nagarro

webinar

Resilience Engineering

Using chaos to elevate your customers' experience

Speakers

Kyle Hultman

Chaos Account Manager Lead,
Gremlin

Uma Mukkara

CEO, ChaosNative

Yury Niño Roa

SRE Technical Program Manager,
ADL Digital Lab

Moderator

Dushyant Anoop Sahni

Practice Head - ISV and Horizontal Tech,
Nagarro

Cloud Native Chaos Engineering

Business benefits for enterprises

Uma Mukkara, CEO - ChaosNative

Chaos Engineering

Chaos Engineering

What?

Break things on purpose

Don’t wait; fail it

Proactive resilience engineering

Being better prepared for disasters

Why?

Outages are expensive

Outages are inevitable; Be better prepared

Too many unknowns in the modern microservices environment

Because you can; Smart solutions and tools can make the process easier

How?

It is a culture thing; Start with it

Create a strategy; Choose a platform; Build a chaos service

Gamedays

Start small and expand

Chaos Engineering - Business benefits

Shorten your outages

Prevent large revenue losses

Retain customer satisfaction

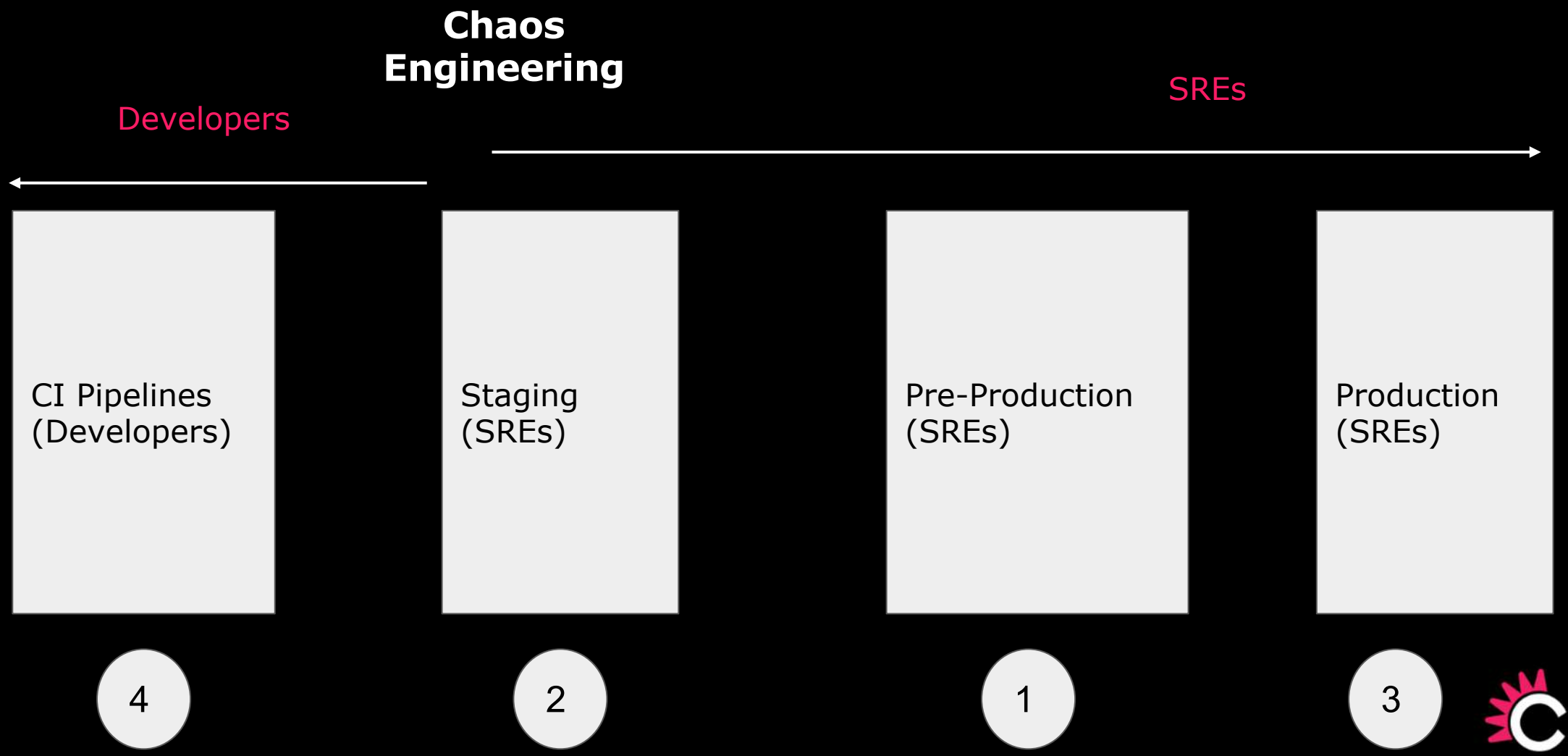
Move to new architectures with confidence

Scale your services faster

Proven training for your response teams

Use cases of Chaos Engineering

Chaos Engineering - Use cases (for who)



Chaos Engineering - Use cases (which areas)

- Benchmark the resilience in the digital transformation journey to micro services
- Accelerate the journey to containerization
  - Benchmark, measure and scale your service
- Large scale critical services moving to containers
  - Banking
  - Retail
  - eCommerce
  - EDGE computing

CI pipelines

Trigger from CD

GameDays

Advanced failure testing in staging

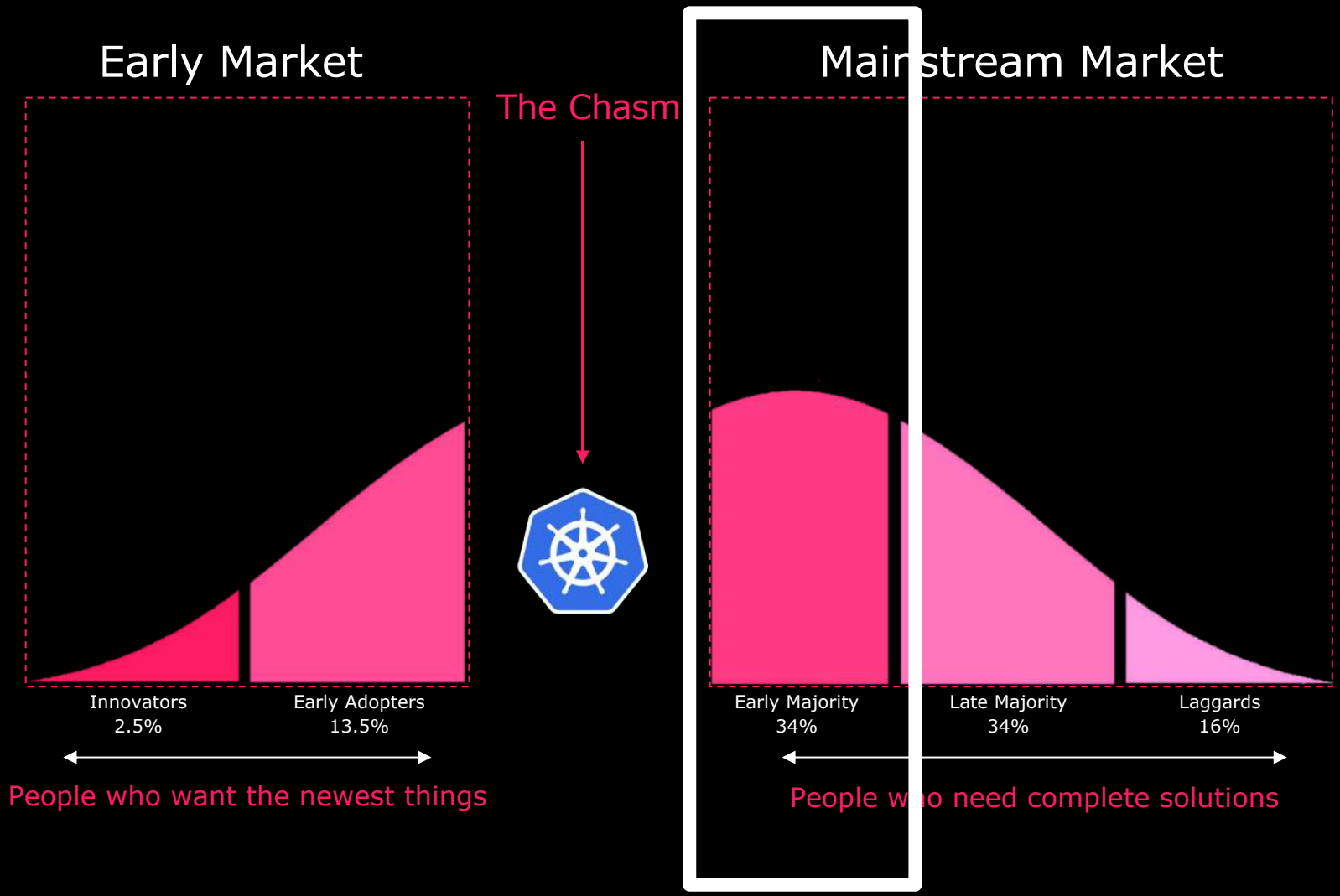
Trigger from GitOps

Trigger to CD

Simple failure testing in production

Cloud Native
Chaos Engineering

Crossing the Chasm - Cloud Native



Reliability in Cloud Native - Why ?

Because - the changes happen too fast; and there are many such services

Cloud native reliability

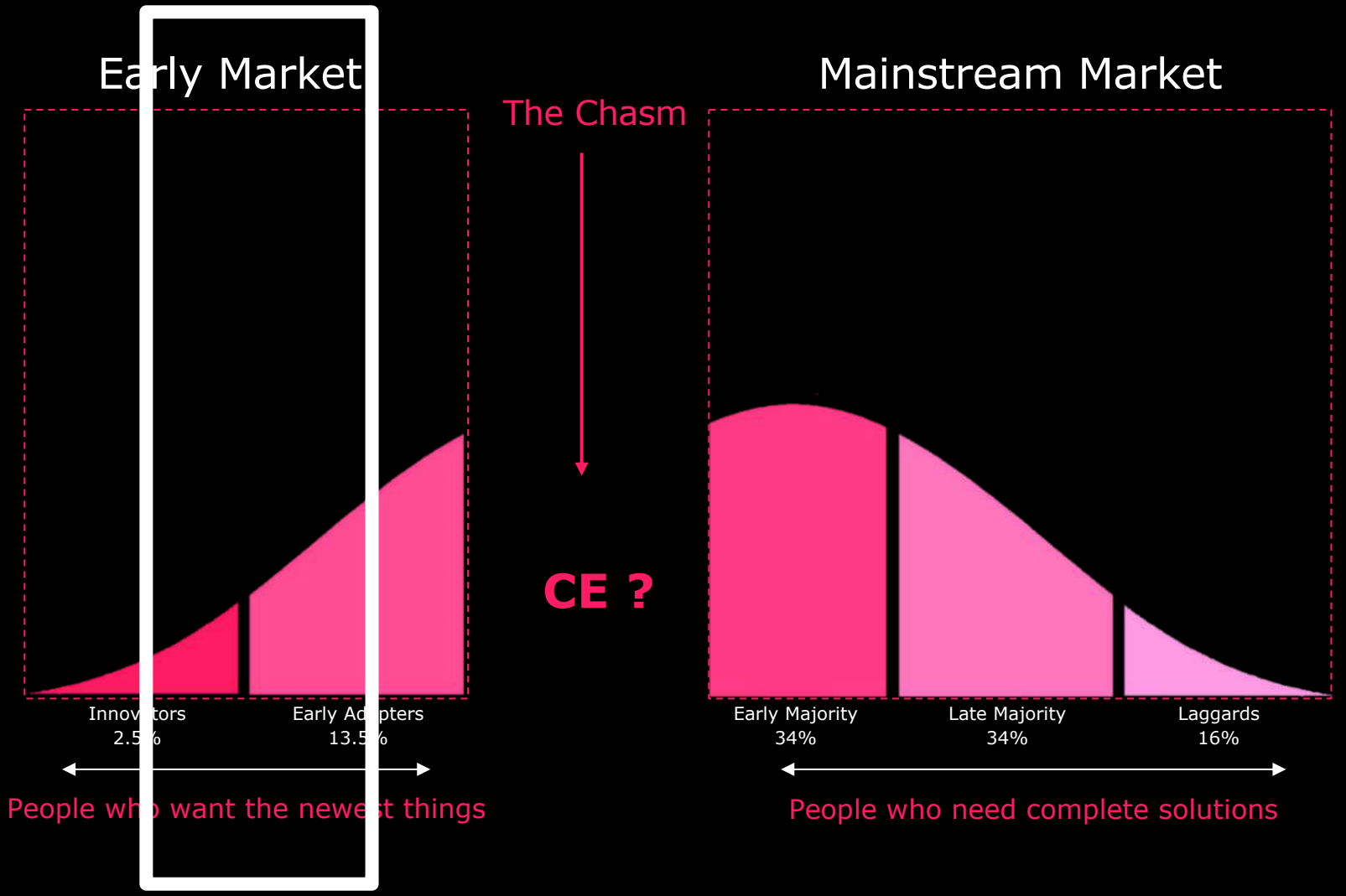
Run services without an outage

Run services the meet the business SLAs or SLOs

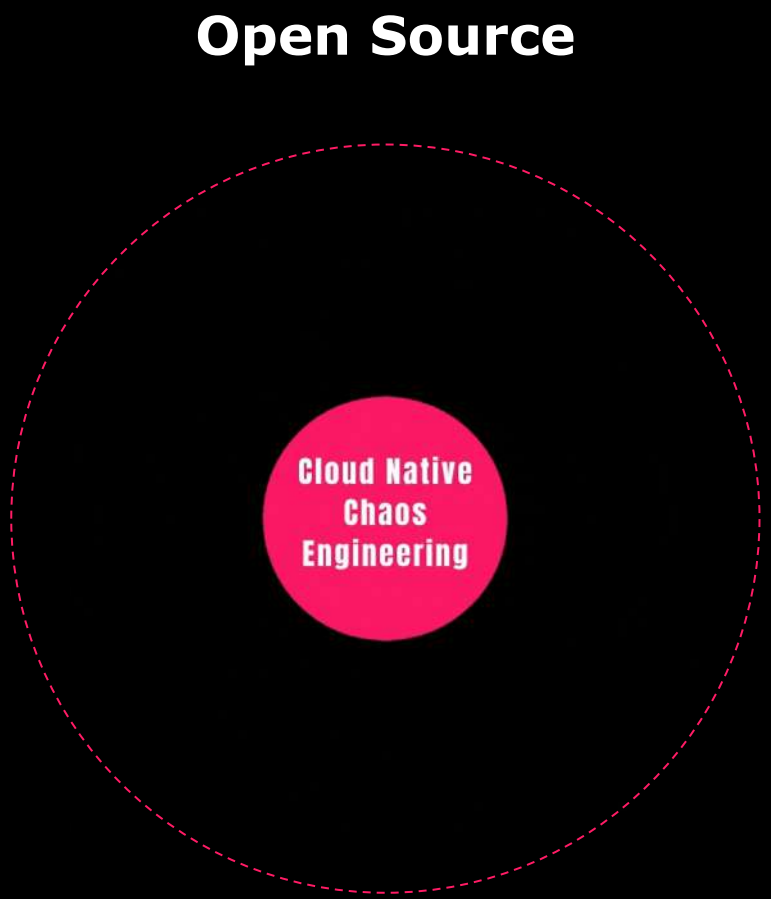
Scale your services on demand

Upgrade your services without an outage

Crossing the Chasm -
Chaos Engineering in Cloud Native?



Principles of Cloud Native Chaos Engineering



NSure/Litmus

Chaos Engineering Platform for Cloud Native

NSure is built for modern Chaos Engineering

Open Source

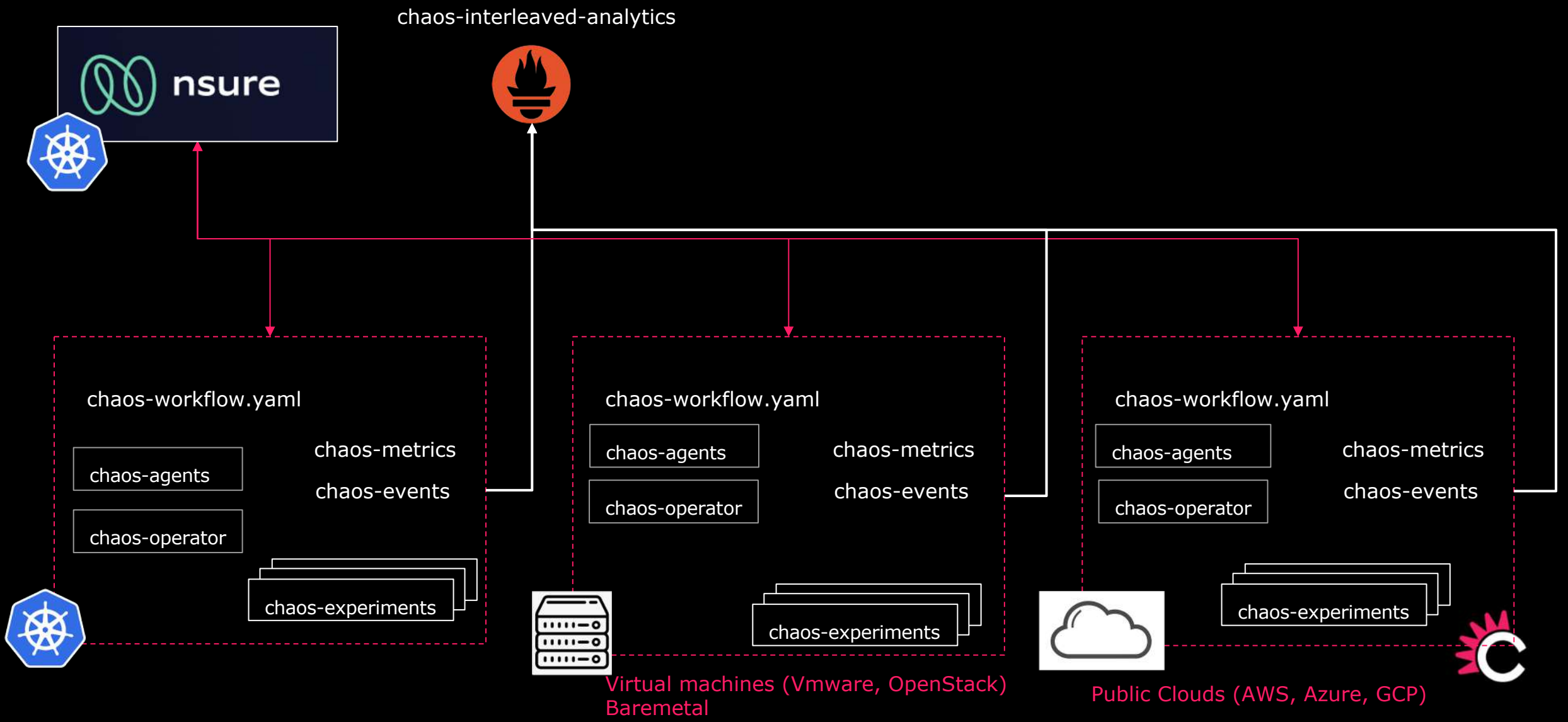
Open Observability

Community Collaboration

GitOps

Open API and lifecycle management

NSure/Litmus in a nutshell



CE for hybrid or multi cloud

Chaos Metrics

ChaosHub(s)

Chaos Execution Plane

SREs/
Developers/
IT Managers

nsure

Chaos
Workflows

GitOps

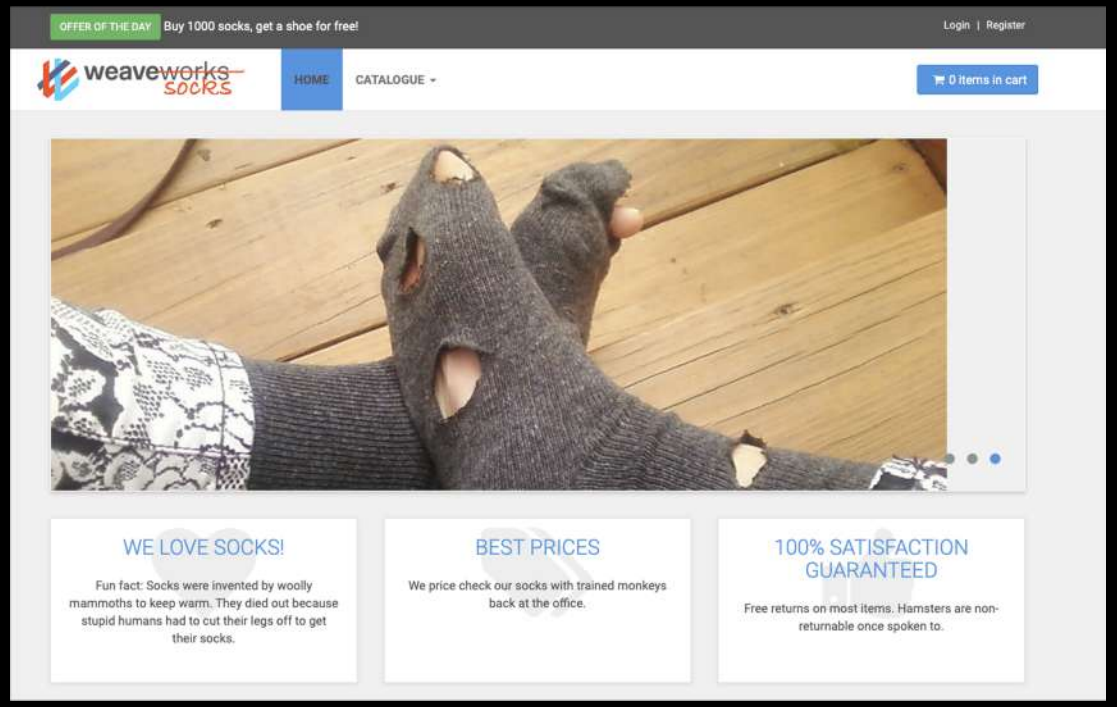
Chaos Control Plane

OnPrem K8s

Legacy
Applications

Let’s see
Chaos Engineering in Action

A quick look at our NSure demo setup



Microservices application - K8s
(catalog, orders, frontend etc)

Steady state hypotheses: orders continue to be processed at certain performance levels

Catalog microservice is not highly available (only one pod in the service)

Test:

- Kill a pod (experiment)
- Measure orders processing during chaos (Probe)
  - Steady state retained?
- Measure orders processing after chaos (Probe)
  - Steady state regained?

Observe:

- See chaos experiment result
- See analytics

Chaos Engineering for your Cloud Native applications

Create, orchestrate, analyze and collaborate your Chaos Workflows.

Username

Password

Login

Forgot your password?

Security Chaos Engineering

Securing the Cloud Differently

02

Nice to meet you

Yury Niño Roa

SRE Technical Program Manager
Chaos Engineering Advocate

www.yurynino.com

What is Security Chaos Engineering?

It is the identification of security control failures through proactive experimentation to build confidence in the system's ability to defend against malicious conditions in production.

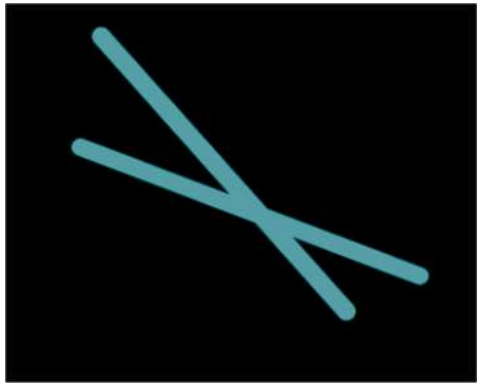
Chaos Engineering Book. 2020

Red/Blue Teaming Exercises

- They were originated with the US Armed Forces by Bryce Hoffman.
- Adversarial approach that imitates the behaviors and techniques of attackers in the most realistic way possible.
- Two common forms of **Red Teaming** seen in the enterprise are:
  - Ethical hacking
  - Penetration testing.
- **Blue Teams** are the defensive counterparts to the **Red Teams** in these exercises.

Purple Team Exercises

- The “Purple” in Purple Teaming reflects the cohesion of Red and Blue Teaming.
- They were intended as an evolution of Red Team exercises by delivering a more cohesive experience between the offensive and defensive teams.
- The goal: collaboration of offensive and defensive tactics to improve the effectiveness of both groups.
- They increase transparency and allow to learn about how effective is the preparation of engineers.



It is not the intention to overlook the value of Red and Purple Team Exercises!

PenTests, Red/Blue Exercises even Purple Teaming are not enough!

This requires a fundamentally new approach to cybersecurity, one that keeps pace with the rapidly evolving world of software engineering.

Chaos Gamedays

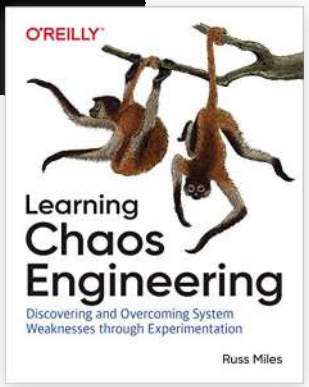
GameDays are interactive team-based learning exercises designed to give players a chance to put their skills to the test in a real-world, gamified, risk-free environment.

powered by aws

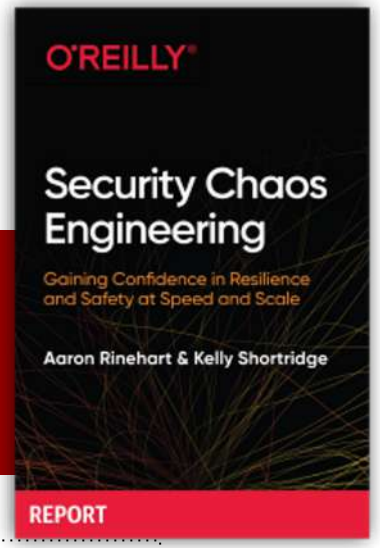
A **Chaos GameDay** is a practice event, and although it *can* take a whole day, it usually requires only a few hours. The goal of a **GameDay** is to practice how you, your team, and your supporting systems deal with real-world turbulent conditions.

DIUS

TARGET



Framework for running a Gameday



Before

- Pick a hypothesis
- Pick a style
- Decide who
- Decide where
- Decide when
- Document
- Get approval

During

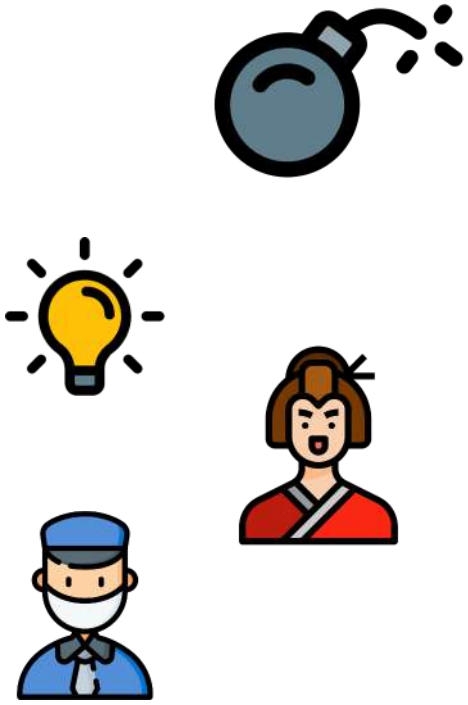
- Detect the situation
- Take a deep breath
- Communicate
- Visit dashboards
- Analyze data
- Propose solutions
- Apply and solve

After

- Write a postmortem
- What happened
- Impact
- Duration
- Resolution time
- Resolution
- Timeline
- Action items

During in a Security Chaos Gameday

- Introduce latency on security controls.
- Drop a folder like a script would do in production.
- Software secret clear text disclosure.
- Permission collision in a shared IAM role policy.
- Disable service event logging.
- API gateway shutdown.
- Unencrypted S3 Bucket.
- Disable MFA.



Infection **Monkey**

1. Run C&C Server ✓
2. Run Monkey ✓
3. Infection Map ✓
4. Security Report ✓
5. Start Over

Configuration

Log

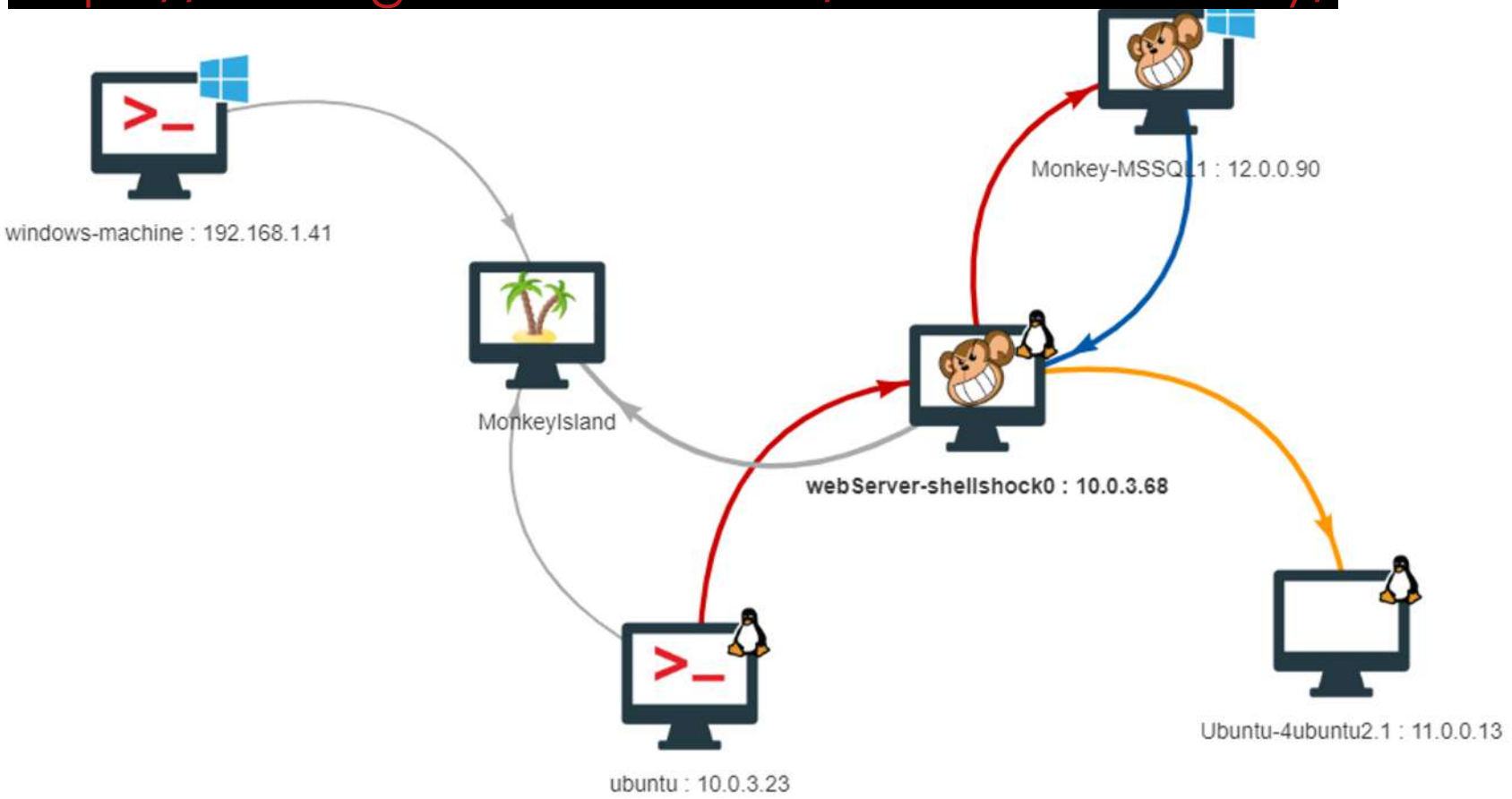
Powered by GuardiCore

License

3. Infection Map

Legend: Exploit | Scan | Tunnel | Island Communication

https://www.guardicore.com/infectionmonkey/



4. Security Report

Print Report

Security Report
Infection Monkey

Overview

Critical security issues were detected!

The first monkey run was started on 31/10/2017 11:00:44. After 21 days, 6 hours, 16 minutes and 47 seconds, all monkeys finished propagation attempts.

The monkey started propagating from the following machines where it was manually installed:

- ubuntu
- windows-machine

The monkeys were run with the following configuration:

Usernames used for brute-forcing:

- Administrator

Passwords used for brute-forcing:

- Pas******

Note: Monkeys were configured to avoid scanning of the local network.

Security Findings

Immediate Threats

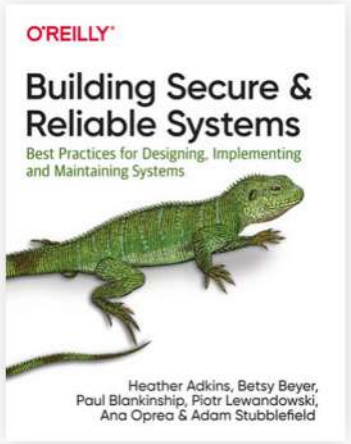
During this simulated attack the Monkey uncovered 2 threats:

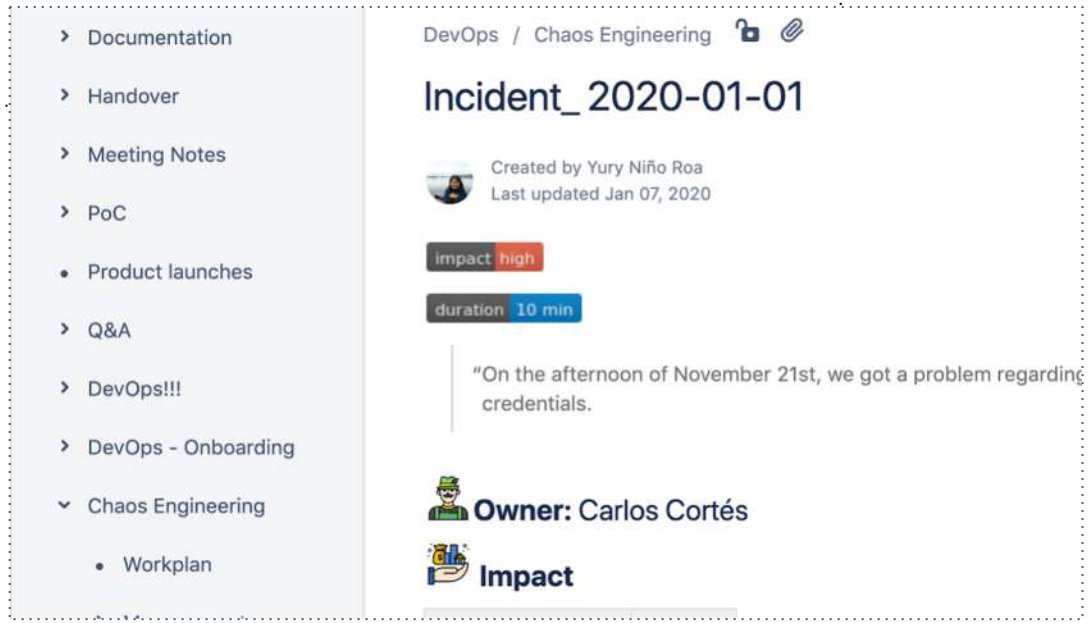
- Machines are vulnerable to 'Shellshock' (CVE-2014-6271).

After

A **security postmortem** covers technology issues that the attacker exploited, and also recognizes opportunities for improved incident handling.

Document the time frames and efforts associated with these action items, and decide which action items.





During

Hypothesis:

After the owner of Root account in AWS left the company, we could use our cloud in a normal way.

Result

Hypothesis disproved. In this experiment the access to AWS was connected to the Active Directory. When an employee left the company his account is dropped and we lost the access to AWS.

Side Effect

Thinking in this scenario allows to consider another applications connected to Active Directory.

If you know the enemy and know yourself, you need not fear the result of a hundred battles …

The Art of War. Sun Tzu

Chaos Engineering

Enterprise Grade Gremlin

Kyle Hultman
Director: Professional Services, Agent of Chaos
@BlueHairGremlin
kyle@gremlin.com

go.gremlin.rocks/nagarro

Failures are inherent to **complex systems** and will cause downtime **unless tested for**.

Measuring the Cost of Downtime

Cost = R + E + C + (B + A)

During the Outage

R = Revenue Lost

E = Employee Productivity

After the Outage

C = Customer Chargebacks

(SLA Breaches)

Unquantifiable

B = Brand Defamation

A = Employee Attrition

The average company loses **$300,000/hour** of downtime

Gremlin

Break Things On Purpose

Build more reliable software & processes with our Chaos Engineering platform.

go.gremlin.rocks/nagarro

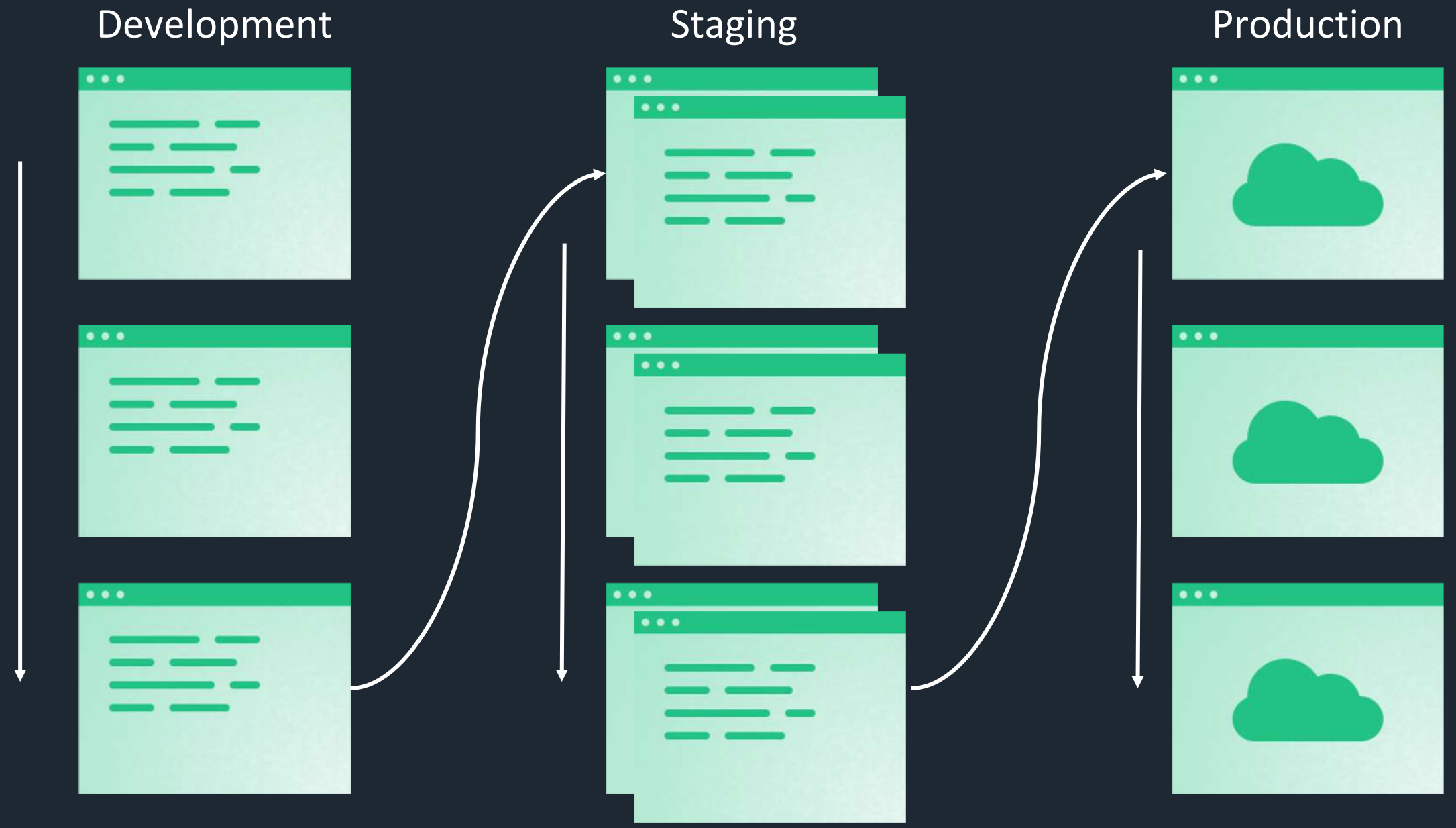
Thoughtful, **planned** experiments designed to **reveal weakness** in our systems.

JPMorgan Chase & Co.

The rigorous compliance requirements of the financial industry add additional challenges to developer velocity. JPMorgan Chase automated all new cloud environments that included Gremlin in order to keep compliance out of the way and streamline developer productivity.

The Enterprise Journey

to Chaos Engineering


Development
Staging
Production

**“Without production testing,
recovery won't work when called upon.”**

James Hamilton
Distinguished Engineer, AWS

Chaos Engineering

01 Resource failure

02 Service failure

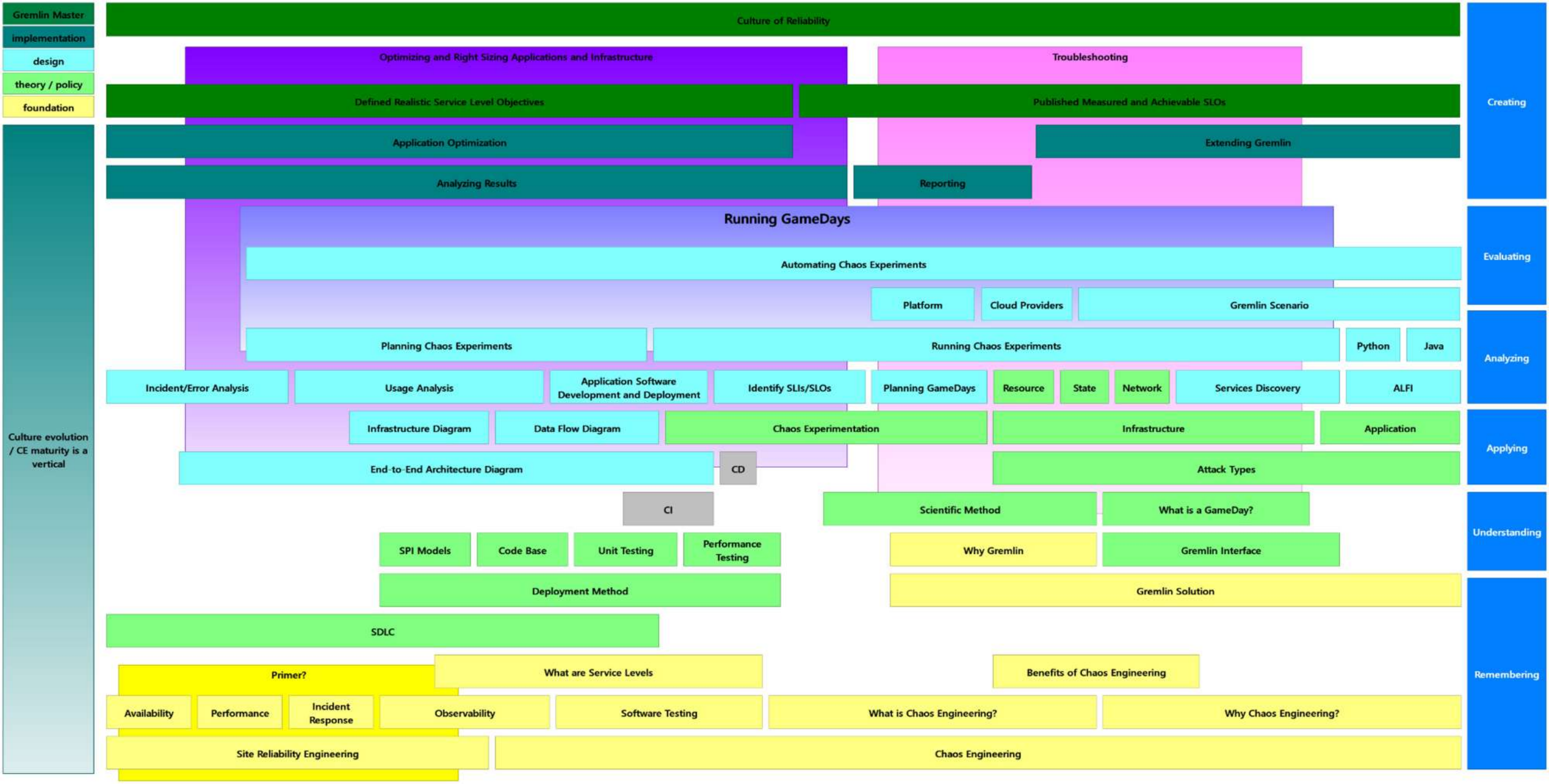
03 Dependency failure

04 Application failure

05 Continuous Chaos

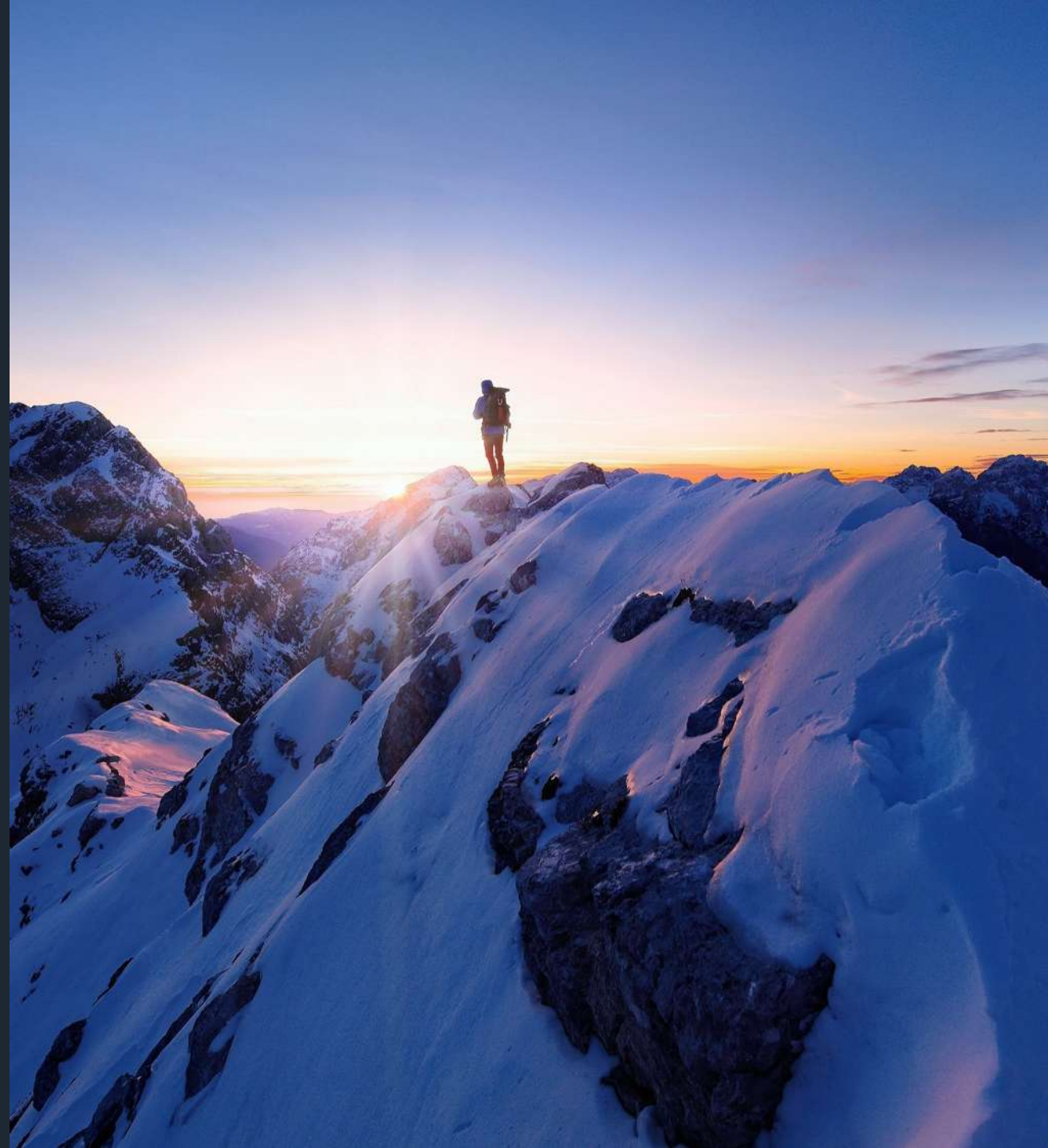
Gremlin

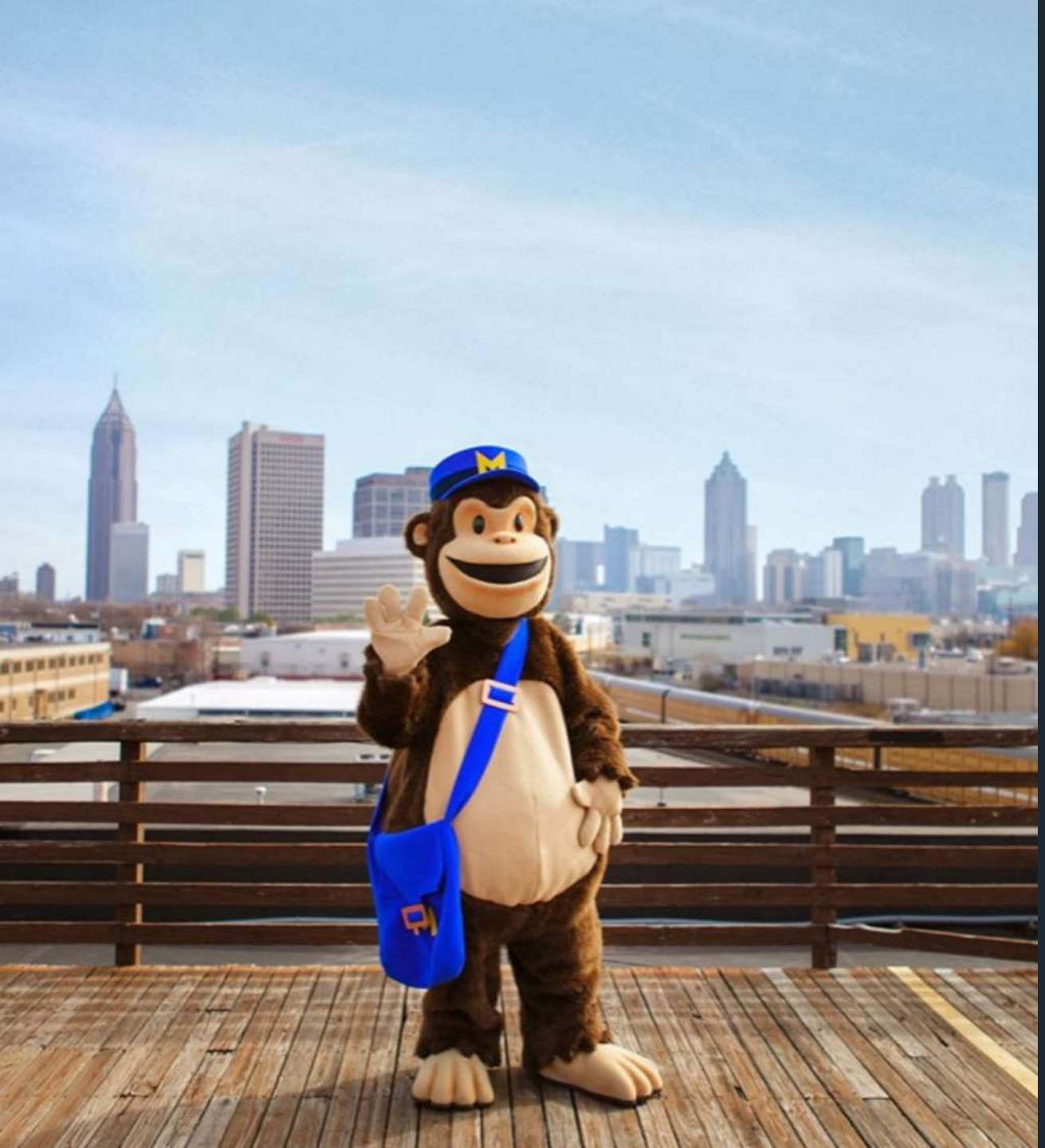
Does it really work?


Gremlin Master
implementation
design
theory / policy
foundation
Culture evolution / CE maturity is a vertical
Culture of Reliability
Optimizing and Right Sizing Applications and Infrastructure
Troubleshooting
Defined Realistic Service Level Objectives
Published Measured and Achievable SLOs
Application Optimization
Extending Gremlin
Analyzing Results
Reporting
Running GameDays
Automating Chaos Experiments
Platform
Cloud Providers
Gremlin Scenario
Planning Chaos Experiments
Running Chaos Experiments
Python
Java
Incident/Error Analysis
Usage Analysis
Application Software Development and Deployment
Identify SLIs/SLOs
Planning GameDays
Resource
State
Network
Services Discovery
ALFI
Infrastructure Diagram
Data Flow Diagram
Chaos Experimentation
Infrastructure
Application
End-to-End Architecture Diagram
CD
Attack Types
CI
Scientific Method
What is a GameDay?
SPI Models
Code Base
Unit Testing
Performance Testing
Why Gremlin
Gremlin Interface
Deployment Method
Gremlin Solution
SDLC
Primer?
What are Service Levels
Benefits of Chaos Engineering
Availability
Performance
Incident Response
Observability
Software Testing
What is Chaos Engineering?
Why Chaos Engineering?
Site Reliability Engineering
Chaos Engineering
Creating
Evaluating
Analyzing
Applying
Understanding
Remembering

backcountry

Following a 72-hour SLO breach on Black Friday in 2017, Backcountry introduced latency to their production systems. This allowed them to verify the fix to an issue affecting their picking robots.





mailchimp

Blackholed a service **NOT** in their critical path, which resulted in all pages serving a 503 error page and ultimately rendered their entire app unusable.


Walmart


Expedia


mailchimp


MEDIALAAN


qualtrics


DREAMWORKS


MoneyLion


FUJITSU


Mobacar
Intelligent Mobility Solutions


SIEMENS


UNDER ARMOUR


twilio

Thank you!

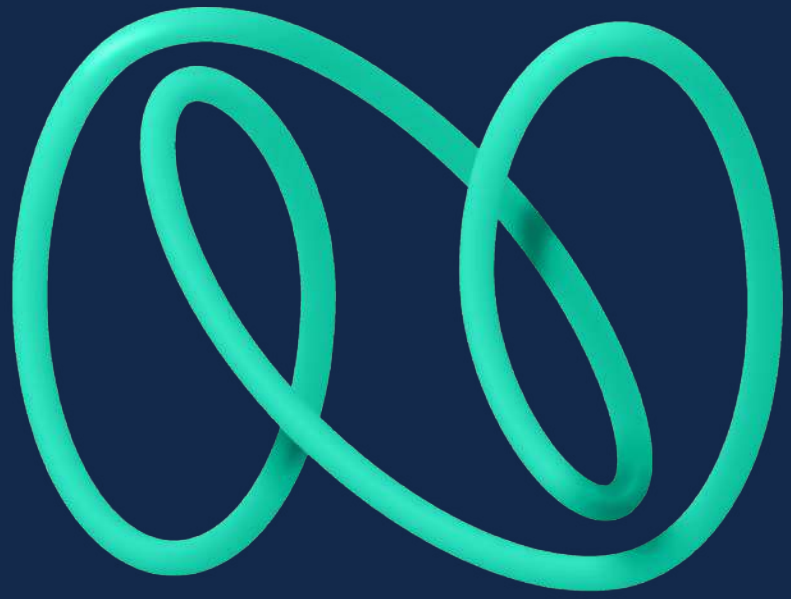
go.gremlin.rocks/nagarro

Questions?

Kyle Hultman
Agent of Chaos, Gremlin

kyle@gremlin.com
@BlueHairGremlin

Questions?



You can also send in your questions to *horizontal.tech@nagarro.com*.