



Practical security in the brave new Kubernetes world

Me harty's guide
fer all ye scallywags



The modern application stack



Application



Orchestrator



Container

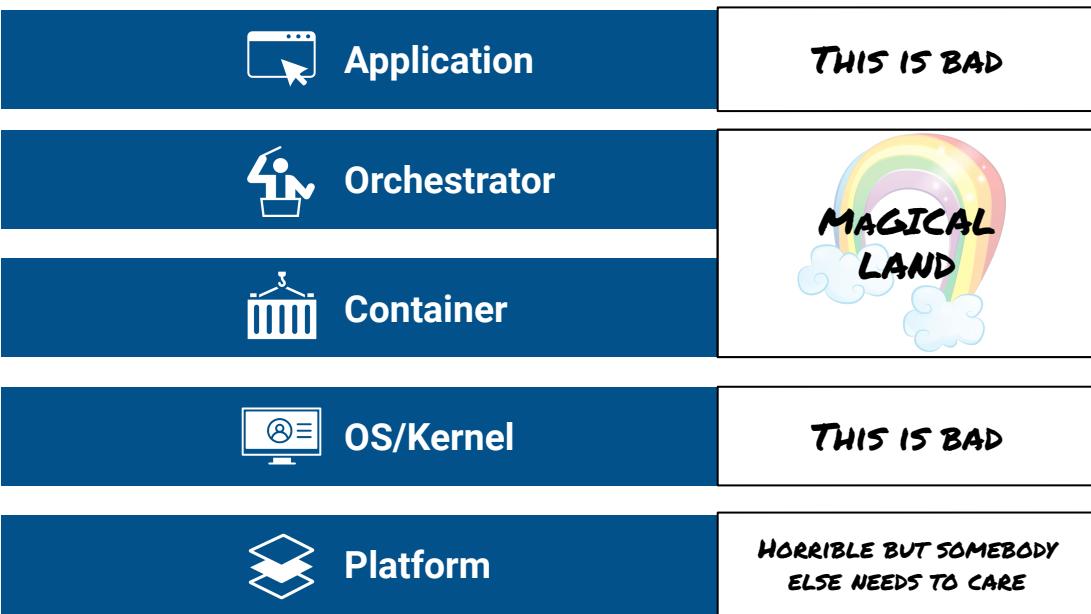


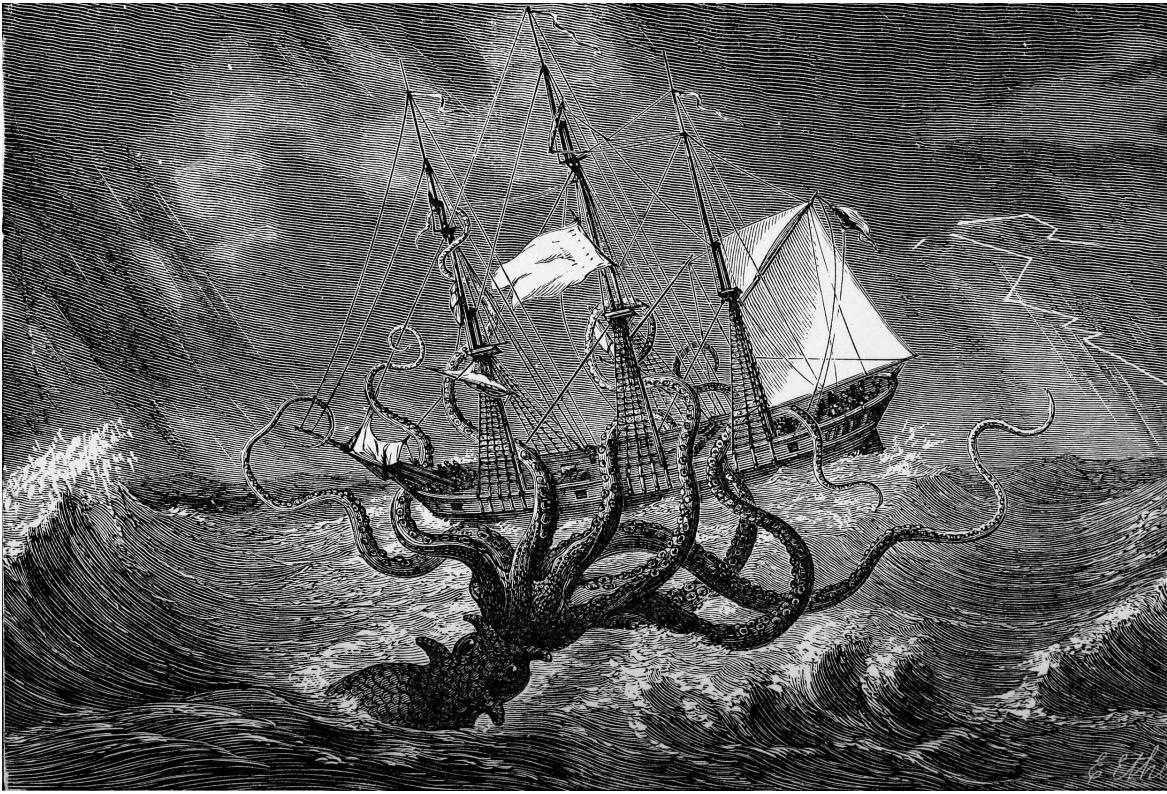
OS/Kernel



Platform

The modern application stack







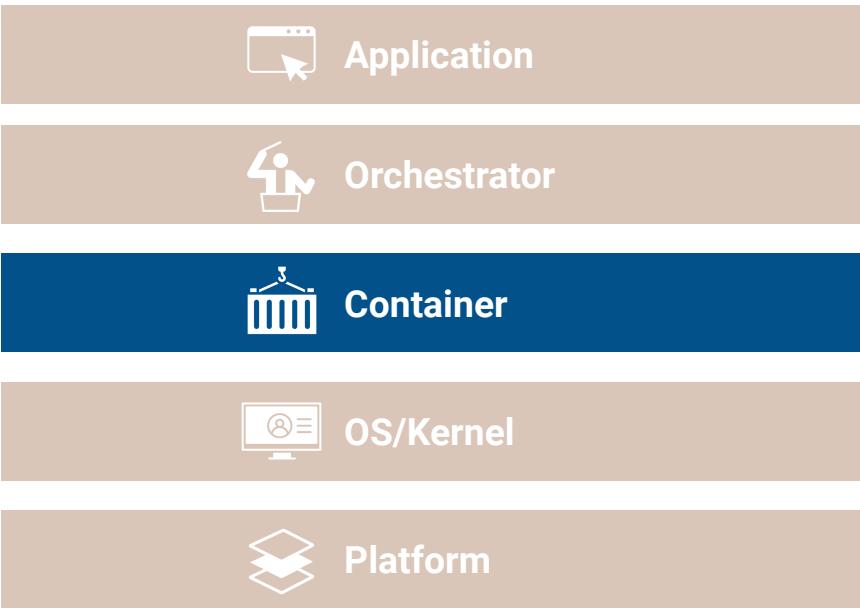
Ahoy!

I am **Alex Ivkin**, Director of Solutions at Eclipsium, a US firmware security company, from Portland, Oregon.

I do secure deployments in (in)secure environments, including container orchestration, application security, and firmware security.



The modern application stack



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

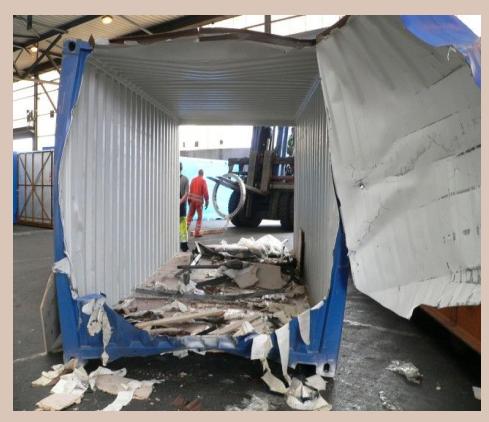
- Persistence
- Tooling/living-of-the-land
- Path traversal
- Uncontrolled resource consumption/DoS



Where containers will not help

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

- Injection/Insecure Deserialization/RCE
- Improper input validation
- Use-after-free, TOCTOU, overflow
- XSS, CSRF, SSRF



But wait there's more!

- Software supply chain, componentes with known vulnerabilities and updating these
- iptables mess
- Kernel 'exploits
- Change docker defaults, get yourself in trouble



'ow do ye ship yer containers?



The modern application stack



Application



Orchestrator



Container



OS/Kernel



Platform

Infrastructure as a Code



Chef

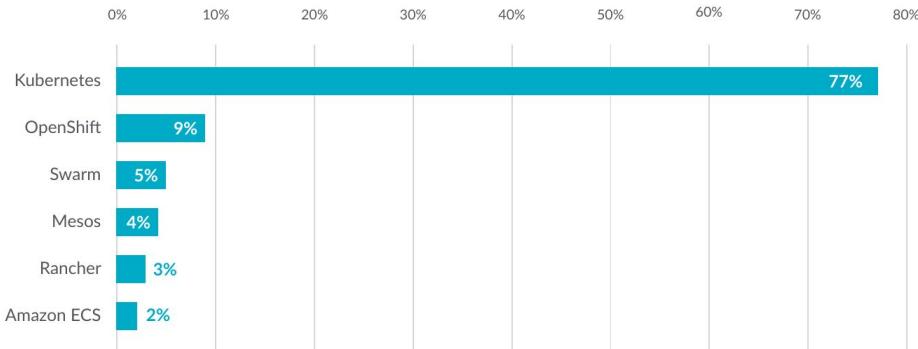


Puppet



Terraform

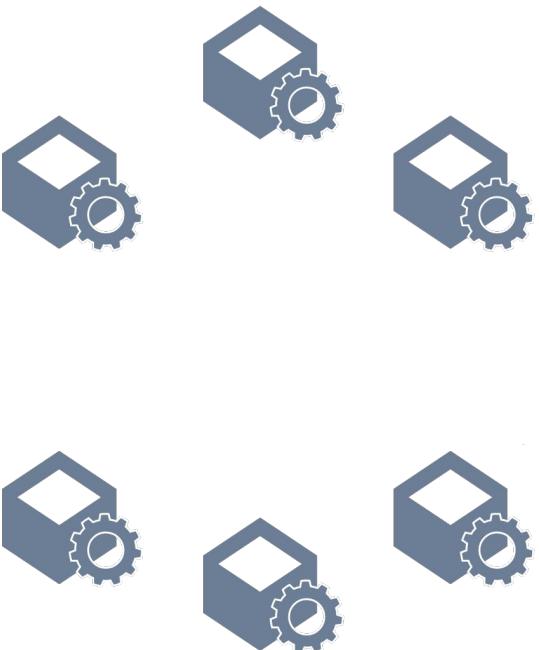
Kubernetes!

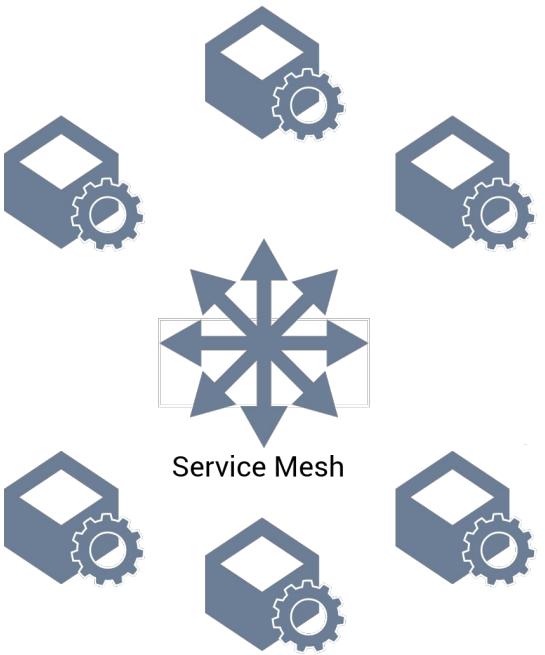


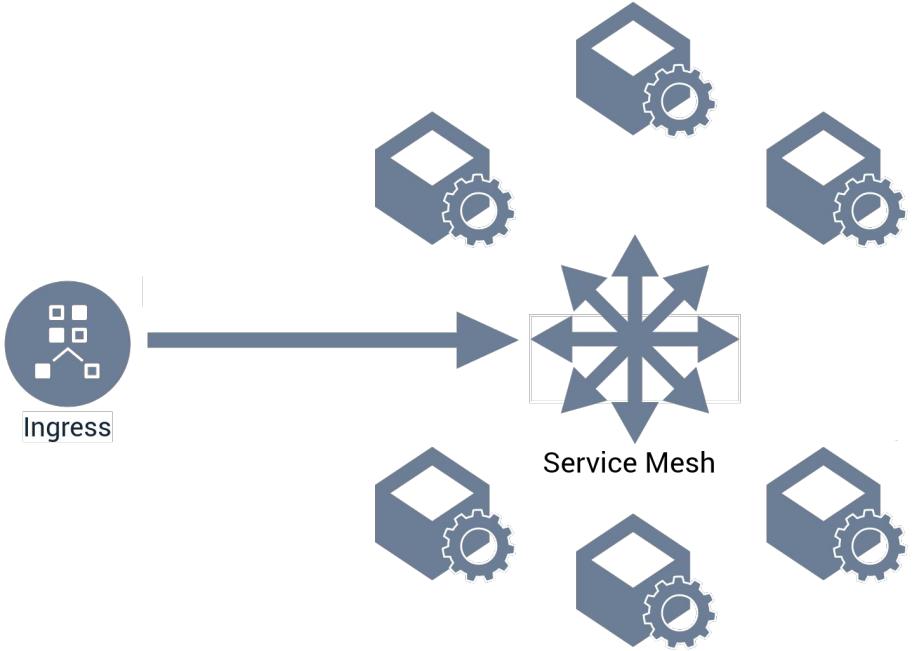
<https://sysdig.com/blog/sysdig-2019-container-usage-report/>

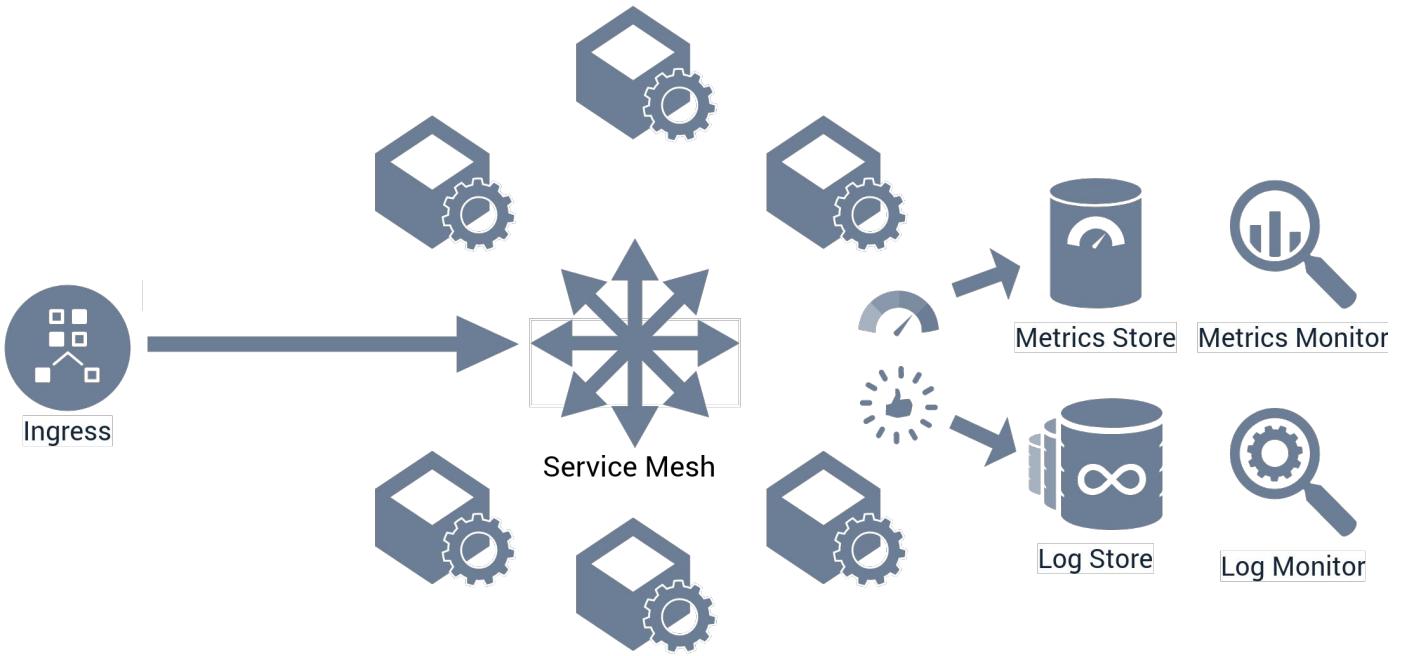
Core K8s concepts

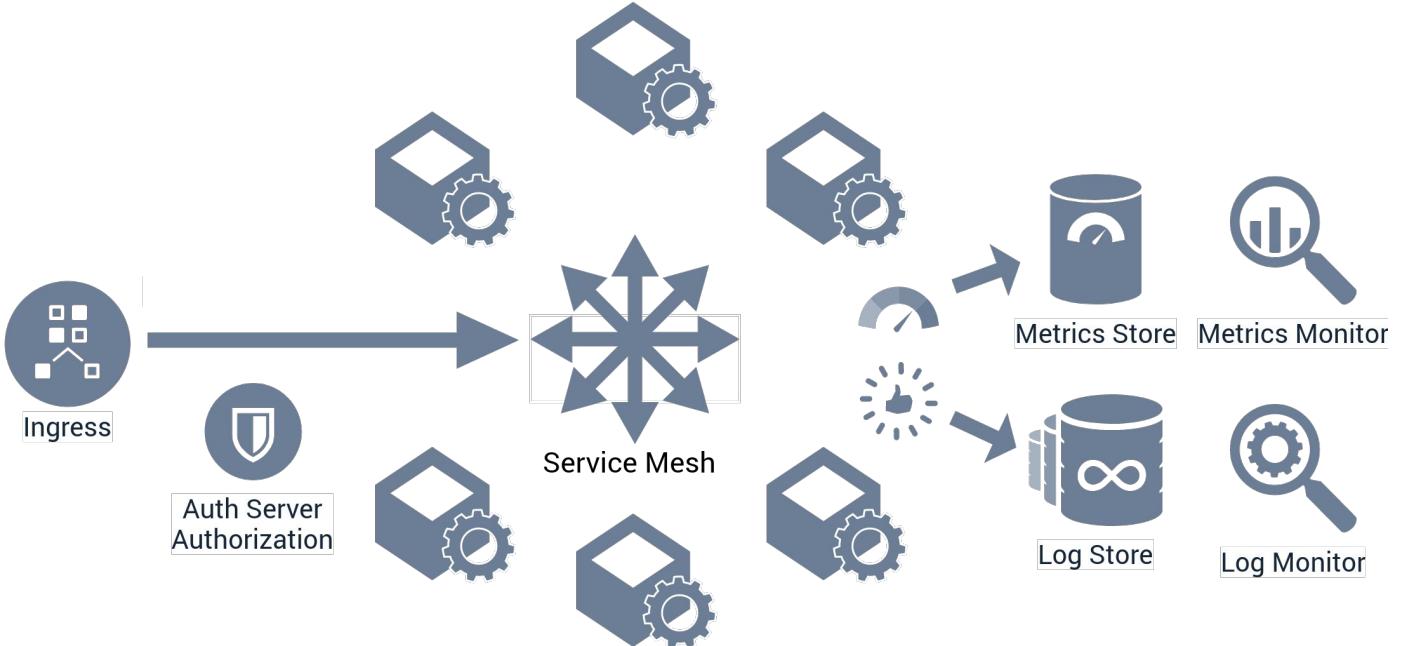
1. Declarative deployment
2. No hidden APIs
3. Meet the developer where they are at

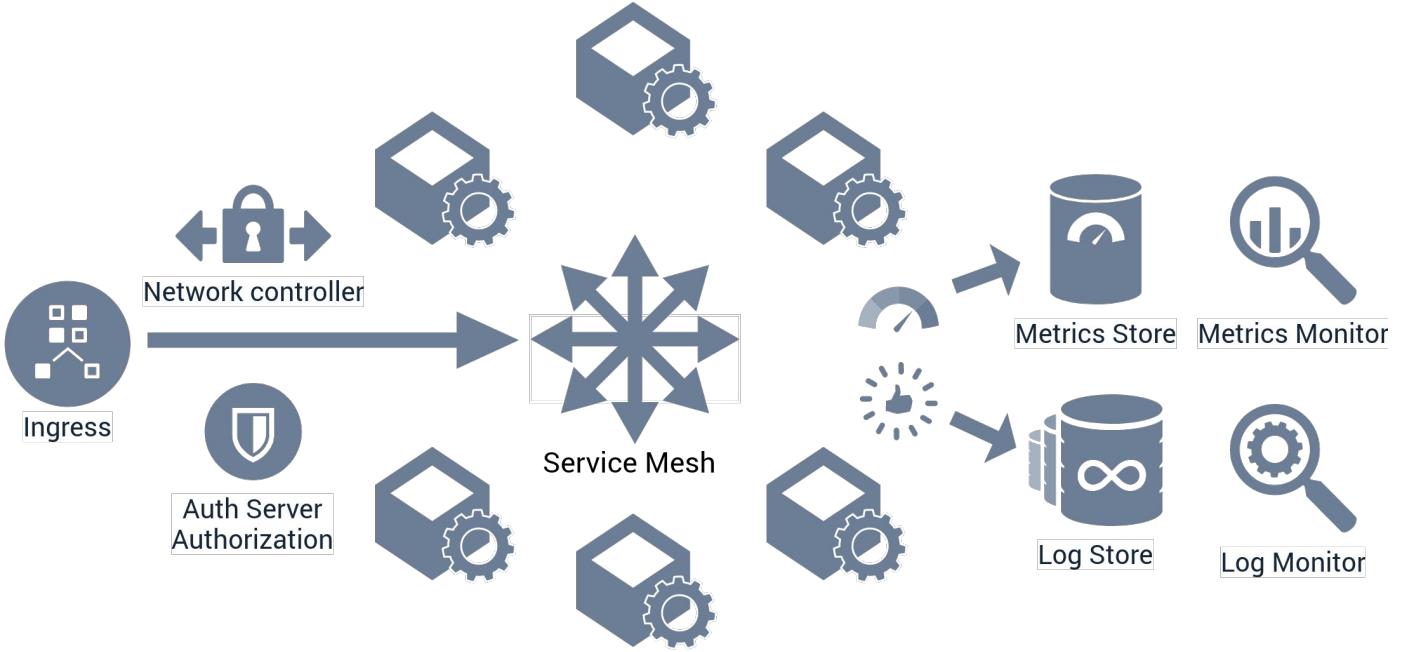


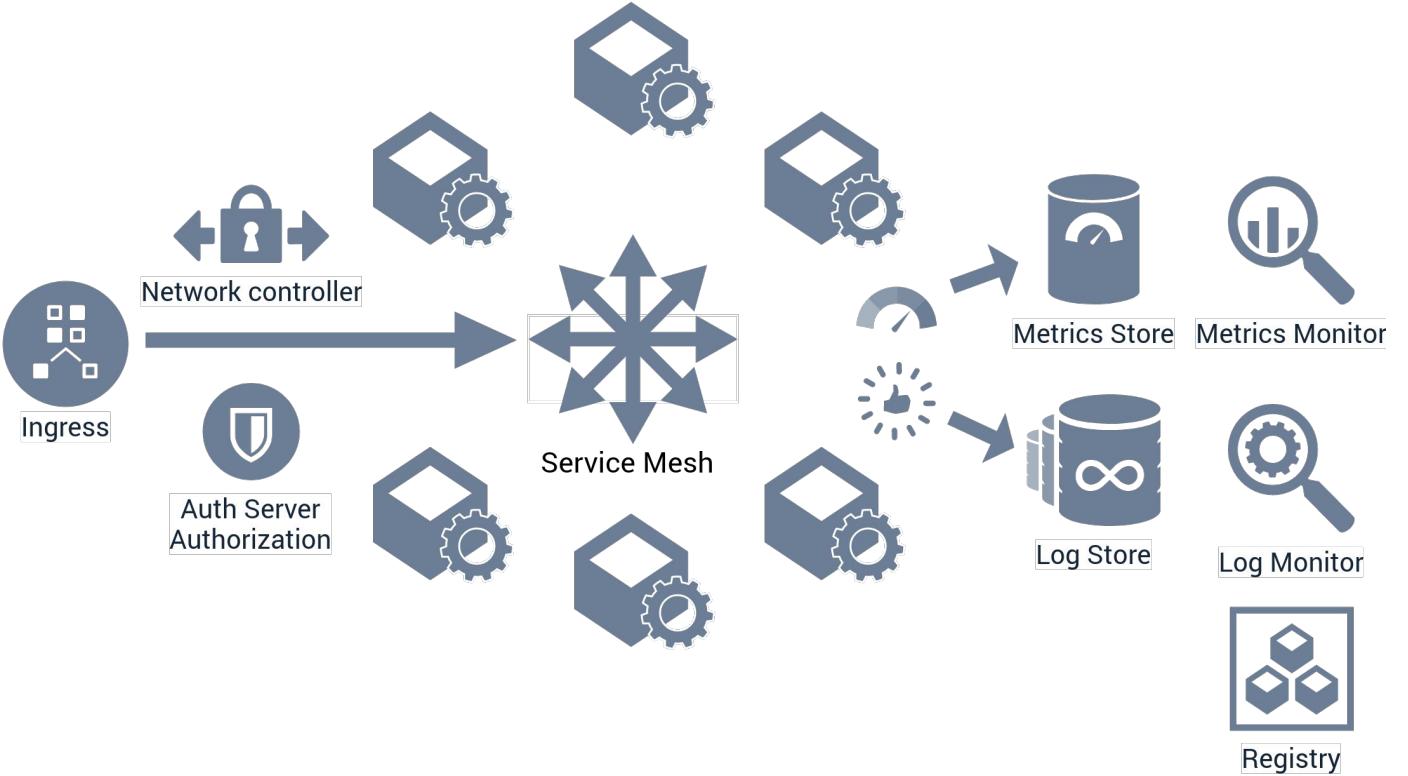


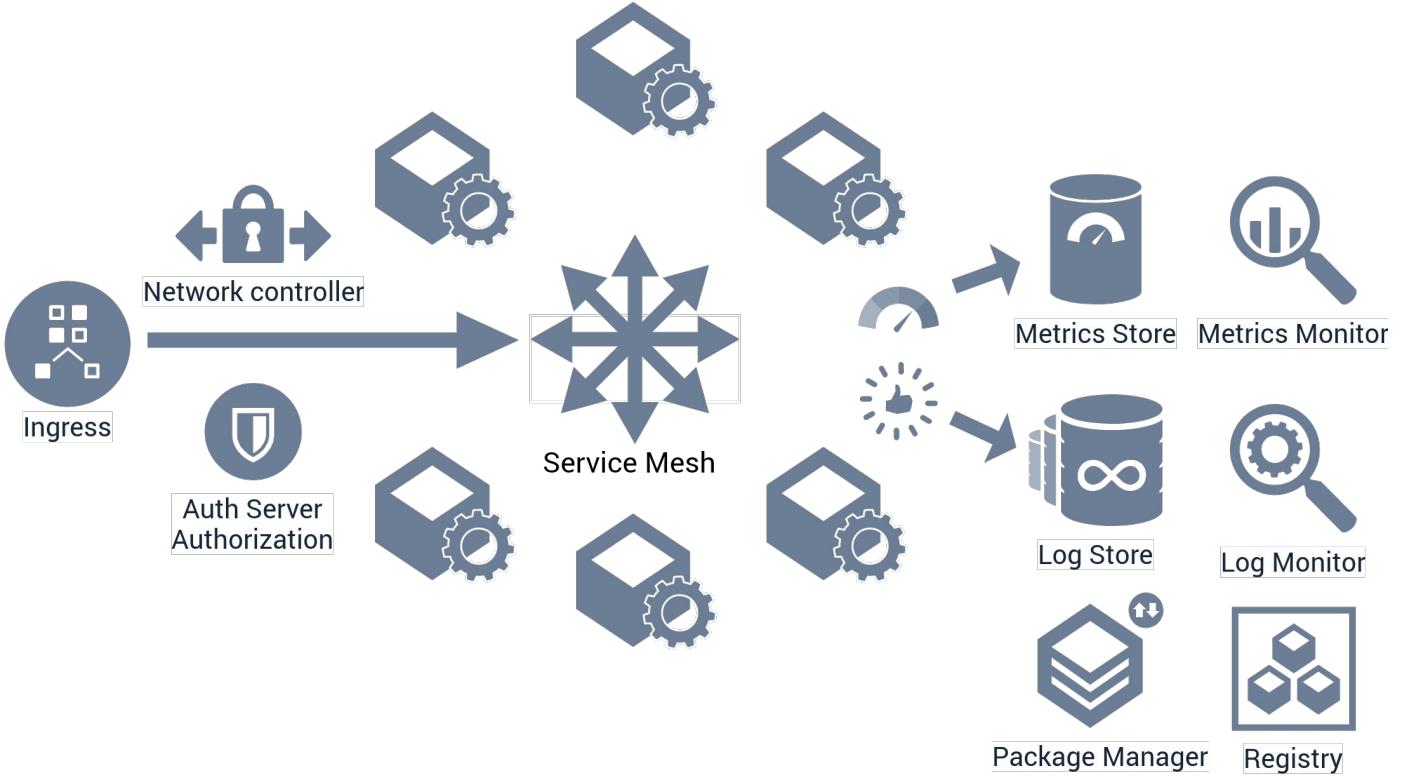














TLS Manager

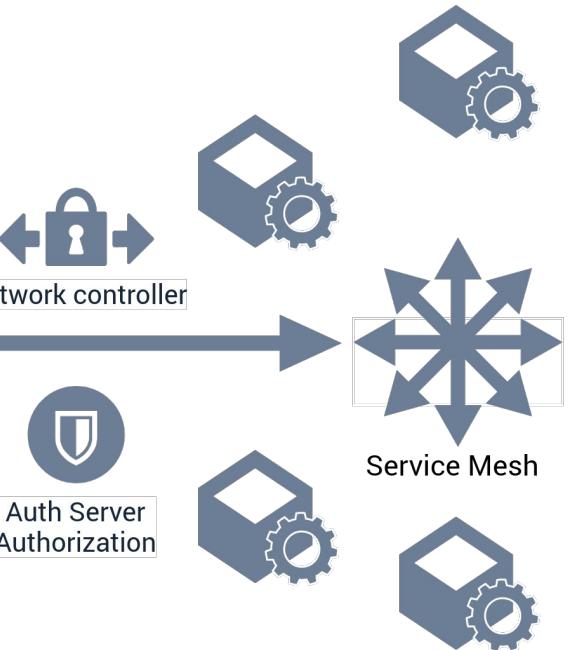


Ingress

Network controller



Auth Server
Authorization



Metrics Store



Metrics Monitor



Log Store



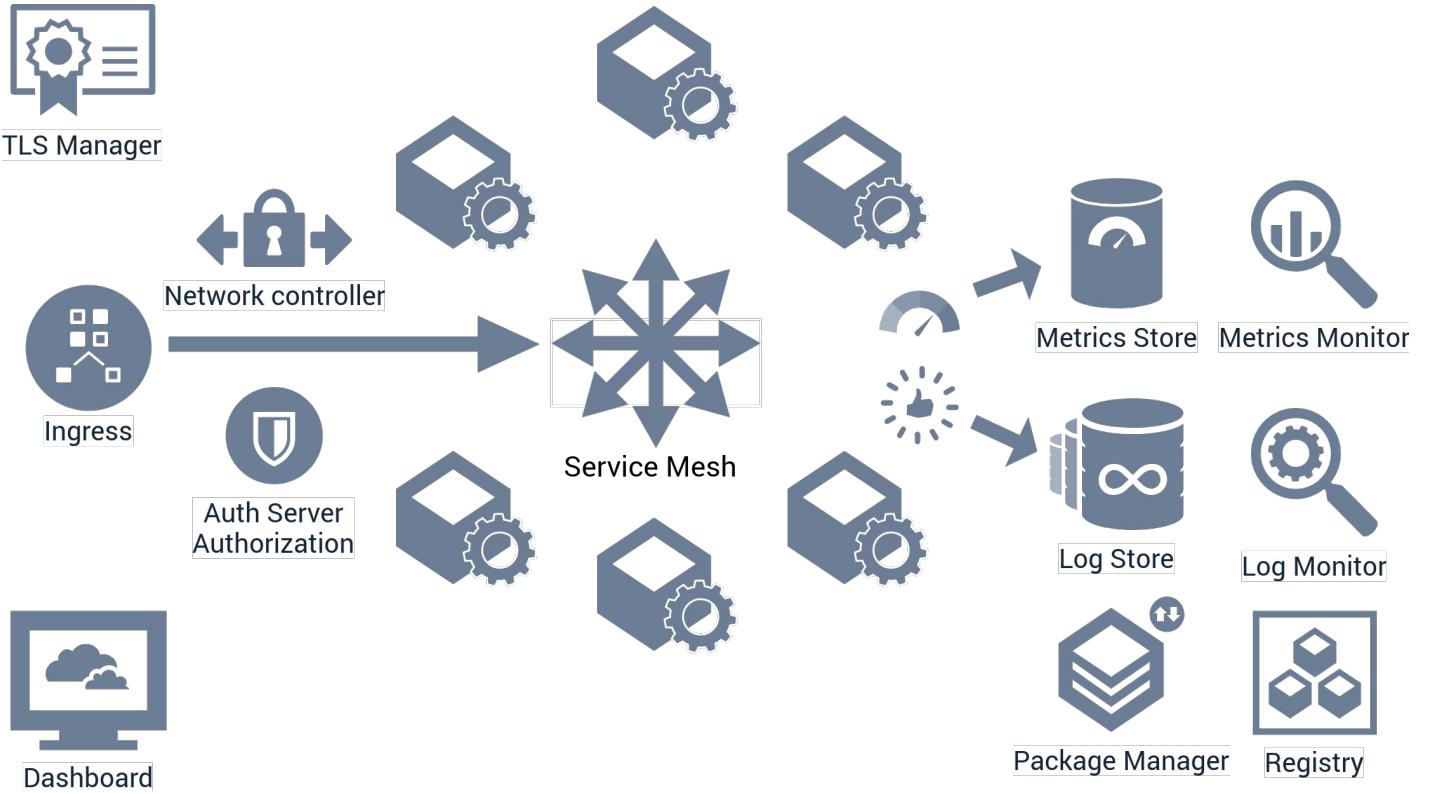
Log Monitor

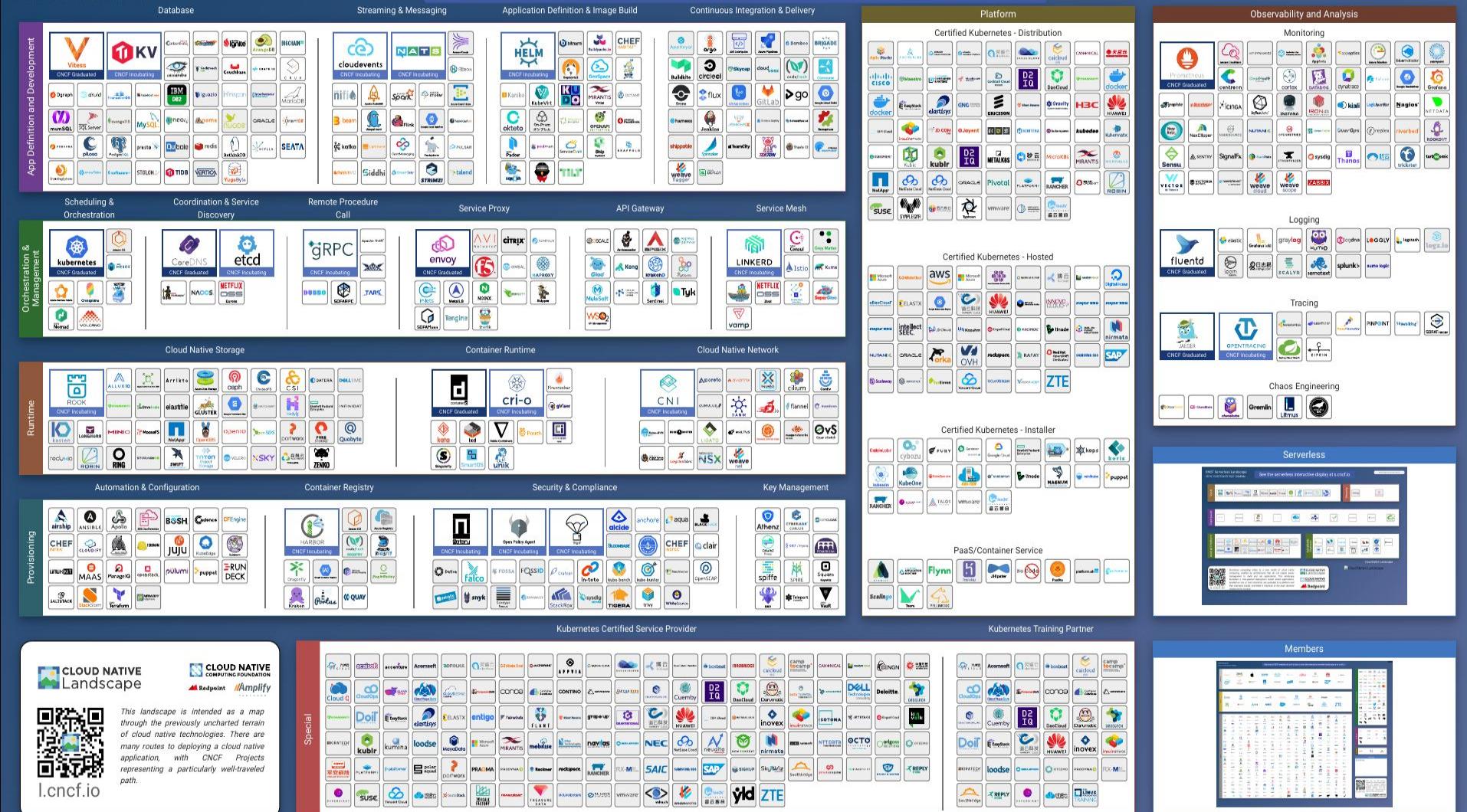


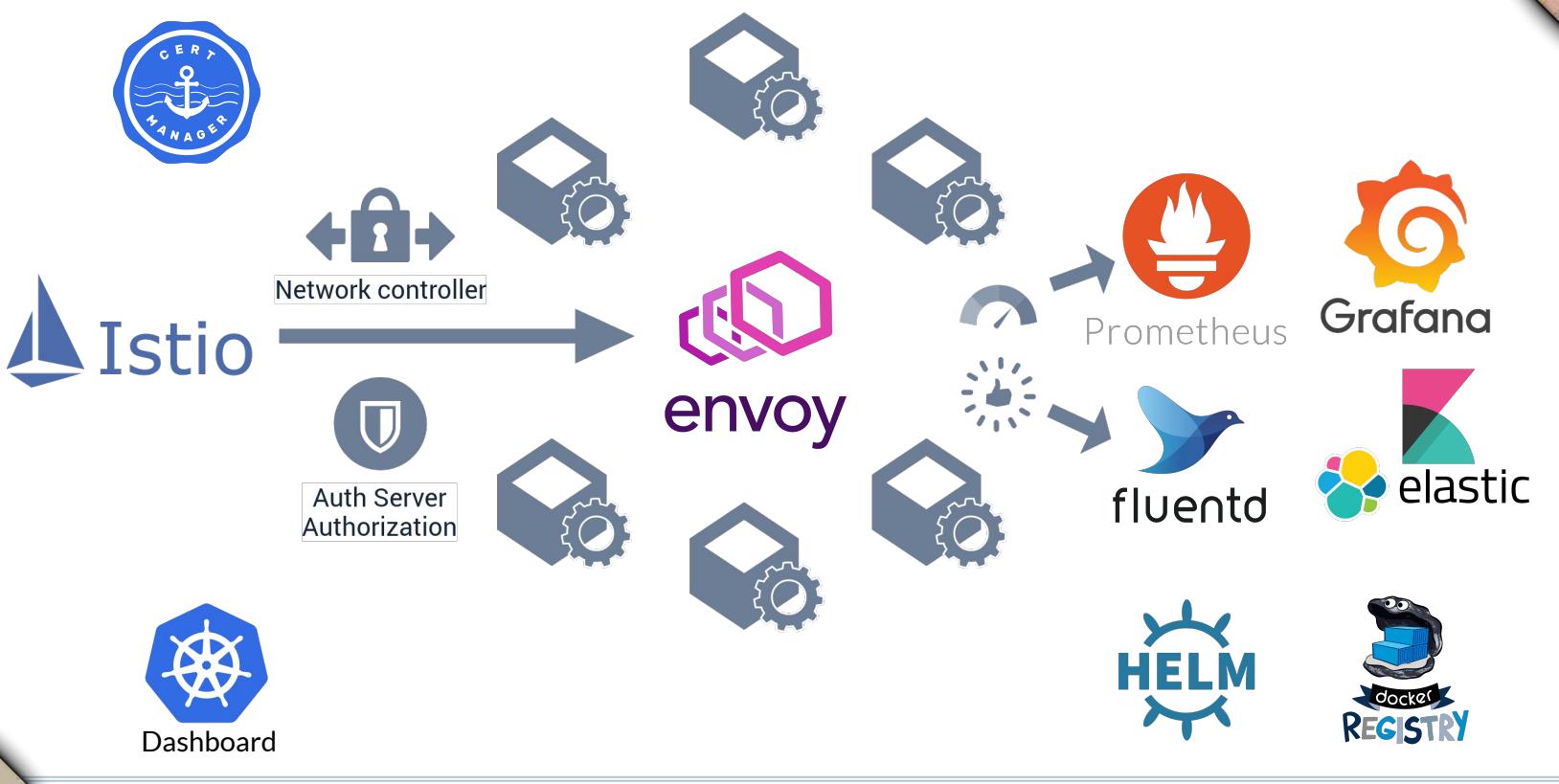
Package Manager



Registry









@alexivkinx

cc by 4

Declarative deployment

- You can find everything you need in K8s resources
- Kubernetes will do what you need for you



No hidden APIs

API's, API's everywhere

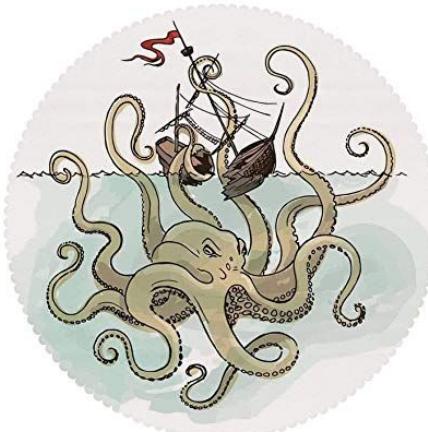
- K8s master API, kubelet API, Application APIs
- RBAC and network policies are the only things between you and the cluster controls.



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
- An average dev is not that good with security

Pillage!

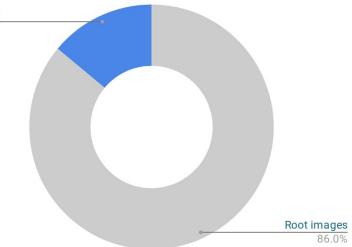
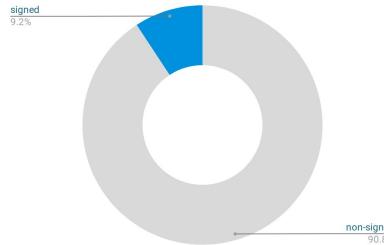


cc by 4

@alexivkinx

Get a code exec in a pod

- Pentest
 - The app - SSRF is the best
 - The APIs
 - No need for the privesc
- Supply chain - pre-implant the image
- Look for known vulns in the k8s 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 davikal
Open

Gene #16026 Open

🔍 is:open label:area/backend/security

✖ Clear current search query, filters, and sorts

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

- Broken Authentication, Access Control
- Security Misconfiguration
- Insufficient Logging and Monitoring
- Service accounts/secrets distributions
- mTLS and In-transit encryption



What do I 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, use COS/Bottlerocket
- 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 Policies
- 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
- Unmix sensitive workloads
 - ➔ Container separation is weaker than hypervisors
 - ➔ 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 

