# Checking your --privileged container

Sam “Frenchie” Stewart, Cruise
Maya Kaczorowski, GitHub

BSides SF, February 23 2020

## Sam “Frenchie” Stewart

InfraSec Eng Mgr, Cruise
@nfFrenchie

cruise

## Maya Kaczorowski

Product Manager, GitHub
@MayaKaczorowski

# Agenda

## What's a container

and why do I care about containerd or seccomp anyways

## --privileged

All the features you can control

- What does it do?
- What happens if you don't block it?

## Isolation in Kubernetes

Controlling --privileged containers

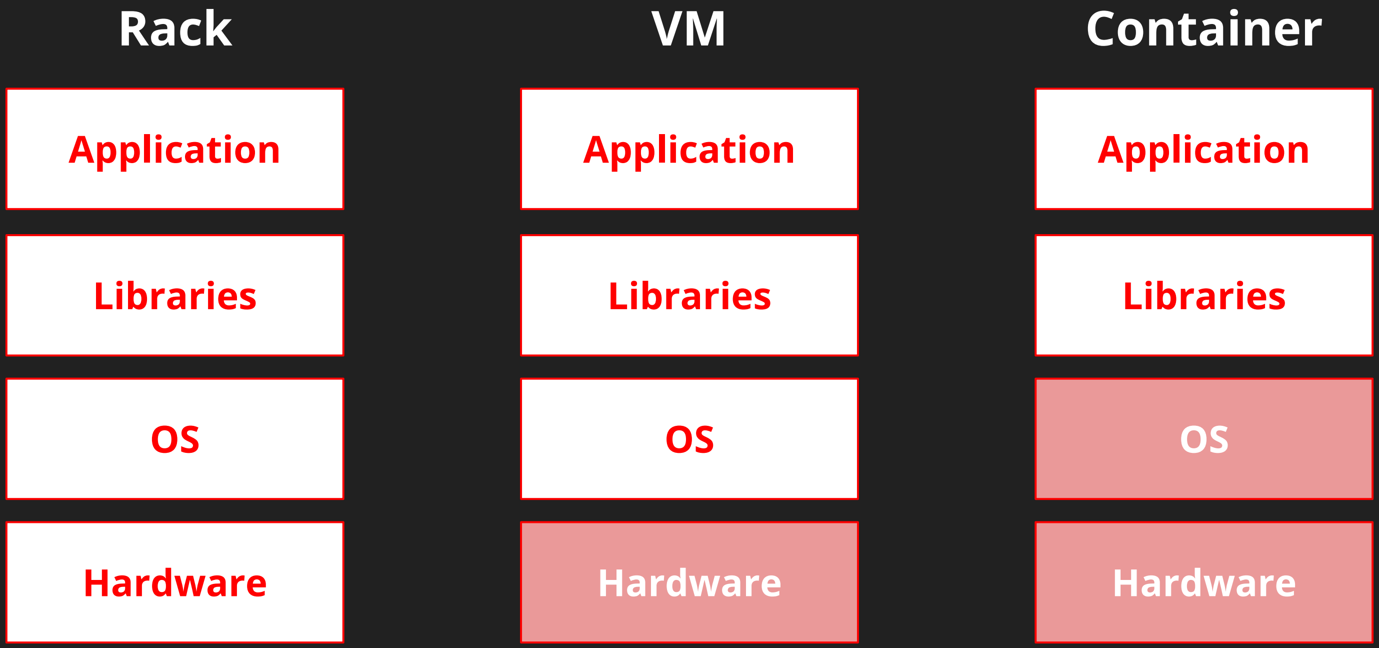
## Summary

# Audience Participation!

Very scientific demographic analysis

- Who has never heard of containers?
- Who has heard of them, used it once or twice, but not familiar?
- Who is familiar, prod users?
- Who is just here to post memes on twitter?

# What’s a container?



| Rack | VM | Container |
|---|---|---|
| Application | Application | Application |
| Libraries | Libraries | Libraries |
| OS | OS | OS |
| Hardware | Hardware | Hardware |

# **What's a container?** Docker and Kubernetes

## Container runtime

**Where privilege controls are enforced**

docker

containerd

RUNC

## Container orchestration

**where privilege controls may be defined**

RED HAT® OPENSHIFT

# What’s a container? cgroups and namespaces

**cgroups:** resource limits

**Namespaces:** process separation

See also: https://jvns.ca/blog/2016/10/10/what-even-is-a-container/

# **What’s a container?** capabilities

**Individual privileges a process can use**, like:

- CAP_AUDIT_CONTROL
- CAP_AUDIT_READ
- CAP_AUDIT_WRITE
- CAP_BLOCK_SUSPEND
- CAP_CHOWN
- etc.

See MAN pages: [http://man7.org/linux/man-pages/man7/capabilities.7.html](http://man7.org/linux/man-pages/man7/capabilities.7.html)

# **What's a container?** AppArmor, SELinux, seccomp

**AppArmor**

- Linux Security Module that lets you restrict your program's actions, e.g., file functions like read, write, execute
- Tied to process path

**SELinux**

- Linux Security Module that lets you restrict Mandatory Access Controls (MAC)
- Tied to process inode number

**seccomp**

- Filters a process' syscalls to limit what syscalls the process allows
- Puts application in 'secure' state with whitelist of allowed syscalls
- Docker seccomp default denies ~50 uncommon or potentially unsafe syscalls

# **--privileged** What does it do?

IDDQD DNCORNHOLIO ↑↑↓↓←→←→BA(Start)

# **--privileged** What does it do?

IDDQD

DNCORNHOLIO

↑↑↓↓←→←→BA(Start)

# **--privileged** What does it do?

`--privileged` is container `setenforce 0`

**https://stopdisablingselinux.com/**

# **--privileged** What is it?

**Lets your process run free**

with all the capabilities

like a `root` user

# **--privileged** How do you implement it?

Before:

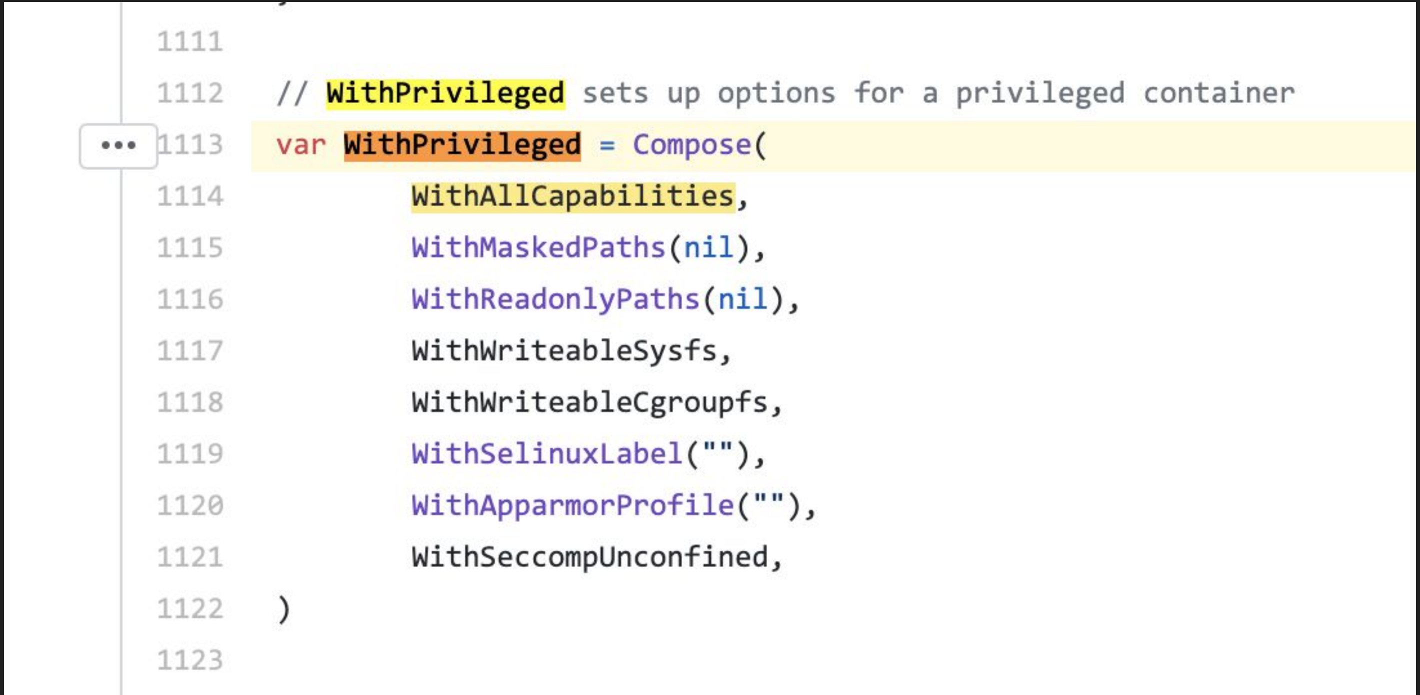
```
docker run nginx ...
```

After:

```
docker run --privileged nginx ...
```

# **--privileged** code walk

source: https://github.com/containerd/containerd/blob/master/oci/spec_opts.go#L1113



```go

// WithPrivileged sets up options for a privileged container
var WithPrivileged = Compose(
		WithAllCapabilities,
		WithMaskedPaths(nil),
		WithReadonlyPaths(nil),
		WithWriteableSysfs,
		WithWriteableCgroupfs,
		WithSelinuxLabel(""),
		WithApparmorProfile(""),
		WithSeccompUnconfined,
)

```

# **--privileged** code walk

```
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

# WithAllCapabilities

```
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

**What does it do?**

Adds all Linux capabilities

**Instead, have you tried?**

| | |
|---|---|
| Exposing ports <1024 | **CAP_NET_BIND_SERVICE** |
| Bind to arbitrary ports | **CAP_NET_RAW** |
| send RAW packets | **CAP_NET_RAW** |
| Other networking? | **CAP_NET_ADMIN** |
| Change host file perms | **CAP_CHOWN** |
| Killing host processes | **CAP_KILL** |
| Raise process niceness | **CAP_SYS_NICE** |

# WithMaskedPaths()

```
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

**What does it do?**

Sets masked paths to nil

**What happens if you don’t block it?**

“Everything is a file” – Linux

```
/proc/acpi
/proc/asound
/proc/kcore
/proc/keys
/proc/latency_stats
/proc/timer_list
/proc/sched_debug
/sys/firmware
/proc/scci
```

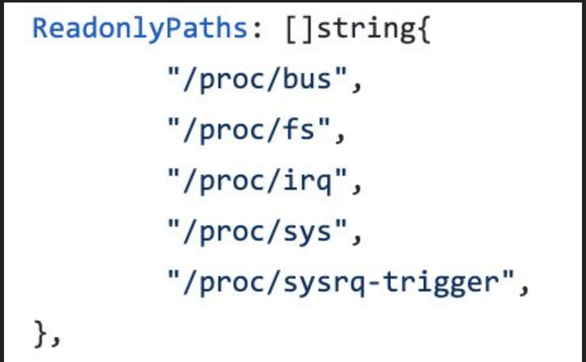
# WithReadonlyPaths()

```
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

## What does it do?

Sets read-only paths to nil

## What happens if you don’t block it?

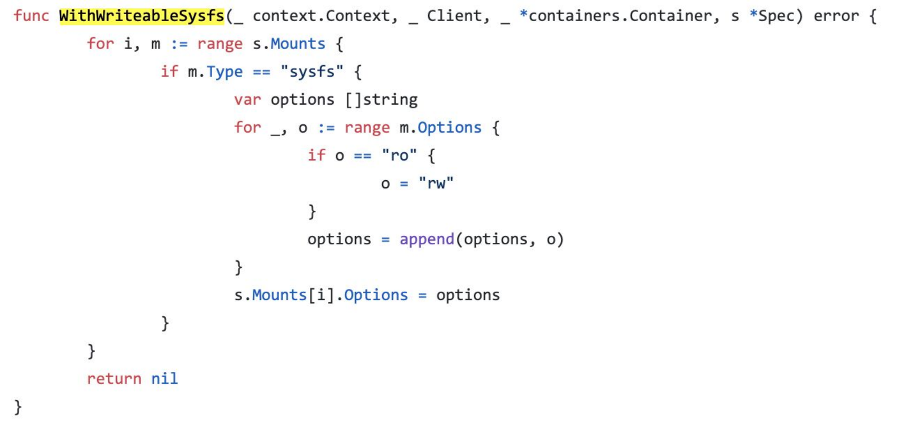


```
ReadonlyPaths: []string{
        "/proc/bus",
        "/proc/fs",
        "/proc/irq",
        "/proc/sys",
        "/proc/sysrq-trigger",
},
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

# WithWriteableSysfs

## What does it do?

Forces rw



```
func WithWriteableSysfs(_ context.Context, _ Client, _ *containers.Container, s *Spec) error {
	for i, m := range s.Mounts {
		if m.Type == "sysfs" {
			var options []string
			for _, o := range m.Options {
				if o == "ro" {
					o = "rw"
				}
				options = append(options, o)
			}
			s.Mounts[i].Options = options
		}
	}
	return nil
}
```

# WithWriteableCgroupfs

```
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

## What does it do?

Controls cgroups

## What happens if you don’t block it?

Potential for DoS

# WithSeLinuxLabel(), WithApparmorProfile()

```
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

**What does it do?**

Mandatory Access Control

**What happens if you don’t block it?**

If enabled on the host, this effectively disables it

# WithSeccompUnconfined

```
var WithPrivileged = Compose(
  WithAllCapabilities,
  WithMaskedPaths(nil),
  WithReadonlyPaths(nil),
  WithWriteableSysfs,
  WithWriteableCgroupfs,
  WithSelinuxLabel(""),
  WithApparmorProfile(""),
  WithSeccompUnconfined,
)
```

## What does it do?

In docker default, ~50 syscalls are blocked, removes that

## What happens if you don’t block it?

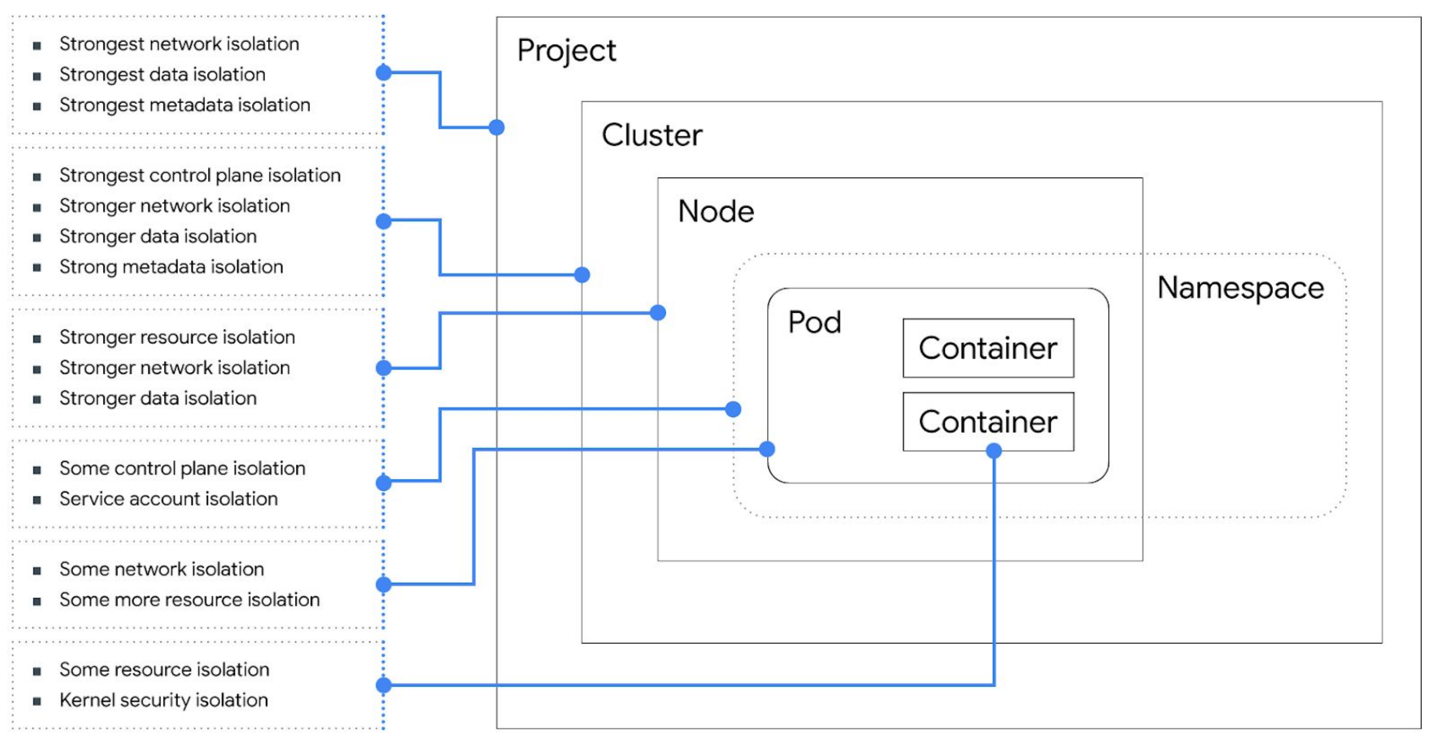
For the full list:

[https://docs.docker.com/engine/security/seccomp/](https://docs.docker.com/engine/security/seccomp/)

# Isolation in Kubernetes

CONTAINERS ARE NOT A SECURITY BOUNDARY

# **Isolation in Kubernetes** Security boundaries



- Strongest network isolation
- Strongest data isolation
- Strongest metadata isolation

- Strongest control plane isolation
- Stronger network isolation
- Stronger data isolation
- Strong metadata isolation

- Stronger resource isolation
- Stronger network isolation
- Stronger data isolation

- Some control plane isolation
- Service account isolation

- Some network isolation
- Some more resource isolation

- Some resource isolation
- Kernel security isolation

# **Isolation in Kubernetes** Pod Security Policy, OPA Gatekeeper

## In Kubernetes

**Security context**, part of a Pod specification

- Applies to the specified pod
- Enforced at runtime

**Pod Security Policy** admission controller

- Can apply to many pods
- Enforced at pod creation time

## In Open Policy Agent (OPA)

**Constraint Template**

- Define requirements

**Gatekeeper** admission controller

- Ensures pod meets Constraint Template
- Can apply to many pods
- Enforced at pod creation time

# **Isolation in Kubernetes** Other tools: k-rail

11.05am tomorrow – Theater 14

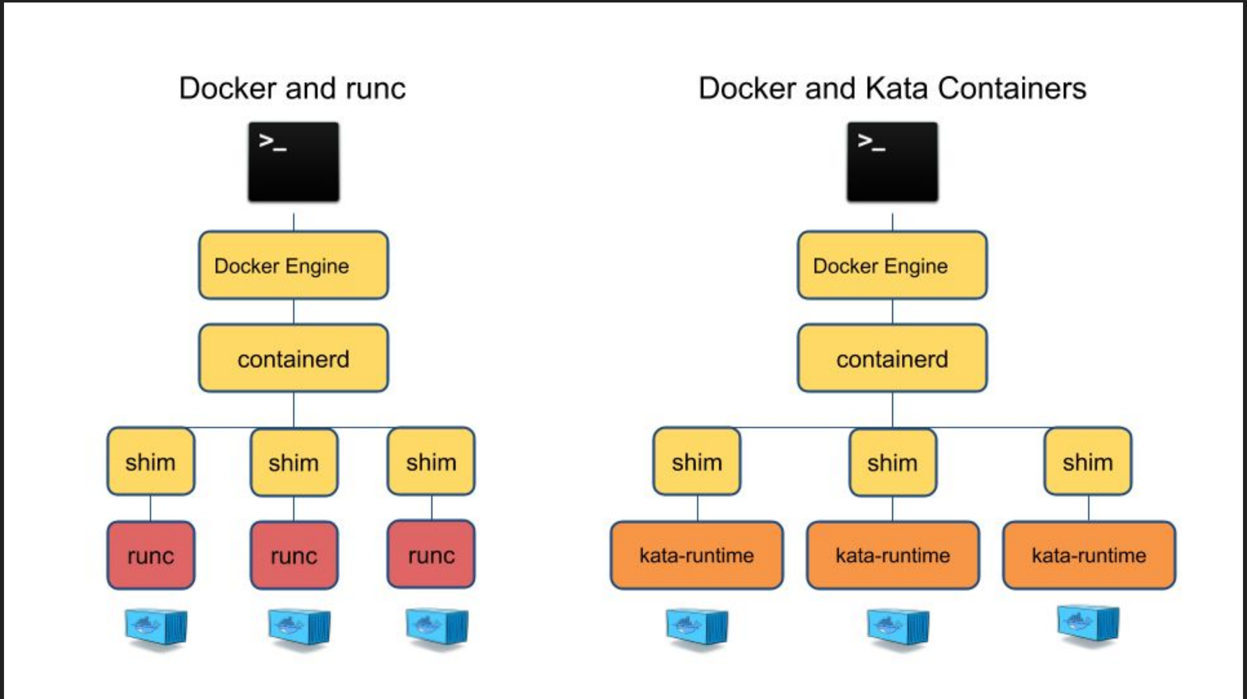
# **Isolation in Kubernetes** Runtime Class

In Kubernetes, use **RuntimeClass** to specify pod-level sandboxes

Some options for sandboxing:

- Kata containers
- gVisor
- Nabla containers
- Firecracker

# **Isolation in Kubernetes** Runtime Class



# Summary

## --privileged lets your processes run free

- Containers are just cgroups and namespaces
- Capabilities are what a process can do
- Use AppArmor, SELinux, seccomp to limit capabilities

## There are LOTS of privileges

- Drop CAPs where not needed

## Where you need it, use two layers of isolation

## Kubernetes has many isolation options

- Some isolation comes from Kubernetes constructs
- Use Pod Security Policy, OPA Gatekeeper, or k-rail
- For multi-tenant environments, consider sandboxing

# Learn more

- What even is a container: https://jvns.ca/blog/2016/10/10/what-even-is-a-container/
- Linux capabilities: http://man7.org/linux/man-pages/man7/capabilities.7.html
- Privileges in containerd: https://github.com/containerd/containerd/blob/master/oci/spec_opts.go#L1113
- Docker default seccomp profile: https://docs.docker.com/engine/security/seccomp/
- Stop disabling SELinux: https://stopdisablingselinux.com/
- Privileged containers aren’t containers: https://ericchiang.github.io/post/privileged-containers/
- Isolation in layers of Kubernetes: https://cloud.google.com/blog/products/gcp/exploring-container-security-isolation-at-different-layers-of-the-kubernetes-stack
- OPA Gatekeeper: https://github.com/open-policy-agent/gatekeeper
- k-rail: https://github.com/cruise-automation/k-rail
- Kubernetes runtime class: https://kubernetes.io/docs/concepts/containers/runtime-class/
- Sandboxing options: https://unit42.paloaltonetworks.com/making-containers-more-isolated-an-overview-of-sandboxed-container-technologies/

# Q&A

sli.do

#BSidesSF2020