



Kubernetes Runtime Security

What happens if a container goes bad?



About us

Jen Tong

Security Advocate

 [@MimmiingCodes](https://twitter.com/MimmiingCodes)

Maya Kaczorowski

Product Manager, Security

 [@MayaKaczorowski](https://twitter.com/MayaKaczorowski)





How many of you...

- ...are familiar with the NIST cybersecurity framework?
- ...have had a container incident?
- ...are monitoring containers for security issues?

Agenda

- 1 What is container security
- 2 Why containers are different from VMs
- 3 How to detect bad things at runtime
- 4 Demo

What is container security





**Kubernetes is so new that
lots of practitioners don't
know what security
controls come with it.**

So one of the first things to do is
study up on what controls are there
and use them to strengthen your
security posture

– *Chenxi Wang, Jane Bond Project*

What kinds of threats are there to containers?

LILY HAY NEWMAN SECURITY 02.20.18 05:06 PM

HACK BRIEF: HACKERS ENLISTED TESLA'S PUBLIC CLOUD TO MINE CRYPTOCURRENCY



- Hackers accessed the Kubernetes console, which was **not password protected**
- Console contained **privileged AWS account credentials**
- Used credentials to access AWS resources and **mine cryptocurrency**

What kinds of threats are there to containers?

Is my infrastructure **secure for developing** containers?

- Kubernetes API compromise
- Privilege escalation
- Credential compromise

Is my container image **secure to build and deploy?**

- Unpatched vulnerability
- Supply chain vulnerability
- Zero day exploit on common library

Is my container **secure to run?**

- DDoS
- Node compromise and exploit
- Container escape
- Flood event pipeline

So, what is container security?

Infrastructure security

Is my infrastructure **secure for developing** containers?

Software supply chain

Is my container image **secure to build and deploy?**

Runtime security

Is my container **secure to run?**

So, what is container security?

 Focus of today's talk

Infrastructure security

Is my infrastructure **secure for developing** containers?

Software supply chain

Is my container image **secure to build and deploy?**

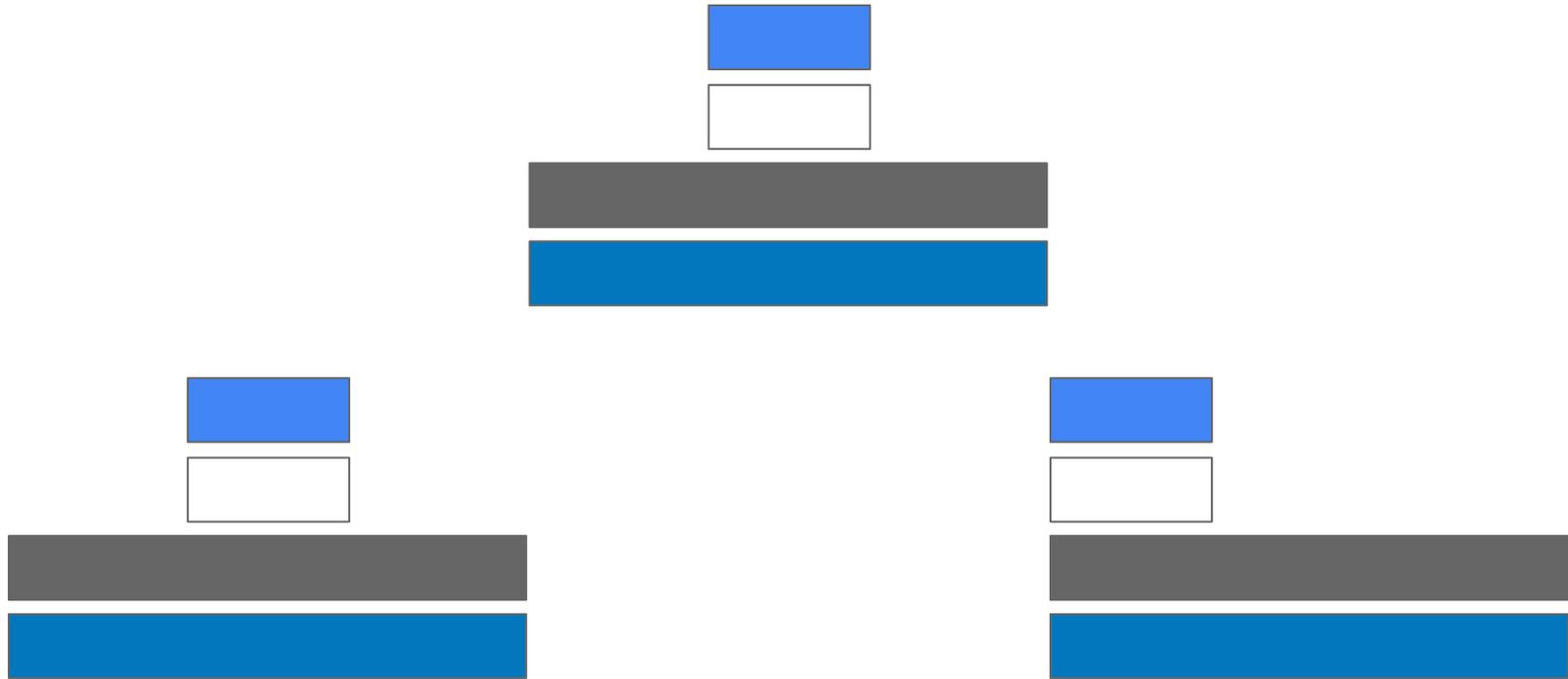
Runtime security

Is my container **secure to run?**

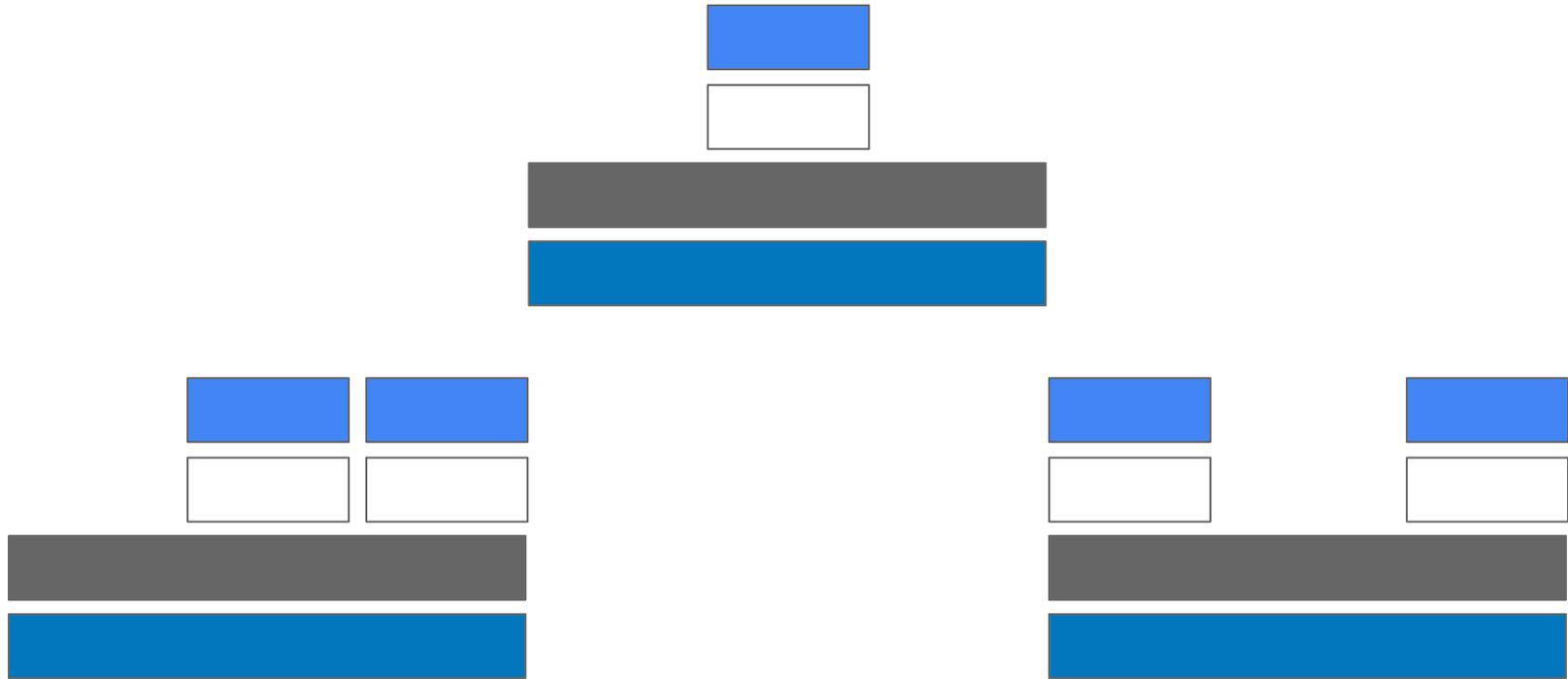
Why containers are different from VMs

Containerization changes some things

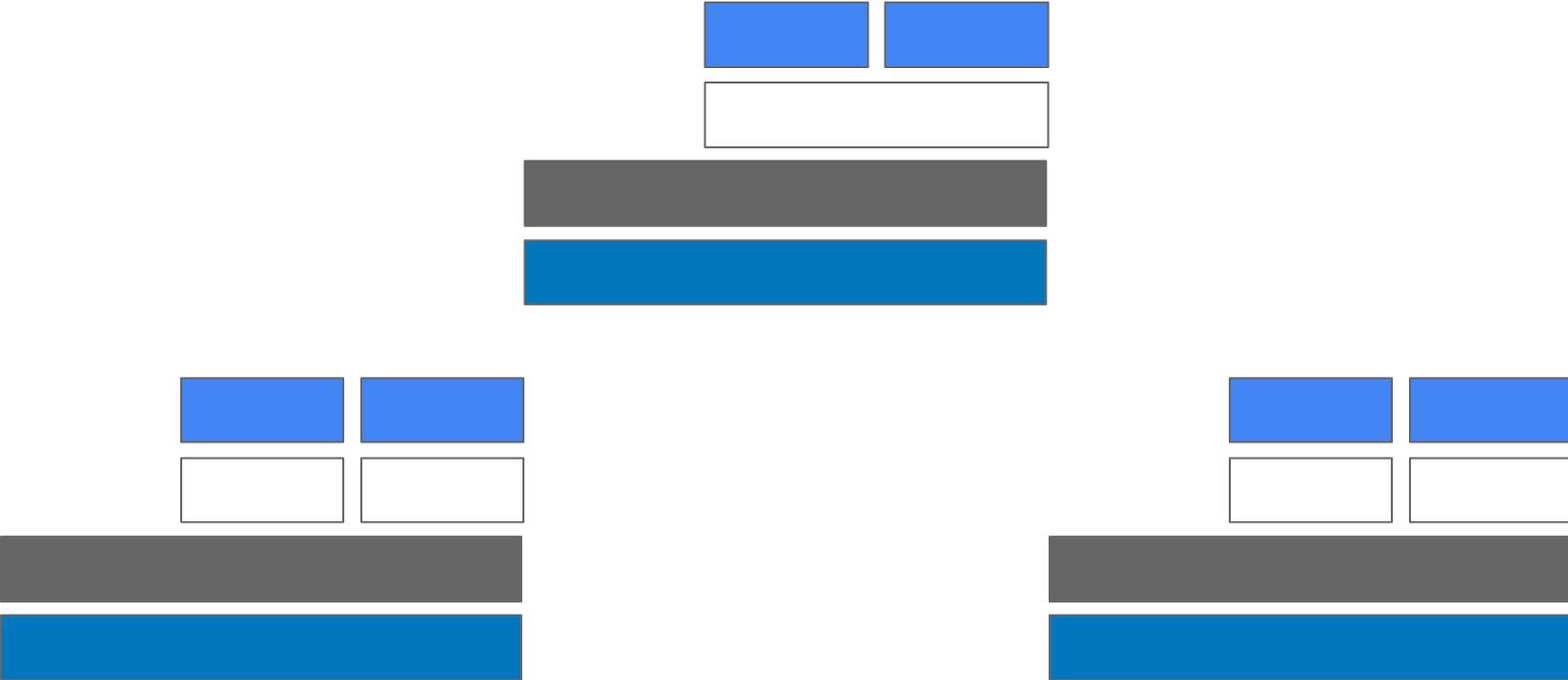
Containers are dynamic



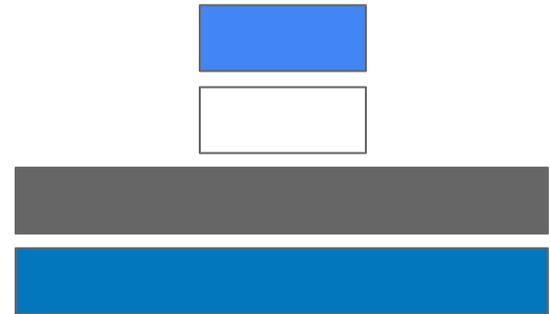
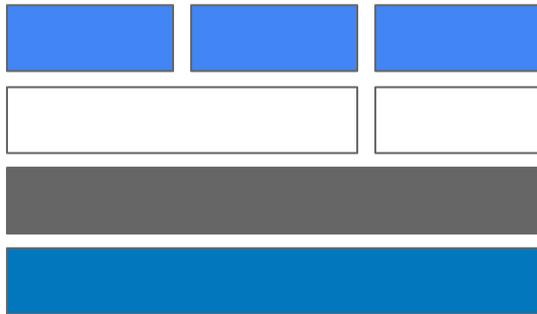
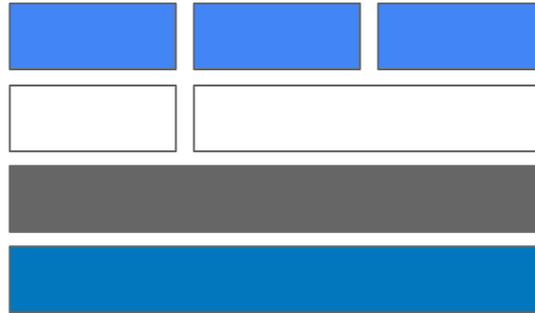
Containers are dynamic



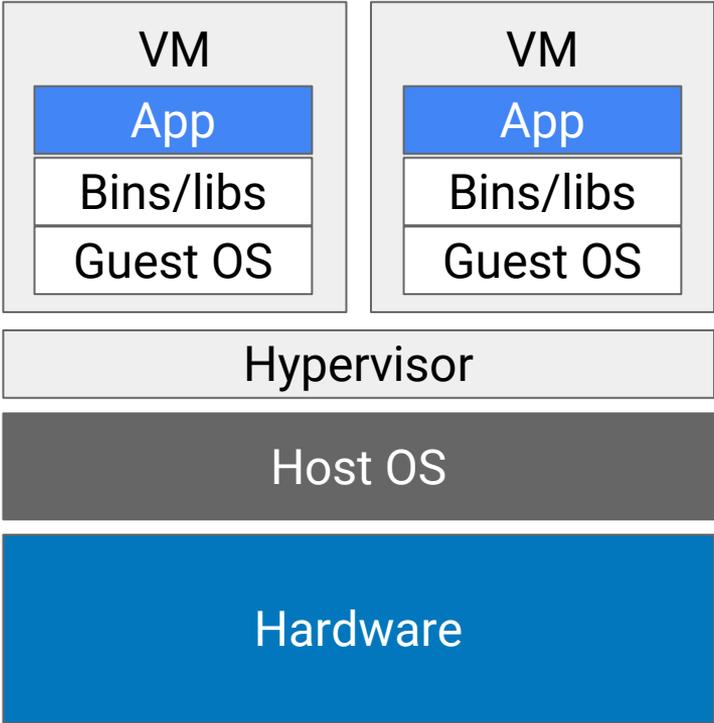
Containers are dynamic



Containers are dynamic

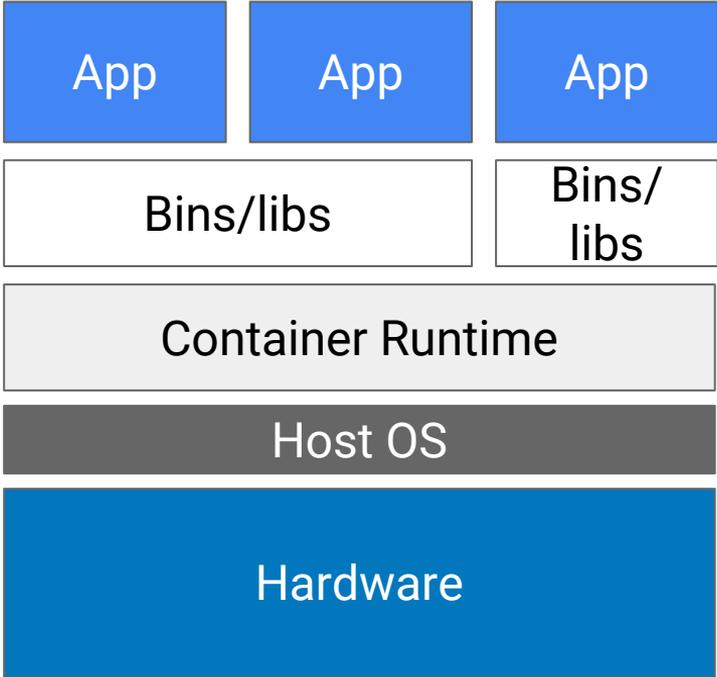


Virtual machine



vs

Container



How is securing a container different than a VM?

How containers help

Surface of attack

Minimalist host OS limits the surface of attack

Resource isolation

Host resources are **separated using namespaces and cgroups**

Root permissions

Access controls for app privileges and shared resources

Lifetime

Containers have a **shorter average lifetime**

How containers hurt

Hypervisors are a strong security boundary

Host resources are **not all well separated**

Containers have access to **wider set of syscalls** to the kernel

It's **harder to do forensics** on a container that isn't there

... but it's more the same than different

How to detect bad things at runtime

Why bother?

My secure supply chain prevents vulnerabilities!

But...

- Incomplete vuln scans
- Misconfigurations
- Zero days

Software supply chain is not perfect.

A fence is better than tall fence posts

NIST cybersecurity framework

Identify

Asset management, business environment, governance, risk assessment, risk management

Protect

Access control, awareness, data security, information protection processes, maintenance, protective technology

Detect

Anomalies and events, continuous monitoring, detection process

Respond

Response planning, communications, analysis, mitigation, improvements

Recover

Recover planning, improvements, communications

NIST cybersecurity framework

Identify

Protect

Detect

Respond

Recover

NIST cybersecurity framework

Identify

Know what your containers are

Protect

Detect

Respond

Recover

NIST cybersecurity framework

Identify

Know what your containers are

Protect

Use secure defaults to protect your containers

Detect

Respond

Recover

NIST cybersecurity framework

Identify

Know what your containers are

Protect

Use secure defaults to protect your containers

Detect

Detect container behaviour that deviates from the norm

Respond

Recover

NIST cybersecurity framework

Identify

Know what your containers are

Protect

Use secure defaults to protect your containers

Detect

Detect container behaviour that deviates from the norm

Respond

Respond to a suspicious event in your container and mitigate the threat

Recover

NIST cybersecurity framework

Identify

Know what your containers are

Protect

Use secure defaults to protect your containers

Detect

Detect container behaviour that deviates from the norm

Respond

Respond to a suspicious event in your container and mitigate the threat

Recover

Complete forensics and fix things so this doesn't happen to your container again

NIST cybersecurity framework

Identify

Know what your ~~e~~containers assets are

Protect

Use secure defaults to protect your ~~e~~containers applications

Detect

Detect ~~e~~container behaviour that deviates from the norm

Respond

Respond to a suspicious event ~~in-your~~ ~~e~~container and mitigate the threat

Recover

Complete forensics and fix things so this doesn't happen ~~to-your-e~~container again

Detect: How does container monitoring work?

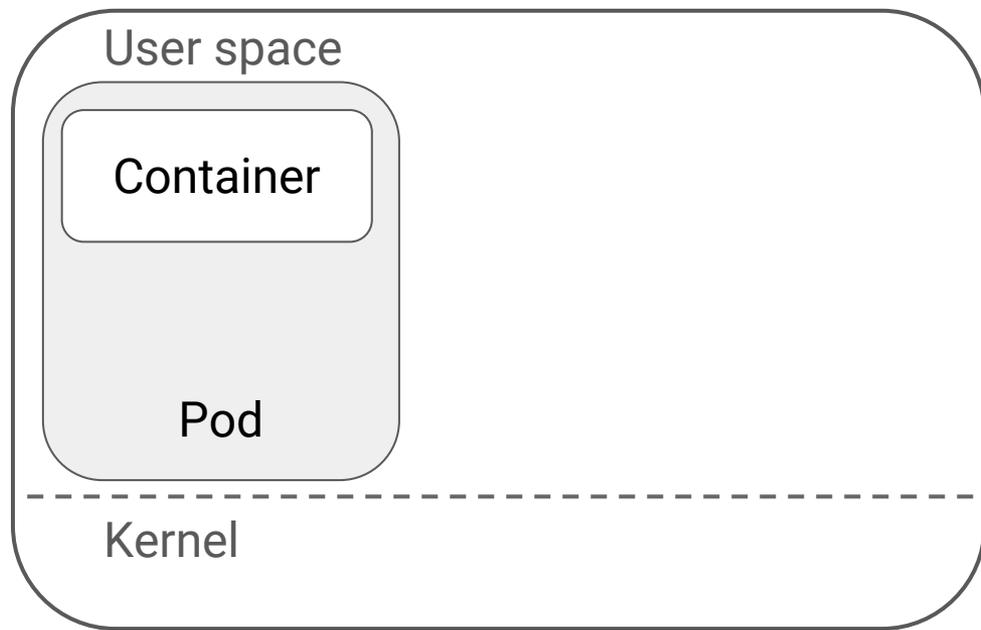
- Hook (something) into your container
- **Log** a bunch of stuff
- **Set policies** for:
 - alerts
 - automatic remediation
- **Do forensics** afterwards

Detect options

Examine process activity, network activity, file activity, ... **HUGE VOLUME**

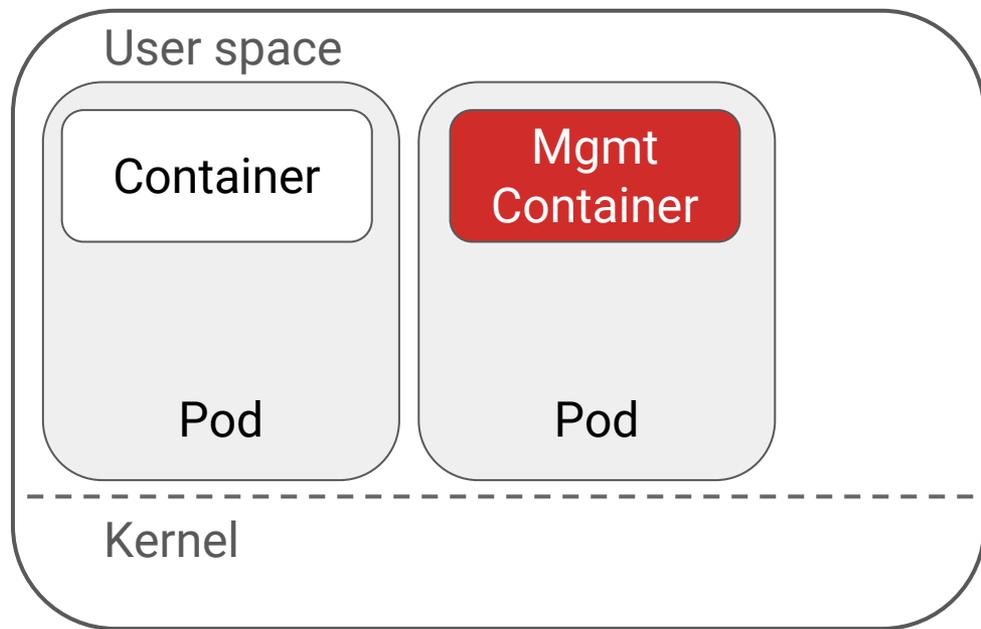
- **ptrace, kprobes, tracepoints**
- **Audit logs**
- **eBPF**: kernel introspection
- **XDP**: uses eBPF for filtering network packets
- **User-mode API**: for kernel events like inotify

Detect: Deployment models



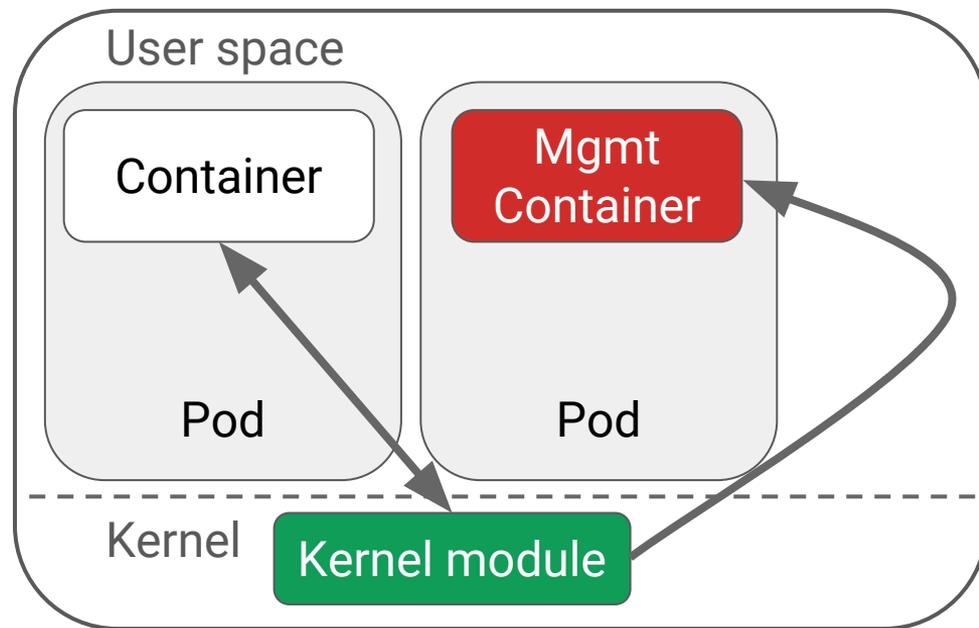
Node

Detect: Deployment models



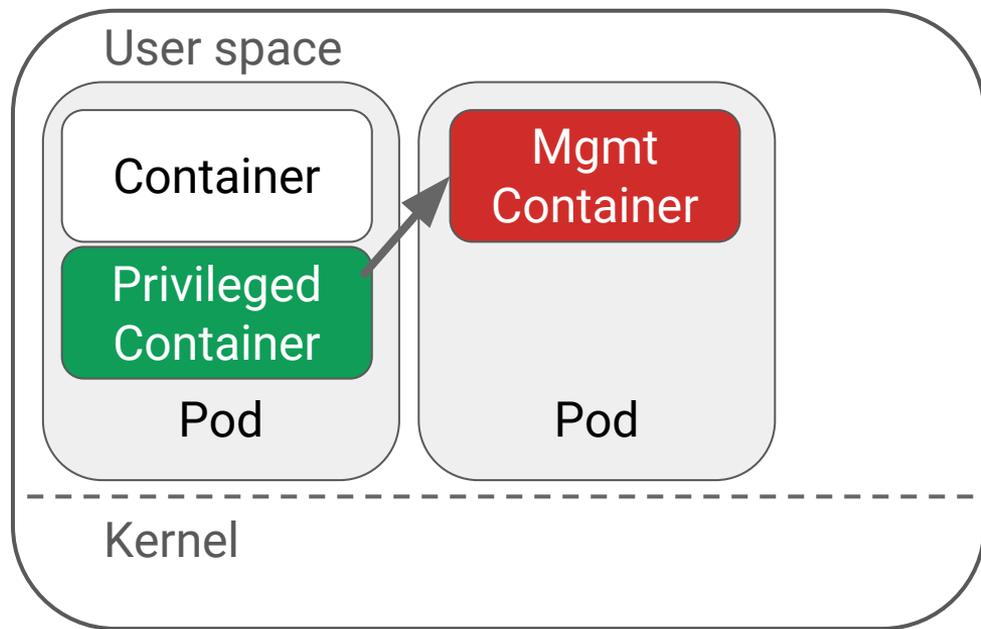
Node

Detect: Deployment models



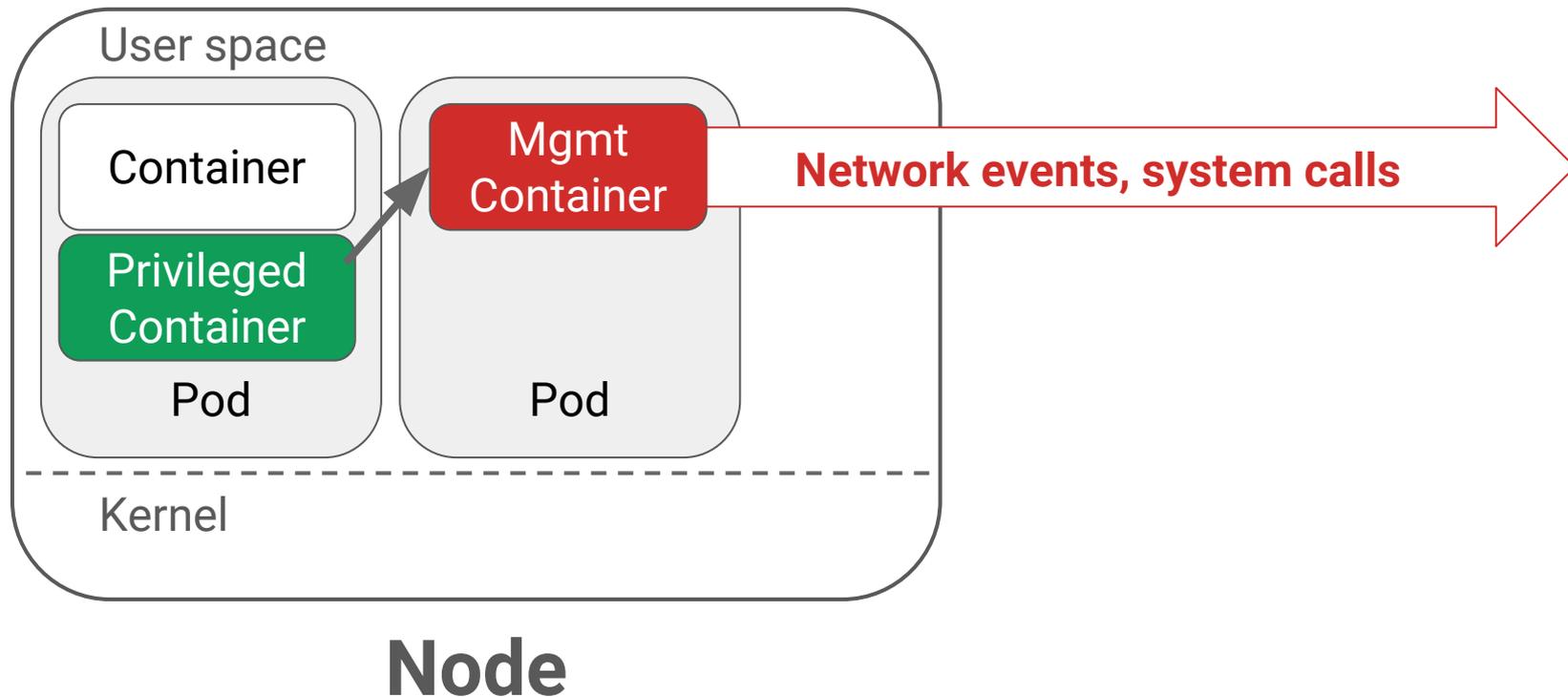
Node

Detect: Deployment models

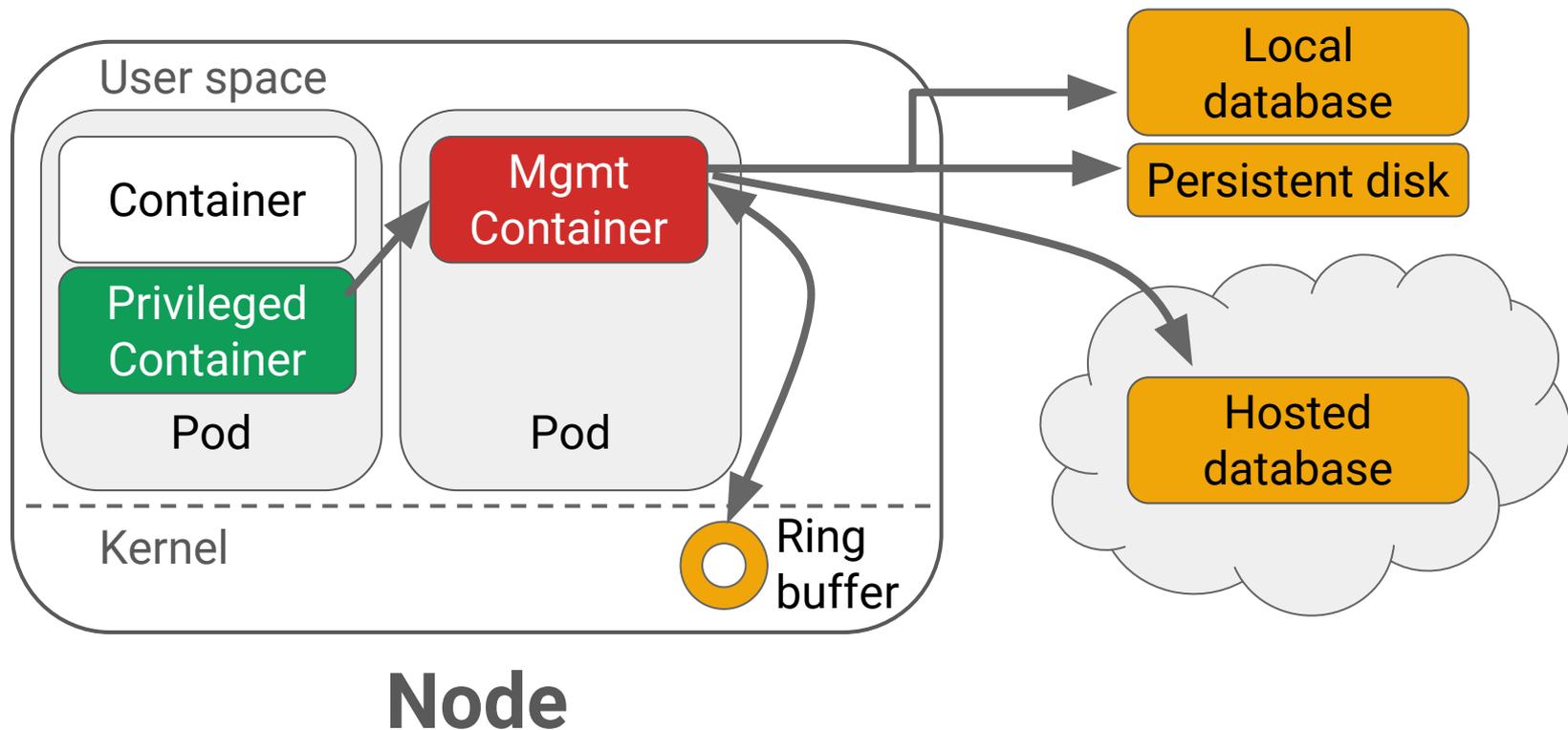


Node

Detect: Deployment models



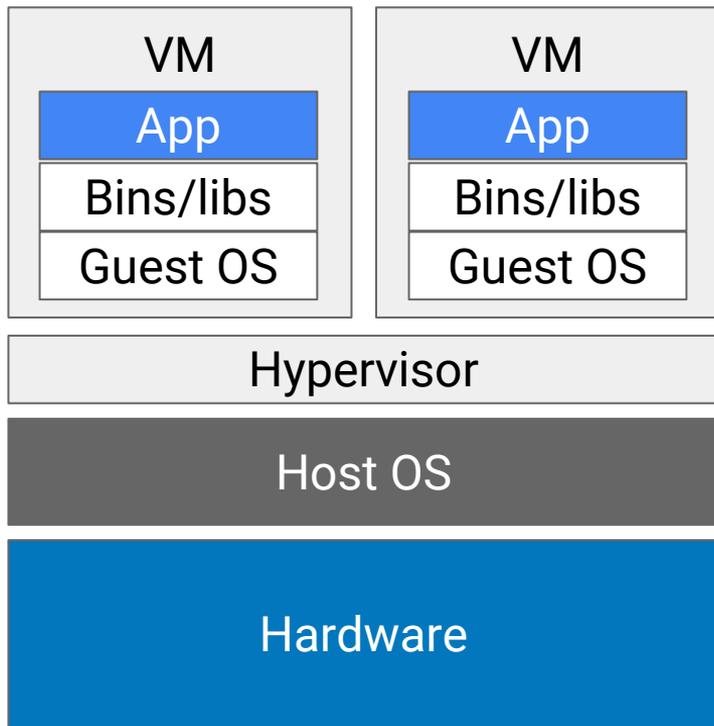
Detect: Deployment models



Respond options

- **Send an alert**
- **Isolate a container**, i.e. move it to a new network
- **Pause a container**, i.e. stop all running processes
- **Restart a container**, i.e. kill and restart processes
- **Kill a container**, i.e. kill processes without restart

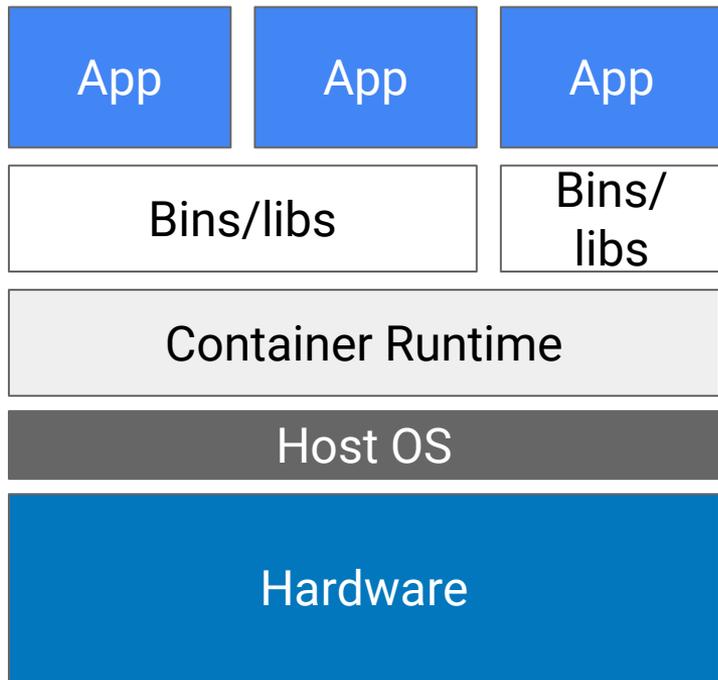
So, why are containers special again?



Virtual machine

- Long lived systems
 - Manual security patches and reviews
- Per-host software
 - IDS for host software
- Shared, physical network
 - Host-centric appliance for network traffic

So, why are containers special again?



Container

- Dynamic short-lived containers
 - **Need to redeploy often**
- Load isolation by container
 - **Need container IDS**
- Overlay network
 - **Need container network monitoring**

What can you do today?

- Make it part of your security plan
 - Try out open source options
 - Evaluate commercial options
- Deploy early
 - Get baseline readings
 - Tune your signals
- Rehearse an event

What can you do on GKE?

The screenshot displays the Google Cloud Platform Security Command Center interface. On the left, a sidebar shows navigation options like 'Security Command Center', 'Identity-Aware Proxy', and 'Encryption Keys'. The main area is divided into 'Assets' and 'Findings' sections. The 'Assets' section shows a table of asset types and counts. The 'Findings' section contains three summary cards: 'Findings: Summary' (772 total findings), 'Findings: Security scanner' (228 total findings), and 'Findings: DLP scanner' (544 total PII findings).

Type	Total	New	Deleted
ORGANIZATION	1	0	0
PROJECT	17	0	0
APPLICATION	11	0	0
SERVICE	8	0	0
ADDRESS	1	0	0
DISK	5	0	0
FIREWALL	32	0	0
INSTANCE	5	0	0
NETWORK	8	0	0
ROUTE	128	0	0
SUBNETWORK	120	0	0
KIND	1	0	0
BUCKET	37	0	0

Type	Count
MIXED_CONTENT	64
OUTDATED_LIBRARY	1
XSS_CALLBACK	114
XSS_ERROR	46
XSS_FLASH_INJECTION	3

PII type	Count
CREDIT_CARD_NUMBER	90
EMAIL_ADDRESS	112
FDA_CODE	87
IBAN_CODE	59
IP_ADDRESS	73



New! Kubernetes Engine resources



New! 5 partner integrations

Demo

Of a really bad day :(



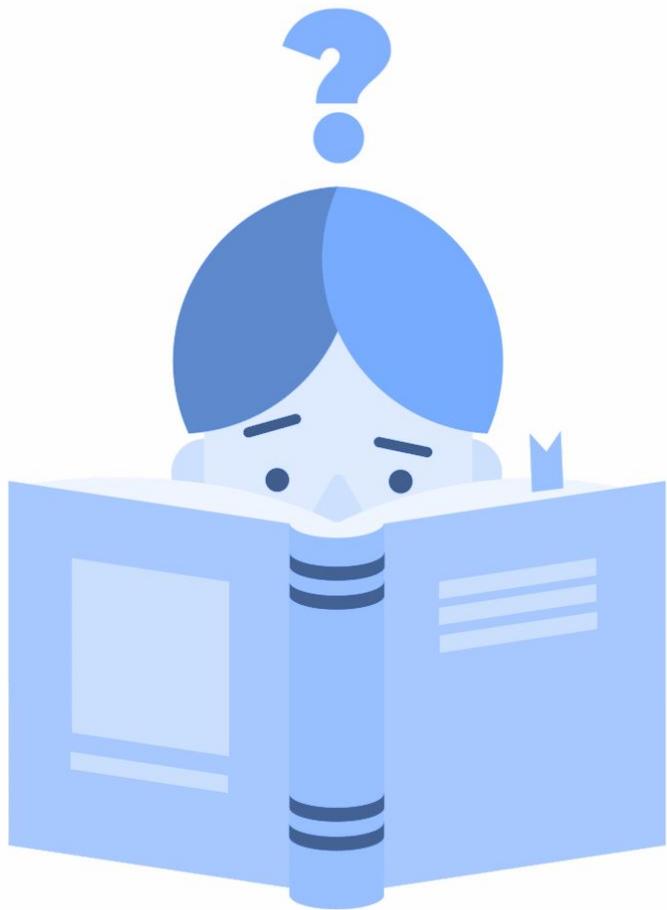
What we discussed

Container security overview

Practical differences from VMs

NIST and fence posts

What you can do today, including on GKE



Thank you!

Slides:

<http://sched.co/Dqvx>



