
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Automating Zero-Trust For Cloud Native Applications

Erin Quill & Raul Mahiques

Automating Zero-Trust For Cloud Native Applications

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

SUSE

Raul **M**ahiques

TMM

SUSE

Erin Quill

TMM

Automating Zero-Trust For Cloud Native Applications

Agenda

1. Overview
2. Implementation
3. Demo
4. Conclusions
5. QA

Overview

Zero-Trust concept

What is it about?

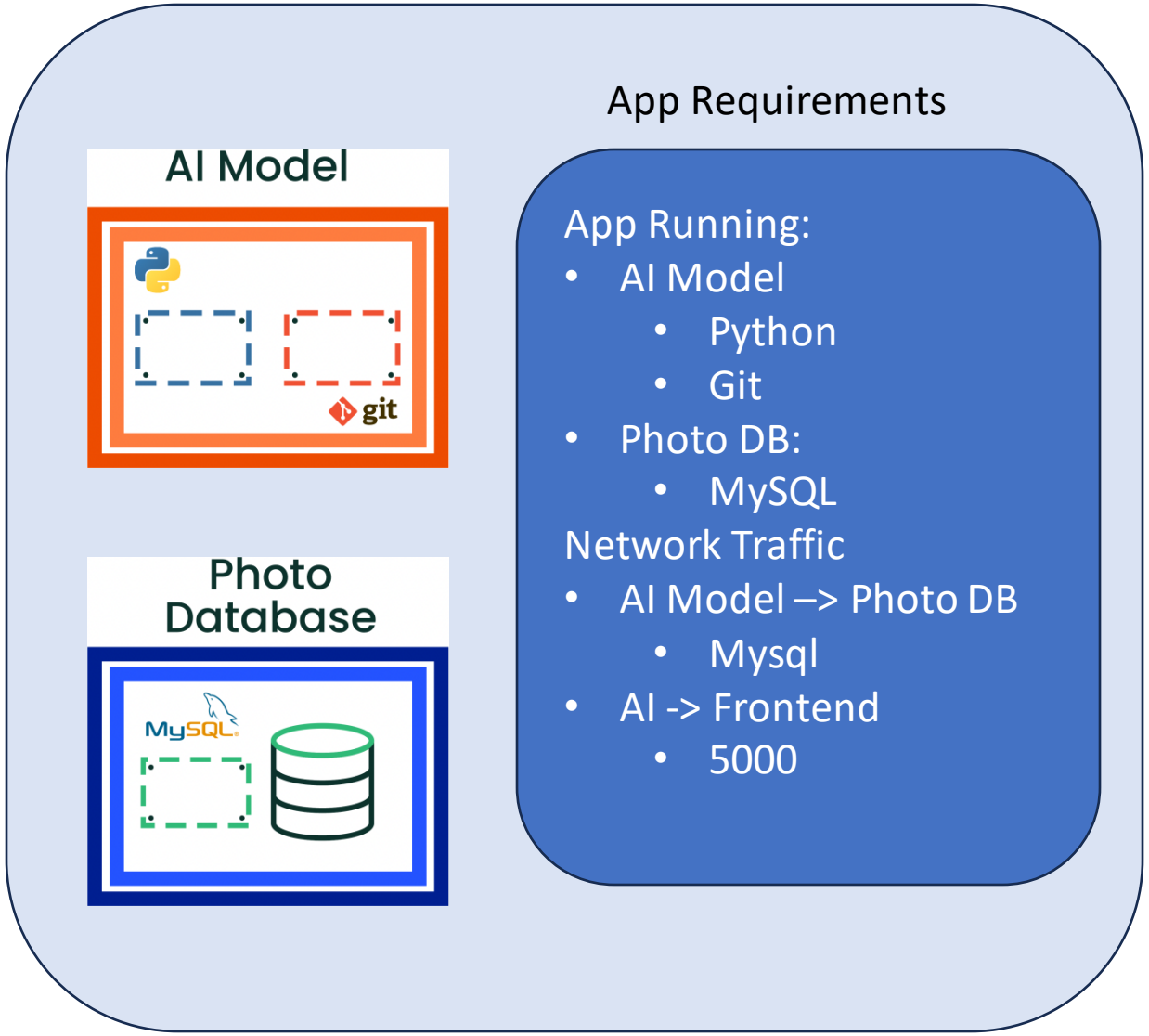
- No trust implicit
- Trust must be defined.

Zero-Trust concept

But trust who/what? for doing what? When? From where?

Trust

- A network packet,
- To reach mybackend app,
- When using mysql protocol,
- Coming from myfrontend app.



Zero-Trust concept

How?

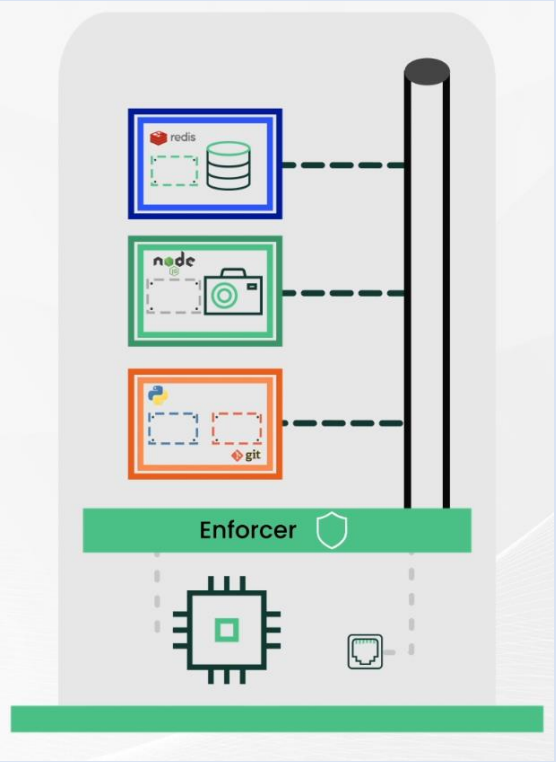
- Define security policies
  - What does the App need to work properly
- Enforced and managed by a security platform.

Security Policy

App Requirements

App Running:
- AI Model
  - Python
  - Git
- Photo DB:
  - MySQL

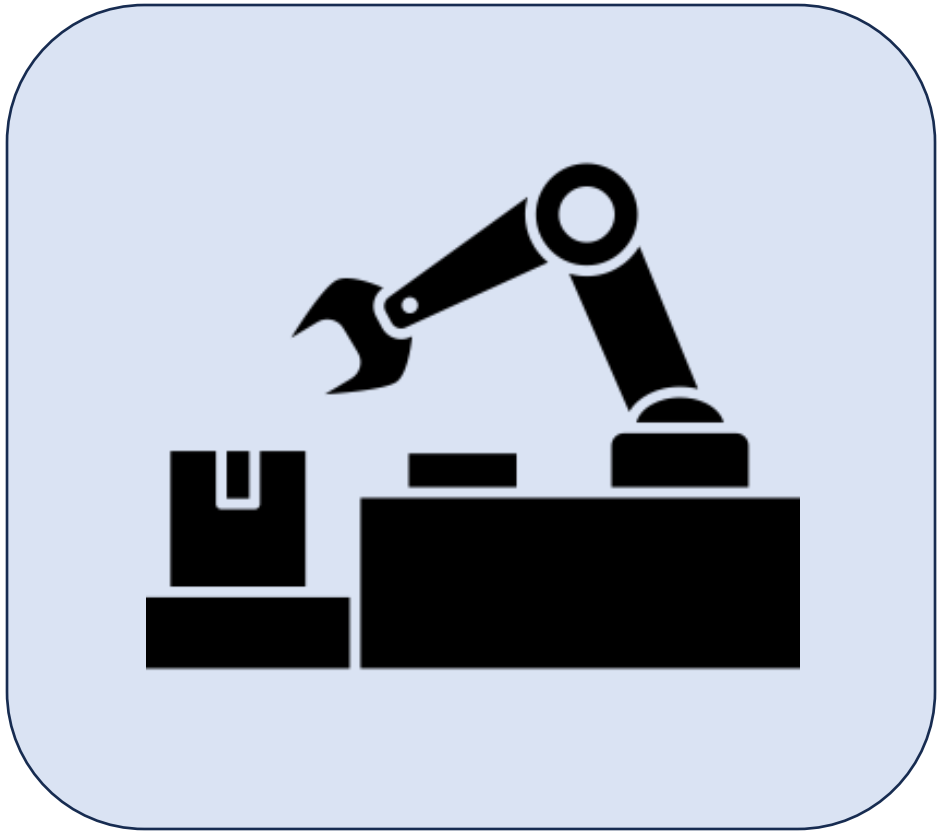
Network Traffic
- AI Model –> Photo DB
  - Mysql
- AI -> Frontend
  - 5000



Automation

Why do we need to automate?

- Manual procedures are prone to errors
- Slower processes with manual intervention
- Security is complex.
- Importance of swift response to threats
- Applications are always evolving
    - Security integration during feature
- Don’t slow down Development
- Kubernetes is a great centralized place for Automation



Automating Zero-Trust For Cloud Native Applications

NeuVector

Container Native security platform

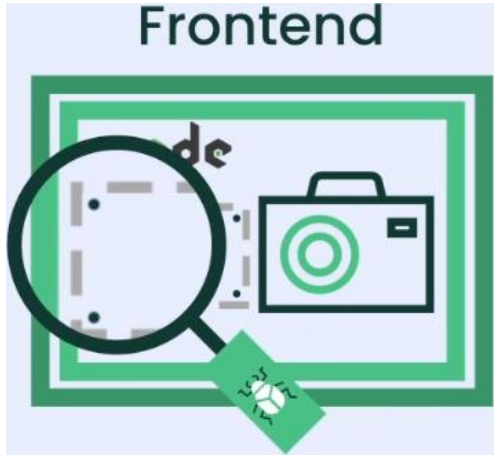
- Behaviour-based Zero-Trust
- CI/CD integration & Admission Control
- Data Loss Prevention and WAF
- Run-time Vulnerability Scanning
- Compliance & Auditing.
- Endpoint/Host Security.
- Multi-cluster Management

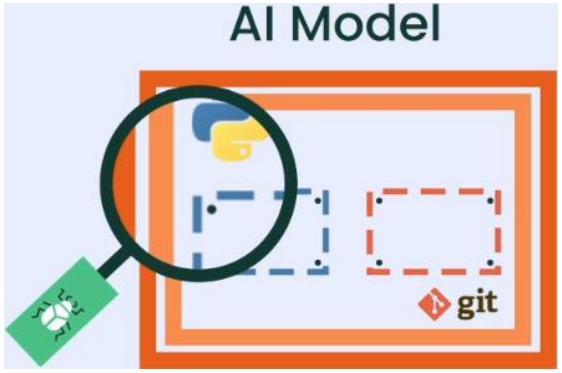
NeuVector BY SUSE

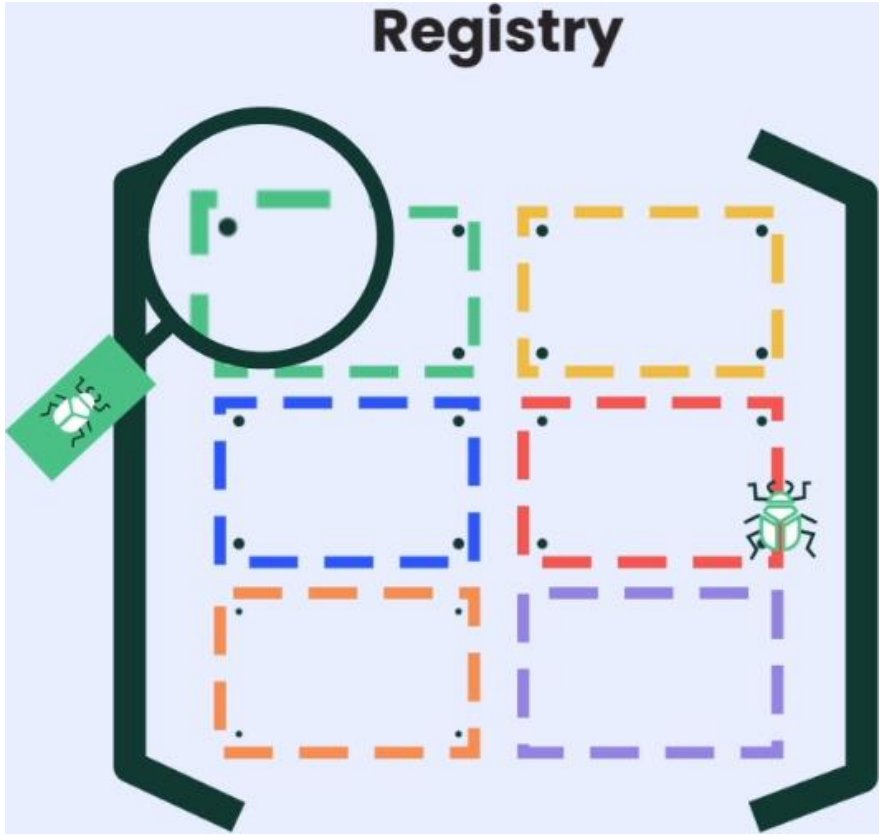
https://github.com/neuvector/neuvector

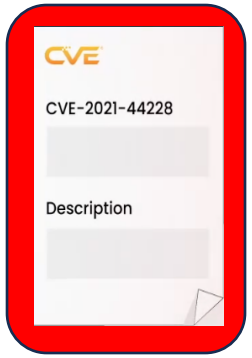
Protecting Against the Known

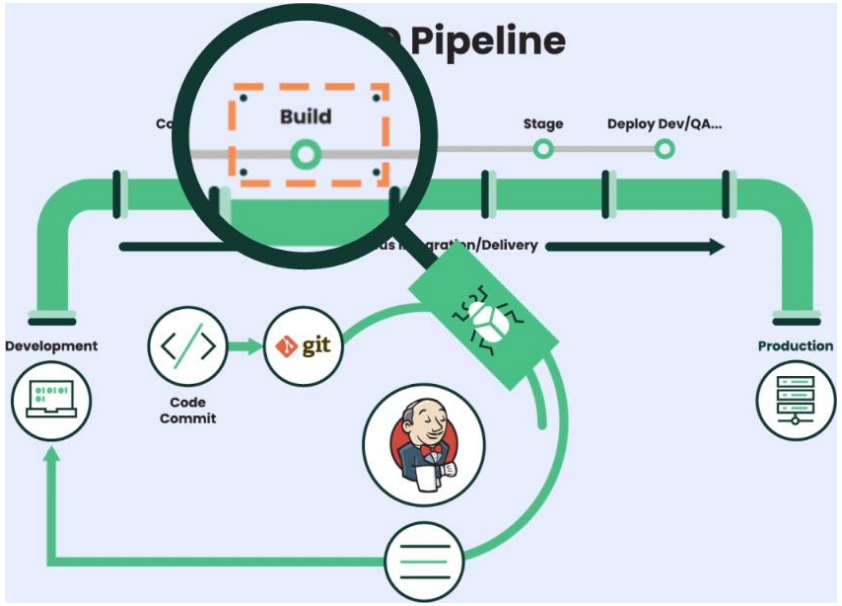
Protecting against the Known
Scanning for Vulnerabilities





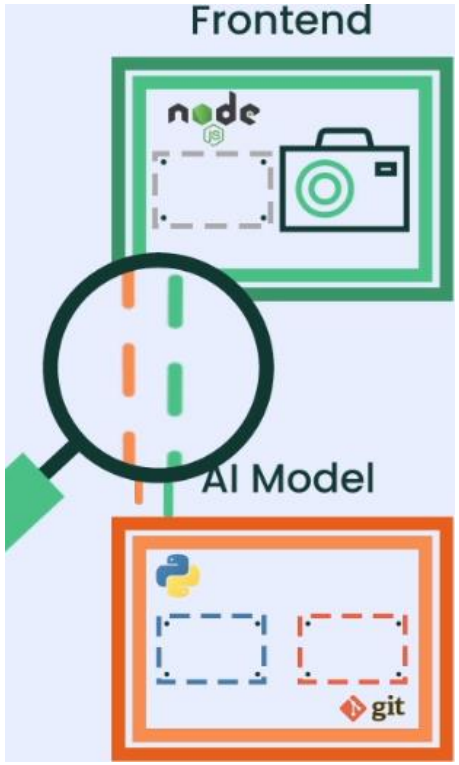


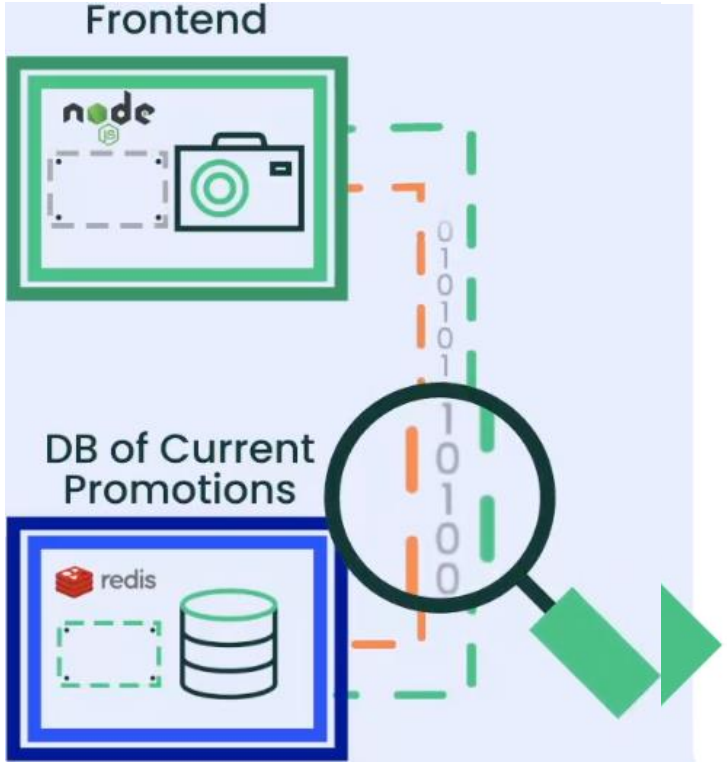




Protecting against the Known Deep Packet Inspection

Front End -> AI Model - Port 5000
Internet -> Frontend – Port 80

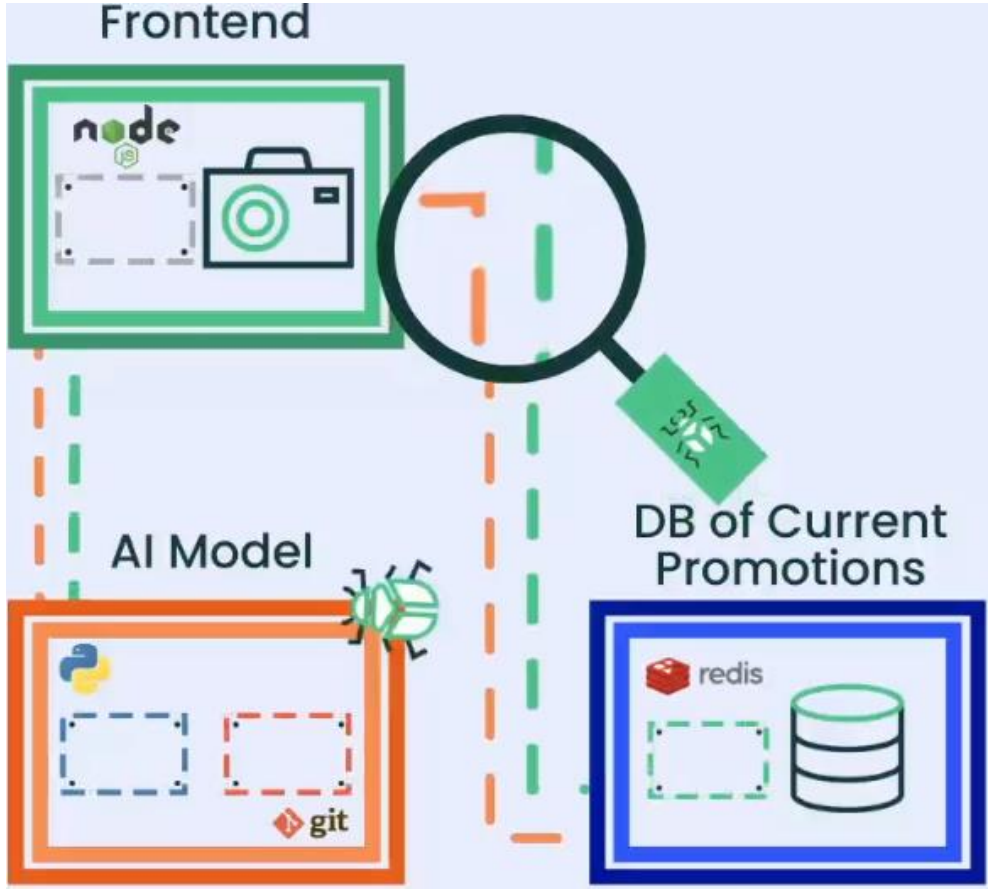




Application Protocols Recognized

HTTP/HTTPS	MySQL	RabbitMQ
SSL	Redis	Radius
SSH	Zookeeper	VoltDB
DNS	Cassandra	Consul
DNCP	MongoDB	Syslog
NTP	PostgresSQL	Etcd
TFTP	Kafka	Spark
ECHO	Couchbase	Apache
RTSP	ActiveMQ	Nginx
SIP	ElasticSearch	Jetty
ICMP	MemCache	NodeJS
gRPC	Oracle	

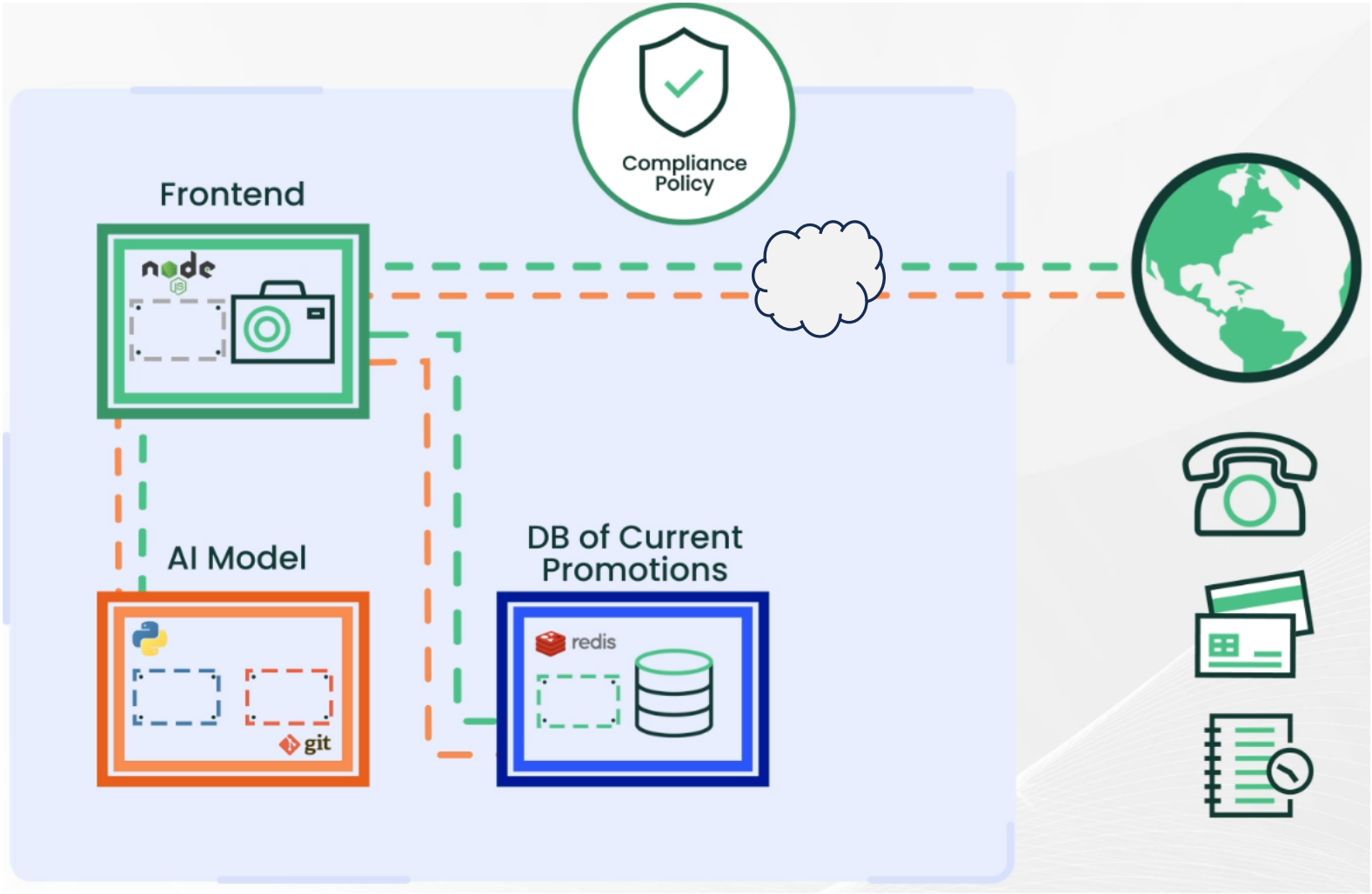
Protecting against the Known Deep Packet Inspection



Automatically Detected Threats

SYN Flood	DNS Buffer Overflow
TCP Split Handshake	ICMP Tunneling
Detect SSH1, 2, or 3	Apache Struts RCE
HTTP Neg Content	Cipher Overflow
TCP small window	IP Teardrop
DNS Zone Transfer	DNS Flood DDoS
SQL Injection	SSL Heartbleed
TCP Small MSS	MySQL Access Deny
ICMP Flood	DNS Null Type
Ping Death	DNS Tunneling
Detect SSl TLS v1.0	K8's MitM
HTTP Smuggling	CVE-202-8554

Protecting against the Known Deep Packet Inspection



Automating Zero-Trust For Cloud Native Applications

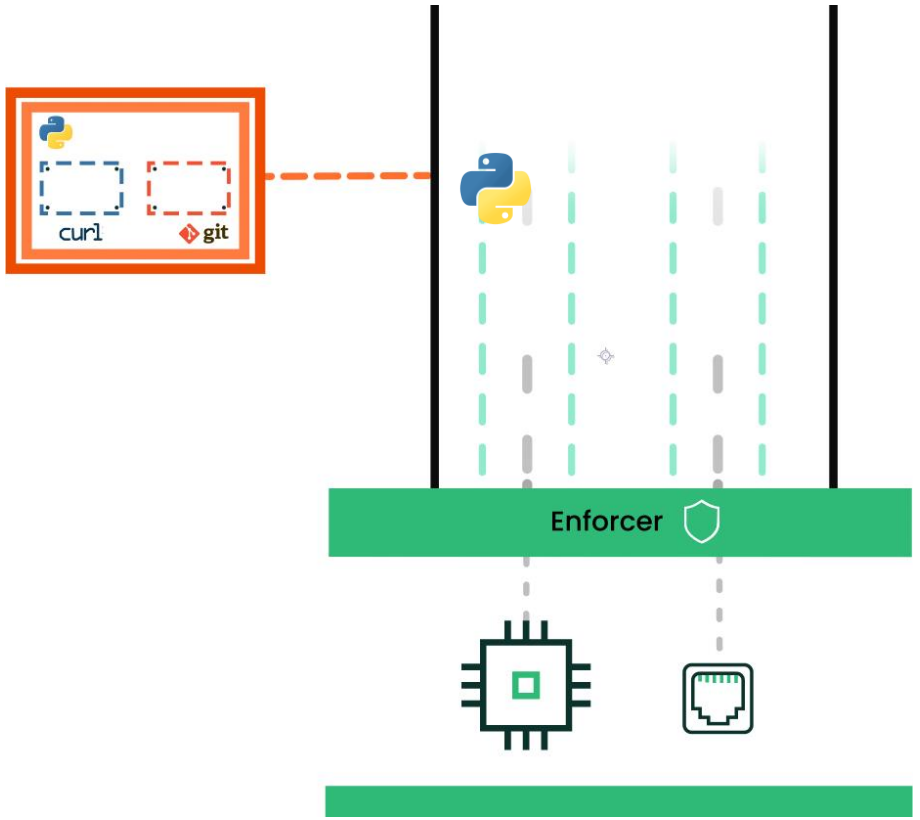
Protecting against the Unknow
Zero-Trust For Cloud Native Applications

Discovery Mode

Identifies application behavior (Learning Mode)

Process execution

Network Access



App Running:

- AI Model
  - Python
  - Git
- Photo DB:
  - MySQL

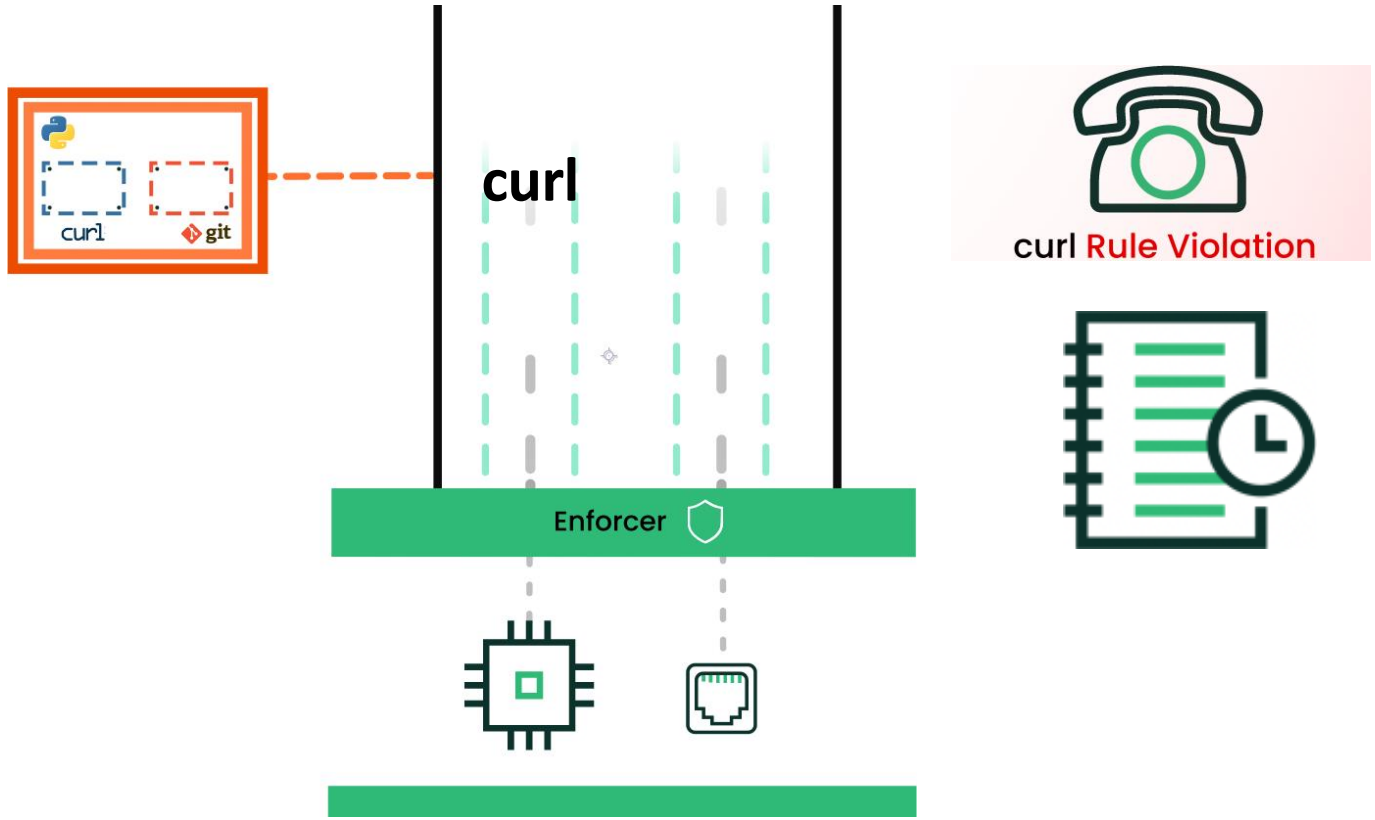
Network Traffic

- AI Model –> Photo DB
  - Mysql
- AI -> Frontend
  - 5000

Protecting against the Unknow
Zero-Trust For Cloud Native Applications

Monitor Mode

Alerts to any anomalous application behavior



App Running:

- AI Model
  - Python
  - Git
- Photo DB:
  - MySQL

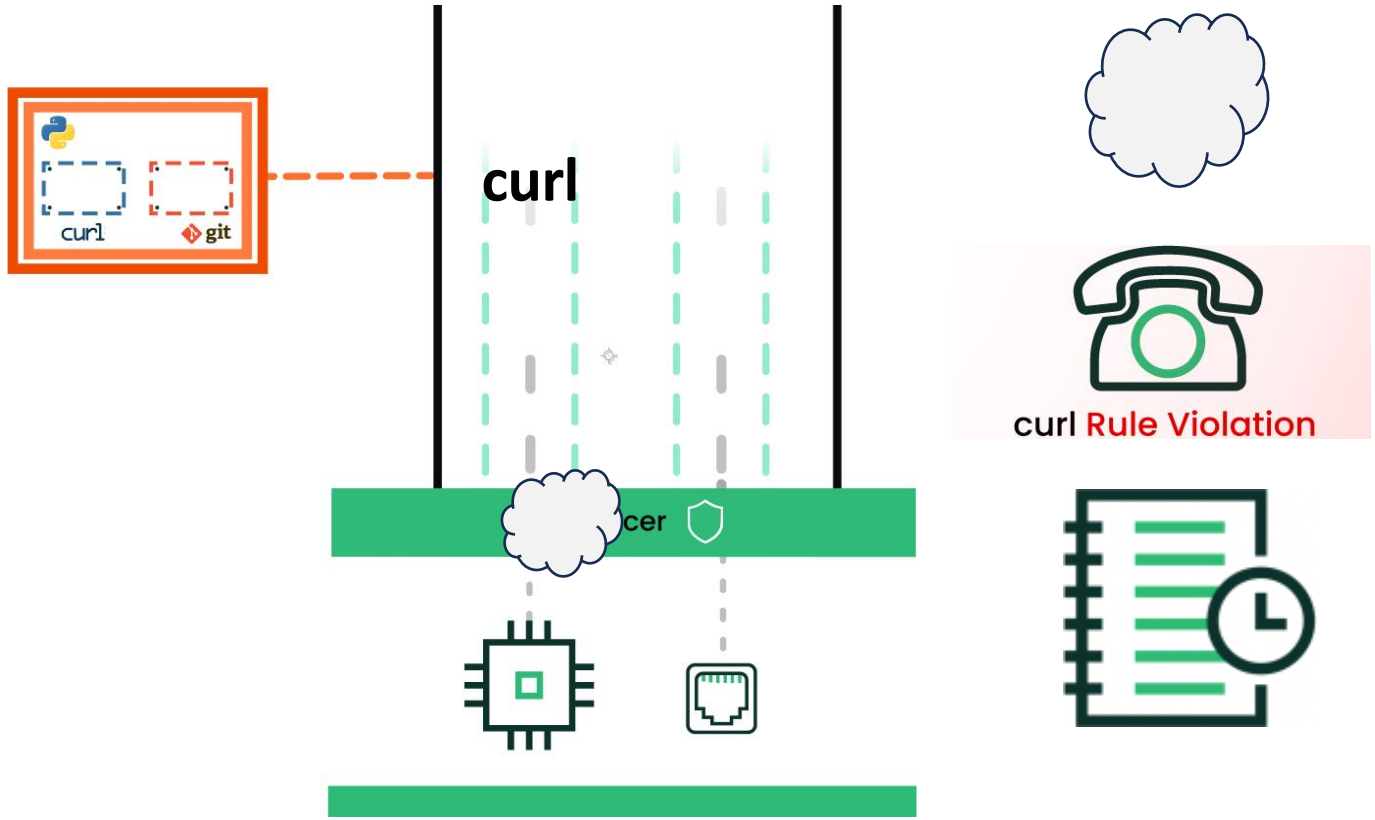
Network Traffic

- AI Model –> Photo DB
  - Mysql
- AI -> Frontend
  - 5000

Protecting against the Unknow
Zero-Trust For Cloud Native Applications

Protect Mode

Denies on any anomalous application behavior



App Running:

- AI Model
  - Python
  - Git
- Photo DB:
  - MySQL

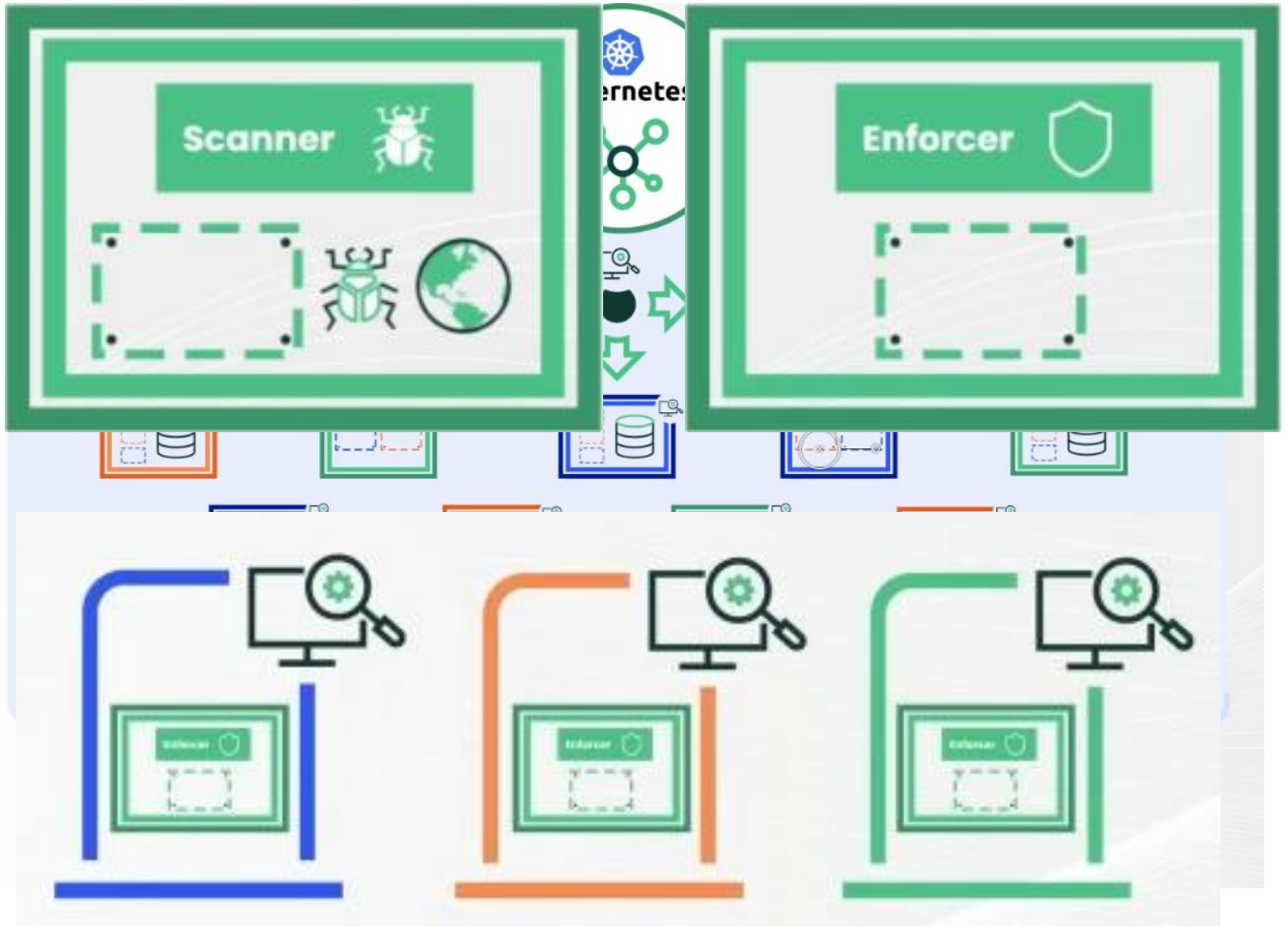
Network Traffic

- AI Model –> Photo DB
  - Mysql
- AI -> Frontend
  - 5000

Automating Zero-Trust For Cloud Native Applications

Zero-Trust Networking

- Privileged pods deployed on every host
  - Operates without sidecars
- Serves as an intermediary between the application and its origin/destination
- Identifies multiple layer 7 protocols in addition to layers 3 and 4

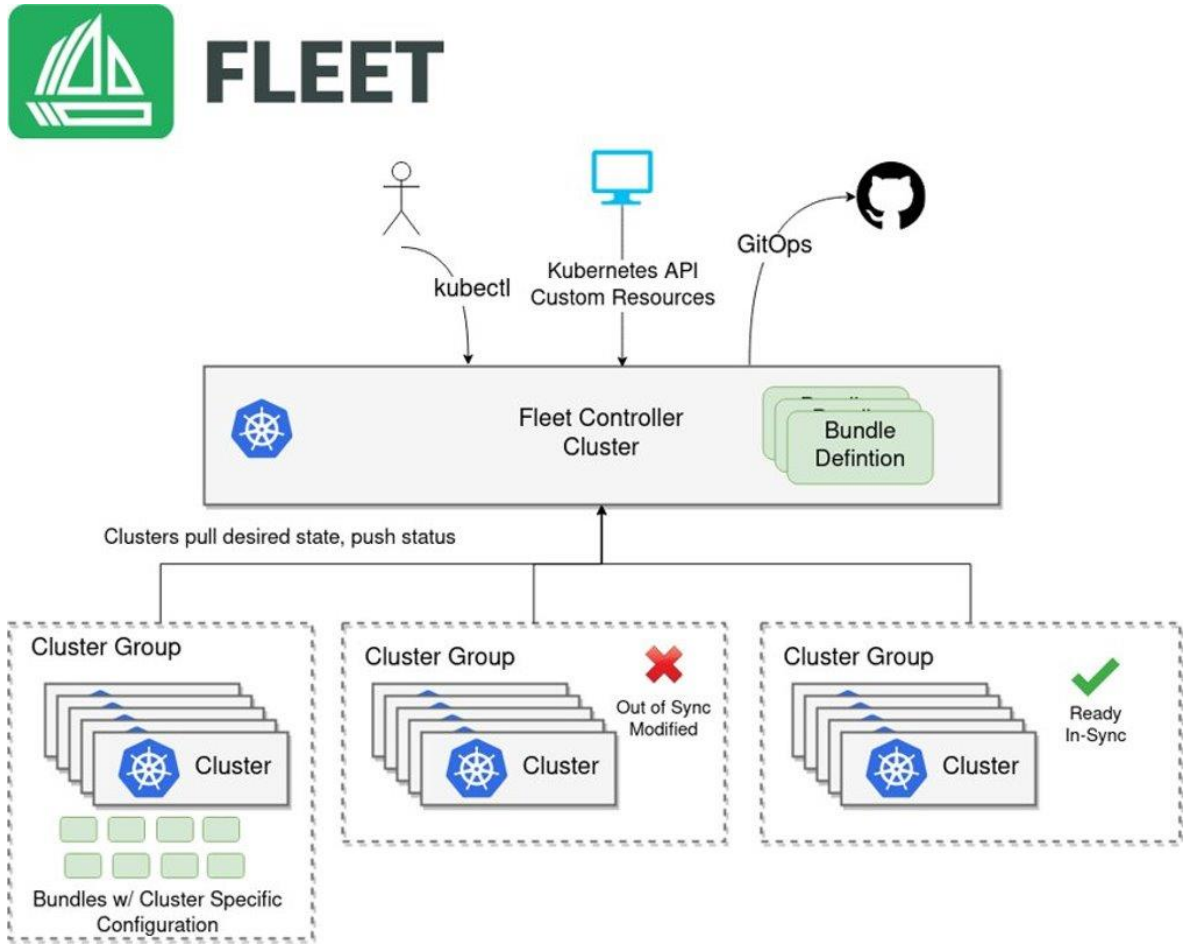


Fleet

Managing fleets of Kubernetes clusters

- Multi-cluster Management
- Deployments via raw Kubernetes YAML, Helm charts, Kustomize or mixed.
- Lightweight
- GitOps at scale

https://github.com/rancher/fleet




Implementation

Creating the Security Policy with NeuVector

- Learning mode
- Run your use cases
- Monitor mode
- Tweak or harden
- Export it.

Create your CI/CD pipeline with Fleet

- Configure Fleet
- Prepare your resource definitions
- Create the repository structure
- Upload all to the repo


Demo

Update your Application

- There is a major update to our app
- Update the security policy
- Stop our first attack

Conclusions

Thank you!

The source code and more:

- NeuVector: https://github.com/neuvector/neuvector
- Fleet: https://github.com/rancher/fleet
- The presentation materials and more:
  https://github.com/SUSE-Technical-Marketing/kcknoss-2023-China

Feel free to meet us at one of the SUSE booths!

Automating Zero-Trust For Cloud Native Applications

KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023