KubeCon CloudNativeCon

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

AI_dev

Open Source GenAI & ML Summit

China 2024

KubeCon

CloudNativeCon

THE LINUX FOUNDATION



China 2024

# Enhancing Cyber Resilience

## Through Zero Trust Chaos Experiments in Cloud Native Environments

Rafik Harabi, Senior Solutions Architect - Sysdig
Sayan Mondal, Senior Software Engineer - Harness

# Who we are?



- Senior Solution Architect at Sysdig, Cloud Security Advocate
- Focus on Cloud Native Security
- Previously working on go to Cloud programmes

rafikharabi

@rafik8_

- Senior Software Engineer II at Harness
- Maintainer of LitmusChaos (CNCF Incubating)
- LFX Mentor
- Chaos Engineering Practitioner

s-ayanide

@s_ayanide

# Agenda



- Cloud Native Application and Threat Landscape
- Chaos Engineering and Cyber Resilience
- Enhance Security with Chaos Engineering
- Solutions Architecture
- Tooling and Architecture

- Hands on demo
- Next steps
- Takeaways

# Once, there was a perimeter



You had a perimeter **guarded by a firewall**

---

**Detecting intrusions** was your breach indicator

# Now, there is no perimeter in the cloud



Cloud providers own external connections

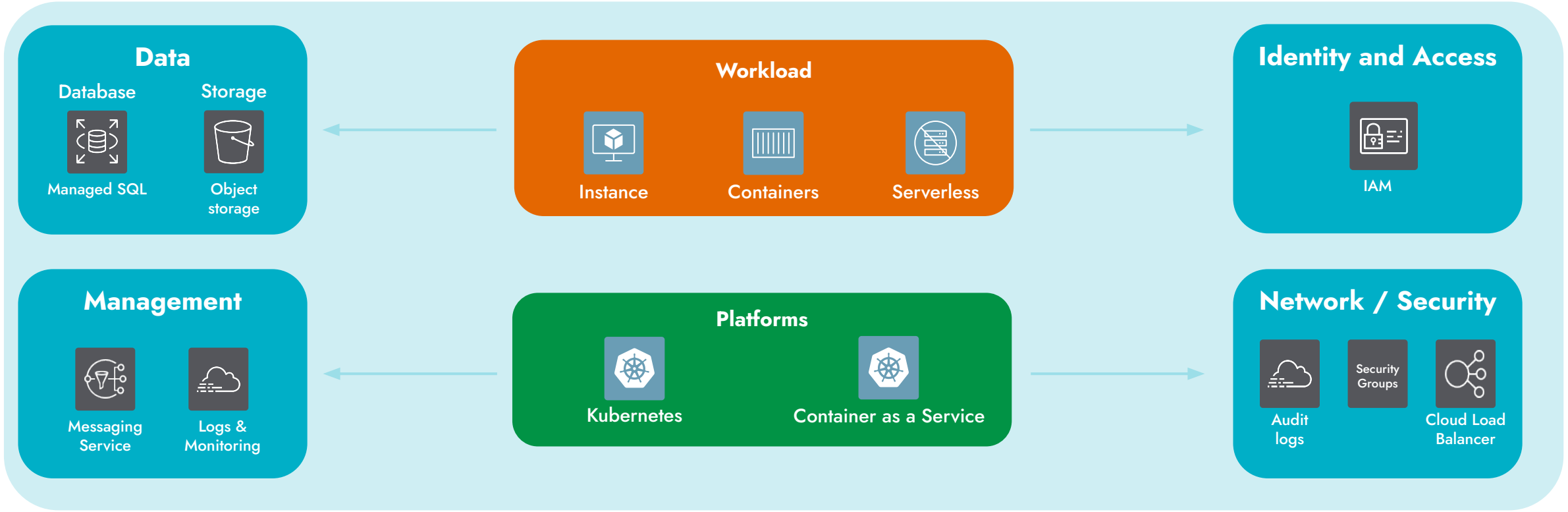
Cloud is exposed to the outside world

You need to control access to services your team uses

You need to detect unusual activity

# Cloud Native Application Architecture





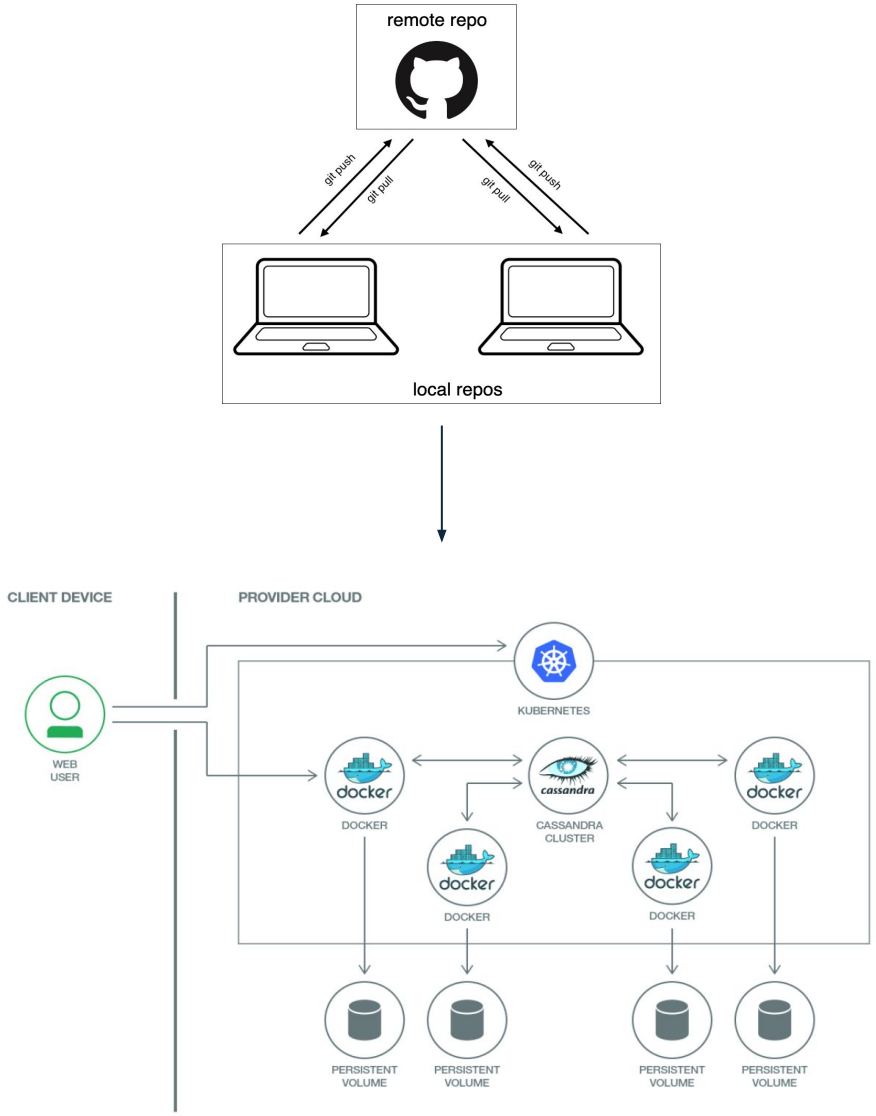
# Cloud Application Security Challenges



- Dynamic attack surface,
- Threat actors are using your tools today,
- Distributed systems and microservices enlarge attack surface,
- Number of calls generated by distributed systems,
- Lack of visibility,
- Cloud delivery vs security process speed.

# Manufacturing software in Cloud Native era



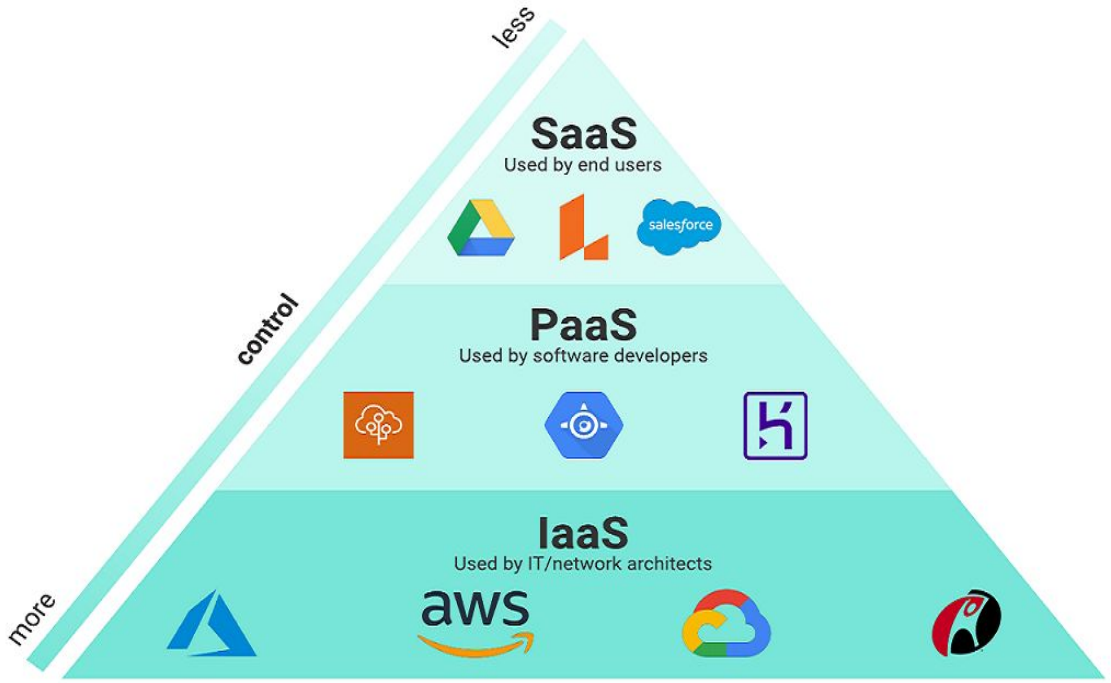



- Runtime architecture, CI/CD, DevOps, Environments, SecOps, Configuration Management, Version Management, Testing, Observability, Analytics, SRE
- Devops goes to canary, etc
- Self Service and Policy Driven
- Zero Trust environment

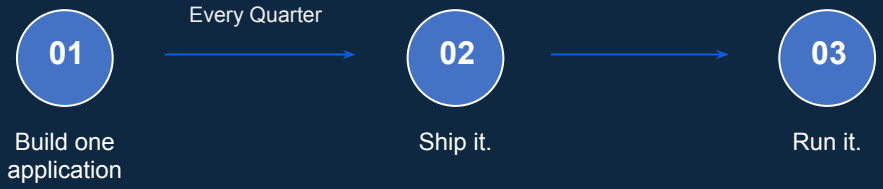
# The *Cloud Native* problem

## Microservices proliferation leads to a RELIABILITY challenge

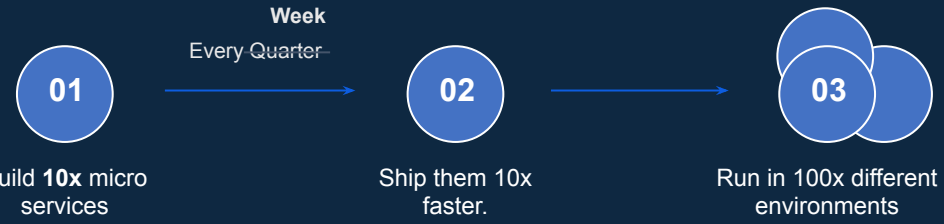
Cloud-native code's reliance on numerous microservices and platforms heightens failure risks.




### Legacy DevOps




### Cloud-Native DevOps



# What causes Downtime?



## Application Failures

- Excessive Logging to debug
- Too many retries
- Service Timeout

## Infrastructure Failures

- Device failures
- Network failures
- Region not available

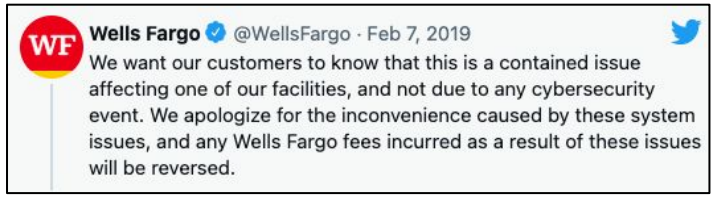
## Operational Failures

- Capacity issues
- Incident management
- Monitoring dashboards not available

## Reputational Impact

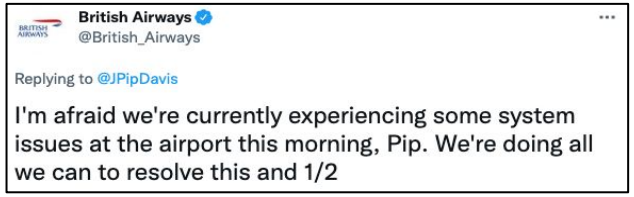

Slack's Outages

## Financial Impact




Est. >$55M in losses to WF

## Poor User Experience




75,000+ passengers travel plans impacted

CROWDSTRIKE

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete

For more information about this issue and possible fixes, visit https://www.windows.com/stopcode

If you call a support person, give them this info:

Stop code: CRITICAL_PROCESS_DIED

# Bad Actors Exploiting Vulnerabilities



# What is Chaos Engineering?



> **Chaos engineering is the process of testing a distributed computing system to ensure that it can withstand unexpected disruptions.**

— Tech Target (https://www.techtarget.com)

# What is Cyber Resilience?



> **Security Chaos Engineering (SCE) is a novel approach to cyber security; its core fundamentals are based on the principles of chaos engineering, though the objective is to enable cyber resiliency.**

— Mitigant (mitigant.io)

# Red Team strategies



## Pen Testing

- focus on a specific asset and have a defined scope that restricts the penetration tester.
- conducted periodically.

## Adversary Emulation

- Emulating specific threat actors/attack scenarios.
- focusing on specific attack vectors and techniques used by particular adversaries.

## Security Chaos Eng

- Introducing controlled security failures.
- observe how the system responds and recovers.
- Ongoing practice

# Why Security Chaos Engineering ?

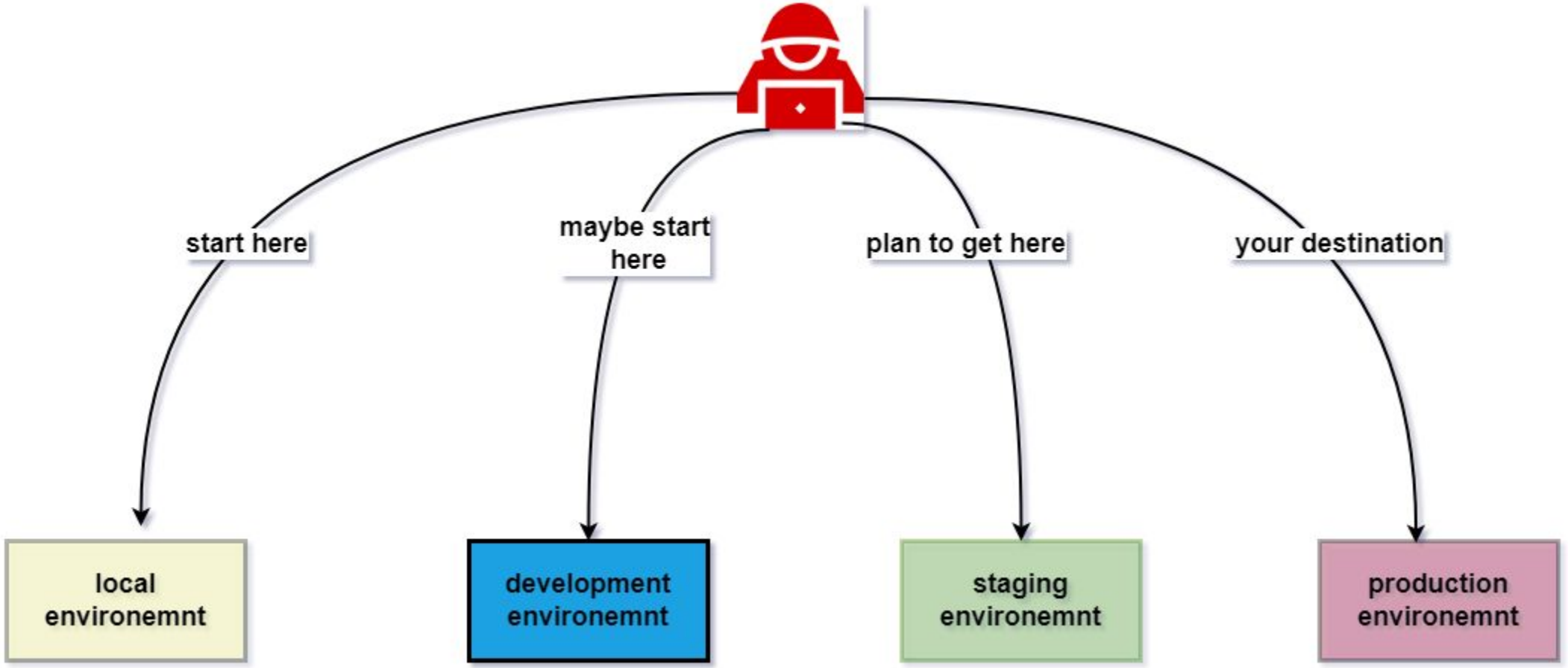


Security Chaos Engineering complements traditional security practices :

- Proactive approach,
- Integrated into ongoing security practices,
- Providing continuous feedback and improvement.

# Where to practise this?





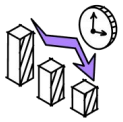
# Is Reliability a goal in Security?



It is not a direct goal usually, but Reliability of the end product or service is being affected while solving the other challenges.

Have you verified that the known resilience status is intact? No new bugs being leaked into the product?

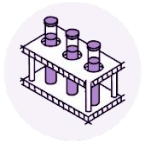

Are you sure you are not compromising the reliability?



How much of developer time is being spent on issues related to reliability?

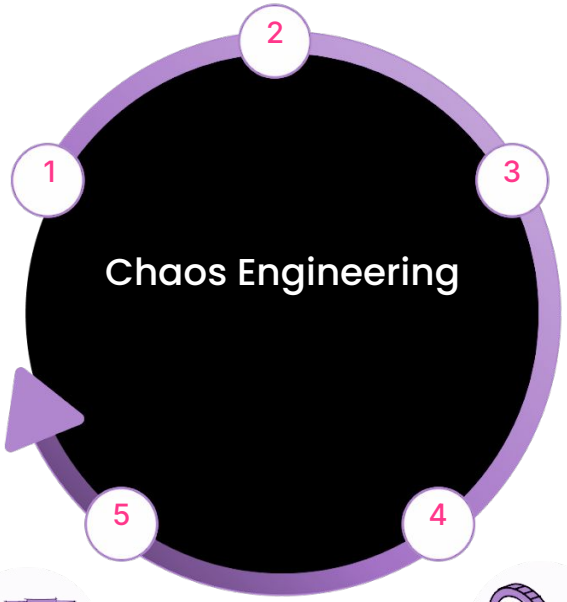
# The Chaos Engineering Process

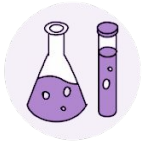




Select Chaos Experiments
Ex: Simulate Region Goes Down, etc

Select systems to test



Run Set of Chaos Experiments on Target System

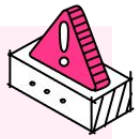
Use learnings to make targeted reliability improvements

Observe results of experiments on target system

# The Problems in current solutions



## Existing solutions



**Failures impacting resiliency is inevitable**

- Not proactively managed
- Downtimes maybe expensive

**Failure Scenarios are Difficult to Implement**

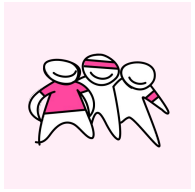
- Isn't implemented in a safe/controlled environment
- Isn't collaborative
- Not scalable

**Failure Testing isn't automated**

- Believed to be just for Ops
- Difficult to manage chaos in CI/CD
- No monitoring of impact

# A Better Solution





### Chaos engineering is *collaborative*

Collaborative chaos experiments in a centralized control plane

**SREs + Developers**
*Experiments are in Git just like code*

### Robust *Experiments*

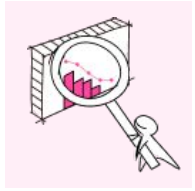
Public and private chaos hubs with ready to use experiments

**Optimize initial investment**
*Reduce the inertia for starting chaos*

### *Integrate* into CI/CD systems

Rollout automated and controlled chaos experiments across prod/non-prod environments

**Find weaknesses during build/test phase**
*Verifying at dev stage saves money*



### Enables *observability* for Chaos

Chaos metrics used to assess impact and manage SLOs/Errors

**Measure the impact of inducing chaos**
*Build confidence by starting small*

# Is it really a better solution?



**Gaining Kernel Level Visibility**

Kernel-level visibility helps detecting sophisticated threats that traditional security approaches might miss

**Comprehensive Security Coverage**

Ensures comprehensive security coverage addresses potential blind spots in the current chaos engineering framework.

***Real Time Threat Detection***

Enables faster response to potential security incidents.

**Customisable Rules and Policies**

Flexibility in creating customizable rules and policies tailored to specific security needs and threat models.

# Potential Tools



**Litmus Chaos** is an Open Source Cloud-Native Chaos Engineering Framework with cross-cloud support. It is a CNCF Incubating project with adoption across several organizations.

https://litmuschaos.io

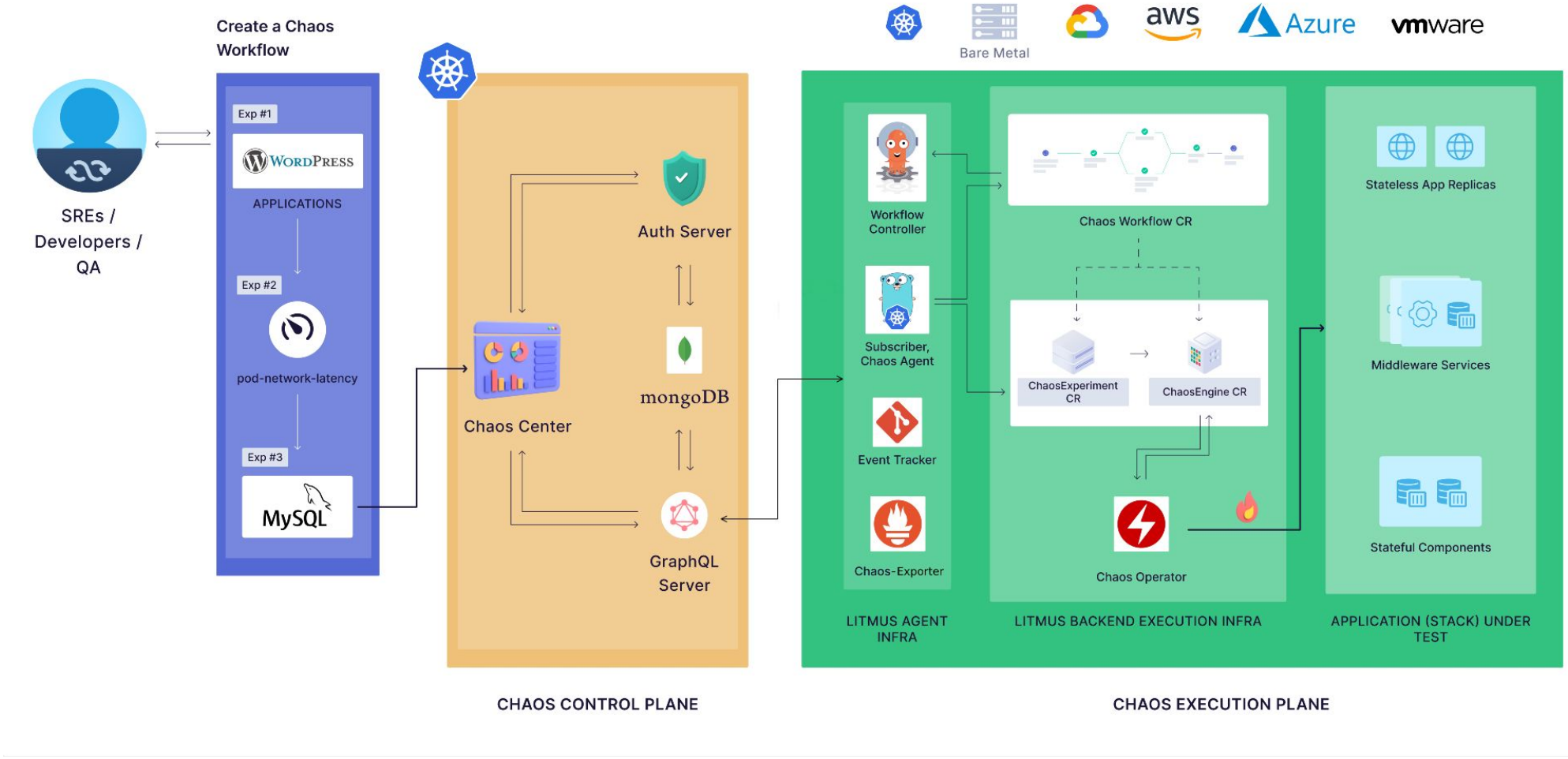
falco

**Falco** is a cloud-native security tool designed for Linux systems. It employs custom rules on kernel events, which are enriched with container and Kubernetes metadata, to provide real-time alerts

https://falco.org

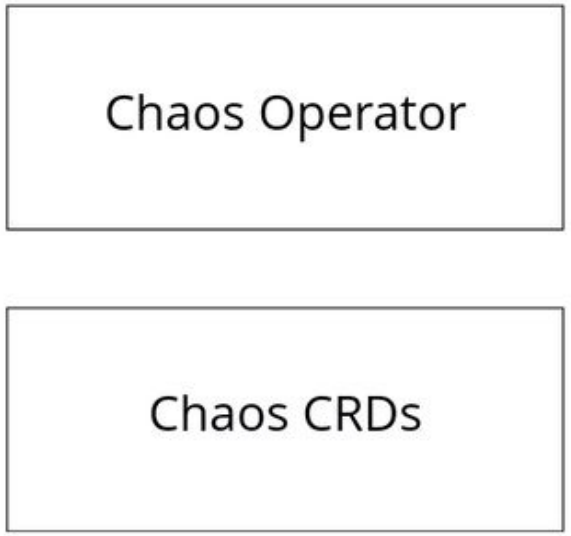
# How does Litmus work?





# How does Litmus work?





# Threat Detection with Falco



**Falco** is an **open source** runtime security solution for **threat detection** across **Kubernetes**, containers, hosts and **the cloud**.

CLOUD NATIVE COMPUTING FOUNDATION

**CNCF Graduated Project**
(Feb. 2024)

**7.2k**

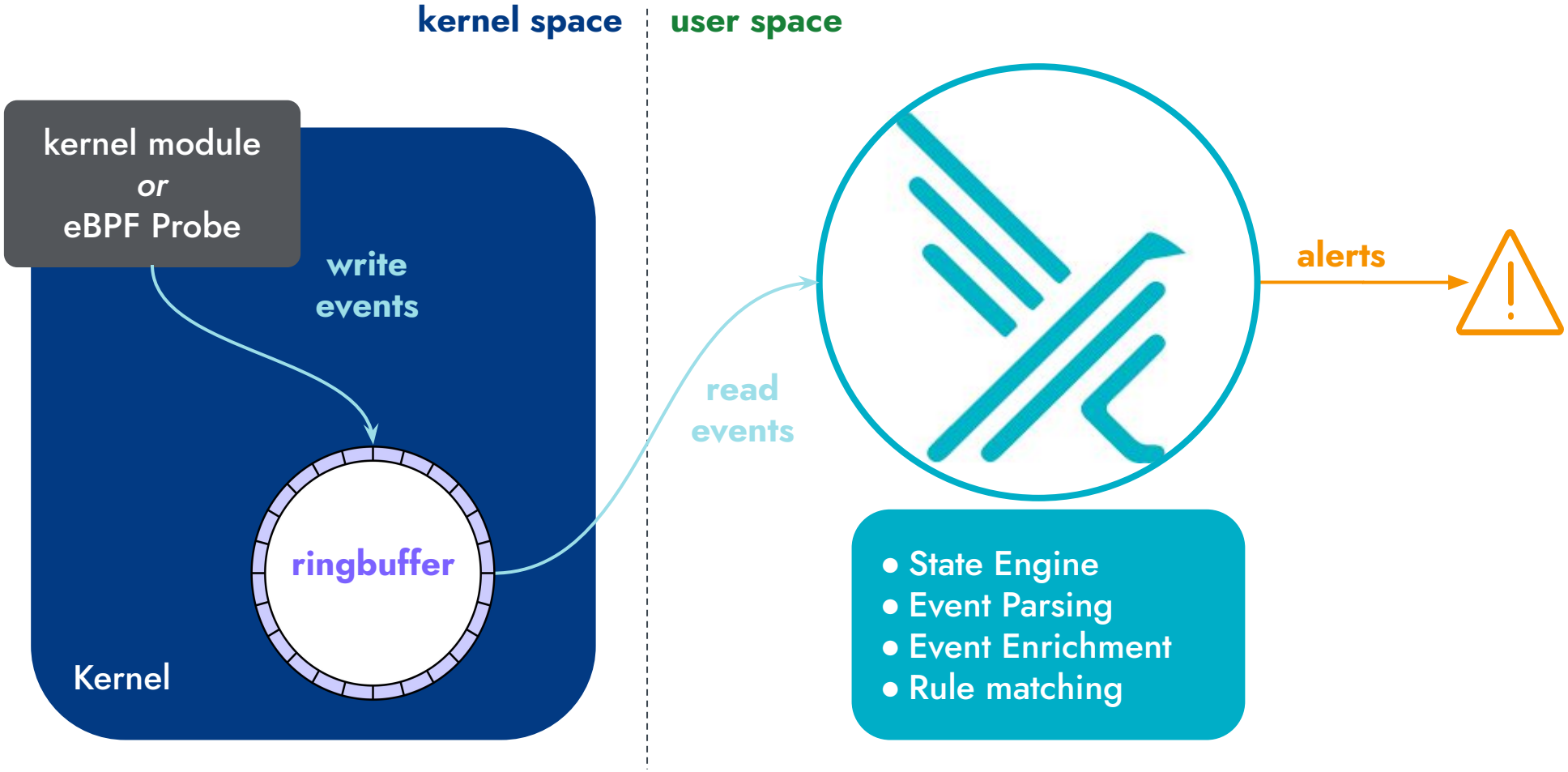
**50M+ pulls**

**https://falco.org**

**https://github.com/falcosecurity/falco**

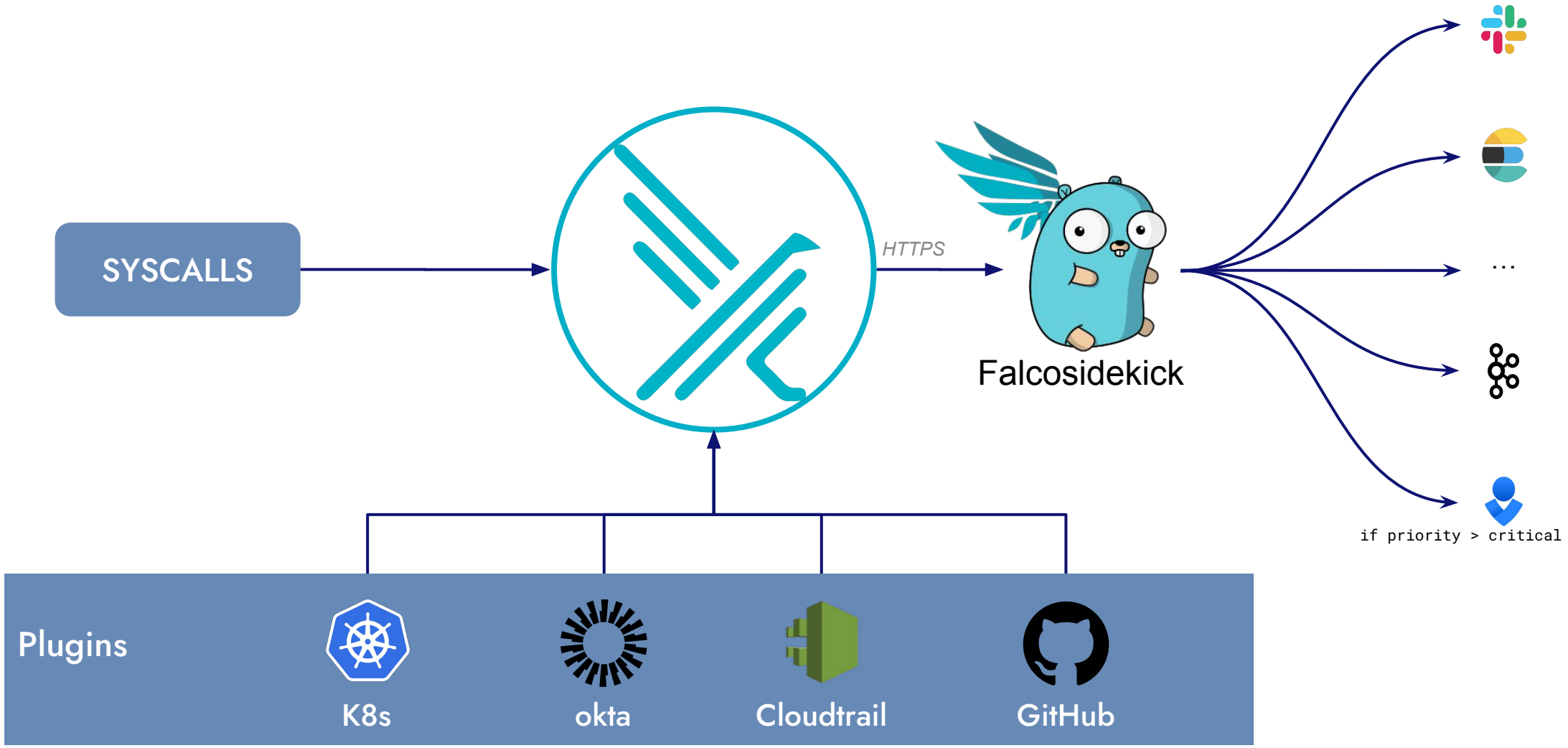
# Falco Architecture Overview





# Falco Ecosystem





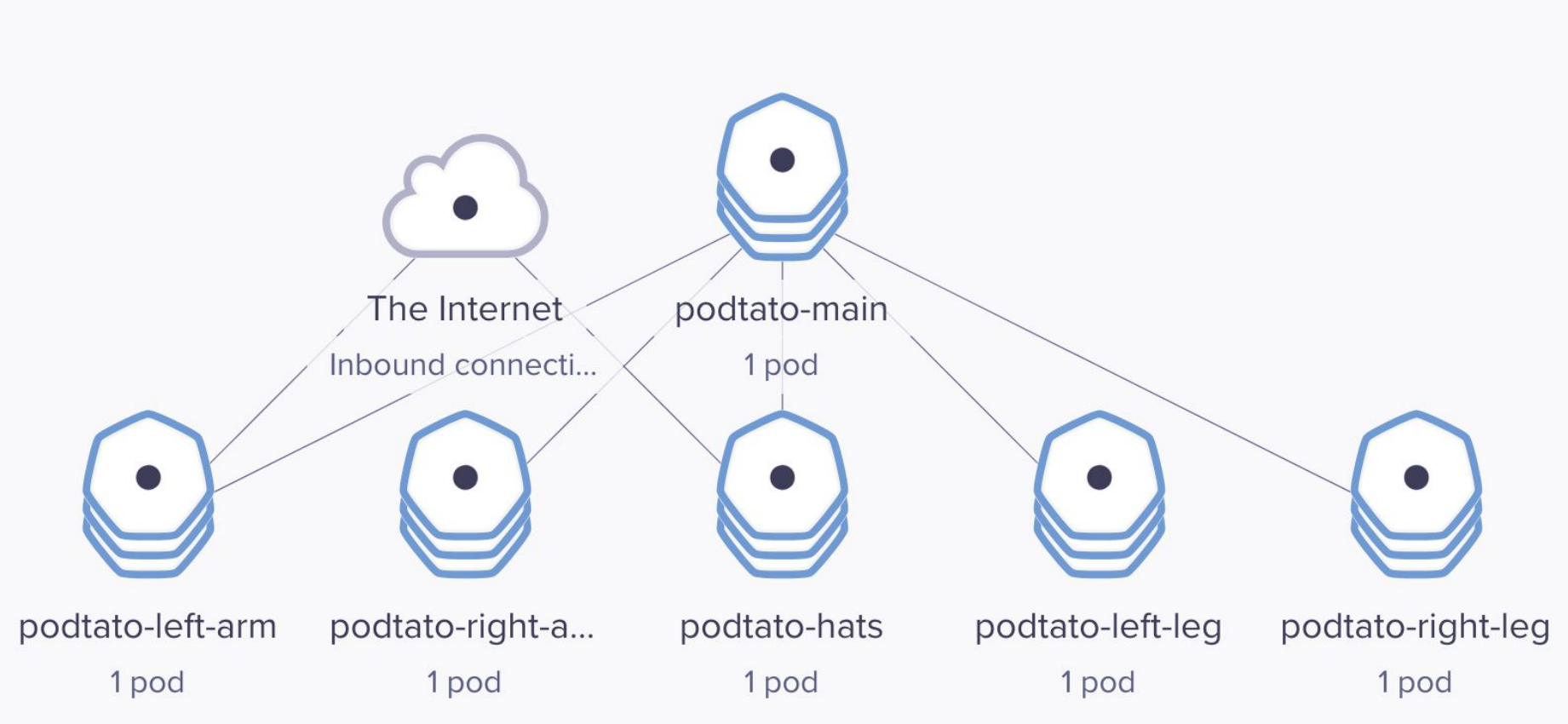
# Hands on Demo

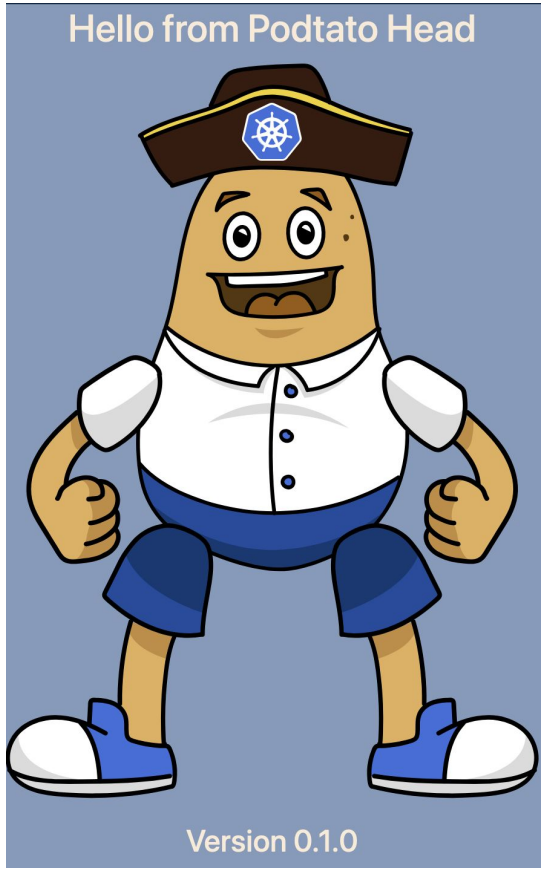
https://github.com/S-ayanide/kubecon-china-2024-cyber-resilience-falco-litmuschaos

https://tinyurl.com/kubecon-china-24

# Podtato Demo Application



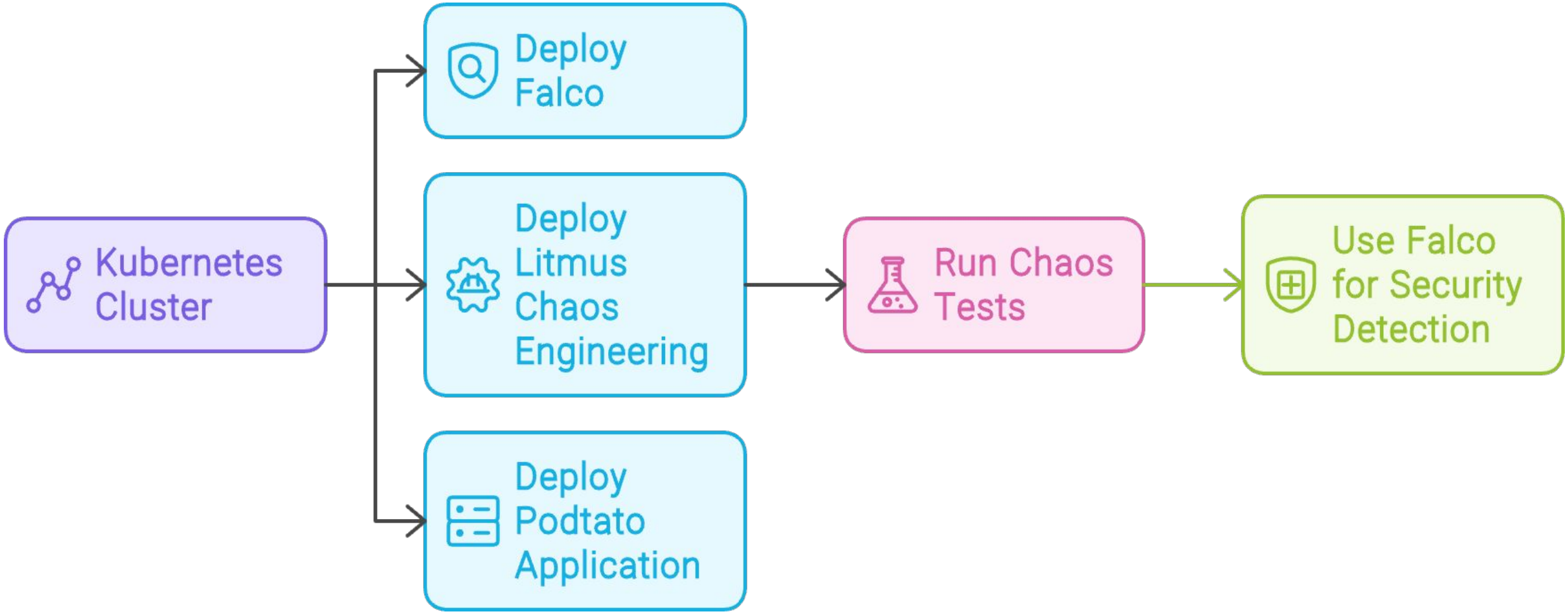



https://github.com/podtato-head/podtato-head/tree/main

# Chaos Engineering in practice

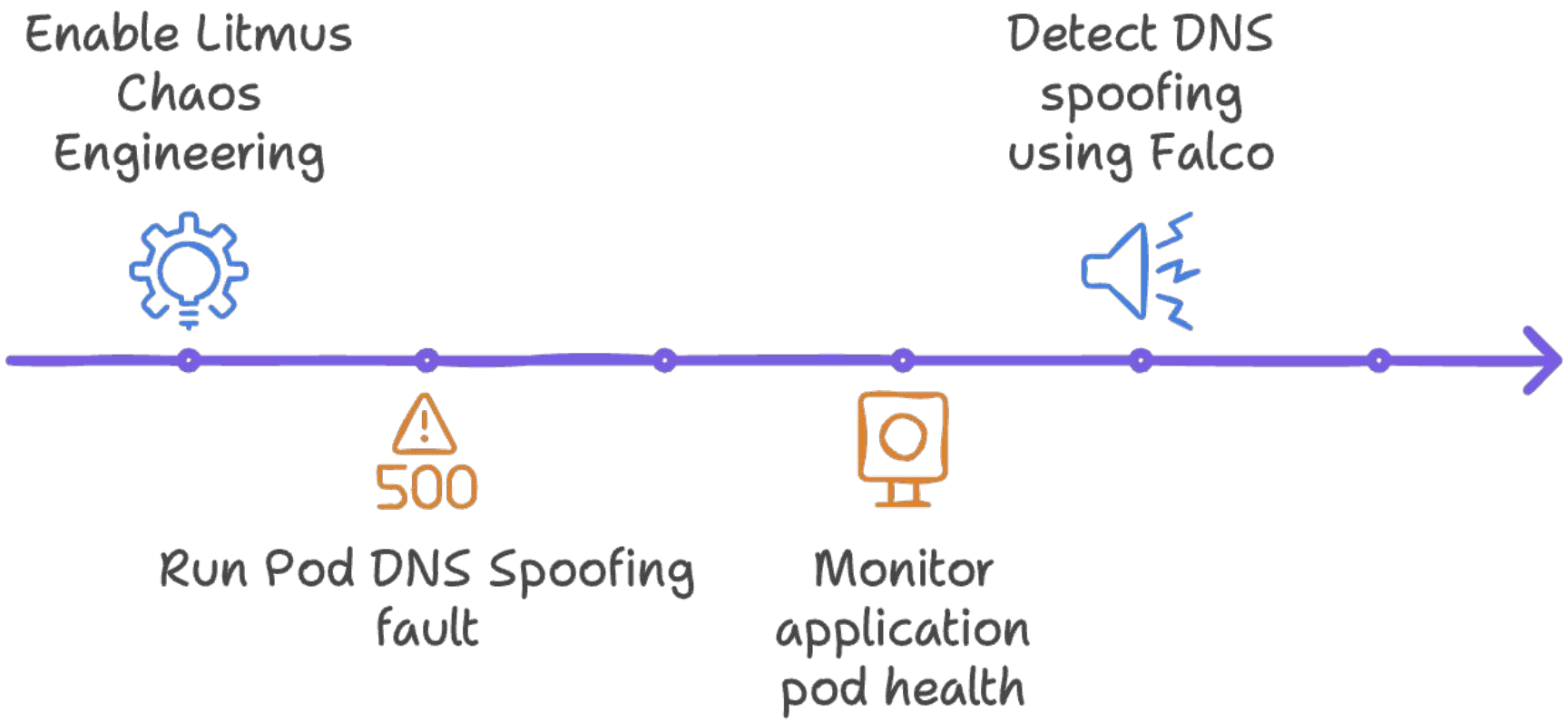




# Scenario 2 - DNS spoofing



## Run Litmus Chaos Engineering to perform DNS spoofing of Kubernetes Pods



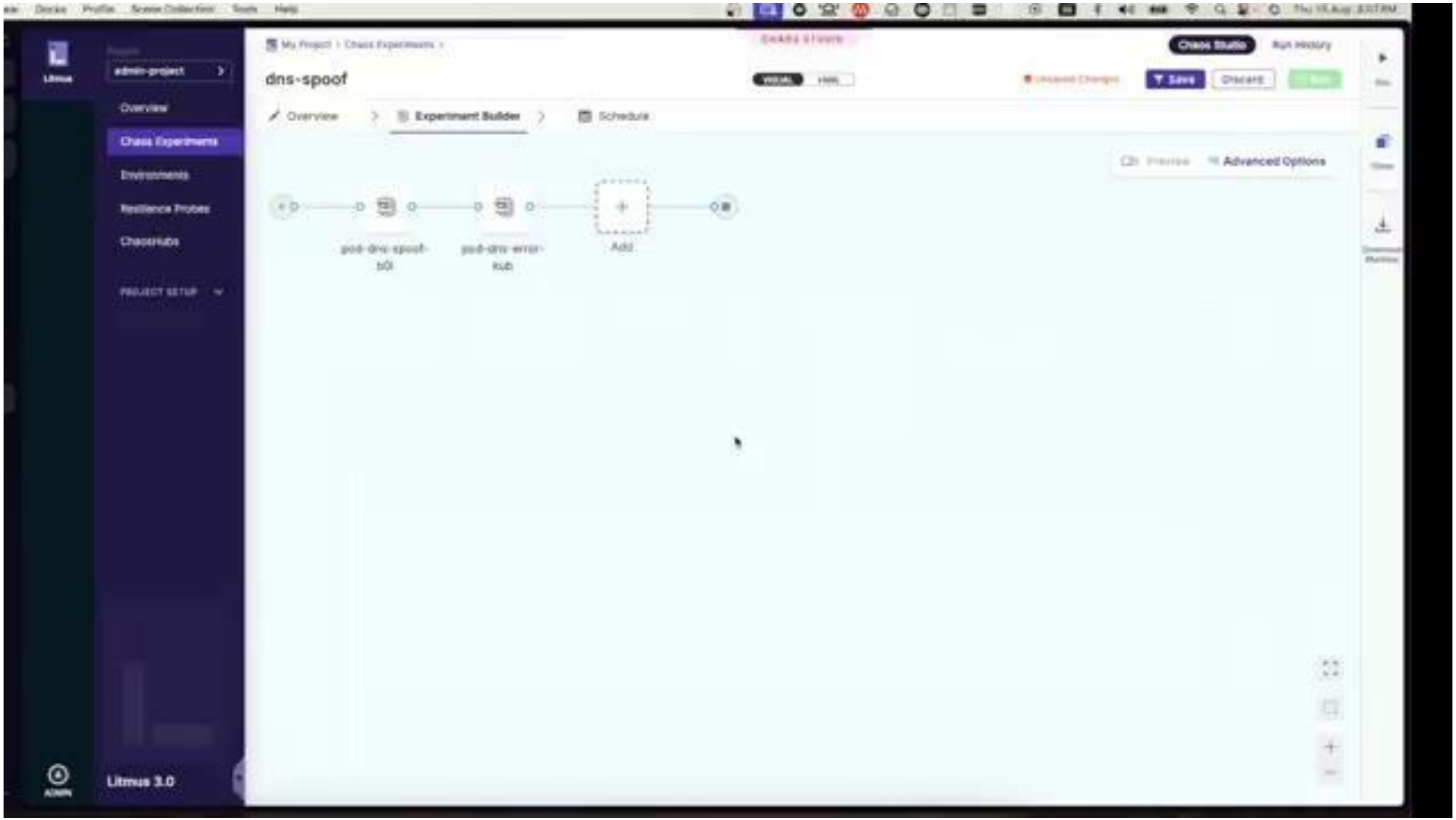
# Scenario 2 - Falco Detection Rule



```
- rule: Detect Access to resolv.conf by Root User
  desc: Detect access to /etc/resolv.conf only when performed by root user
  condition: >
    container and
    fd.name = /etc/resolv.conf and
    user.name = "root"
  output: "[KubeCon] Access to /etc/resolv.conf detected by root user (container=%container.name
user=%user.name command=%proc.cmdline)"
  priority: WARNING
  tags: [filesystem, container]
```

# Scenario 2 - Video Demo

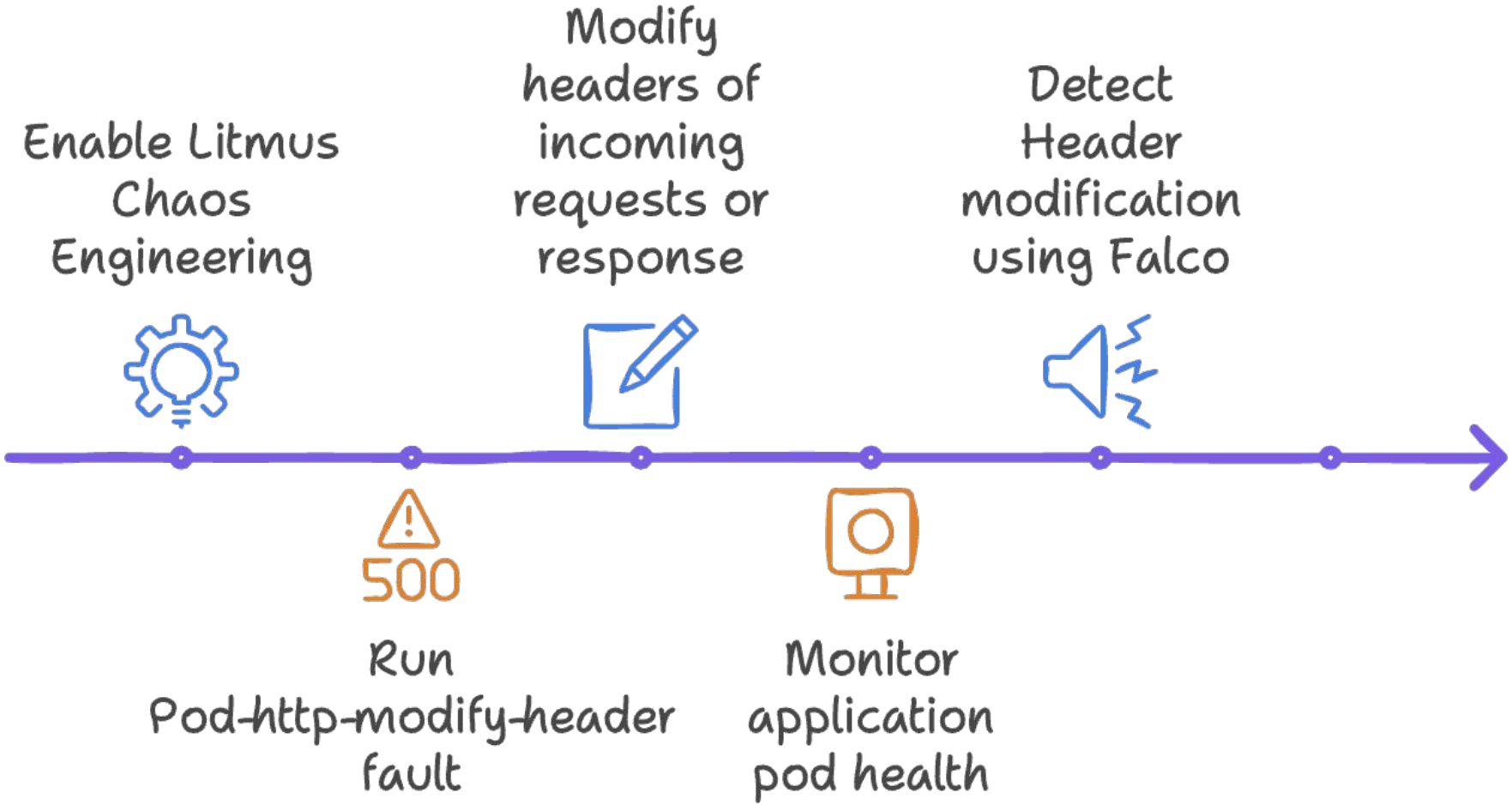




# Scenario 1 - Modify HTTP header



## Run Litmus Chaos Engineering to Modify HTTP Headers of Kubernetes Pods



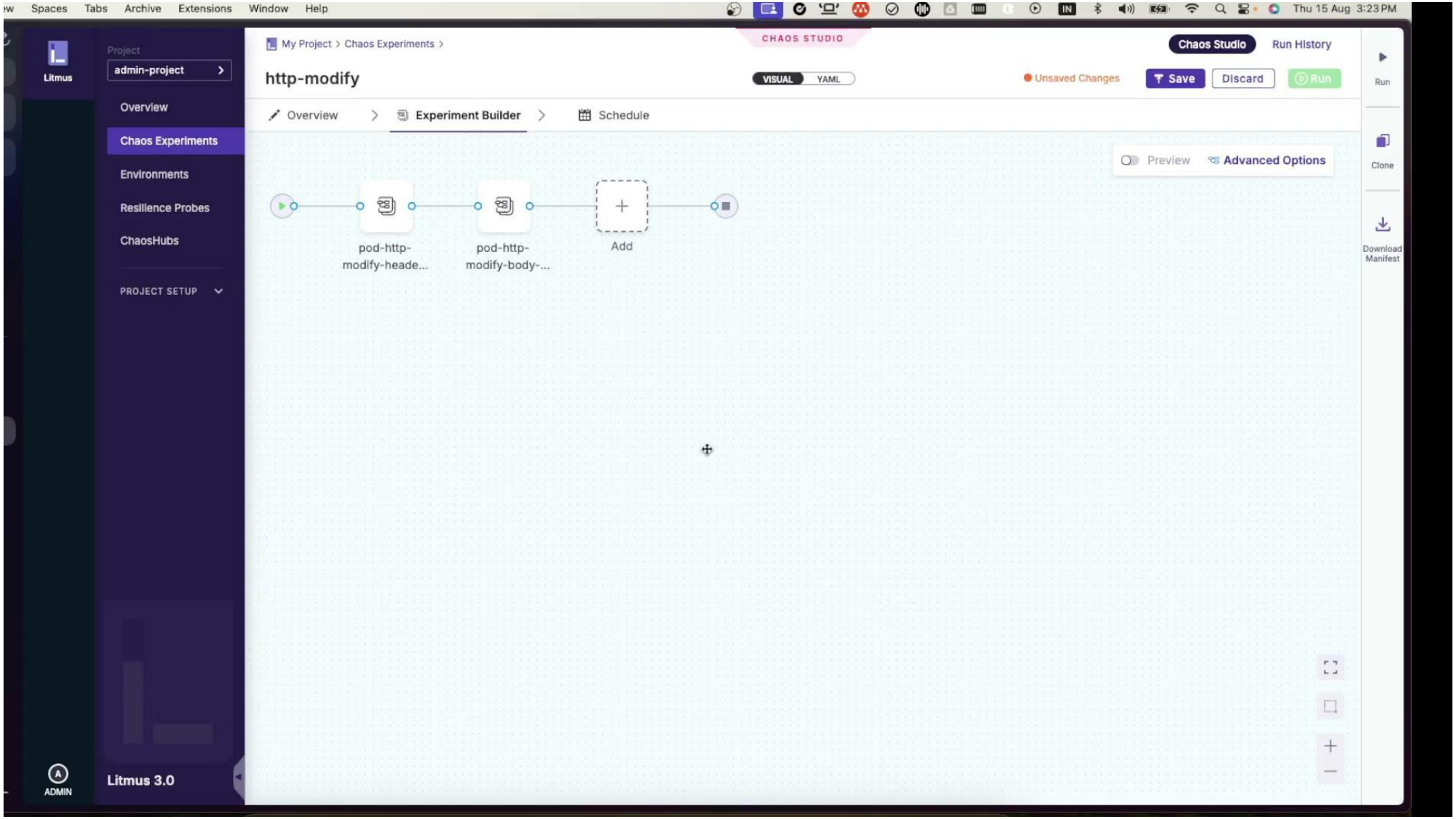
# Scenario 1 - Falco Detection Rule



```
- rule: Detect Header Detection in Log File
  desc: Detects when the specific text "Header X-Litmus-Test-Header detected" is logged
  condition: >
    container and
    fd.name = /tmp/http_header_check.log and
    evt.type in (open, read, write) and
    (proc.cmdline contains "cat" or proc.cmdline contains "grep" or proc.cmdline contains "tail")
  output: "[KubeCon] [HTTP Headers Modify] Detected log entry indicating HTTP header detection
(container=%container.name user=%user.name command=%proc.cmdline file=%fd.name)"
  priority: WARNING
  tags: [filesystem, http]
```

# Scenario 1 - Video Demo





# Go to the next level



- Red Teaming
- Cross functional collaboration
- Enhance Automation using GitOps
- Introduce feedback loop
- Advanced Metrics
- Community and ecosystem shift

# Takeaway



- Cloud-native systems exceed traditional security.
- Cyber-criminals exploit advancements in cloud.
- Learned about Zero Trust Chaos.
- Discovered vulnerabilities with chaos experiments.
- Enhanced detection and response capabilities.
- Gained actionable Zero Trust strategies.

# Further Reading



- Increased support for chaos against Non-Kubernetes infrastructure components
- More Application specific chaos experiments with native faults and health checks
- Improved Chaos SDK for creation of user-defined experiments
- Additional probe types for diverse steady state-hypothesis validation
- Improved Observability for chaos experiments
- More community supported Chaos Types
- Falco training: https://falco.org/training
- Litmus training: https://v2-docs.litmuschaos.io/tutorials

KubeCon | CloudNativeCon | THE LINUX FOUNDATION OPEN SOURCE SUMMIT | AI_dev Open Source GenAI & ML Summit

China 2024

# Thank You

## Scan QR for Feedback


**Contact Sayan on**

@s_ayanide

/s-ayanide

**Contact Rafik on**

@rafik8_

/rafikharabi


Leave us a feedback