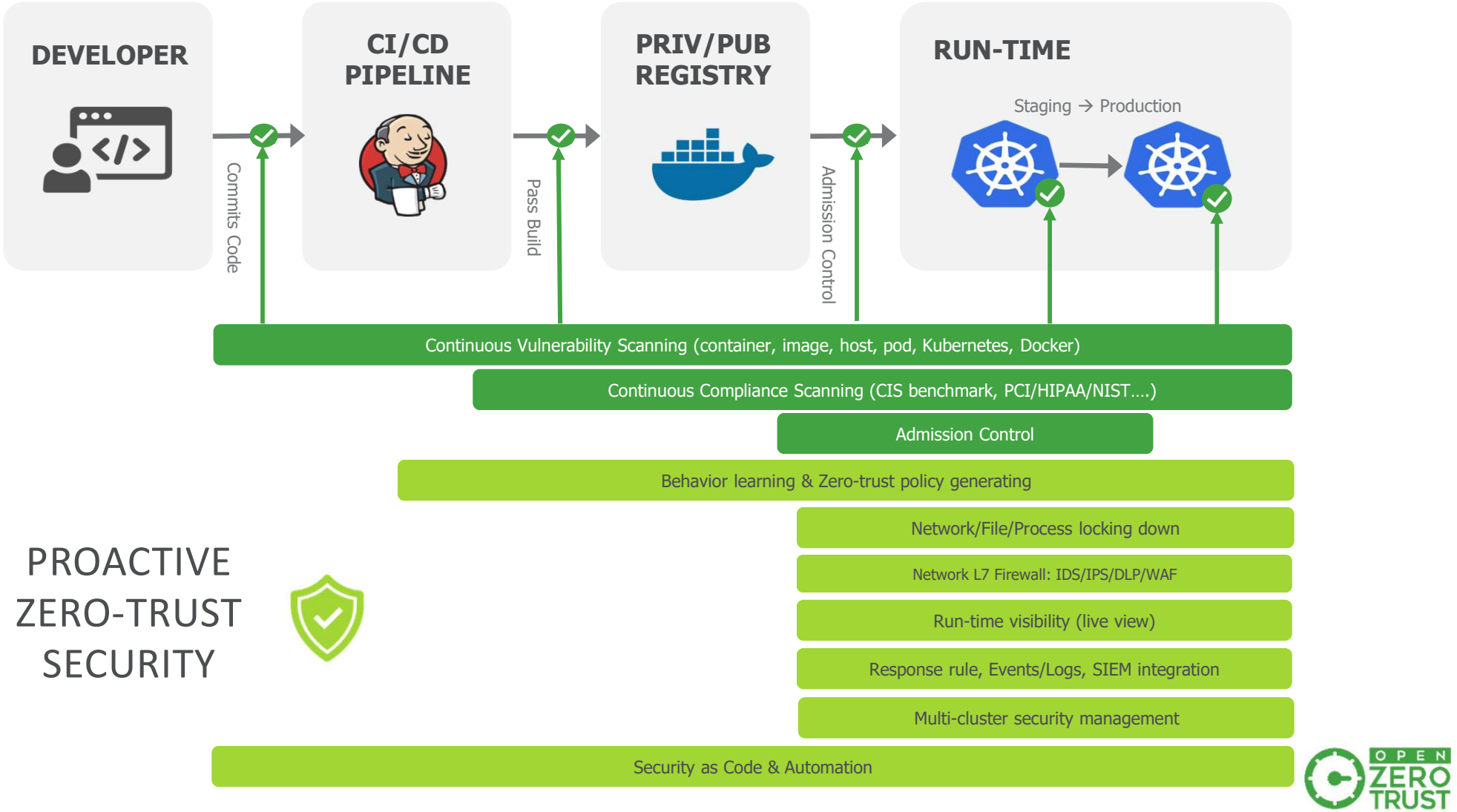
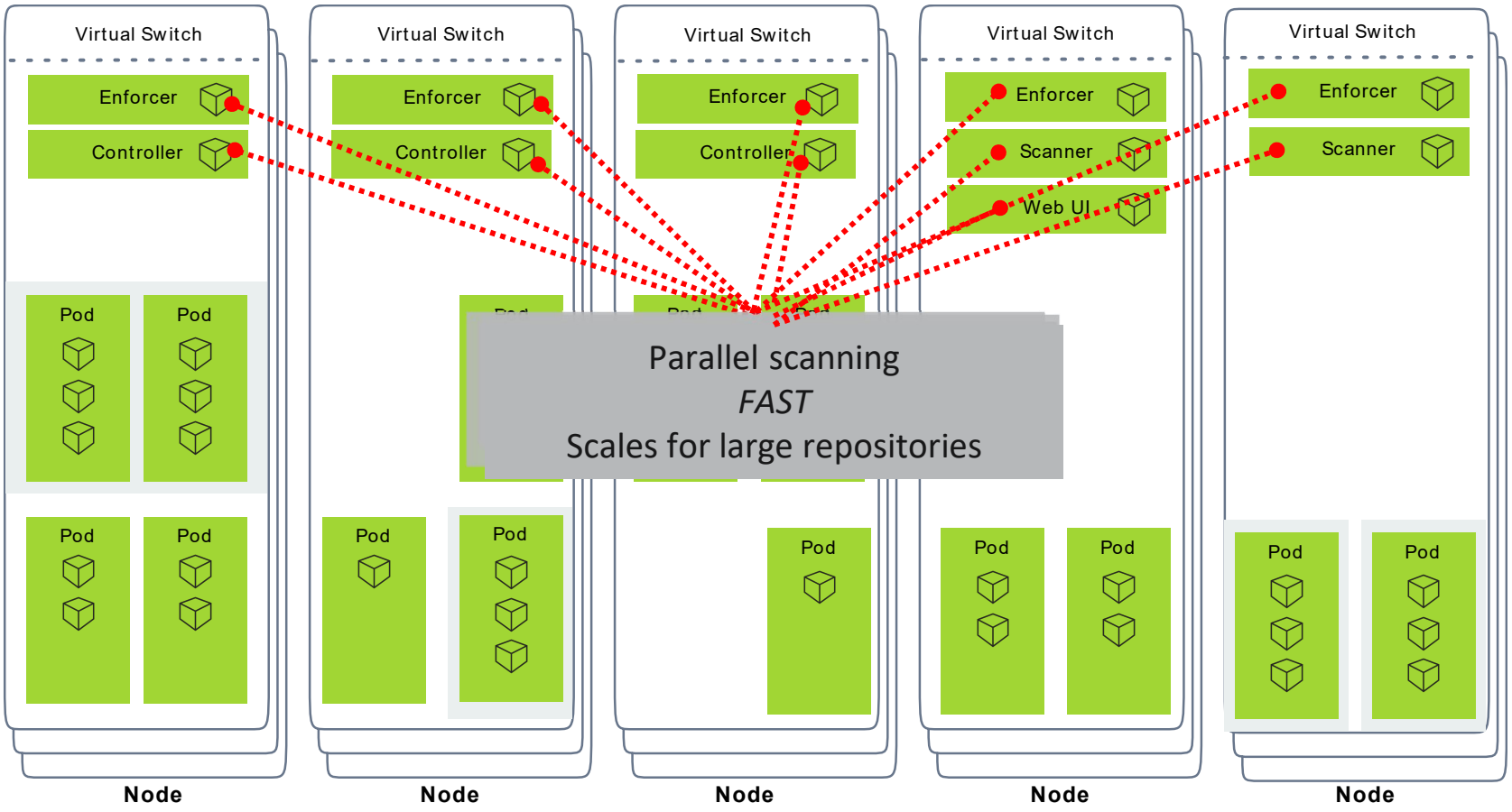
FULL LIFECYCLE CONTAINER SECURITY

FROM DEV TO PRODUCTION

May 2022


DEVELOPER
CI/CD PIPELINE
PRIV/PUB REGISTRY
RUN-TIME
Staging → Production
Commits Code
Pass Build
Admission Control
Continuous Vulnerability Scanning (container, image, host, pod, Kubernetes, Docker)
Continuous Compliance Scanning (CIS benchmark, PCI/HIPAA/NIST….)
Admission Control
Behavior learning & Zero-trust policy generating
Network/File/Process locking down
Network L7 Firewall: IDS/IPS/DLP/WAF
Run-time visibility (live view)
Response rule, Events/Logs, SIEM integration
Multi-cluster security management
Security as Code & Automation
PROACTIVE ZERO-TRUST SECURITY
OPEN ZERO TRUST

ARCHITECTURE / DEPLOYMENT



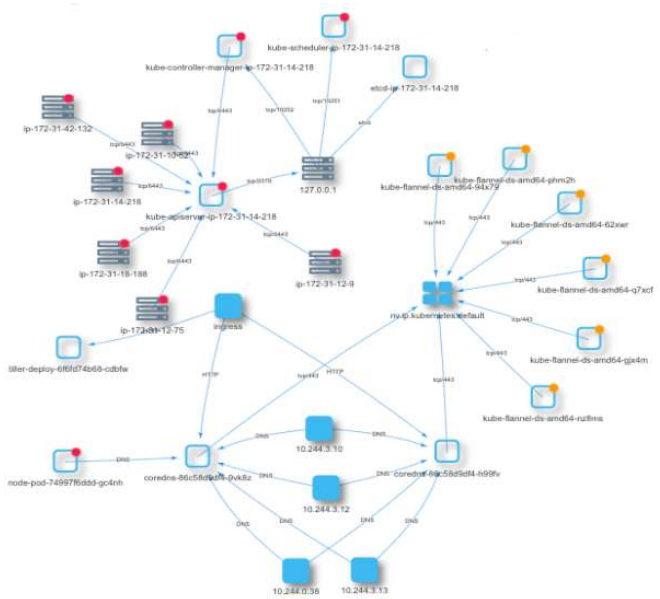
DETAILED INFRASTRUCTURE SPECS

Container	# of Instances	vCPU / Memory		Notes
Controller	1 - Minimum 3 for HA *(odd # only)*	Recommended vCPU	1	vCPU core may be shared.
		Minimum Memory	1GB	
Enforcer	1 per node/vm	Recommended vCPU	1+	One or more Dedicated vCPU for higher network throughput in Protect mode.
		Minimum Memory	1GB	Deployed as daemonset in Kubernetes
Scanner	1 - Minimum 2+ for HA/Performance	Recommended vCPU	1	vCPU core may be shared for standard workloads. Dedicate 1 or more vCPU for high volume (10k+) image scanning.
		Minimum Memory	1GB	The minimum memory recommendation assumes images to be scanned are not larger than .5GB. When scanning images larger than 1GB, scanner memory should be calculated by taking the largest image size and adding .5GB. *Example* - largest image size = 1.3GB, the scanner container memory should be 1.8GB.
Manager	1 - Minimum 2+ for HA	Recommended vCPU	1	vCPU core may be shared.
		Minimum Memory	1GB	

* Being stress tested and validated by large Cloud provider to 1000 node clusters!

OPEN ZERO TRUST

OPEN ZERO TRUST: FULL LIFECYCLE CONTAINER SECURITY PLATFORM



Unique Attack Protection in Production

- Complete Run-Time Attack Detection & Prevention – Network, Process, File, Host, Orchestrator
- Deep Network Packet Inspection for Real-Time Attack Prevention

Complete Security Automation

- Automated CI/CD Security, Security Policy As Code, Automated Alerting & Response

Vulnerability & Compliance Management for DevOps

- ‘Shift-Left’ CI/CD Scanning with Admission Control
- Kubernetes CIS Benchmark, PCI Controls

True Cloud-Native Solution

- Deploys and Updates as a Container
- Integrated into CI/CD Tools and Container Orchestrators - Kubernetes

OPEN ZERO TRUST

MITRE ATTACK COVERAGE

MITRE ATT&CK® is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

Covered by Open Zero Trust

Partially covered by Open Zero Trust

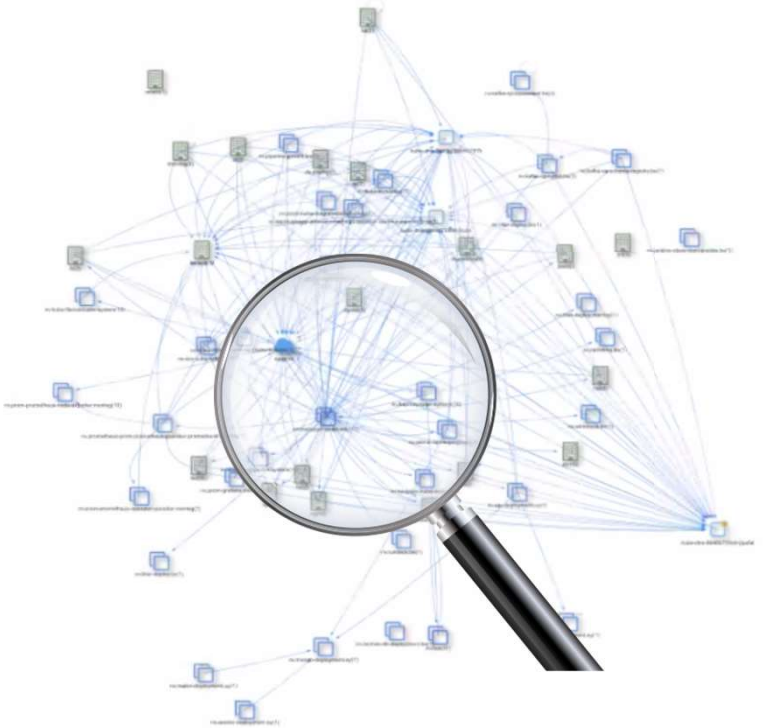
Covered by 3rd party solutions

https://blog.neuvector.com/article/how-to-use-neuvector-with-the-mitre-attck-framework

OPEN ZERO TRUST

KUBERNETES NETWORK DPI USE CASES

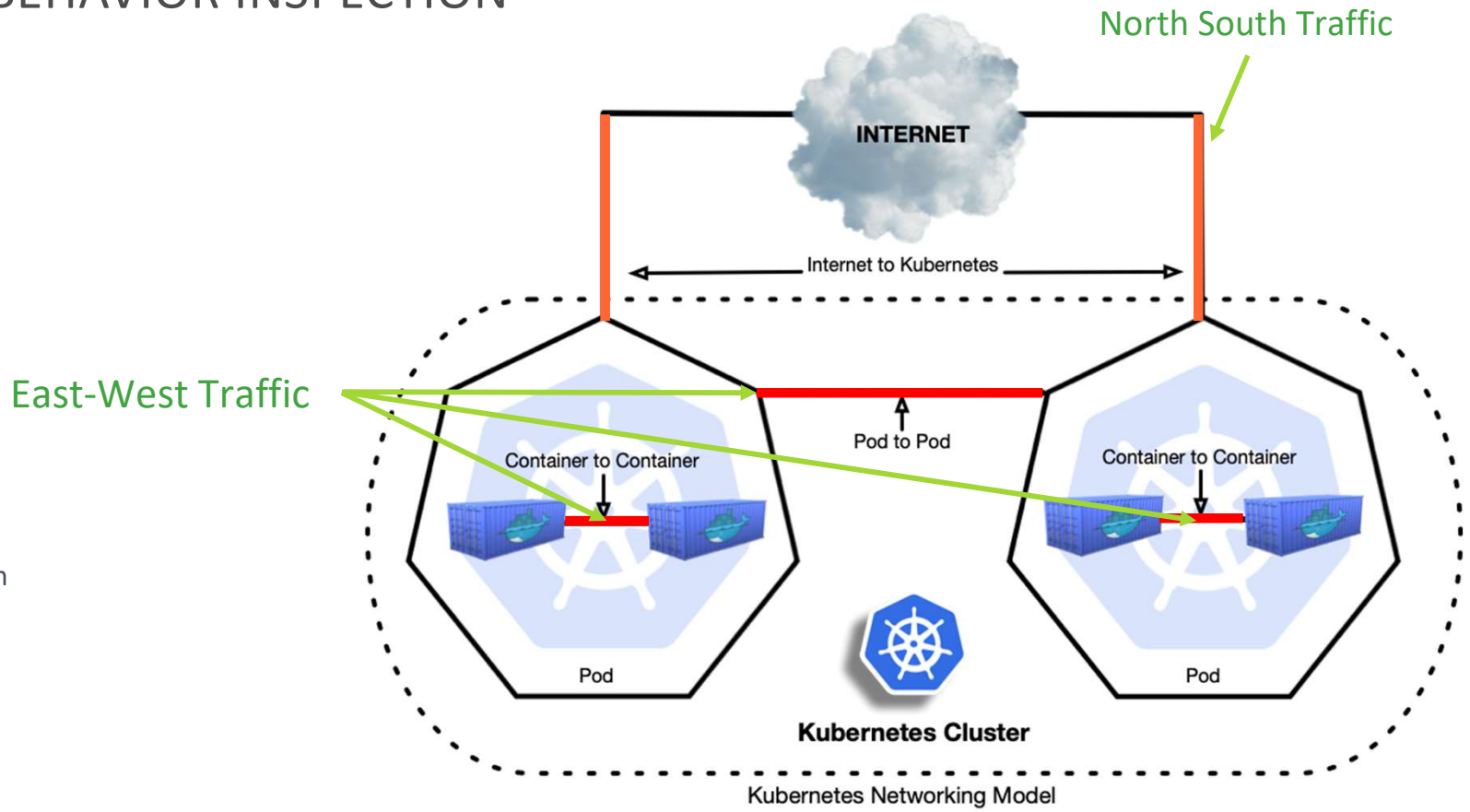
- Application Layer Segmentation
- Break Out Detection
- Strict, declarative egress controls
- Command & Control Connections
- Sensitive Data Detection
- North/South and East/West Threat Detection
- North/South In- & Exfiltration
- WAF rules including OWASP Top 10 and Log4j
- API Security
- PCI DLP Compliance
- Full Network Packet Capture / Forensics
- Application Connection Debugging



OPENSHIFT kubernetes docker RANCHER

OPEN ZERO TRUST

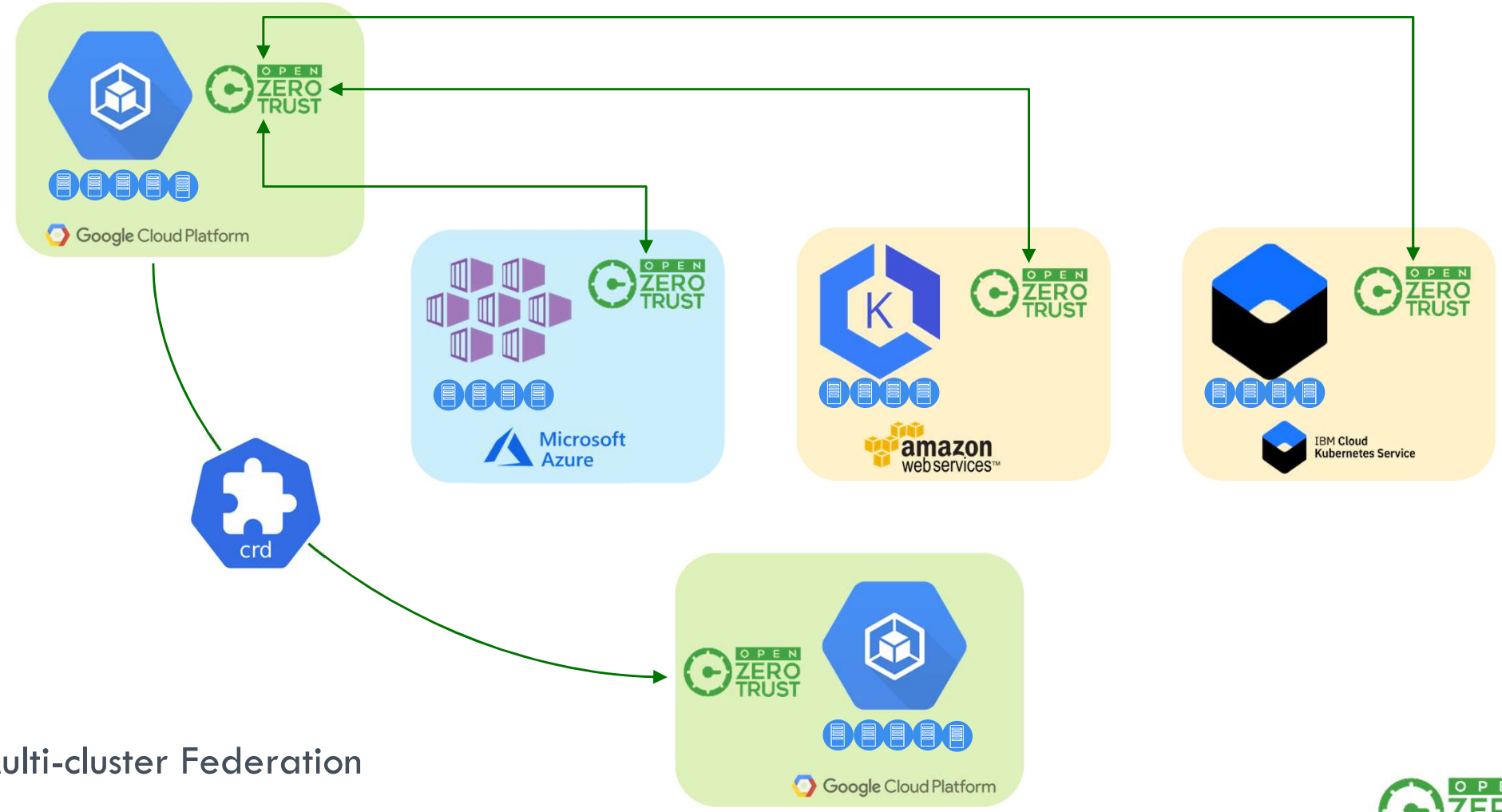
L7 NETWORK BEHAVIOR INSPECTION



Deep Packet Inspection

- Layer 3/4 Port
- Layer 7 Protocol
  +
- Processes

OPEN ZERO TRUST



Multi-cluster Federation

CVE DATABASE SOURCES

Updated nightly

Source	URL
nvd and Mitre	https://nvd.nist.gov/feeds/json/cve/1.1
SUSE Linux	https://ftp.suse.com/pub/projects/security/oval/
Ubuntu	https://launchpad.net/ubuntu-cve-tracker
RedHat	https://www.redhat.com/security/data/oval/
Debian	https://security-tracker.debian.org/tracker/data/json
Alpine	https://github.com/alpinelinux/alpine-secdb
Amazon	https://alas.aws.amazon.com/
Rancher OS	https://rancher.com/docs/os/v1.x/en/about/security/
Busybox	https://www.cvedetails.com/vulnerability-list/vendor_id-4282/Busybox.html
NGINX	http://nginx.org/en/security_advisories.html
NodeJS	https://www.npmjs.com/advisories/
Ruby	https://github.com/rubysec/ruby-advisory-db
OpenSSL	https://www.openssl.org/news/vulnerabilities.html
Apache	https://www.cvedetails.com/vendor/45/Apache.html
Java	https://openjdk.java.net/groups/vulnerability/advisories/
python	https://github.com/pyupio/safety-db
Microsoft Mariner	https://github.com/microsoft/CBL-MarinerVulnerabilityData

NeuVector CVE Database is Updated via 17 Security Sources Nightly

OPEN ZERO TRUST

APPLICATION PROTOCOLS RECOGNIZED

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

OPEN ZERO TRUST

THREATS AUTOMATICALLY DETECTED

SYN Flood	ICMP Flood	IP Teardrop
TCP Split Handshake	Ping Death	DNS Flood DDoS
Detect SSH 1, 2, or 3	Detect SSL TLS v1.0	SSL Heartbleed
HTTP Neg Content	HTTP Smuggling	MySQL Access Deny
TCP small window	DNS Buffer Overflow	DNS Null Type
DNS Zone Transfer	ICMP Tunneling	DNS Tunneling
SQL Injection	Apache Struts RCE	K8's Man-in-the-middle
TCP Small MSS	Cipher Overflow	

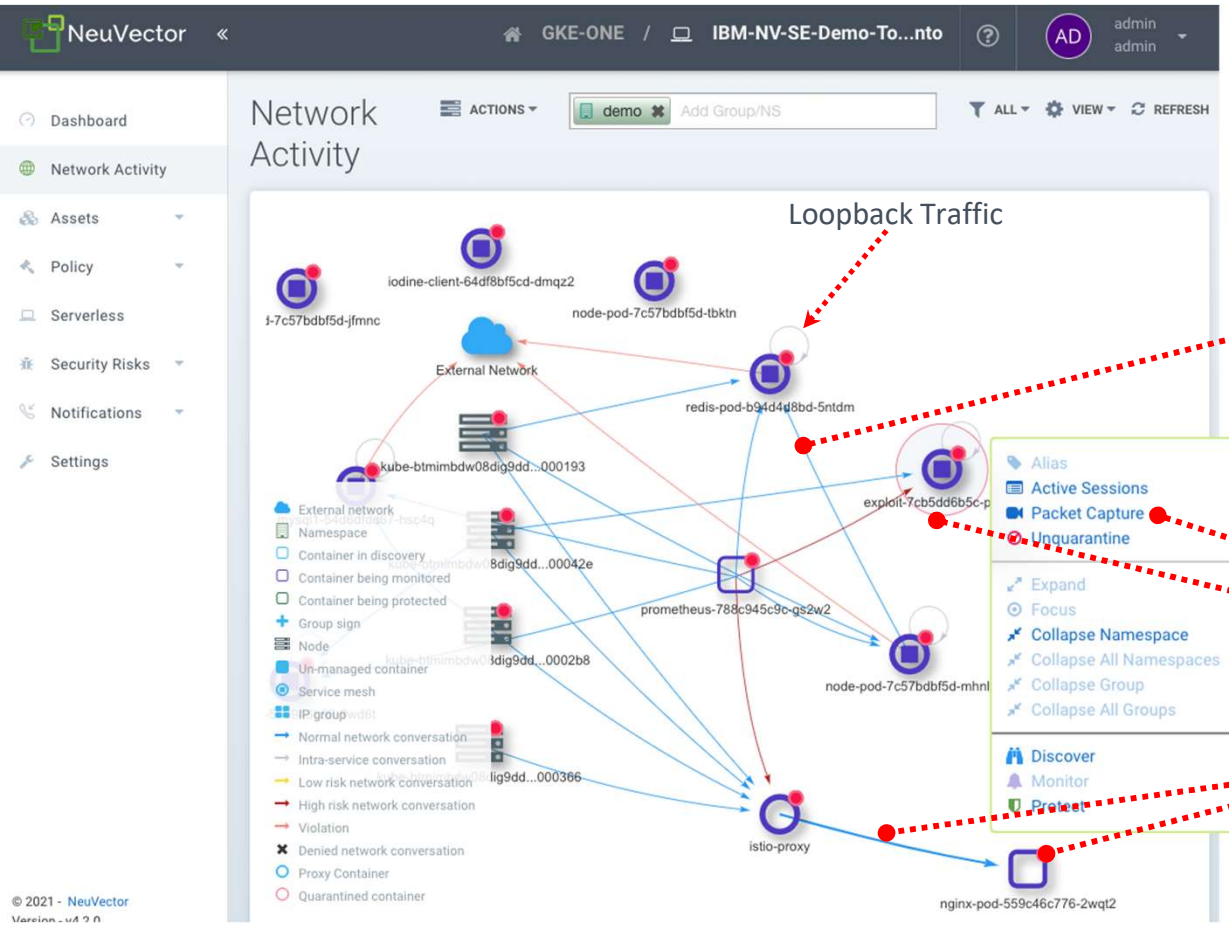
OPEN ZERO TRUST

SECURITY AS CODE

- ✓ Define Application Behaviors in Kubernetes-native yaml
  - ✓ Network Connections and Protocols
  - ✓ Ingress/egress controls
  - ✓ Processes & File System Protection
- ✓ Version Control of Security Policies
- ✓ Deploy & Enforce Global Security Rules
  - ✓ Ingress / Egress, DLP detection, etc.
- ✓ RBAC Integrated
  - ✓ Kubernetes enforcement of CRD creation permissions
- ✓ Eases migration from staging to production
- ✓ Supports Open Policy Agent (OPA), other integrations

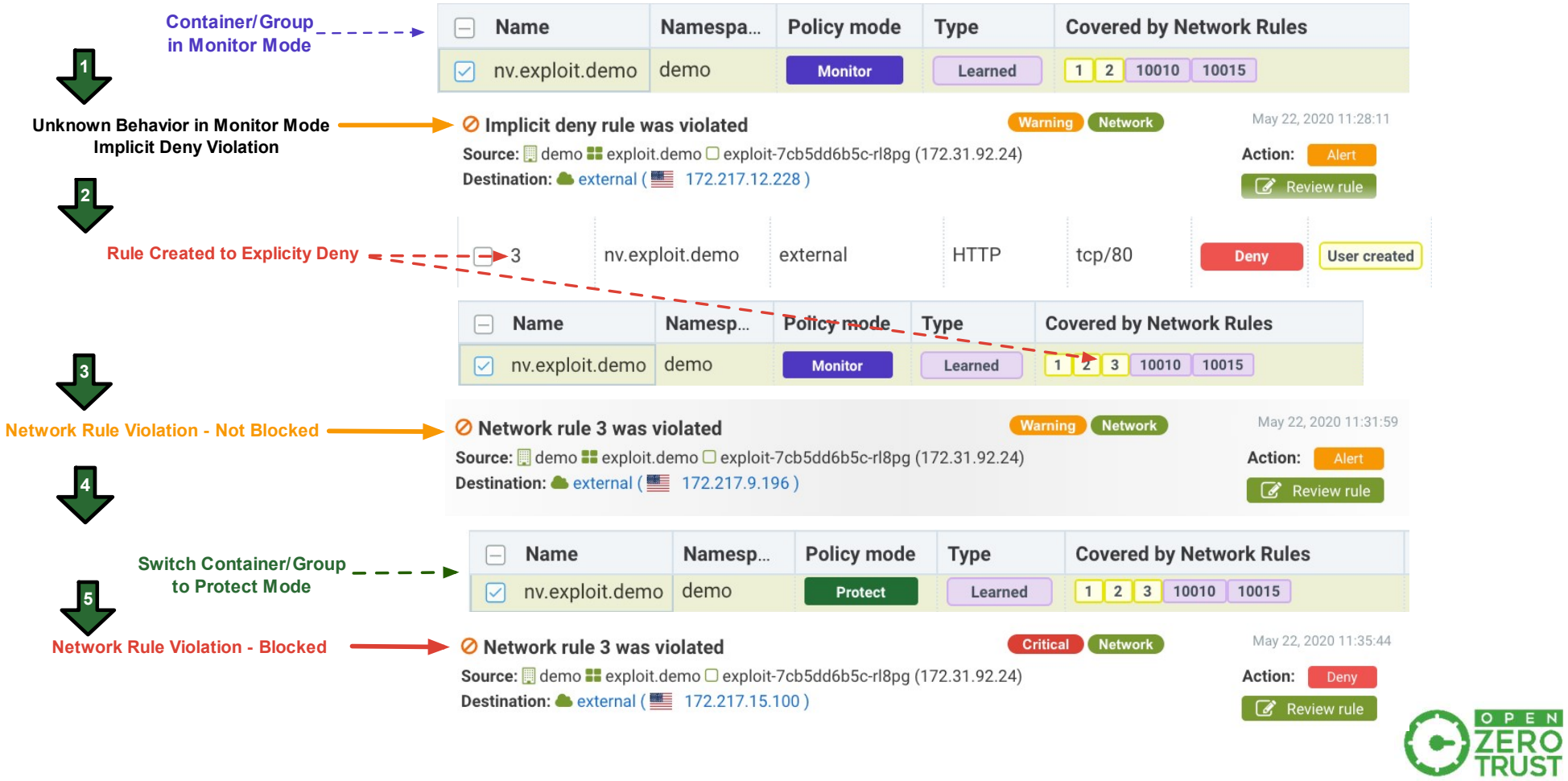
```
kind: NvSecurityRule
metadata:
  name: nv.nginx-pod.demo
  namespace: demo
spec:
  egress:
  - Selector:
      criteria:
      - key: service
        op: =
        value: node-pod.demo
      - key: domain
        op: =
        value: demo
      name: nv.node-pod.demo
    action: allow
    applications:
    - HTTP
    name: nv.node-pod.demo-egress-0
    ports: any
  file:
  - app:
    - /bin/nano
    behavior: block_access
    filter: /var/neuvector
    recursive: false
  ingress:
  - Selector:
      criteria: []
      name: nodes
    action: allow
    applications:
    - HTTP
    - Wordpress
    name: nv.nginx-pod.demo-ingress-0
    ports: any
  process:
  - action: allow
    name: nginx
    path: /usr/sbin/nginx
  target:
    Selector:
      criteria:
      - key: service
        op: =
        value: nginx-pod.demo
      - key: domain
        op: =
        value: demo
      name: nv.nginx-pod.demo
    policymode: Monitor
```

VISUALIZE & PROTECT SERVICE MESHES



- ✓ Shows Application Workloads ONLY
  - View/Hide Istio System Containers
- ✓ Automates Istio System Monitoring
  - Learn & Allow-list Istio Control Plane & Proxy Connections
- ✓ Automates Segmentation
  - App & System Allow-Lists
- ✓ Detects Attacks Even Via Trusted Connections
- ✓ Automated Response Capabilities
  - Packet Capture
  - Quarantine Without Killing
  - Webhooks
- ✓ Inspects Traffic with Pod-to-Pod Encryption On
  - Between container & side-car proxy
  - U.S. Patent 11,075,884 issued July 27, 2021

OPERATING – SETTING & ENFORCING RULES



THANK YOU!