



AWS Cloud Day

Prague



ISV 301

Mastering resilience at every layer of the cake

Dragos Madarasan
Solutions Architect Team Lead

AWS

Adrian Bere
Senior Solutions Architect

AWS



What to expect



Explore advanced
resilience features &
patterns



“It doesn’t
depend”



Demos



Case studies

Disambiguation

Our scope for today

Resilience
(the subject)

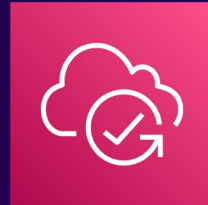


Disaster Recovery (DR)
Resilience (Chaos) Testing

Include



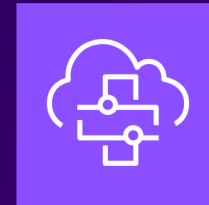
AWS Cloud Resilience



AWS
Resilience
Hub



AWS Fault
Injection
Service



Amazon
Application
Recovery
Controller



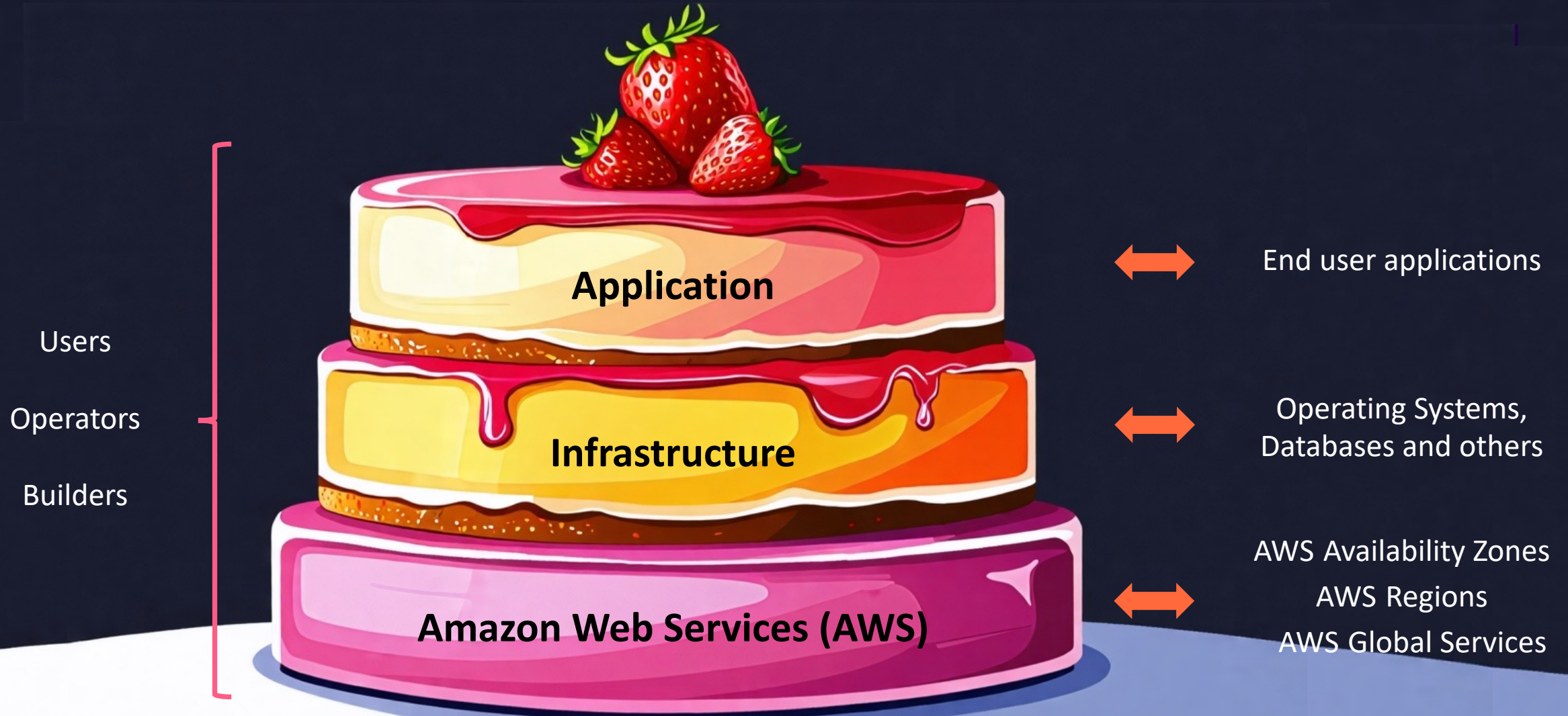
AWS
Backup



AWS Elastic
Disaster
Recovery

Introduction: What is cloud resilience?

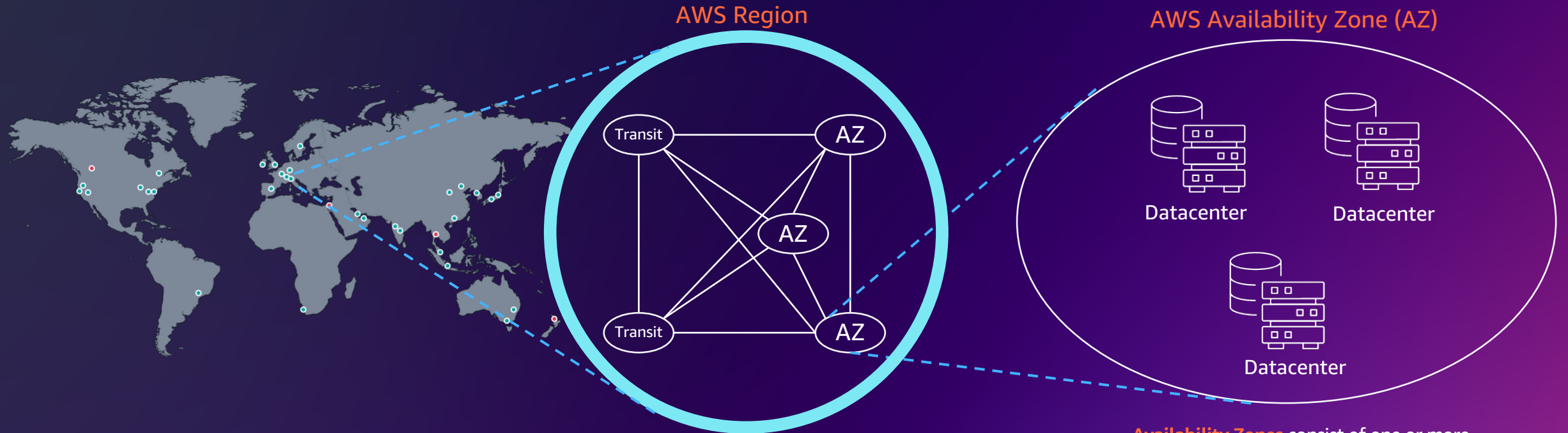
Our scope



Our infrastructure is designed to support operational resiliency

The AWS Cloud spans 120 Availability Zones within 38 Geographic Regions, with announced plans for 10 more Availability Zones and 3 more AWS Regions in the Kingdom of Saudi Arabia, Chile, and the AWS European Sovereign Cloud.

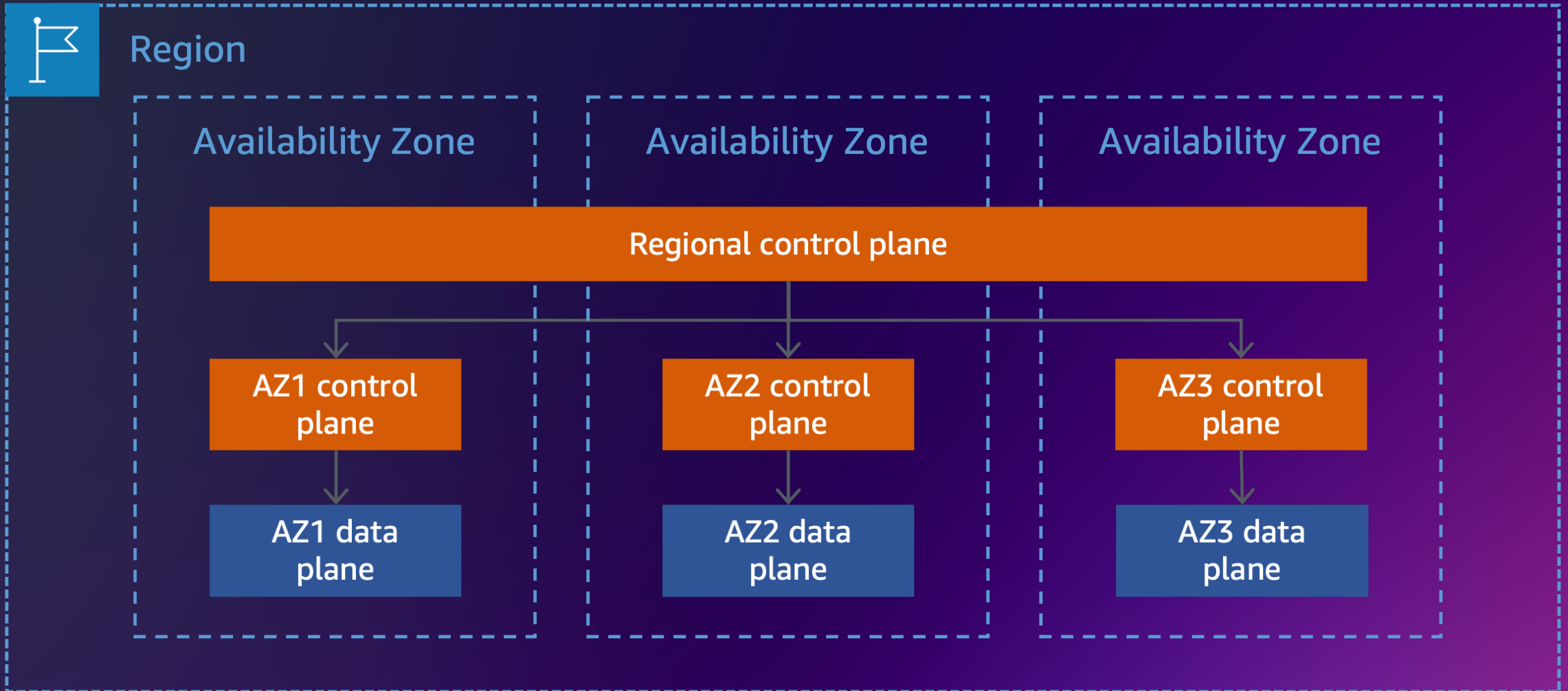
AWS Regions are comprised of multiple AZs for high availability, high scalability, and high fault tolerance. Applications and data can be replicated in real time and consistent in the different AZs.



A Region is a physical location in the world where we have multiple **Availability Zones**.

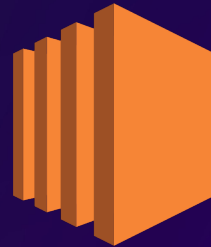
Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities.

Zonal services



Understand planes of access

Amazon Elastic
Compute Cloud
(Amazon EC2)



Data plane – VPC connection



Control plane – AWS API

Regional services



Amazon Simple Storage Service (Amazon S3)



Amazon DynamoDB



Amazon Simple Queue Service (Amazon SQS)



AWS Lambda



Amazon API Gateway

Understand planes of access

Amazon DynamoDB

Data plane – AWS API

Control plane – AWS API



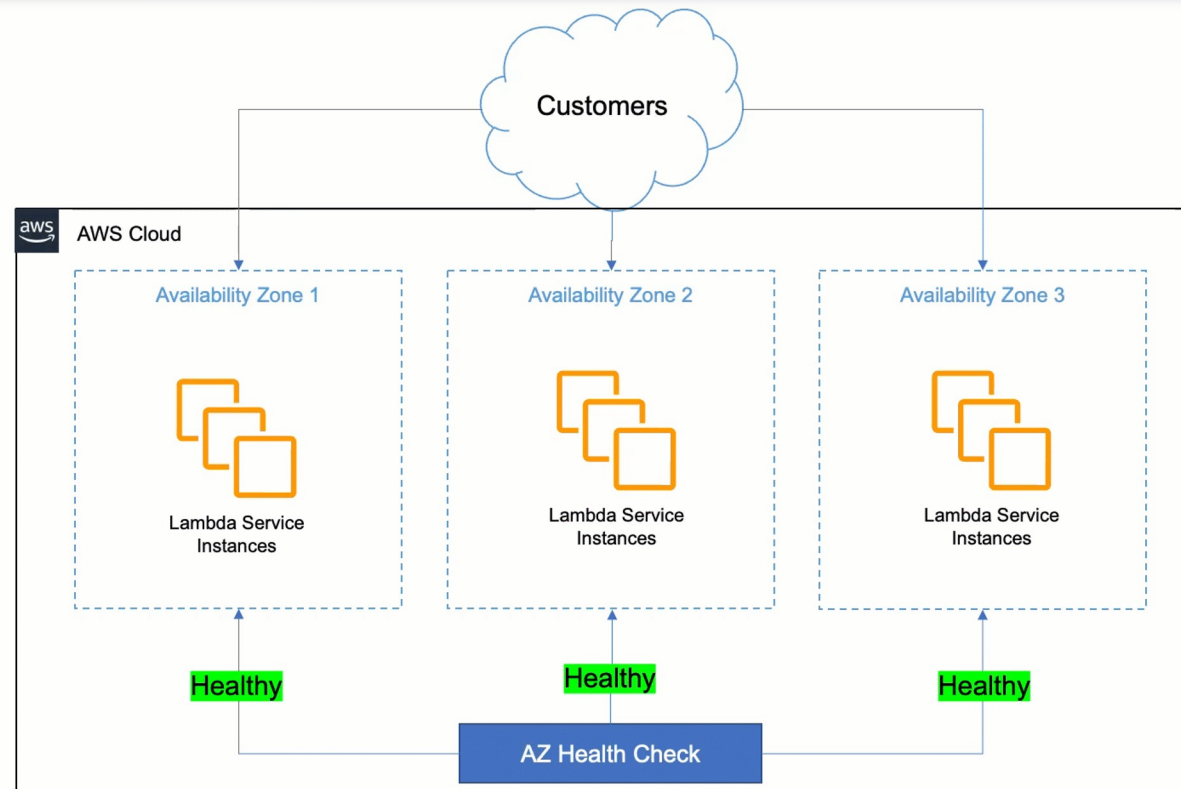
Global services

Recommendation

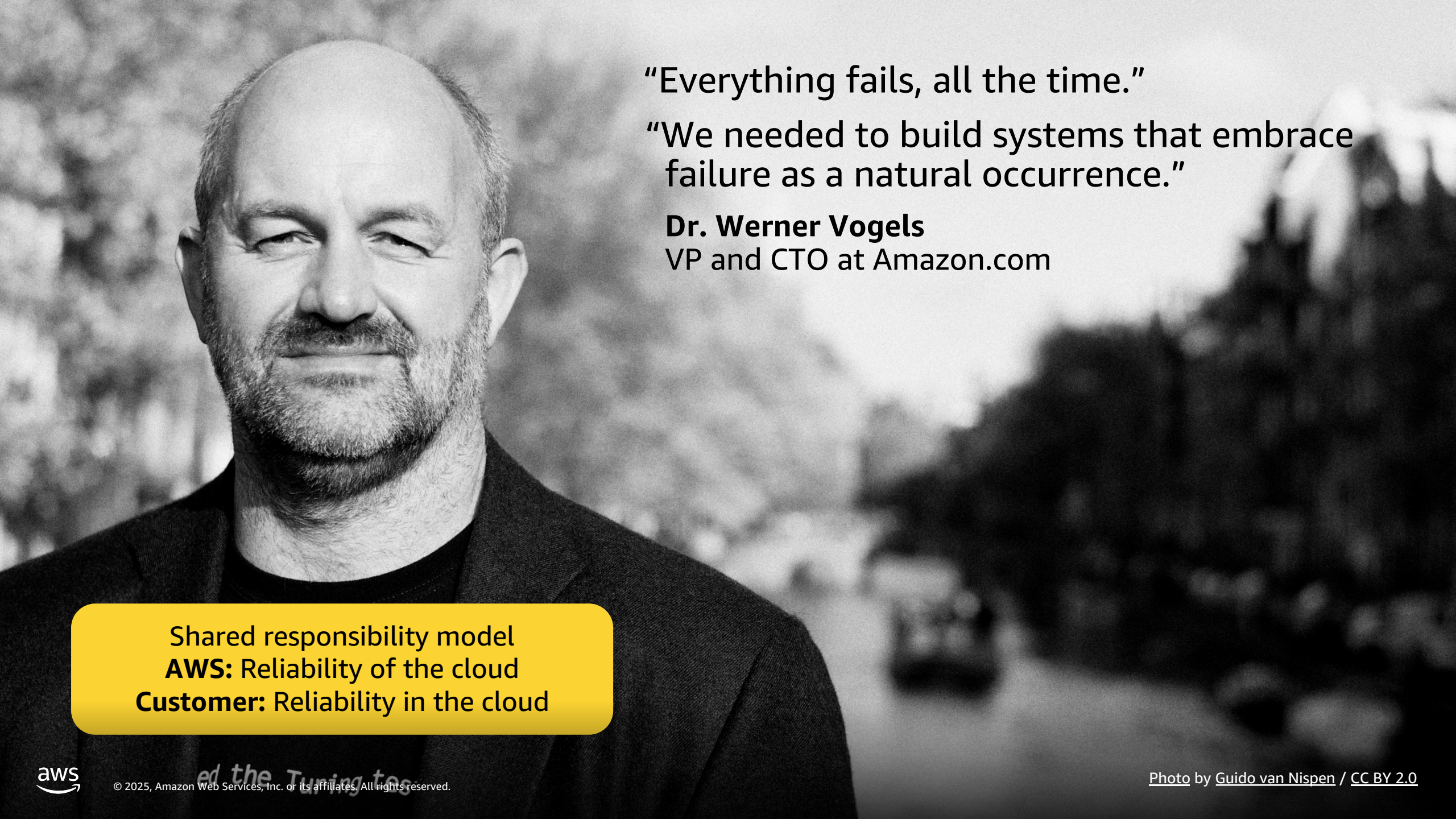
Do not rely on the control planes of partitioned services in your recovery path.

Instead, rely on the data plane operations of these services.

Case study: AWS Lambda: Resilience under-the-hood



AWS Lambda



“Everything fails, all the time.”

“We needed to build systems that embrace failure as a natural occurrence.”

Dr. Werner Vogels

VP and CTO at Amazon.com

Shared responsibility model
AWS: Reliability of the cloud
Customer: Reliability in the cloud



Introduction to continuous resilience

Foundational resilience



Continuous 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 Application Recovery Controller (ARC)

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

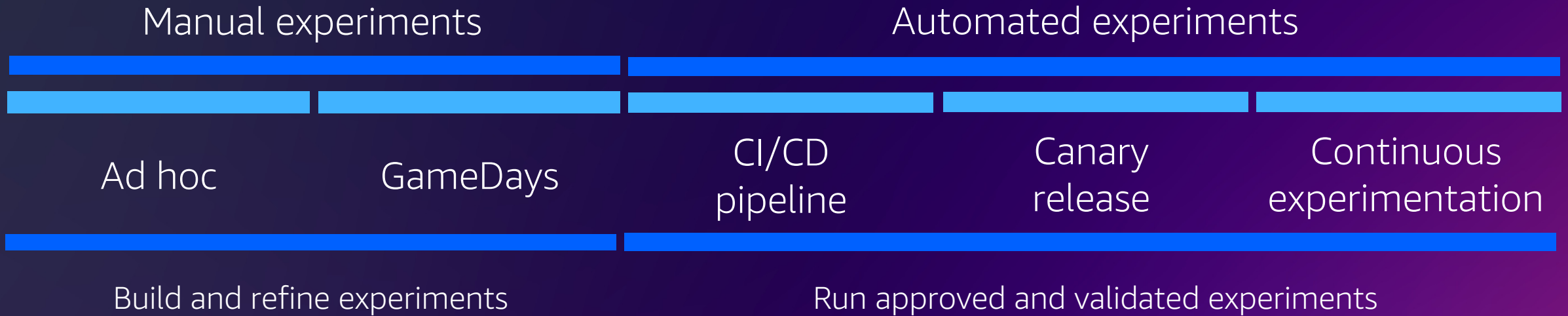
AWS Solutions Library

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



What are the current trends in resilience?

1: Growing demand for resilience testing



2: Multi Region

Financial Services

Enhanced business continuity, data redundancy, and compliance with local data protection regulations

Healthcare and Life Sciences

Enhanced data redundancy, disaster recovery, and compliance with regional data privacy regulations (e.g., HIPAA in the US, GDPR in the EU).

Media, Entertainment, and Gaming

Provide a reliable and seamless experience for their customers.

Automotive and Aviation

Scale globally by using geo-location, comply with local data storage, sovereignty, and privacy requirements.

3: Customers building resilience culture



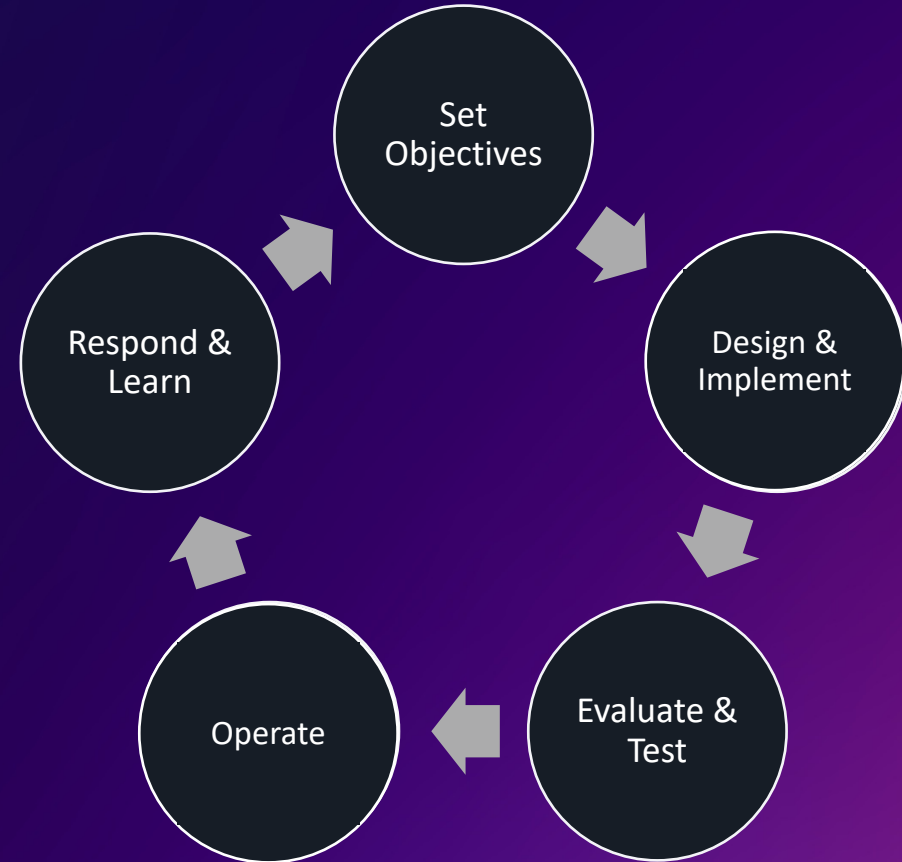
The AWS Resilience Lifecycle Framework

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



AWS continuous resilience services



Practices at each step



Setting objectives



Workload prioritization
and tiering

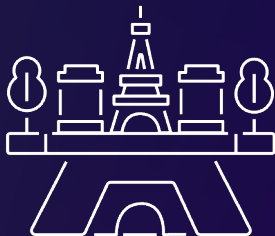


Objective metrics
identification

Design and implement



Architecture recommendation



Foundational resilience services and features



CI/CD



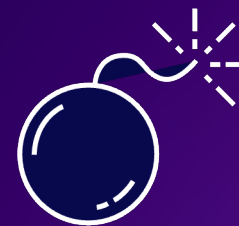
Logging



Dependency mapping



Code reviews and frameworks

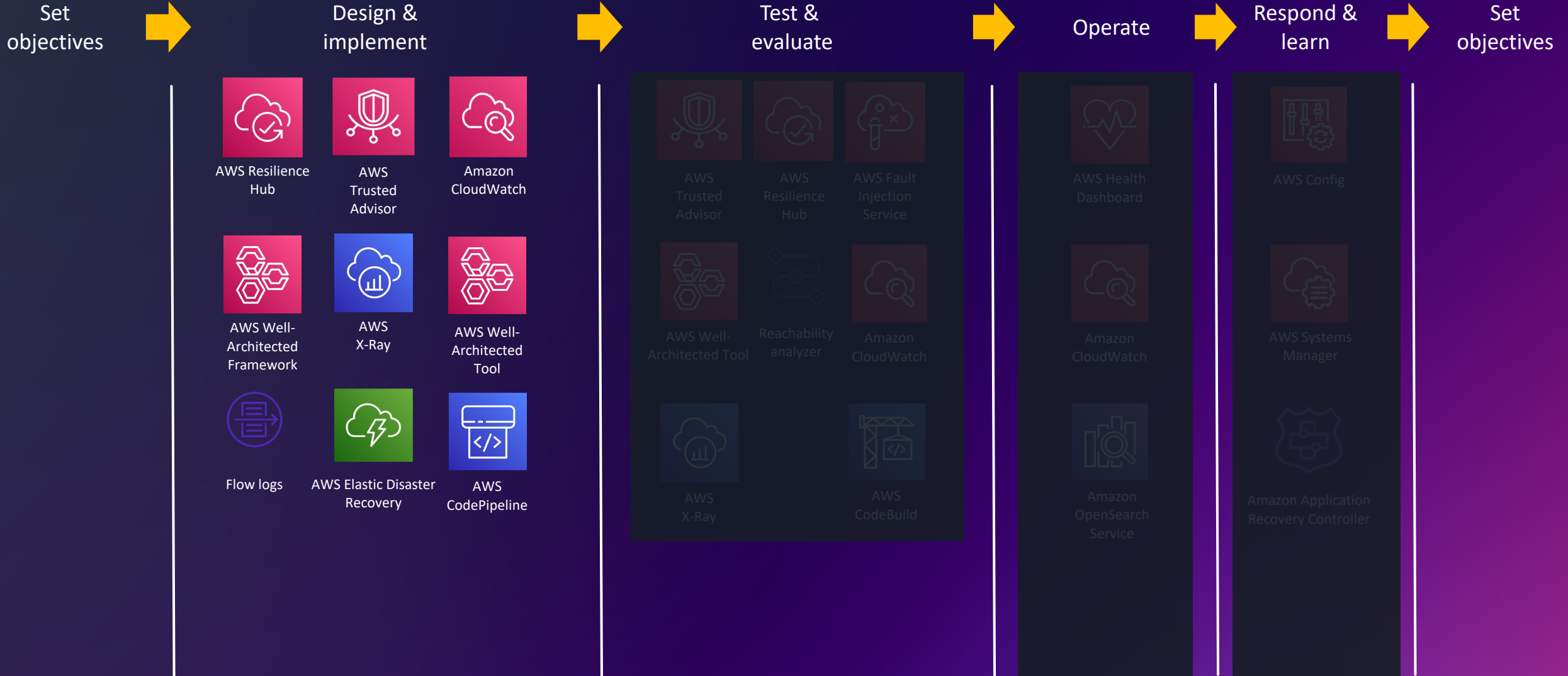


Backup and disaster recovery



Training

AWS continuous resilience services



Reliability

The ability of a workload to perform its required function correctly and consistently over an expected period of time.

– Reliability Pillar, AWS Well-Architected Framework



<https://bit.ly/reliability-pillar>

Reliability design principles

- ⌘ Automatically recover from failure

- ⌘ Test recovery procedures

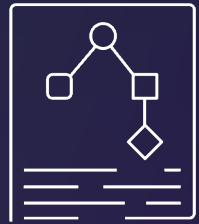
- ⌘ Scale horizontally to increase aggregate workload availability

- ⌘ Stop guessing capacity

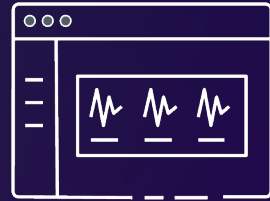
- ⌘ Manage change through automation



Evaluate and test



Tracing



Performance
benchmarking



Fault injection and
GameDays



Load testing

AWS continuous resilience services

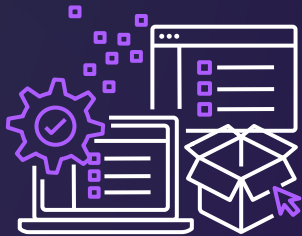


AWS Fault Injection Service

A fully managed service for running fault injection experiments

RESILIENCE LIFECYCLE: TEST & EVALUATE

Easy to
get started



Real-world
conditions



Safeguards



Scenario library

- List of scenarios
- Detailed description
- Create experiment templates
- Bulk edit parameters

Scenario library (1/12) [Info](#)

Select a scenario to see its details.

AZ Availability: Power Interruption

Description
Affect multiple resource types in a single AZ, targeting by tags and explicit ARNs, to approximate power interruption in one AZ.

Cross-Region: Connectivity

Description
Block application network traffic from experiment Region to target Region and pause cross-Region replication

EC2 Stress: Network Latency

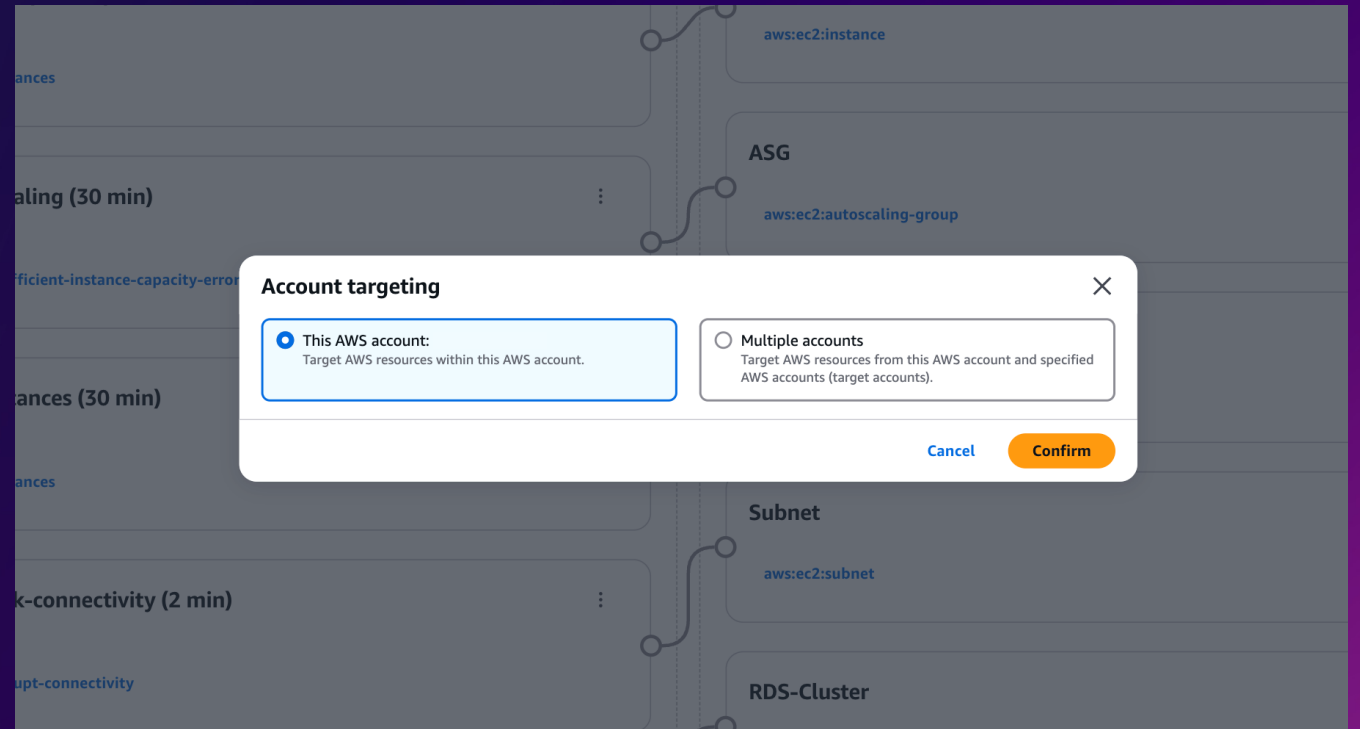
Description
Inject increasing network latency on one or more instances, targeting based on instance tags

EKS Stress: CPU

Description
Inject increasing CPU load on one or more EKS pods, targeting based on cluster and application label

Scenario library

- List of scenarios
- Detailed description
- Create experiment templates
- Bulk edit parameters



Scenario library

- List of scenarios
- Detailed description
- Create experiment templates
- Bulk edit parameters

Create experiment template [Info](#)

Description and name

Description
Add a description for your experiment.

Affect multiple resource types in a single AZ, targeting by tags and explicit ARNs, to approximate power interruption in one AZ.
The description must have 1 to 512 characters.

Name - optional
Creates a tag with a key of 'Name' and a value that you specify.

AZ Availability: Power Interruption
The name must have 1 to 256 characters

Actions and targets [Info](#)

Edit shared parameters of the AZ Availability: Power Interruption scenario
In addition to using the shared parameter editing dialog, you may make changes to the experiment template directly. If you remove all actions or targets required for the scenario, shared parameter editing will be disabled. [Learn more about scenarios](#) [?](#)

[Edit shared parameters](#)

Actions (7) [Hide details](#) **Targets (7)** [Hide details](#)

Pause-EBS-IO (30 min) Action aws:ebs:pause-volume-io	⋮	○	EBS-Volumes aws:ec2:ebs-volume	⋮
Stop-Instances (30 min) Action aws:ec2:stop-instances	⋮	○	EC2-Instances aws:ec2:instance	⋮
Pause-ASG-Scaling (30 min) Action aws:ec2:asg-insufficient-instance-capacity-error	⋮	○	ASG aws:ec2:autoscaling-group	⋮
		○	ASG-EC2-Instances	⋮

AZ Availability: Power interruption scenario

Power interruption scenario to verify application is resilient to an AZ power interruption

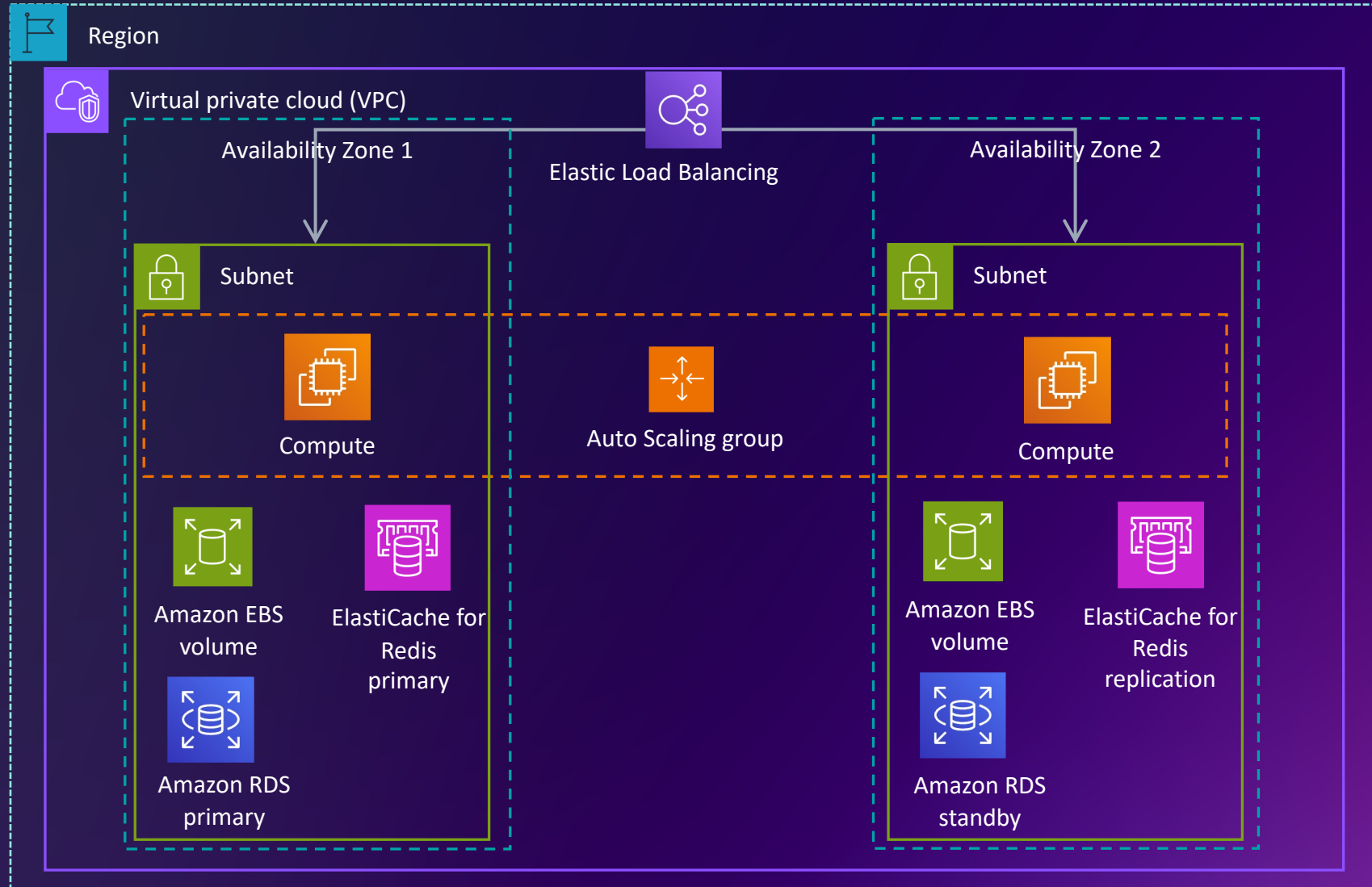
Test Multi-AZ application against the most common symptoms of a complete AZ impairment

AZ Availability: Power Interruption

Description








Affect multiple resource types in a single AZ, targeting by tags and explicit ARNs, to approximate power interruption in one AZ.

Multi-AZ architectures



AZ Availability: Power interruption scenario

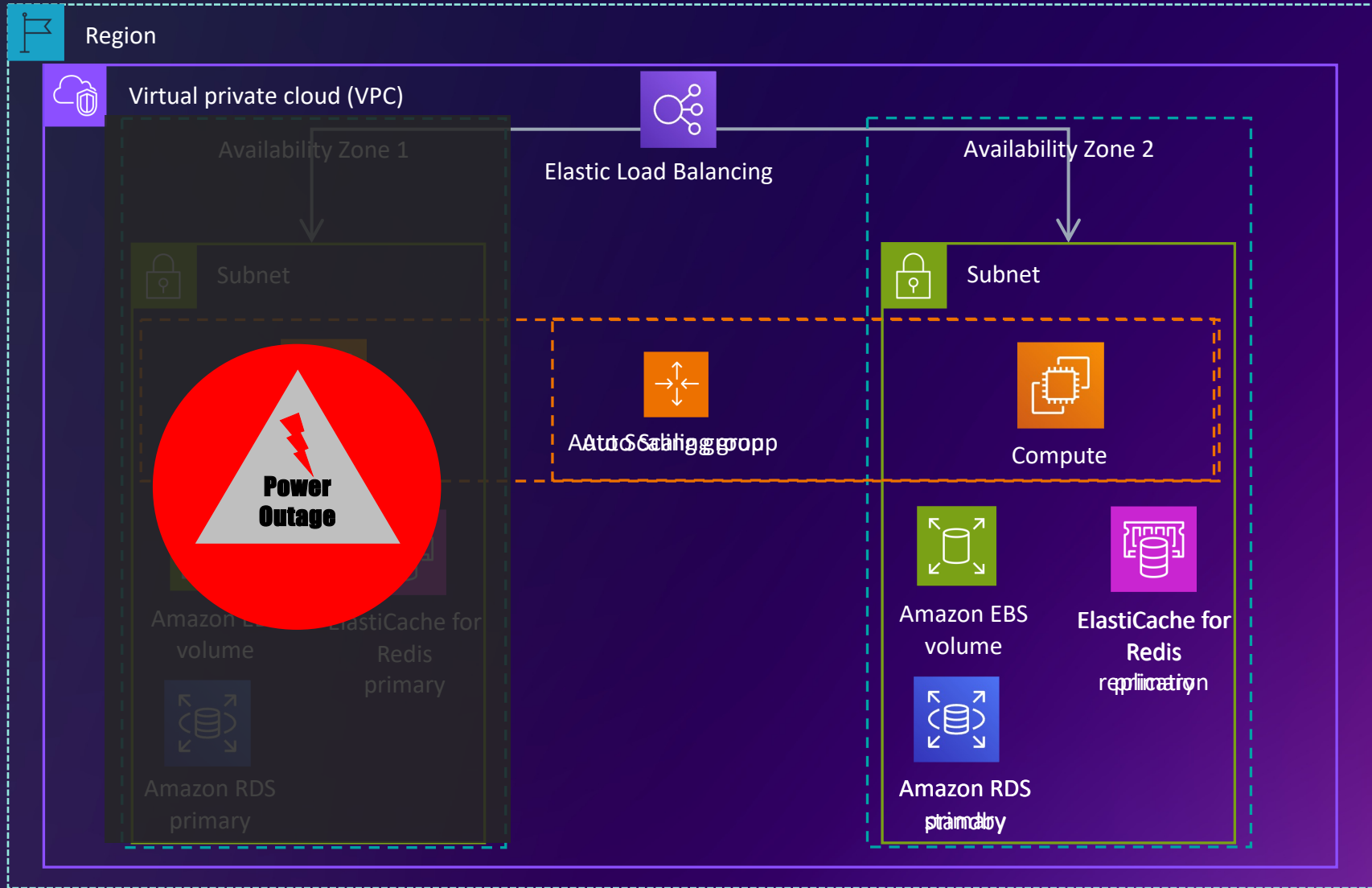
Power interruption scenario to verify application is resilient to an AZ power interruption

	Amazon EC2	Stop instances, insufficient capacity errors for provisioning APIs
	Auto Scaling group	Impair provisioning to AZ via insufficient capacity errors
	Amazon EKS*	Stop EC2 instances (delete pod), prevent EKS provisioning (via Auto Scaling group, Karpenter)
	Amazon ECS*	Stop EC2 instances (stop task), prevent ECS provisioning (via Auto Scaling group)
	Amazon ElastiCache**	AZ unavailability
	Amazon RDS	AZ failover
	Amazon EBS	Pause I/O (unresponsive volume) post-recovery

* No AWS Fargate support

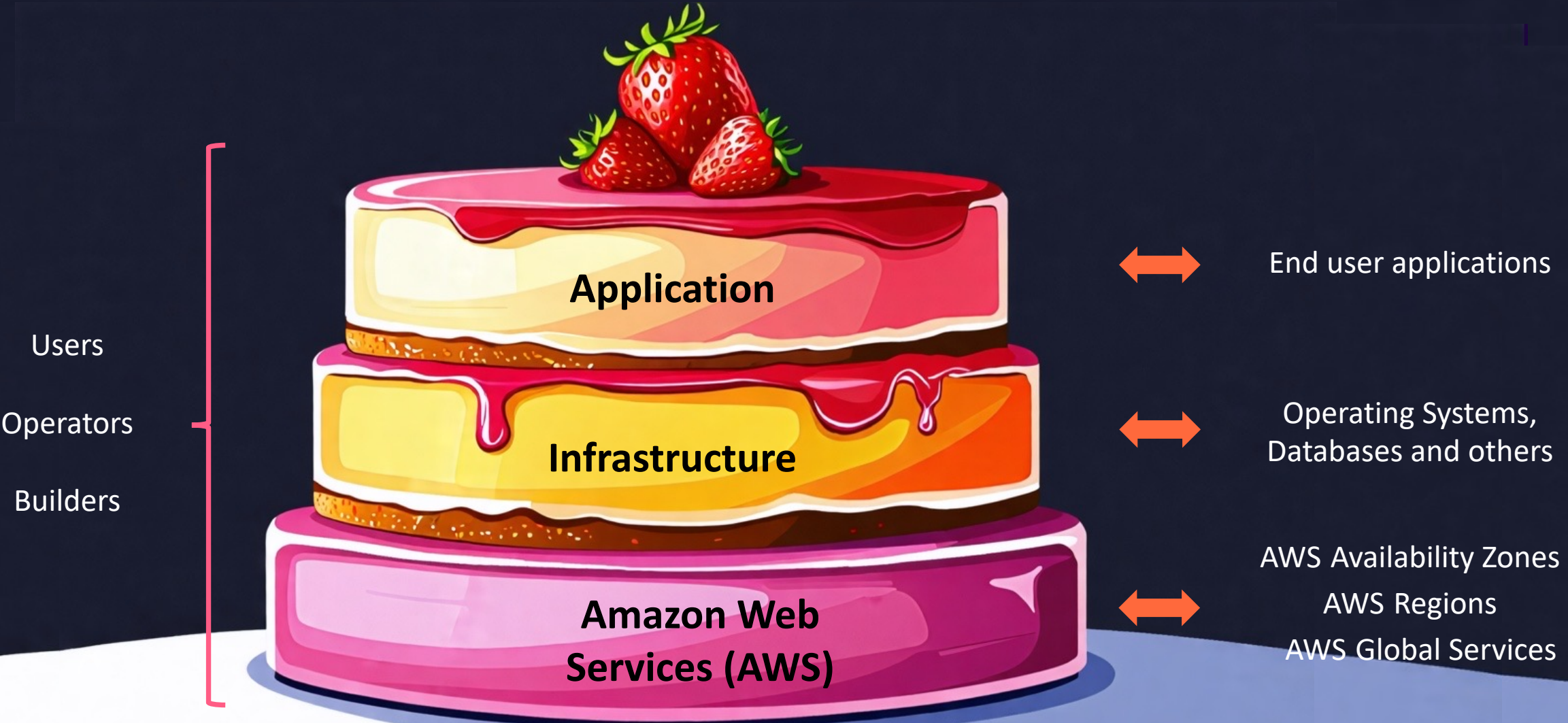
** Redis only

AZ Availability: Power interruption scenario

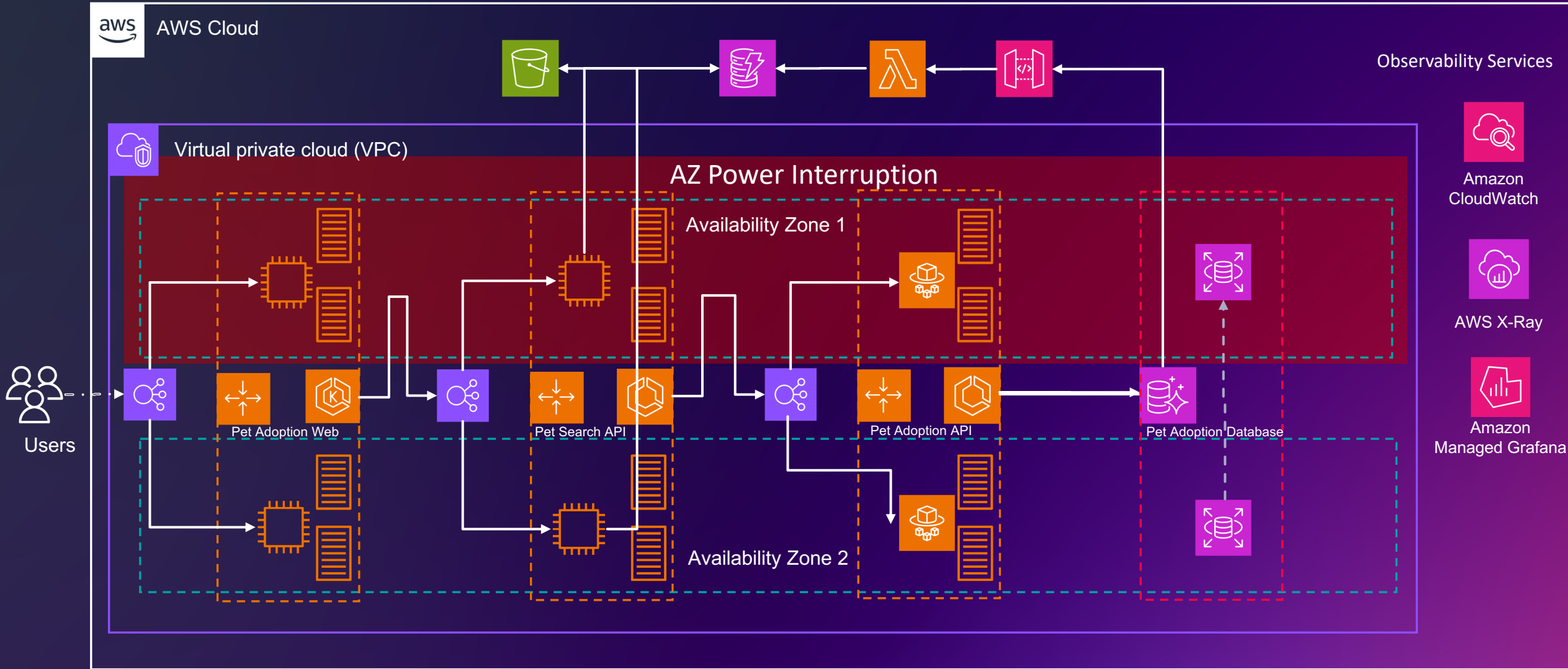


Demo: AZ Availability power interruption

Reminder



Demo - Application architecture



Adopt a Pet, Save a Life!

TYPE
All

COLOR
All

Search



Not available

puppy-black

\$ 89

☆☆☆☆



Take me home

puppy-black

\$ 99

☆☆☆☆



Take me home

puppy-brown

\$ 99

☆☆☆☆



Take me home

Take me home

Cross-Region: Connectivity scenario

Test for unknown dependencies from one region to another

Test that multi-Region applications can operate when their primary Region is inaccessible

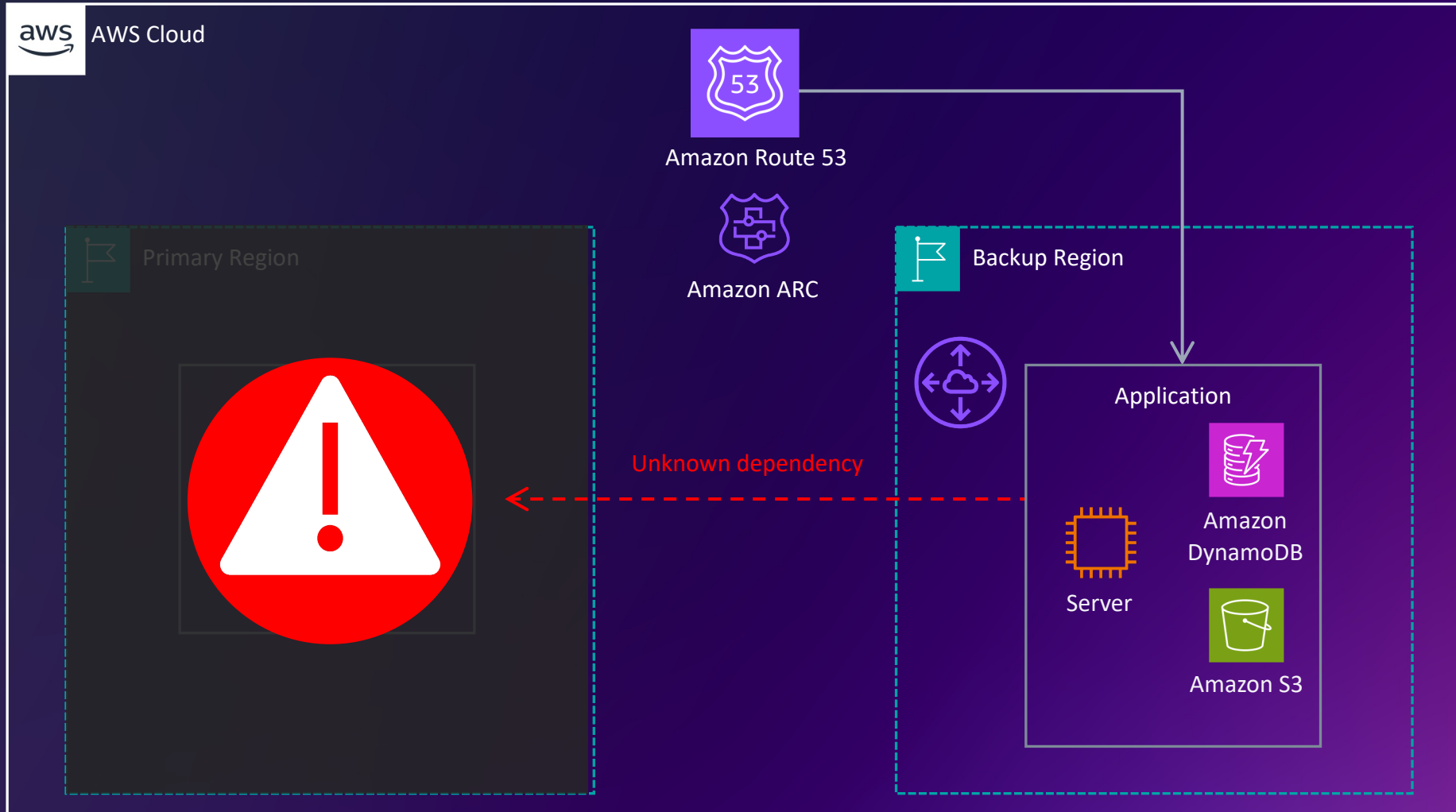
Cross-Region: Connectivity

Description

Block application network traffic from experiment Region to target Region and pause cross-Region replication

Cross-Region: Connectivity scenario

Test for unknown dependencies from one region to another



Cross-Region: Connectivity scenario

Cross-region connectivity disruption to test for hidden dependencies from one region to another



Network* – block cross-Region traffic originating from VPC (internet gateway, VPC peering, AWS Transit Gateway)



Amazon DynamoDB** – pause global tables cross-Region replication

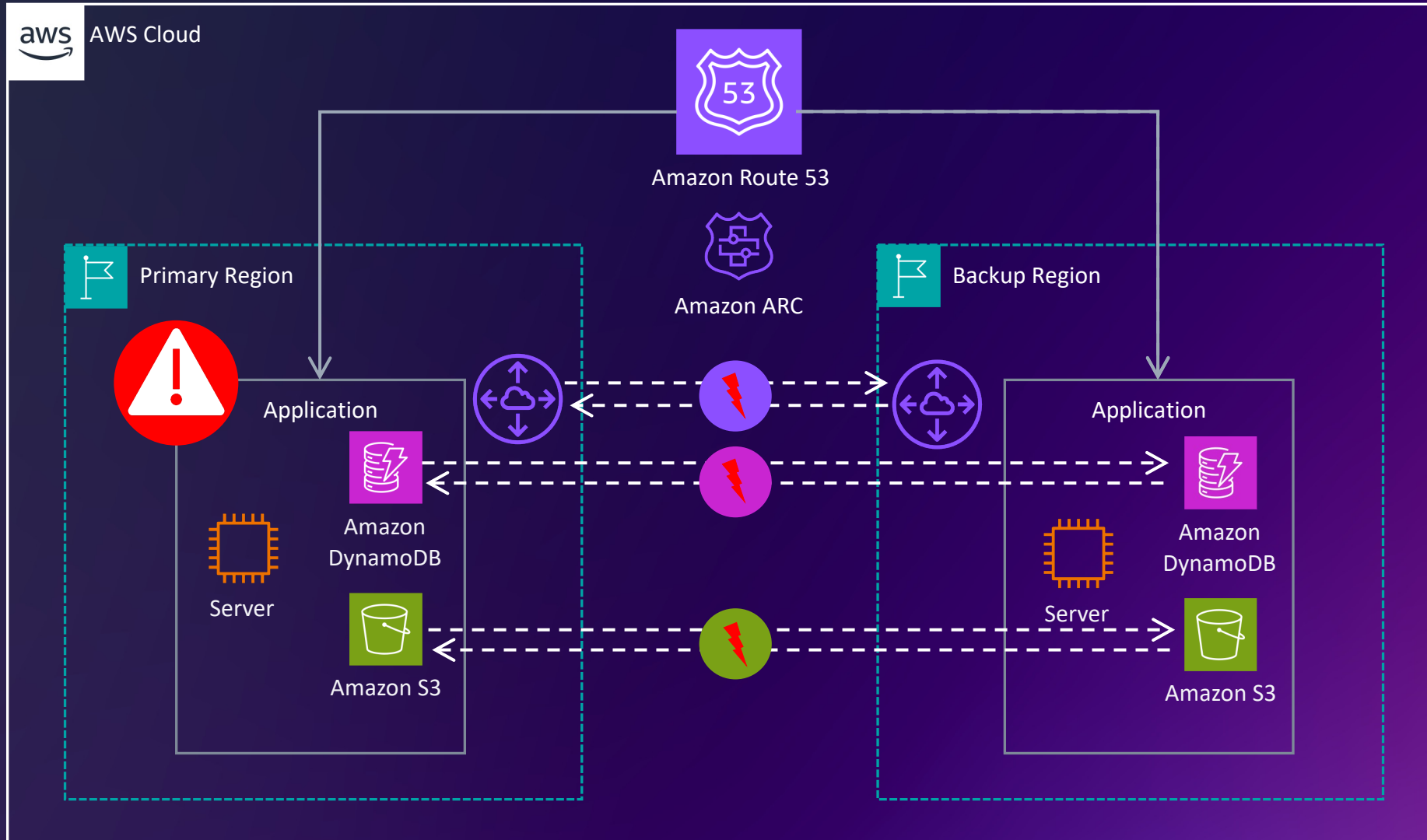


Amazon S3 – pause replication from source to destination buckets

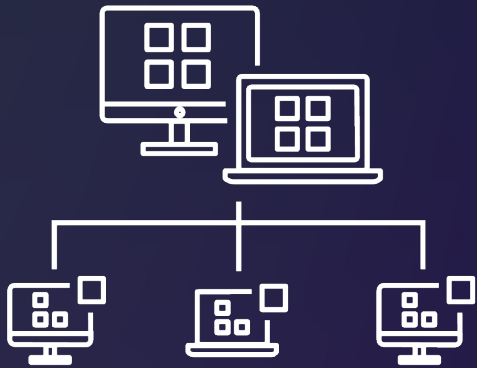
*AWS Direct Connect not supported (hybrid)

**Table encrypted with customer managed keys only

Multi-Region architectures



Operate



Synthetic traffic



Alarming



Operational reviews



Load testing

AWS continuous resilience services

Set objectives



Design & implement



Test & evaluate





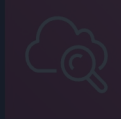


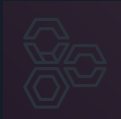



Operate











Respond & learn



Set objectives

 AWS Resilience Hub	 AWS Trusted Advisor	 Amazon CloudWatch
 AWS Well-Architected Framework	 AWS X-Ray	 AWS Well-Architected Tool
 Flow logs	 AWS Elastic Disaster Recovery	 AWS CodePipeline

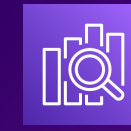
 AWS Trusted Advisor	 AWS Resilience Hub	 AWS Fault Injection Service
 AWS Well-Architected Tool	 Reachability analyzer	 Amazon CloudWatch
 AWS X-Ray		 AWS CodeBuild






AWS Health Dashboard



Amazon CloudWatch



Amazon OpenSearch Service

 AWS Config
 AWS Systems Manager
 Amazon Application Recovery Controller

Respond and learn



Auto remediation



Escalation paths



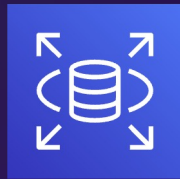
Event management



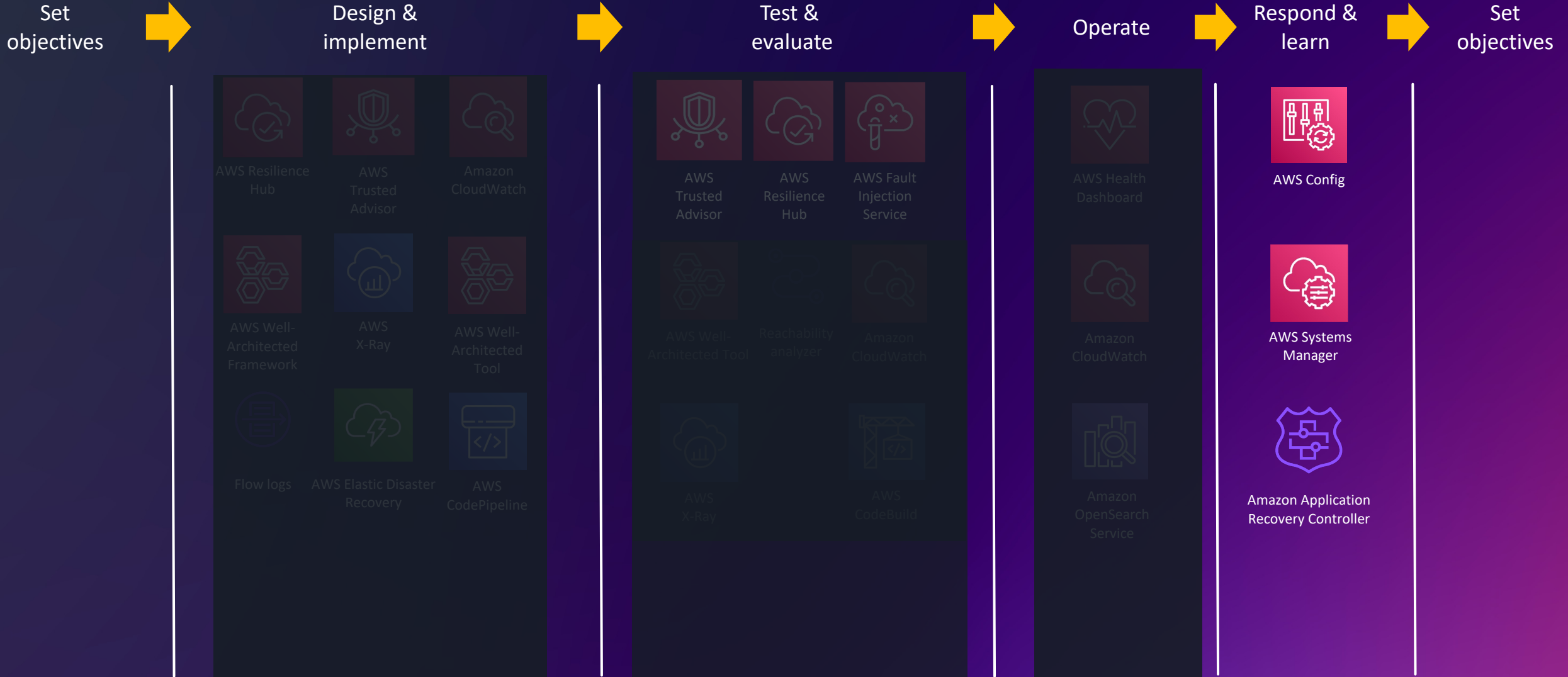
Correction of errors

Automated recovery

- Manual recovery will take much longer
- Use AWS offerings to automate
 - Amazon Auto Scaling
 - Amazon Relational Database Service (Amazon RDS)
 - Amazon Application Recovery Controller
 - Amazon Simple Storage Service (Amazon S3)



AWS continuous resilience services



Further exploration



Workshop:
Controlled chaos for
resilient systems



AWS Resilience Hub: Getting
Started



AWS Fault Injection
Service:
Getting Started



Resilience Lifecycle
Framework



AWS Fault Isolation
Boundaries



Static stability using
Availability Zones

Thank you!

Dragos Madarasan

Adrian Bere

