
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

BOA324

# Resilient and well-architected applications with chaos engineering


Gunnar Grosch (he/him)

Sr. Developer Advocate
Amazon Web Services

Sara Gerion (she/her)

Sr. Solutions Architect
Amazon Web Services



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Agenda

AWS Well-Architected

Chaos engineering

AWS Fault Injection Simulator

Scenario

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Well-Architected

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

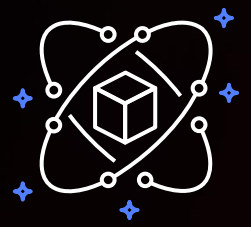
# Pillars of AWS Well-Architected



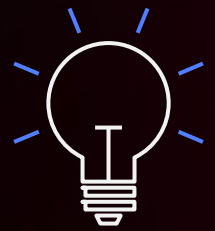
Operational excellence



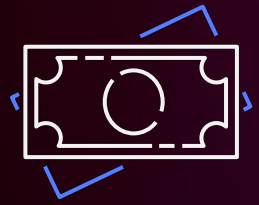
Security



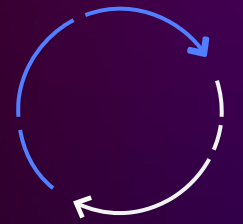
Reliability



Performance efficiency



Cost optimization



Sustainability

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

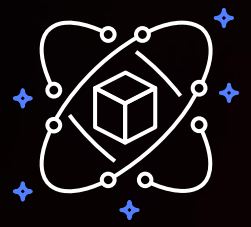
# Pillars of AWS Well-Architected



Operational excellence



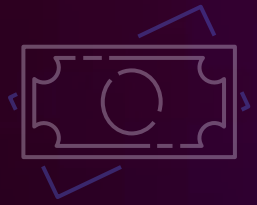
Security



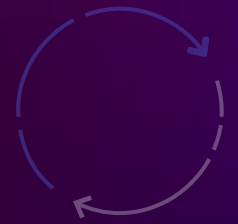
Reliability



Performance efficiency



Cost optimization



Sustainability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

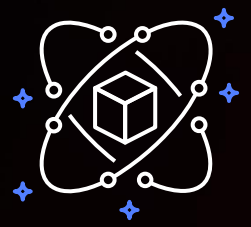
# Pillars of AWS Well-Architected



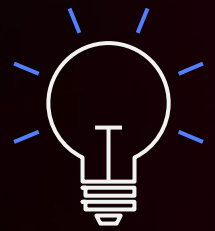
Operational excellence



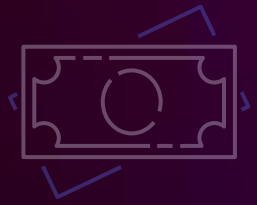
Security



Reliability



Performance efficiency



Cost optimization



Sustainability

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

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

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Chaos engineering

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

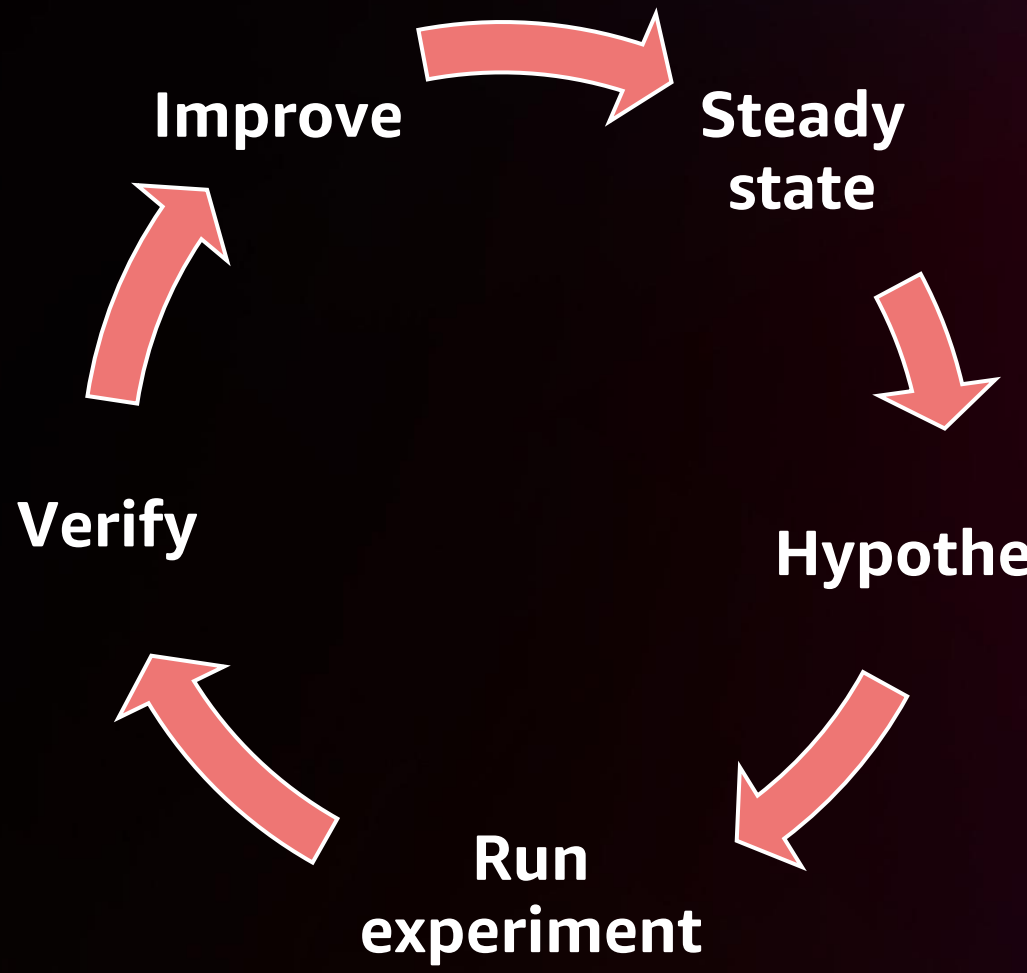
"Chaos engineering is the process of stressing an application by creating disruptive events, observing how the system responds, and implementing improvements"

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

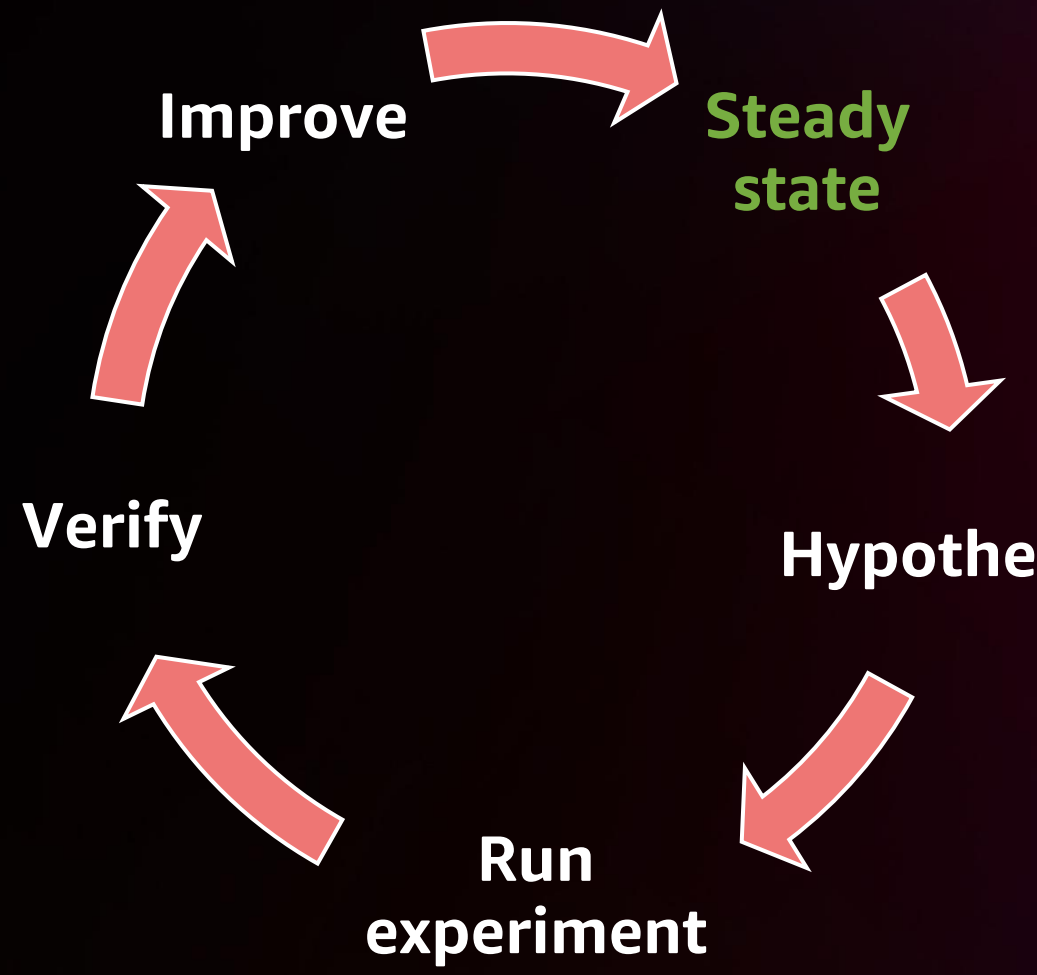
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phases of chaos engineering



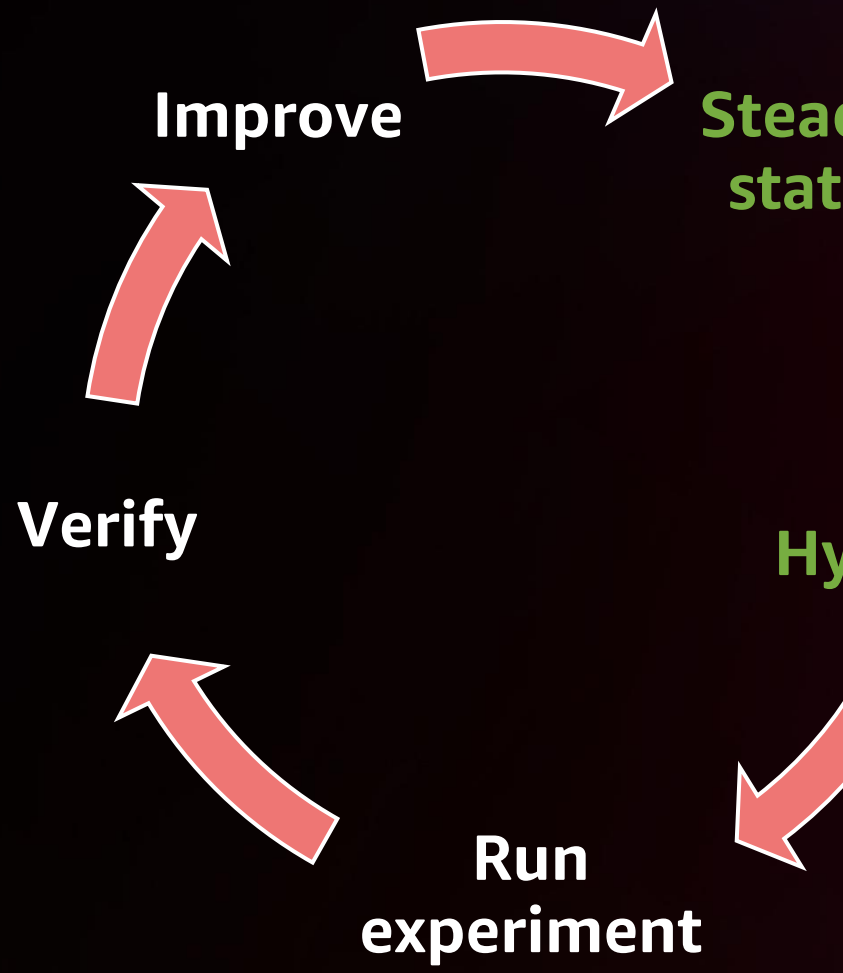

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phases of chaos engineering



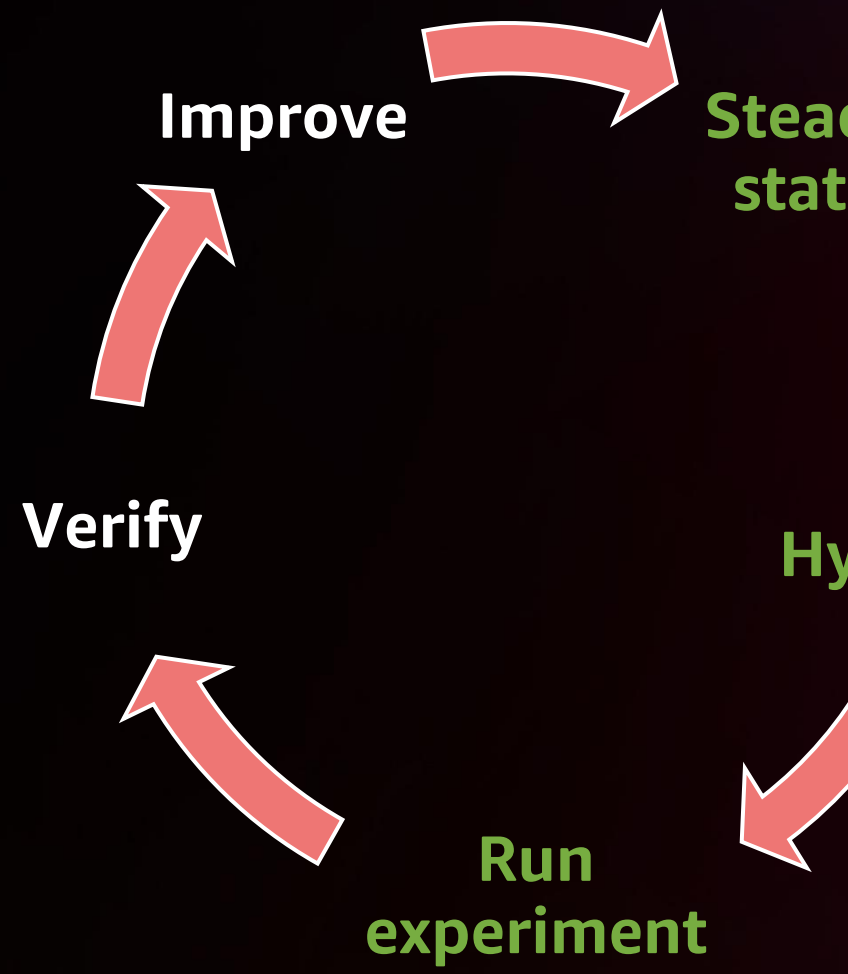

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phases of chaos engineering



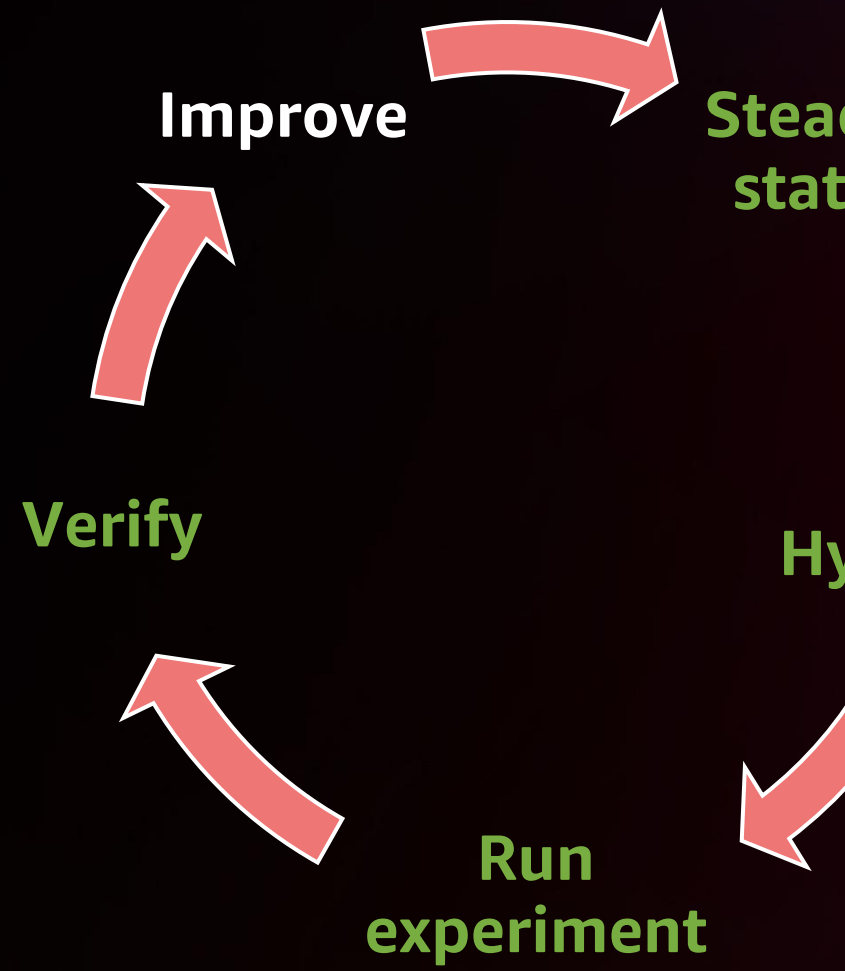

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phases of chaos engineering



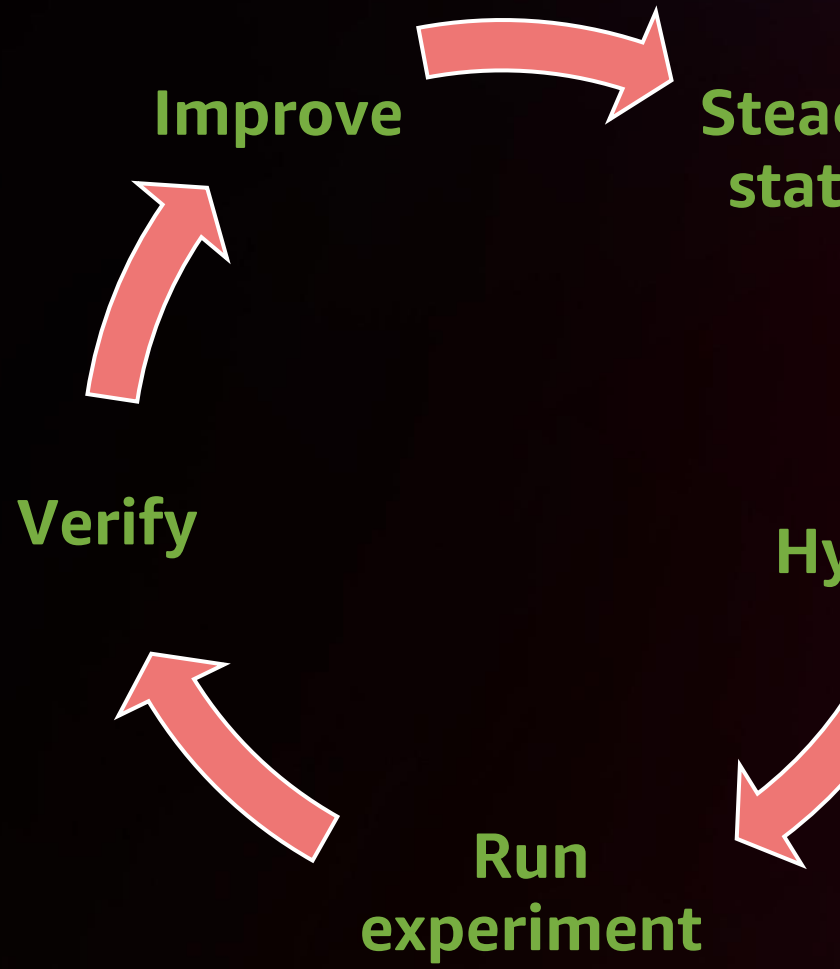

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phases of chaos engineering




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phases of chaos engineering




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

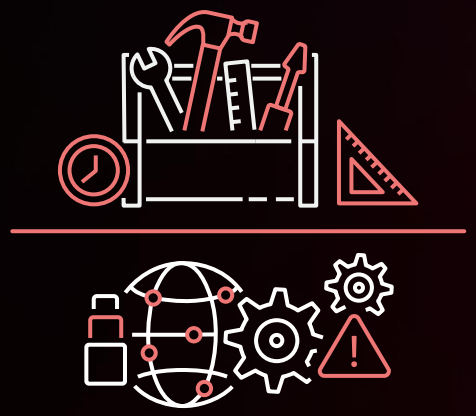
# AWS Fault Injection Simulator

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# AWS Fault Injection Simulator



Easy to
get started



Real-world
conditions



Safeguards

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Components



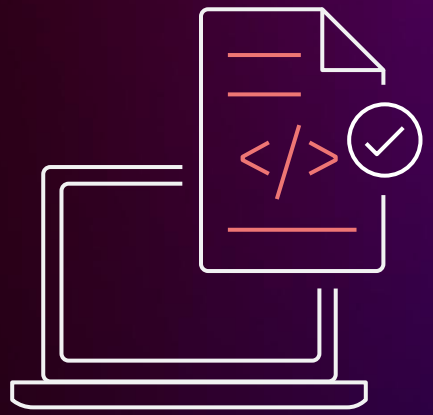
Actions



Targets



Experiment templates



Experiments

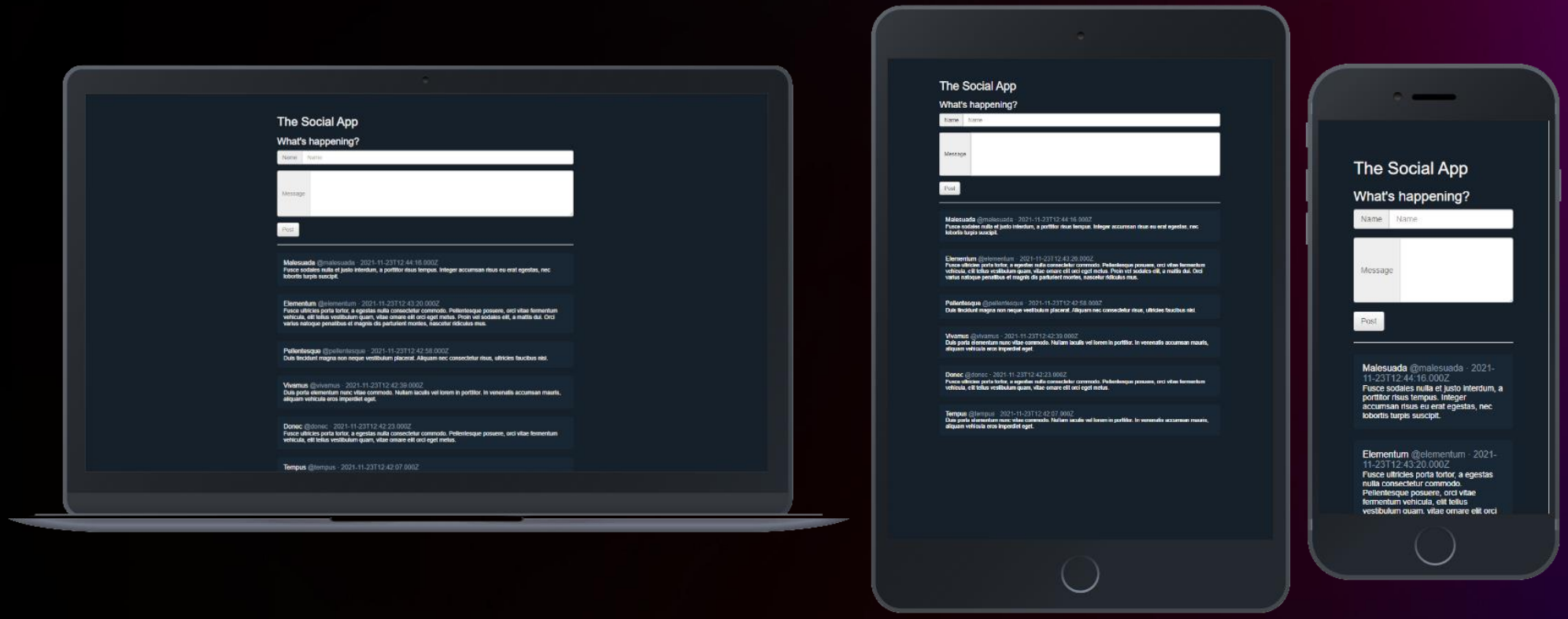
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Scenario

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Scenario




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# The Social App

## What's happening?

Name | Name

Message

Post

---

**Joyce** @joyce
I may struggle with geography, but I'm sure I'm somewhere around here.

**Reyes** @reyes
He uses onomatopoeia as a weapon of mental destruction.

**Lazaro** @lazaro
I'd rather be a bird than a fish.

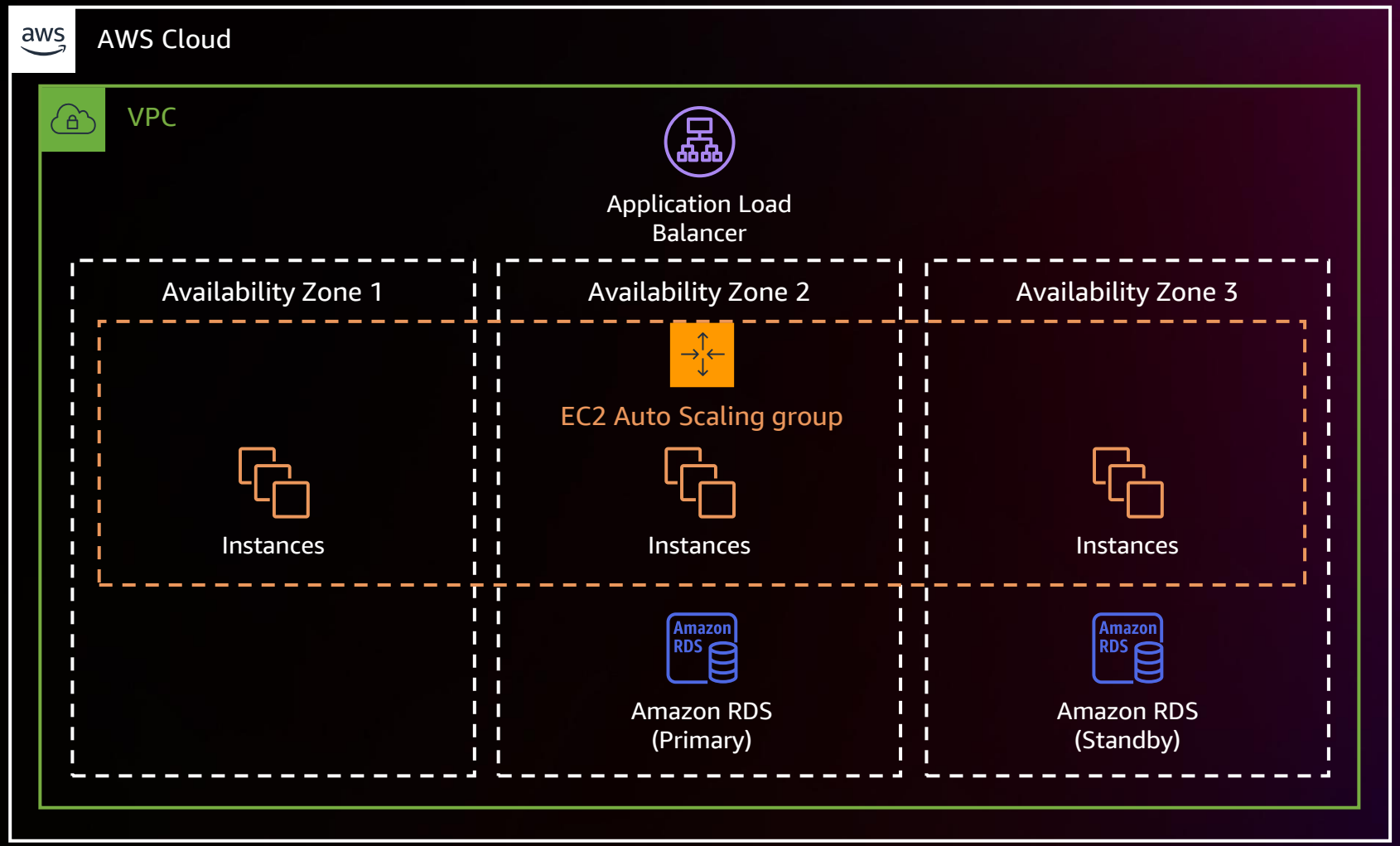
**Carly** @carly
The light that burns twice as bright burns half as long.

**Russo** @russo

# Instance-based architecture

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Instance-based architecture

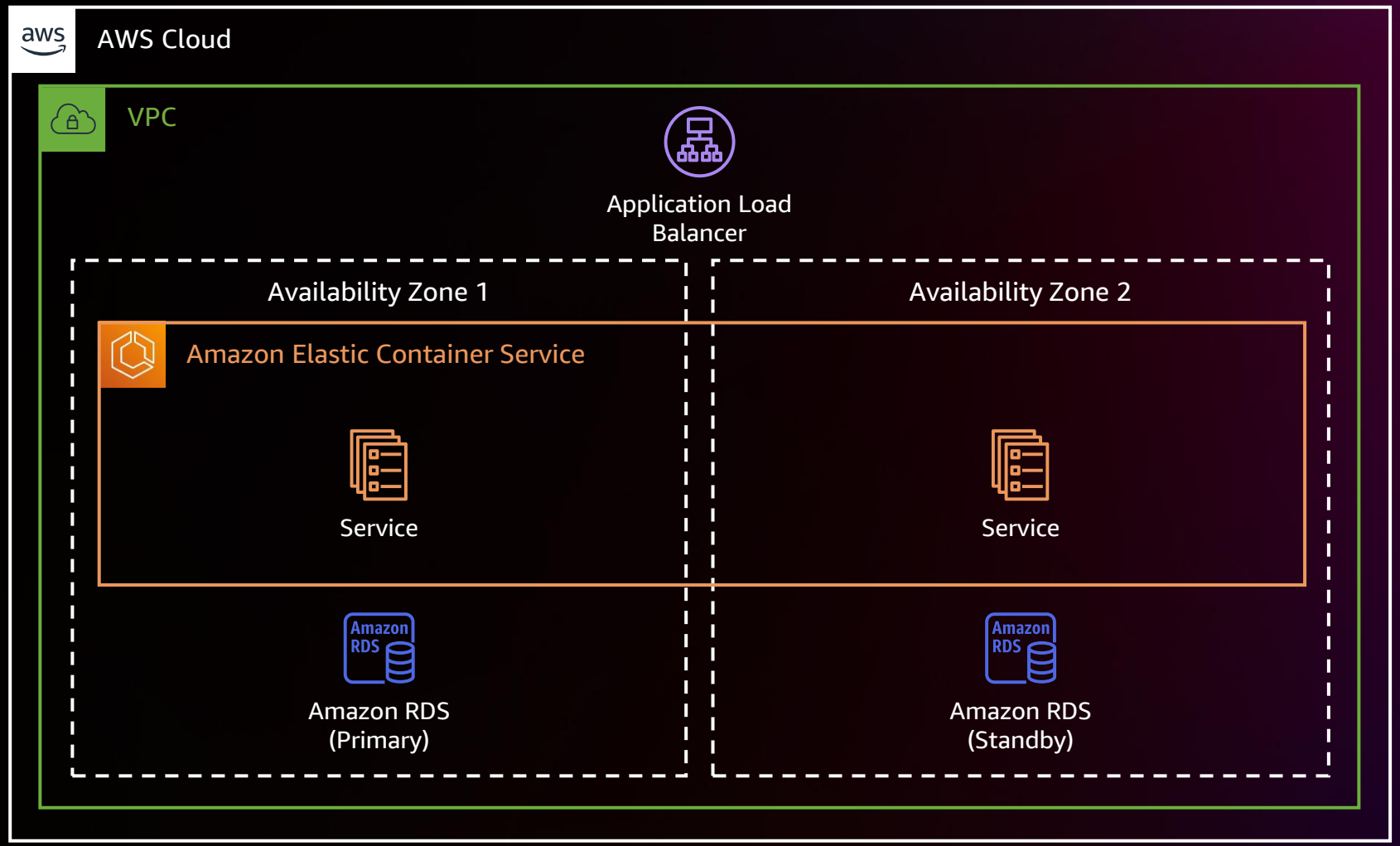



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Container architecture

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Container architecture

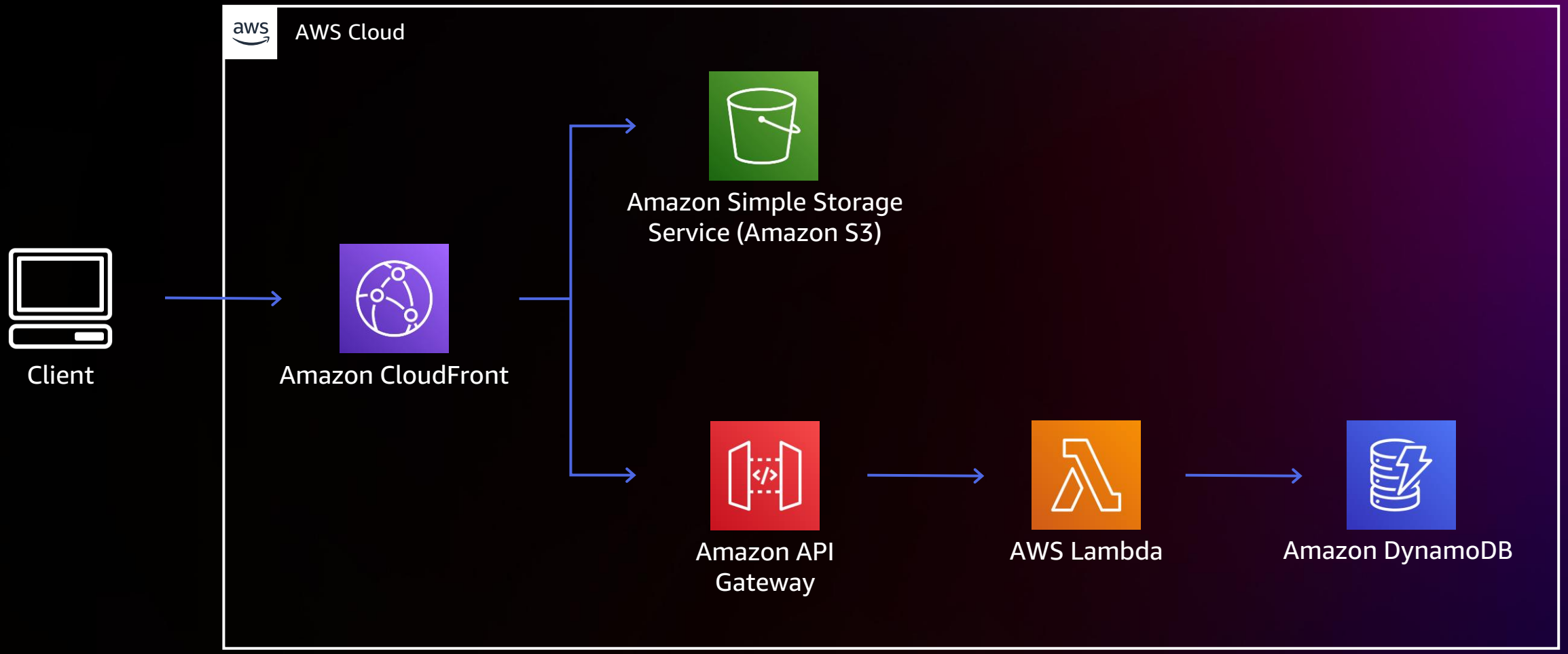



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Serverless architecture

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

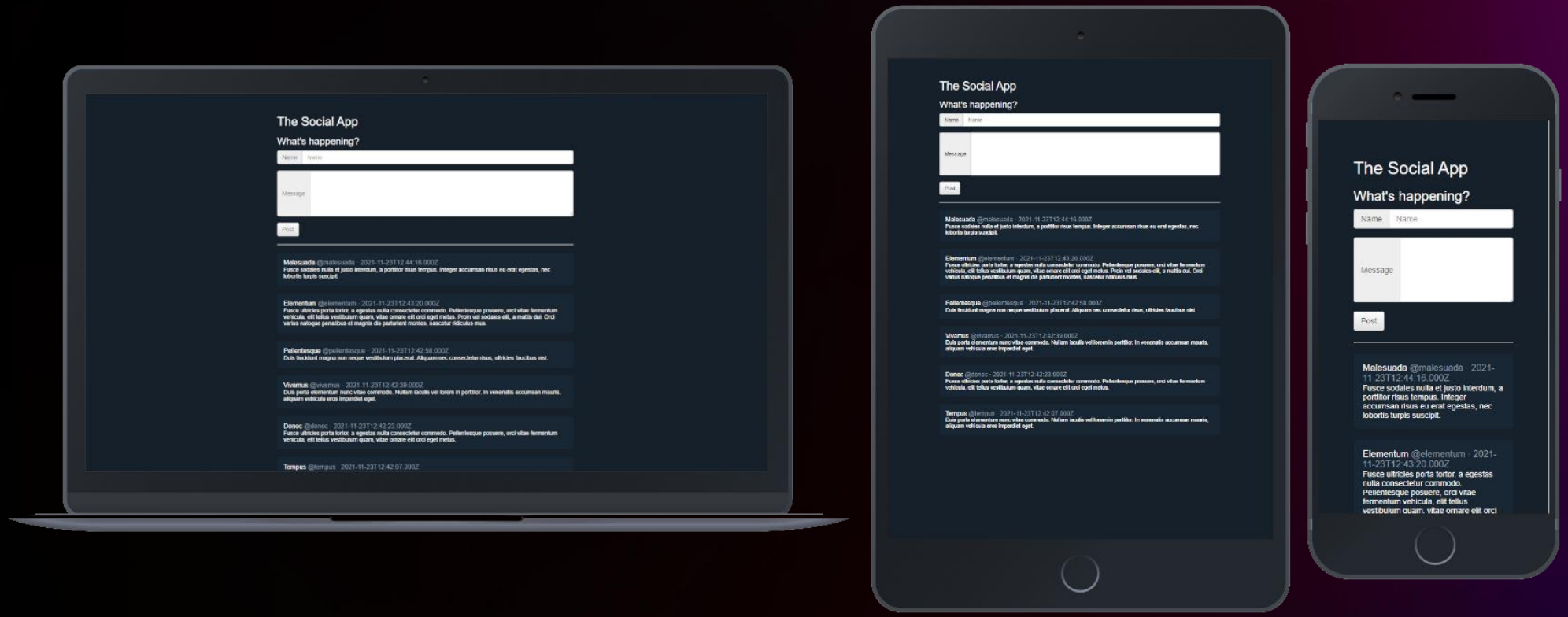
# Serverless architecture




aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Scenario

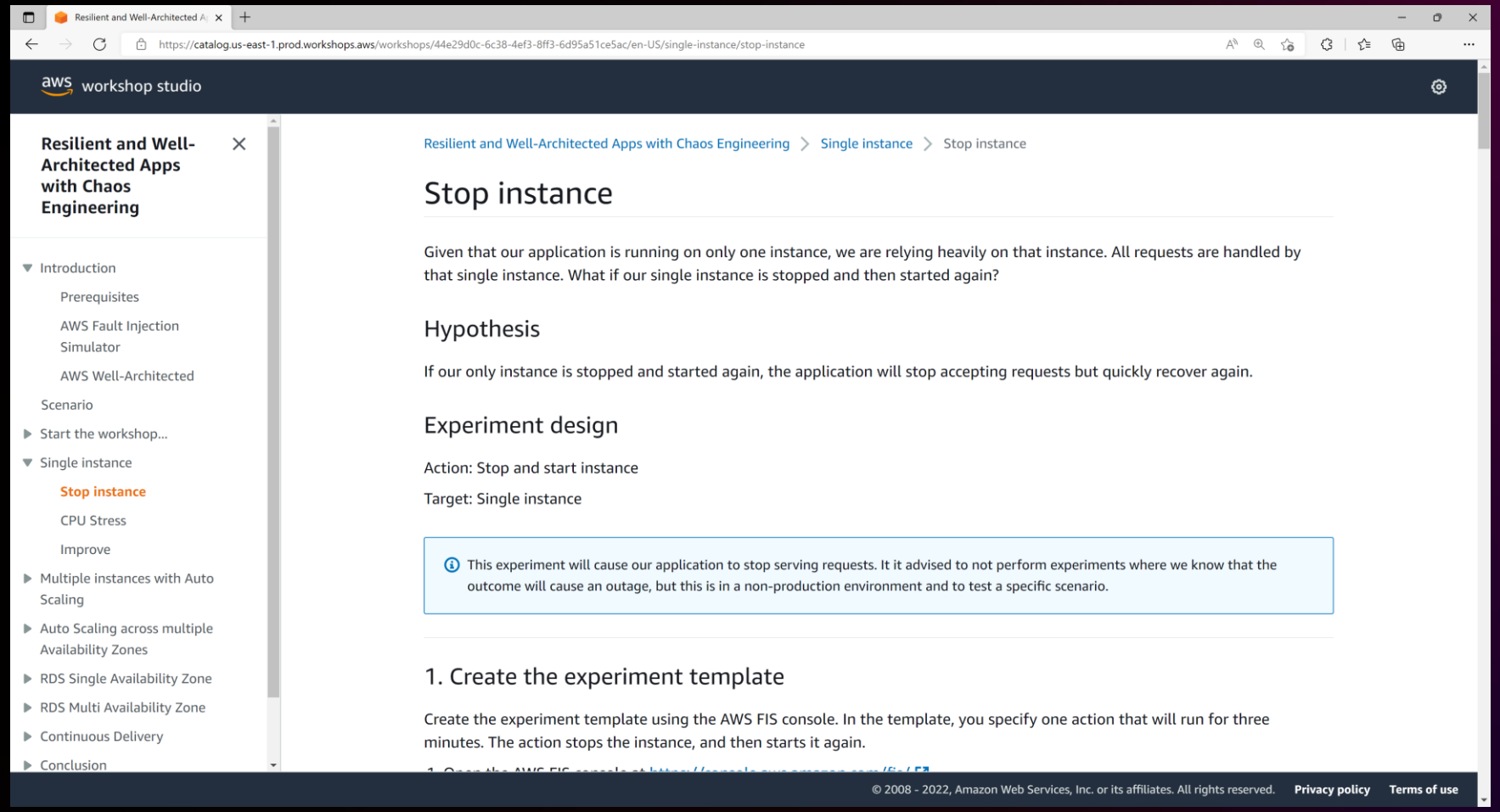
## SUMMARY




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Workshop

## RESILIENT AND WELL-ARCHITECTED APPS WITH CHAOS ENGINEERING




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Resources



grosch.link/wace

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Thank you!

Gunnar Grosch
@gunnargrosch

Sara Gerion
@sarutule



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.