

The logo for AWS re:Invent features the words "AWS" and "re:Invent" stacked vertically. "AWS" is in a smaller, sans-serif font above "re:Invent", which is in a larger, bold, sans-serif font. The entire logo is white against a dark blue background.

AWS
re:Invent

A R C 2 0 9

Architecture Patterns for Multi-Region Active-Active Applications

Amy Che
Principal Solutions Delivery Manager
AWS Solutions Architecture

Darin Briskman
AWS Chief Evangelist
AWS Database, Analytics, & ML

Christopher Lane
Enterprise Architect
Chick-fil-A Corporate

Agenda

So you want a multi-region architecture

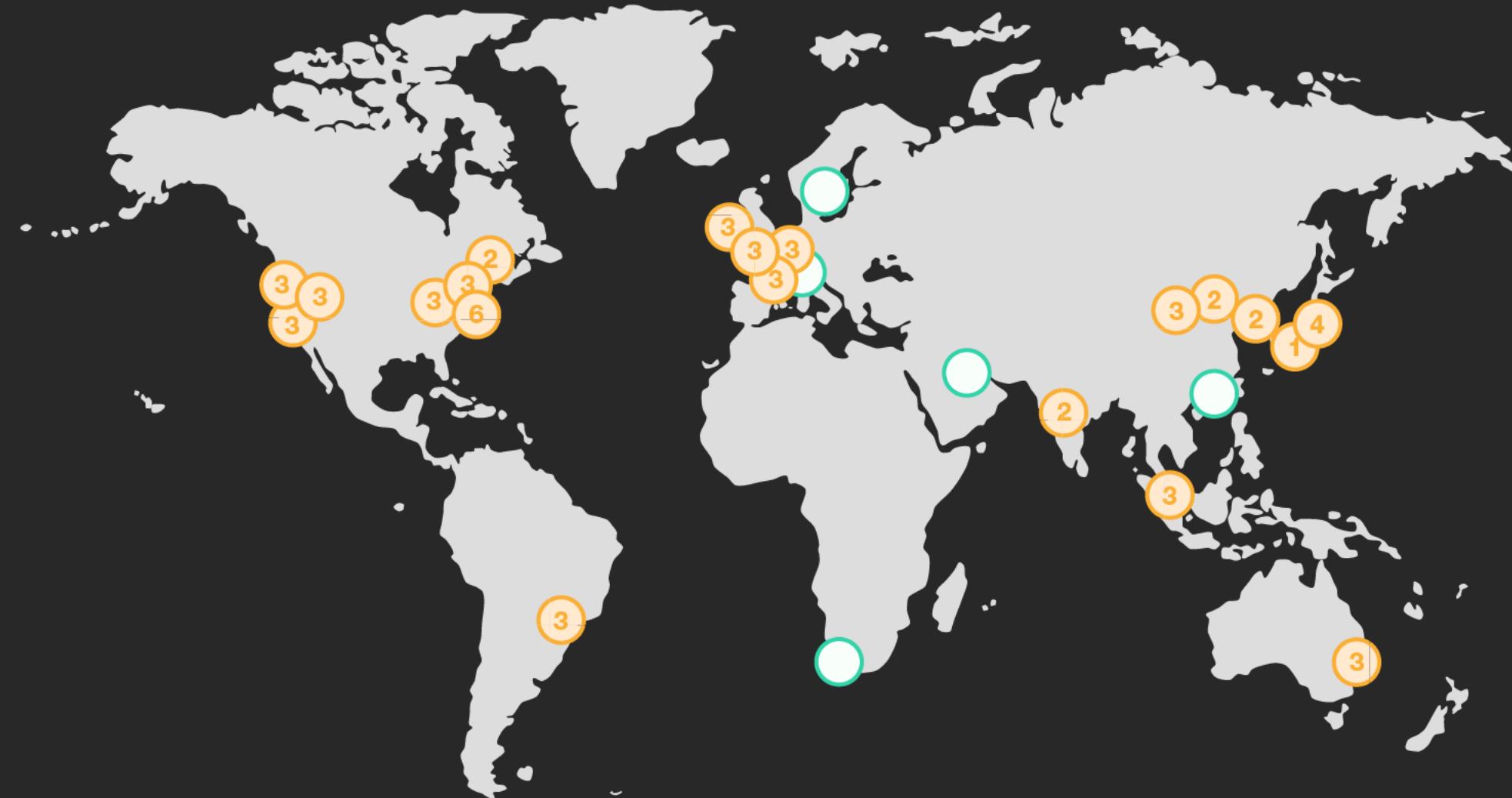
Multi-region business continuity

Re-architecting a single region to multi-region active-active

A customer's journey: Chick-fil-A's path to multi-region

So you want a multi-region architecture

What does multi-region mean to AWS?

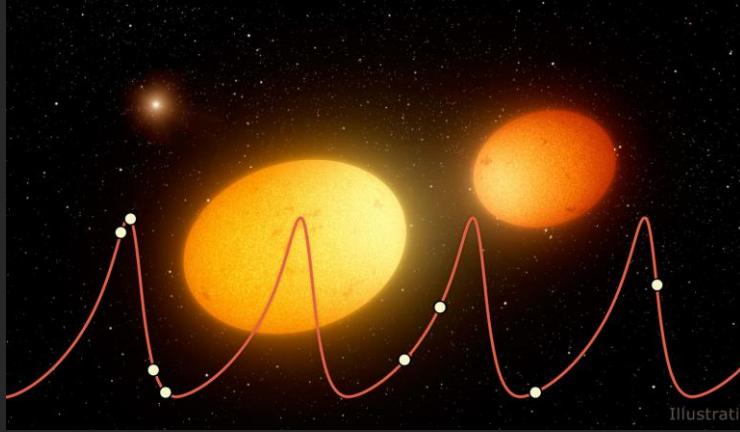


multiRegionActiveActive

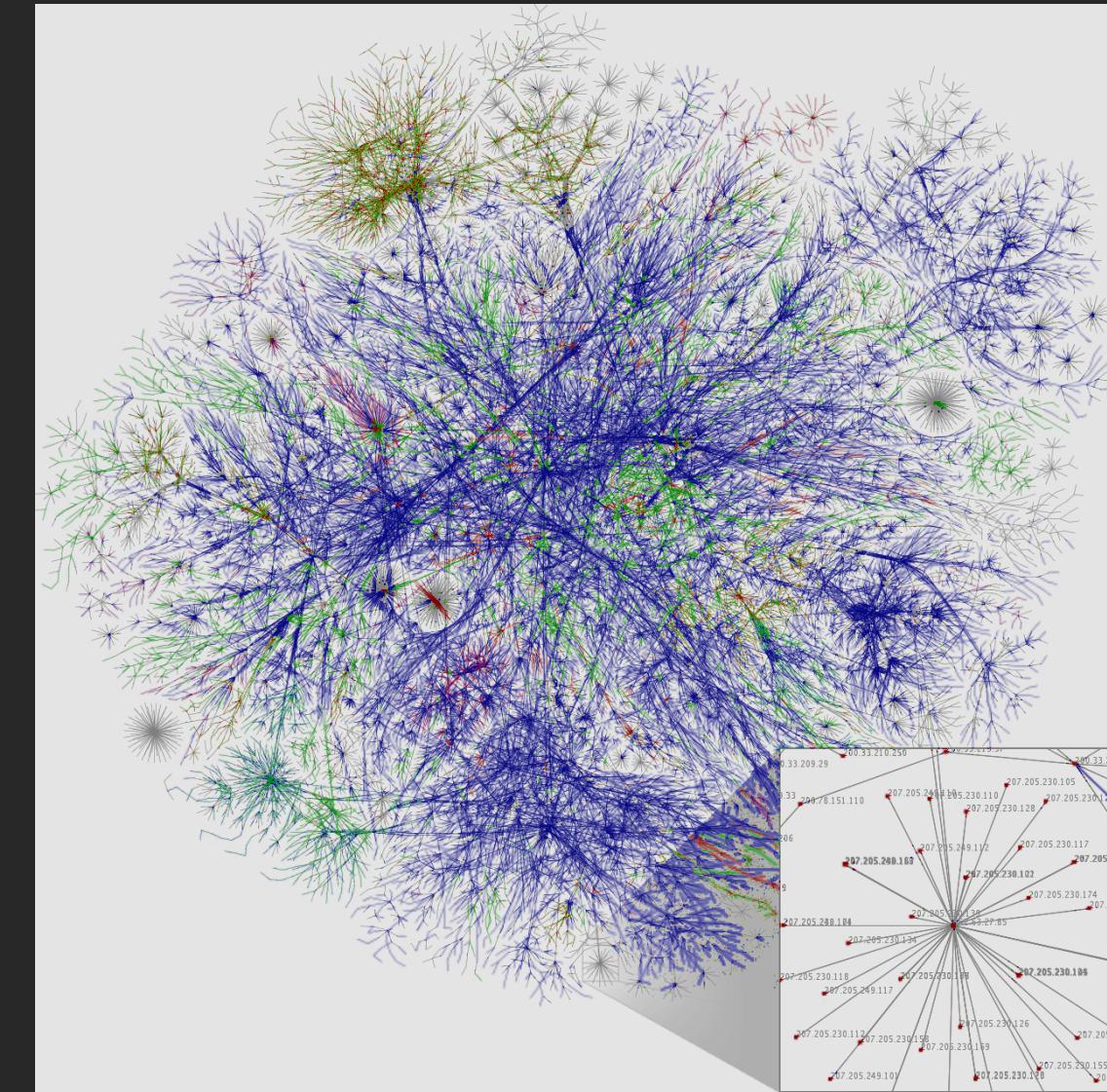
=

2+ Active full stack AWS Regions

Why NOT multi-region?



Replication Lag



Complexity & Cost

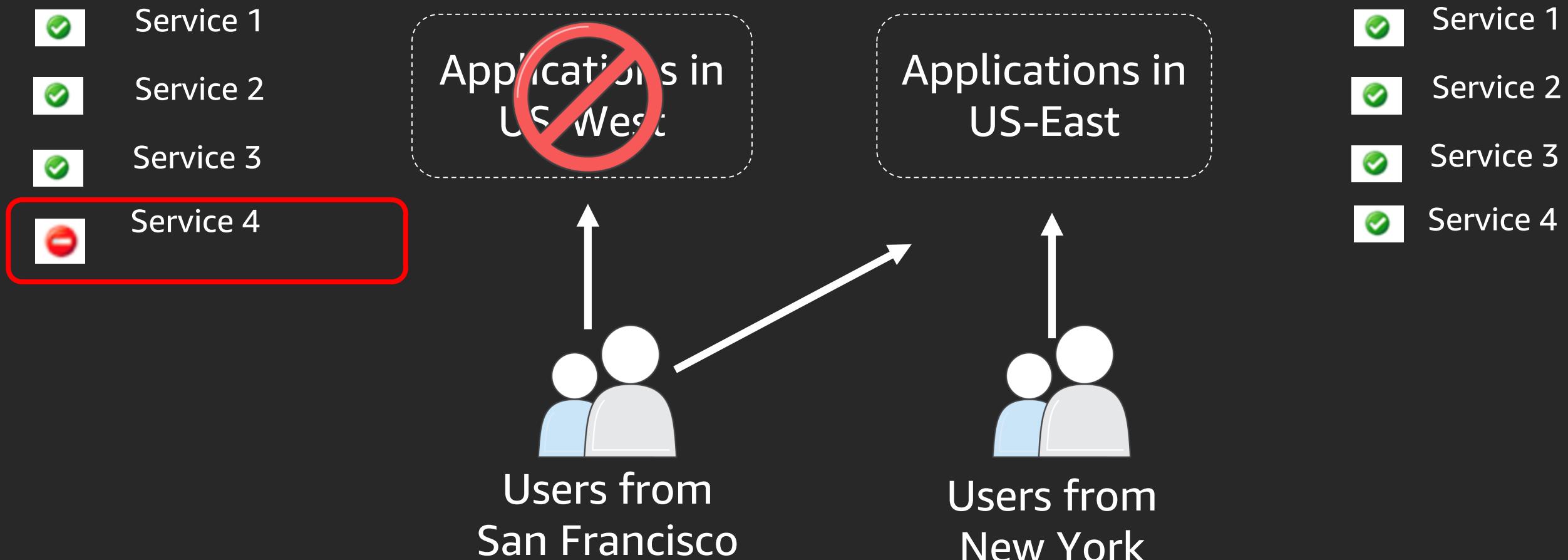
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Why multi-region?

Why multi-region?

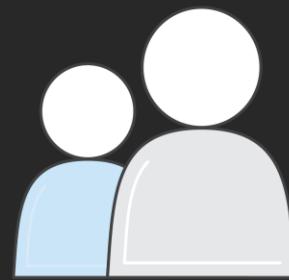
1. Business continuity / disaster recovery



Why multi-region?

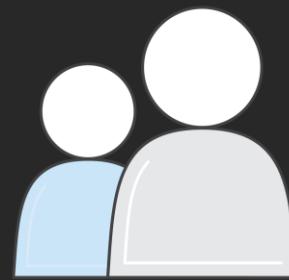
1. Business continuity / disaster recovery
2. Geographically distributed customer base

US-West



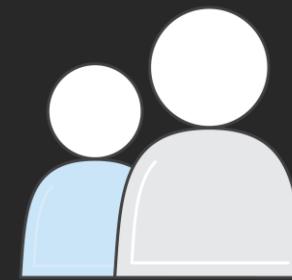
Users from
San Francisco

US-East



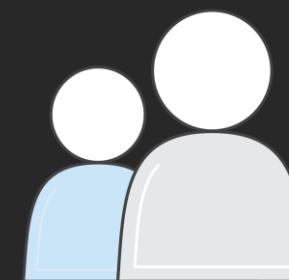
Users from
New York

EU-West



Users from
London

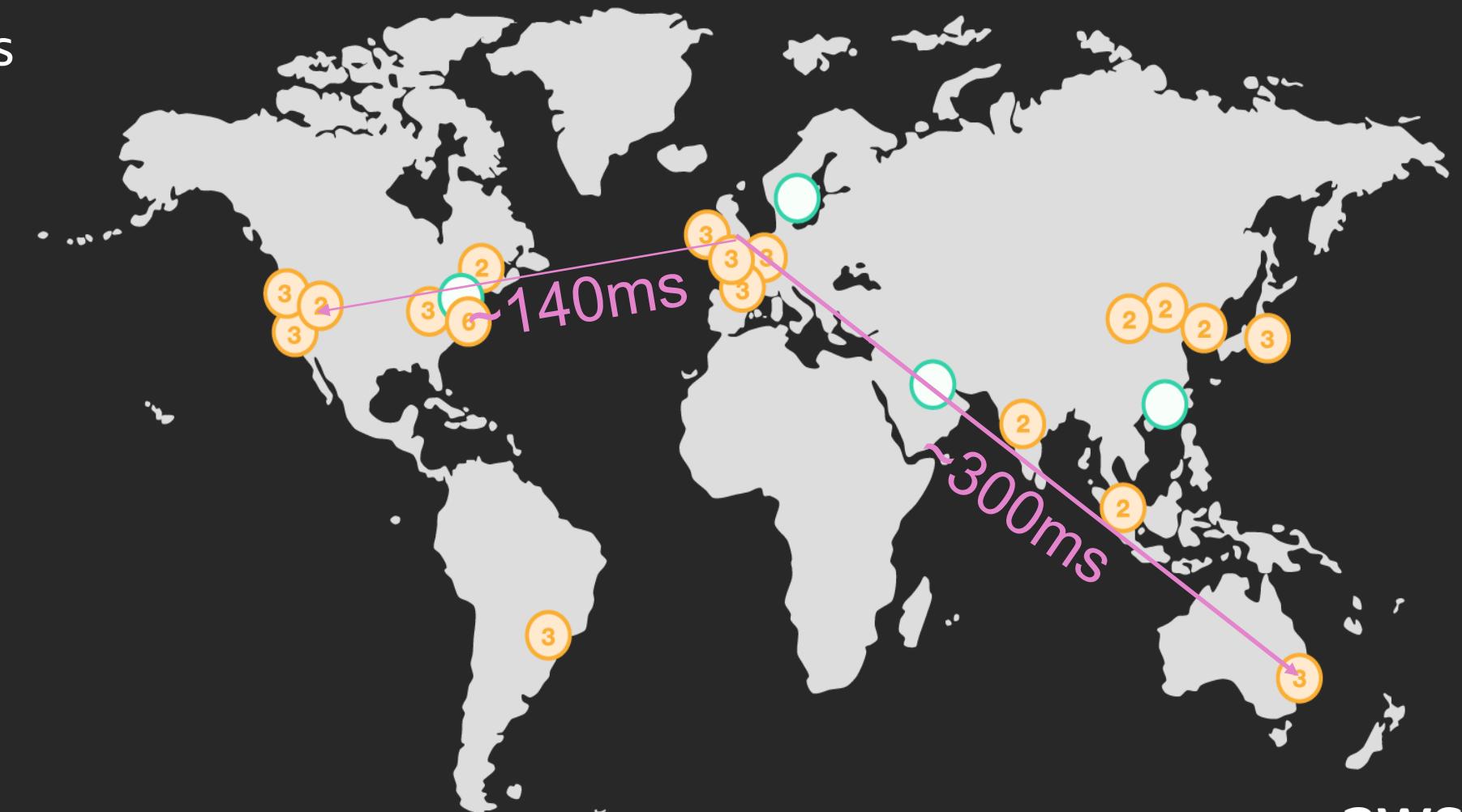
China



Users from
Shanghai

Why multi-region?

1. Business continuity/disaster recovery
2. Geographically distributed customer base
 - a. Improve latency for end-users



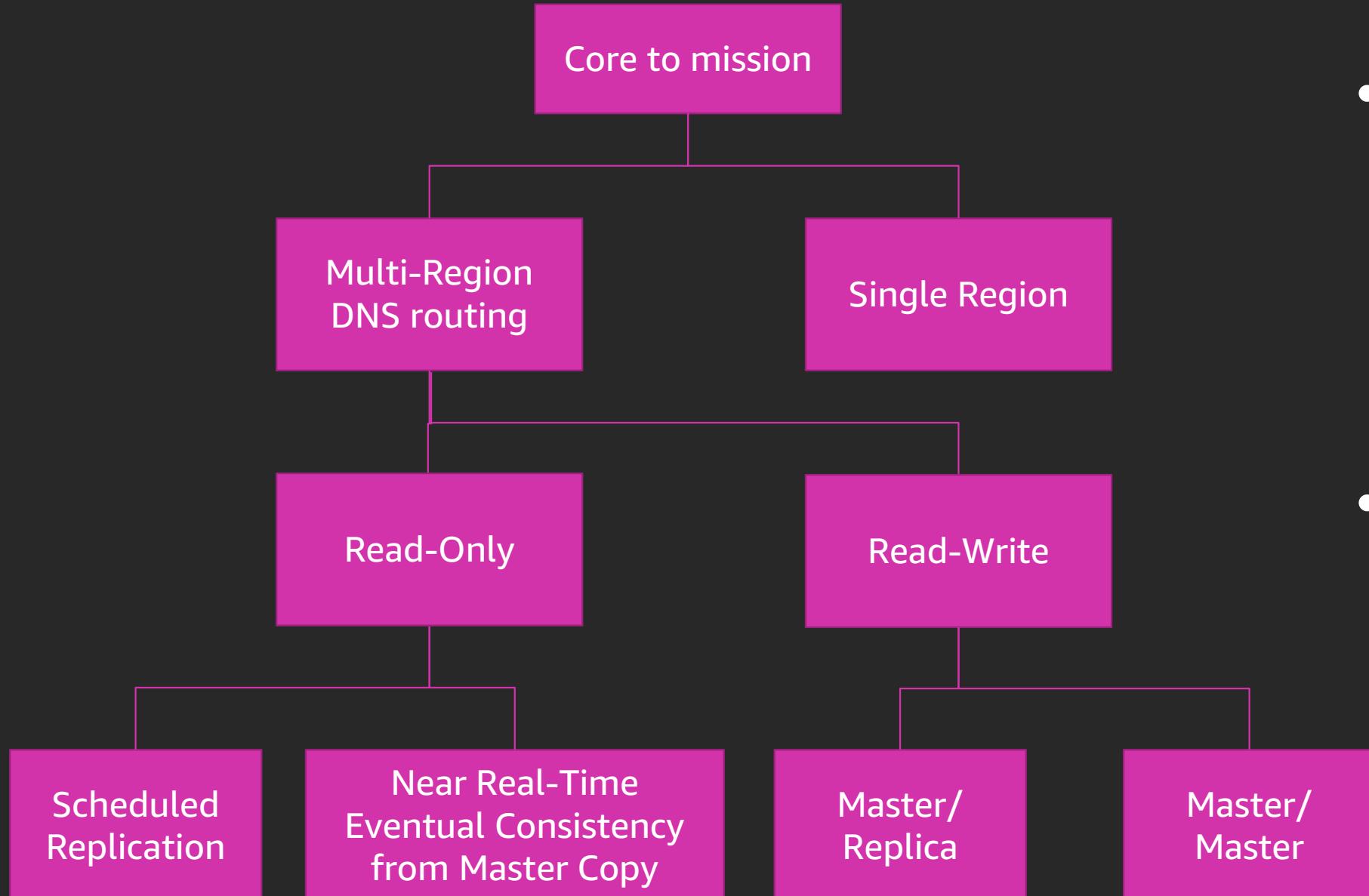
* Latency numbers are only examples

Why multi-region?

1. Business continuity/disaster recovery
2. Geographically distributed customer base
 - a. Improve latency for end-users
 - b. Meet legal and data regulatory compliance

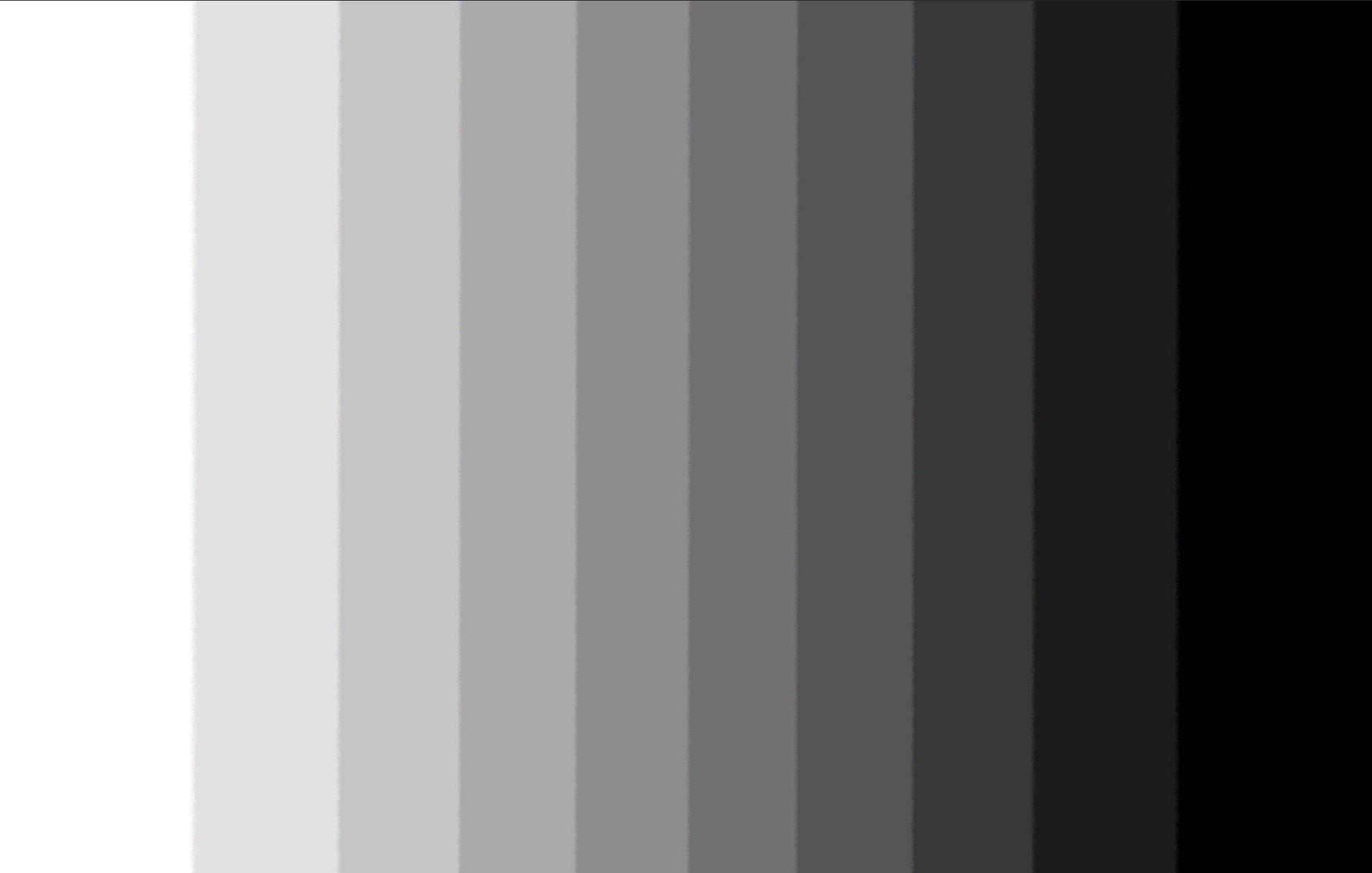


Does it need to be multi-region?



- Can we be successful **without this service?**
How much redundancy do we need?
- How does this service access **persistent storage?**

Deciding multi-region is *not* black or white



multiRegionArchitecture

!=

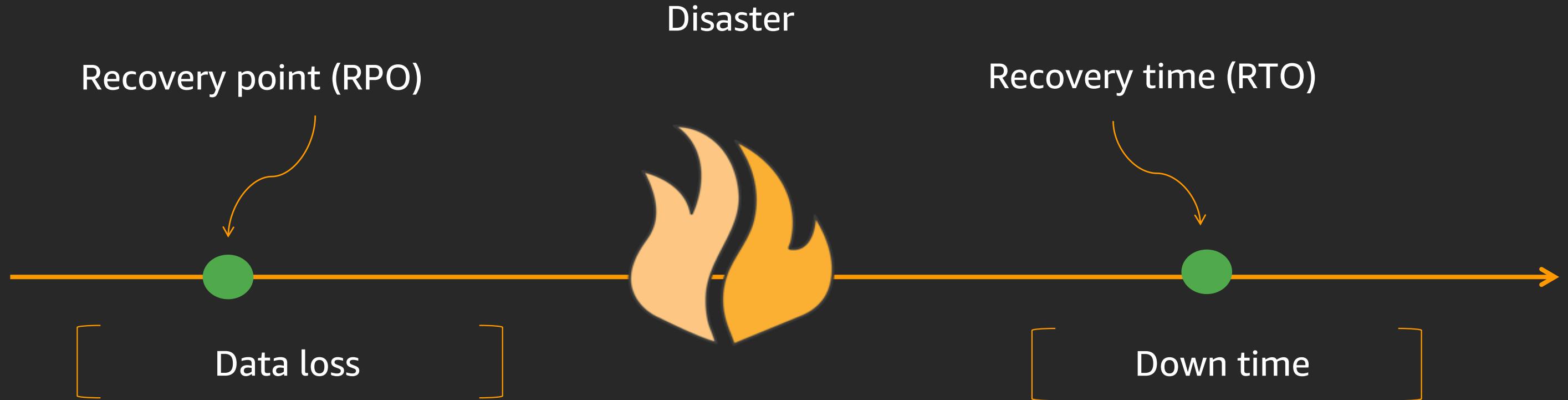
multiRegionActiveActive

How do I design for *multi-region business continuity?*

Business continuity

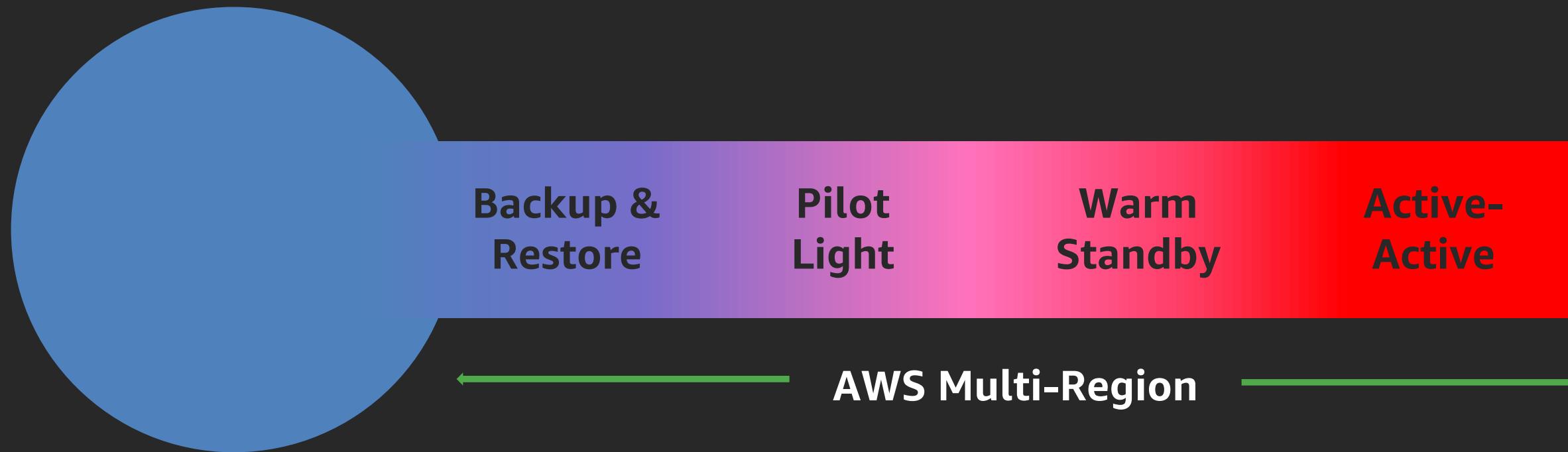
How much data can you afford to recreate or lose?

**How quickly must you recover?
What is the cost of downtime?**

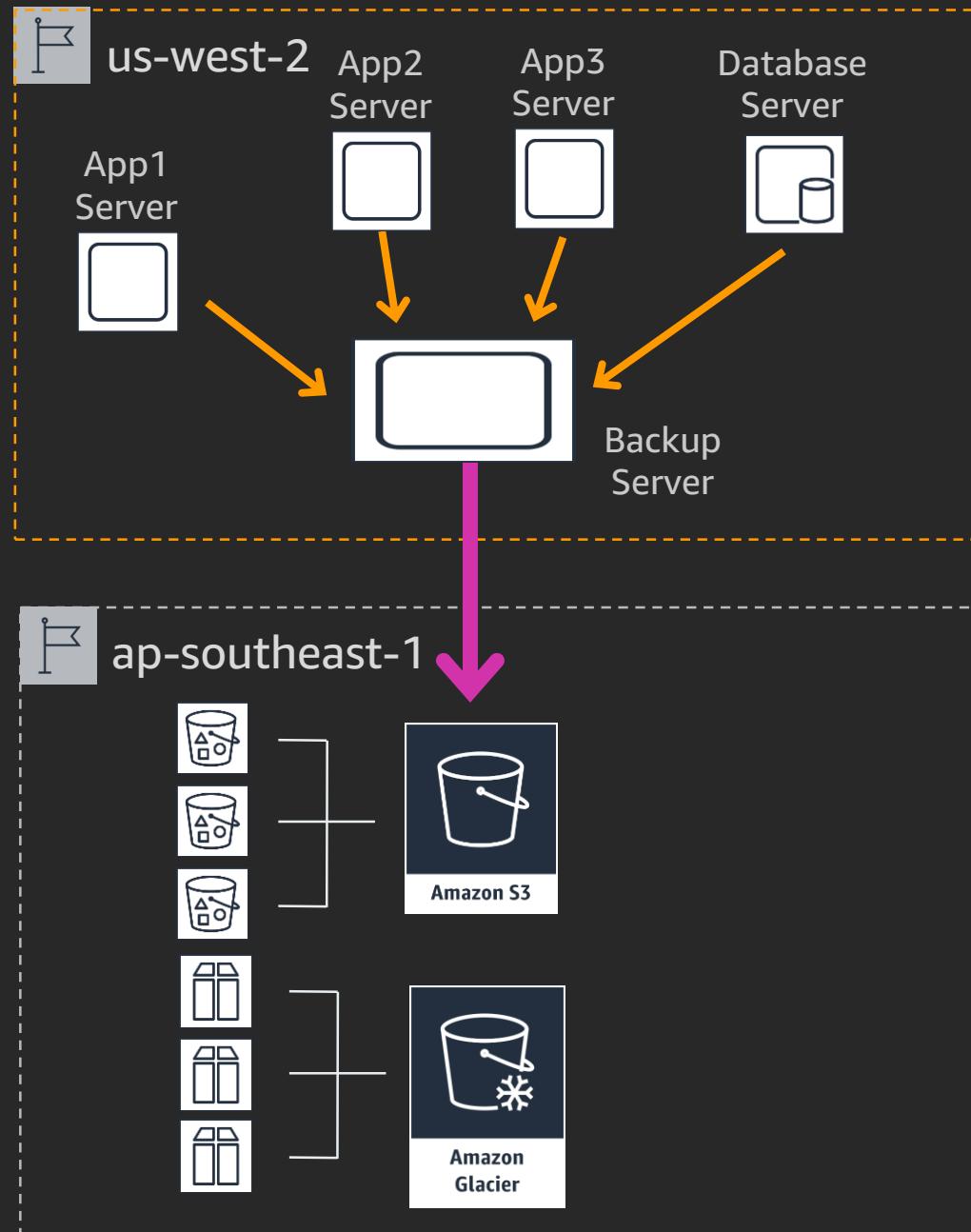


*It's not about the data, it's about the **mission***

The four strategies for business continuity

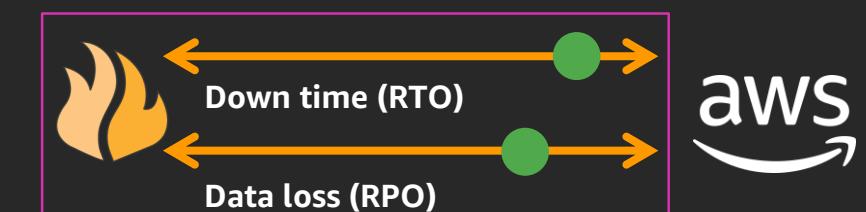


Strategy: Backup & restore (multi-region)



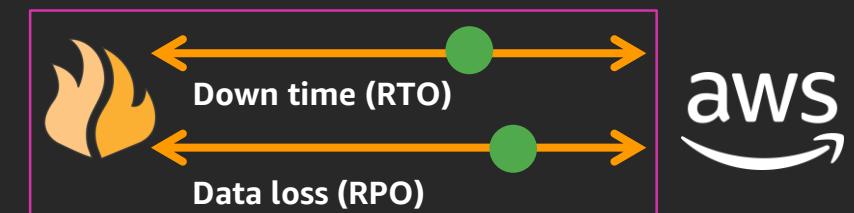
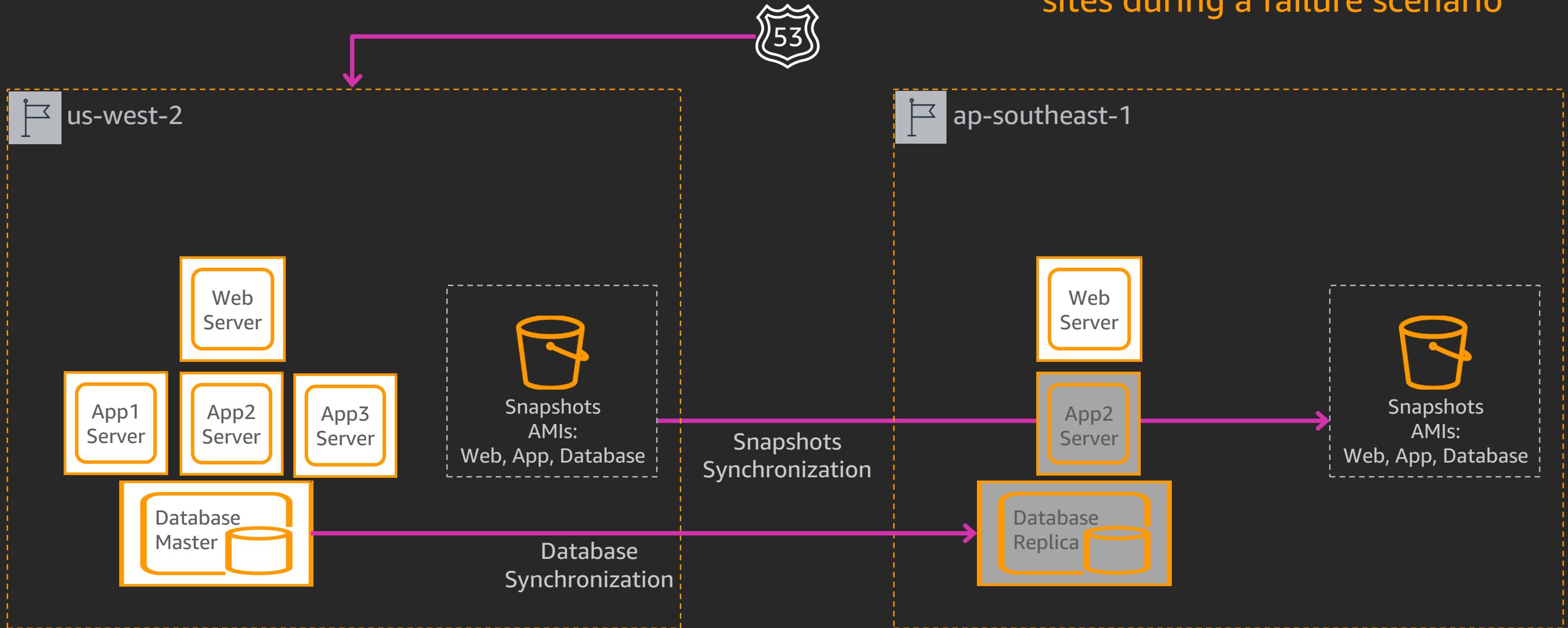
Backup to another region

- Use managed database services with Amazon Simple Storage Service (Amazon S3) or Amazon Glacier
- Data stored with high durability in multiple locations



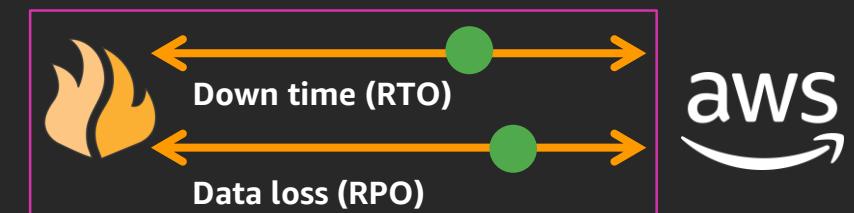
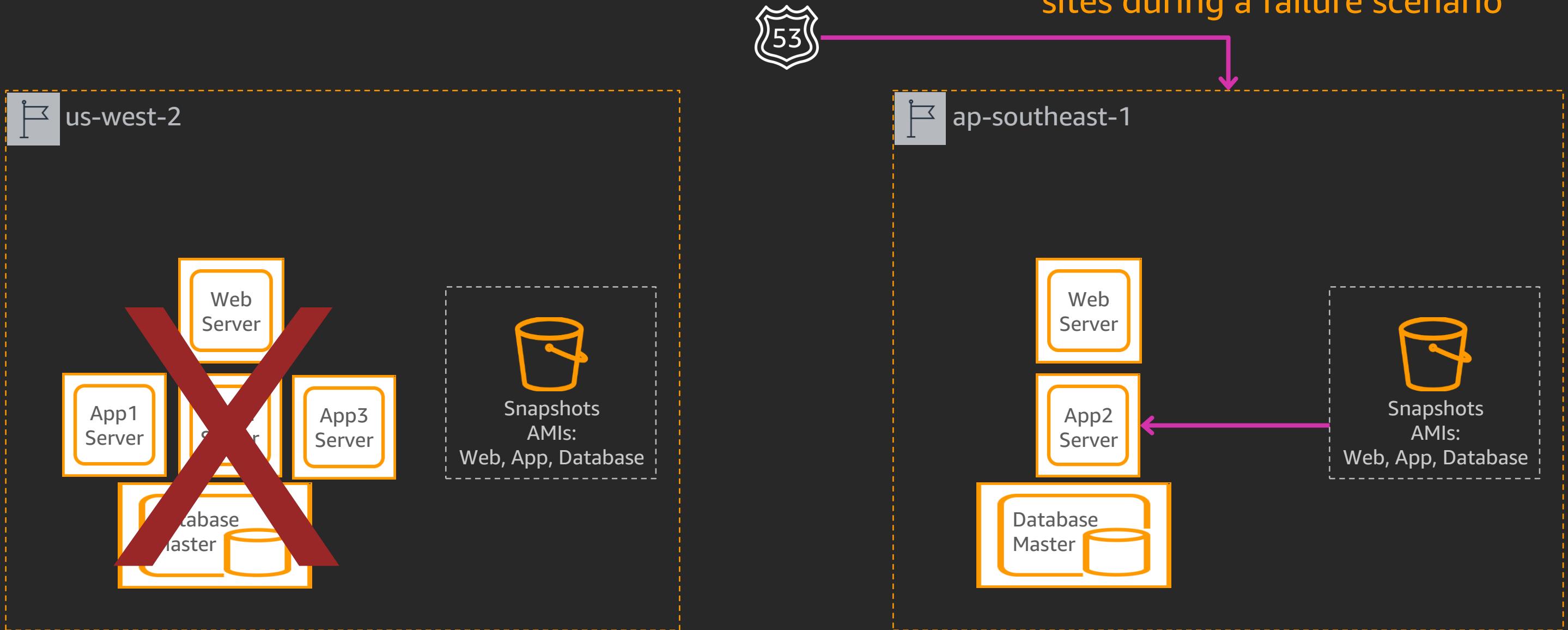
Strategy: Pilot light (multi-region)

Allows the scaling of redundant sites during a failure scenario

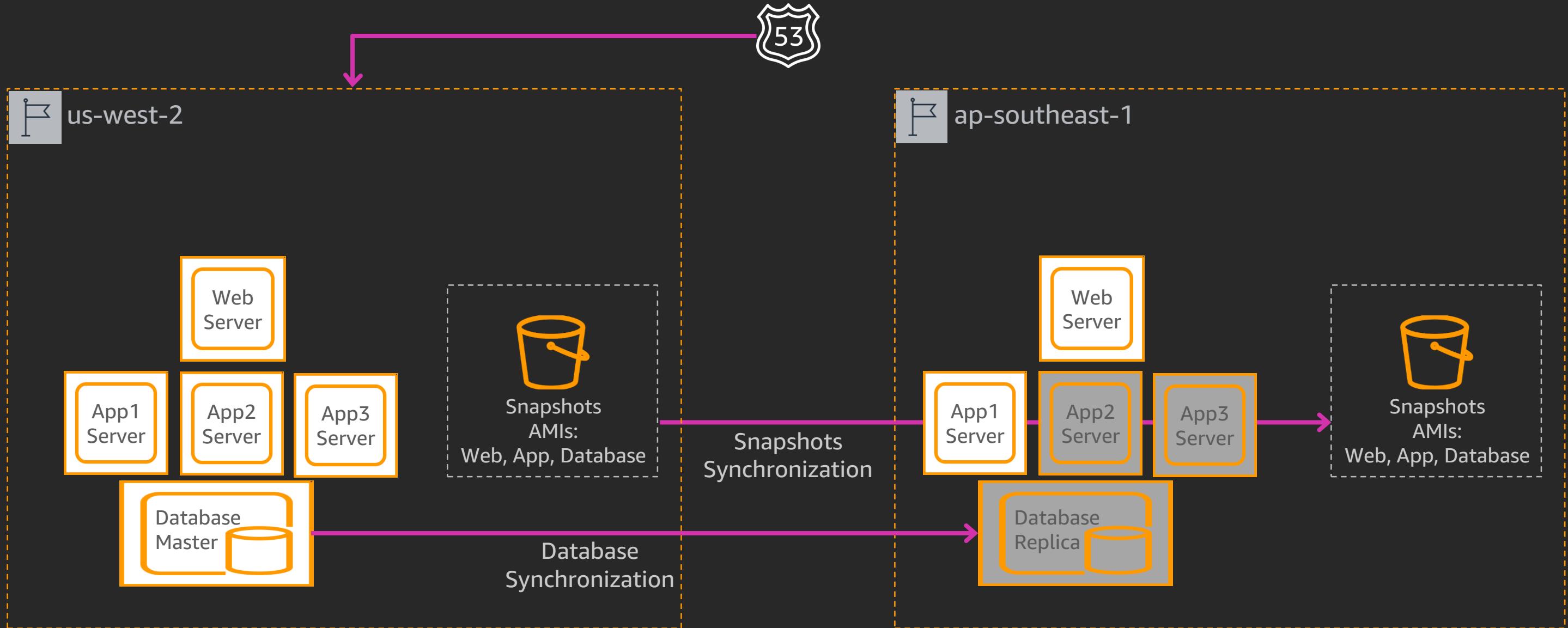


Strategy: Pilot light (multi-region)

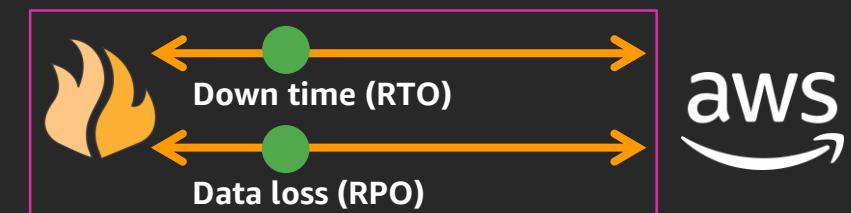
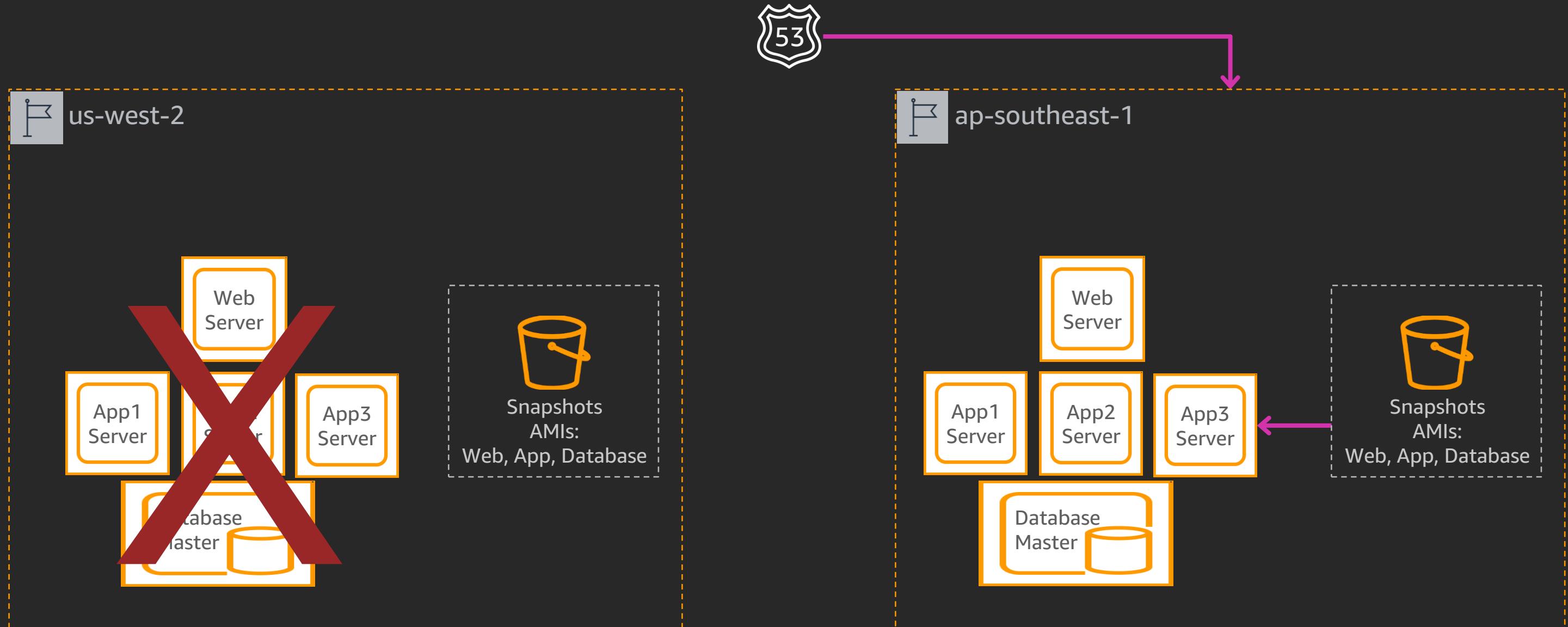
Allows the scaling of redundant sites during a failure scenario



Strategy: Warm Standby (Multi-Region)

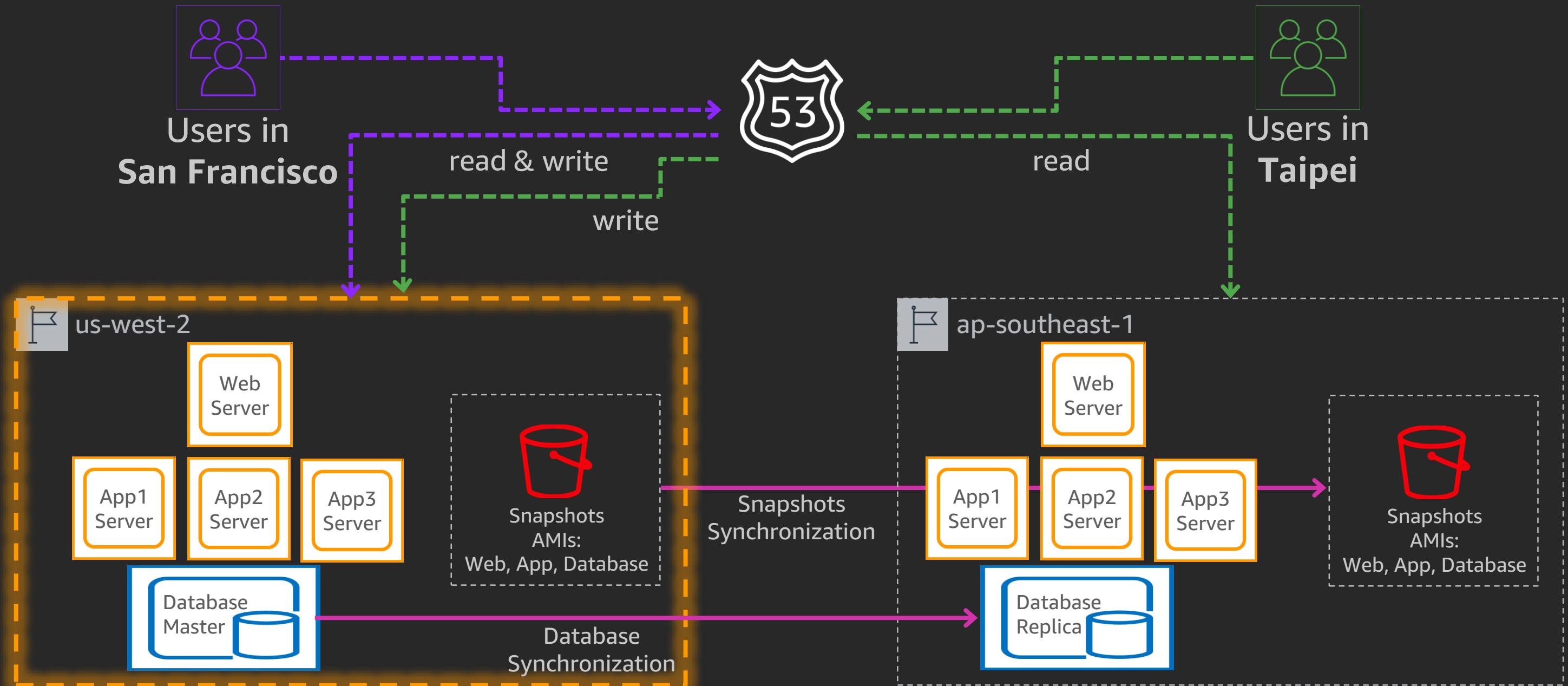


Strategy: Warm standby (multi-region)

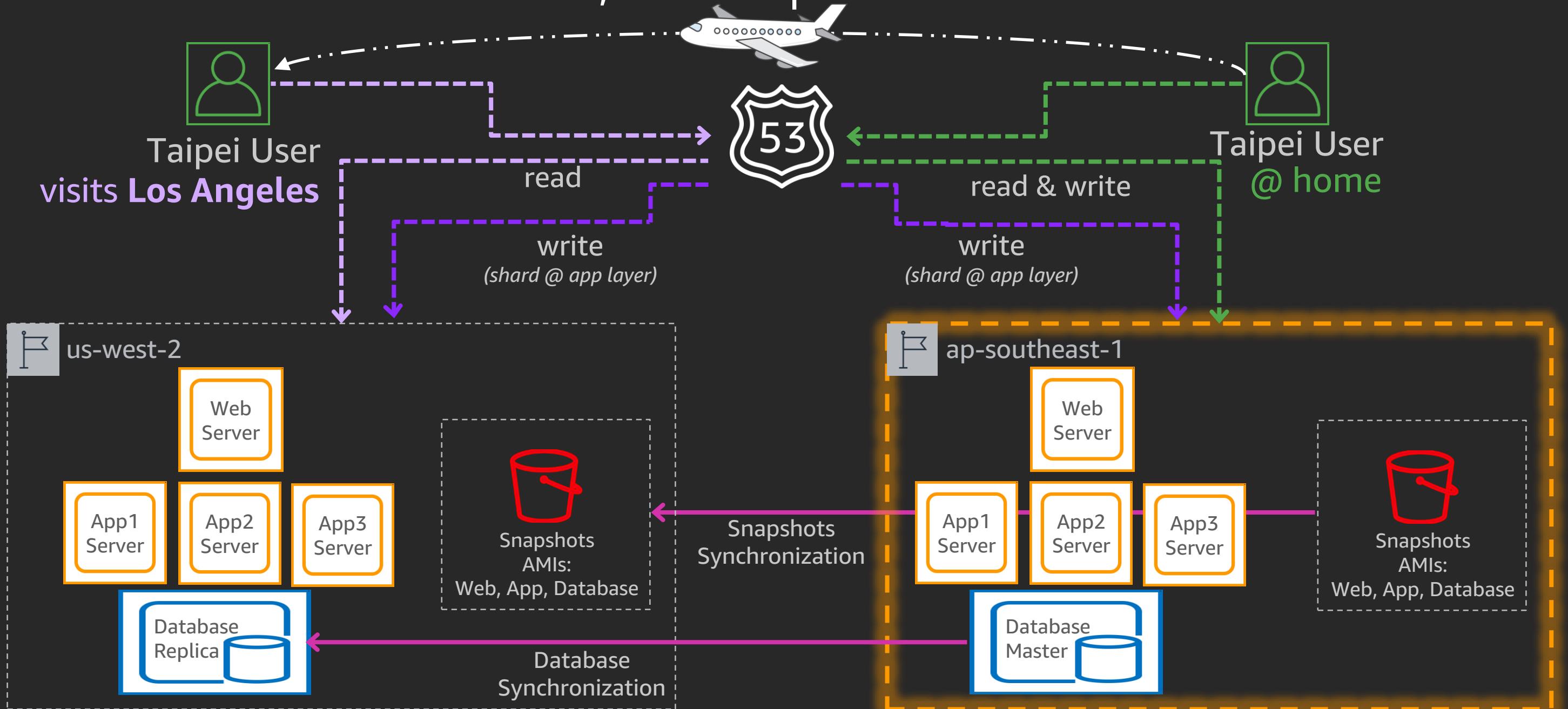


How do I re-design
a single region to multi-region active-active?

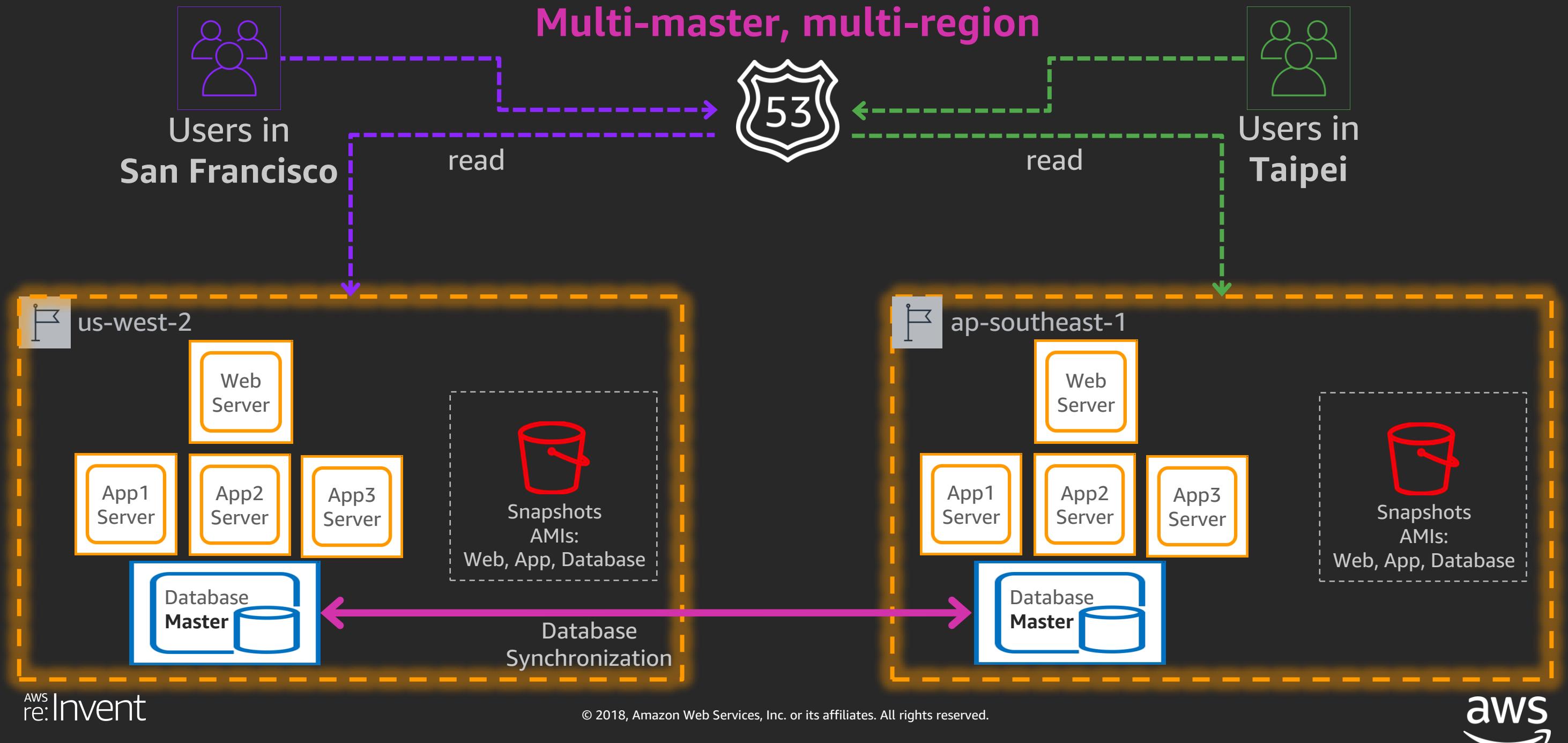
Pattern 1: Read local, write global



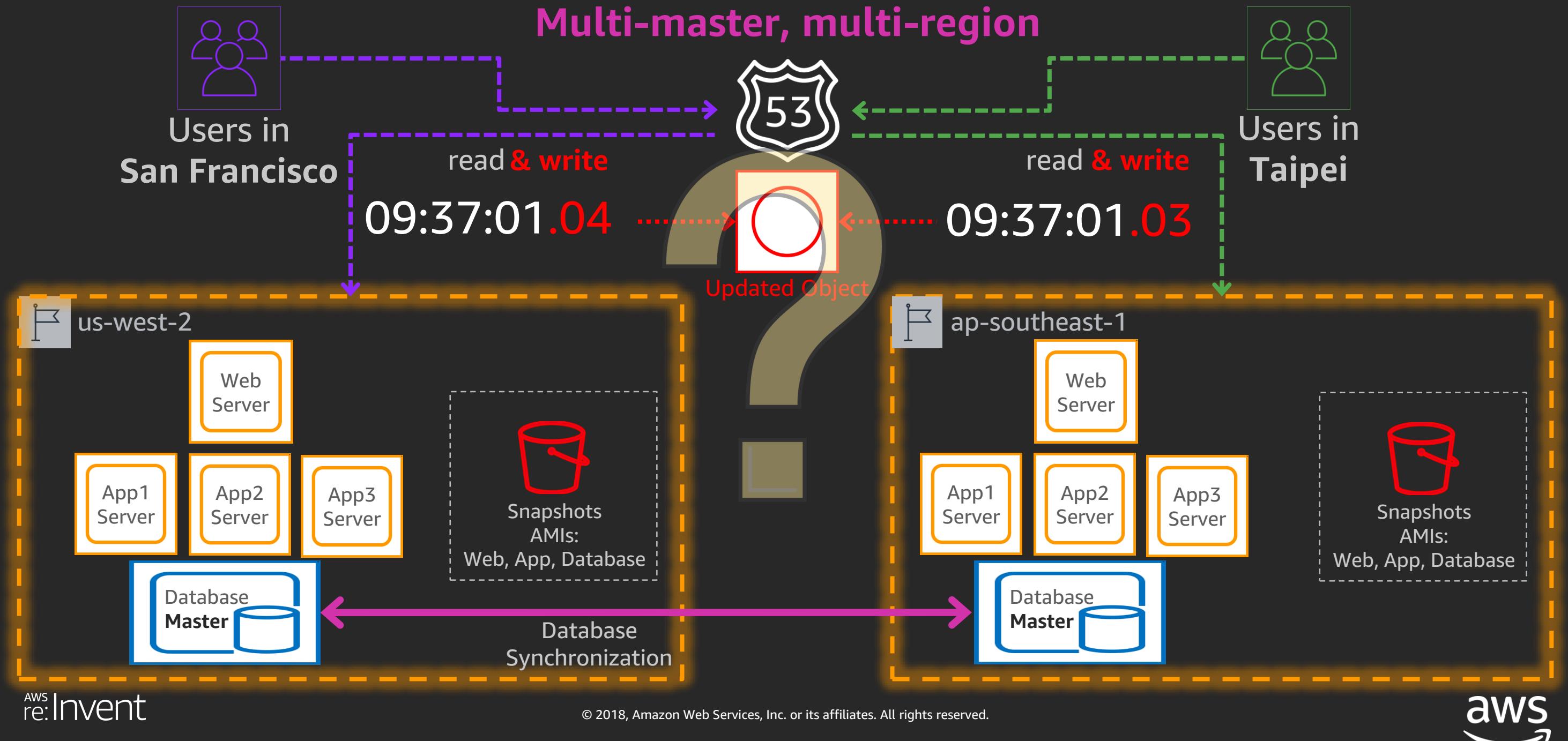
Pattern 2: Read local, write partitioned



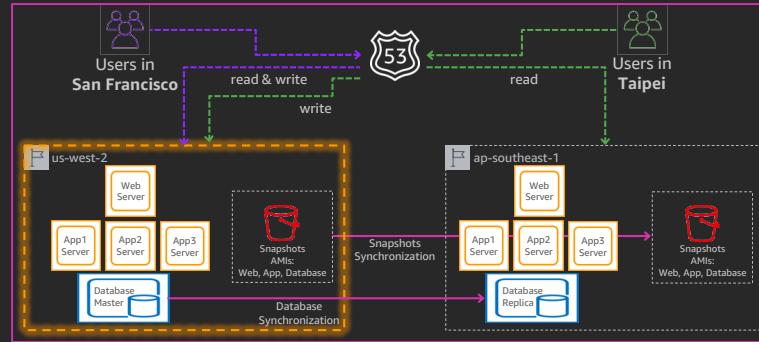
Pattern 3: Read local, write local



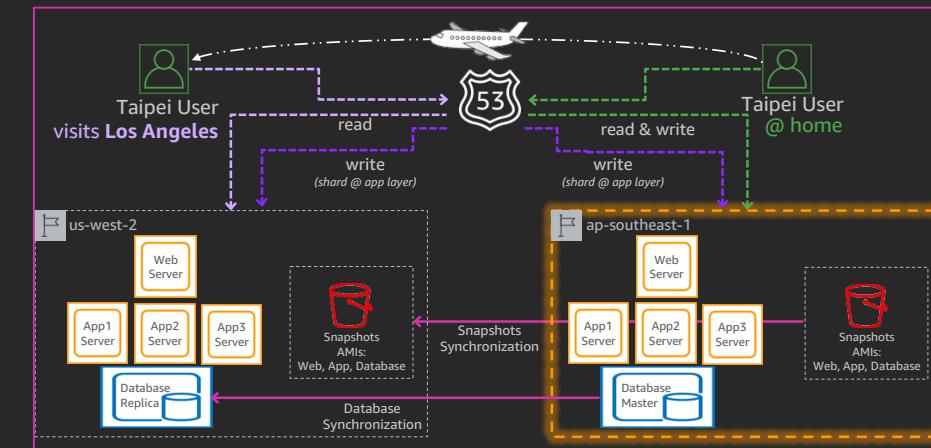
Pattern 3: Read local, write local



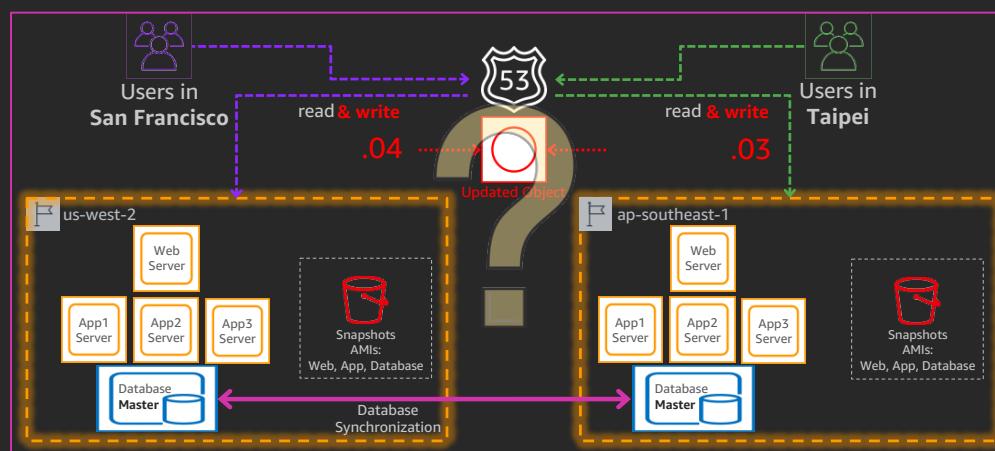
Multi-region active-active patterns



Read local, write global

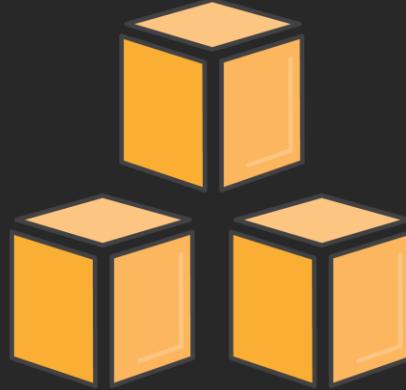


Read local, write partitioned

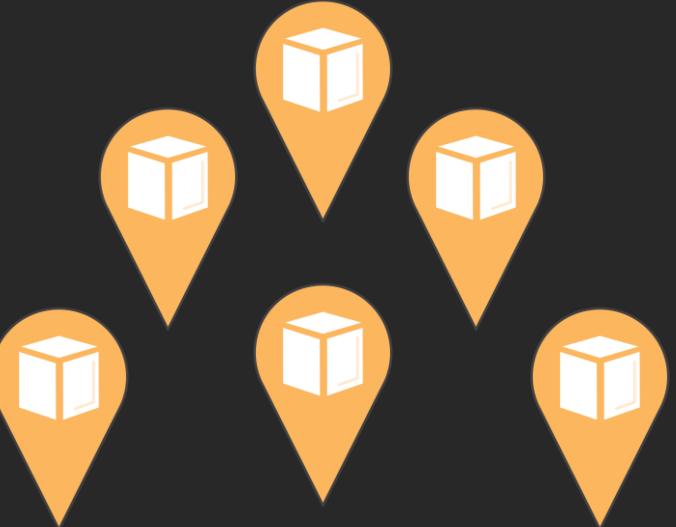


Read local, write local

Amazon S3 Availability Zones



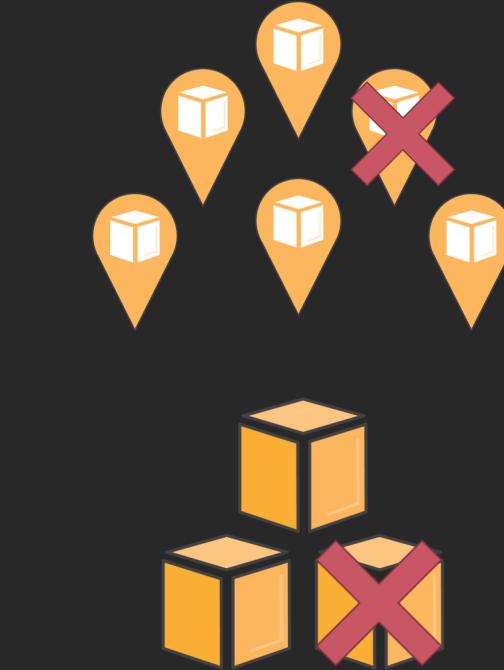
S3 stores data in at least 3 Availability Zones (AZ's)



Each AZ can be up to 8 physical data centers

Physically separate – even extremely uncommon disasters would only affect a single AZ

Low latency private network connect data centers and AZ's



Unavailability of a data center or an AZ does not impact overall S3 availability

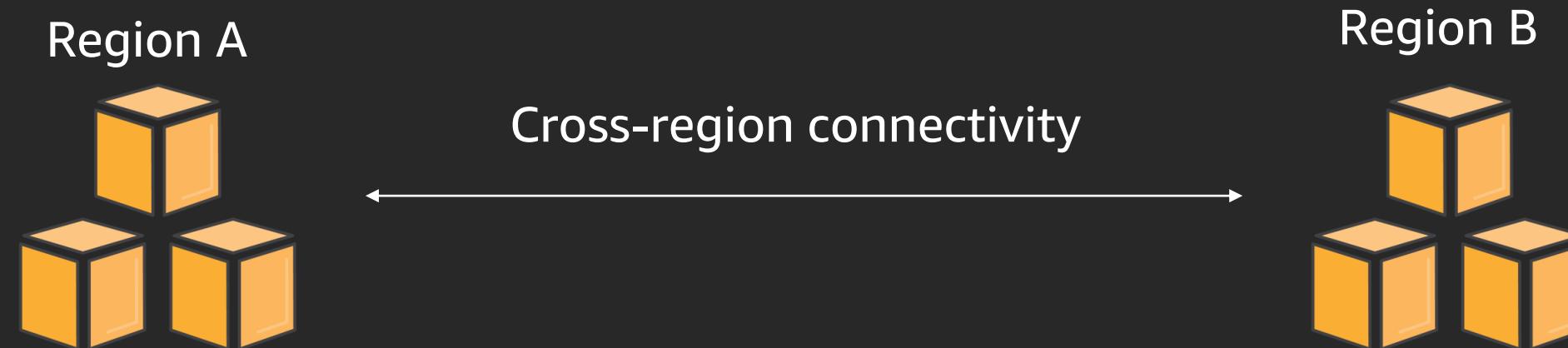
Data is automatically distributed across a minimum of 3 AZ's GEO separated within an AWS Region

S3 cross-region replication



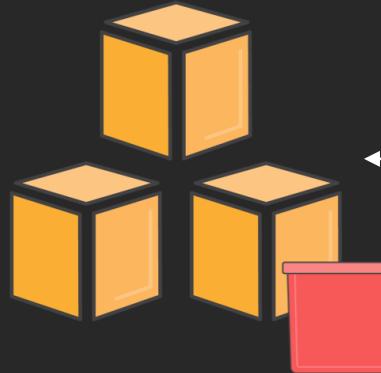
Automatically replicate data to any other AWS Region

- Replicate by object, bucket, or prefix
- Support for server-side encryption on AWS Key Management Service (AWS KMS) encrypted objects
- Ownership overwrite
 - Change the object owner in the destination region

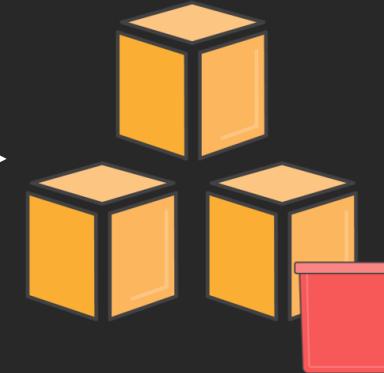


S3 cross-region replication examples

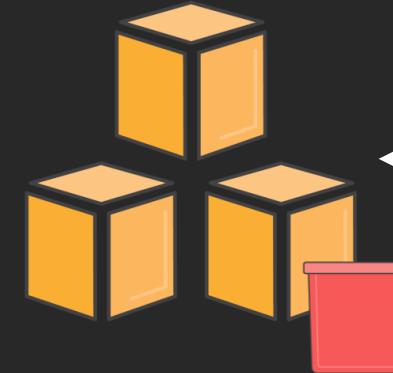
S3 Standard



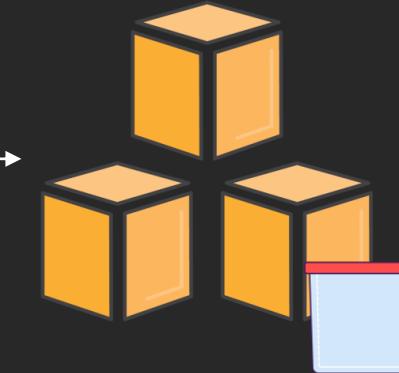
S3 Standard



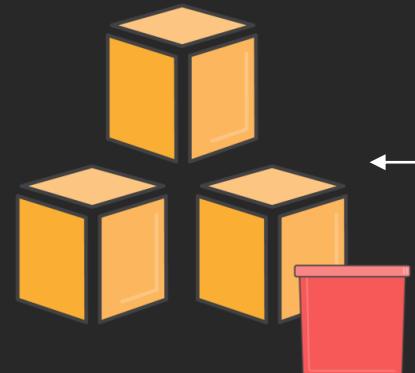
S3 Standard



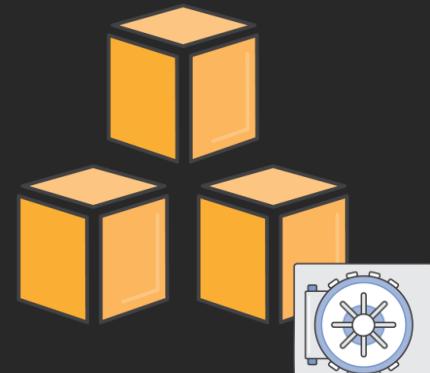
S-IA



S3 Standard

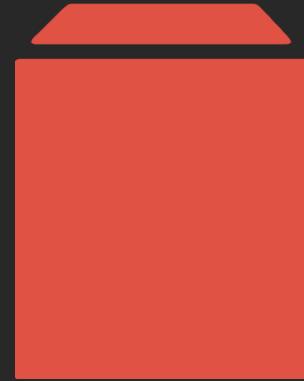


Amazon Glacier

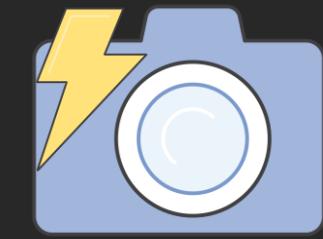


*Zero-day lifecycle policy
to Amazon Glacier*

Amazon Elastic Block Store (Amazon EBS) snapshots



EBS volume



EBS snapshot

Elastic block storage

- Point-in-time backup of modified volume blocks
- Stored in S3, accessed via Amazon EBS APIs
- Subsequent snapshots are incremental
- Deleting snapshot will only remove data exclusive to that snapshot
- Copies in same region or cross region

Amazon DynamoDB global tables

Fully managed, multi-master, multi-region database



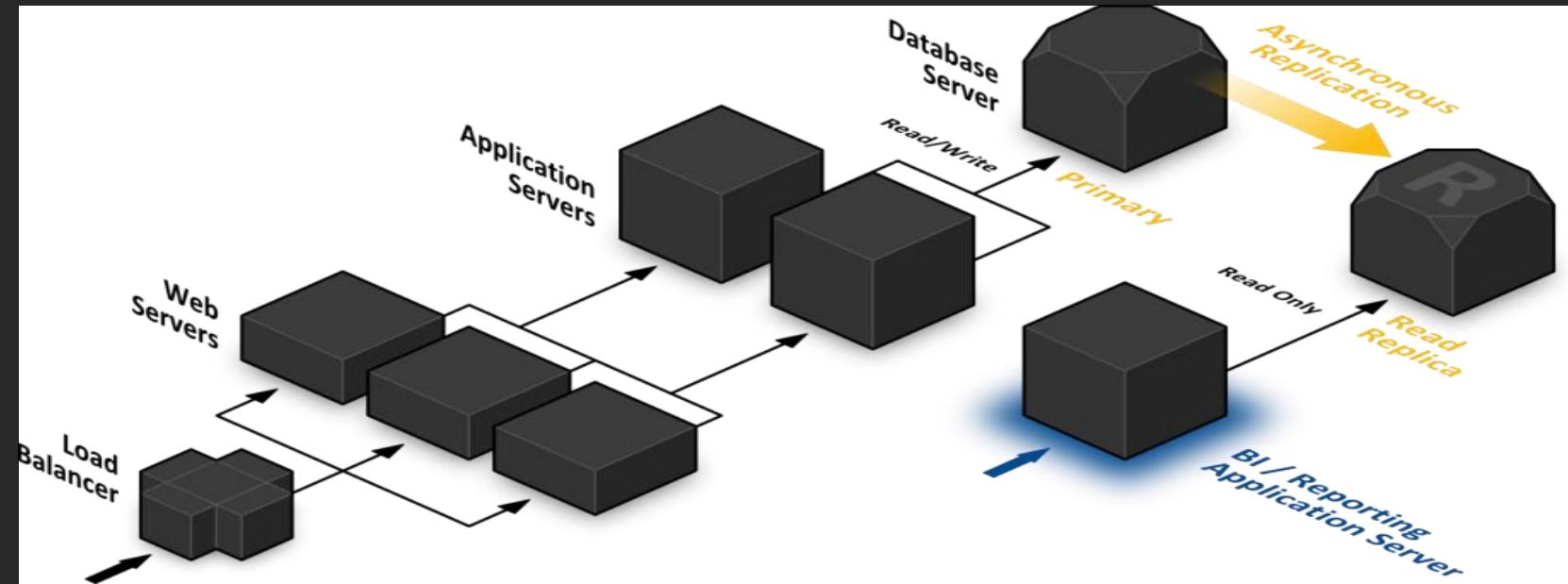
Build high performance, globally distributed applications

Low latency reads & writes to locally available tables

Disaster proof with multi-region redundancy

Easy to setup and no application re-writes required

Cross-region read replicas with Amazon Relational Database Service (Amazon RDS) and Amazon Aurora



- Horizontal scaling of read heavy workloads
- Offload long reporting queries
- Easy to re-create if fallen behind
- Cross-region read replicas bring data close to your applications in different regions
- Promote read replica to a master for faster recovery in the event of disaster

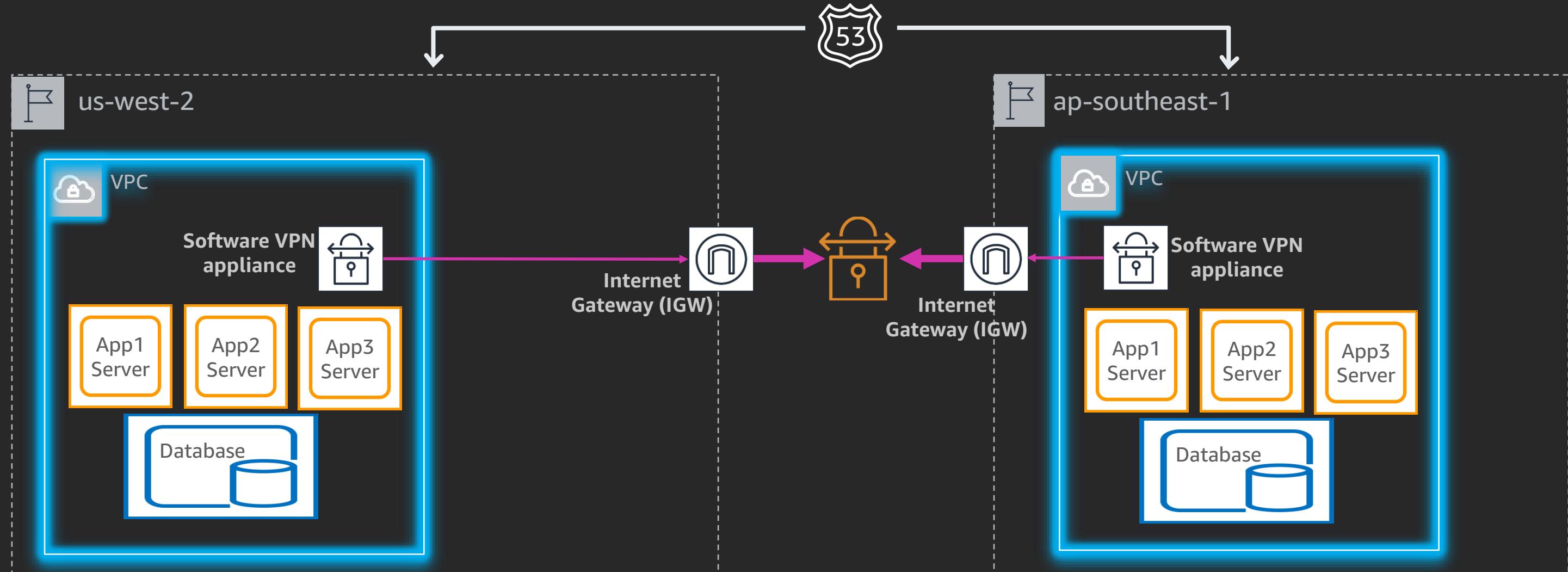
Designing beyond data

Inter-Region Virtual Private Cloud (VPC) Peering

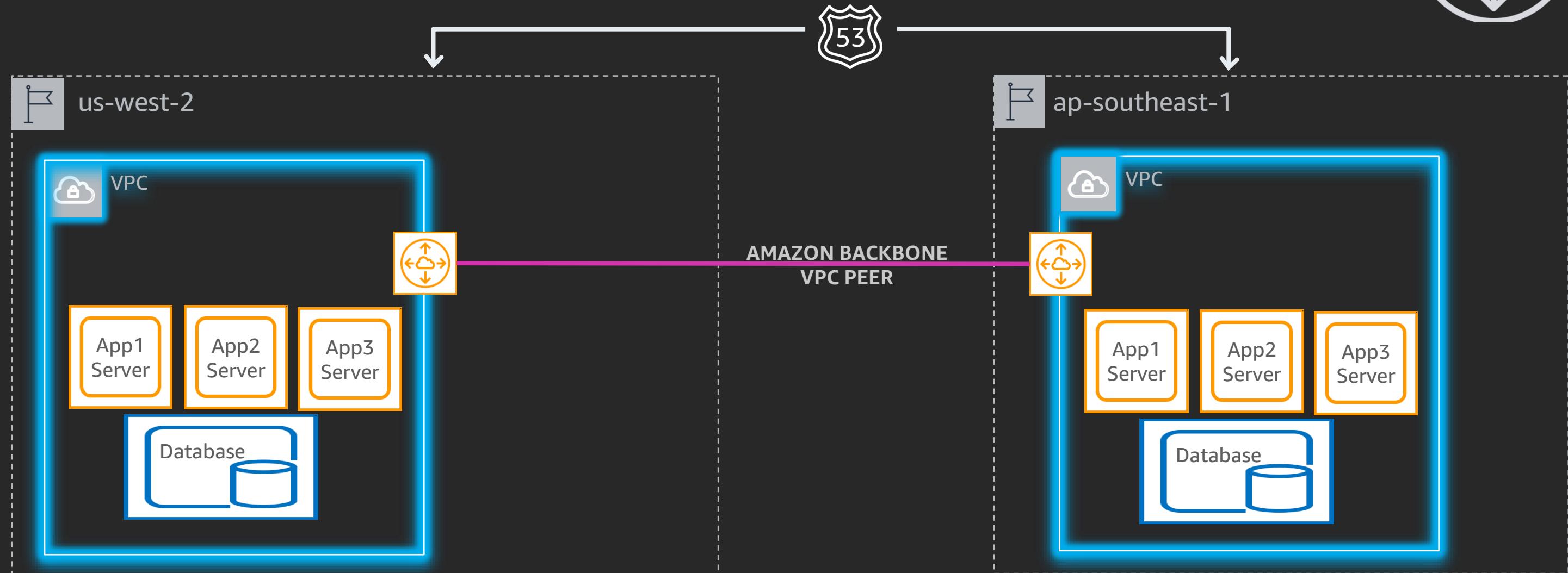
Why is this important to my architecture?



Amazon VPC + VPN



Inter-Region VPC Peering



Some notes...

Inter-Region VPC Peering **encrypts** with no single point of failure or bandwidth bottleneck

Traffic using Inter-Region VPC Peering always **stays on the global AWS backbone**

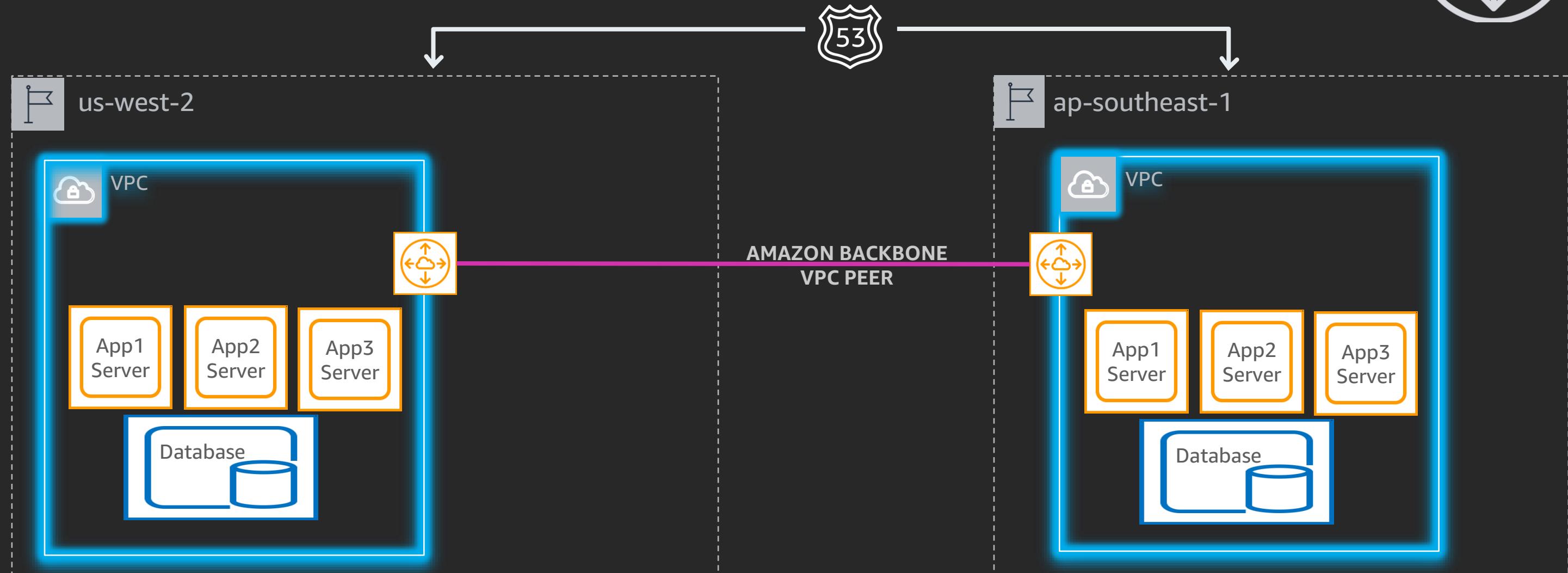
Inter-Region VPC Peering **shares** resources between regions or replicates data for **geographic redundancy**

Multi-VPC connectivity

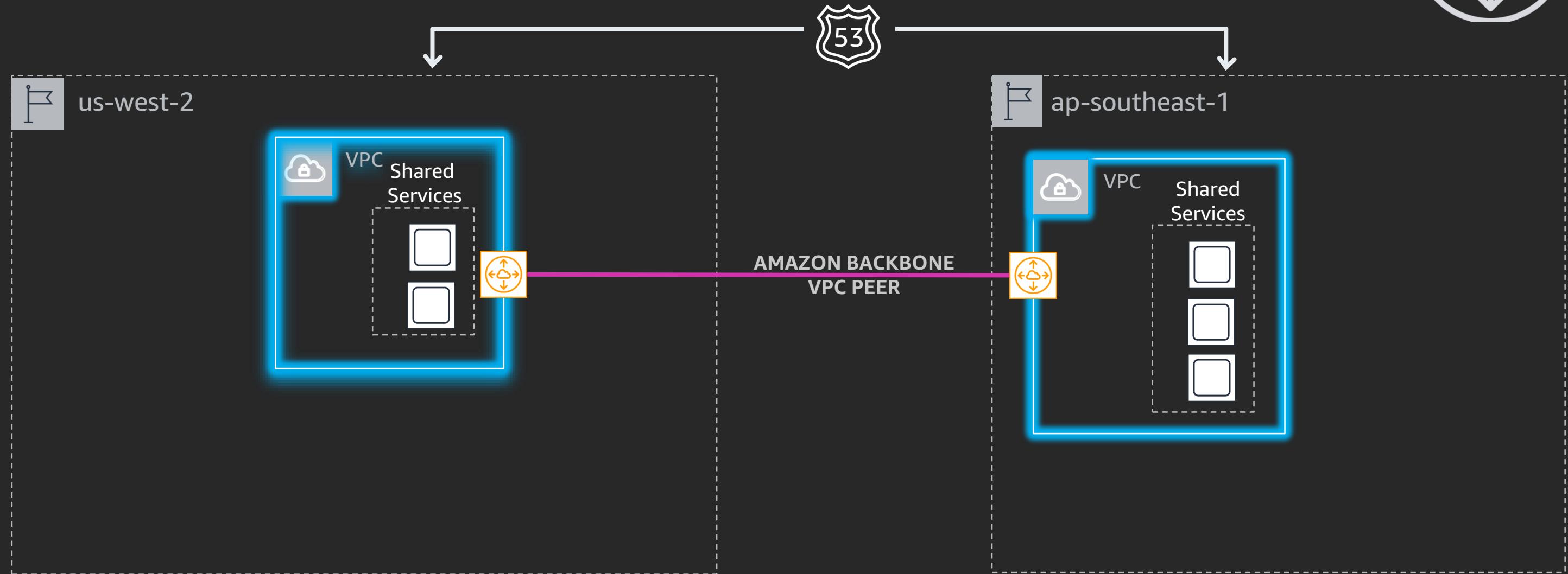
Regions can have multiple VPCs ...
connected?



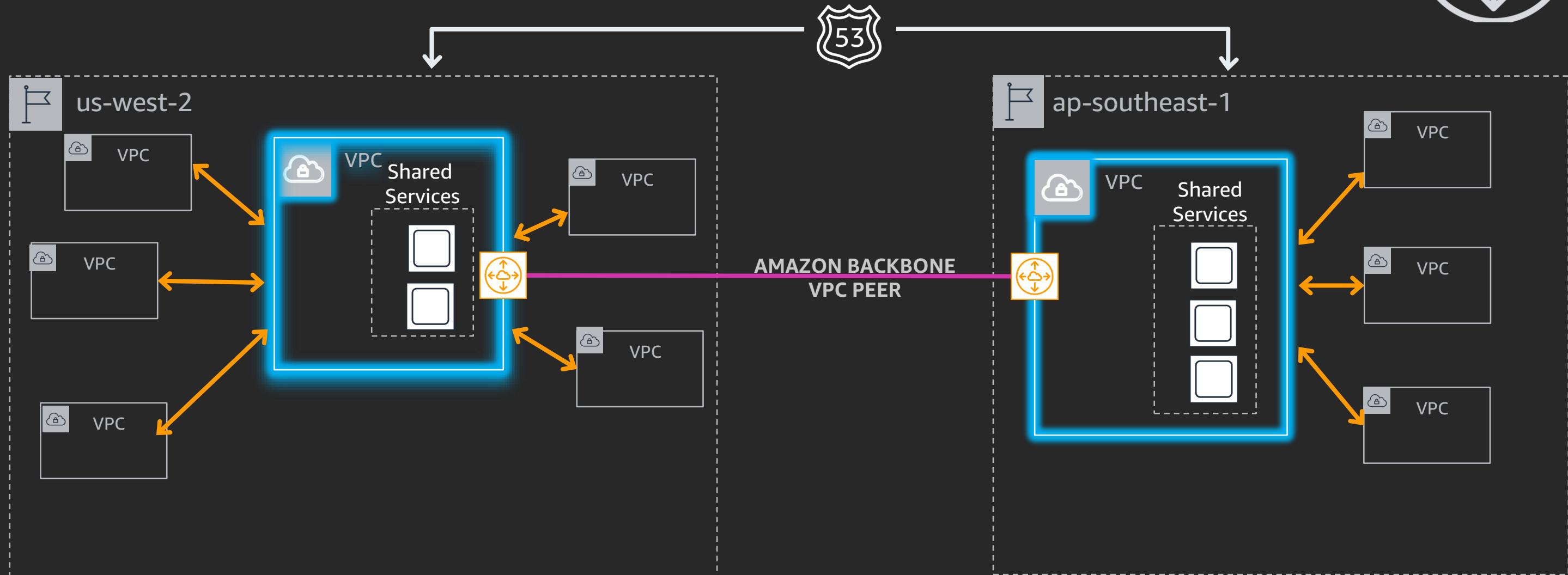
Inter-Region VPC Peering



Inter-Region VPC Peering



Multi-region multi-VPC connectivity



Amazon Route 53

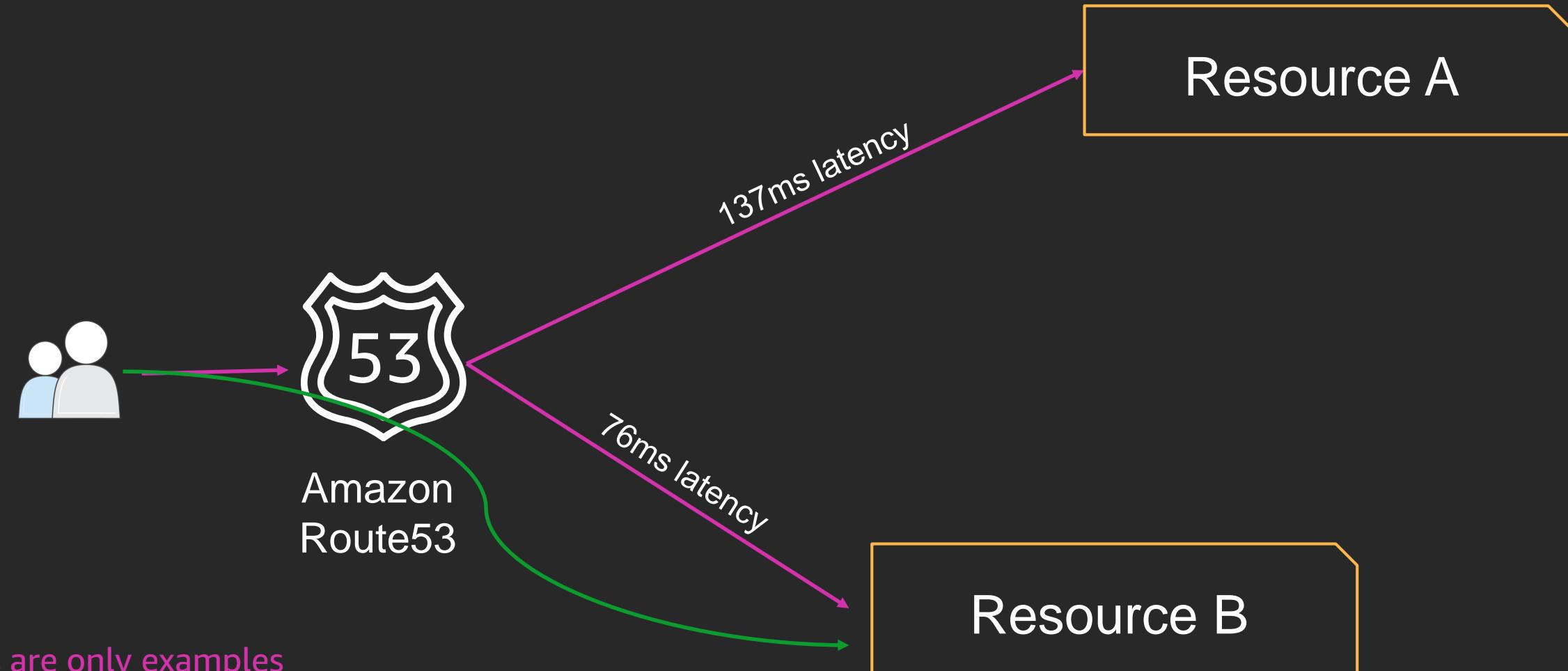
Why is this important to my architecture?



Traffic routing with Route 53



Latency-based routing



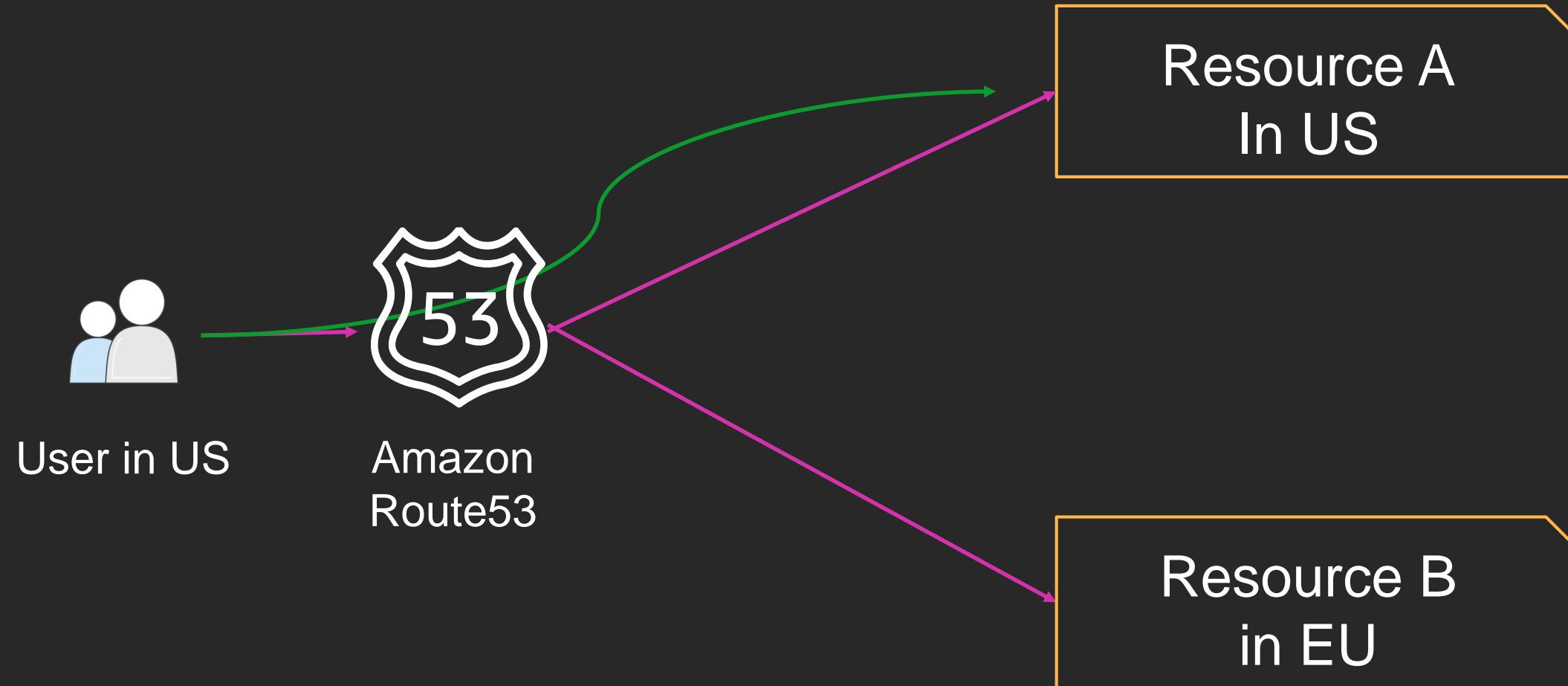
* Latency numbers are only examples

Traffic routing with Route 53



Latency-based routing

Geolocation routing



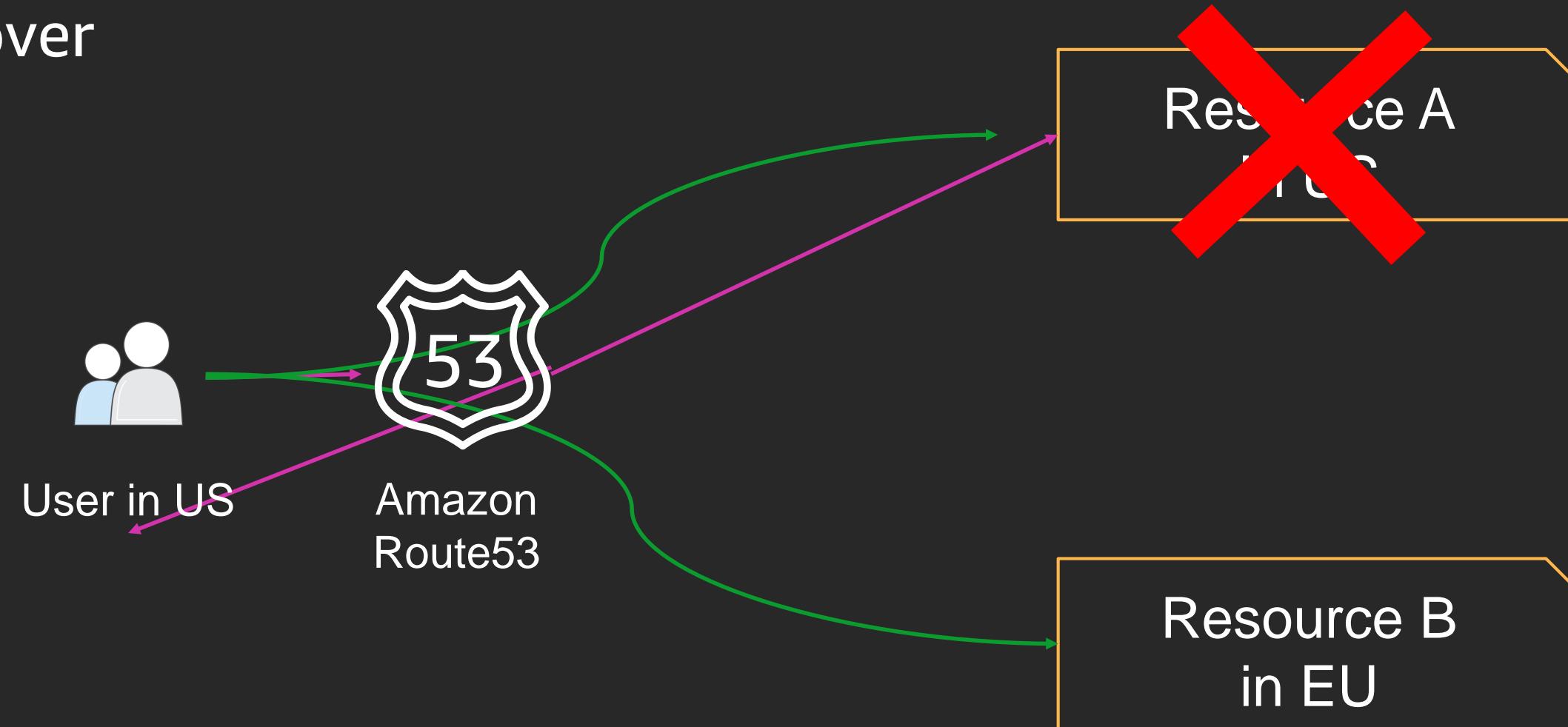
Traffic routing with Route 53



Latency-based routing

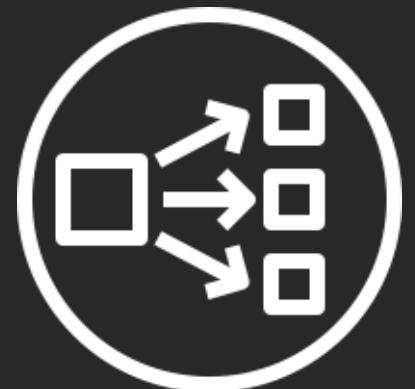
Geolocation routing

DNS failover

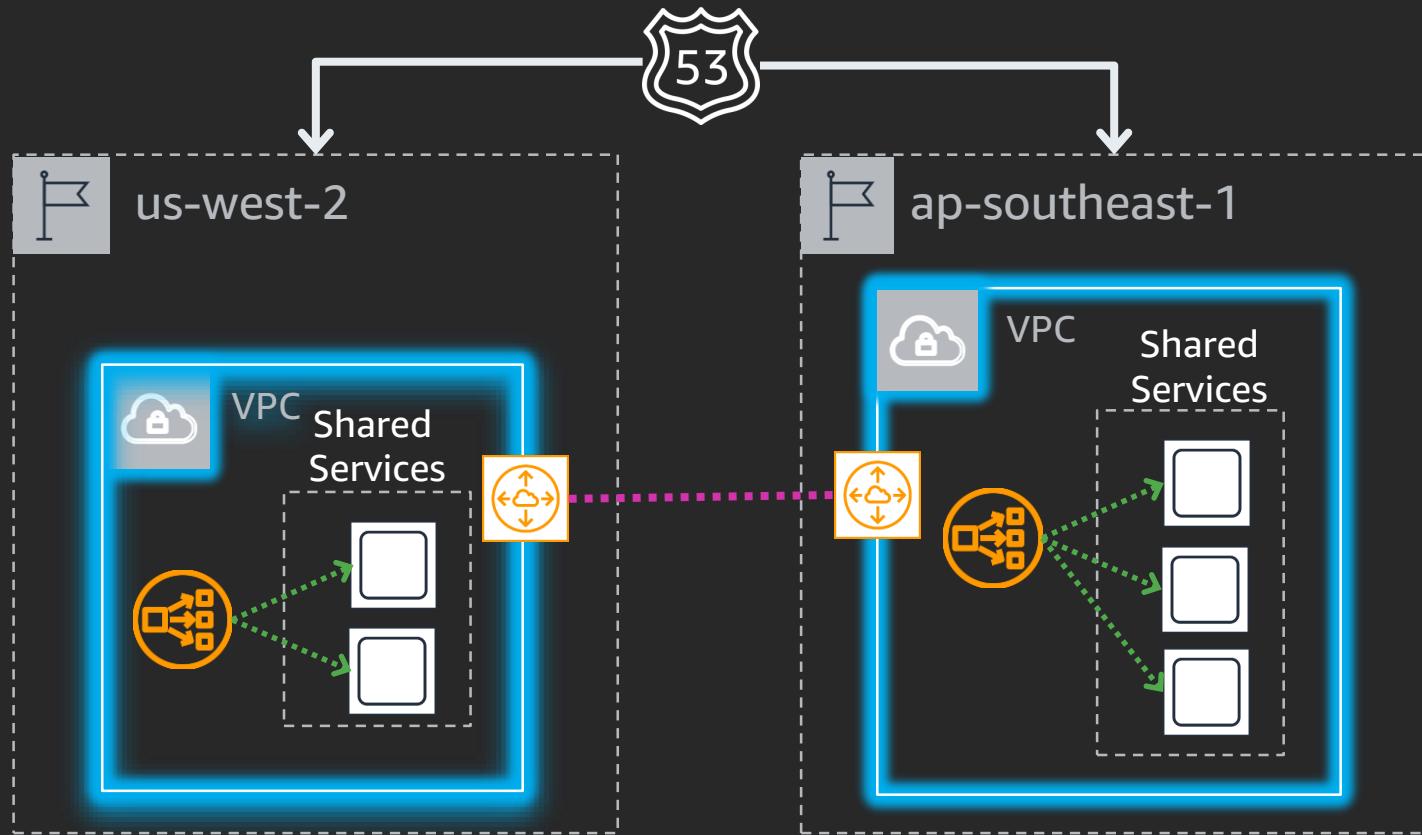
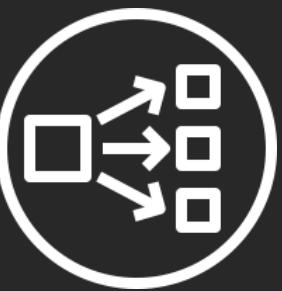


Amazon Network Load Balancer (NLB)

Why is this important to my architecture?



NLB + Inter-Region VPC Peering



- Access NLBs over an inter-region peered VPC
- Load balance to IP-based targets deployed in an inter-region peered VPC
- Support connections from clients to IP-based targets in peered VPCs across different Regions
- Route 53 health checks allow for traffic to be shifted away from failed instances or regions

Some notes...

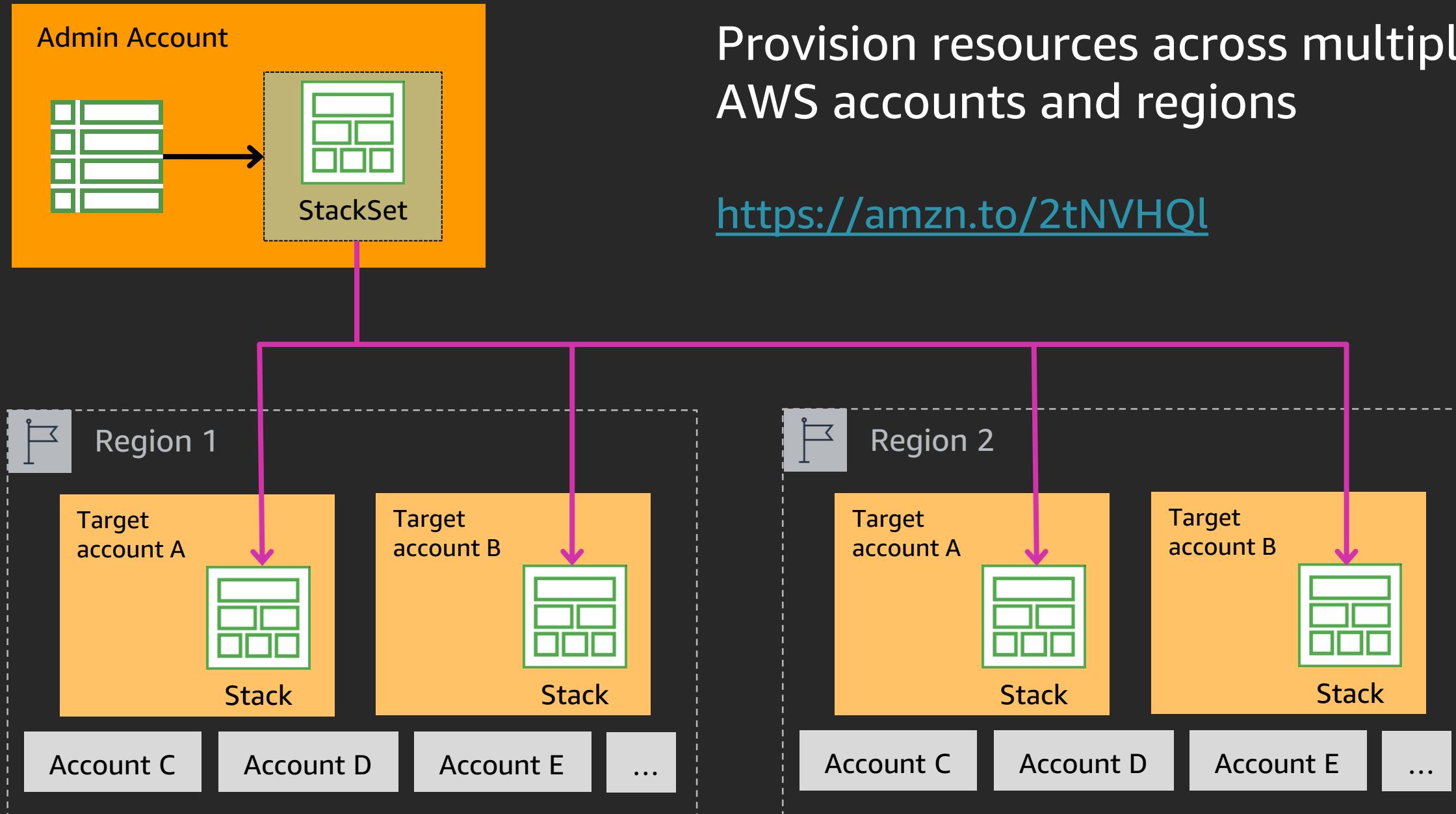
Multi-Region Multi-VPC Connectivity encrypts and **shares** resources across the **global AWS backbone**

Route 53 **health checks** and routes **traffic** based on routing policy

NLB can **load balances** across inter-region peered VPCs and **integrates** with Route 53 **health checks**

How do I manage *a multi-region active-active environment?*

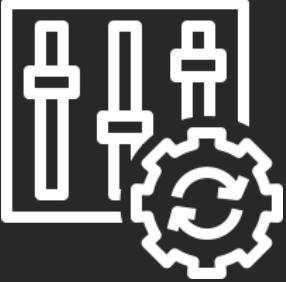
AWS CloudFormation StackSets



Provision resources across multiple AWS accounts and regions

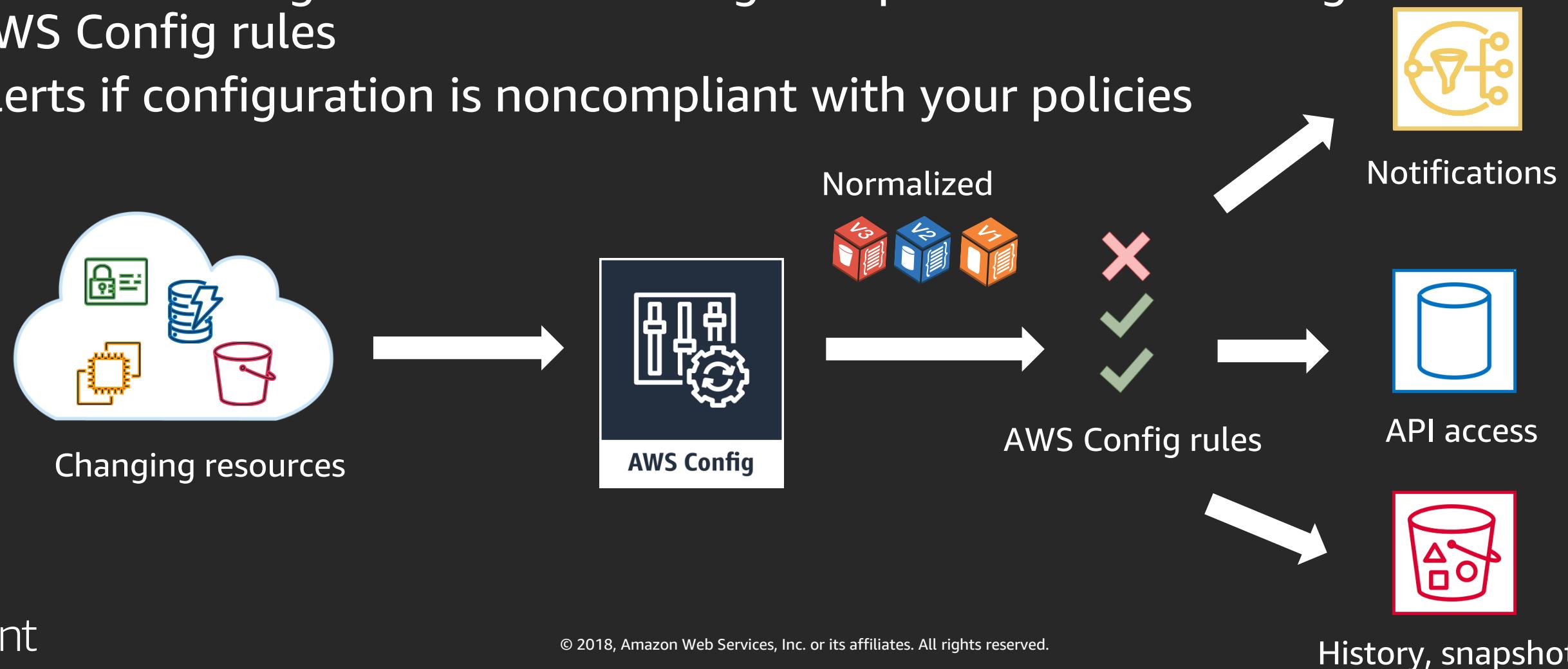
<https://amzn.to/2tNVHQI>

AWS Config rules



Multi-account, multi-region data **aggregation**

- Integrates with AWS Organizations
- Evaluates configuration over time against policies defined using AWS Config rules
- Alerts if configuration is noncompliant with your policies

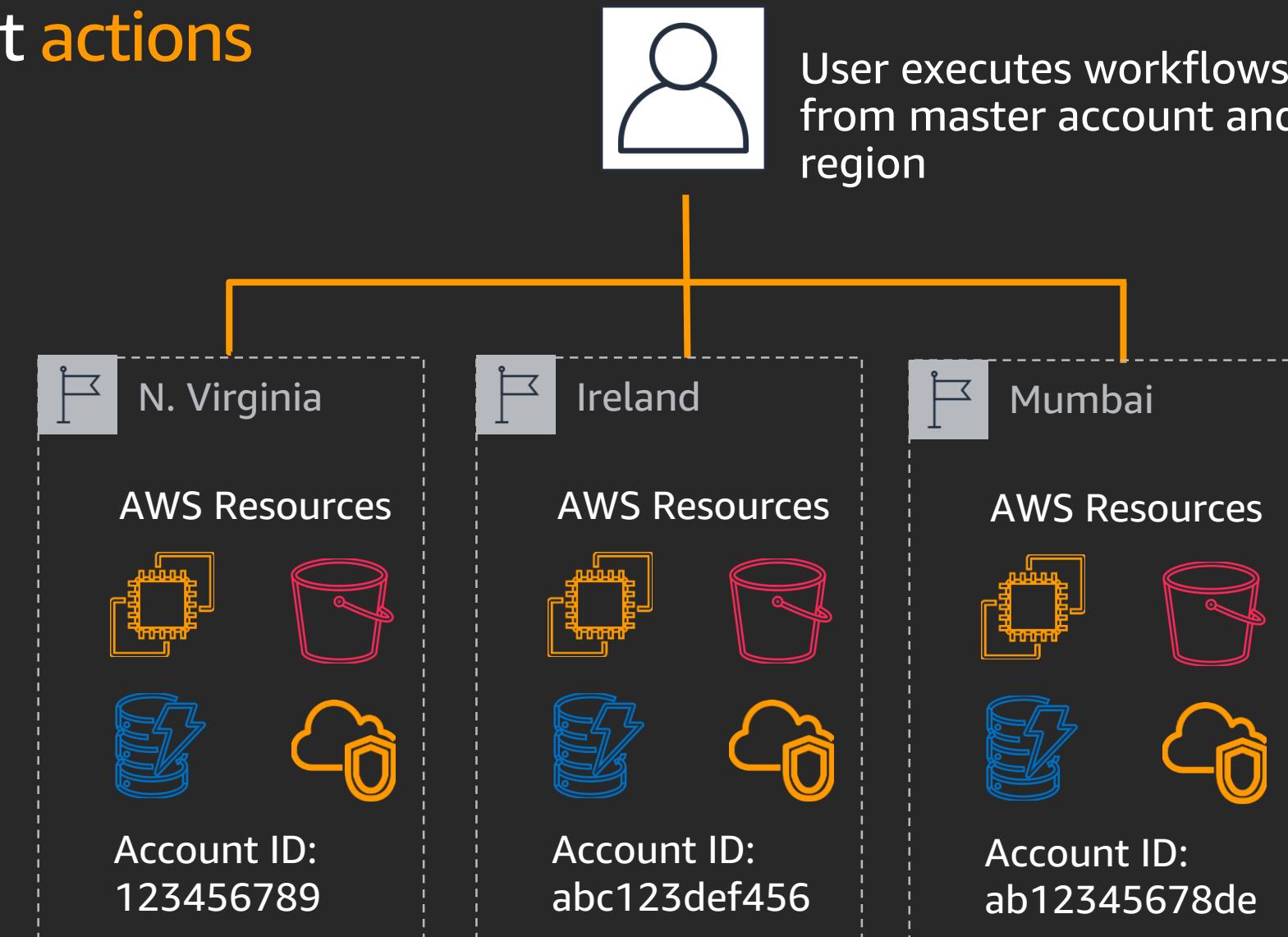


AWS Systems Manager (SM) Automation



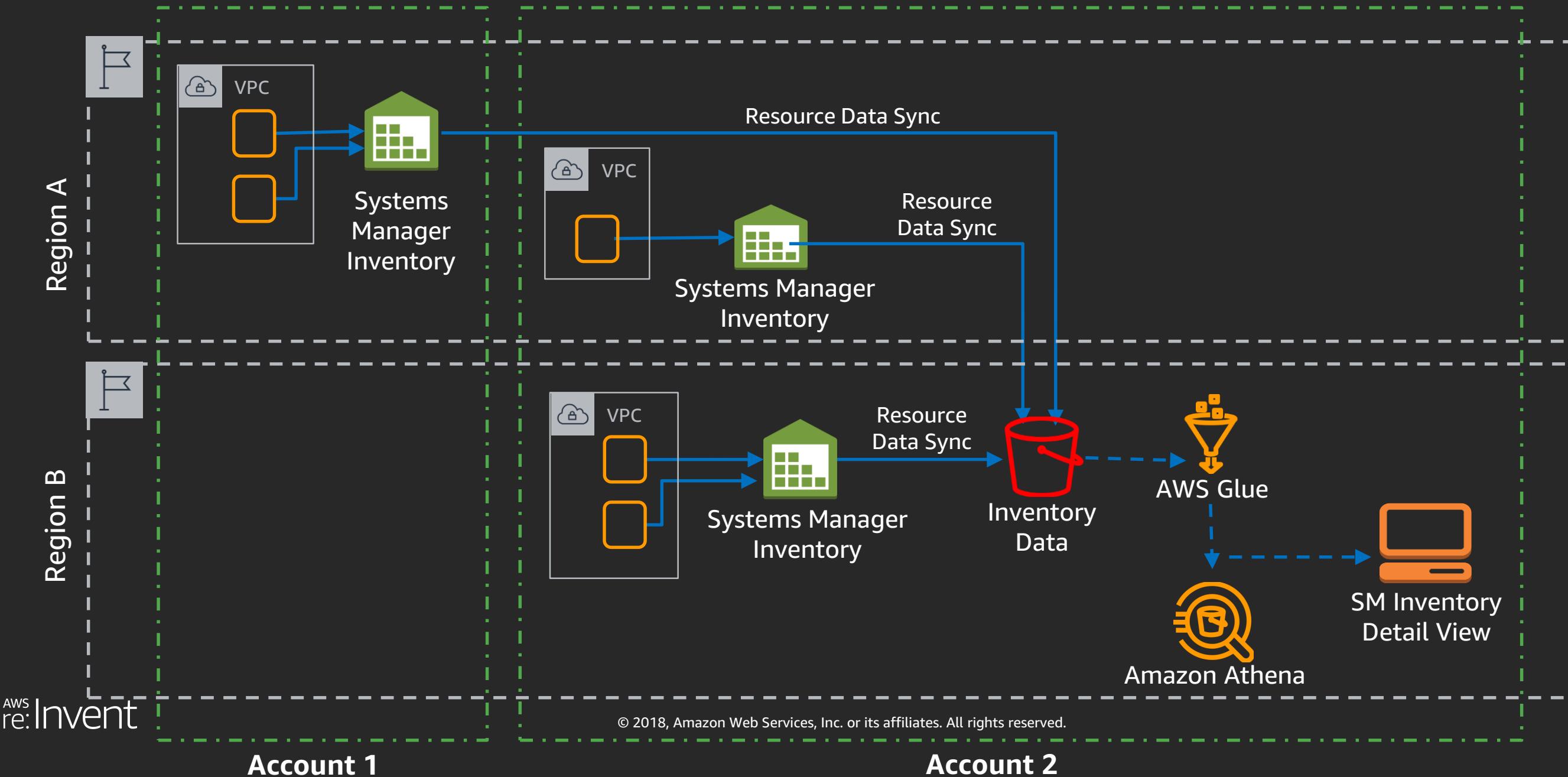
Multi-region and multi-account **actions**

- Take actions on AWS resources centrally
- Remediate compliance drifts
- Backup key resources

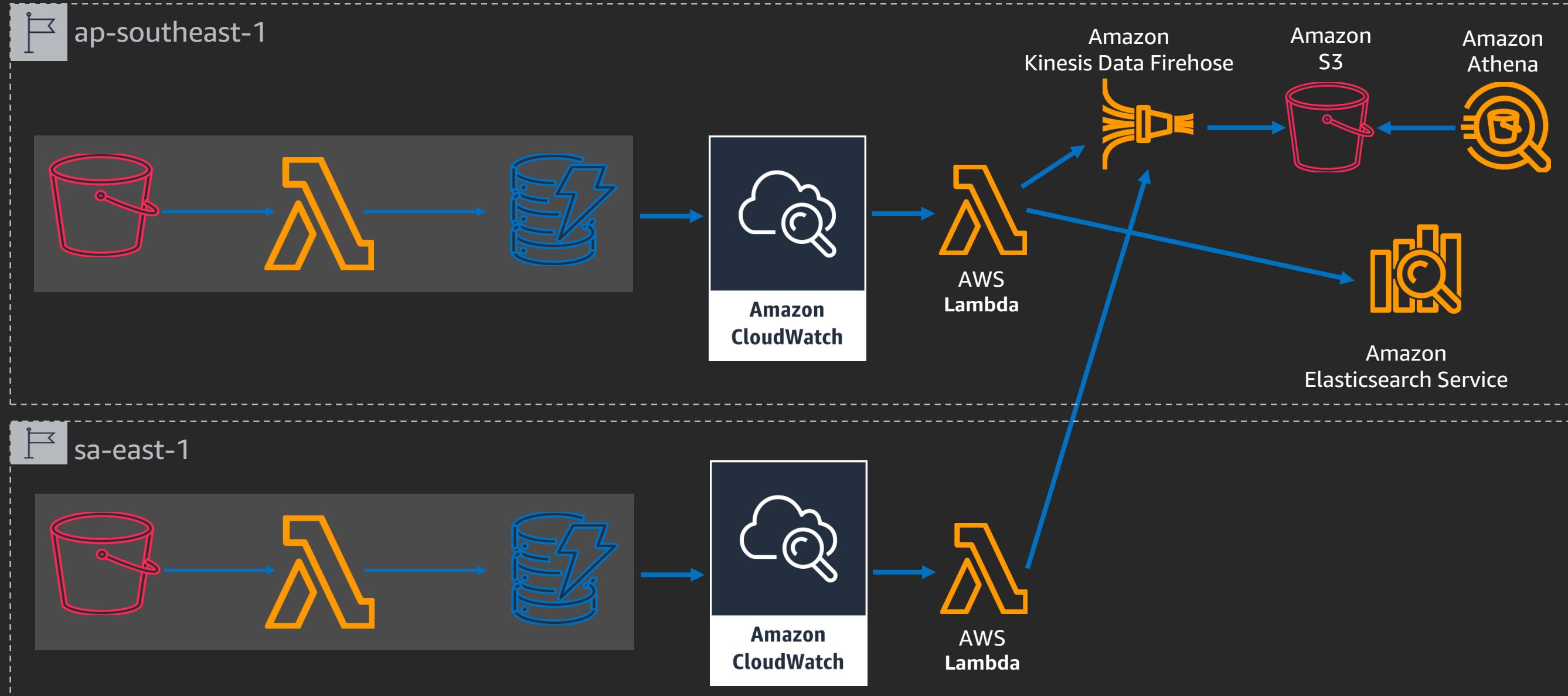


AWS Systems Manager (SM) inventory

Multi-region and multi-account **view**



Centralize Amazon CloudWatch Logs

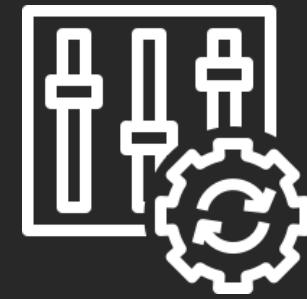


Multi-region management



AWS CloudFormation
StackSets

Provision/Configure



AWS Config
Data Aggregation

Track resource
changes



AWS Systems Manager
Automation & Inventory

Perform operational
management



Amazon CloudWatch

Monitor and then
centralize logging

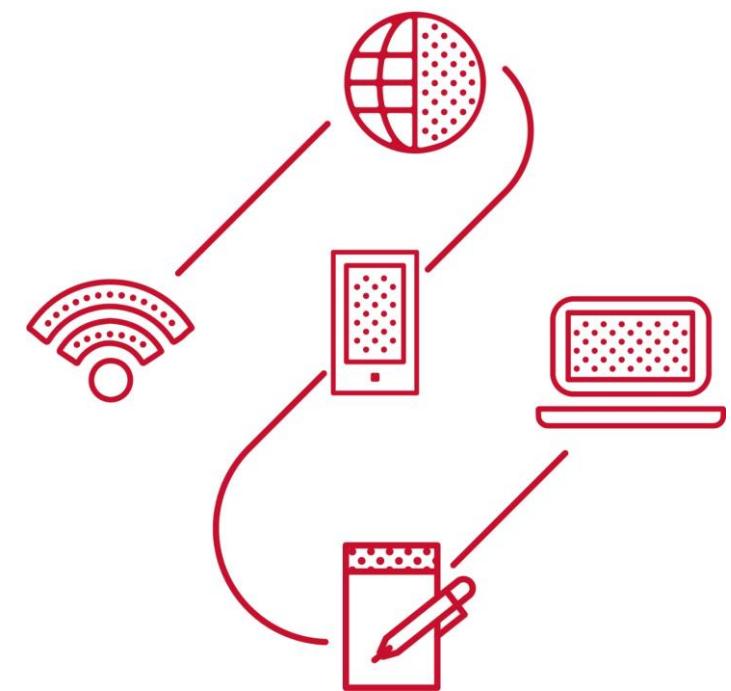
Cross accounts and multi-region

Testing failure modes



ARC209

Christopher Lane
Enterprise Architect



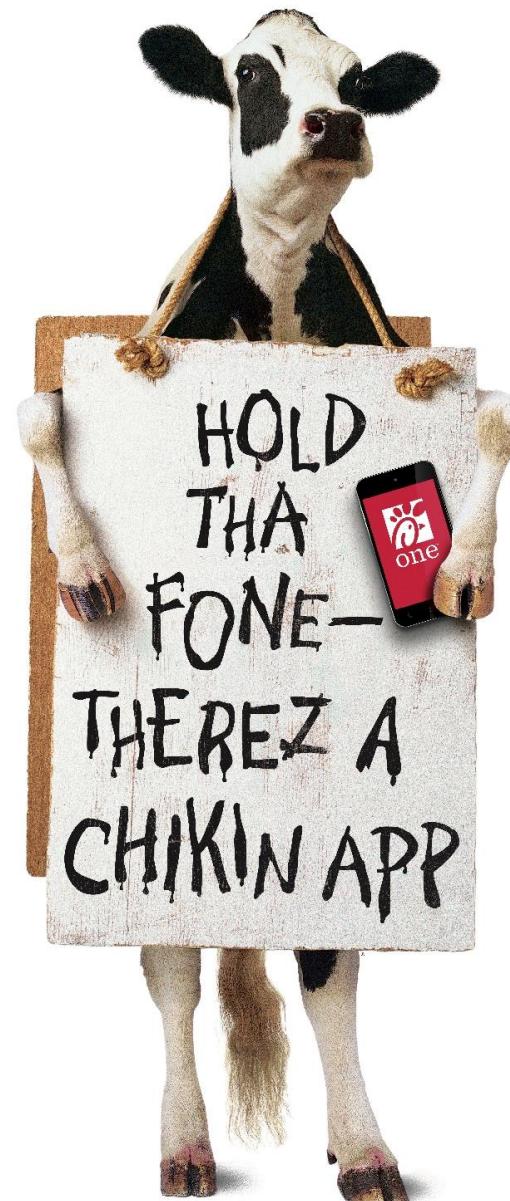
Who is Chick-fil-A?

Chick-fil-A is a privately owned quick service restaurant (QSR) company based in Atlanta, GA

2200+ restaurants in US (and opening in Canada soon!)

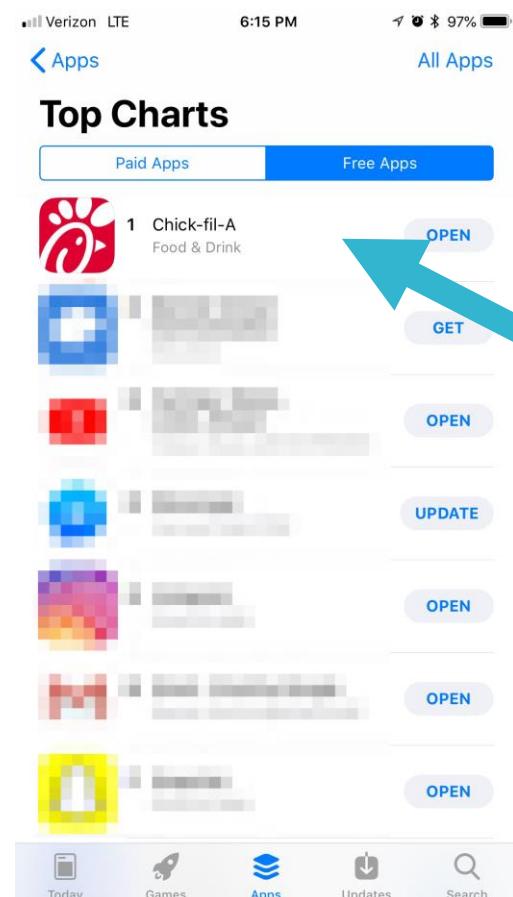
Highest per restaurant sales in QSR industry

More than 10% of revenue flowing through digital channels



Redesigned app and national giveaway

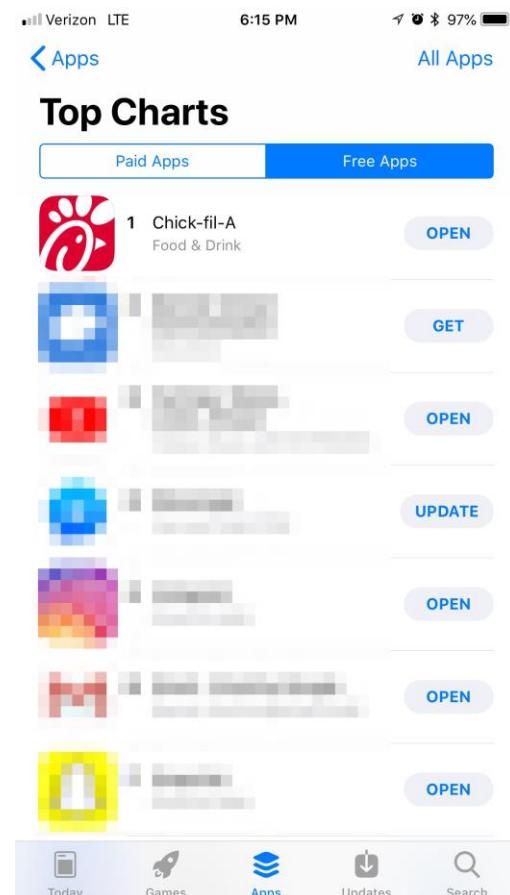
Aug 1



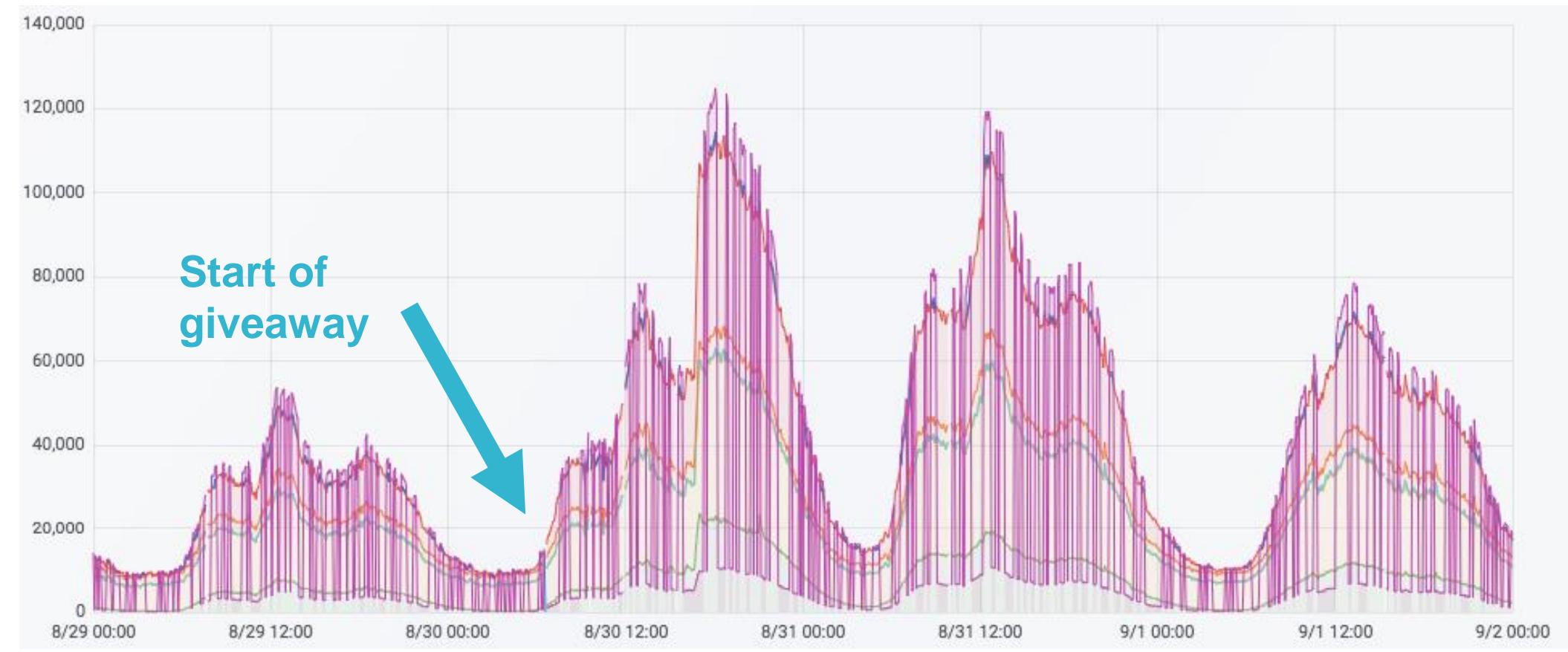
No. 1 in App
Store!

Redesigned app and national giveaway

Aug 1

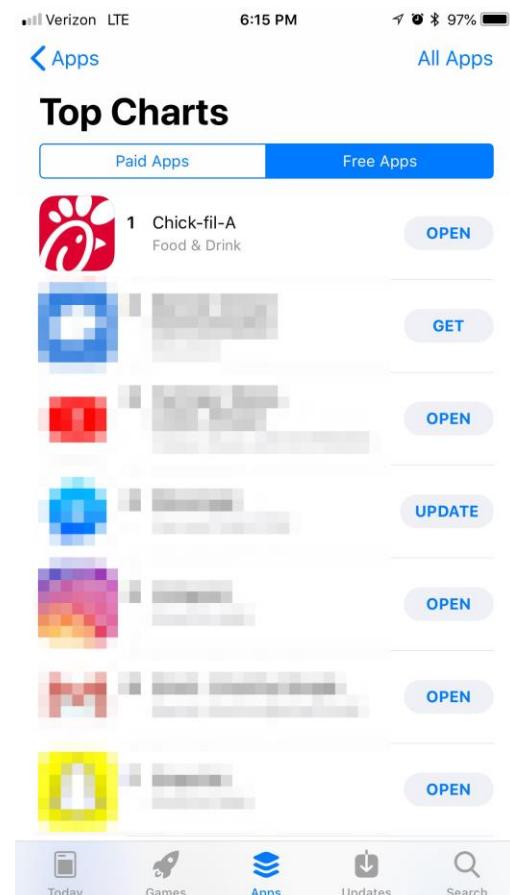


Aug 30– Sep 29

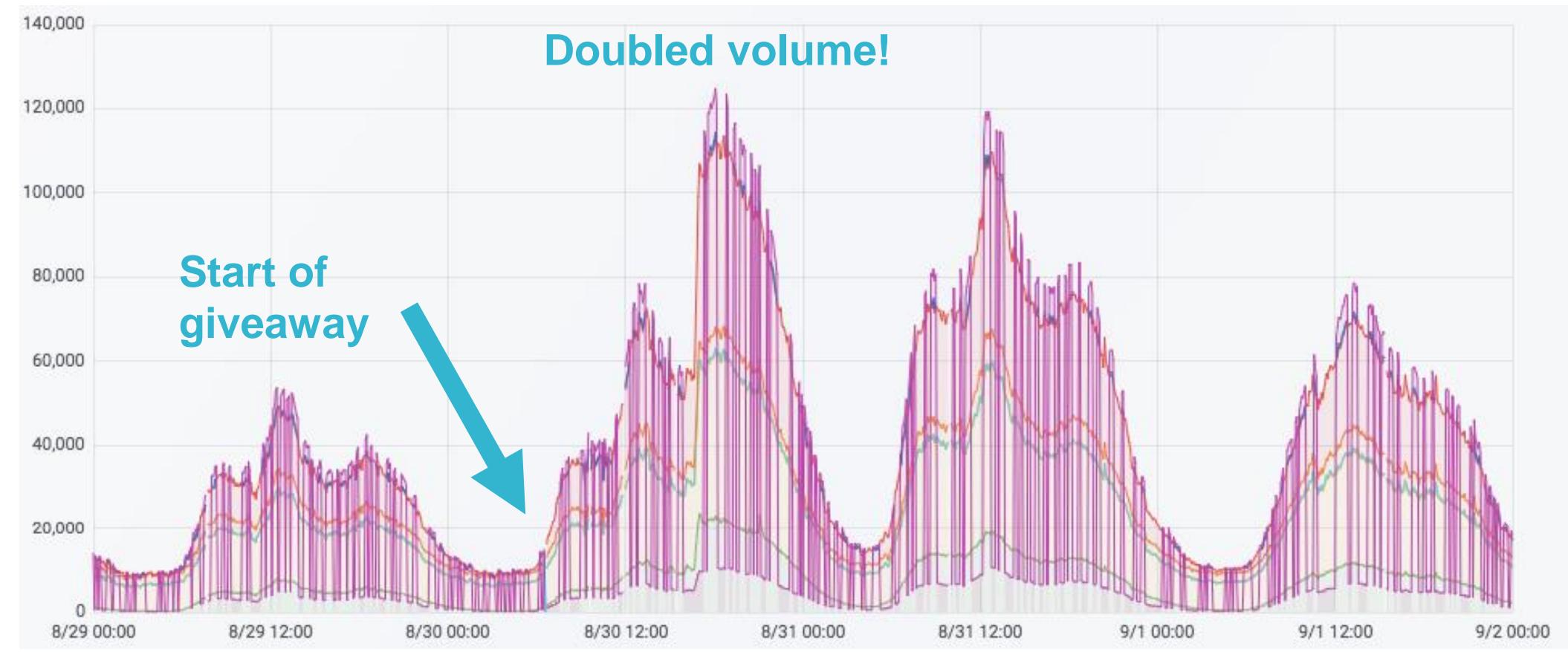


Redesigned app and national giveaway

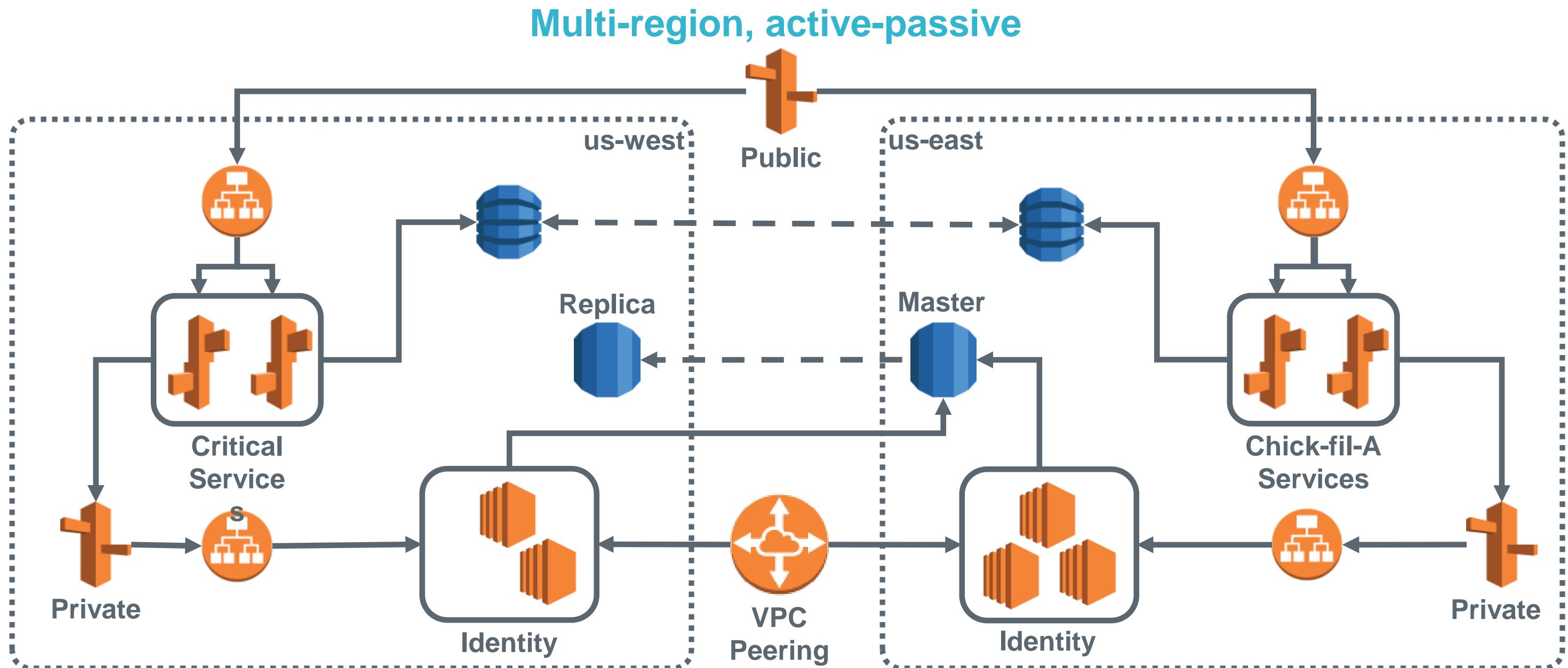
Aug 1



Aug 30– Sep 29

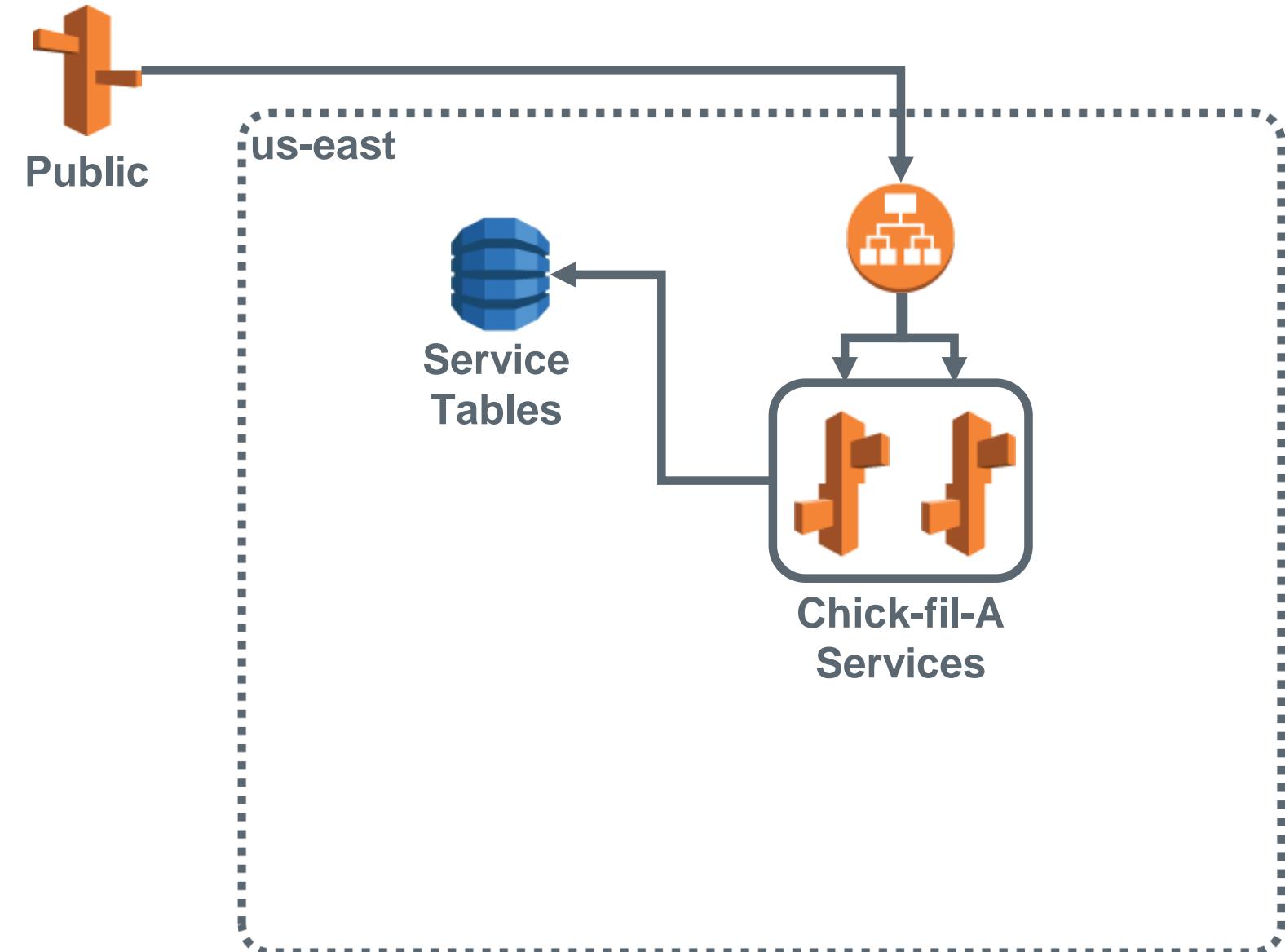


Current architecture



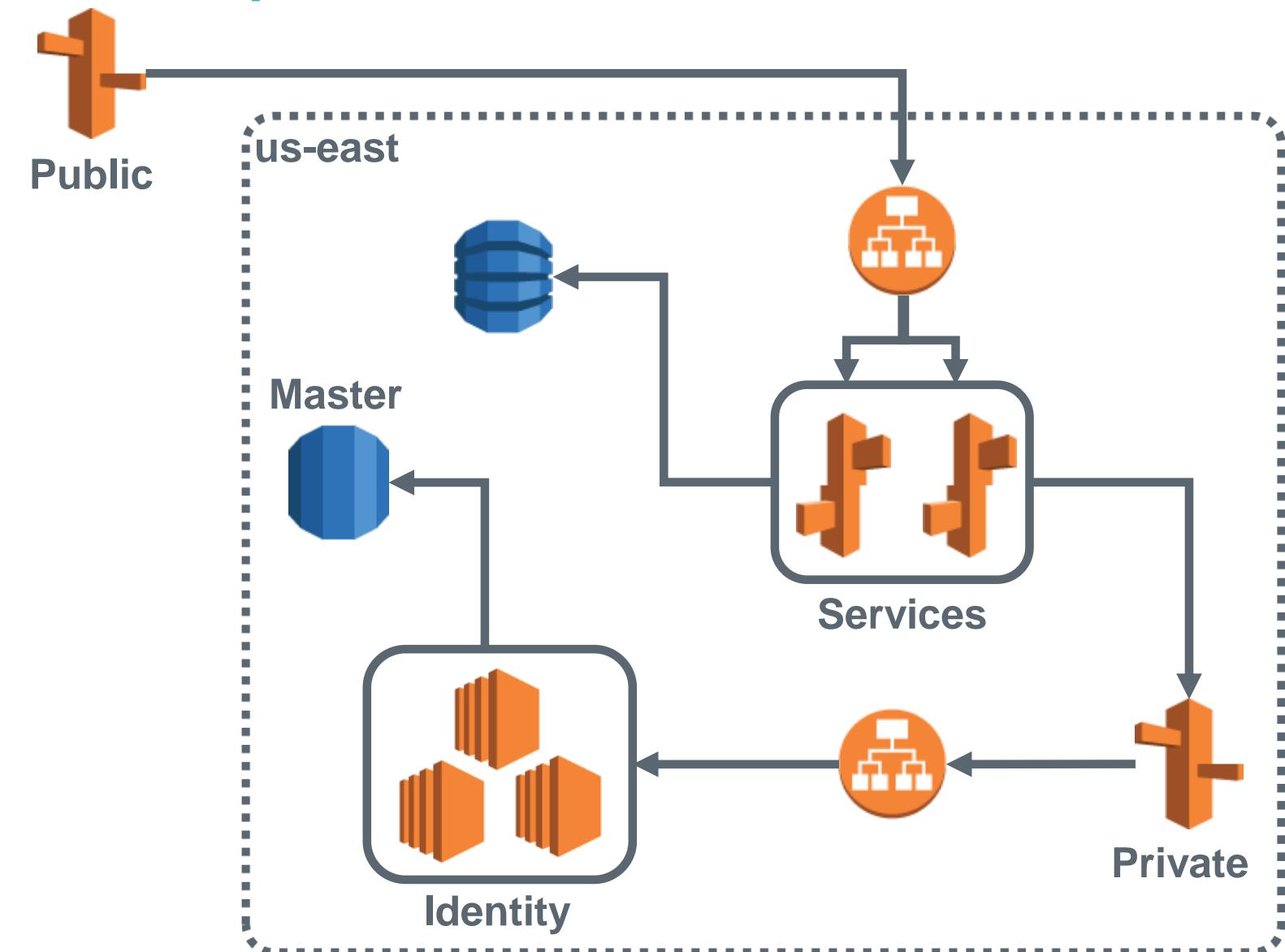
Current architecture

Multi-region, active-passive

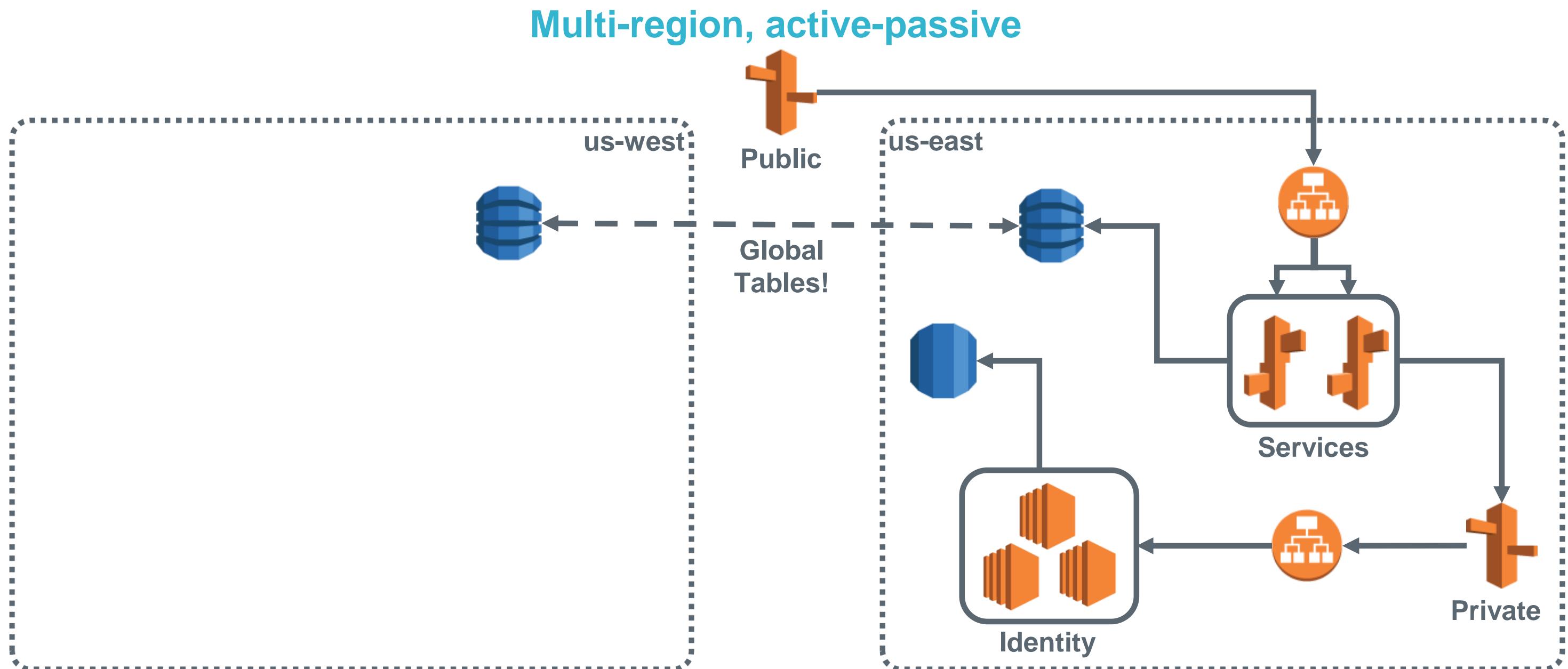


Current architecture

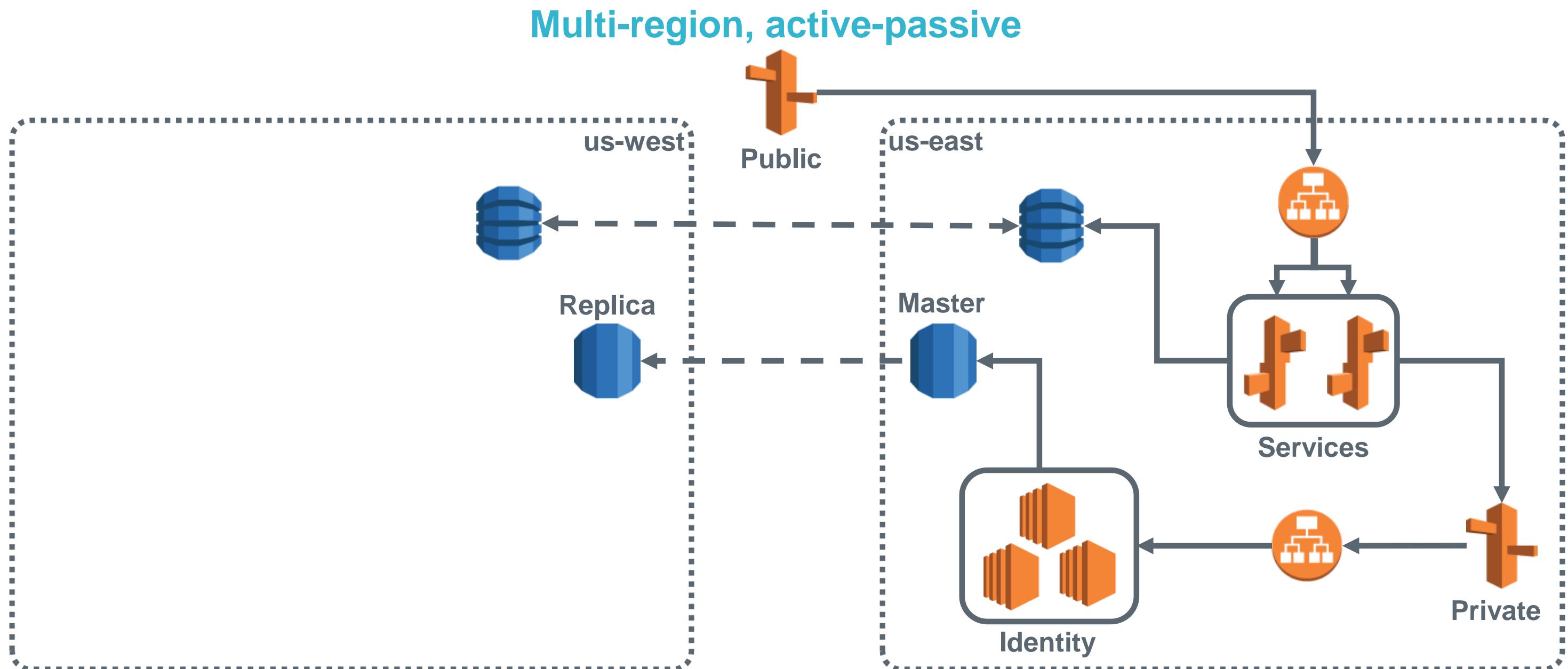
Multi-region, active-passive



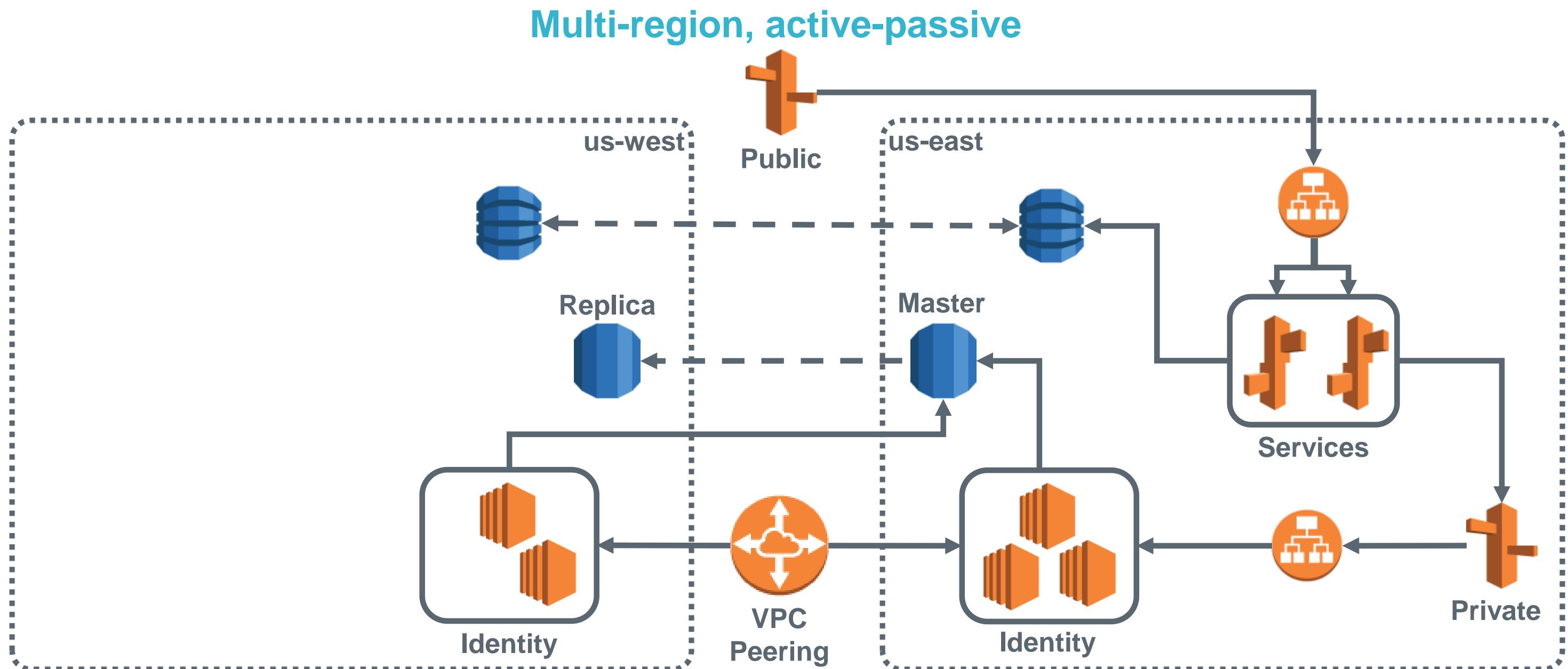
Current architecture



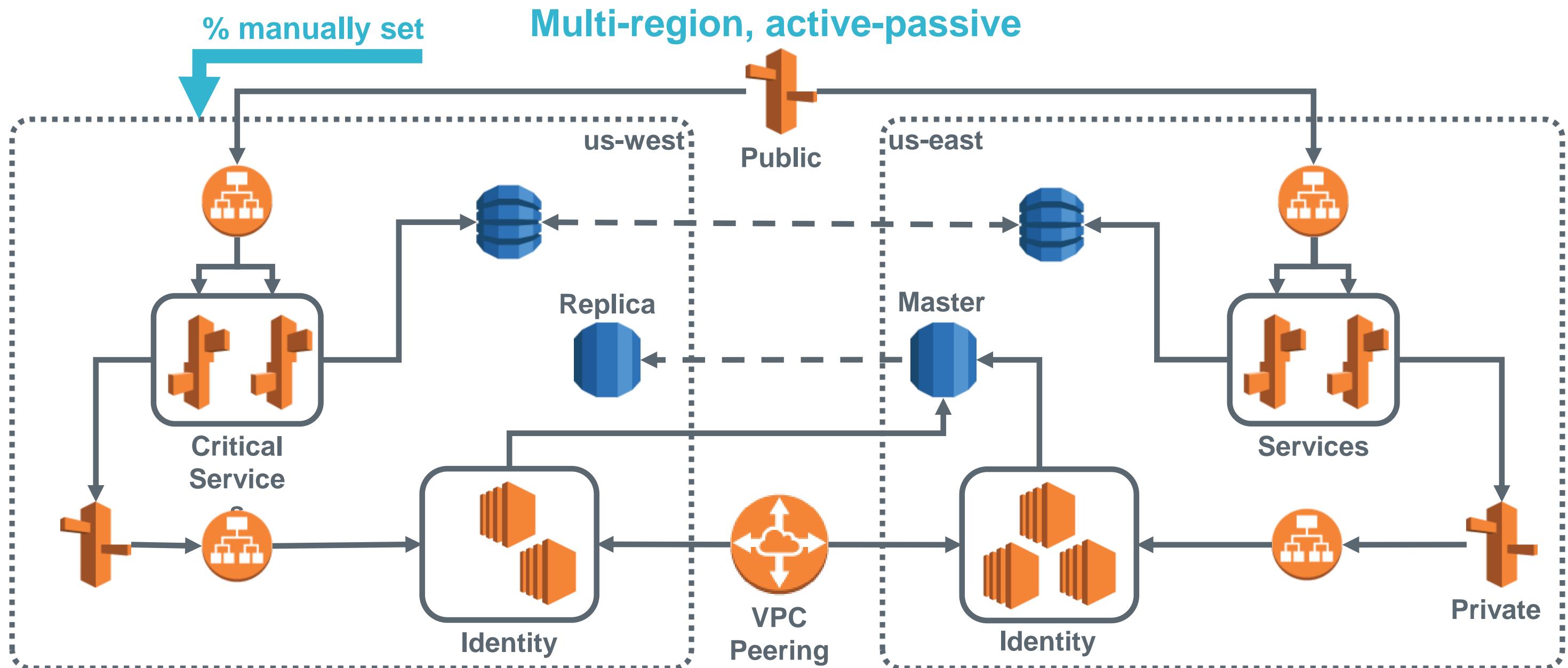
Current architecture



Current architecture

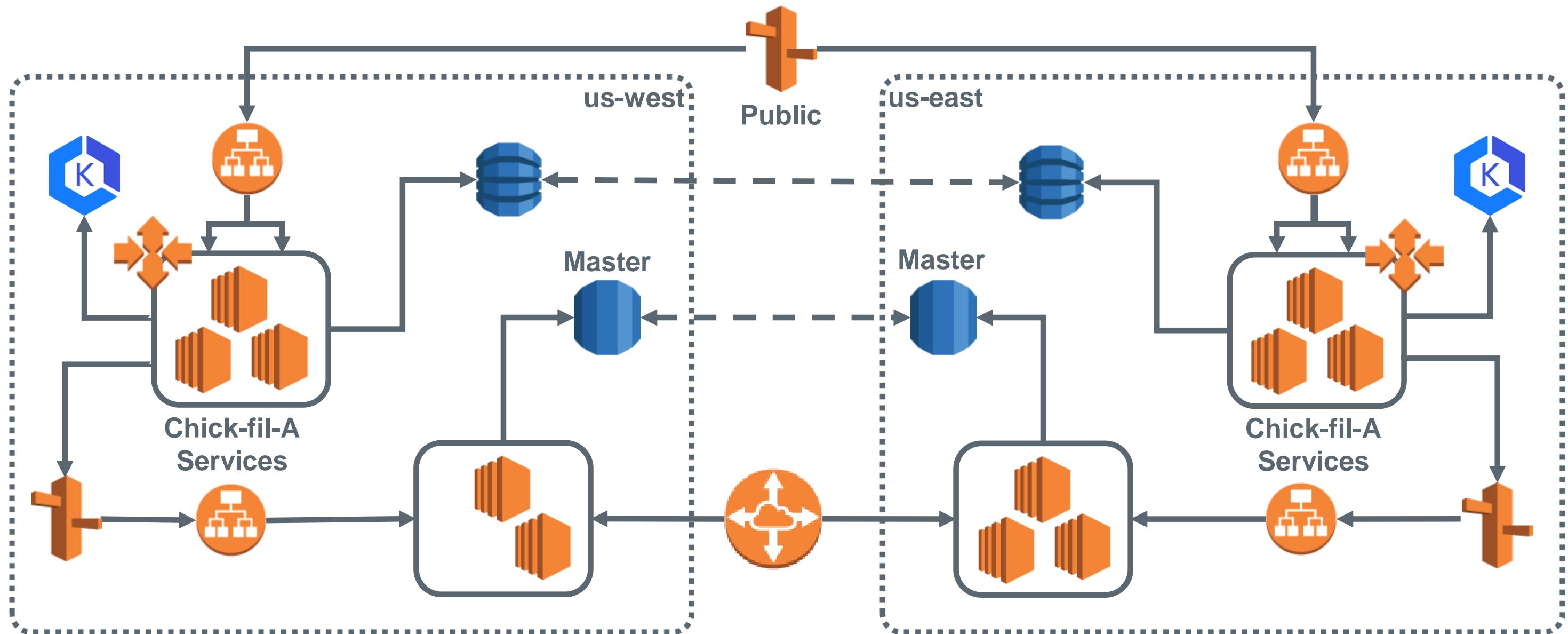


Current architecture



Future architecture

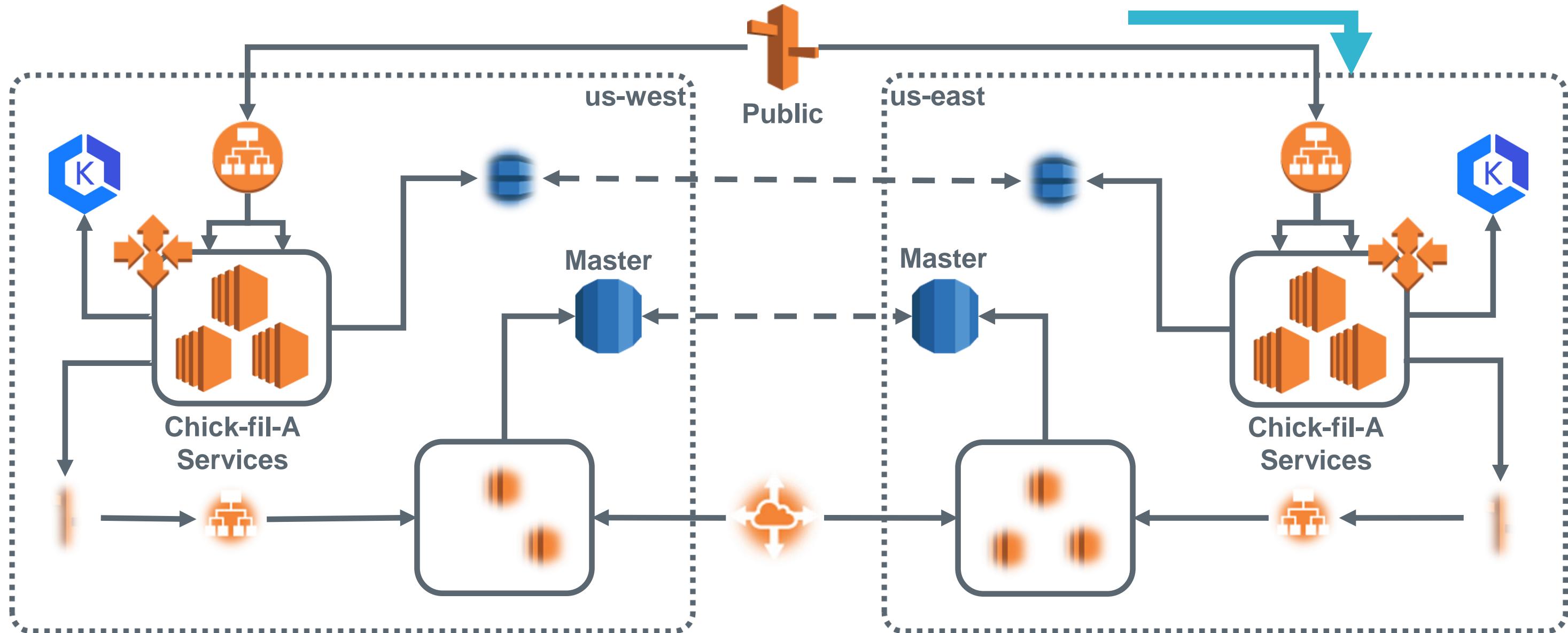
Multi-region, active-active +
Amazon Elastic Container Service for Kubernetes (Amazon EKS)



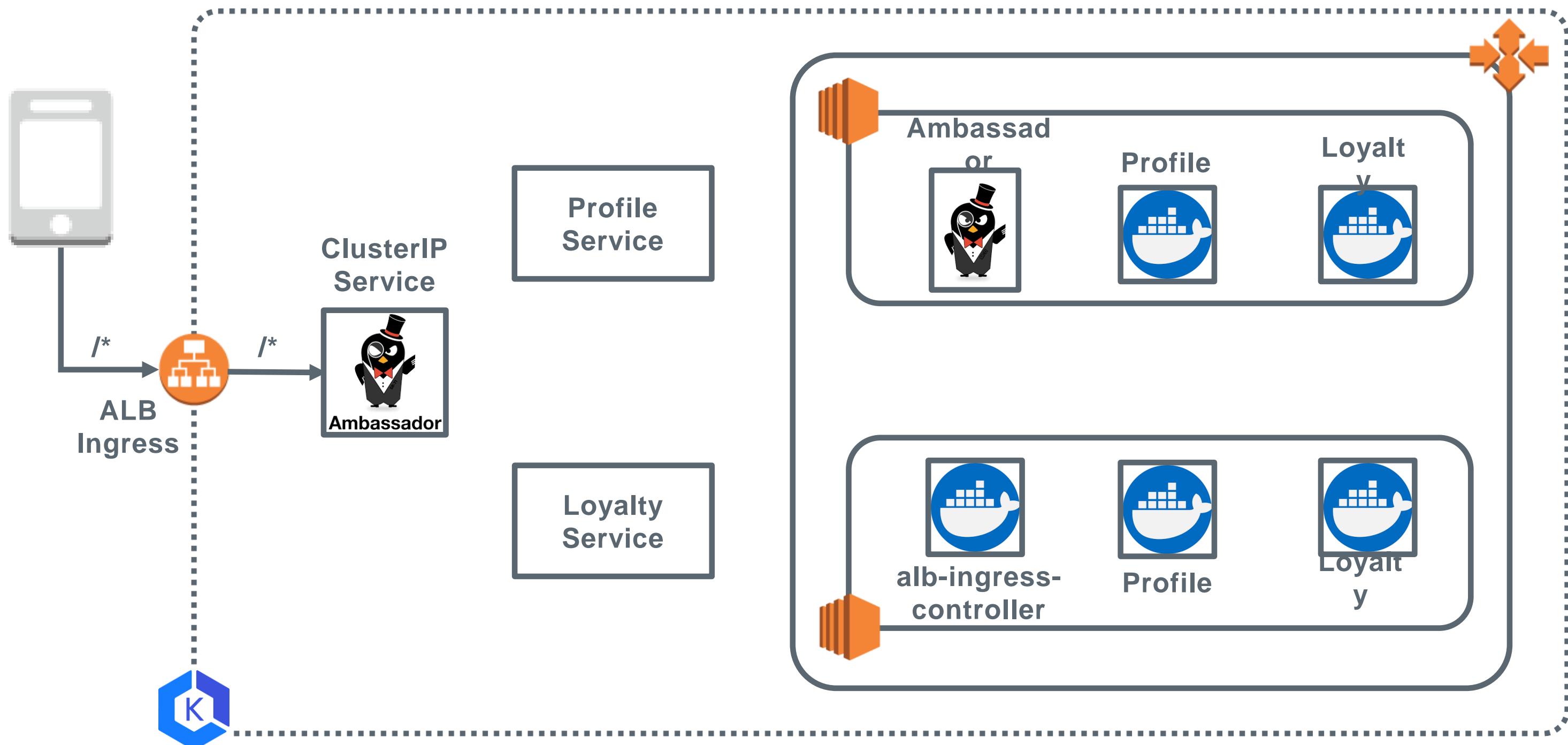
Future architecture

Multi-region, active-active + Amazon EKS

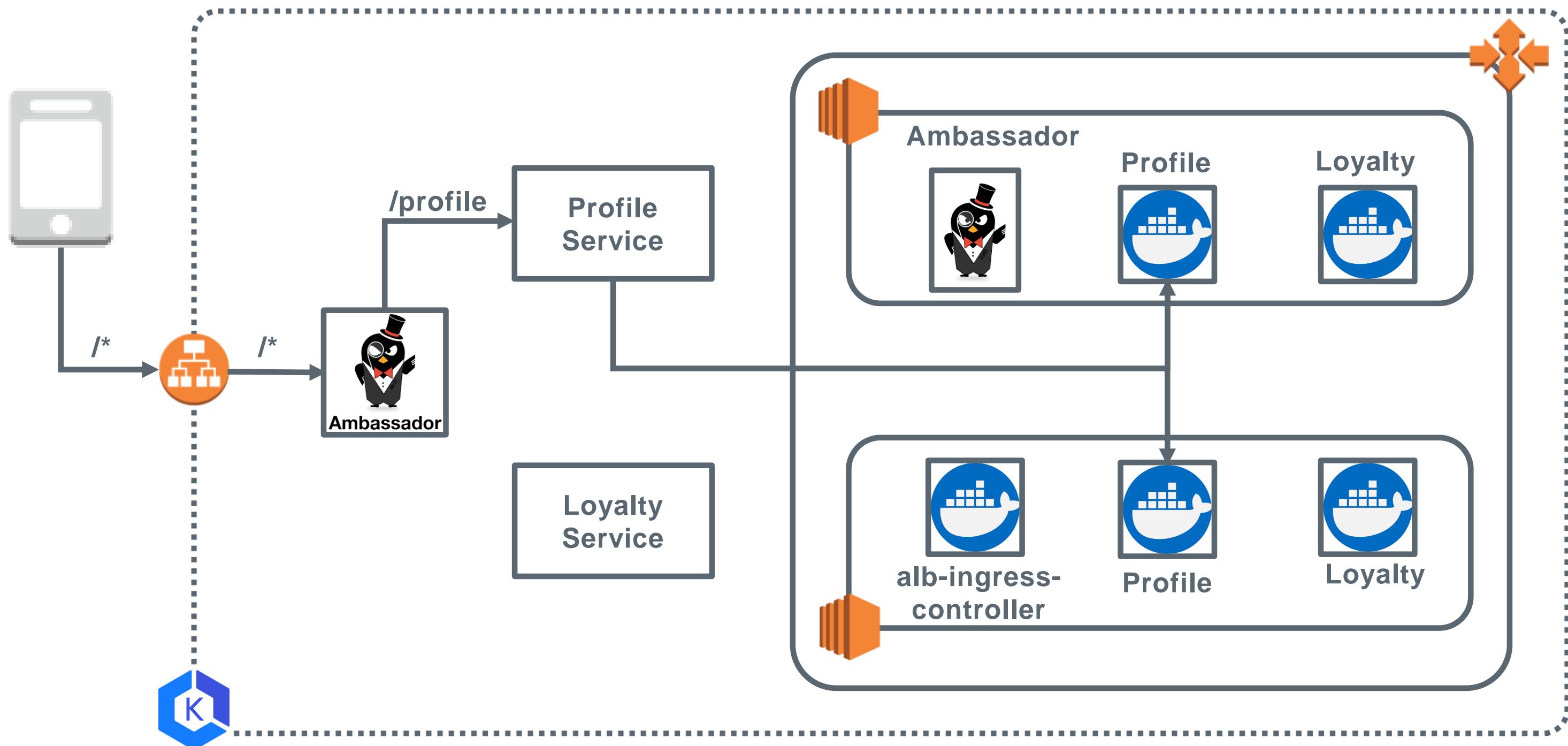
% based on health,
location



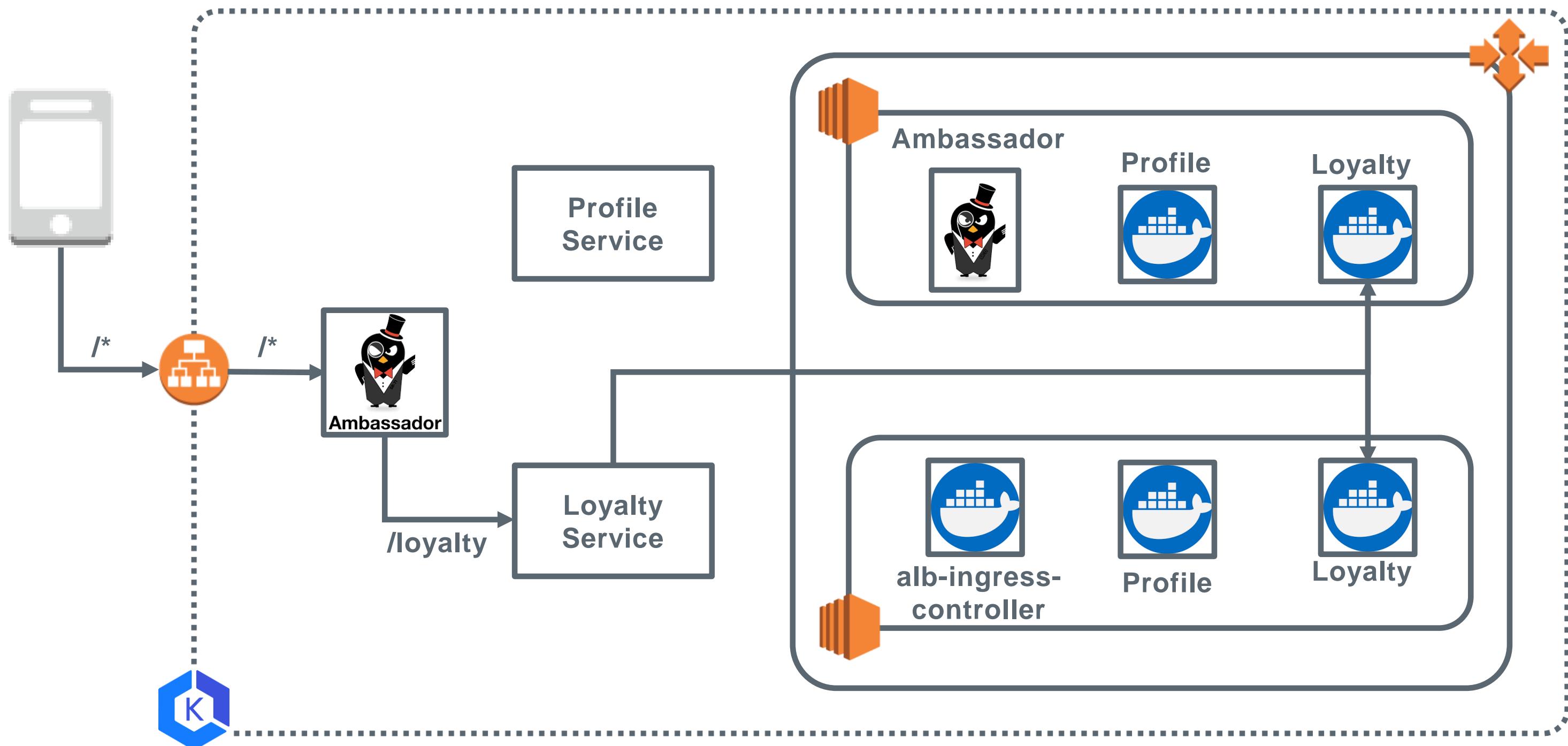
Routing tier



Routing tier



Routing tier



“No goal is too high if you climb with care and confidence.”

...

Truett Cathy



medium.com/@cfatechblog
github.com/chick-fil-a
bit.ly/chickfilacareers

linkedin.com/in/lanecm



Takeaways

- Multi-region active-active architectures add **significant complexity** to your stack – think carefully about your services and whether they need to be multi-region at all.
- Consider whether less complex multi-region architectures, such as **multi-region backup**, **pilot light**, or **warm standby** can meet your needs.
- Design to **avoid race conditions**, by using read local/write global or partitioned write.

Takeaways

- Use storage tools to **keep data synchronized**, including S3 cross region replication and EBS snapshot cross-region copies.
- Keeping data consistent across regions is challenging. Use **DynamoDB Global Tables** or **Read Replicas** with RDS and Aurora.
- **VPC peering** allows consistent security across regions.
- Route 53 can perform failover checks, and **most operations can be automated** – but the service still needs to be monitored. Create relevant metrics and set alarms on them.

Takeaways

- Plan to **manage the environment!** Use AWS CloudFormation StackSets, AWS Config rules, AWS System Manager, and other DevOps tools.
- **If you don't test, it won't work in a crisis.** Test a lot.
- Eat mor chickin'.



Please complete the session
survey in the mobile app.

Related sessions

[ARC315 - Hands-On: Building a Multi-Region Active-Active Solution](#)

[ARC316 - Hands-On: Breaking a Multi-Region Active-Active Solution](#)

[ARC402 - Building Multi-Region Microservices](#)

[ARC415 – Building Multi-Region Persistence with MySQL](#)

[SRV214 - Multi-Region REST APIs with Automatic Failover](#)

[SRV326 - Build a Multi-Region Serverless Application for Resilience & High Availability](#)

[SRV425 - Best Practices for Building Multi-Region, Active-Active Serverless Applications](#)

Speaker meet & greet

Time: 15 minutes after this session

Location: Speaker Lounge (ARIA East, Level 1, Willow Lounge)

Duration: 30 min.

Thank you!

Amy Che
Darin Briskman
Christopher Lane