
Navigating the shallow waters of Kubernetes Security
Me harty's guide fer all ye scallywags

The modern application stack


Application


Orchestrator


Container

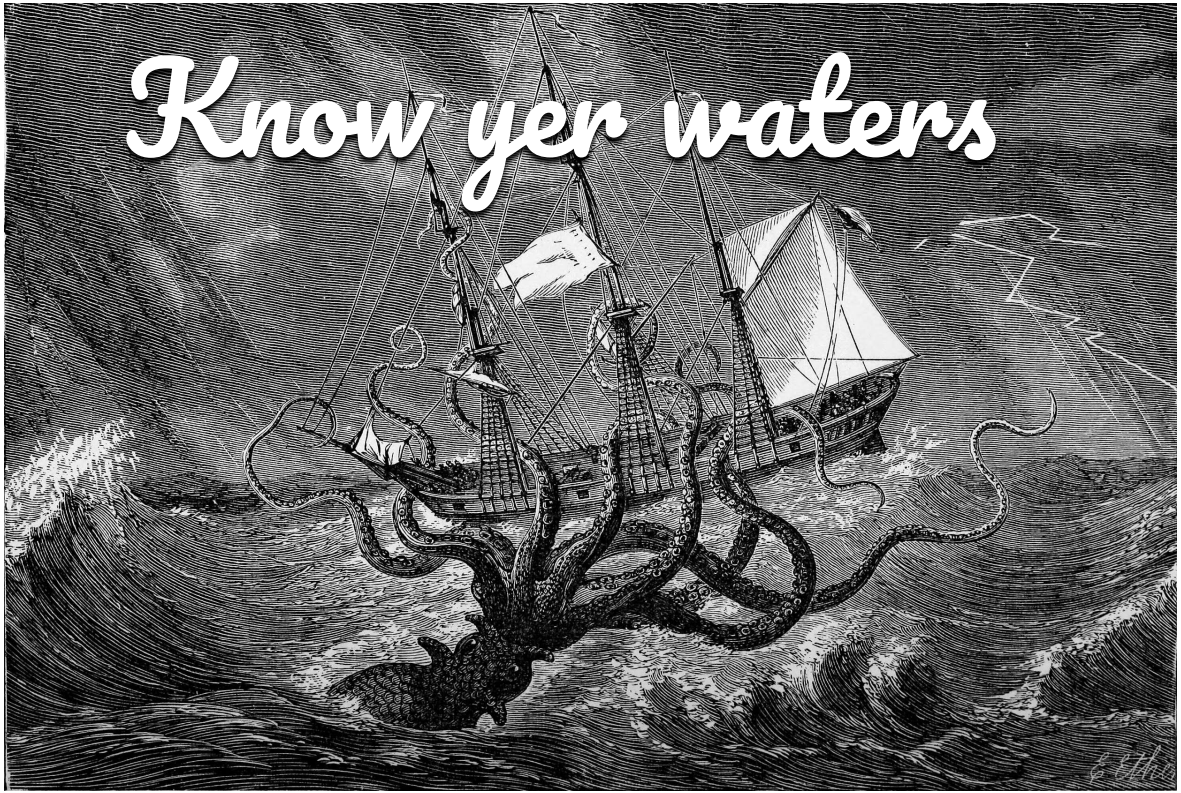

OS/Kernel


Platform

The modern application stack

Application	THIS IS BAD
Orchestrator	
Container	
OS/Kernel	THIS IS BAD
Platform	HORRIBLE BUT SOMEBODY ELSE NEEDS TO CARE

Know yer waters



The modern application stack

Application

Orchestrator


Container

OS/Kernel

Platform

Can containers help with my security?

Yes, if you build them, deploy and run them properly.

Types of issue a container can help with limiting the damage

- Persistence
- Tooling/living-of-the-land
- Path traversal
- Uncontrolled resource consumption/DoS

Where containers will not help

If your app is bad, it won't fix it

- Injection/Insecure Deserialization/RCE
- Use-after-free, TOCTOU, overflow
- XSS, CSRF, **SSRF**
- Change CRI defaults, get yourself in trouble

Get'er going

The modern application stack

Application

Orchestrator

Container

OS/Kernel

Platform

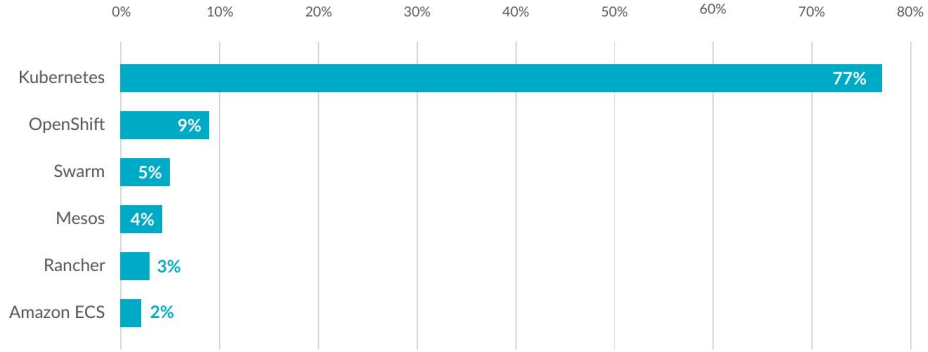
Infrastructure as a Code

Chef

Puppet

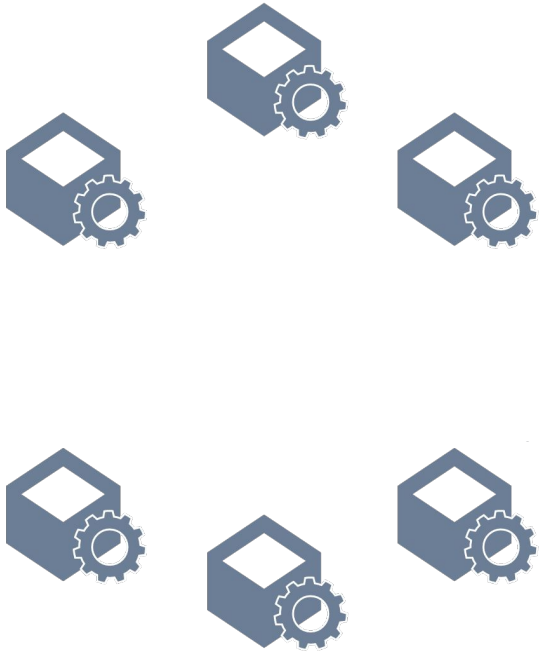
Terraform

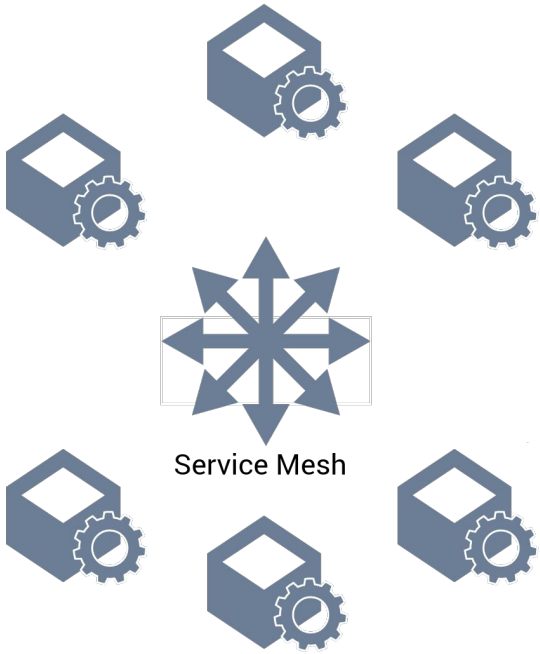
Kubernetes!

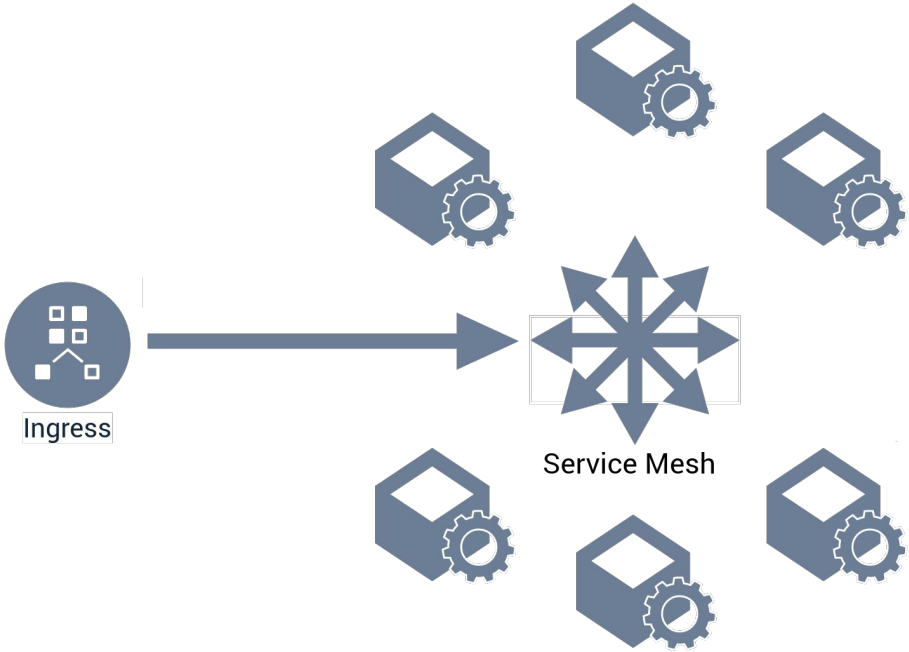


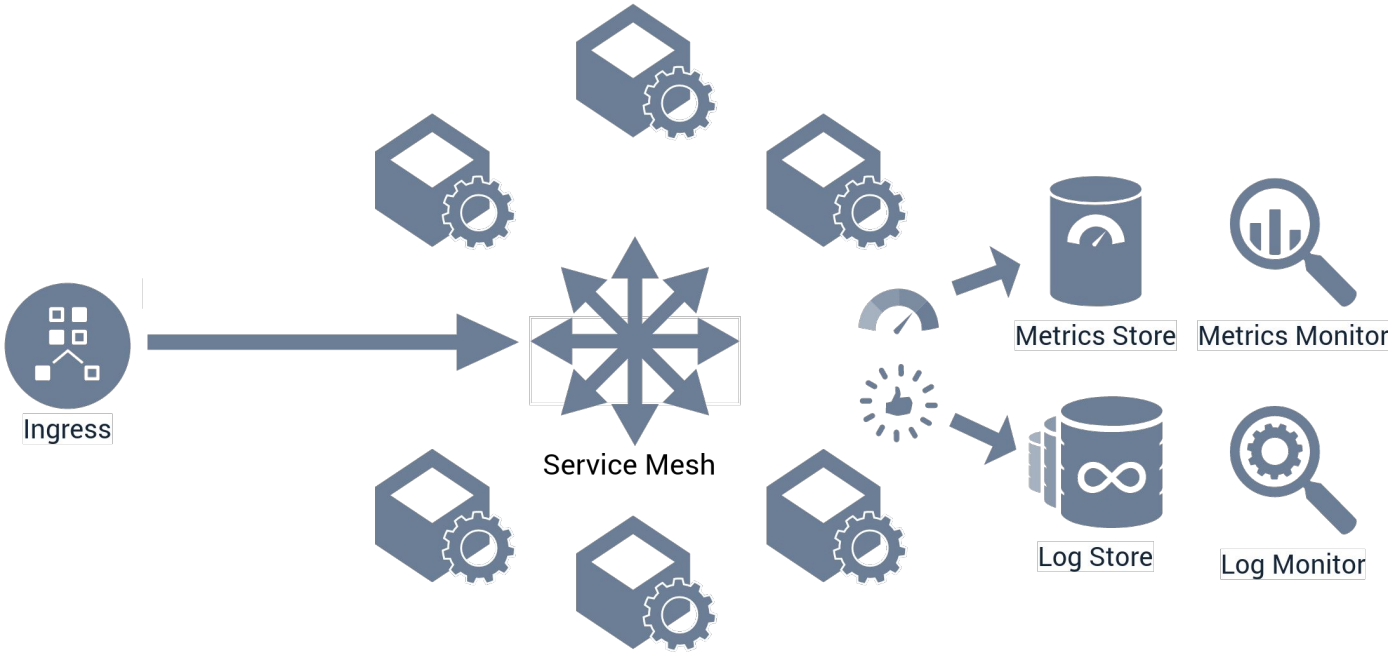
Core K8s concepts

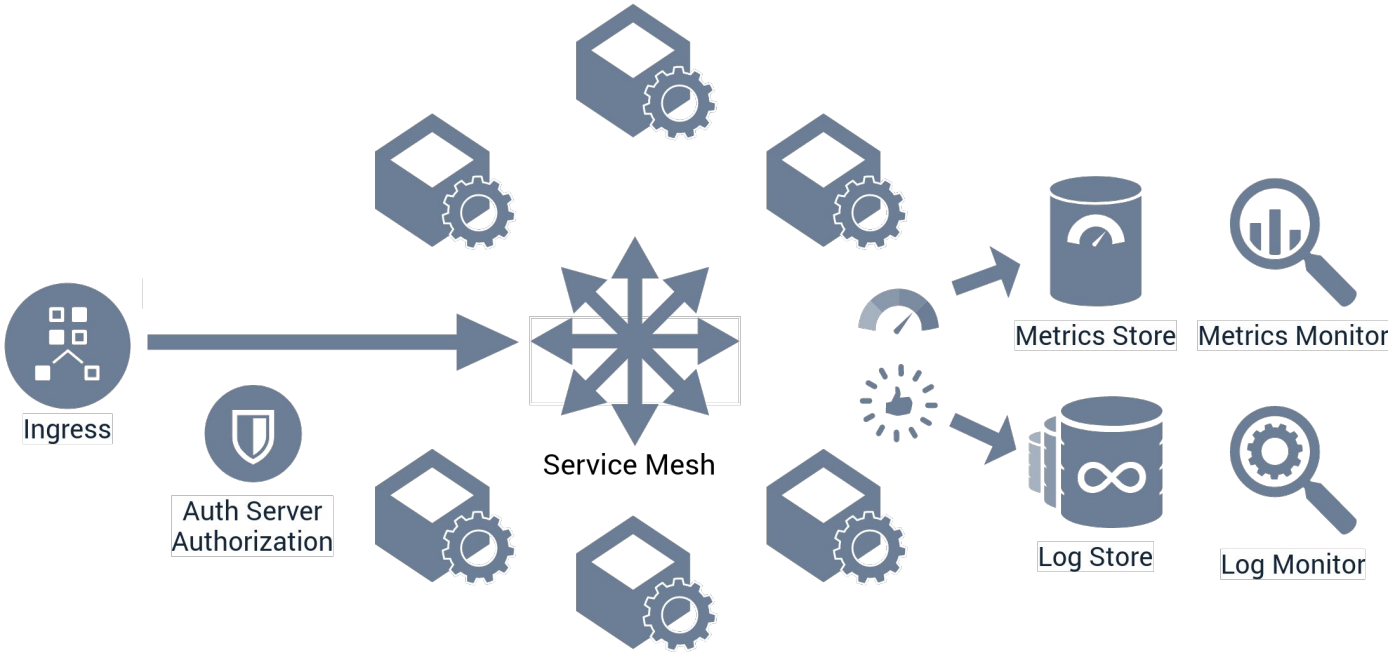
1. Declarative deployment
2. No hidden APIs
3. Meet the developer where they are at

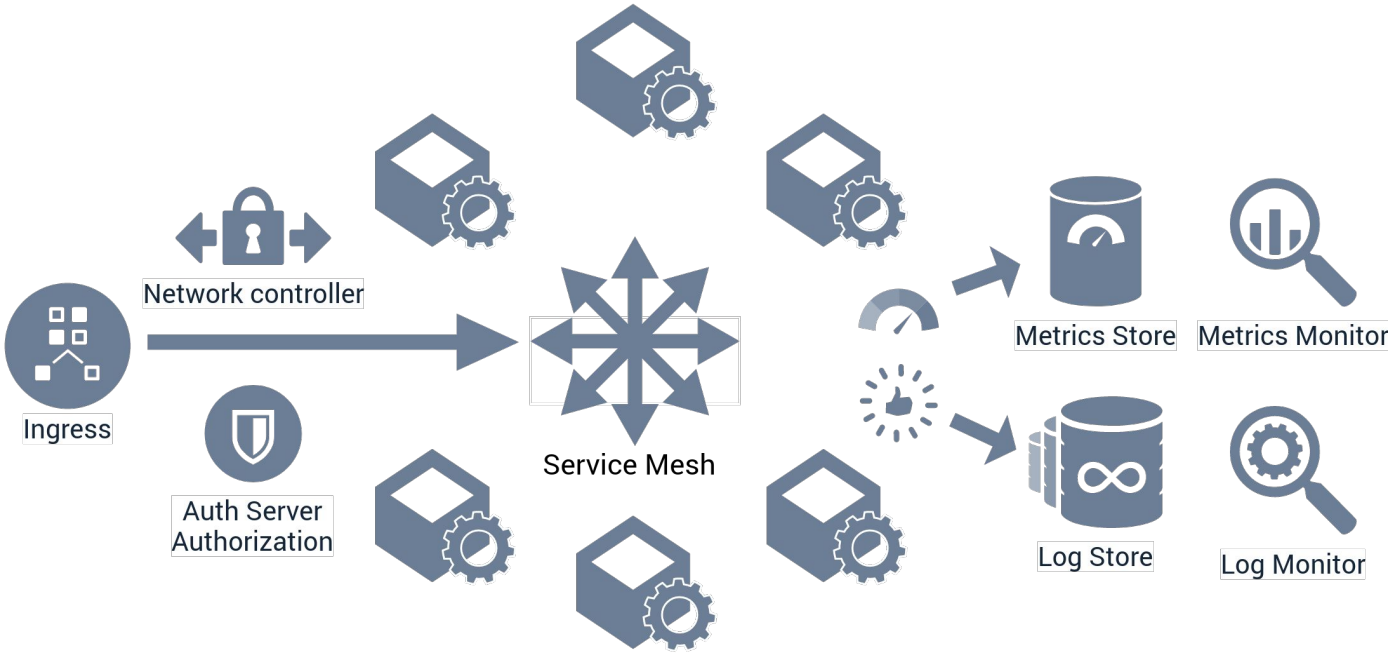


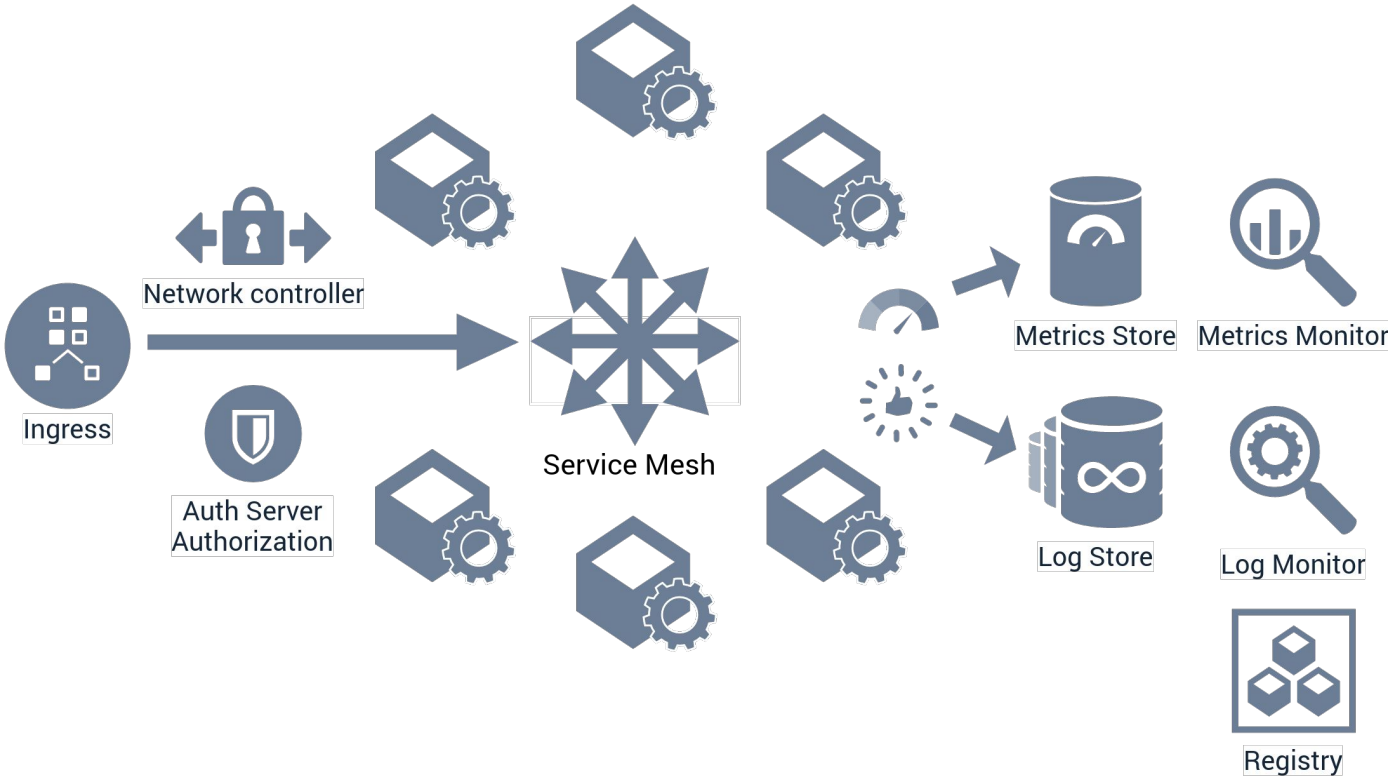


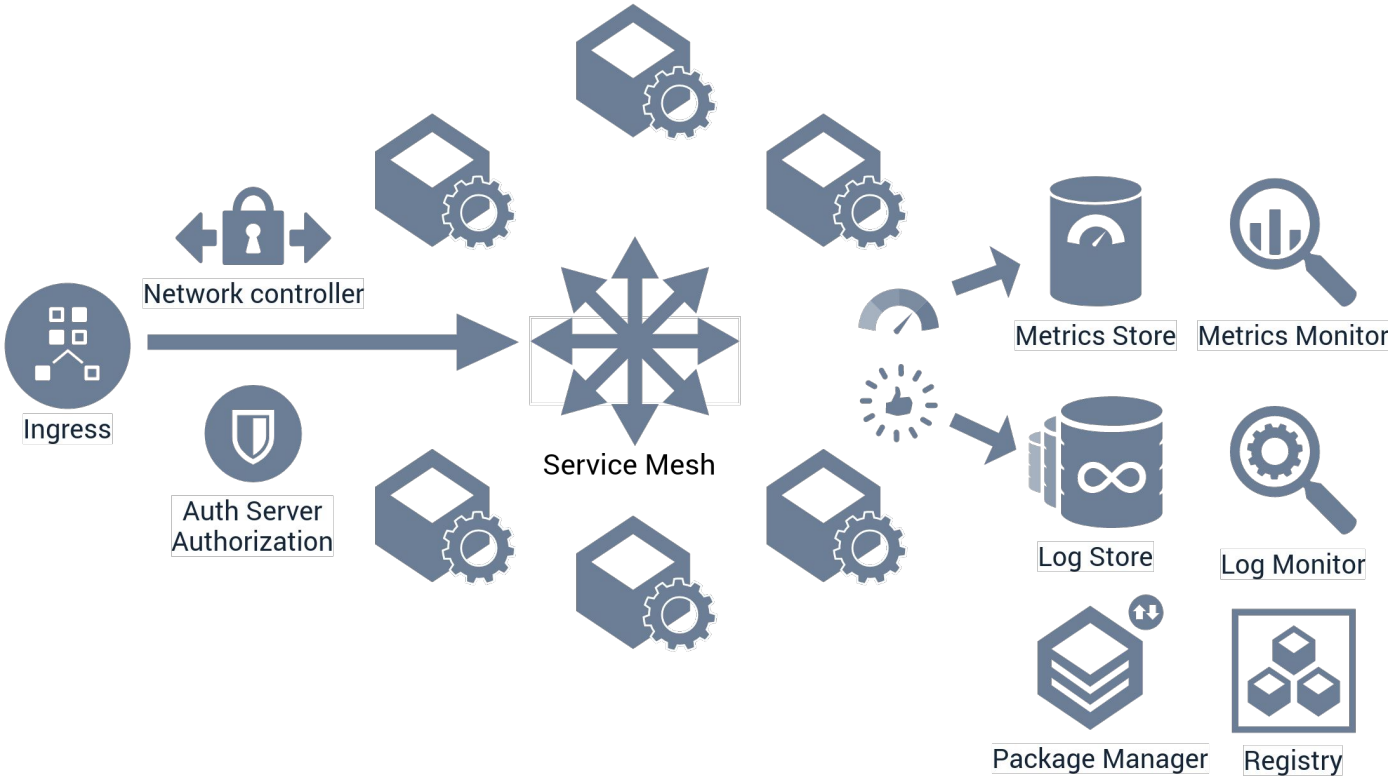


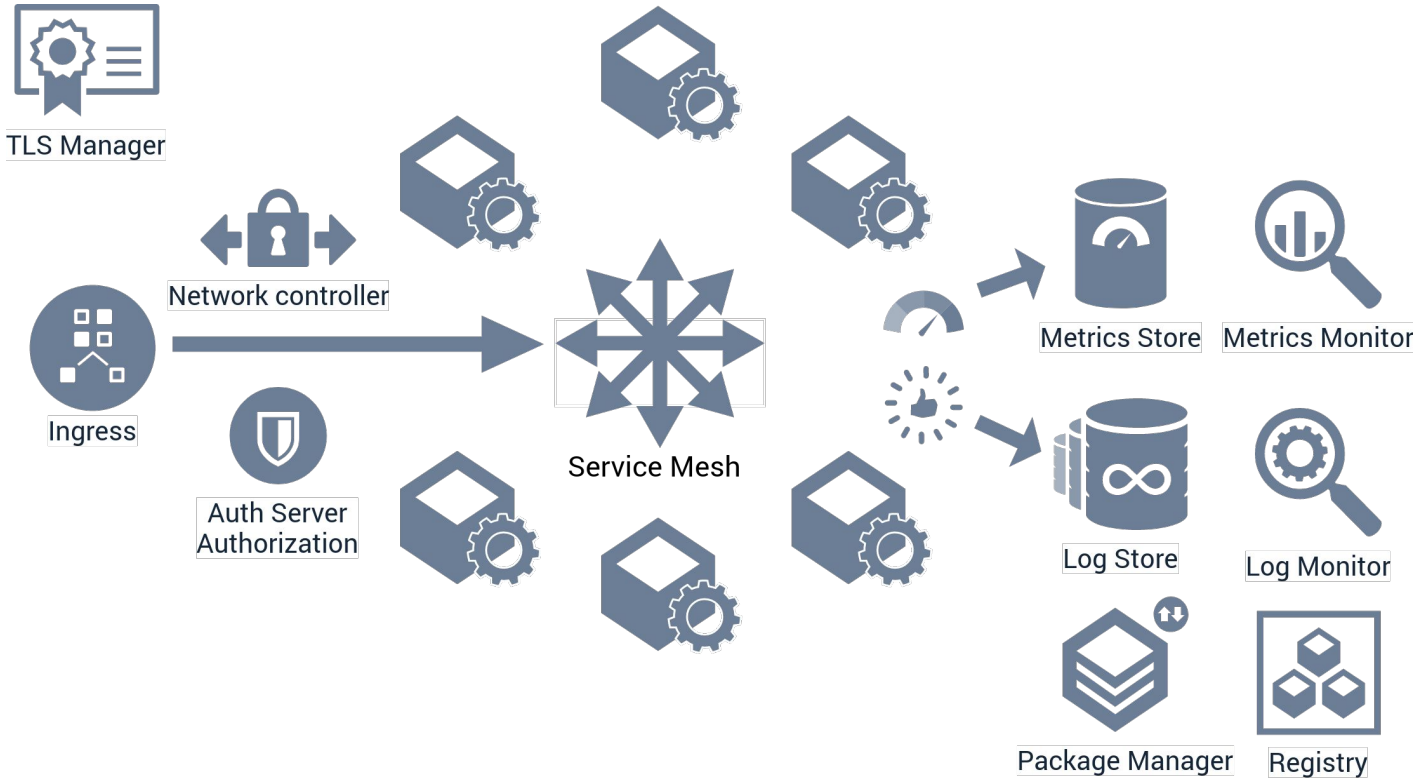

Ingress
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor

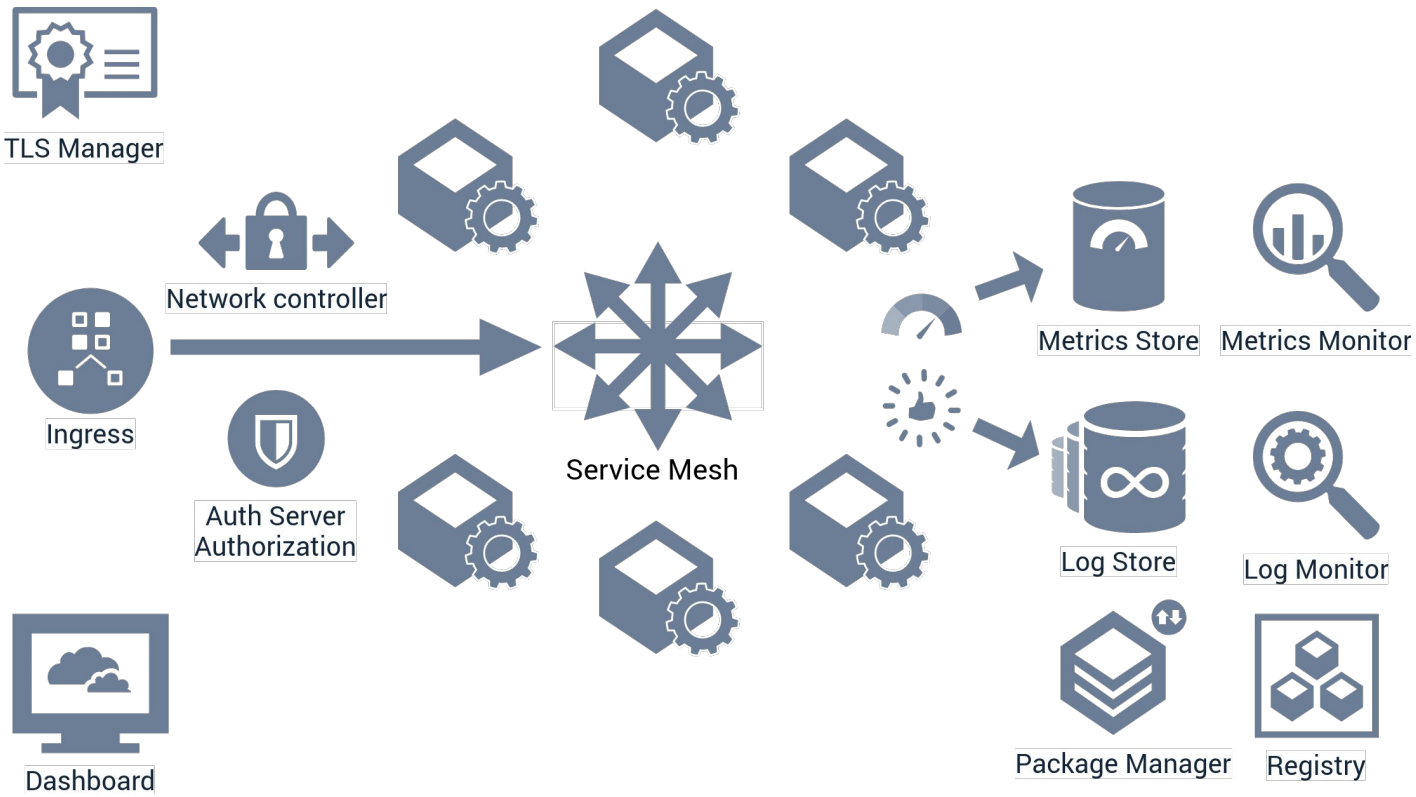

Ingress
Auth Server
Authorization
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor

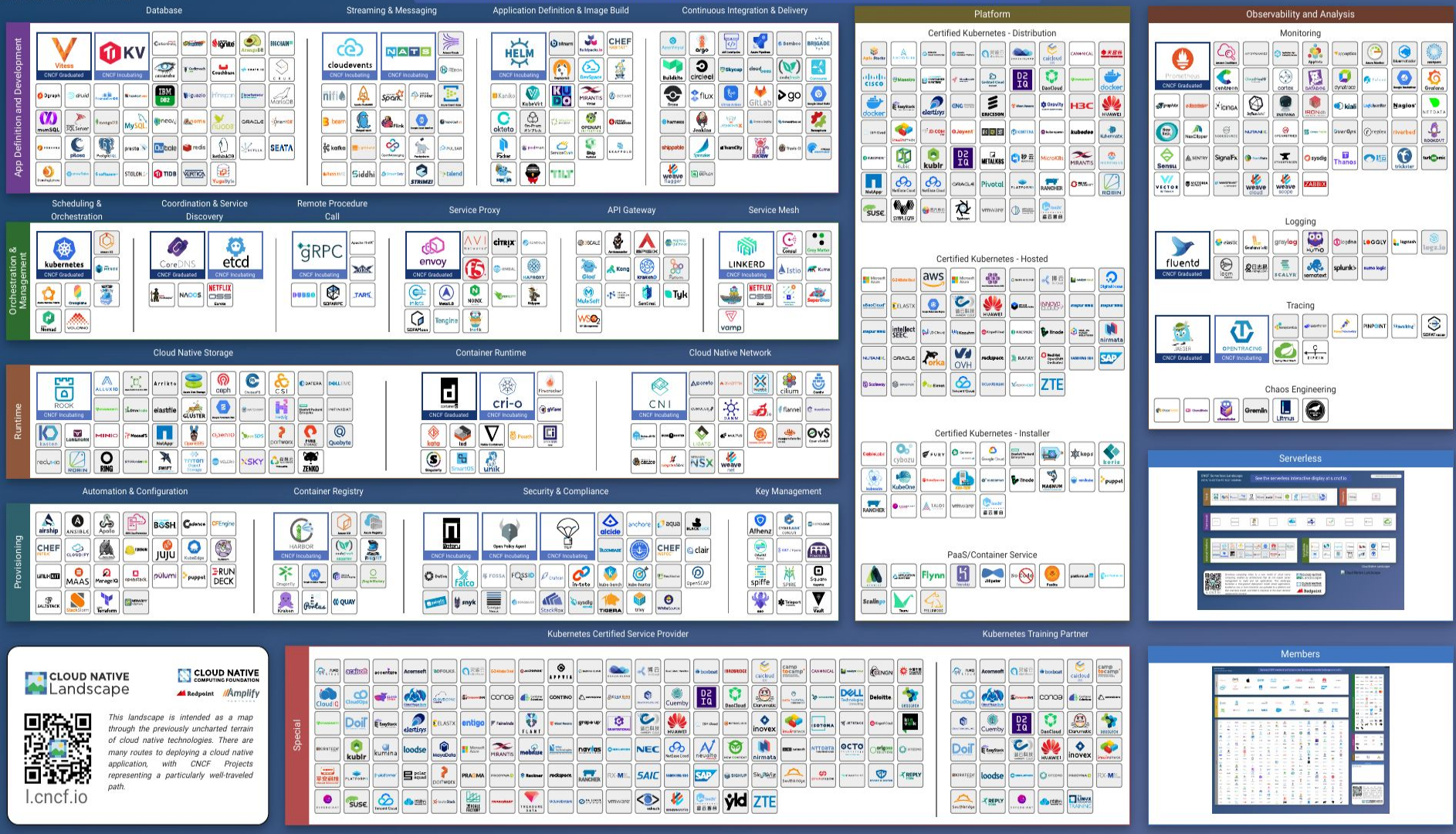

Network controller
Ingress
Auth Server
Authorization
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor

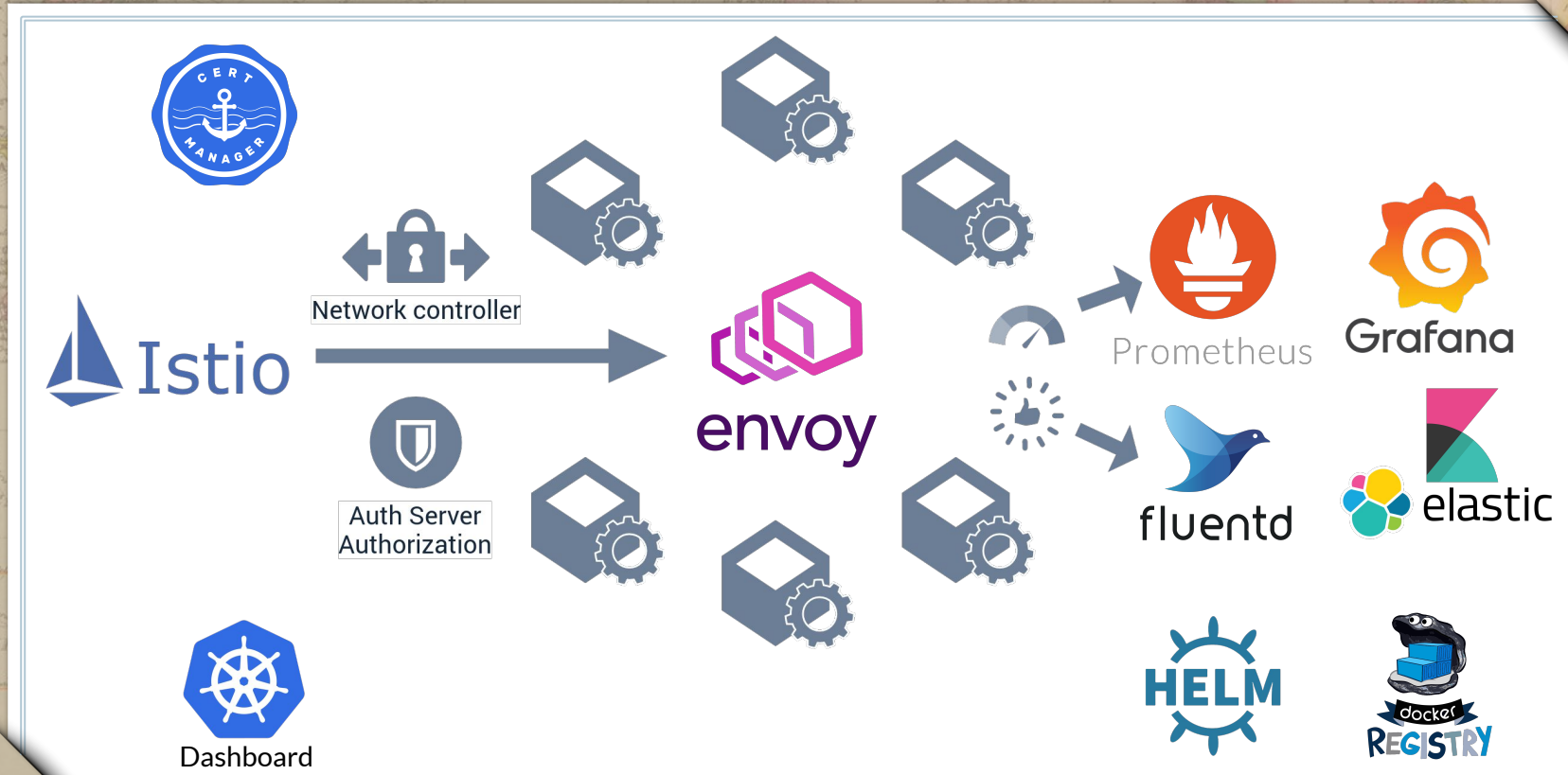

Network controller
Ingress
Auth Server
Authorization
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor
Registry


Network controller
Ingress
Auth Server
Authorization
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor
Package Manager
Registry


TLS Manager
Network controller
Ingress
Auth Server
Authorization
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor
Package Manager
Registry


TLS Manager
Network controller
Ingress
Auth Server
Authorization
Service Mesh
Metrics Store
Metrics Monitor
Log Store
Log Monitor
Package Manager
Registry
Dashboard


Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
App Definition and Development
Vitess
KV
cloudevents
NATS
HELM
CHEF
BRIGADE
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Orchestration & Management
kubernetes
CoreDNS
etcd
gRPC
envoy
LINKERD
Cloud Native Storage
Container Runtime
Cloud Native Network
Runtime
ROOK
cri-o
CNI
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Provisioning
HARBOR
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
Members
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io


CERT MANAGER
Network controller
Istio
Auth Server
Authorization
envoy
Prometheus
Grafana
fluentd
elastic
Dashboard
HELM
docker
REGISTRY

@alexivkinx

Declarative deployment

- You can find everything you need in K8s resources
- Kubernetes will do what you need for you
- It does not know what it does not know

No hidden APIs

API's, API's everywhere

- K8s master API, kubelet API, Application APIs
- RBAC and network policies are the only things between you and the cluster controls.

Meet the developer where they are at

- Secrets, config data and the application state are in env variables and folder mounts
- Legacy apps hastily ported to containers
- Security last

Pillage!

@alexivkinx

Pwnage now

Open ports / unauthed access

Supply chain for build layers, OSS Libs, registries

Container escape - Kernel boundary

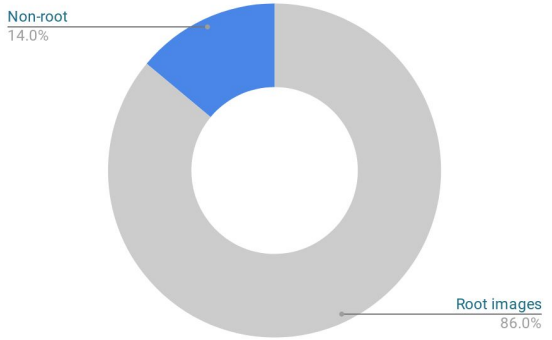
Network sniffing and MITM with eBPF

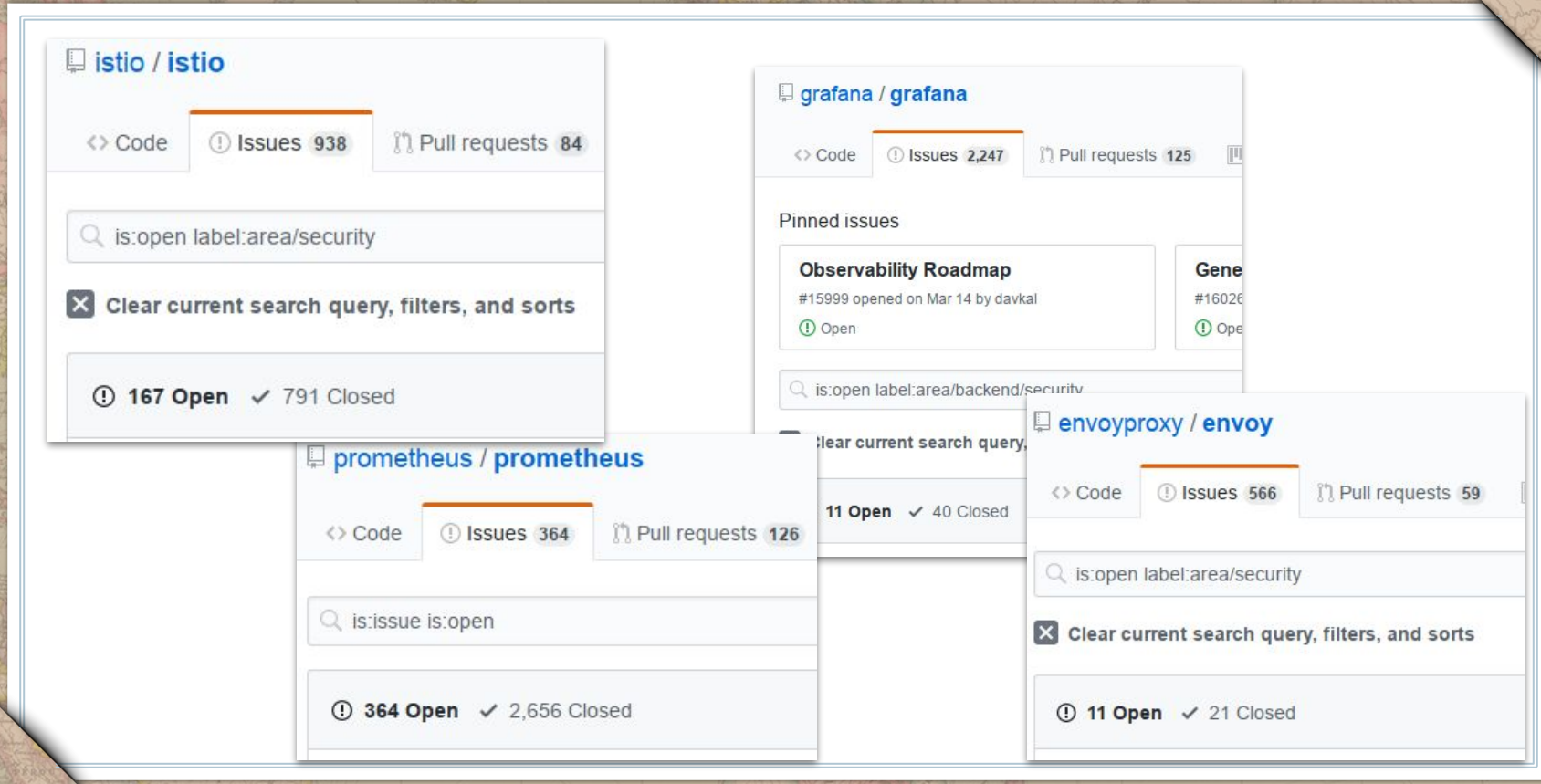
Node/pod keys

Privesc via secrets/daemonsets

Get a code exec in a pod

- Pentest
  - The app - SSRF is the best
  - The APIs
  -
- Look for known vulns in the k8s apps world




istio / istio
Code
Issues 938
Pull requests 84
is:open label:area/security
Clear current search query, filters, and sorts
167 Open
791 Closed
grafana / grafana
Code
Issues 2,247
Pull requests 125
Pinned issues
Observability Roadmap
#15999 opened on Mar 14 by davkal
Open
is:open label:area/backend/security
11 Open
40 Closed
prometheus / prometheus
Code
Issues 364
Pull requests 126
is:issue is:open
364 Open
2,656 Closed
envoyproxy / envoy
Code
Issues 566
Pull requests 59
is:open label:area/security
Clear current search query, filters, and sorts
11 Open
21 Closed

Kubernetes Network Plugins

	Calico	Canal	Cilium	Flannel	Kube-router	WeaveNet
Encryption	No	No	Yes	No	No	Yes
Network Policies	Ingress Egress	Ingress Egress	Ingress Egress	None	Ingress	Ingress Egress

Ingress controllers

	Ingress-nginx	Nginx-ingress	Istio	Ambassador	Traefik	Gloo
Authentication	Yes	No	mTLS, OpenID	Basic, OAuth	Basic,URL	
JWT Support	No		Yes		No	
WAF	ModSecurity		ModSecurity	No	No	

https://docs.google.com/spreadsheets/d/1DnsHtdHbxjvHmxvlu7VhzWcWgLAn_Mc5L1WlhLDA__k/edit#gid=0

Real world Kubernetes

- Security Misconfiguration is the King
  - Broken Authentication, Access Control
  - Insufficient Logging and Monitoring
  - Service accounts/secrets distributions
  - mTLS and In-transit encryption
- Example - Pod Security Policies are dead

What do?

Easy

- Use simple images to start from or grab and reconstruct dockerfile and build the image yourself
- No privileged containers or shared volumes with the node
- Monitor for rogue containers
- In the cloud - secure metadata
- On baremetal - use hardened OS (Flatcar, Minimal Debian, Alpine with CRI)
- Use RBAC (duh!)

https://github.com/alexivkin/docker-historian

Normal

- Build from scratch or use distroless images
- Use PodSecurity Admission Controller
- Have multiple registries and authenticate/authorize access
- Remap root to non-root users for all containers
- Upgrade master nodes and all nodes to the latest version
- SAST/DAST/SCA for images
- Ship out K8s secrets to a vault

Hardmode

- Use Pod Admission Policies
- Aim for zero-trust (mTLS, SPIFFE/SPIRE)
- Sign your images
  - Use TUF/Notary to address supply chain risks
- Confidential nodes/VM sandboxing
  - Container separation is weaker than hypervisors
- Unmix sensitive workloads
  - Namespaces will not provide hard multitenancy
    - Use separate clusters

Security best practices, frameworks and standards

- CTFs and playgrounds
- CIS Benchmarks for Docker and Kubernetes
- Aqua, StackRox, SysDig, Tigera, Neuvector whitepapers
- Night time reading
  - US DoD DevSecOps reference design
  - NIST SP 800-190 - Compliance in Container Environments
  - SANS Checklist for Audit of Docker containers
- https://github.com/alexivkin/containerpwn
- https://github.com/alexivkin/kubepwn

It’s gonna get more complex

Errors, not vulnerabilities, lead to a lot of breaches

Make it secure by default

https://github.com/alexivkin

@alexivkinx

in/alexivkin