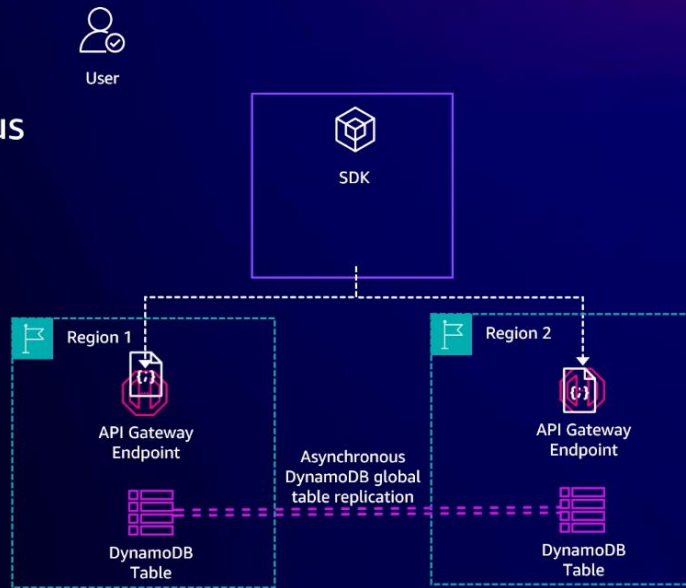
- Multi-writer (active/active)
- Synchronous vs. asynchronous
- Hedging
- Idempotency



# Multi-Region writing

UNDERSTANDING THE DATA

- Multi-writer (active/active)
- Synchronous vs. asynchronous
- Hedging
- Idempotency



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Data replication

UNDERSTANDING THE DATA

- Speed of light =  $\sim 300,000$  km/sec
- Longer distance = more latency
- Strong vs. eventual consistency



Availability

CAP  
Theorem

Pick  
two

Consistency

Partition  
tolerance



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Consistency & observability

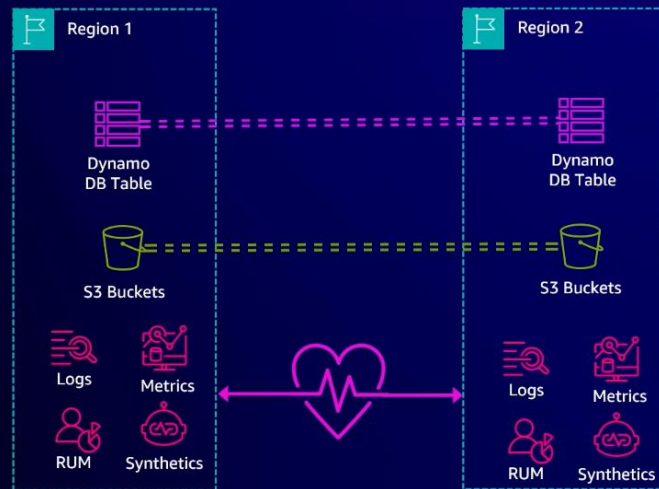
UNDERSTANDING THE DATA

## Data consistency

- Detection & monitoring

## Observability

- ReplicationLatency (DynamoDB)
- Duration (Synthetic)
- PerformanceResourceDuration (RUM)
- S3 replication metrics (bytes & operations pending, replication latency, failed operations)



## Multi-Region best practices

- Understand consistency and replication options
- Utilize asynchronous writes across Regions for increased performance and availability
- Multi-Region writes should be idempotent, avoid conflicts, and have appropriate conflict resolution
- Observability needs to happen from multiple viewpoints

## Regional independence

UNDERSTANDING DEPENDENCIES

- Enforce fault isolation boundaries
- Test to find the unknowns





# Multi-Region operations

## OPERATIONAL READINESS

- Multi-Region observability
- Multi-Region management
- Multi-Region governance
- Multi-Region service quotas



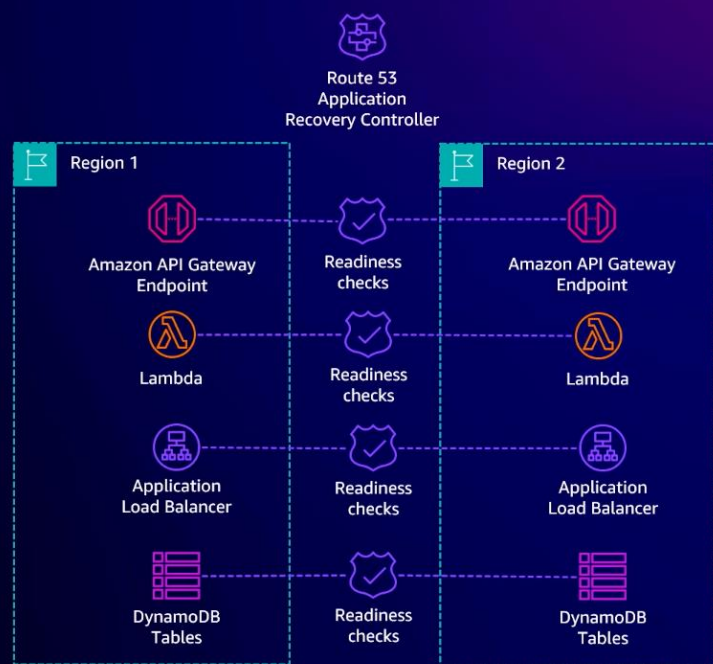
# Multi-Region best practices

- Understand dependencies, architect for Regional independence
- Plan for increased operational needs
- Track AWS service quotas across Regions
- Runbooks should include planned (healthy state) and unplanned (unhealthy state) actions. Initiate actions from healthy locations

# Tracking consistency

## OPERATIONAL READINESS

- Consistent deployments
- Detection of inconsistencies

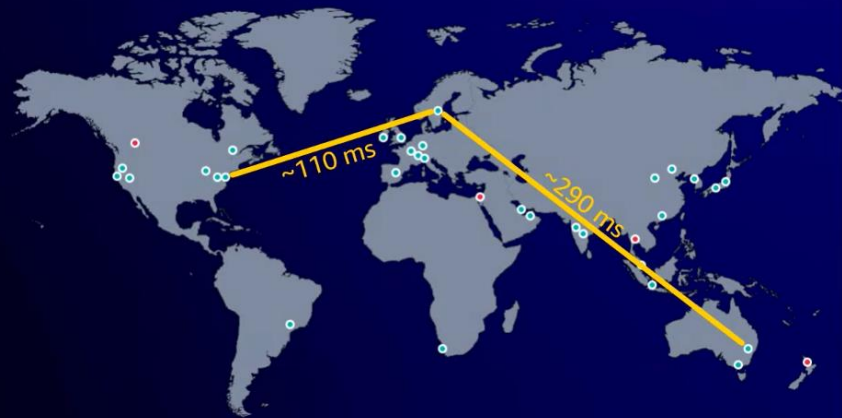


# Continued Regional expansion

OPERATIONAL READINESS

Additional operational considerations:

- Latency to users
- Cost
- Service availability



\*Latency ms are examples

## Multi-Region best practices

- Have tooling in place to actively monitor for drift between workload deployments
- When selecting a Region, understand userbase location characteristics, differences in cost, and service availability between Regions

## Current (evolving) architecture



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Multi-region best practices recap

## Key takeaways

- AWS Regions are built to be highly resilient; most applications do not require multi-Region. Fully understand the requirements first
- Data and dependencies are complex and require special attention
- Observability, code, testing, and deployment best practices for operational readiness
- Don't underestimate added operational cost
- No "one size fits all" solution, consider all tradeoffs and don't rush – we're here to help

## AWS multi-Region fundamentals

- Understanding:
  - Requirements
  - Data
  - Dependencies
- Operational readiness



AWS Multi-Region  
Fundamentals whitepaper



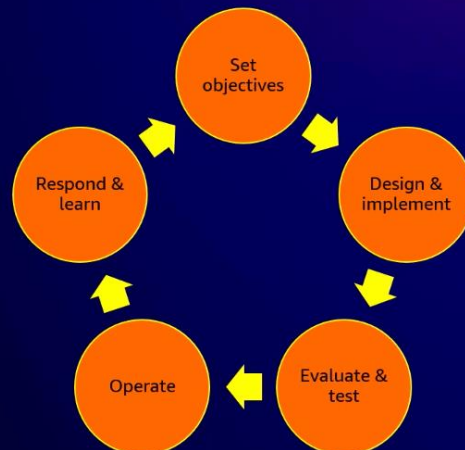
# New! The AWS Resilience Lifecycle Framework

The **Resilience Lifecycle Framework** shares strategies, services, and mechanisms you can use to help improve your resilience posture.

- Continuous process; not a one-time effort
- Modeled after a standard SDLC to easily incorporate into your existing processes



Available in the AWS Prescriptive Guidance Library and [aws.amazon.com/resilience](https://aws.amazon.com/resilience)



## Purpose-built AWS resilience offerings

BUILD RESILIENT, HIGHLY AVAILABLE APPLICATIONS IN THE AWS CLOUD

### AWS Resilience Hub

Analyze the components of your application to uncover potential resilience weaknesses

### AWS Fault Injection Service

Improve application performance, observability, and resilience through controlled fault injection experiments

### AWS Elastic Disaster Recovery

Minimize downtime and data loss with fast, reliable recovery of on-premises and cloud-based applications

### AWS Backup

Protect data at scale using this cost-effective, fully managed, policy-based service

### Amazon Route 53 Application Recovery Controller

Automate management and coordination of recovery for your applications across AWS AZs or Regions

### AWS solutions

Find purpose-built AWS resilience solutions, Partner solutions, and guidance in the AWS Solutions Library

# Thank you!



Please complete the session survey in the mobile app

**Joe Chapman**

✉ [jochp@amazon.com](mailto:jochp@amazon.com)

🌐 [linkedin.com/in/chapmanjoe](https://linkedin.com/in/chapmanjoe)

**Neeraj Kumar**

✉ [neeraku@amazon.com](mailto:neeraku@amazon.com)

🌐 [linkedin.com/in/neerajkumar-tech](https://linkedin.com/in/neerajkumar-tech)