From Crisis to Confidence

Deploying applications for extreme contingency planning

Joel Ponukumatla

Sr Solutions Architect - SLG/EDU

Wednesday, May 15, 2024

Agenda

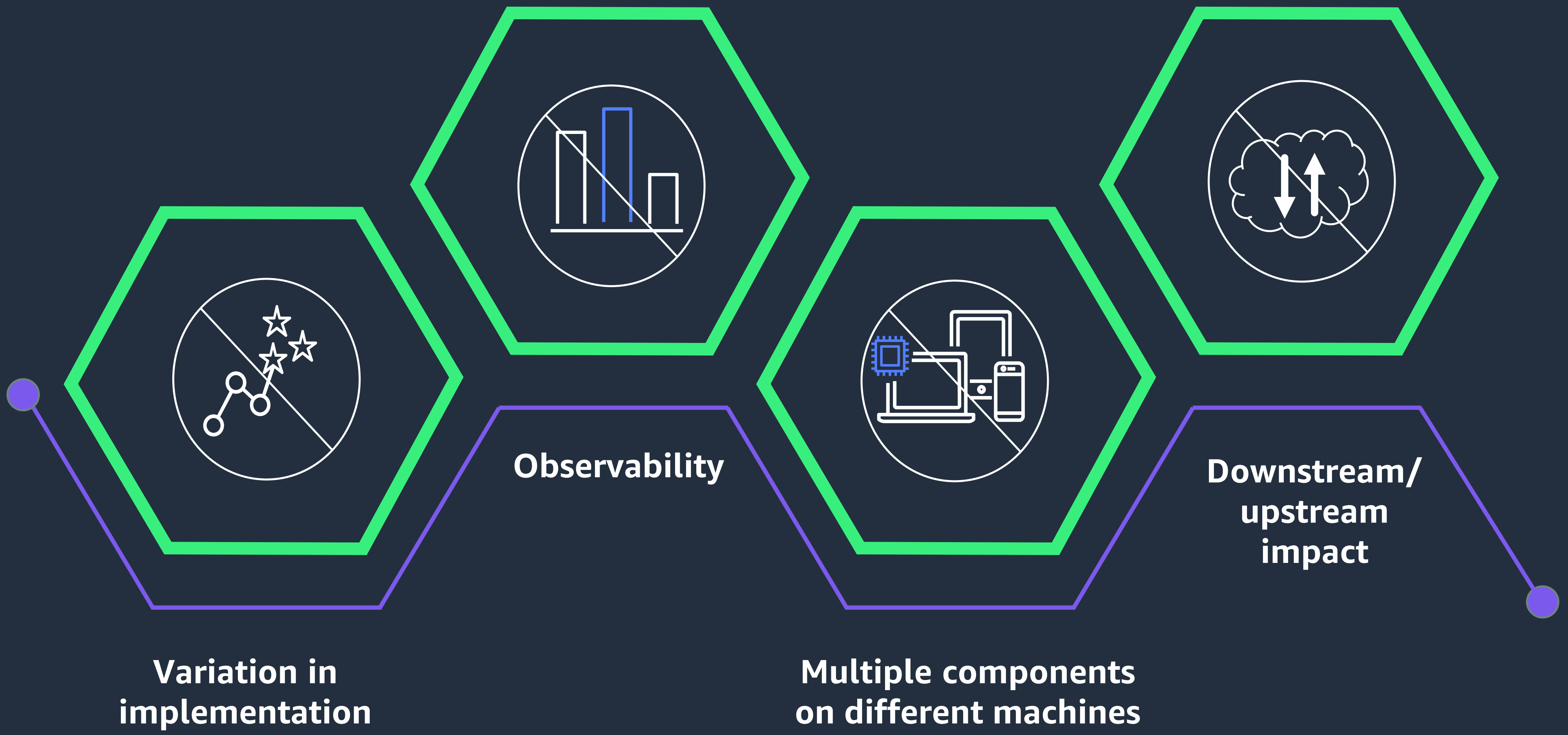
- Failures, resilience, and shared responsibility model
- Resilience of the AWS Cloud
- Resilience of customer workloads in the cloud
- Critical reliability best practices

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

Dr. Werner Vogels

CTO, Amazon.com

Challenges with distributed systems



Variation in implementation

Observability

Multiple components on different machines

Downstream/ upstream impact

Categories of failure

Code deployments & configuration

such as bad deployment, cred expiration

Core infrastructure

such as datacenter failure, host failure

Data and state

such as data corruption

Highly unlikely scenarios

such as all of internet failure, environmental disasters, supplier failure

← Likelihood →

Probable — Rare

```
# rm -rf *
```

Four essential capabilities in a resilient system

Anticipate

Monitoring

Responding

Learning

"Resilience Engineering in Practice," by Hollnagel, Pariès, Woods, Wreathall

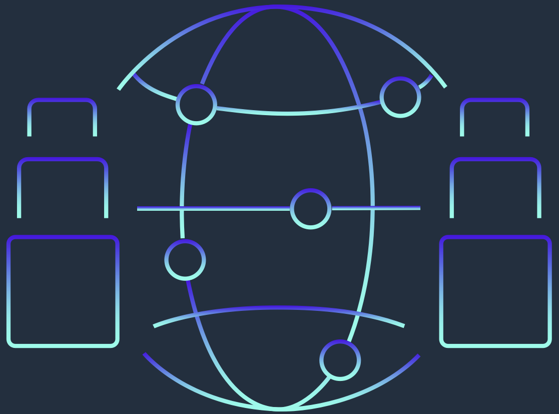
Testing resilience

Resilience: The ability of an application to resist or recover from certain types of faults or load spikes

bit.ly/resilience_essentials

Design principles for reliability

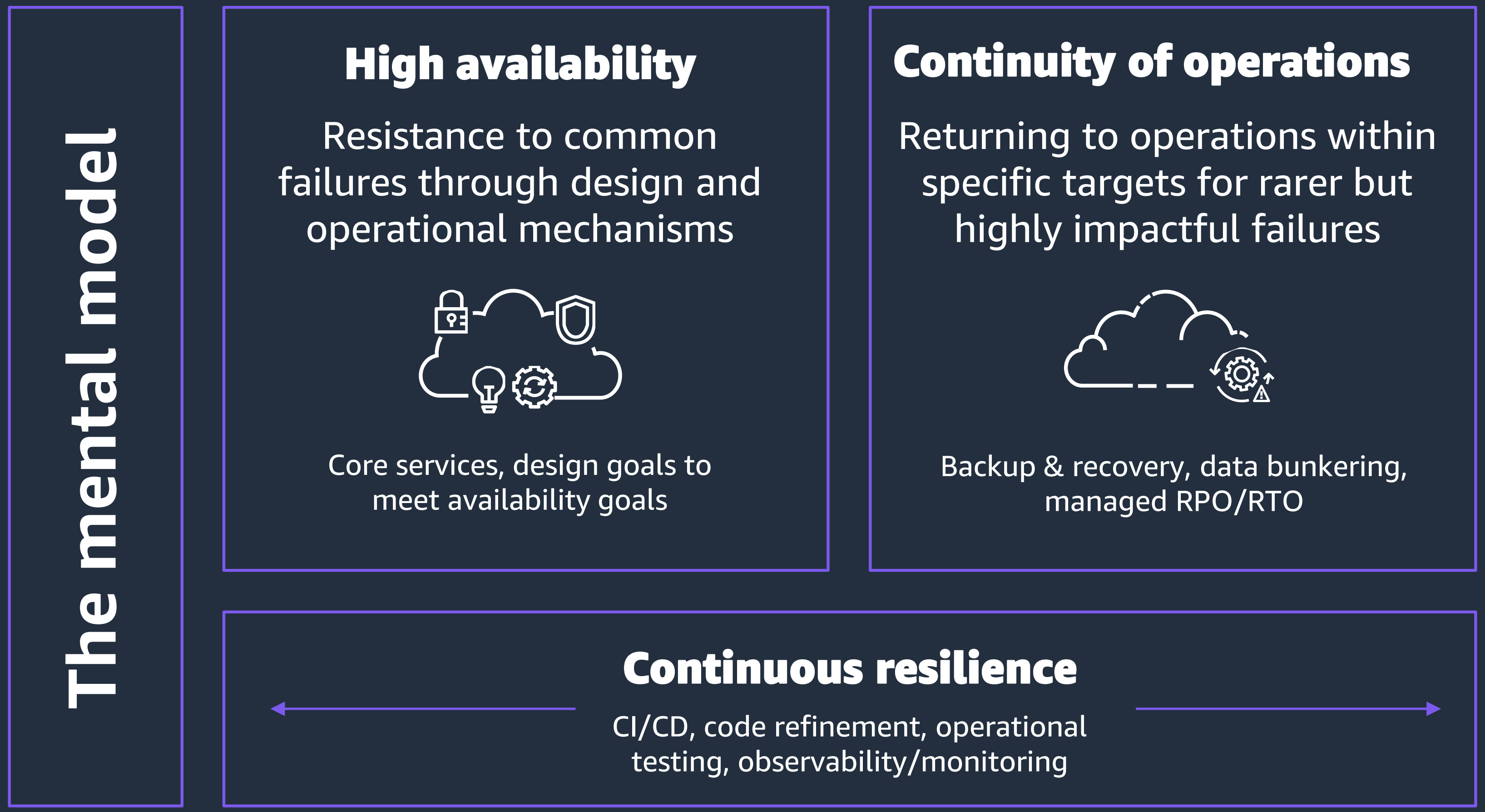
- Automatically recover from failure
- Test recovery procedures



Resiliency of the cloud: Culture of reliability at AWS

What is resilience?

Resilience refers to the ability of workloads to respond to and quickly recover from failures



The mental model

High availability

Resistance to common failures through design and operational mechanisms

Core services, design goals to meet availability goals

Continuity of operations

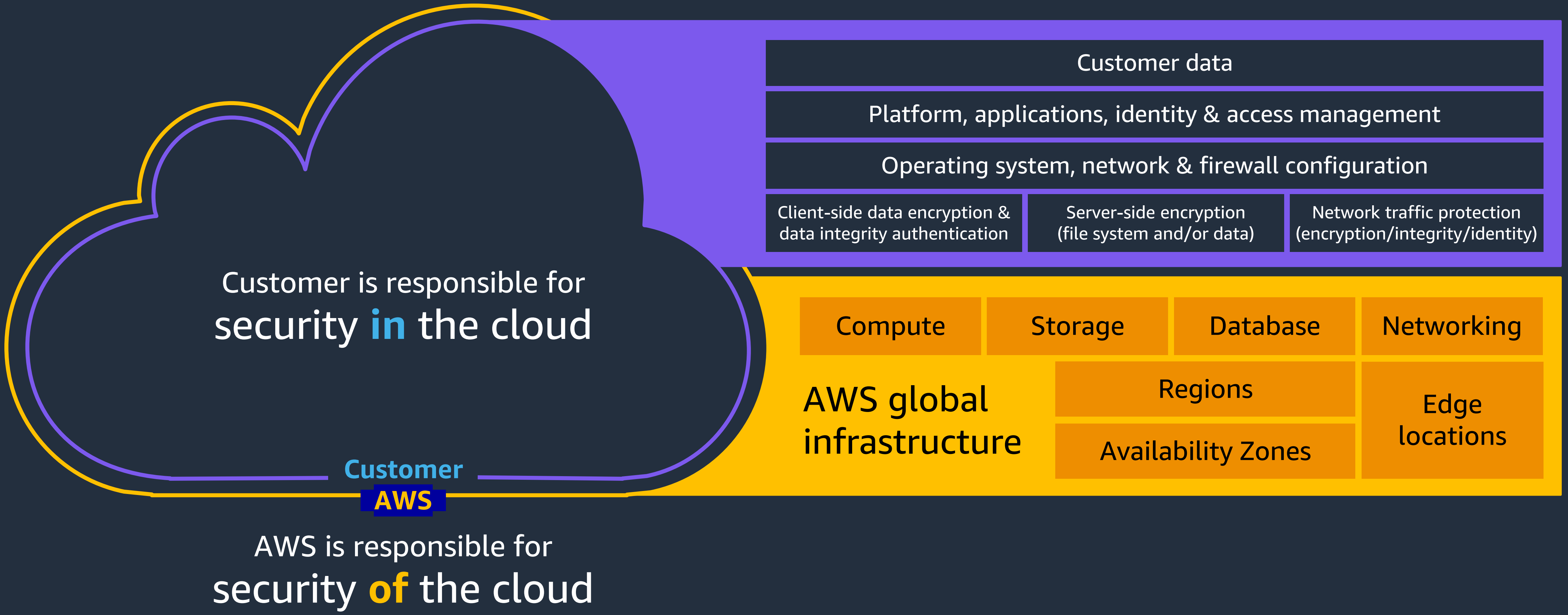
Returning to operations within specific targets for rarer but highly impactful failures

Backup & recovery, data bunkering, managed RPO/RTO

Continuous resilience

CI/CD, code refinement, operational testing, observability/monitoring

Understanding the AWS shared responsibility model



Customer data

Platform, applications, identity & access management

Operating system, network & firewall configuration

Client-side data encryption & data integrity authentication	Server-side encryption (file system and/or data)	Network traffic protection (encryption/integrity/identity)

Compute | Storage | Database | Networking

AWS global infrastructure

Regions

Availability Zones

Edge locations

Customer is responsible for security **in** the cloud

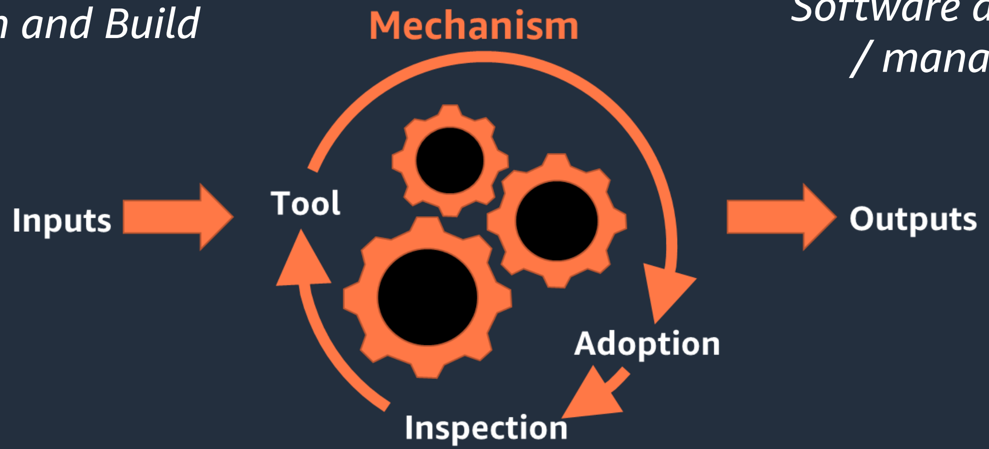
AWS is responsible for security **of** the cloud

Shared responsibility model for resilience

Engineering Culture:
Clear scope of ownership

PR/FAQ,
Design and Build

Software deployment
/ management



Mechanism

Inputs

Tool

Outputs

Adoption

Inspection

Incident Management, COE (learnings, actions), Weekly operations meetings, Principal Engineers

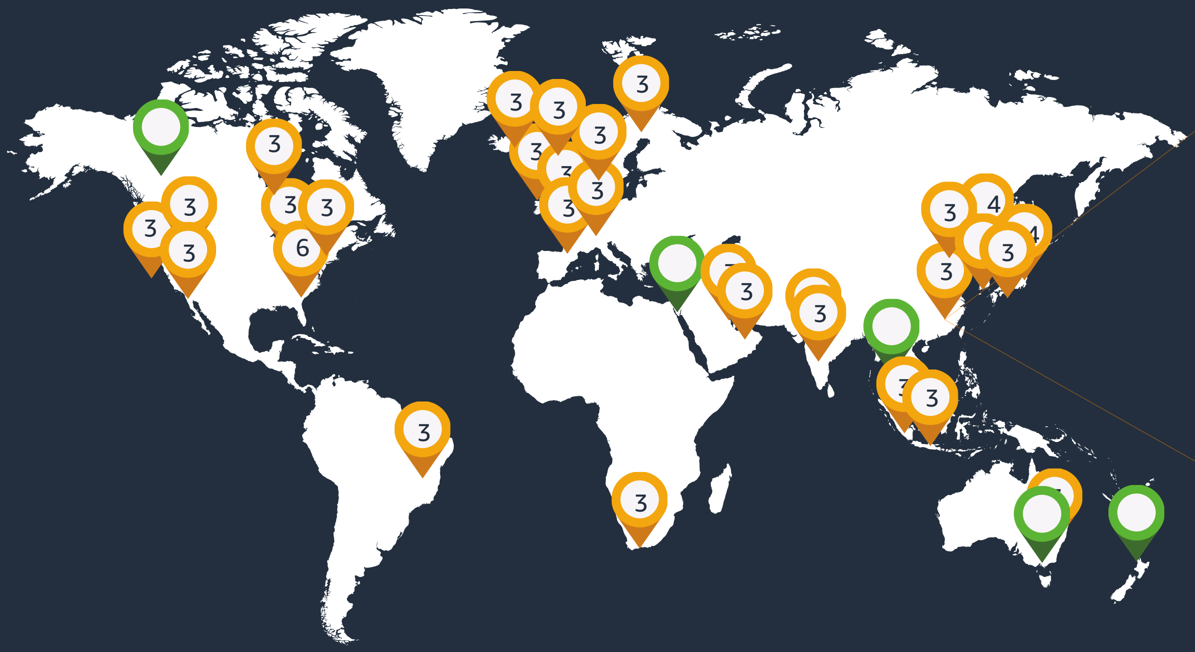
Design and implement
Resilient architectures

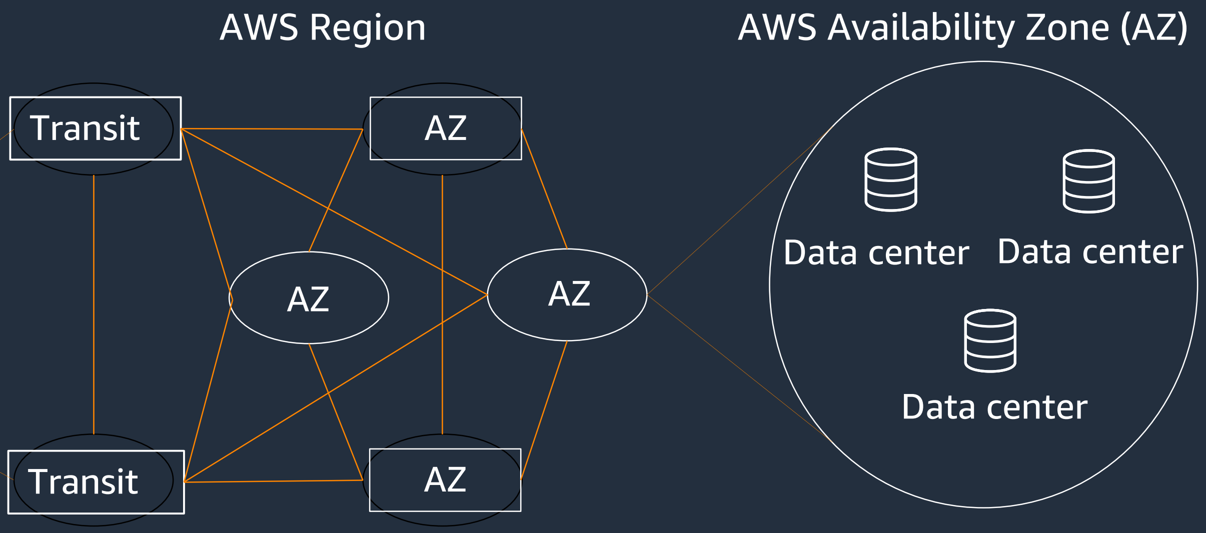
Fault isolation boundaries

- ✓ Workload isolation
- ✓ Failure containment
- ✓ Scale out vs. scale up
- ✓ Testability
- ✓ Manageability

Enabling resilience of the cloud

We offer 200+ fully featured services from 96 Availability Zones (AZs) across 30 Regions, globally





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

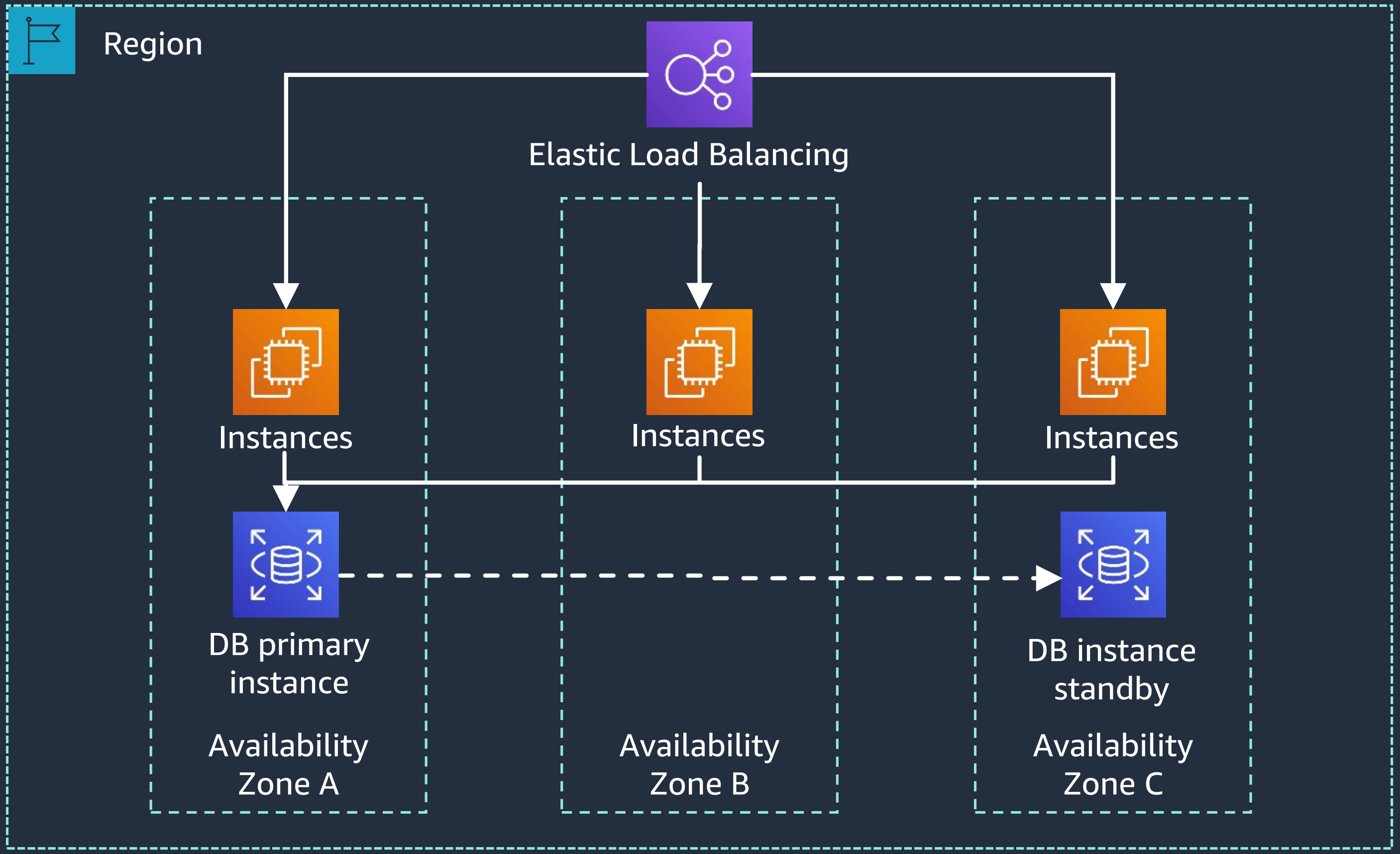
Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities

Region & number of Availability Zones

Announced Regions

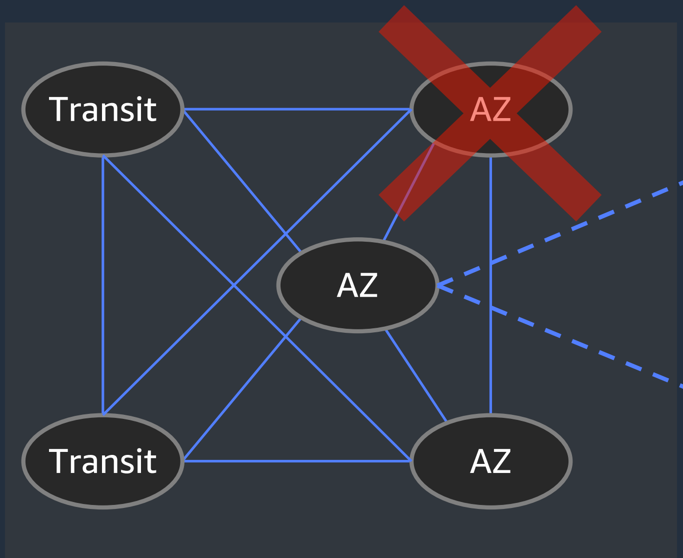
Canada West, Israel, Thailand, Melbourne, Auckland

Multi-AZ application



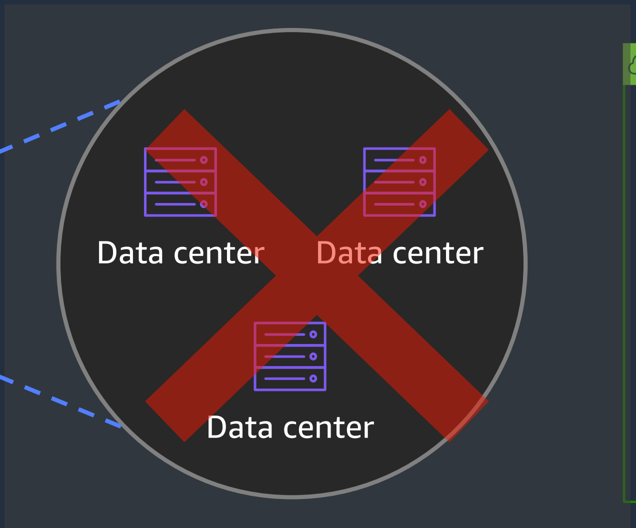
Multi-AZ for Disaster Recovery (DR)

Each AWS Region has multiple AZs

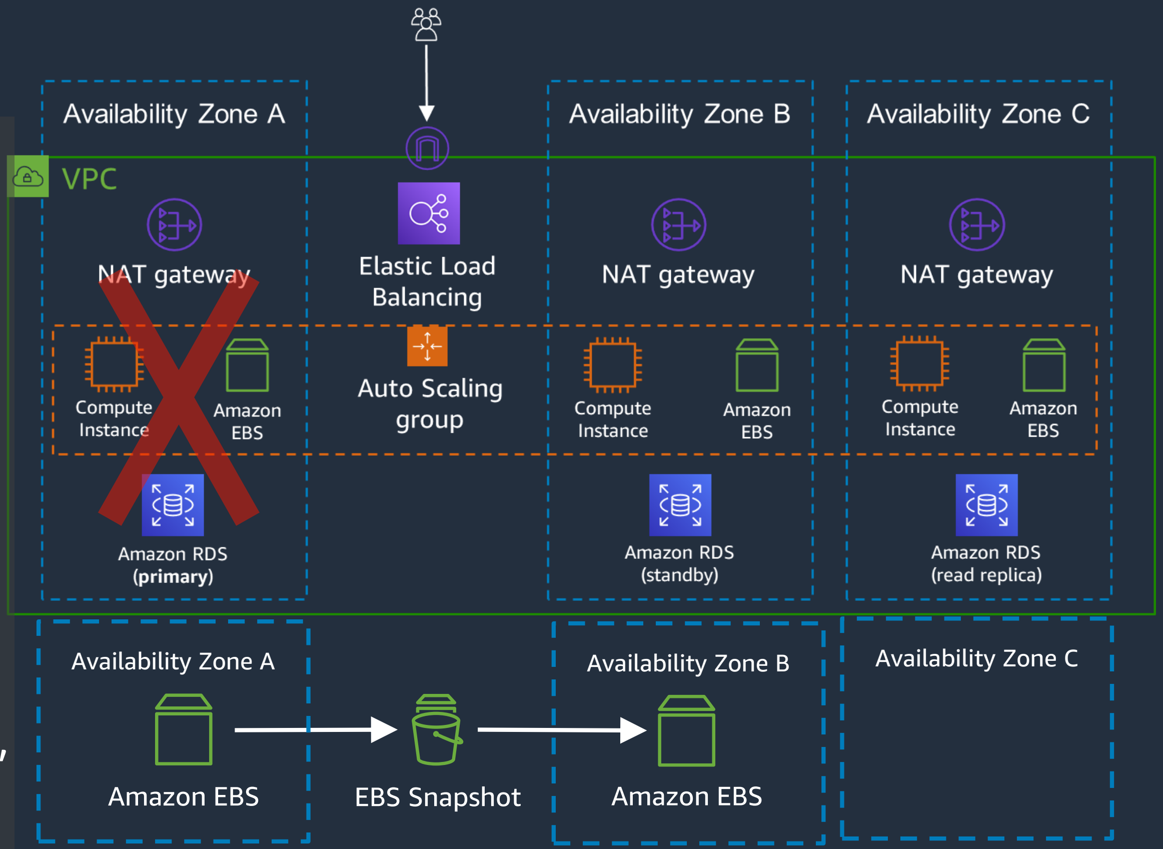


A **Region** is a physical location in the world

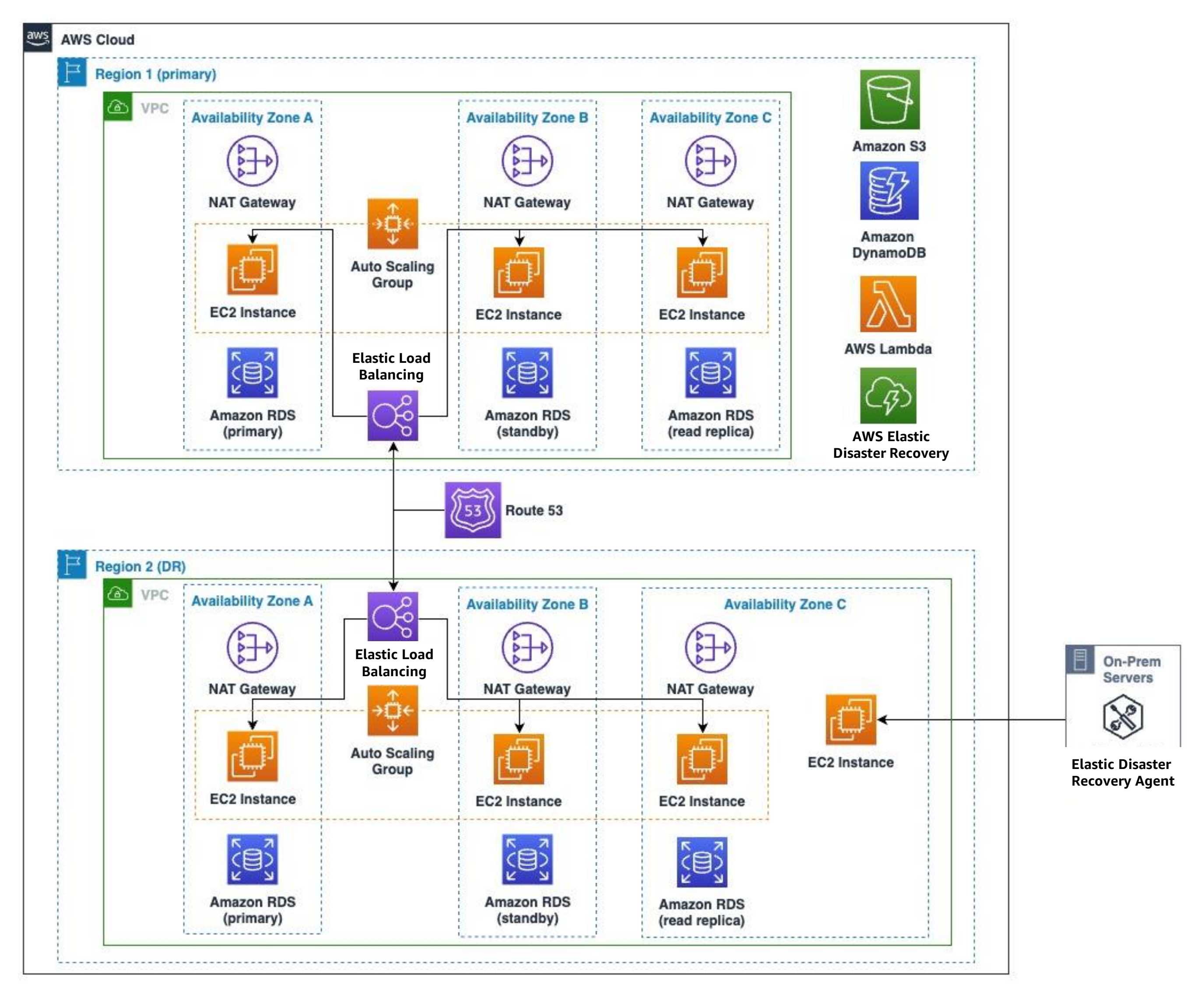
Each AZ includes one or more discrete data centers



Data centers, each with redundant power, networking, and connectivity, housed in separate facilities

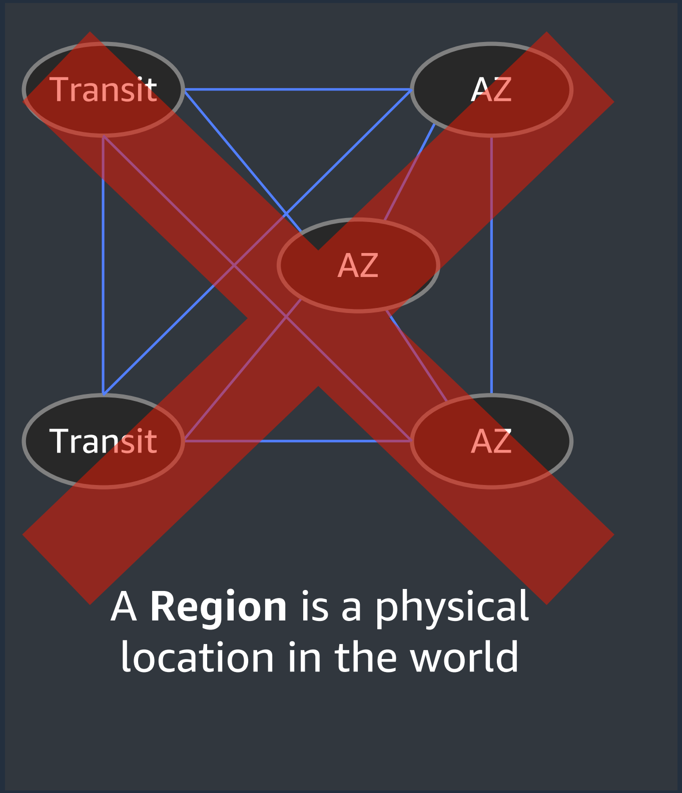


Multi-Region Architecture

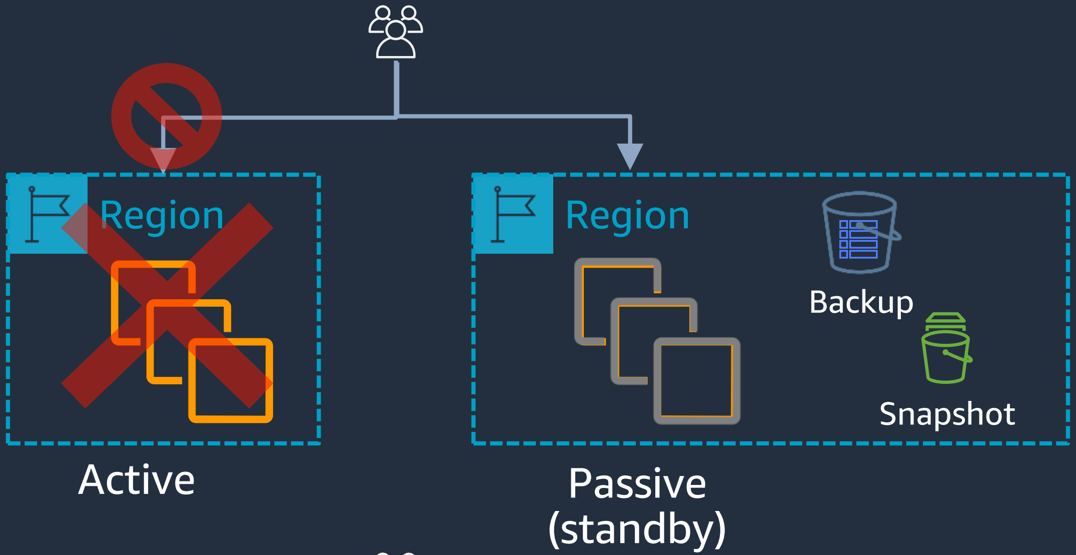


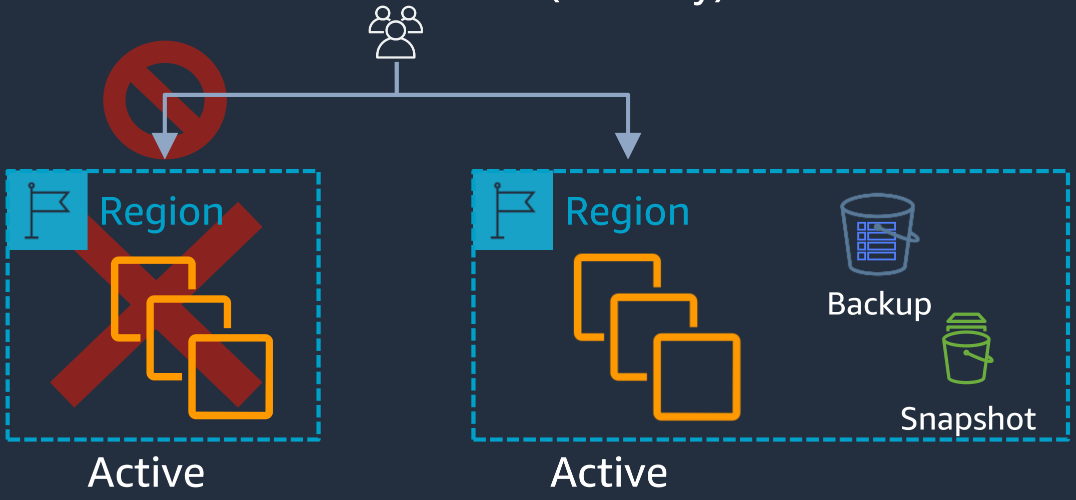
Multi-Region for Disaster Recovery (DR)

Each AWS Region has multiple AZs



A **Region** is a physical location in the world





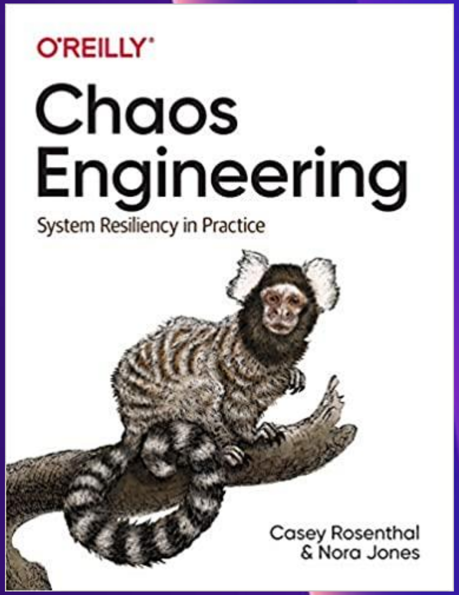
Continuous Resilience: Finding the unknowns

> **Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system’s capability to withstand turbulent conditions in production.**

Casey Rosenthal and Nora Jones

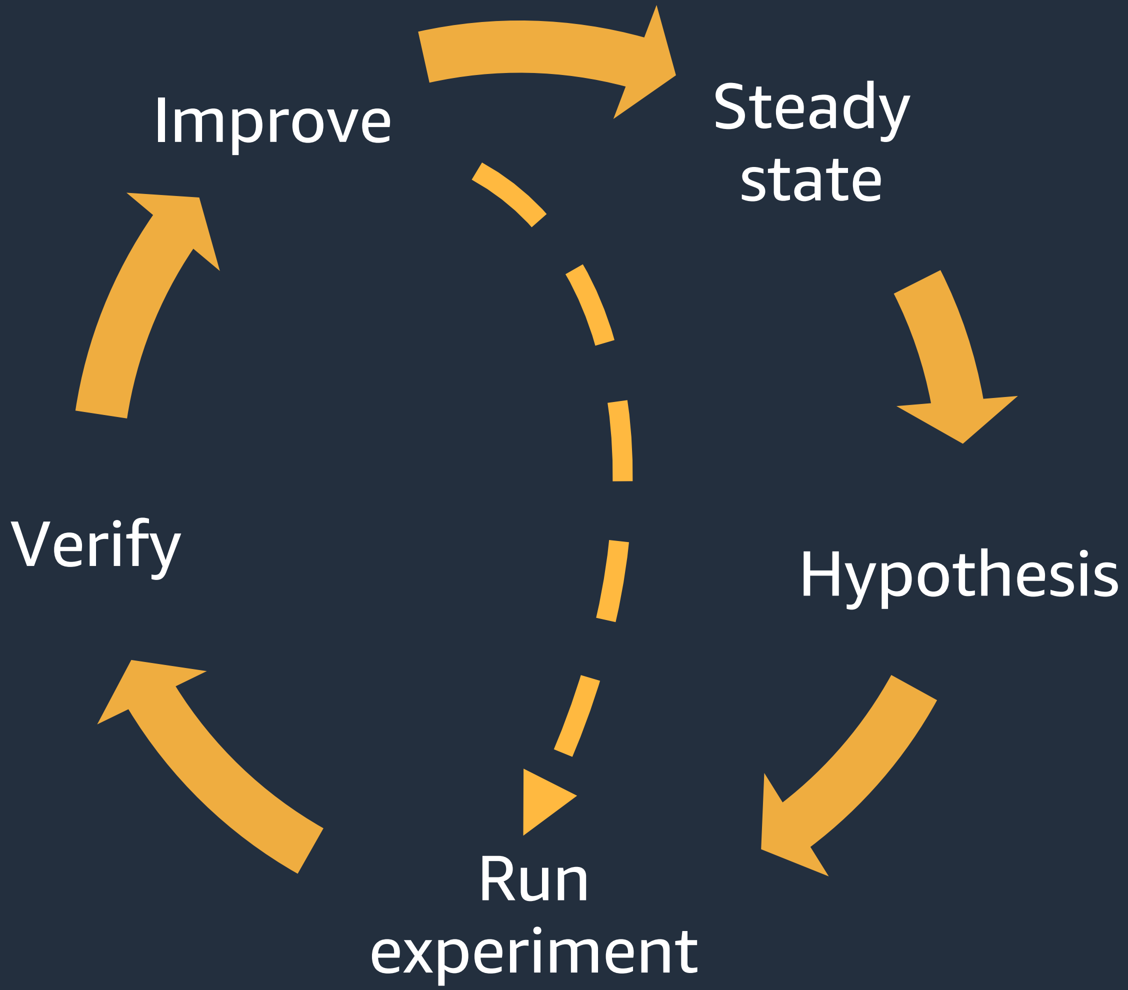
Chaos Engineering: System Resiliency in Practice

principlesofchaos.org



Chaos engineering

A SCIENTIFIC METHOD



Steady state

Hypothesis

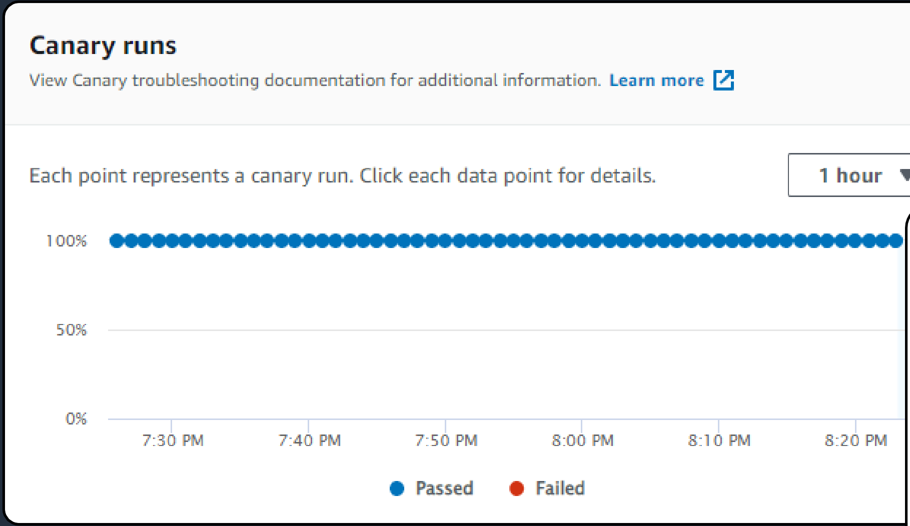
Run experiment

Verify

Improve

Steady state

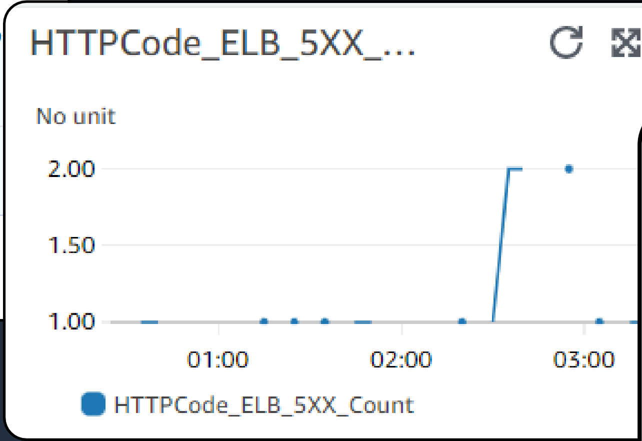
- Your workload exhibits steady state if it is operating reliably and as expected
- Not necessarily no impact – This may mean impact is within acceptable limits

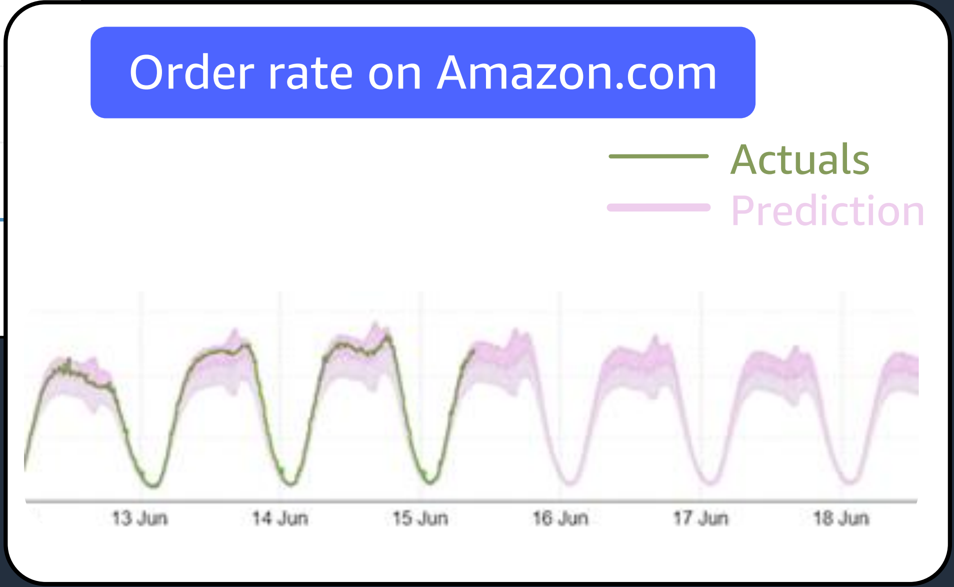


Canary runs

View Canary troubleshooting documentation for additional information. Learn more

Each point represents a canary run. Click each data point for details.





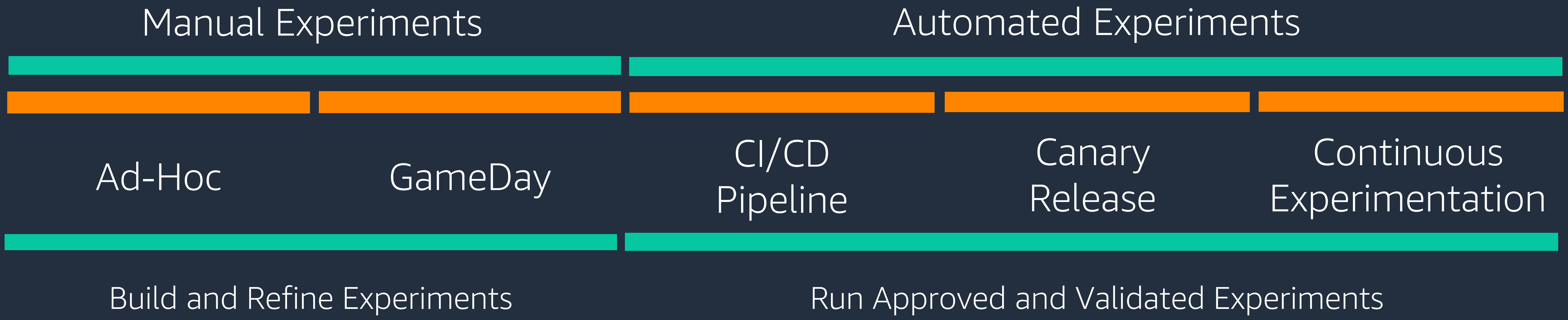
Hypothesis

If [fault] occurs, the [name] workload
will [mitigating controls]
to maintain [steady state metric]

If **a single Amazon EC2 instance failure** occurs,
the **AnyCompany Order System** workload will **send traffic to only healthy instances and replace the unhealthy one** to maintain a **less than 0.01% increase in server-side (5xx) errors**

When and Where to Run Chaos experiments

Test & Evaluate:
Types of Resilience Experimentation



Manual Experiments

Automated Experiments

Ad-Hoc

GameDay

CI/CD Pipeline

Canary Release

Continuous Experimentation

Build and Refine Experiments

Run Approved and Validated Experiments

Chaos Engineering

Chaos engineering is the discipline of experimenting on a software system in ~~production~~ dev/test in order to build confidence in the system's capability to withstand turbulent and unexpected conditions.

Wikipedia

Accessed 22 September 2021. https://en.wikipedia.org/wiki/Chaos_engineering

Get close to production

Traffic patterns

Test in production

Synthetic traffic

Environment

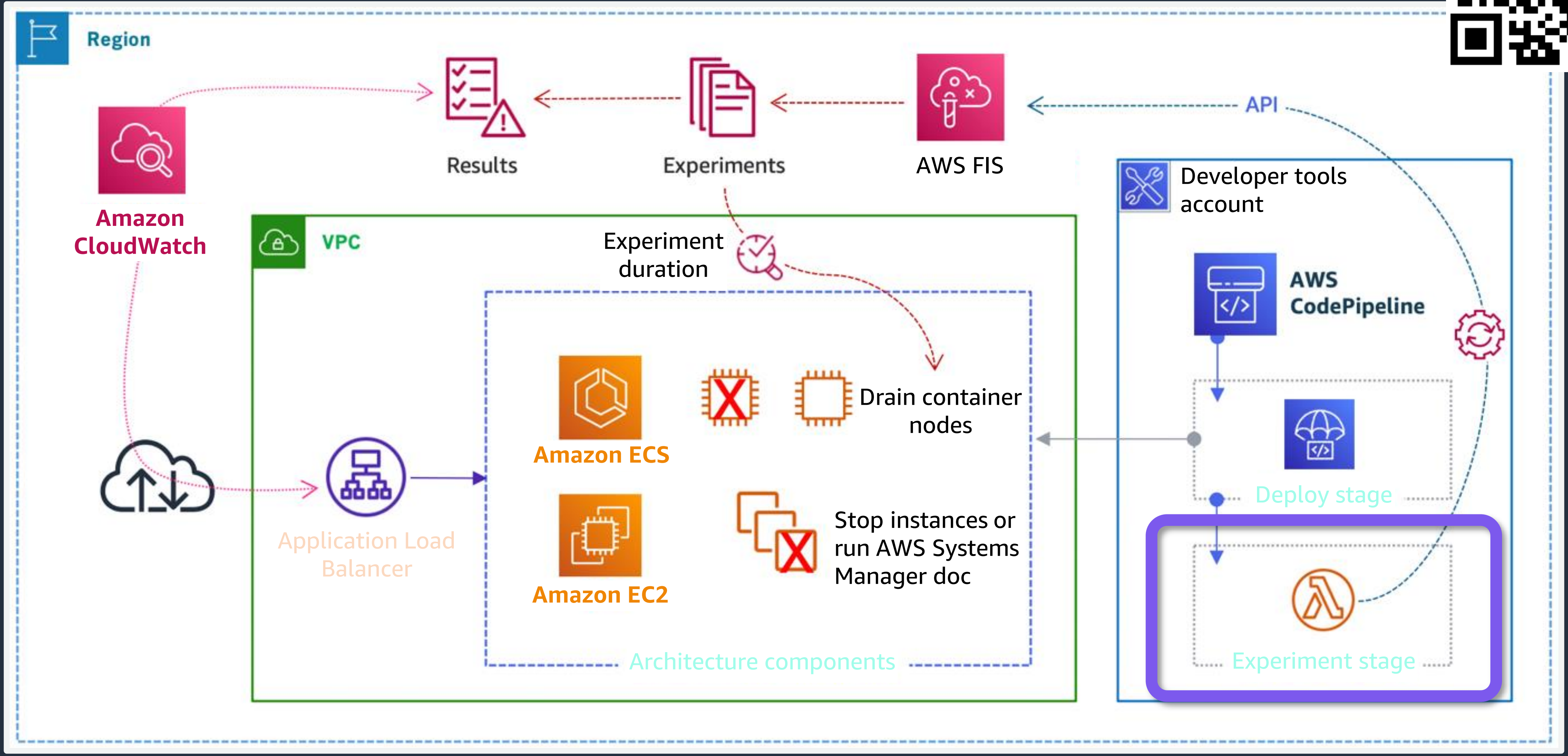
Test in production

Cloud-deployed test environment

Events

Learn from production

Run these experiments regularly

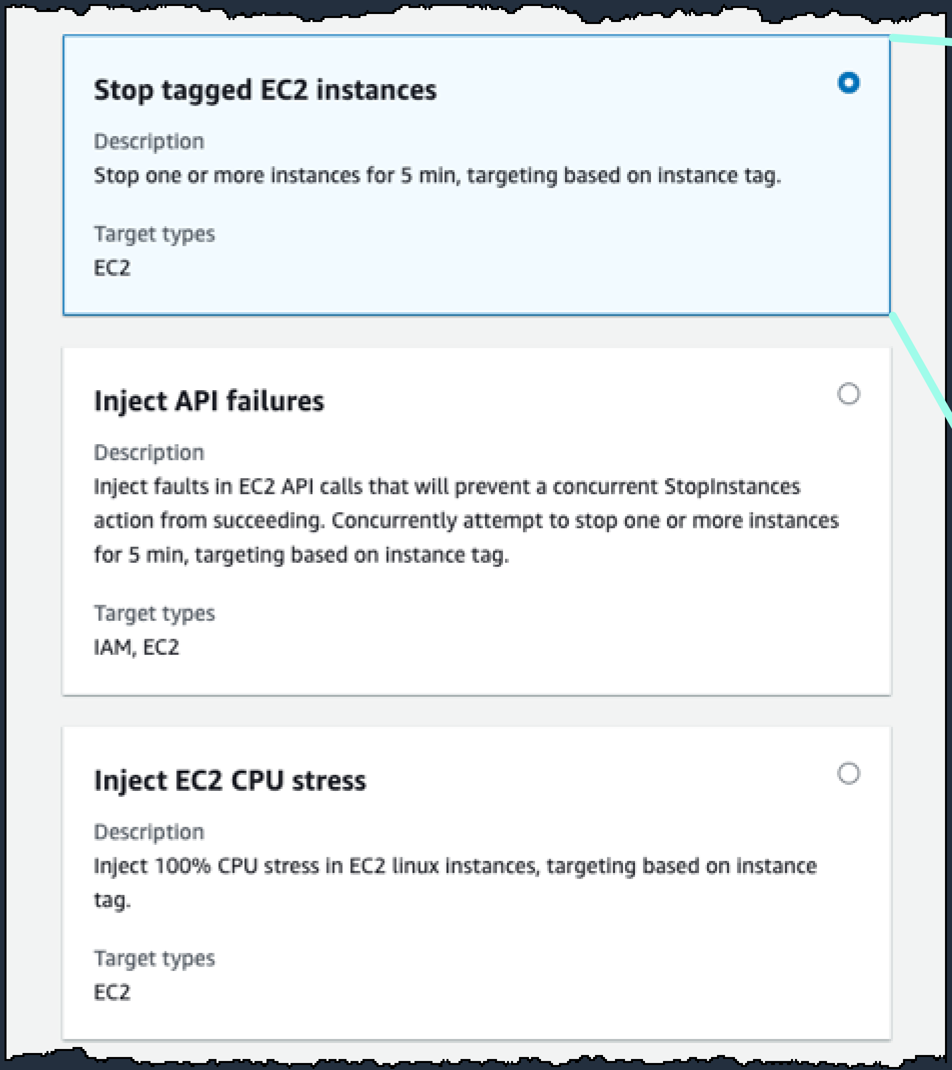


How do I run these experiments on AWS?

Run experiment

AWS Fault Injection Service (AWS FIS)

Use AWS FIS scenario library



Stop tagged EC2 instances ●

Description

Stop one or more instances for 5 min, targeting based on instance tag.

Target types

EC2

Inject API failures ○

Description

Inject faults in EC2 API calls that will prevent a concurrent StopInstances action from succeeding. Concurrently attempt to stop one or more instances for 5 min, targeting based on instance tag.

Target types

IAM, EC2

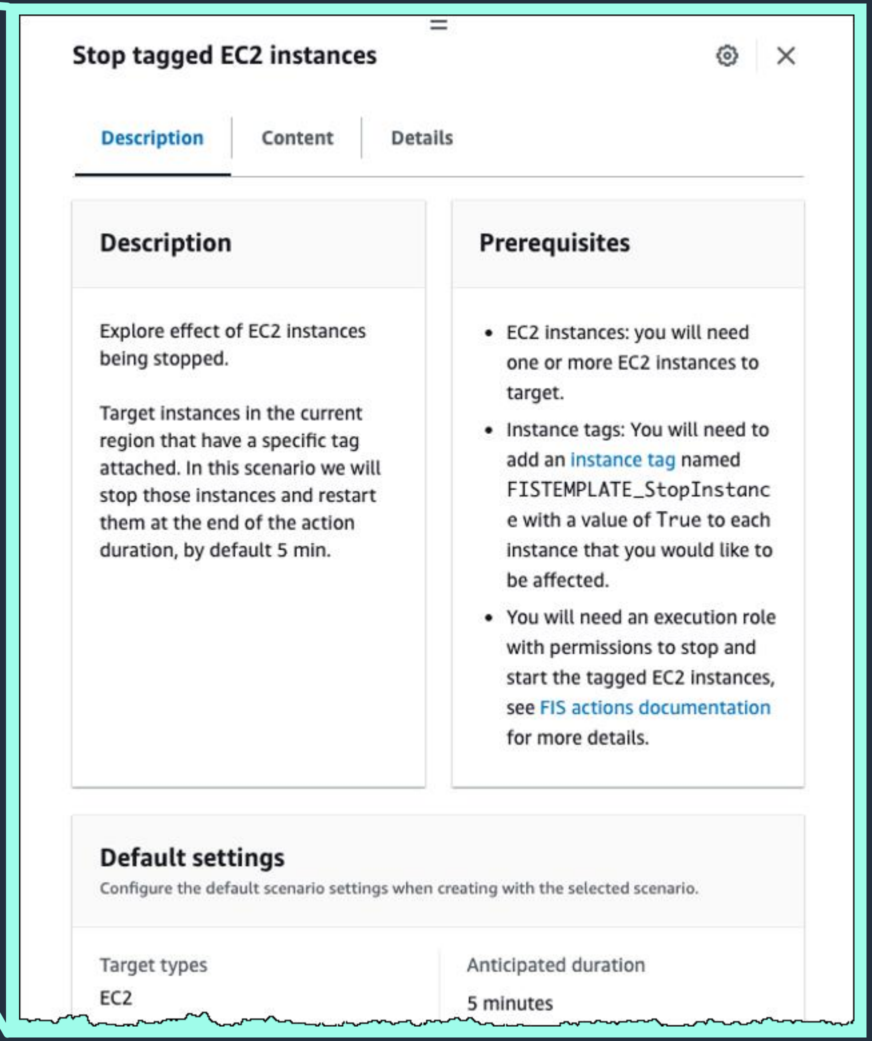
Inject EC2 CPU stress ○

Description

Inject 100% CPU stress in EC2 linux instances, targeting based on instance tag.

Target types

EC2



Stop tagged EC2 instances

Description

Explore effect of EC2 instances being stopped.

Target instances in the current region that have a specific tag attached. In this scenario we will stop those instances and restart them at the end of the action duration, by default 5 min.

Prerequisites

- EC2 instances: you will need one or more EC2 instances to target.
- Instance tags: You will need to add an instance tag named `FISTEMPLATE_StopInstance` with a value of `True` to each instance that you would like to be affected.
- You will need an execution role with permissions to stop and start the tagged EC2 instances, see FIS actions documentation for more details.

Default settings

Configure the default scenario settings when creating with the selected scenario.

Target types

EC2

Anticipated duration

5 minutes

FIS targets

Compute

- Amazon Elastic Compute Cloud (Amazon EC2)
- New Amazon EC2 Auto Scaling
- Amazon Elastic Kubernetes Service (Amazon EKS)
- Amazon Elastic Container Service (Amazon ECS)

Storage

- New Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic Block Store (Amazon EBS)

Networking

- New Amazon Virtual Private Cloud (Amazon VPC)
- New AWS Transit Gateway

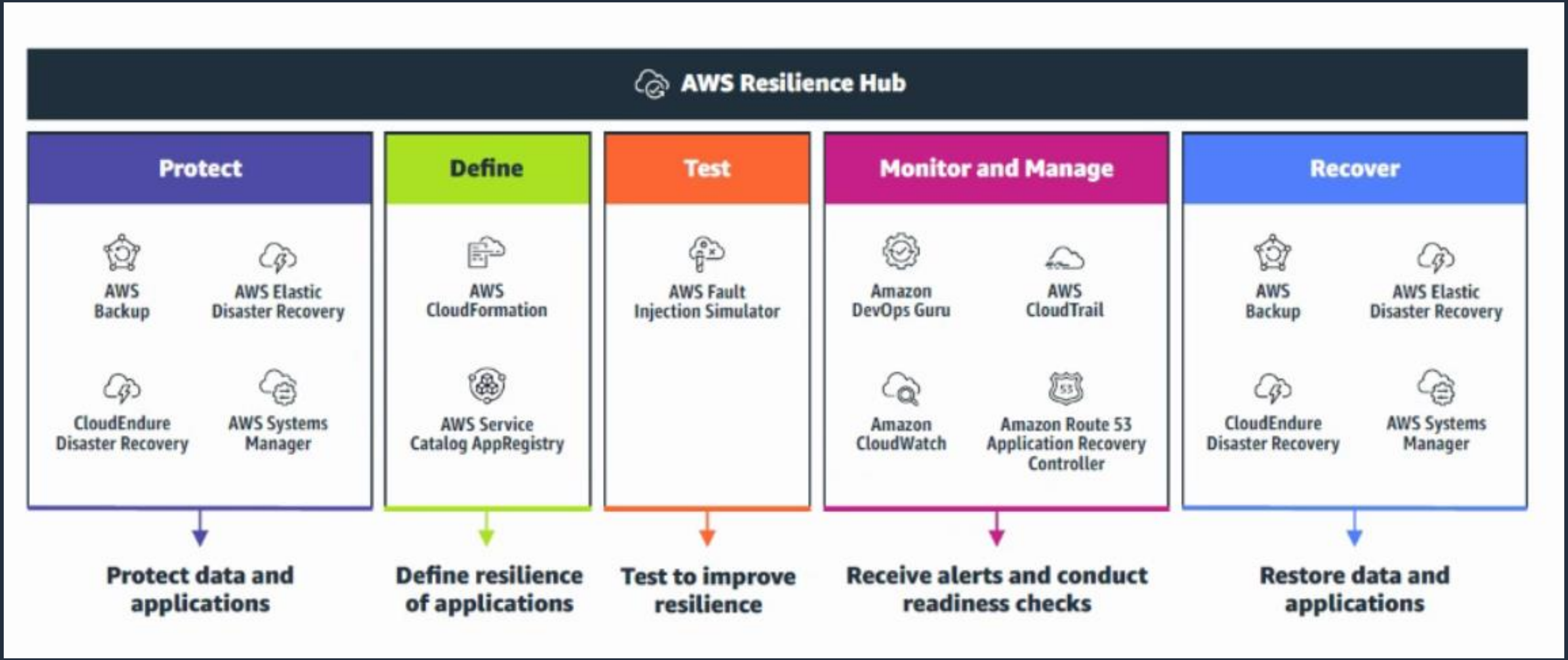
Database

- Amazon Relational Database Service (Amazon RDS)
- New Amazon DynamoDB
- New Amazon ElastiCache

Management

- Amazon CloudWatch
- AWS Systems Manager

How AWS helps you design resilient workloads



AWS Resilience Hub

Protect

- AWS Backup
- AWS Elastic Disaster Recovery
- CloudEndure Disaster Recovery
- AWS Systems Manager

Protect data and applications

Define

- AWS CloudFormation
- AWS Service Catalog AppRegistry

Define resilience of applications

Test

- AWS Fault Injection Simulator

Test to improve resilience

Monitor and Manage

- Amazon DevOps Guru
- AWS CloudTrail
- Amazon CloudWatch
- Amazon Route 53 Application Recovery Controller

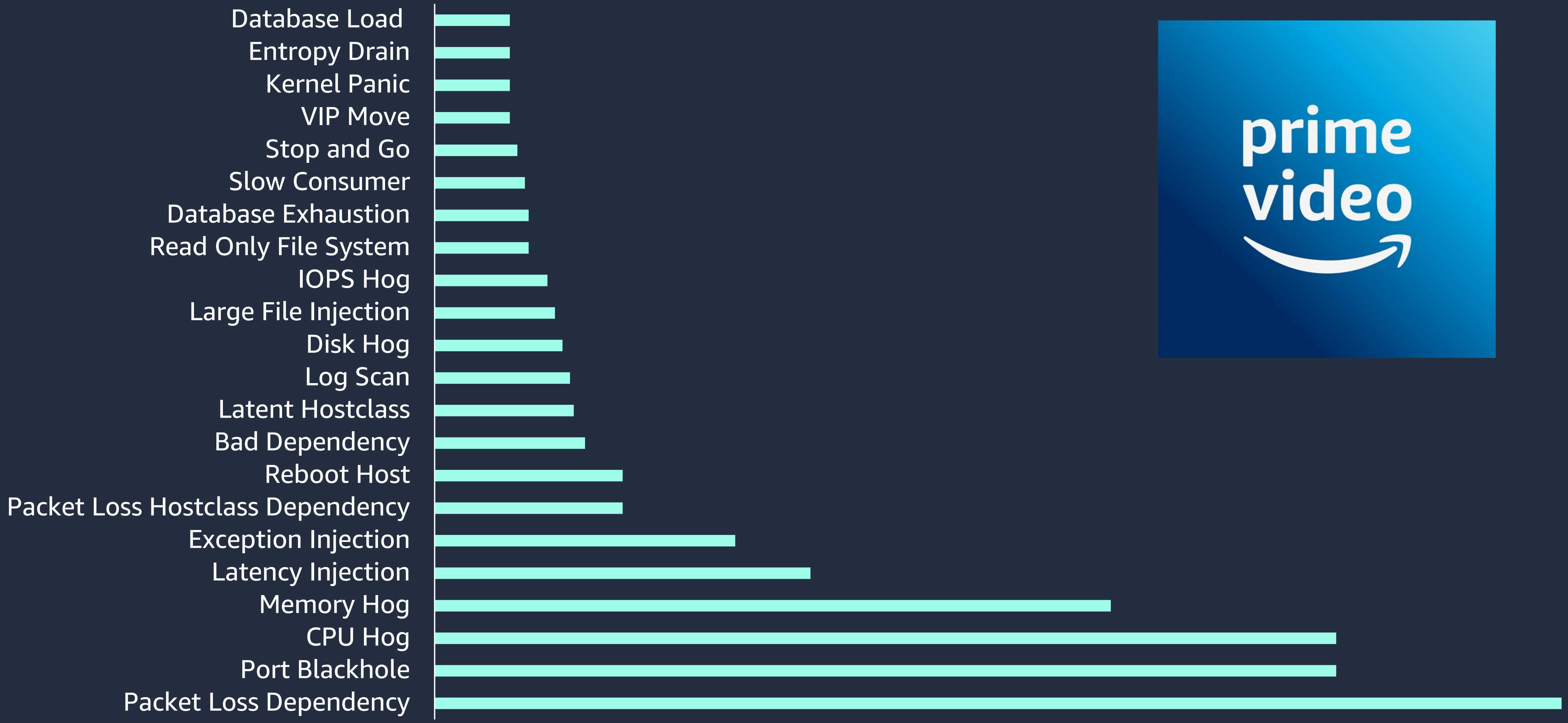
Receive alerts and conduct readiness checks

Recover

- AWS Backup
- AWS Elastic Disaster Recovery
- CloudEndure Disaster Recovery
- AWS Systems Manager

Restore data and applications

Experiments used by Prime Video



Observability

What To Observe

If a tree falls in the forest and no one is around to hear it, does it make a sound?

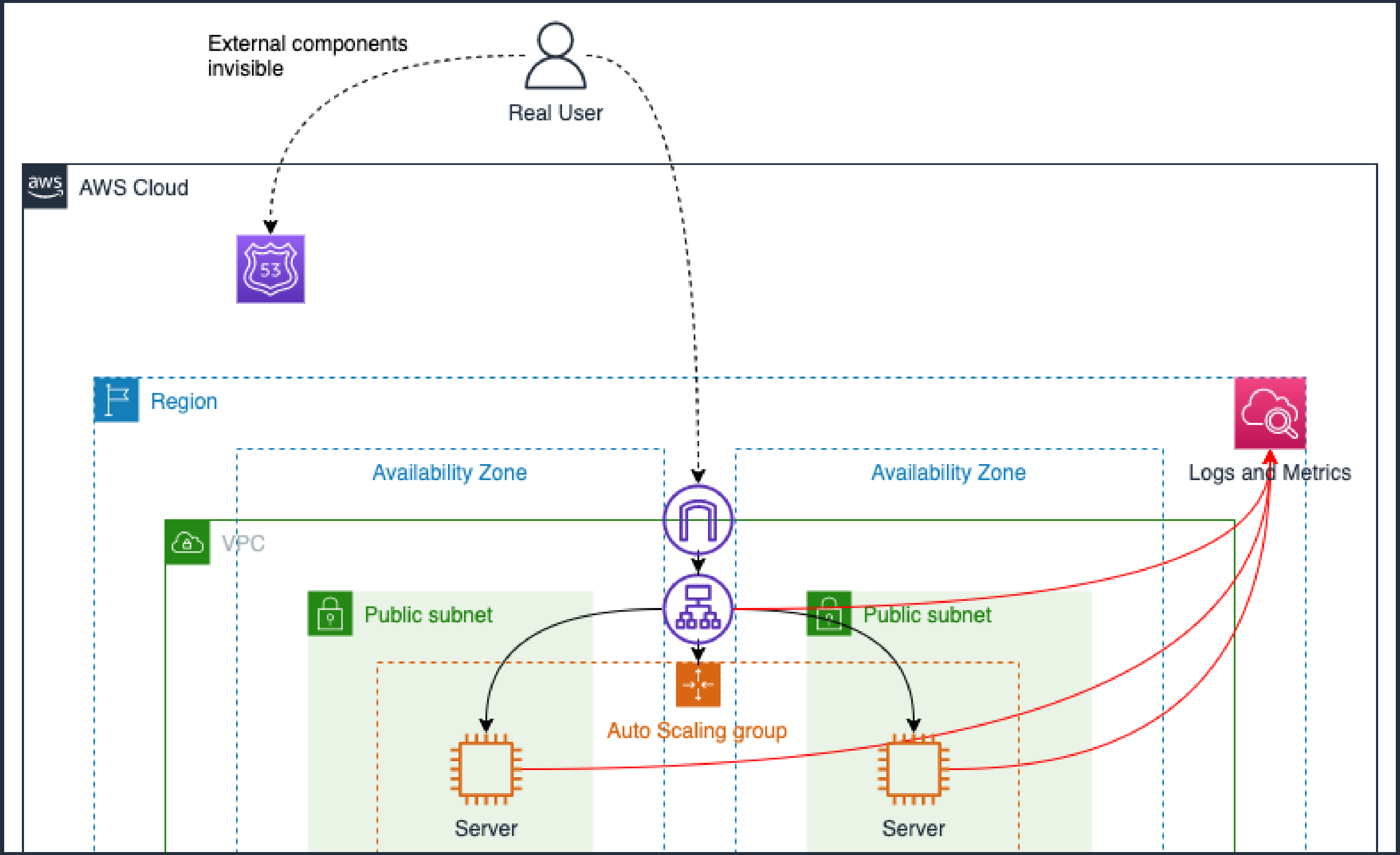
What To Observe

If part of our system is disrupted and we do not receive any irate calls from users, did anything break?

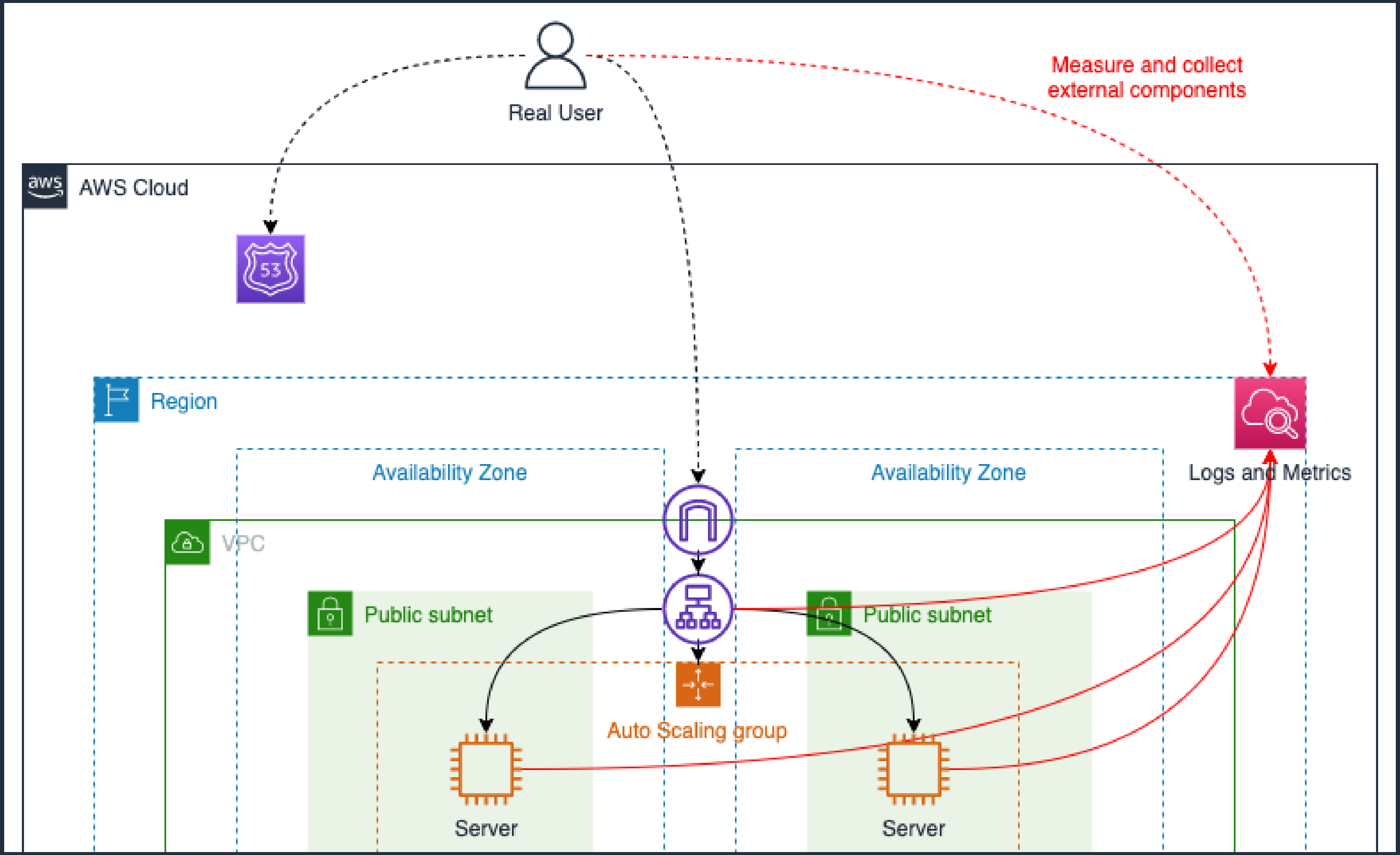
What To Observe

If part of our system is disrupted and SysOps isn’t alerted, did anything break?

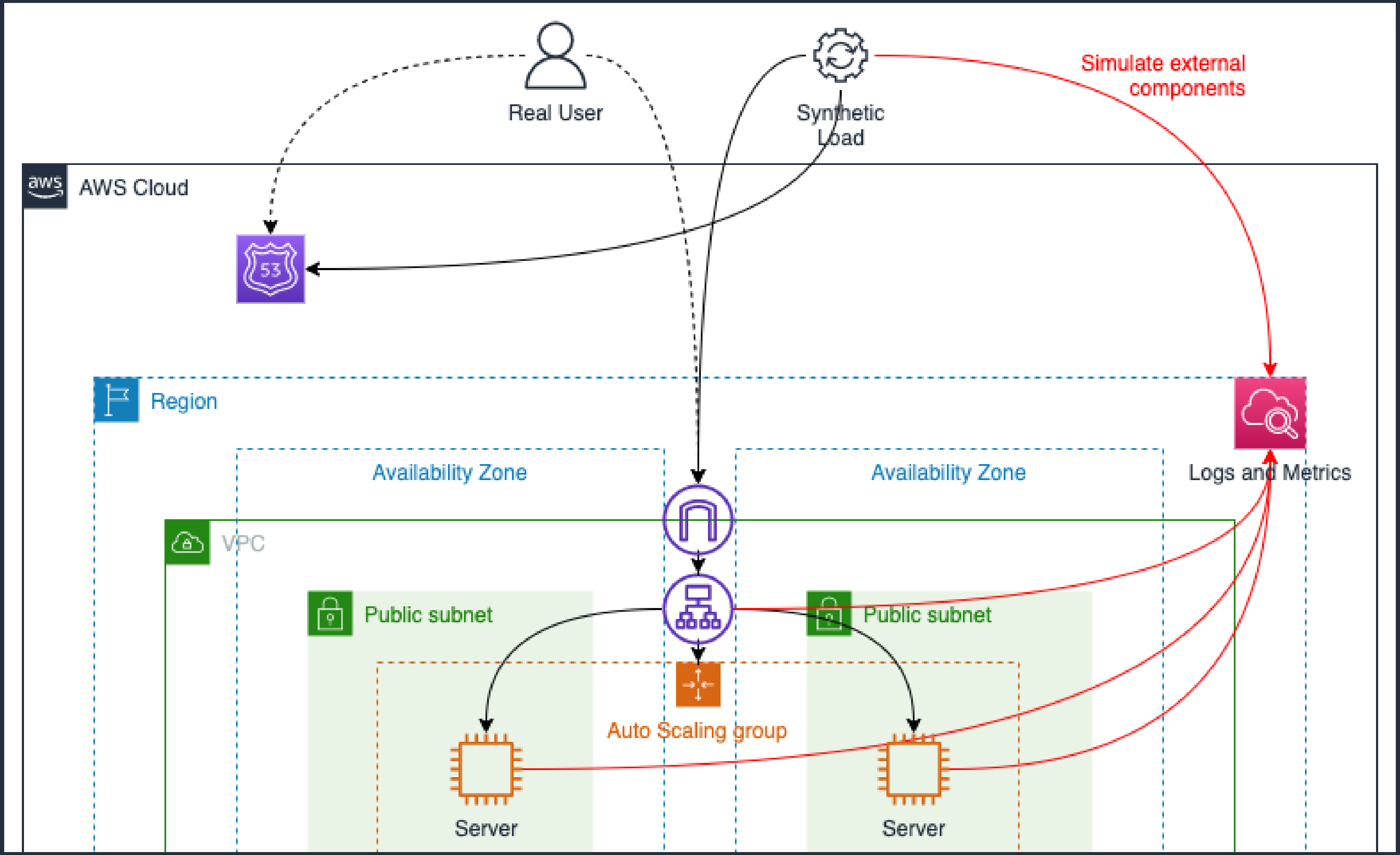
What SysOps Normally Sees



What SysOps Should See



What SysOps Should See



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 Availability Zones or Regions

AWS Solutions

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

Learn more

AWS Fault Injection Service
aws.amazon.com/fis

AWS Resilience Hub
aws.amazon.com/resilience-hub

Lab: Chaos Engineering on AWS (includes serverless)
chaos-engineering.workshop.aws

Any Day Can Be Prime Day: How Amazon.com Search Uses Chaos Engineering
bit.ly/search_chaos_engineering

Chaos Engineering in the cloud (includes link to public chaos engineering stories)
go.aws/3F2sfrF

Crisis to Confidence

Joel Ponukumatla

Ponukuma@amazon.com

Thank you !

Survey