



Resilient and well-architected apps with chaos engineering

Kiranpreet Chawla (she/her)

Solutions Architect
Amazon Web Services

Agenda

AWS Well-Architected

Chaos engineering

AWS Fault Injection Simulator

Examples



AWS Well-Architected



Pillars of AWS Well-Architected



Operational
excellence



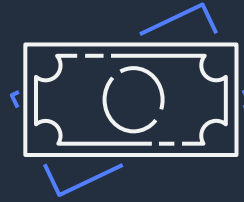
Security



Reliability



Performance
efficiency



Cost
optimization



Sustainability

Pillars of AWS Well-Architected



Operational
excellence



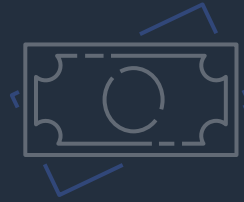
Security



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



Chaos engineering



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

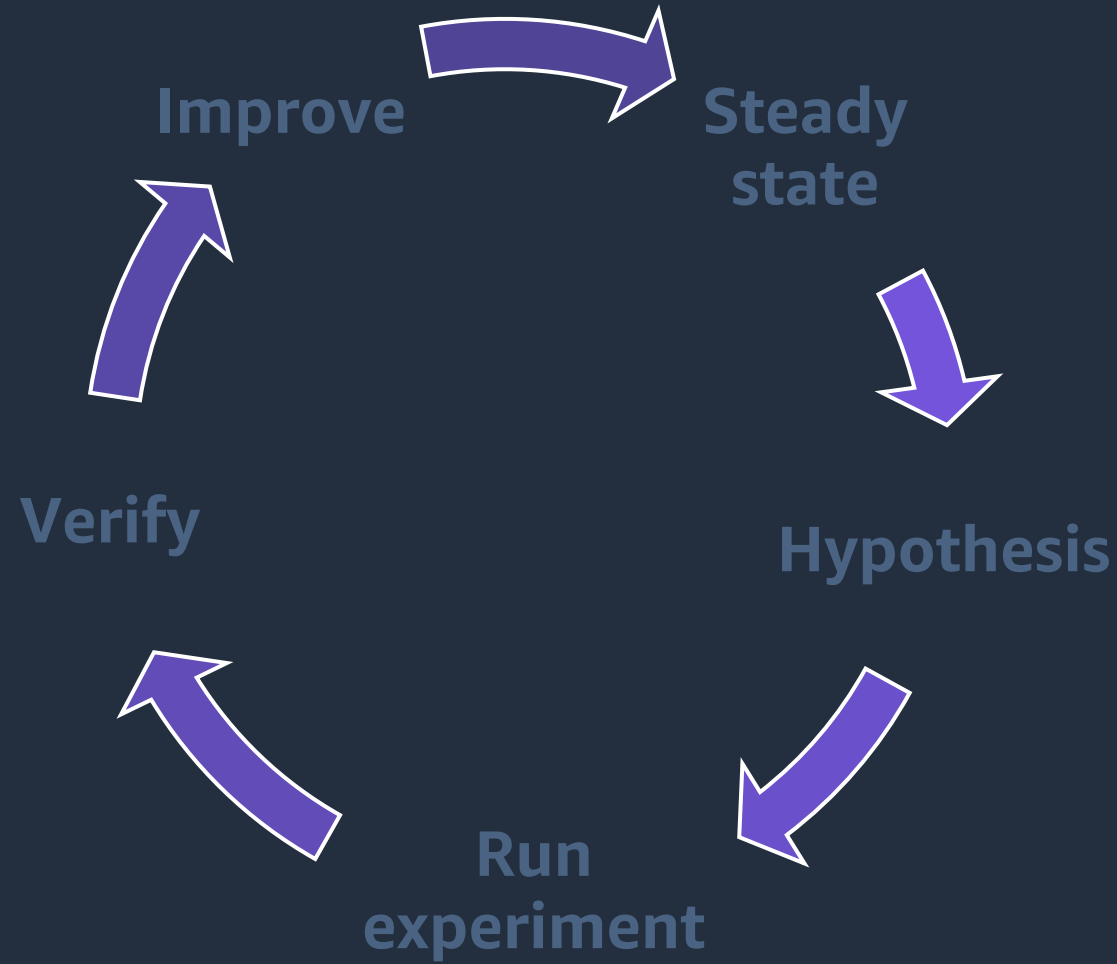
Chaos engineering is the process of **stressing an application by creating disruptive events**, observing how the system responds, and implementing improvements

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

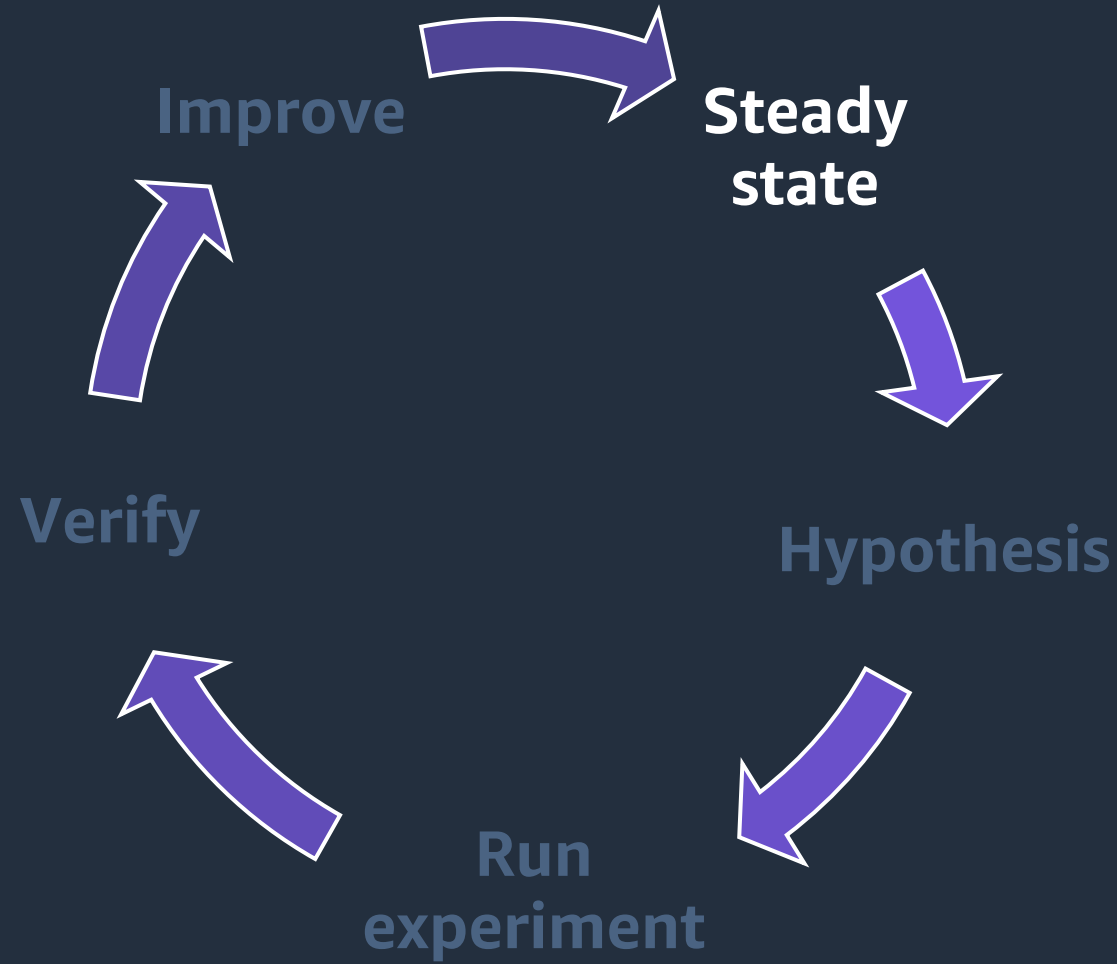
Chaos engineering is the process of stressing an application by creating disruptive events, observing how the system responds, and **implementing improvements**

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

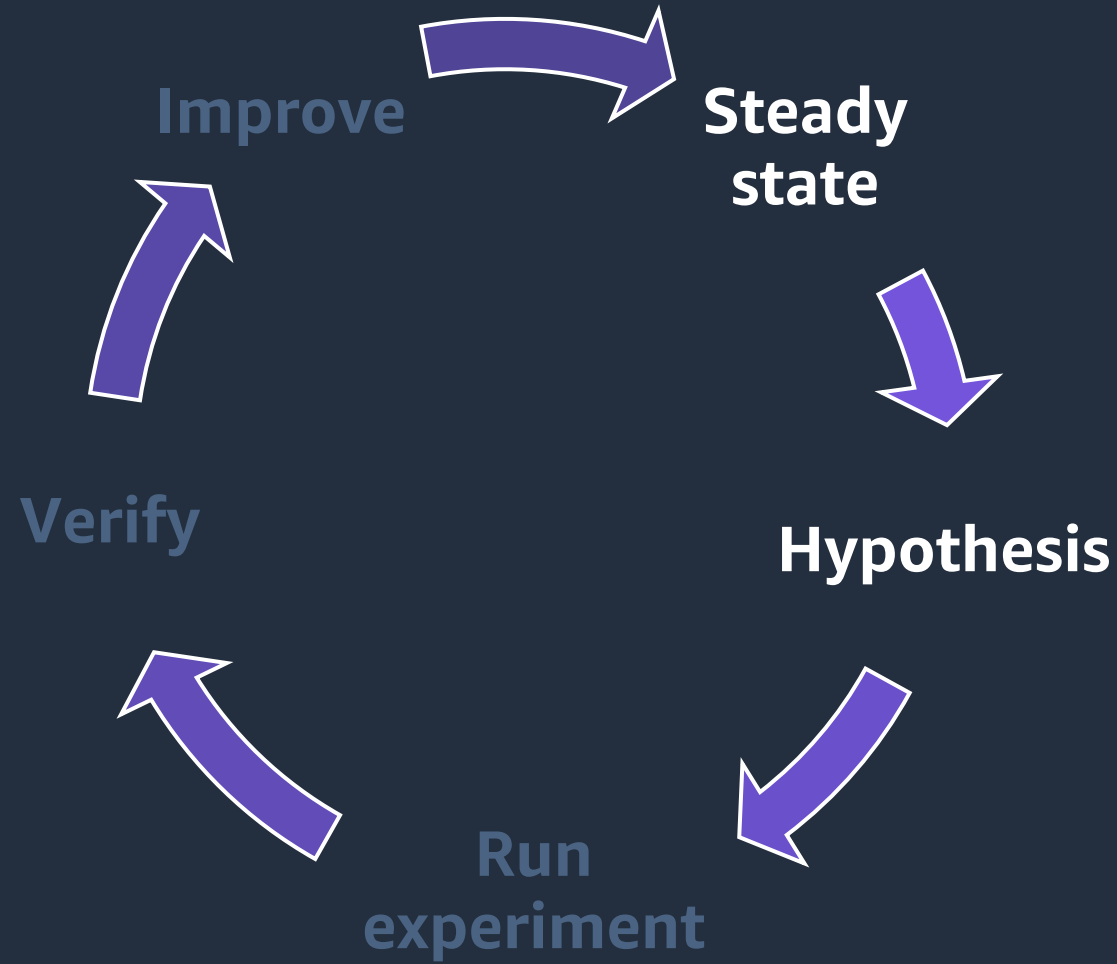
Phases of chaos engineering



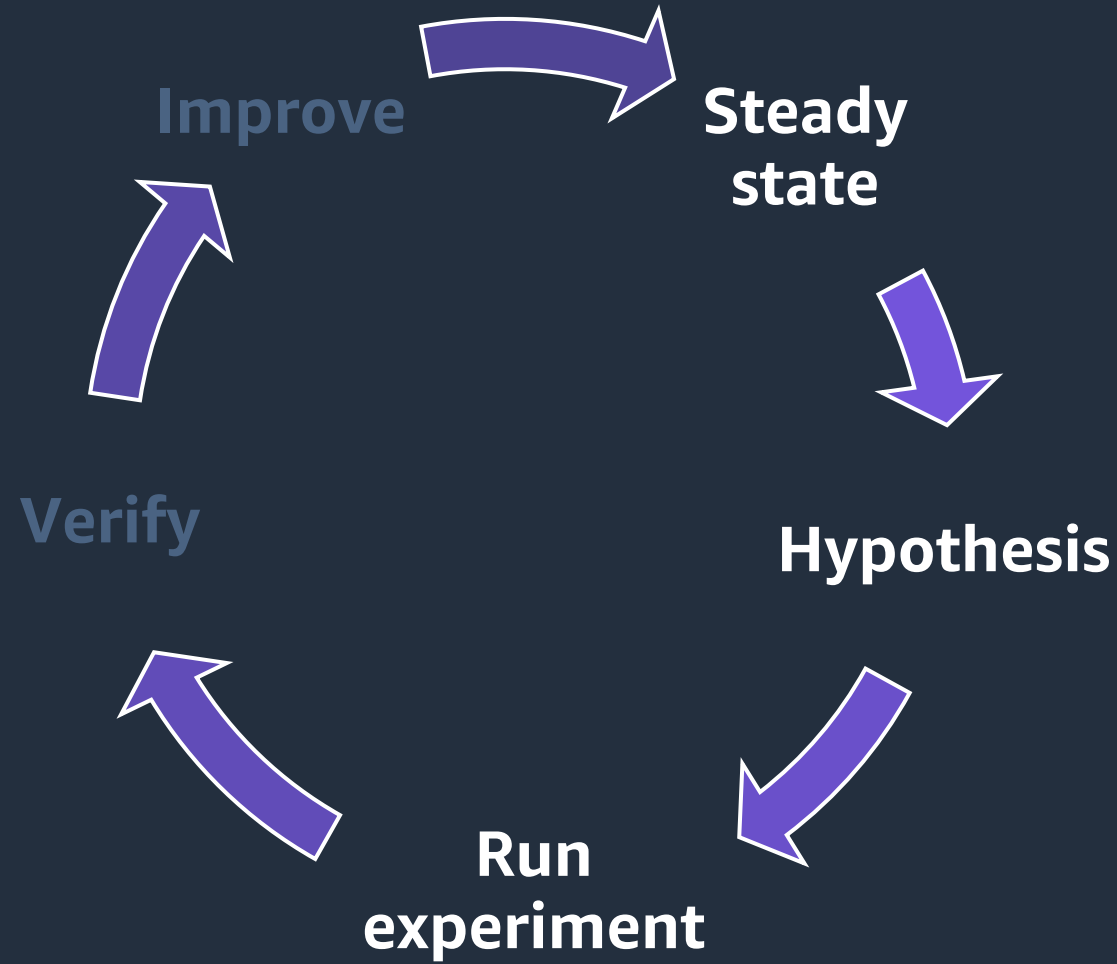
Phases of chaos engineering



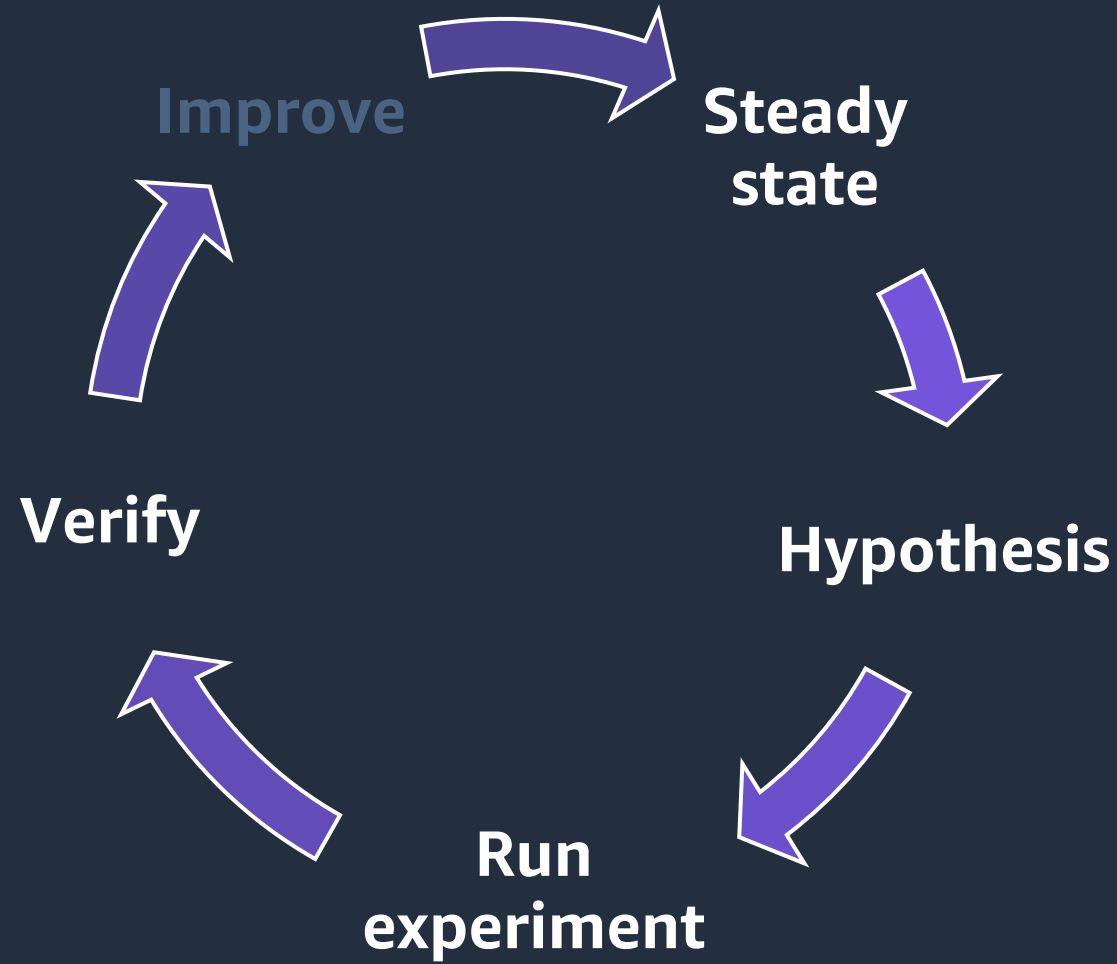
Phases of chaos engineering



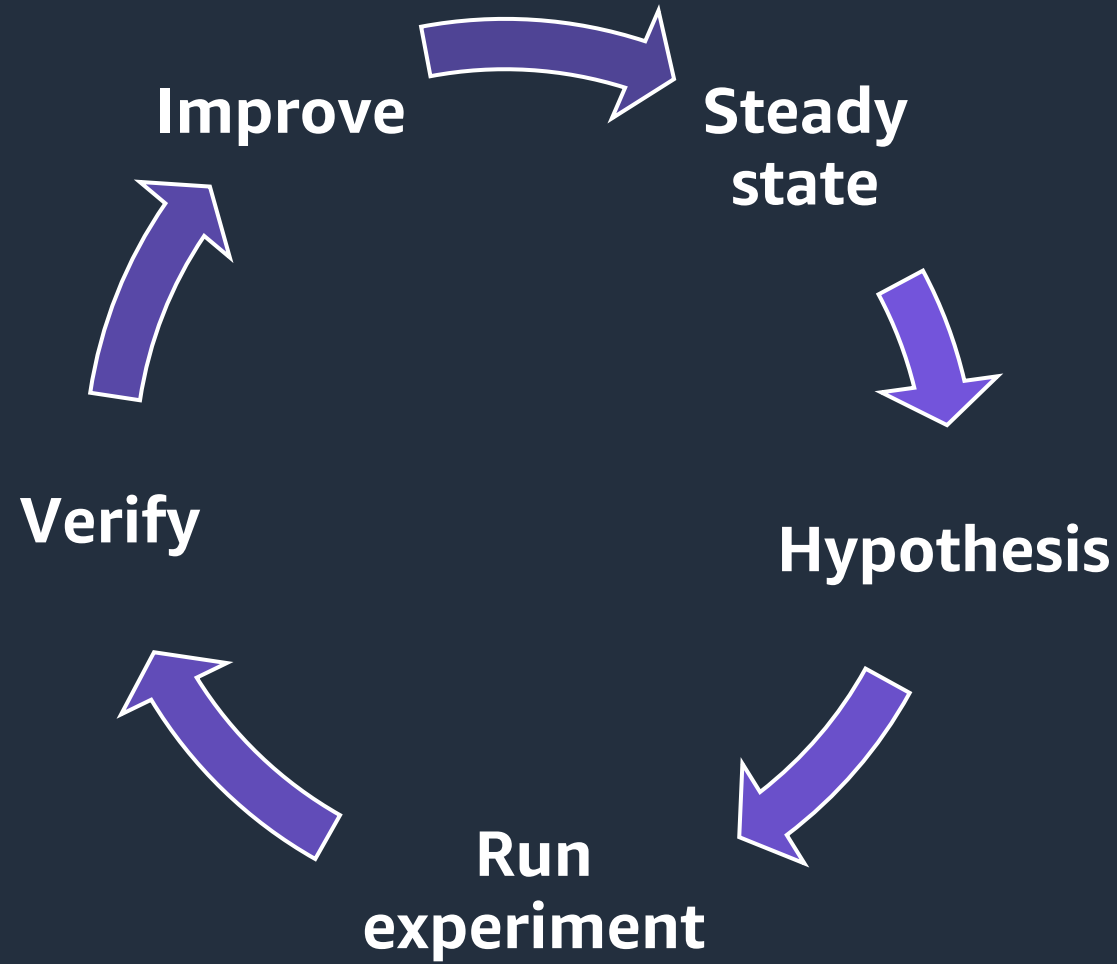
Phases of chaos engineering



Phases of chaos engineering



Phases of chaos engineering



AWS Fault Injection Simulator



AWS Fault Injection Simulator



Easy to
get started



Real-world
conditions



Safeguards

Components



Actions



Targets



Experiment
templates



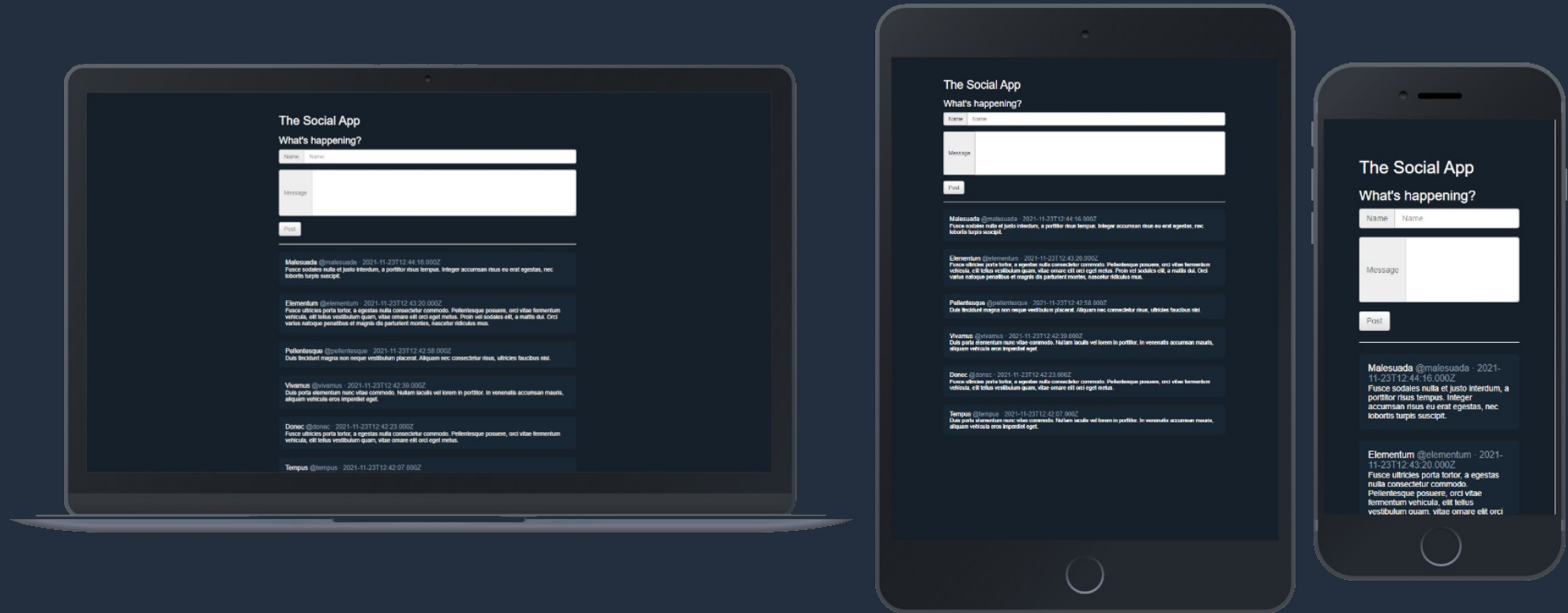
Experiments

Scenario



Scenario

OUR IN-HOUSE BUILT APPLICATION THE SOCIAL APP SHOULD BE MADE PUBLIC



The Social App

What's happening?

Name

Message

Post

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

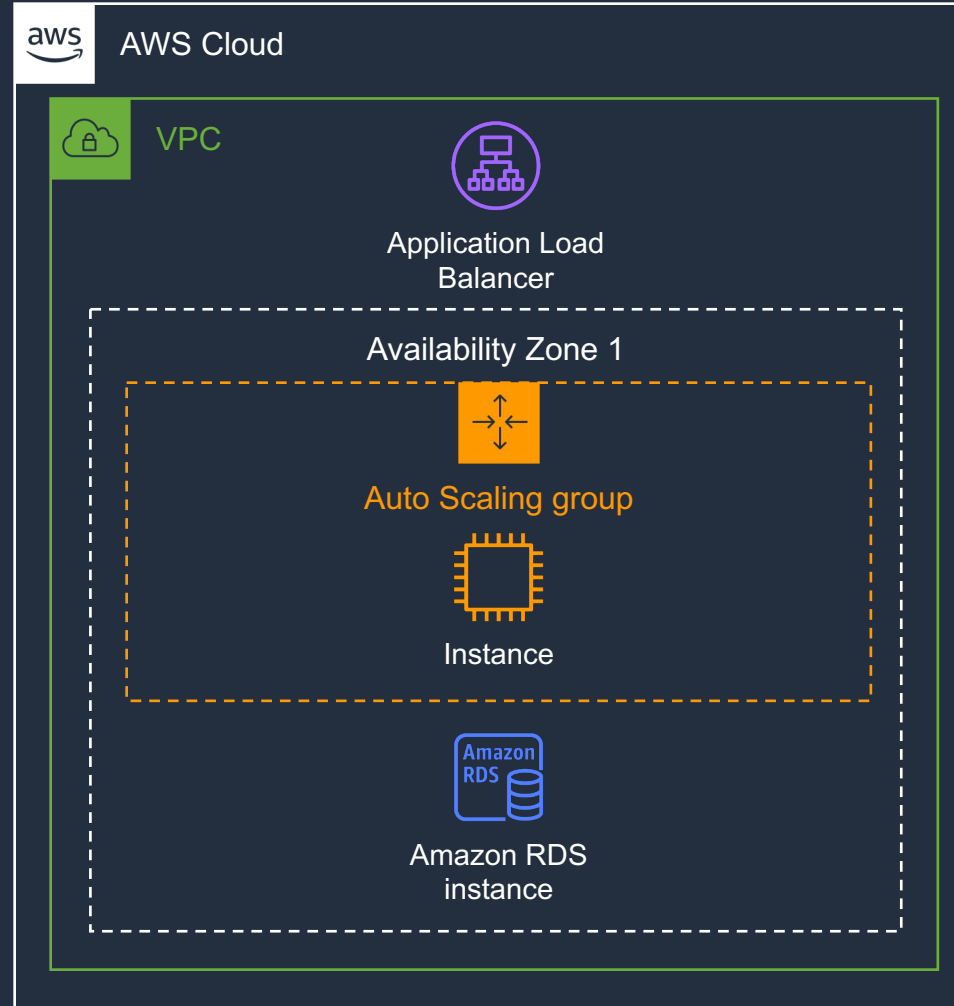
Reyes @reyes
He uses onomatopoeia as a weapon of mental destruction.

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
Each person who knows you has a different perception of who you are.

Single instance



Experiment

STOP INSTANCE

Hypothesis

If our only instance is stopped and started again, the application will stop accepting requests but quickly recover again.

Experiment design

Action: Stop and start instance

Target: Single instance

AWS FIS [Close]

- Experiment templates
- Experiments

Share Feedback [Link]

Developer Tools

AWS Fault Injection Simulator

Improve resiliency and performance with controlled experiments

AWS Fault Injection Simulator is a fully managed service for running fault injection experiments on AWS that makes it easier to continuously improve an application's performance, observability, and resiliency.

Create experiment template

Choose your fault injection actions and the targets to run them on. Then start running your experiments.

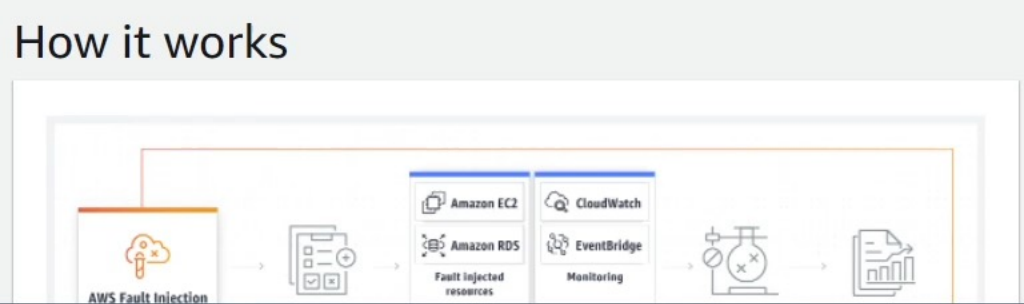
[Create experiment template](#)

Pricing (US)

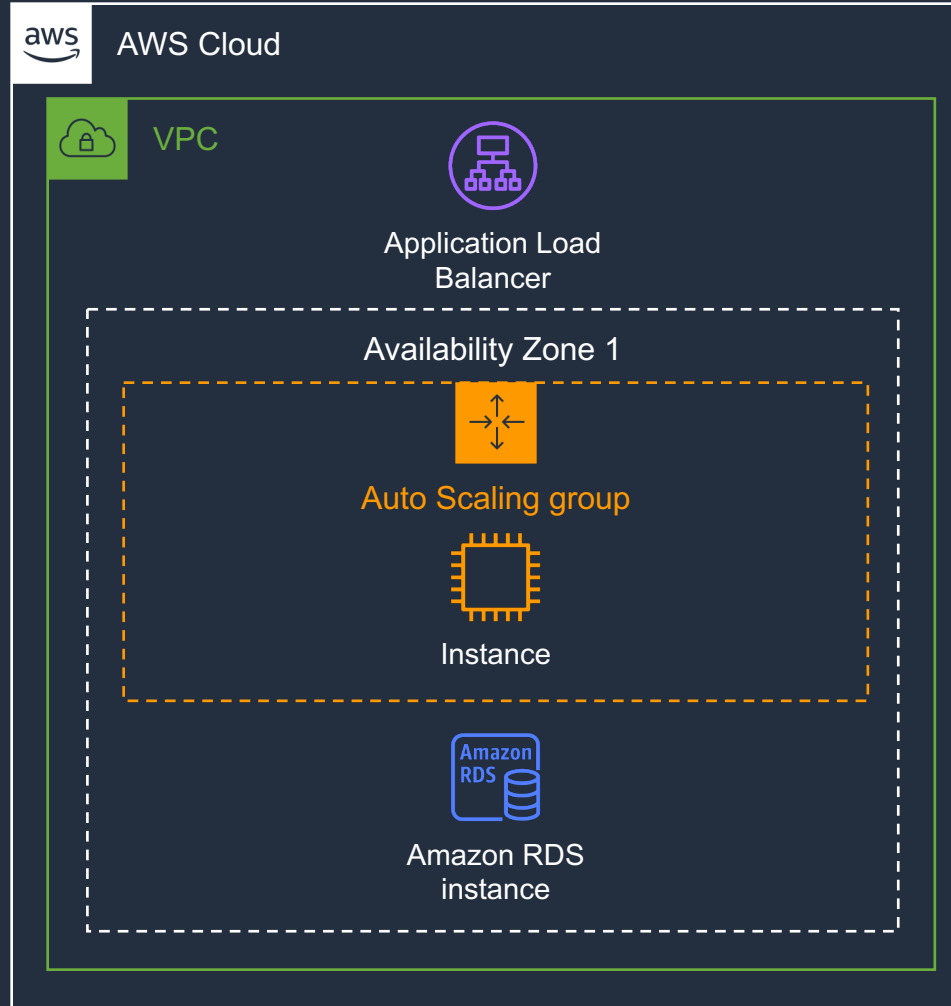
AWS Fault Injection Simulator uses simple pay-as-you-go pricing. There are no upfront costs or minimum fees. You pay per action-minutes you use.

All actions provided	\$0.10 per action-minute
----------------------	--------------------------

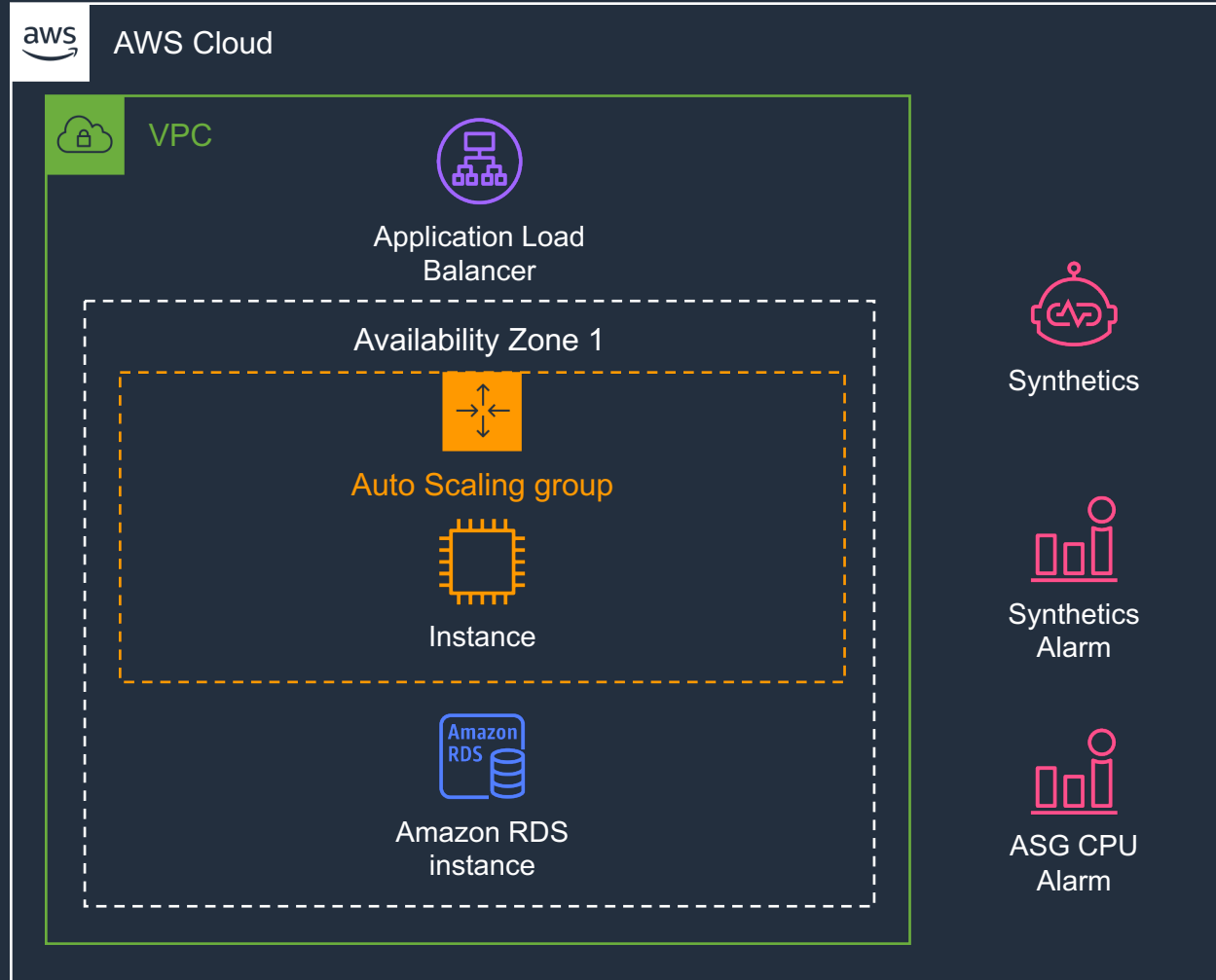
[View pricing](#)



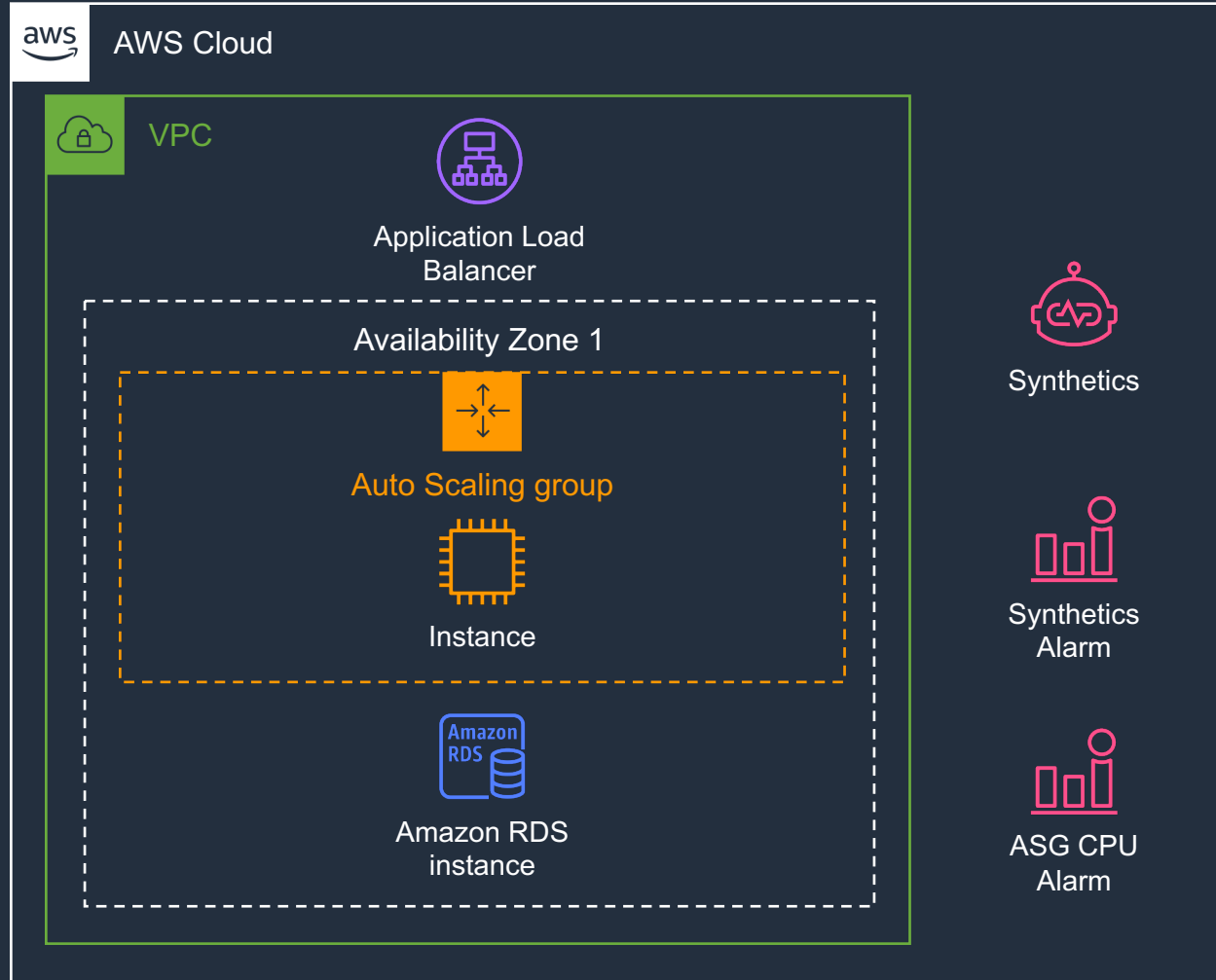
Improve



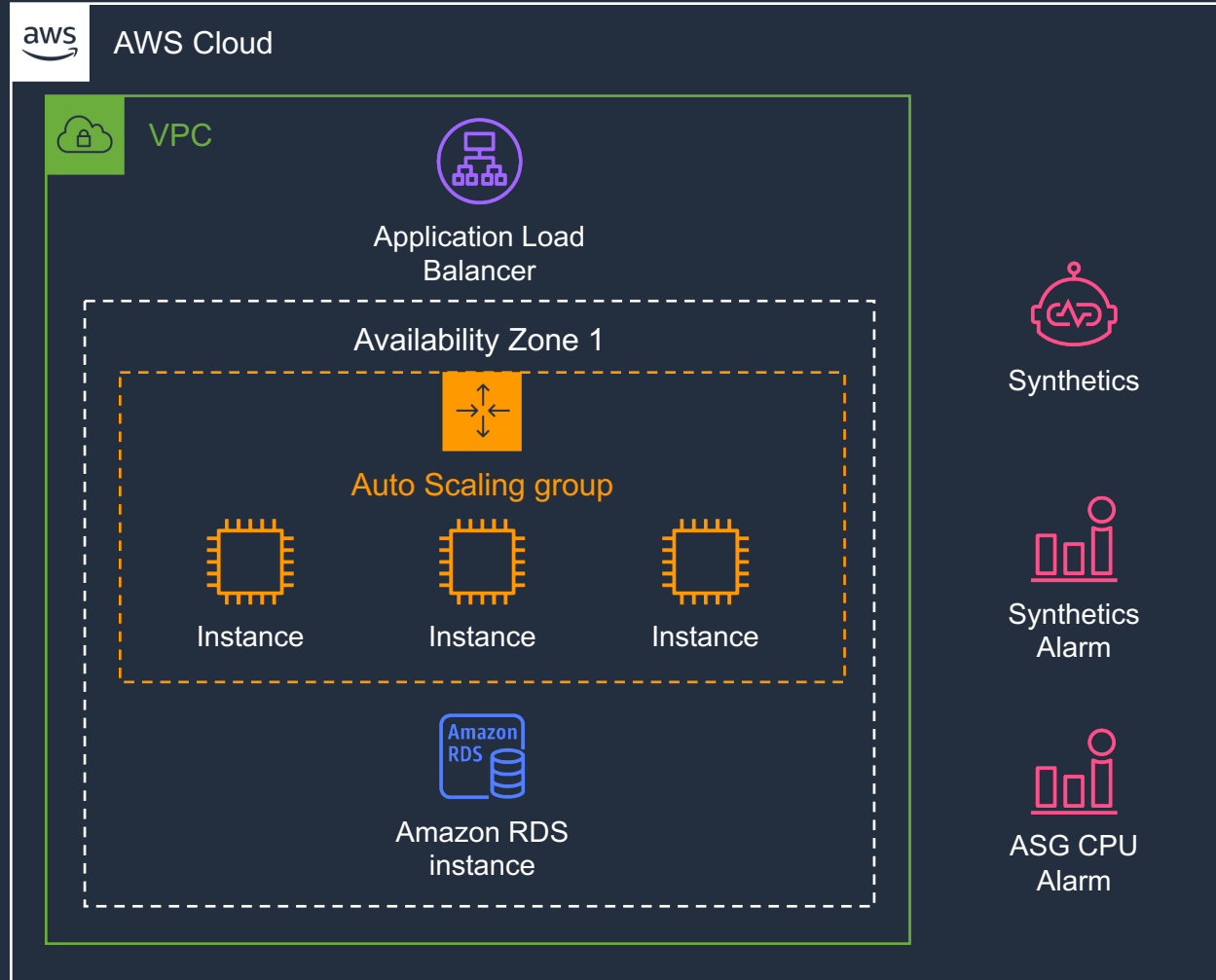
Improve



Improve



Improve



Experiment

STOP INSTANCE

Hypothesis

If one instance in our Auto Scaling group is stopped and started again, the application will continue serving requests to clients.

Experiment design

Action: Stop and start instance

Target: One instance in Auto Scaling group

Summary

- AWS Well-Architected
- Chaos engineering
- AWS Fault Injection Simulator
- Write an experiment



Thank you!