AWS Fault Injection Simulator를 활용한 EKS 안정성 검증하기

정영진
DevOps Engineer
LG U+

“Good intentions never work, you need good mechanisms to make anything happen”

Jeff Bezos

CEO of Amazon

Agenda

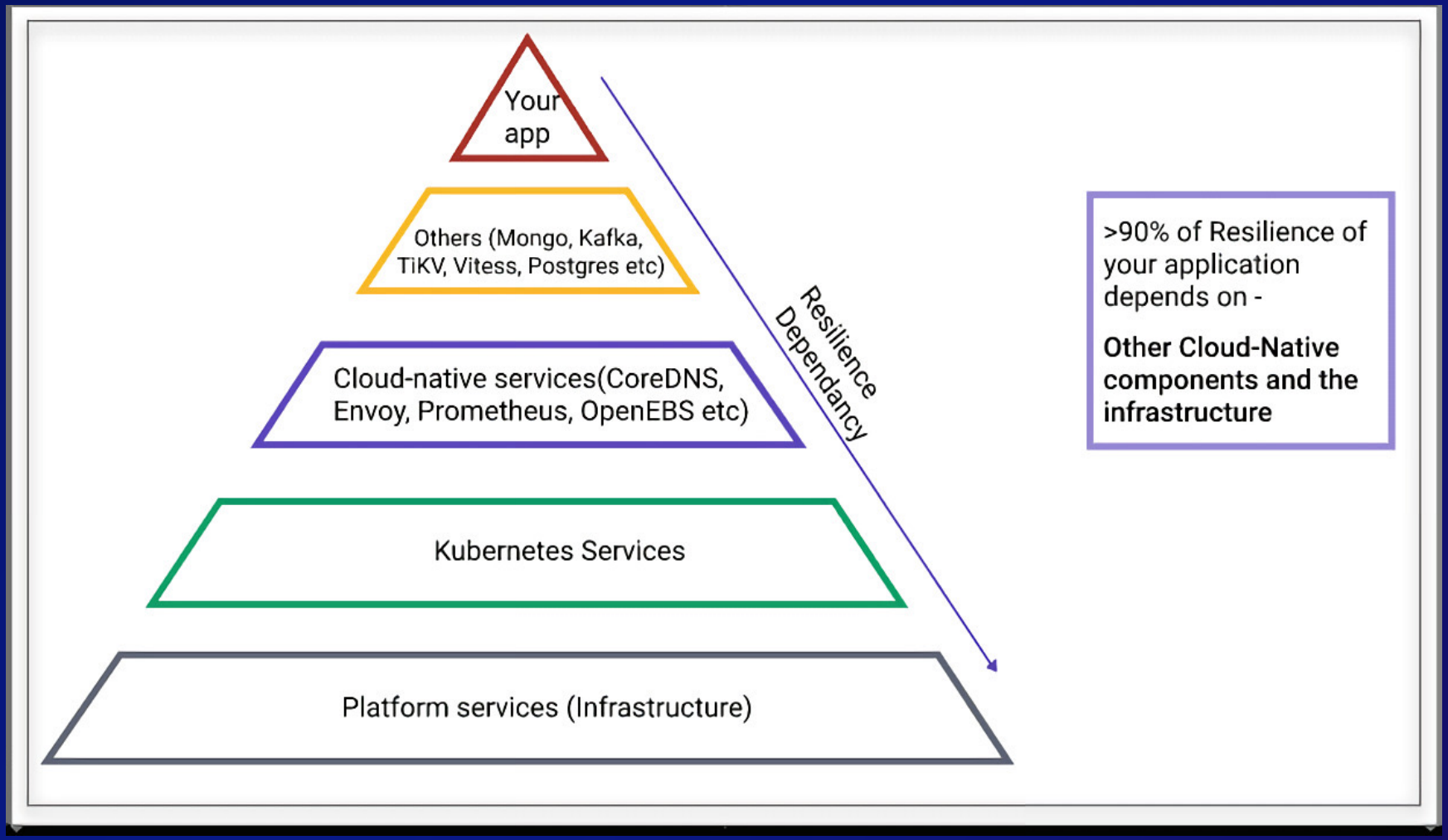
지속적인 서비스를 위한 테스트

AWS Fault Injection Simulator과 Chaos Mesh

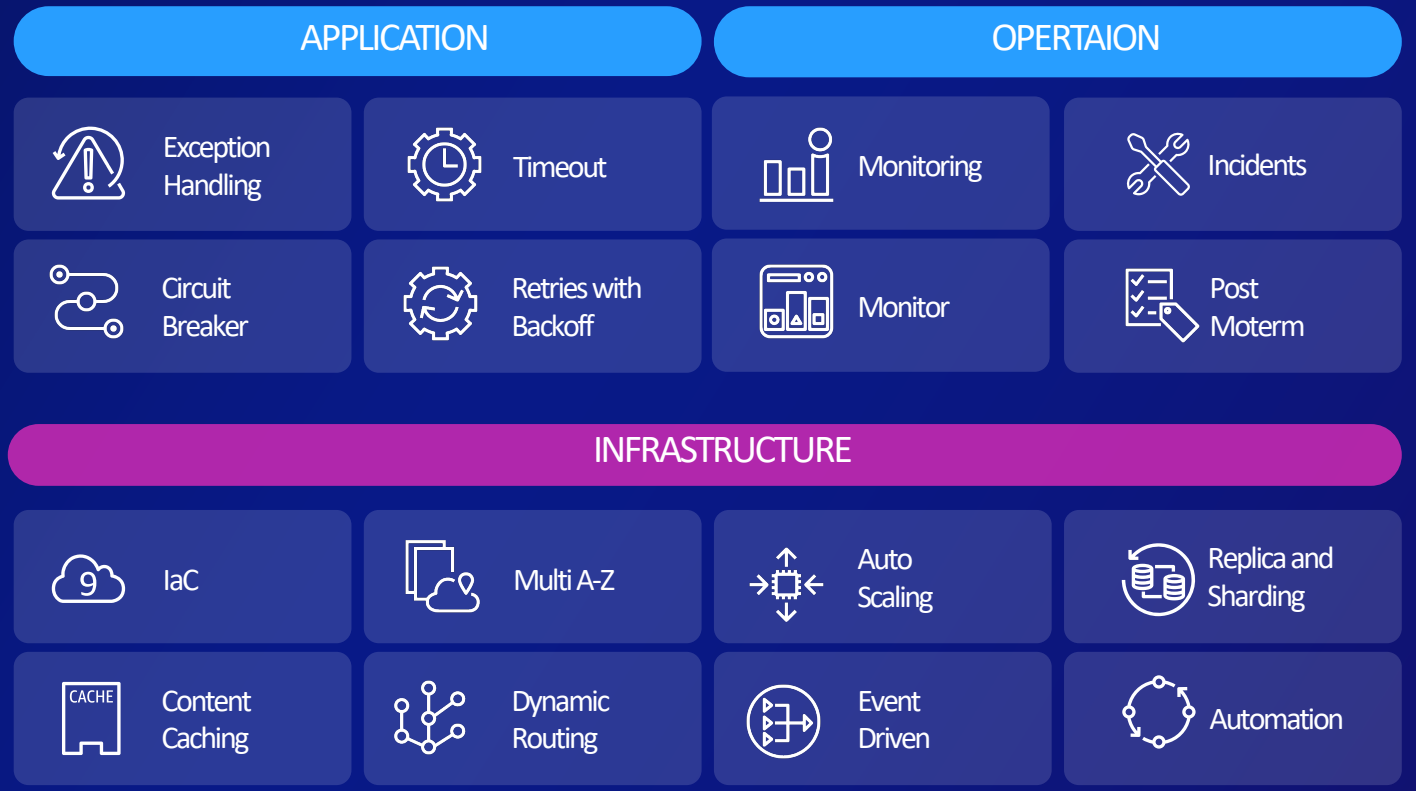
EKS 안정성 검증하기

지속적인 서비스를 위한 테스트

탄력성의 중요성



우리가 장애를 겪지 않기 위해선...



카오스 엔지니어링은 운영
환경에서도 갑작스러운
장애를 견딜 수 있는 시스템을
구축하기 위해 시스템을
실험하는 분야

WHAT

예상치 못한 상황에서의
문제점 발견, 통제된
환경에서의 인프라와
어플리케이션의 복원력, 탄력성

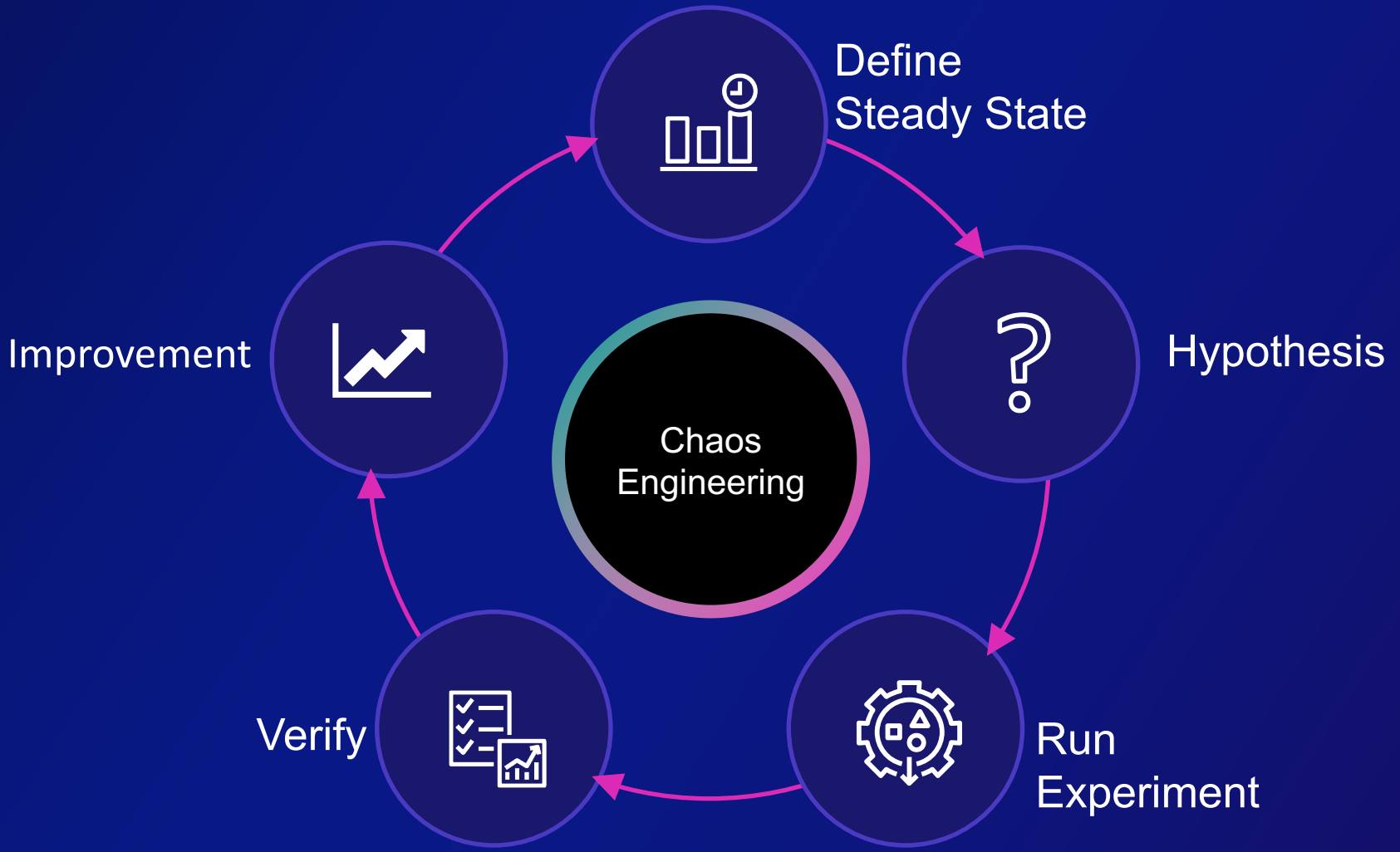
HOW

어플리케이션이 실행되는
환경(OS, 네트워크, 스토리지)에
잘못된 값이나 오류를 주입

WHY

어플리케이션을 검증, 주어진
환경에서 SPOF를 찾아내고
인프라/어플리케이션이 장애를
처리할 만큼 견고한지 확인

카오스 엔지니어링 과정



Observability의 중요성

Visibility
데이터를 통한
인프라와
어플리케이션 가시성

Alerting
이상 동작 혹은 문제
발생 시, 개발자와
운영자에게 알람

Less time in Incident
장애상황 발생 시,
빠른 트러블 슈팅

Acceleration
개발자들의
생산성 향상

AWS Fault Injection Simulator과 Chaos Mesh

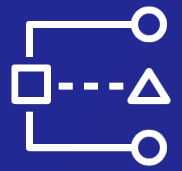
AWS Fault Injection Simulator

완전 관리형 카오스 엔지니어링 서비스

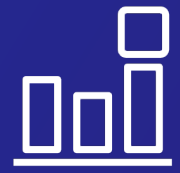
01.
Simple Setup

02.
Run Real World Scenarios

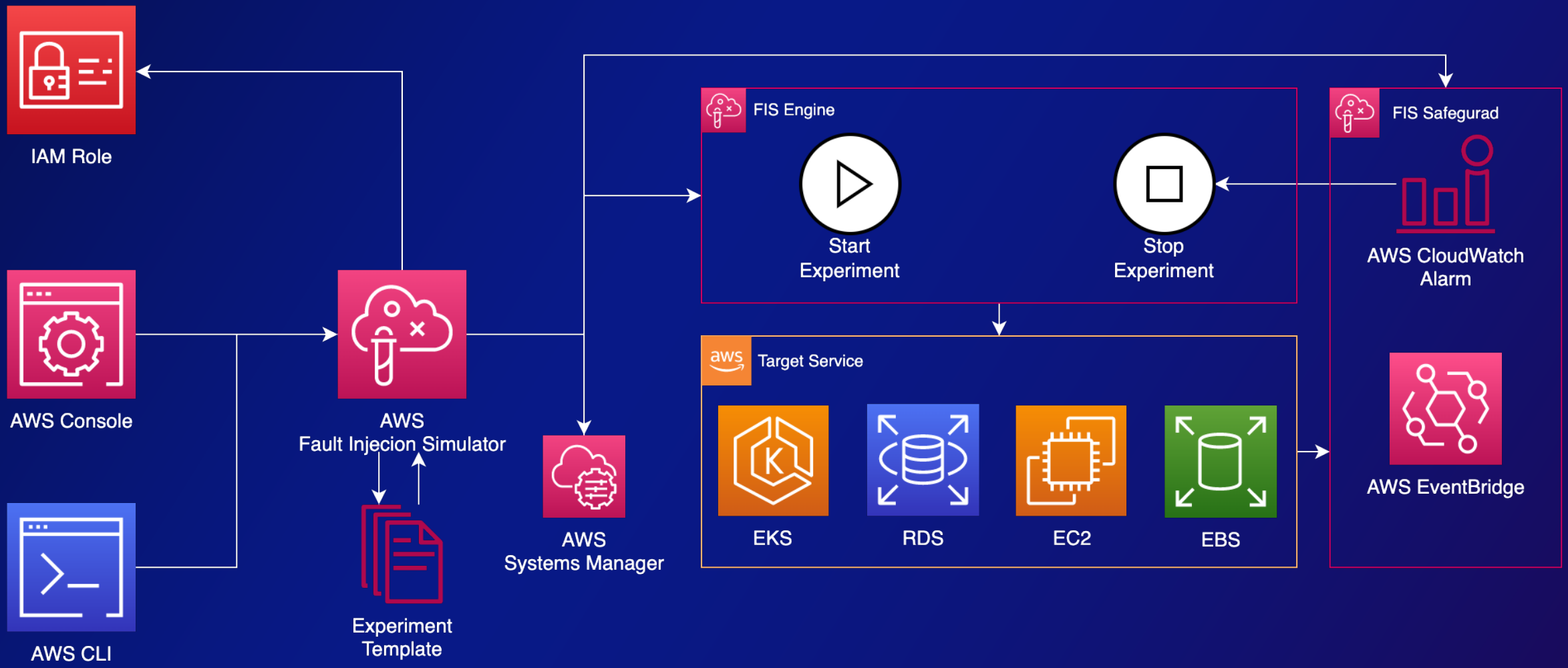
03.
Integrated Security Model



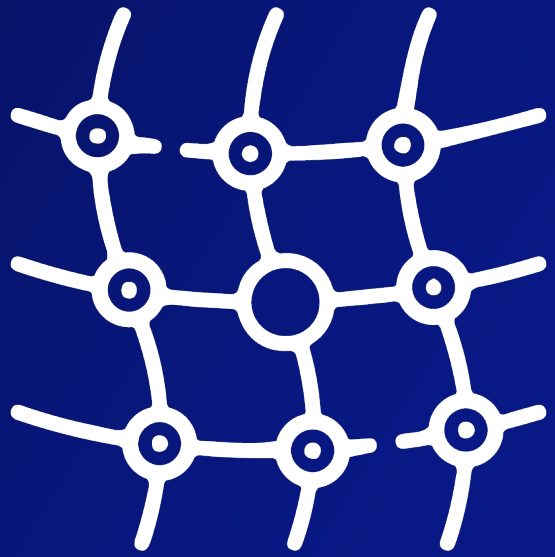
04.
Visibility throughout an Experiment



AWS FIS 동작 원리



Chaos Mesh란?



Chaos Mesh

Cloud Native and easy-to-use system

Kubernetes 기반의 사용 환경과 대시보드로 쉽게 사용 가능

Flexible experiment orchestration

여러 실험을 혼합하거나 어플리케이션 상태 확인까지 할 수 있는 고유한 실험 시나리오 설계 가능

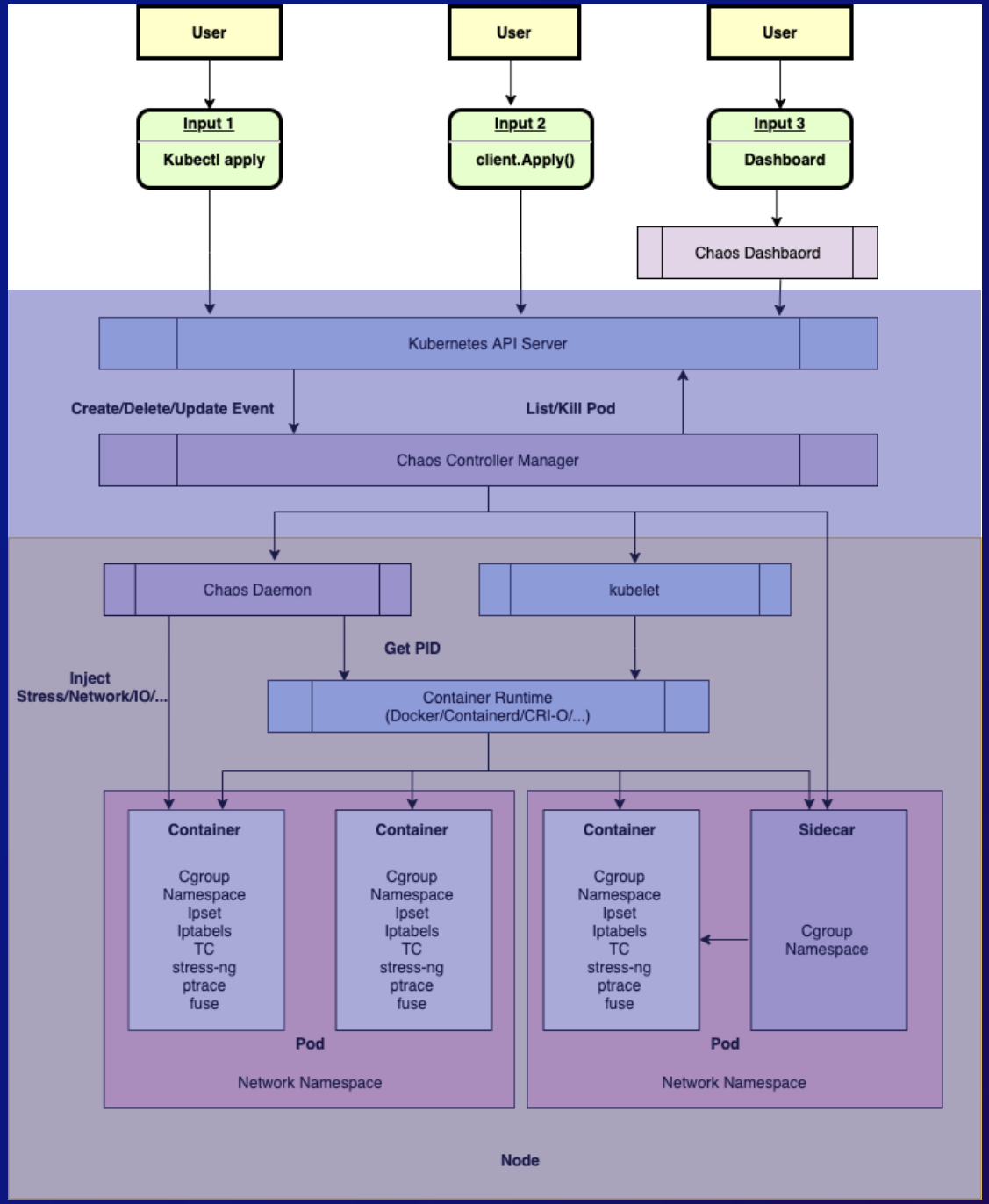
High security and fully authenticated

여러 계층에서의 보안으로 안전하고, 많은 분산 시스템에서 사용

Chaos Mesh 동작

Input Fault From User

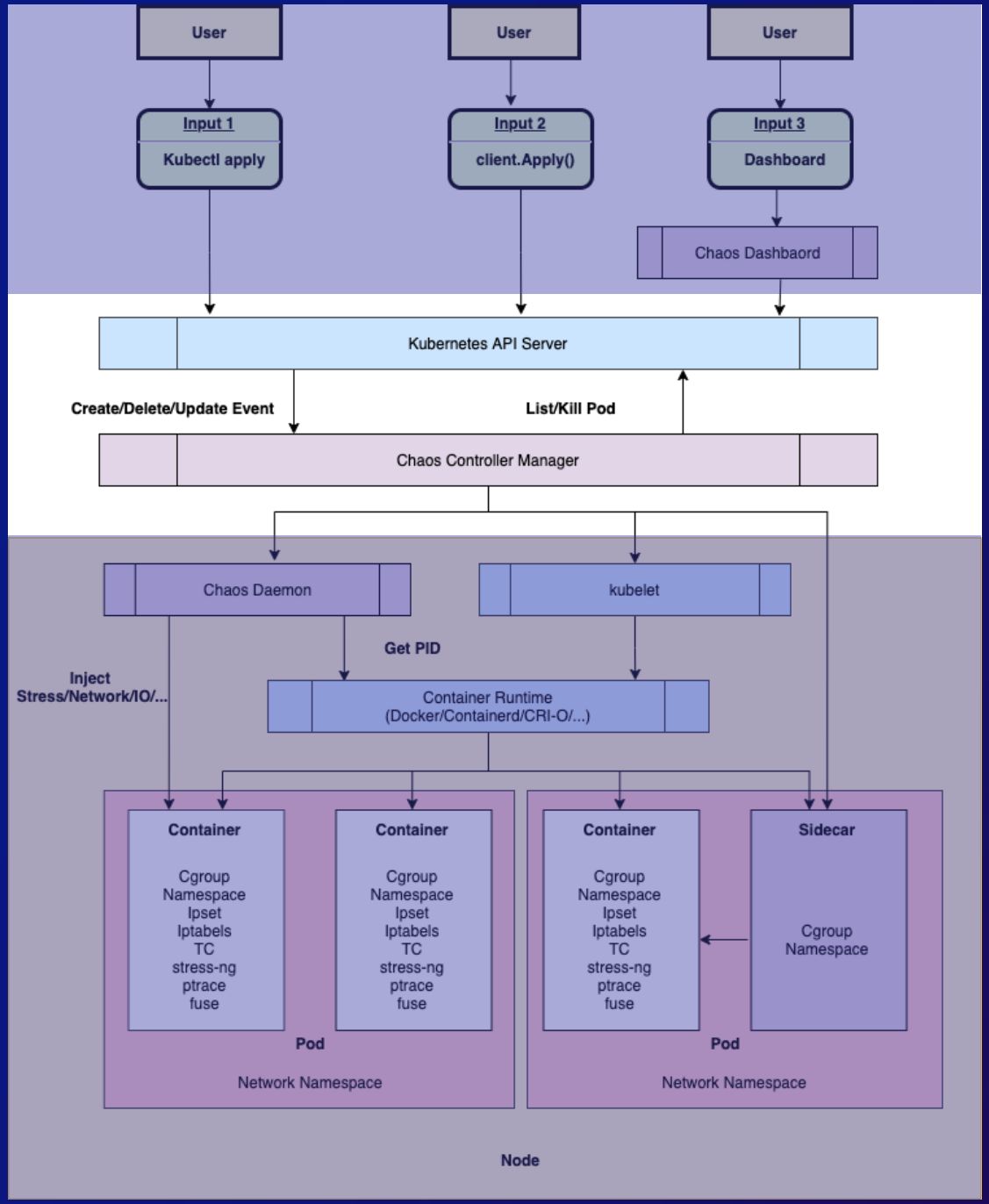
- kubectl
- API
- Dashboard



Chaos Mesh 동작

Monitor Resource and Schedule

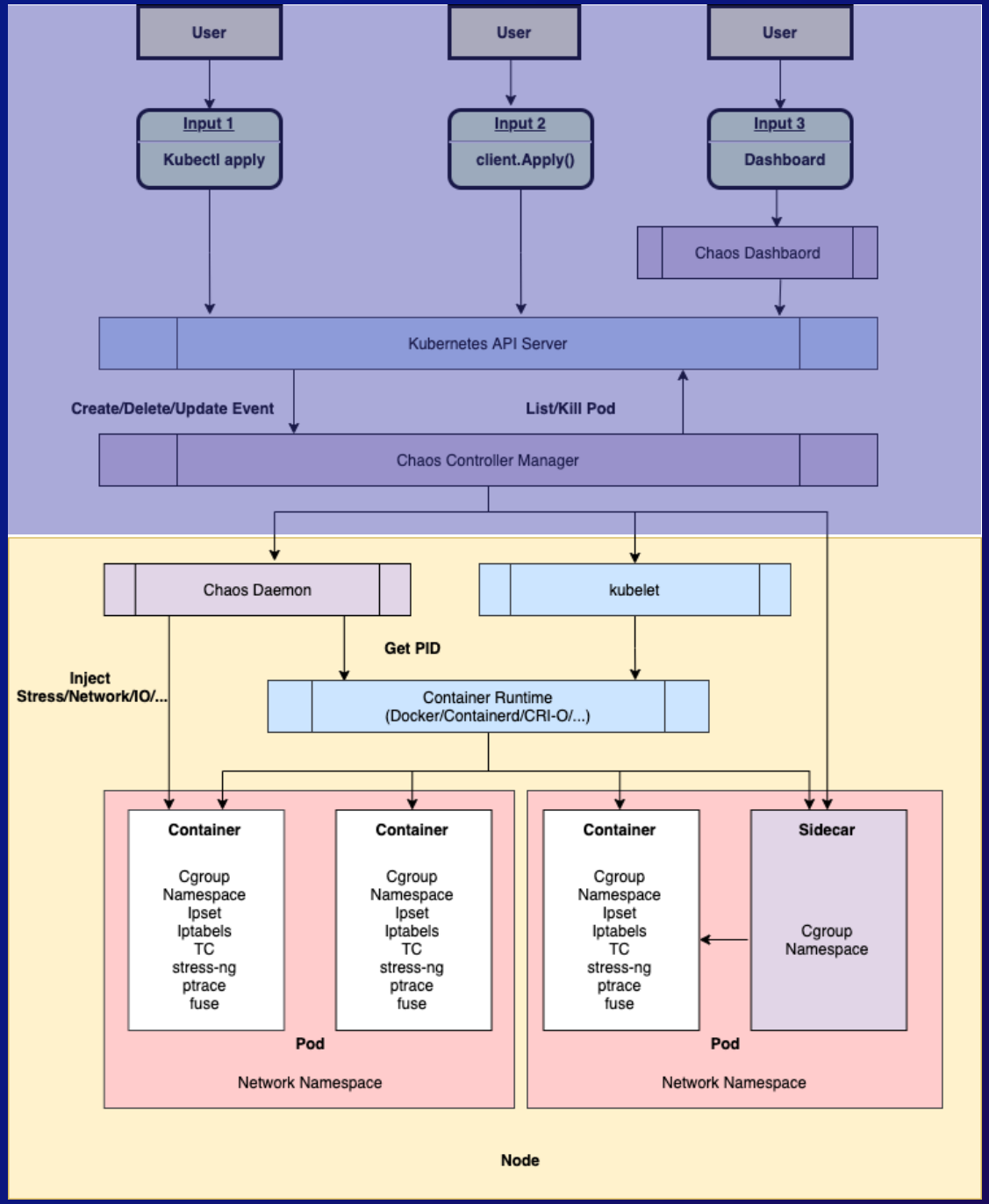
- Kubernetes API 서버로부터 요청 전달



Chaos Mesh 동작

Injection of a specific fault

- Chaos Controller Manager로부터 요청 받음
- 요청 받은 장애를 대상에 주입



AWS FIS

Experiment templates
Experiments

Spotlight

Share Feedback

You successfully deleted 1 experiment templates.

AWS FIS > Experiment templates

Experiment templates (5) Info

Last updated on April 05, 2023, 15:05:33 (UTC+09:00)

Actions | Start experiment | Create experiment template

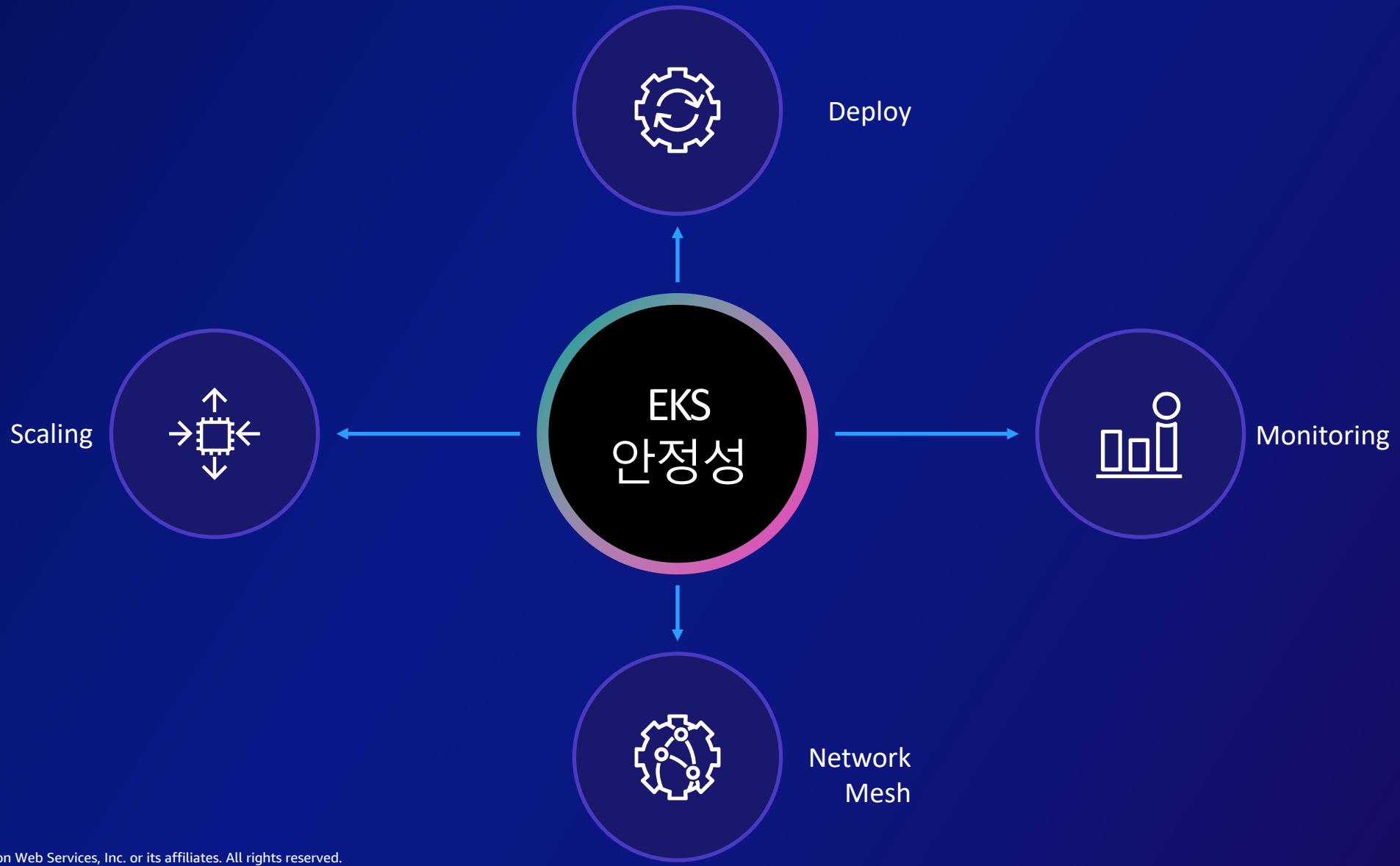
Filter experiment templates

1

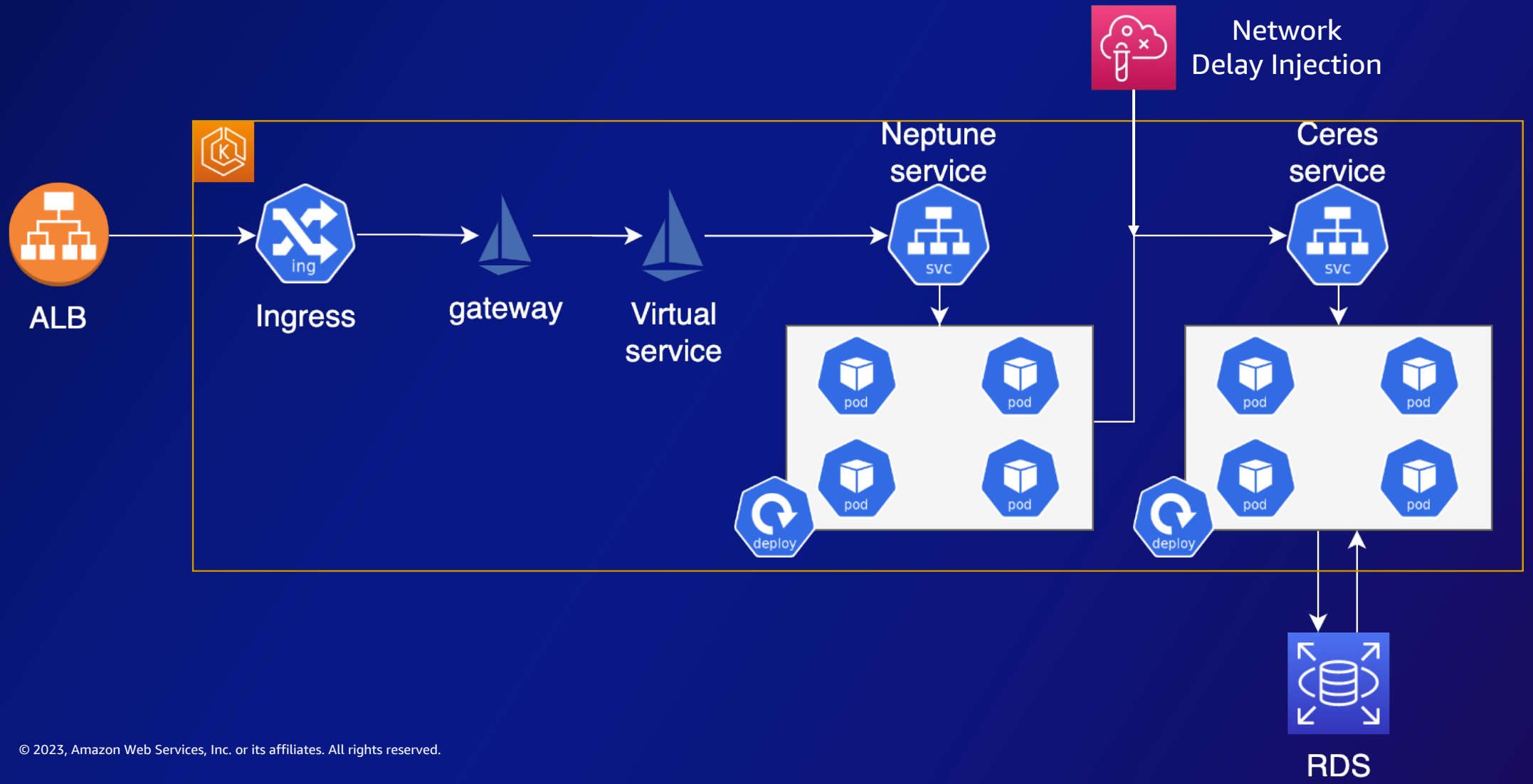
Name	Experiment template ID	Description	Creation time	Last update time
	EXT3tzADbTYWePuYC	daxdapne2 nodegroup termination experiment	March 27, 2023, 13:47:29 (UTC+09:00)	March 27, 2023, 13:47:29 (
	EXT3ySZEgLHbZjpL	Chaos Mesh Stress CPU on Ceres Pods	March 27, 2023, 14:56:59 (UTC+09:00)	March 30, 2023, 13:24:10 (
	EXT4kZkJvKN5Pohzg	karpenter node termination	March 20, 2023, 09:38:01 (UTC+09:00)	March 20, 2023, 09:38:01 (
	EXT6iSggBi5j7rYpx	Chaos Mesh Stress Memory on Ceres Pods	March 27, 2023, 14:57:01 (UTC+09:00)	March 27, 2023, 14:57:01 (
	EXTXJq89hd2M5tt	Chaos Mesh Network Delay	March 20, 2023, 15:04:15 (UTC+09:00)	March 29, 2023, 15:53:41 (

EKS 안정성 검증하기

EKS 안정성을 높이기 위한 구성



Network Chaos 검증



Network Chaos 검증

01	02	03	04	05
				
Steady State	Hypothesis	Run Exp.	Verify	Improve
응답 시간 평균 100ms	Network Delay를 주입 응답시간 증가로 502 발생	서비스 Pod에 Network Delay 300ms 주입	Network Delay로 응답 시간이 1000ms 이상, 502 확인	502 발생 시, Retry 시도로 정상 응답 올 수 있도록 설정 변경

~ (-zsh)

18:41:59

~ (-zsh)

18:42:01

27% 20 GB 4/05 6:42 PM

LG유플러스 CTO 검색

t-dax-summit-alert

1

+ 책갈피 추가

Metric value: 14.0 (11kB)

오늘

Notified

@slack-LG_CTO-t-dax-summit-alert

Mute Monitor Declare Incident

Recovered: [demo] demo - Application LB Response 502 Count is too high

Application LB response 502 Count is too high for last 1m, please check

@slack-LG_CTO-t-dax-summit-alert

The monitor was marked as Recovered by 영진 정.

Notified

@slack-LG_CTO-t-dax-summit-alert

새 항목

Datadog 앱 오후 6:41

Recovered: [demo] demo - Application LB Latency(p99) is too high

Application LB response time(p99) is too high for last 1m, please check

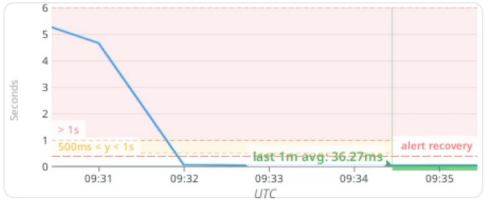
@slack-LG_CTO-t-dax-summit-alert

aws.applicationelb.target_response_time.p99 over name:k8s-istiosys-olymposi-d977b8571c was <= 1.0 on average during the last 1m.

Metric value: 0.036 (13kB)

Notified

@slack-LG_CTO-t-dax-summit-alert



#t-dax-summit-alert에 메시지 보내기

Reference

- **Principles of Chaos Engineering**
  https://principlesofchaos.org/
- **Fault Injection Simulator**
  https://docs.aws.amazon.com/fis/latest/userguide/what-is.html
- **Chaos Mesh**
  https://chaos-mesh.org/docs/
- **AWS Well-Architected Framework - Reliability**
  https://wa.aws.amazon.com/wat.pillar.reliability.ko.html

감사합니다