

# Building Resilience through Chaos Engineering

Shirisha Vivekanand  
Solutions Architect  
AWS

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

**Werner Vogels**

CTO

Amazon.com



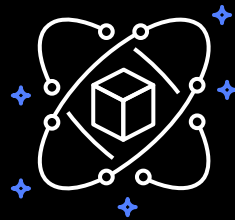
# AWS Well-Architected Framework



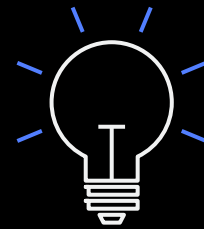
Operational  
excellence



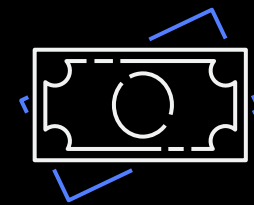
Security



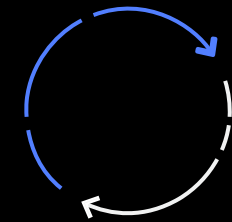
Reliability



Performance  
efficiency



Cost  
optimization



Sustainability

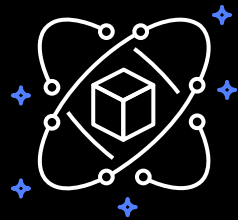
# AWS Well-Architected Framework



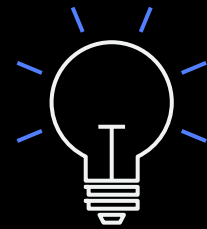
Operational  
excellence



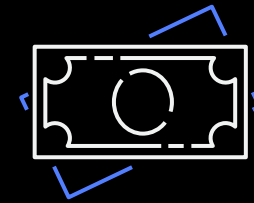
Security



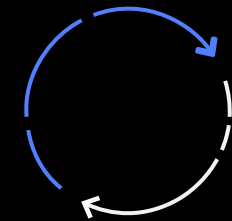
Reliability



Performance  
efficiency



Cost  
optimization



Sustainability

**“ Run tests that inject failures  
regularly into pre-production  
and production environments”**

**Reliability pillar of AWS Well-Architected Framework**

# What, Why and How?



**Chaos engineering is the process of stressing an application by creating disruptive events, observing how the system responds, and implementing improvements**

**Chaos engineering is the process of stressing an application by creating disruptive events, observing how the system responds, and implementing improvements**



**Chaos engineering is the process of stressing an application by creating disruptive events, observing how the system responds, and implementing improvements**

**Chaos engineering is the process of stressing an application by creating disruptive events, observing how the system responds, and implementing improvements**

**Chaos engineering is the process of stressing an application by creating disruptive events, observing how the system responds, and implementing improvements**

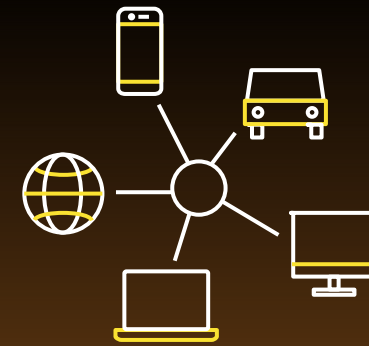
# Why Chaos Engineering?



**Increase the pace  
of innovation**

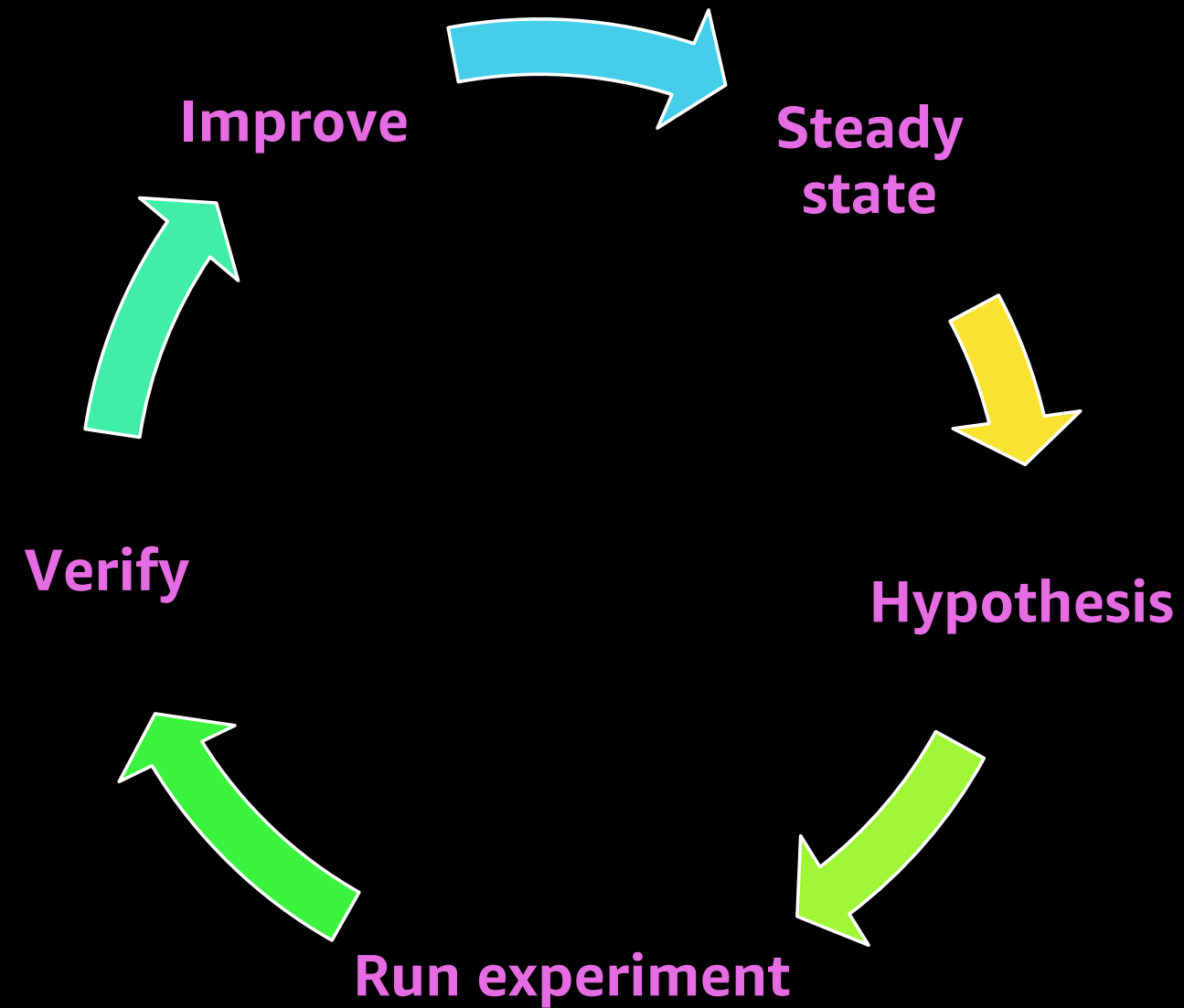


**Get more  
out of data**

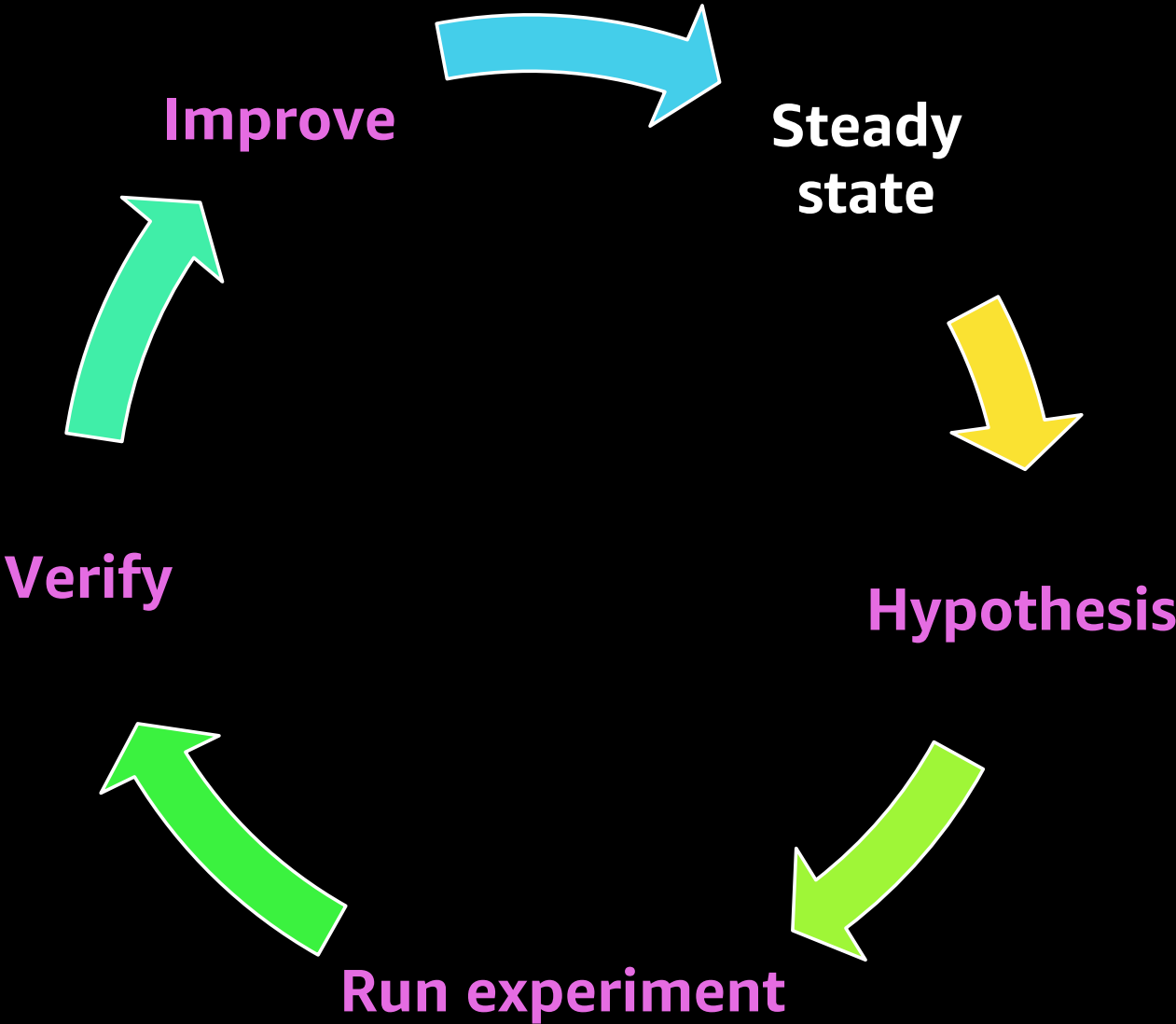


**Build new online  
customer experiences**

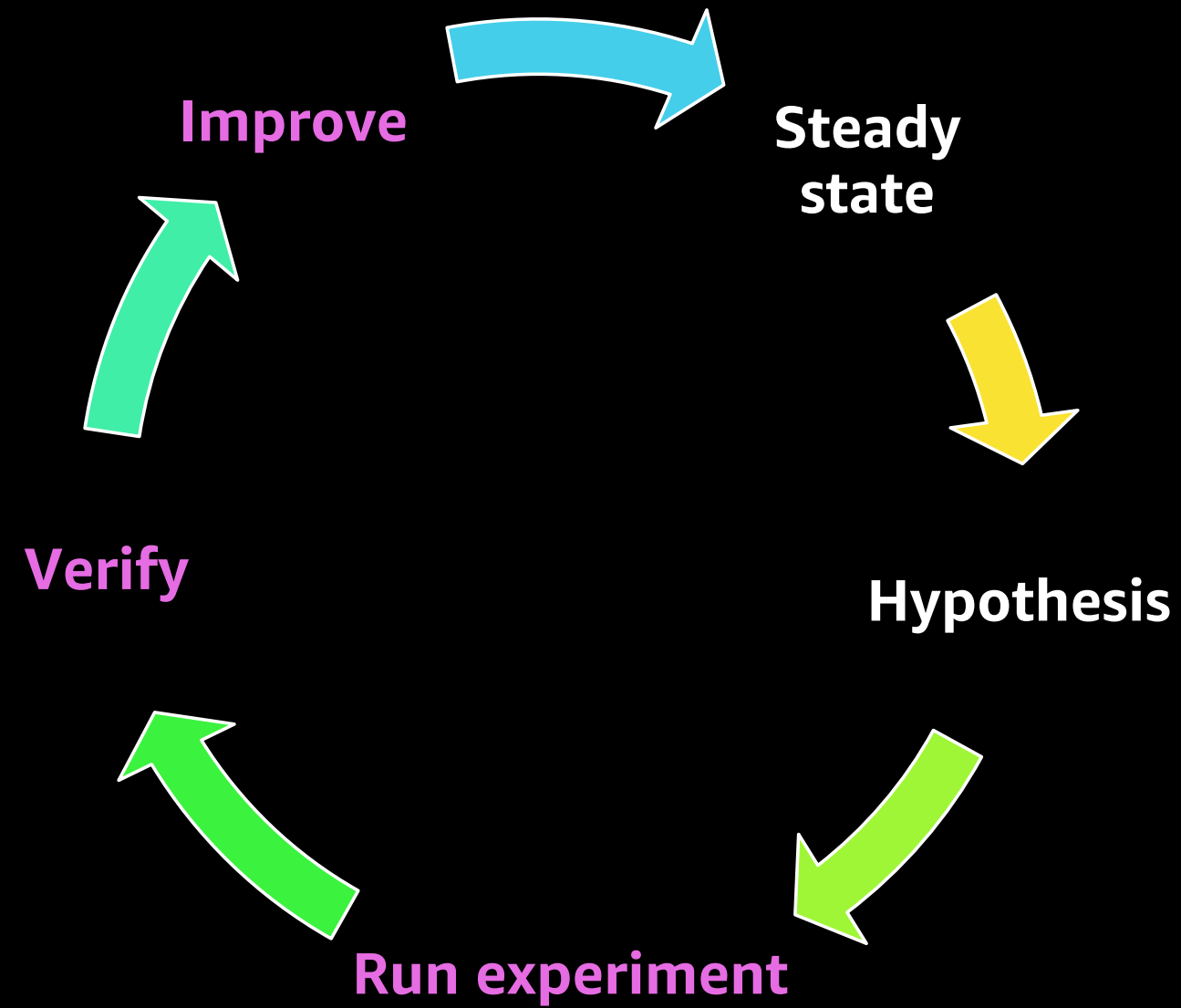
# Phases of chaos engineering



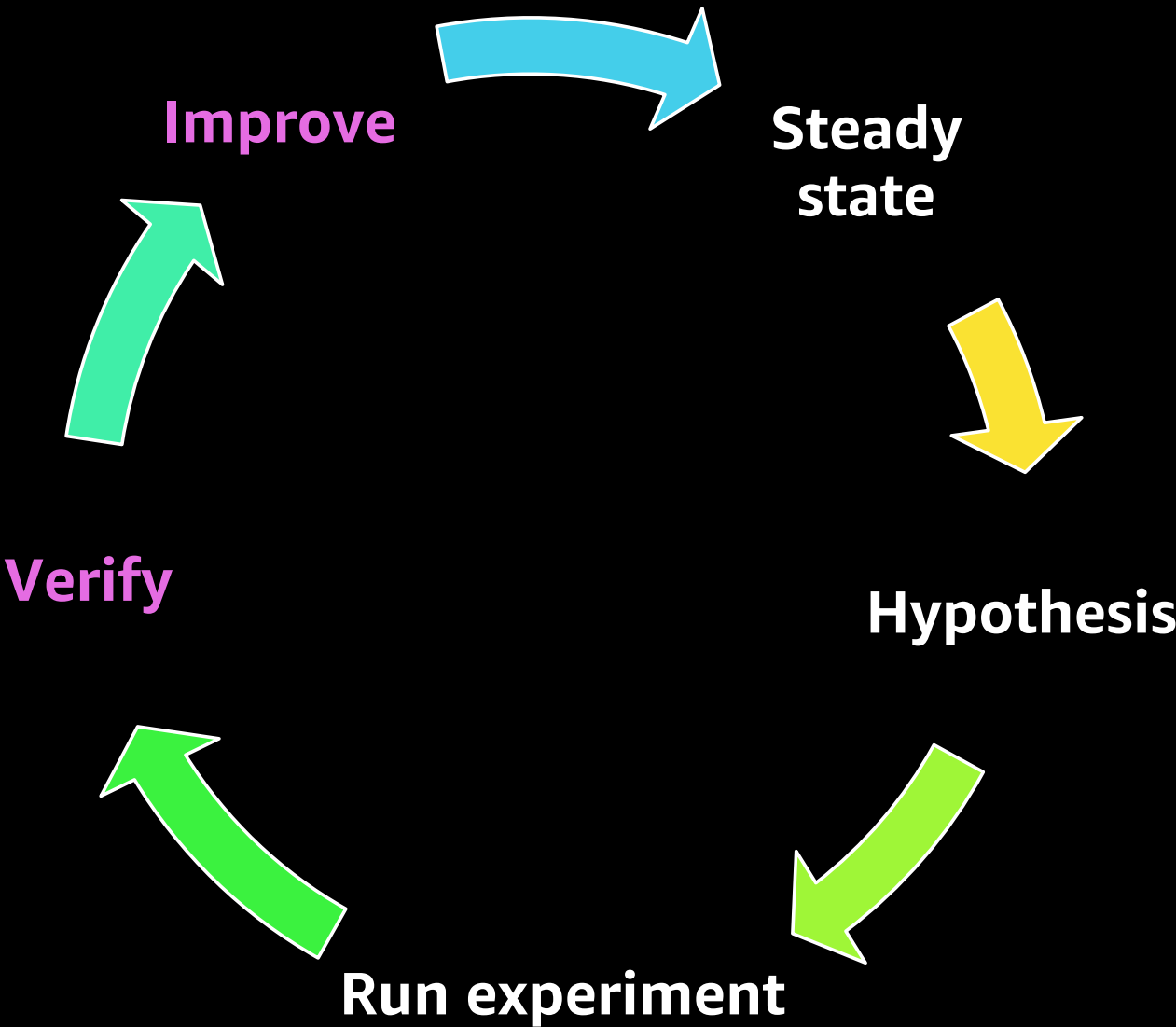
# Phases of chaos engineering



# Phases of chaos engineering

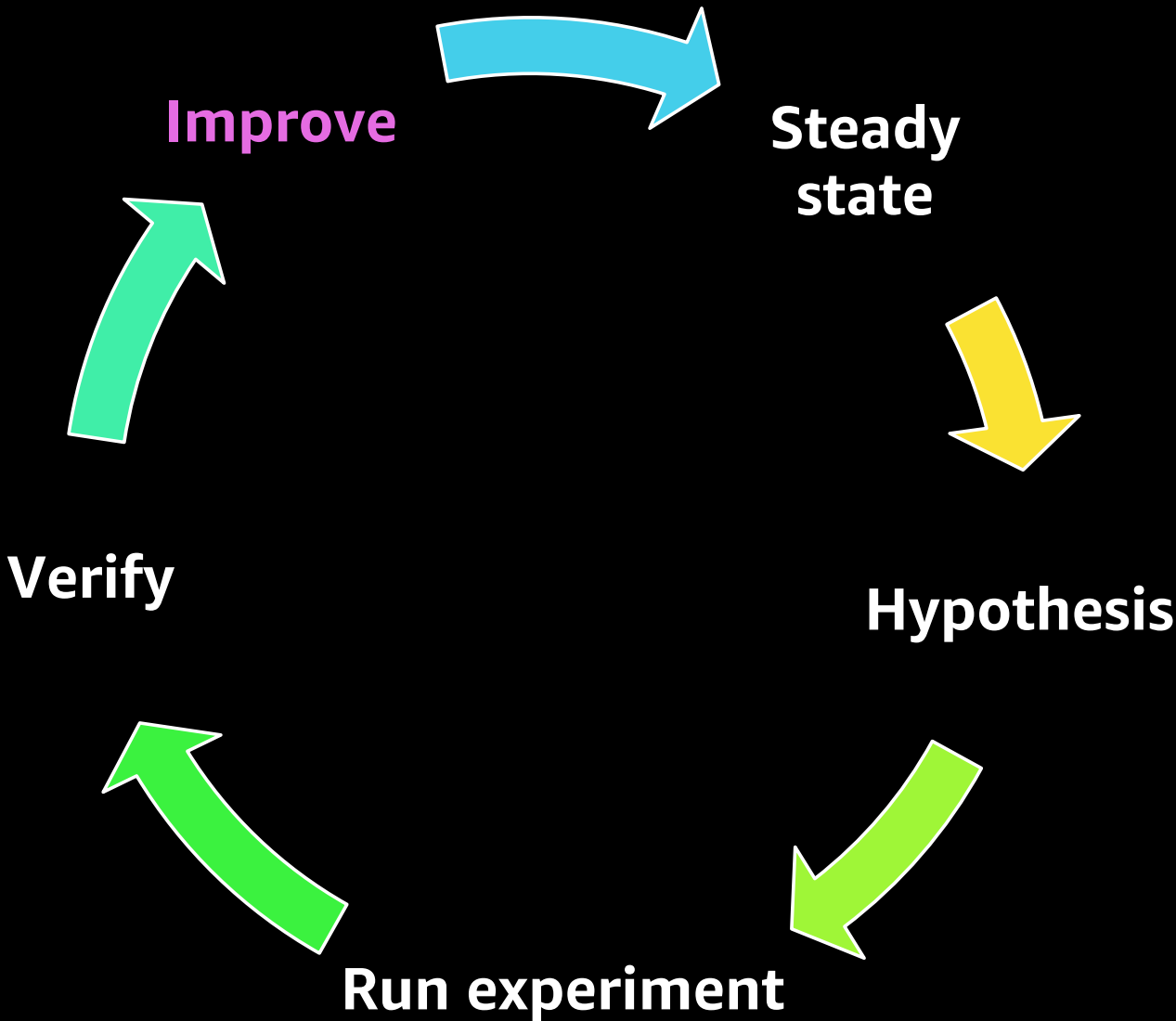


# Phases of chaos engineering

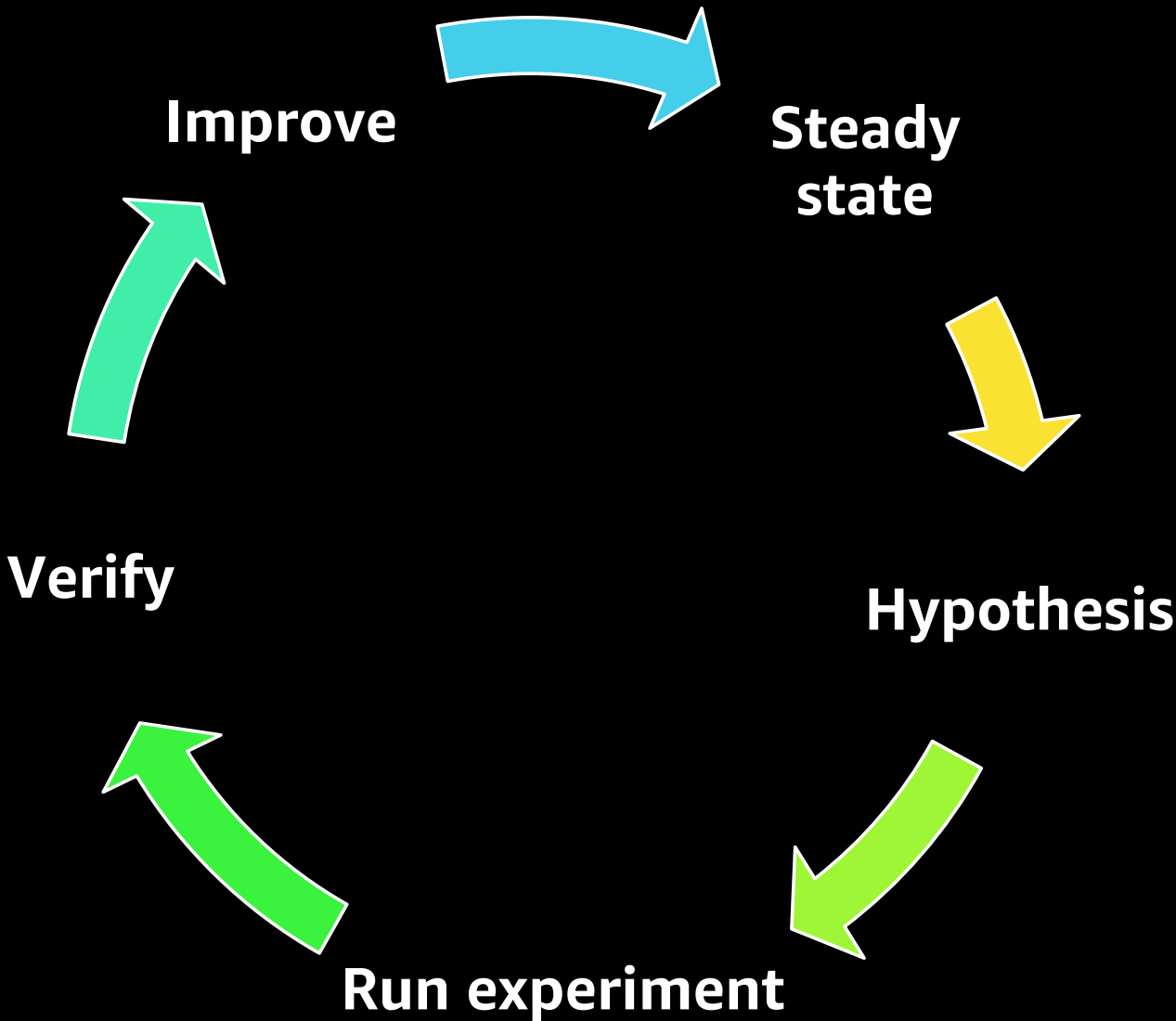




# Phases of chaos engineering



# Phases of chaos engineering



# Testing resilience

## Resilience

“The ability of a system to recover from infrastructure or service disruptions...”

## Design principles for reliability

- ⌘ Automatically recover from failure
- ⌘ Test recovery procedures

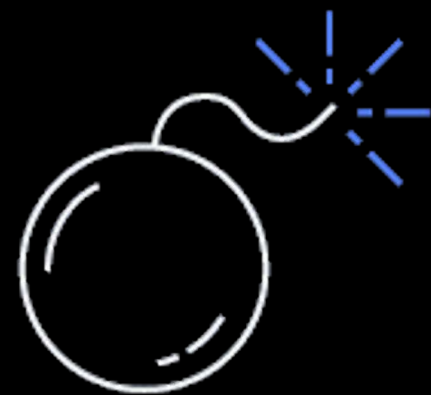
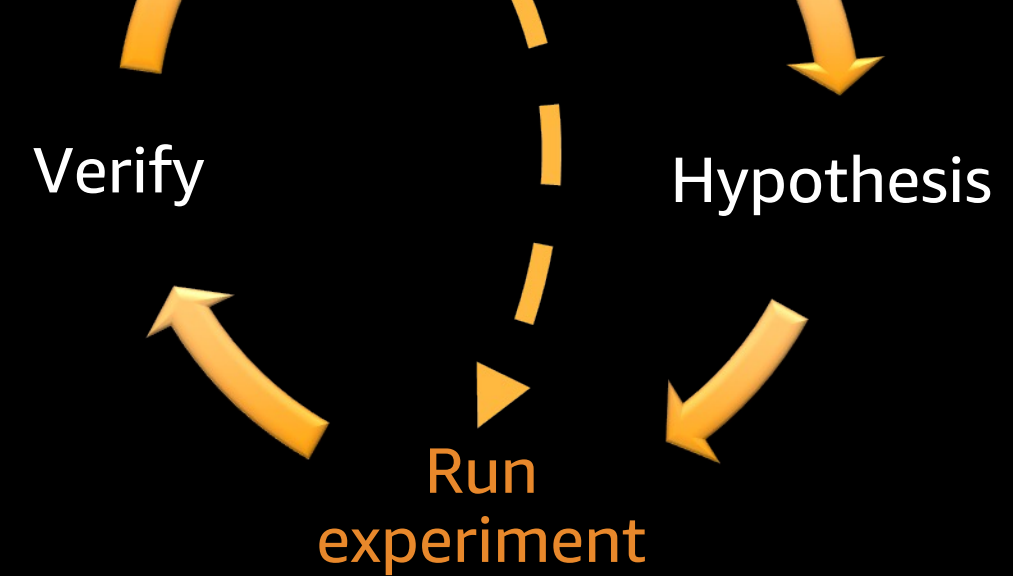


# Chaos experiment

Inject **events** that simulate

- **Hardware failures**, like servers dying
- **Software failures**, like malformed responses
- Nonfailure events, like spikes in traffic or **scaling** events

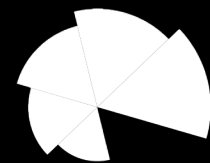
Any event capable of disrupting steady state



# Chaos tools landscape

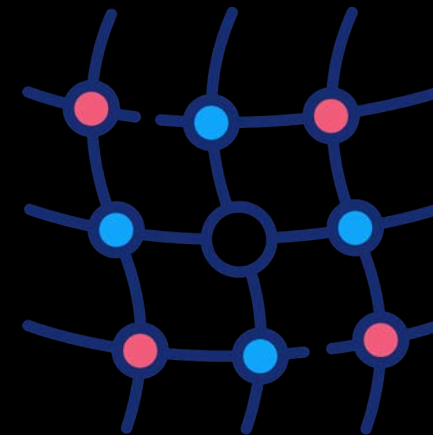


Chaos Monkey  
Simian Army  
2011



ChaosToolkit

Chaos Toolkit  
2017

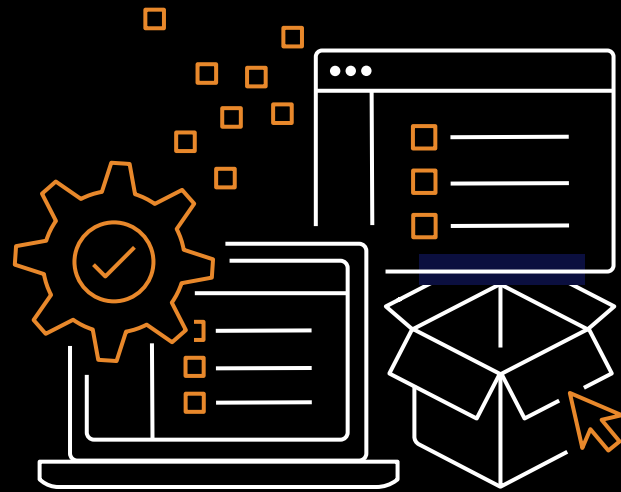


Chaos Mesh  
(Kubernetes)  
2019

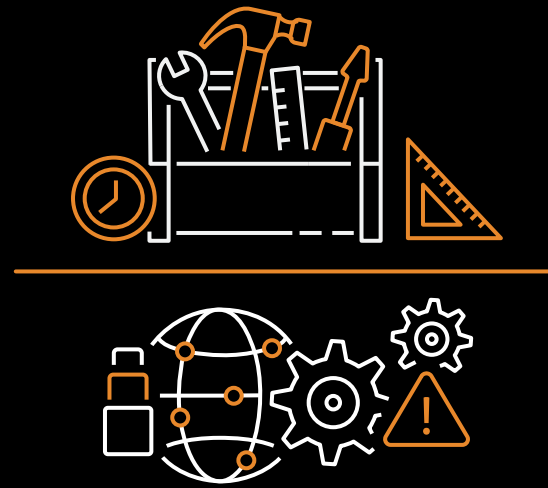


Gremlin  
2019

# AWS Fault Injection Simulator



Easy to  
get started



Real-world  
conditions



Safeguards

# Components



Actions



Targets



Experiment  
templates



Experiments

# Automatically recover from failure

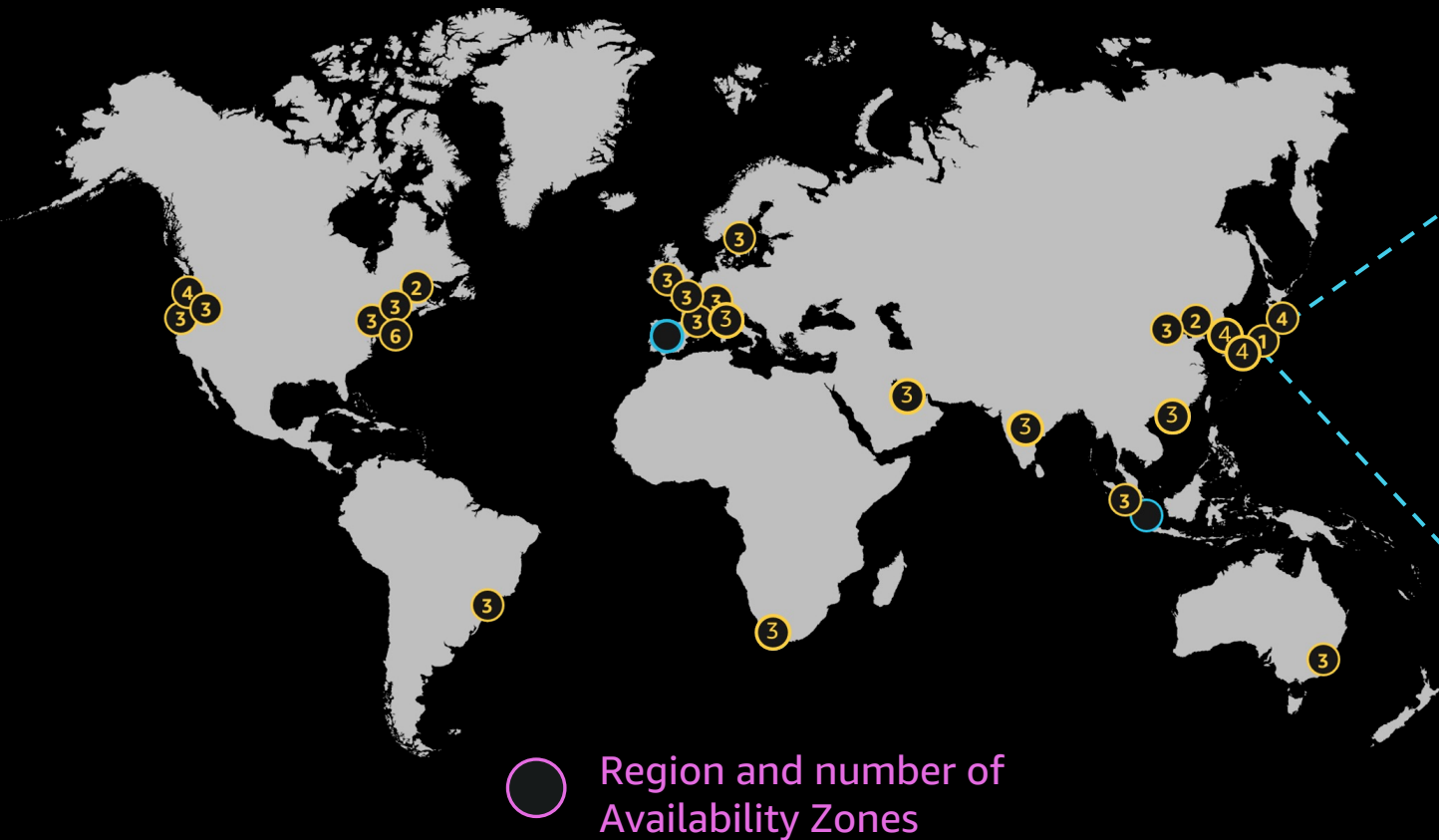




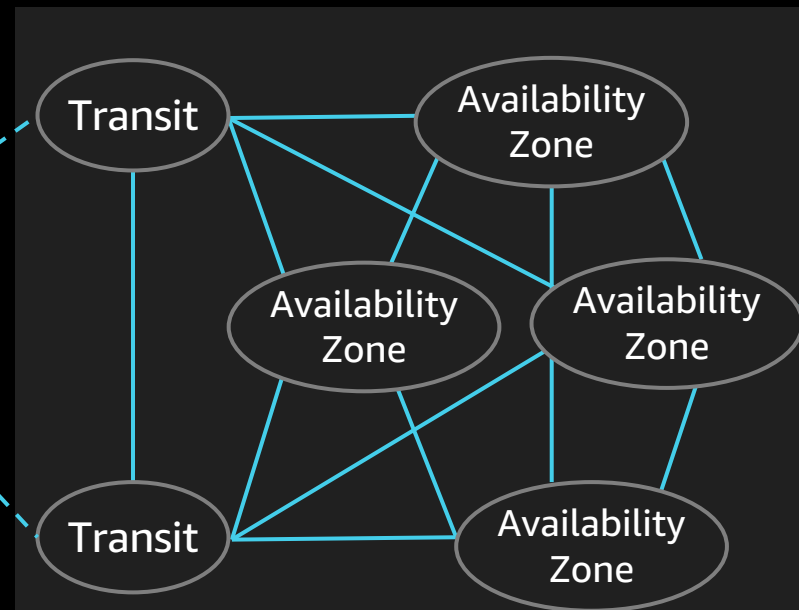
# AWS Regions and Availability Zones

AWS REGIONS ARE PHYSICAL LOCATIONS AROUND THE WORLD WHERE WE CLUSTER DATA CENTERS

24 AWS Regions worldwide

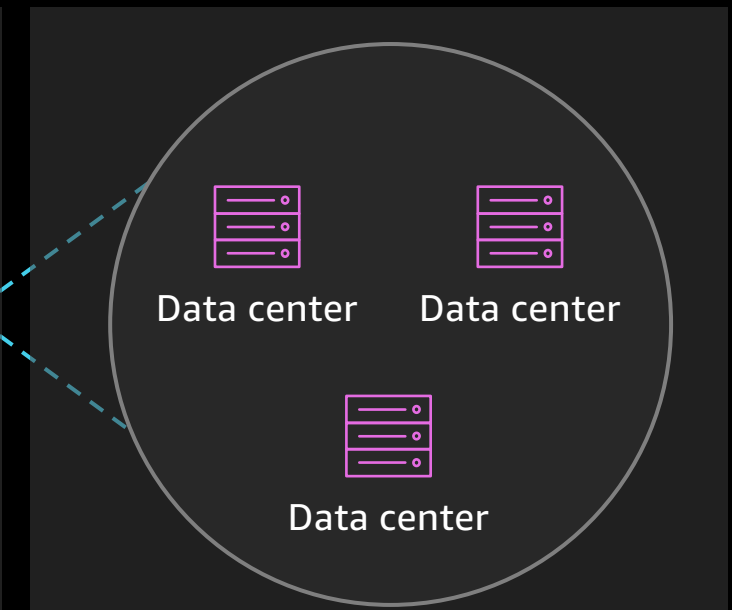


Each AWS Region has multiple Availability Zones



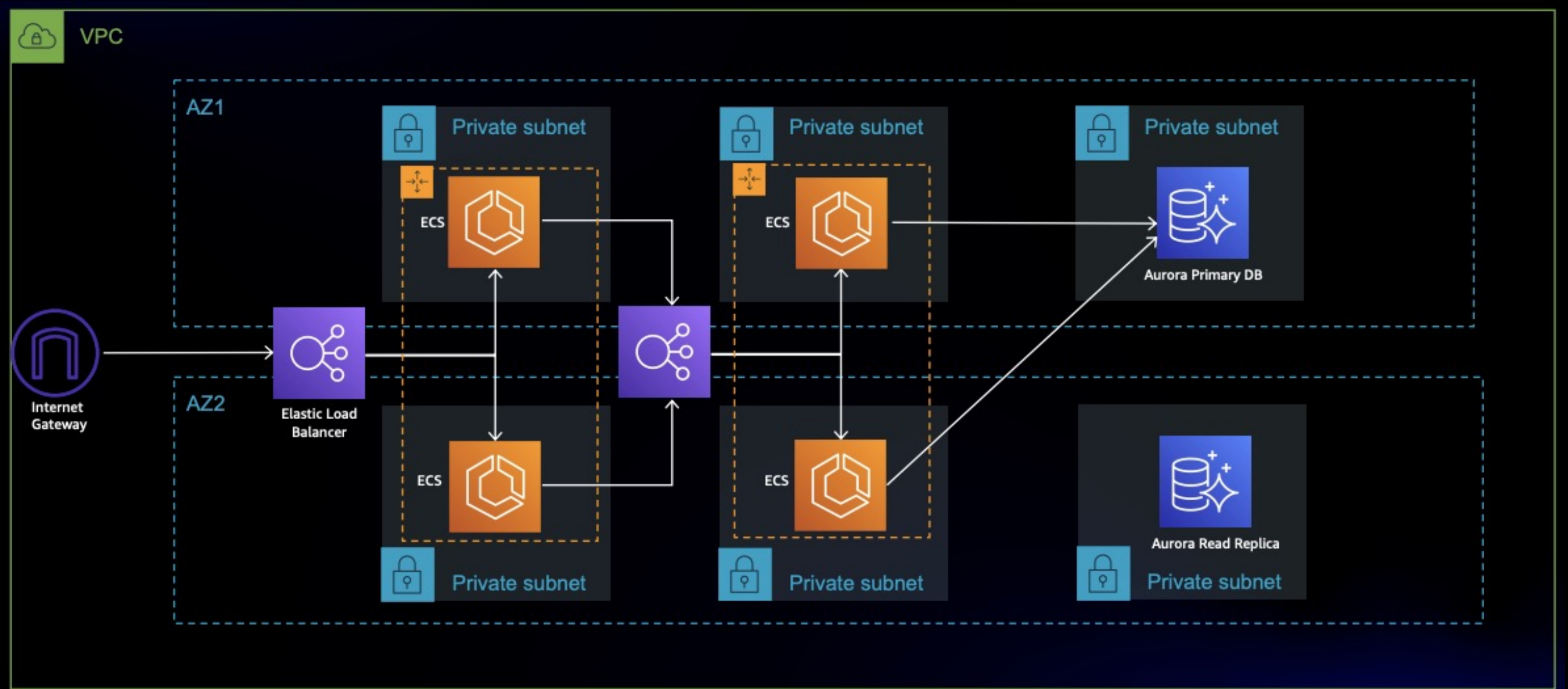
A **Region** is a physical location in the world

Each Availability Zone is one or more discrete data centers



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

# ECS based three tier Multi-AZ architecture



## AWS FIS

## AWS FIS &gt; Experiment templates

## Experiment templates

Experiments

Experiment templates (8) [Info](#)

Actions ▾

[Create experiment template](#)

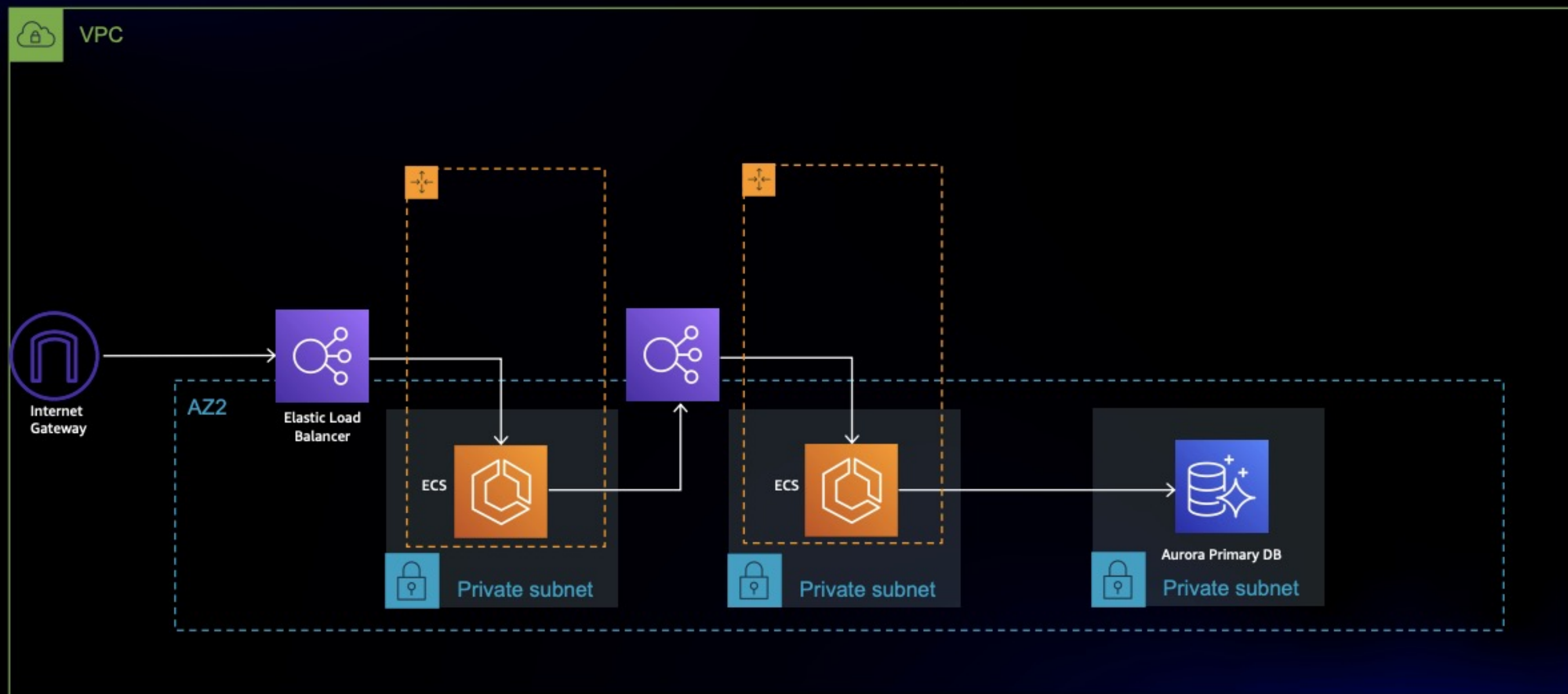
Filter experiment templates

&lt; 1 &gt; ⚙️

	Name ▾	Experiment template ID ▾	Description ▾	Creation time ▾	Last update time ▾
<input type="radio"/>	Az2FailureTemplate	<a href="#">EXT2cRAyzfRru4Fz</a>	Replicate AZ: us-east-1b outag...	November 23, 2021, 12:25:12 (UTC-05:00)	November 23, 2021, 12:25:12 (UTC-05:00)
<input type="radio"/>	MemoryStress	<a href="#">EXT3NXbUgkiragjc</a>	Memory stress via SSM	November 23, 2021, 12:25:06 (UTC-05:00)	November 23, 2021, 12:25:06 (UTC-05:00)
<input type="radio"/>	CPUStress	<a href="#">EXT3TXdG5d8rvmqz</a>	CPU stress via SSM	November 23, 2021, 12:25:06 (UTC-05:00)	November 23, 2021, 12:25:06 (UTC-05:00)
<input type="radio"/>	TerminateEc2	<a href="#">EXT6Sb3xmwXZtYY</a>	Terminate EC2 instance	November 23, 2021, 12:24:50 (UTC-05:00)	November 23, 2021, 12:24:50 (UTC-05:00)
<input type="radio"/>	RdsClusterFailover	<a href="#">EXT85GSXDboxp3tMn</a>	Db failover to other AZ	November 23, 2021, 12:25:06 (UTC-05:00)	November 23, 2021, 12:25:06 (UTC-05:00)
<input type="radio"/>	EcsStopTaskTemplate	<a href="#">EXT8kUECEH3GheH84</a>	Stop a random task in cluster: ...	November 23, 2021, 12:25:12 (UTC-05:00)	November 23, 2021, 12:25:12 (UTC-05:00)
<input type="radio"/>	Az1FailureTemplate	<a href="#">EXTBkj8WBeZaWDgCx</a>	Replicate AZ: us-east-1a outag...	November 23, 2021, 12:25:12 (UTC-05:00)	November 23, 2021, 12:25:12 (UTC-05:00)
<input type="radio"/>	NetworkStress	<a href="#">EXTCwMJrIUywPm37W</a>	Network latency stress via SSM	November 23, 2021, 12:25:05 (UTC-05:00)	November 23, 2021, 12:25:05 (UTC-05:00)



# Simulate AZ failure and auto-recovery



# Making it real



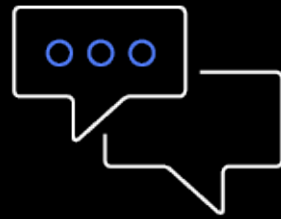
# Game days

SIMULATE FAILURE OR EVENT TO TEST SYSTEMS, PROCESSES, AND TEAM RESPONSES



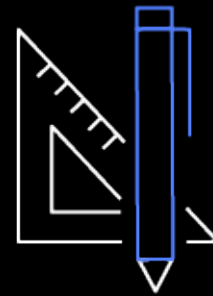
## People

Cross-discipline  
team



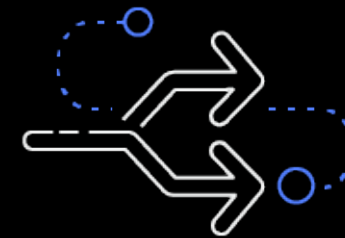
## Briefing

Overview  
Roles



## Planning

Scenario  
Events  
Preparation



## Execution

In production  
Capture  
feedback

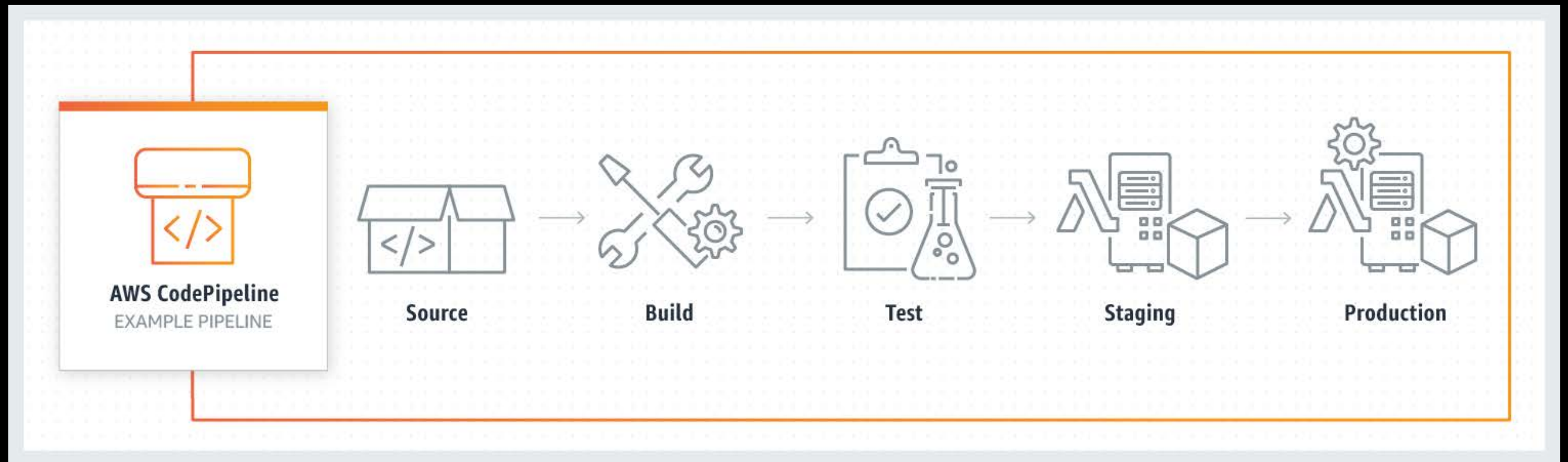


## Analysis

What happened  
Follow-up items

# Run these tests regularly

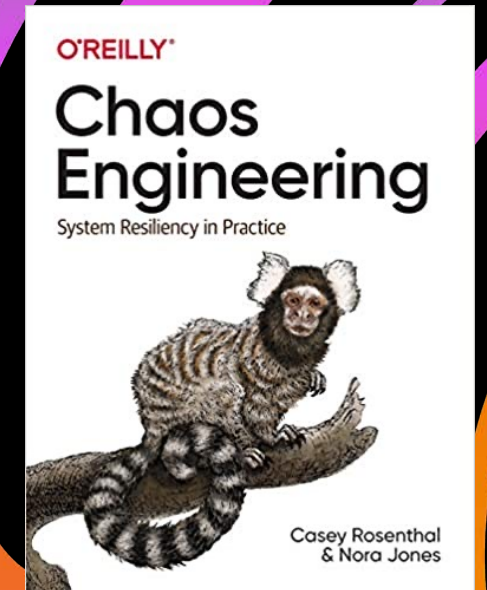
- As part of CI / CD pipeline
- Start with staging (test, pre-production) environments
- Consider which tests to run in production



**“This isn’t about creating chaos. It is about making the chaos inherent in the system visible.”**

**Casey Rosenthal and Nora Jones**

*Chaos Engineering: System Resiliency in Practice*





# Next steps

You *can* do this

You *should* do this

## REL 12. How do you test reliability? [Info](#)

After you have designed your workload to be resilient to the stresses of production, testing is the only way to ensure that it will operate as designed, and deliver the resiliency you expect.

Question does not apply to this workload [Info](#)

Select from the following

- Use playbooks to investigate failures [Info](#)
- Perform post-incident analysis [Info](#)
- Test functional requirements [Info](#)
- Test scaling and performance requirements [Info](#)
- Test resiliency using chaos engineering [Info](#)
- Conduct game days regularly [Info](#)

# AWS Account

## How to create an AWS Account?

The AWS free tier create account page

[\*https://go.aws/3HCmBxm\*](https://go.aws/3HCmBxm)

# Architecture resources

## AWS Well-Architected Framework

The official best practices for Architecting in the AWS Cloud

[aws.com/well-architected](https://aws.com/well-architected) (reliability pillar: [bit.ly/reliability-pillar](https://bit.ly/reliability-pillar))

## AWS Well-Architected Labs

Hands-on labs to help you learn, measure, and build using architectural best practices

[wellarchitectedlabs.com](https://wellarchitectedlabs.com) (reliability labs: [wellarchitectedlabs.com/reliability](https://wellarchitectedlabs.com/reliability))

## AWS Architecture Center

Official AWS repository for all Architecture resources

[aws.amazon.com/architecture](https://aws.amazon.com/architecture)

## AWS Solutions Library

Vetted reference implementations and well-architected patterns

[aws.amazon.com/solutions](https://aws.amazon.com/solutions)



**AWS Well-Architected**

# Workshop resources

## Chaos Engineering Workshop

Hands on workshop on chaos engineering - EC2, EKS, Redis, RDS

<https://bit.ly/40nrva1>

## Testing Resiliency in Financial Services using Chaos Engineering

Hands on workshop on stress testing CPU and Network on ECS

<https://bit.ly/3Hs3qpZ>

## Chaos Engineering on AWS

Hands on workshop on chaos engineering with advanced experiments

<https://bit.ly/3Rq2sPC>

# Thank you!

Shirisha Vivekanand

Solutions Architect  
AWS

