Container Security and Multi-Tenancy Tales from Kata and Nabla

Ricardo Aravena (rico)
branch.io
@raravena80

James Bottomley
IBM Research
@jejb_

KubeCon | CloudNativeCon

North America 2018

Who Are We?

James Bottomley

Distinguished Engineer @ IBM

Working on Nabla Containers

Ricardo Aravena (rico)

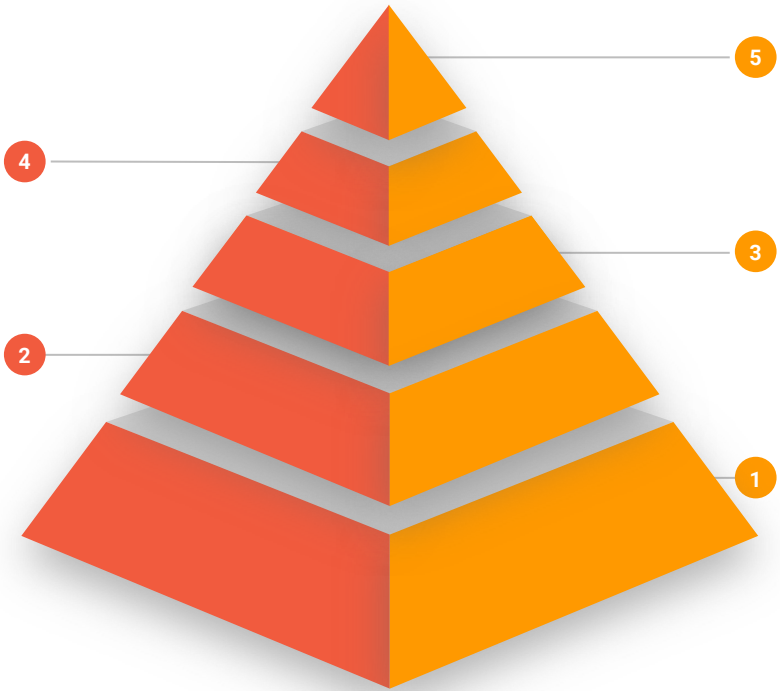
Work @ Branch Metrics

Cloud Ops

Kata Containers contributor

@raravena80
@jejb_

Containers @ Branch



DB & Big data apps
5 Standalone

Mesos qa/dev/prod
3 Data pipelines
Spark/Flink/HDFS/Druid

Where?
1 Everywhere

Runtime
Docker/containerd/runC 4

K8s in qa/dev/prod
API services - Lots of different languages 2

@raravena80
@jejb_

https://branch.io/

Outline

Problem

- Regular containers
- Security is hard

Kata

- What
- Why
- Containerd Shim V2

Nabla

- What/Why
- Metrics
- HAP

Kubernetes

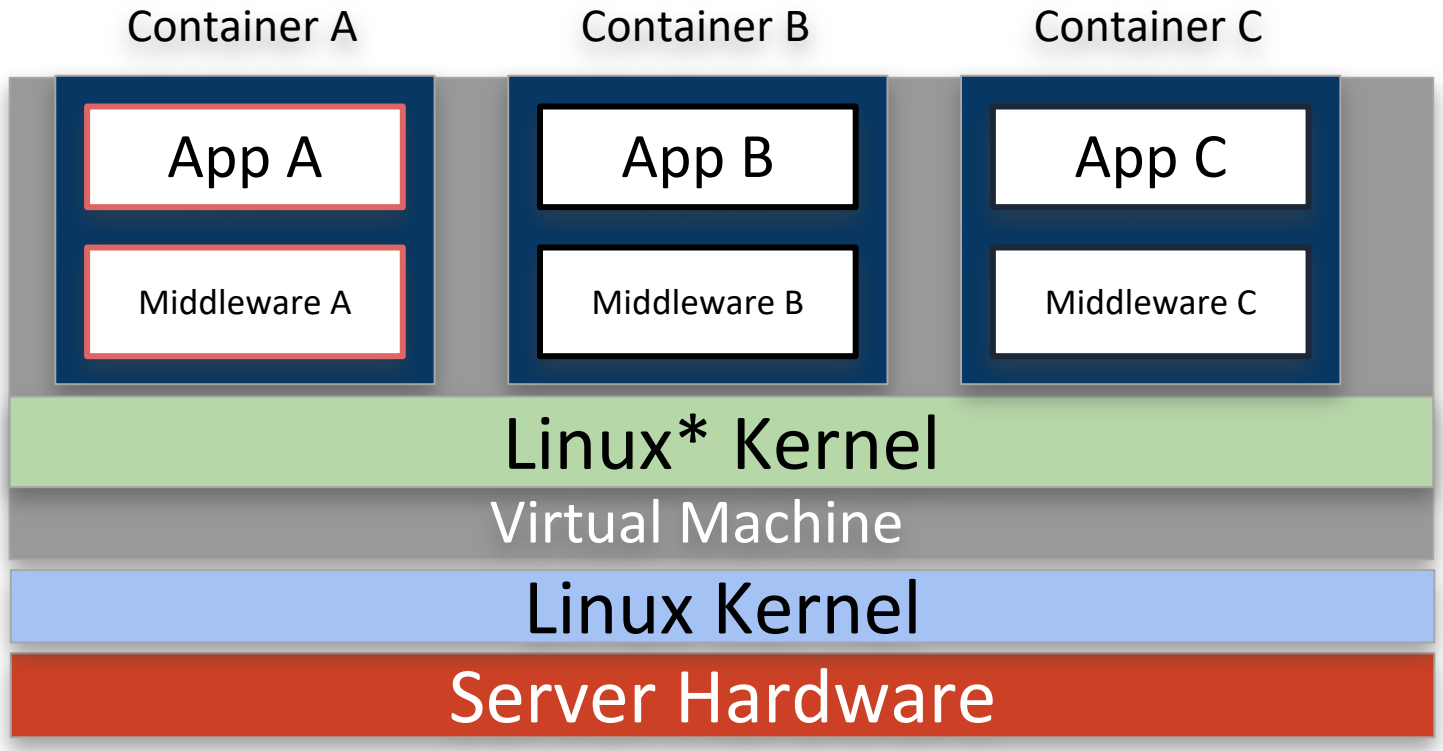
- Containerd
- CRI-O
- RuntimeClass

Future

- CPUs
- Hotplug
- AI/ML
- Blockchain

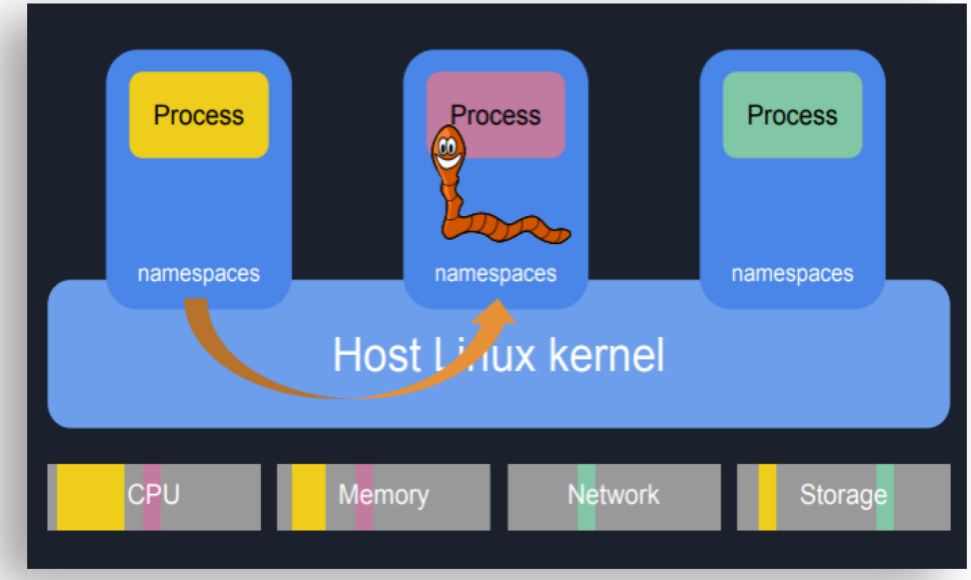
@raravena80
@jejb_

Traditional Containers



@raravena80
@jejb_

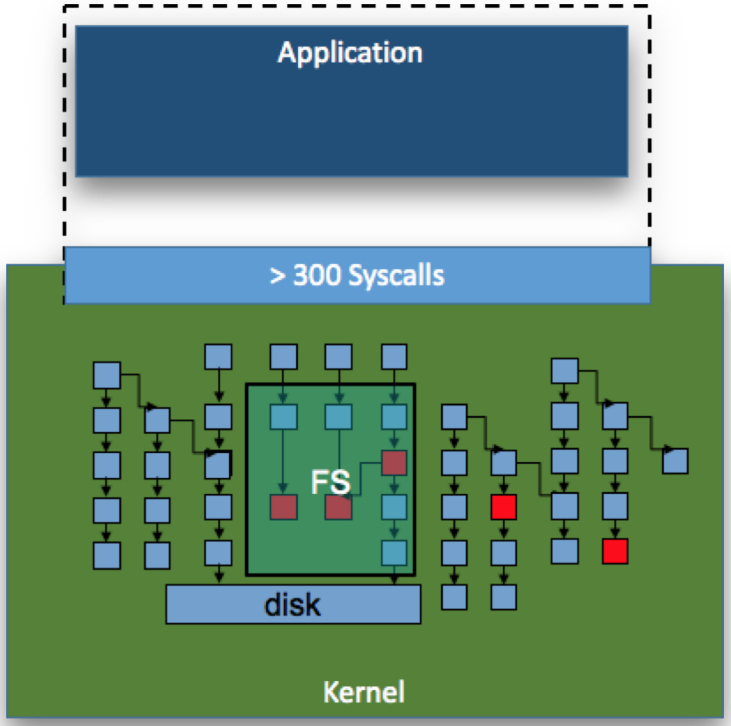
Traditional Containers



Prone to exploits

@raravena80
@jejb_

Kernel Footprint



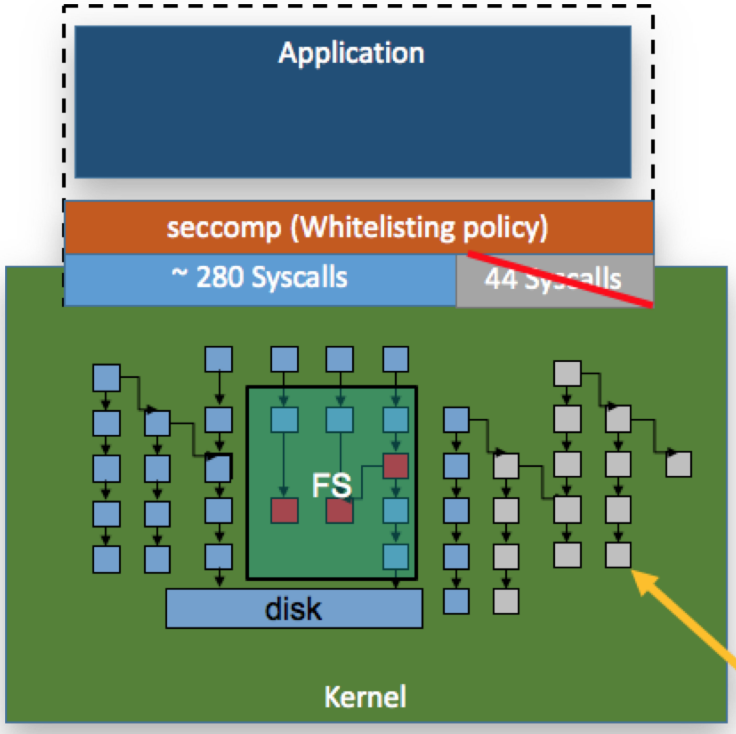
Exploits target vulnerable part of kernel via syscalls

If we restrict the number of syscalls

- Less reachable kernel functions
- Less potential vulnerabilities

@raravena80
@jejb_

Docker Default Seccomp Policy



Docker default seccomp policy disables around 44 system calls out of 300+. It is moderately protective while providing wide application compatibility

It is hard to make a generic seccomp policy for applications

Greyed – unreachable functions

@raravena80
@jejb_

Seccomp Profiles

```
{
        "defaultAction": "SCMP_ACT_ERRNO",
        "architectures": [
            "SCMP_ARCH_X86_64",
            "SCMP_ARCH_X86",
            "SCMP_ARCH_X32"
        ],
        "syscalls": [
            {
                "name": "accept",
                "action": "SCMP_ACT_ALLOW",
                "args": []
            },
            {
                "name": "accept4",
                "action": "SCMP_ACT_ALLOW",
                "args": []
            },
            {
                "name": "access",
                "action": "SCMP_ACT_ALLOW",
                "args": []
            },
..... 1578 Lines!
}
```

1578 lines

@raravena80
@jejb_

https://github.com/docker/labs/blob/master/security/seccomp/seccomp-profiles/default.json

Linux Capabilities

13 +

CAP_AUDIT_CONTROL
CAP_AUDIT_READ
CAP_AUDIT_WRITE
CAP_BLOCK_SUSPEND
CAP_CHOWN
CAP_DAC_OVERRIDE
CAP_DAC_READ_SEARCH
CAP_FOWNER
CAP_DAC_READ_SEARCH;
CAP_FSETID
CAP_IPC_LOCK
CAP_IPC_OWNER
CAP_KILL

13 +

CAP_LEASE
CAP_LINUX_IMMUTABLE
CAP_MAC_ADMIN
CAP_MAC_OVERRIDE
CAP_MKNOD
CAP_NET_ADMIN
CAP_NET_BIND_SERVICE
CAP_NET_BROADCAST
CAP_NET_RAW
CAP_SETGID
CAP_SETFCAP
CAP_SETPCAP
CAP_SETUID

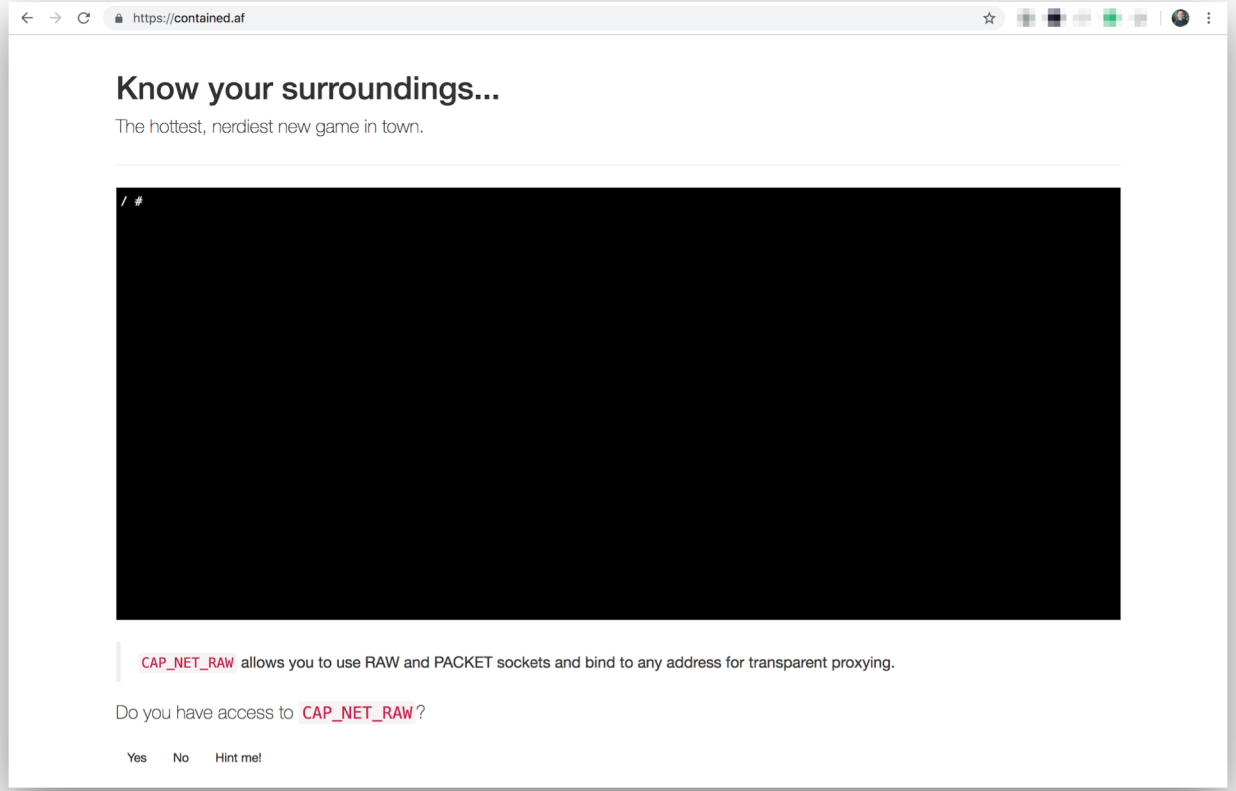
13 = 39

CAP_SYS_ADMIN
CAP_SYSLOG
CAP_SYS_BOOT
CAP_SYS_CHROOT
CAP_SYS_MODULE
CAP_SYS_NICE
CAP_SYS_PACCT
CAP_SYS_PTRACE
CAP_SYS_RAWIO
CAP_SYS_RESOURCE
CAP_SYS_TIME
CAP_SYS_TTY_CONFIG
CAP_WAKE_ALARM

@raravena80
@jejb_

http://man7.org/linux/man-pages/man7/capabilities.7.html

contained.af



JessFraz!

JessFraz!

@raravena80
@jejb_

https://contained.af

SELinux

Building the Base Policy Module

This exercise will build the mandatory base policy module that uses the same policy source file as the monolithic policy discussed above

The basic steps to produce a simple base test policy are:

- Ensure you are logged on as root and SELinux is running in permissive mode (setenforce 0) to perform the build process. It is assun
- Produce a base policy source file with a text editor (such as vi or gedit) containing the following contents: base.conf.
- Produce a base file contexts file with the following contents: base.fc. This will be used to relabel the file system.
- Produce a default_contexts file with the folowing contents default_contexts. This will be used to ensure that the correct context is built).
- Produce an seusers file with the following contents: seusers.
- Produce a dbus_contexts file with the following contents: dbus_contexts. This is required for X-Windows to load as it uses the dbu
- Produce a x_contexts file with the following contents: x_contexts. This is required for the X-Windows object manager.
- Compile the policy with checkmodule to produce an intermediate binary policy file:

```
checkmodule -o base.mod base.conf
```

@raravena80
@jejb_

http://selinuxproject.org/page/Building_a_Basic_Policy

SELinux

@raravena80
@jejb_

SELinux - Disable!

/etc/selinux/config

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

@raravena80
@jejb_

AppArmor

Installation

AppArmor is available in `linux` (since 4.18.8.arch1-1), `linux-zen` (since 4.18.8.zen1-1) and `linux-hardened` (since 4.17.4.a-1).

To enable AppArmor as default security model on every boot, set the following kernel parameters:

```
apparmor=1 security=apparmor
```

Install `apparmor` for userspace tools and libraries to control AppArmor. To load all AppArmor profiles on startup, enable `apparmor.service`.

Custom kernel

When compiling the kernel, it is required to set at least the following options:

```
CONFIG_SECURITY_APPARMOR=y
CONFIG_AUDIT=y
```

To use AppArmor as the default Linux security model and omitting the need of setting kernel parameters, also set the following options:

```
CONFIG_SECURITY_APPARMOR_BOOTPARAM_VALUE=1
CONFIG_DEFAULT_SECURITY_APPARMOR=y
```

@raravena80
@jejb_

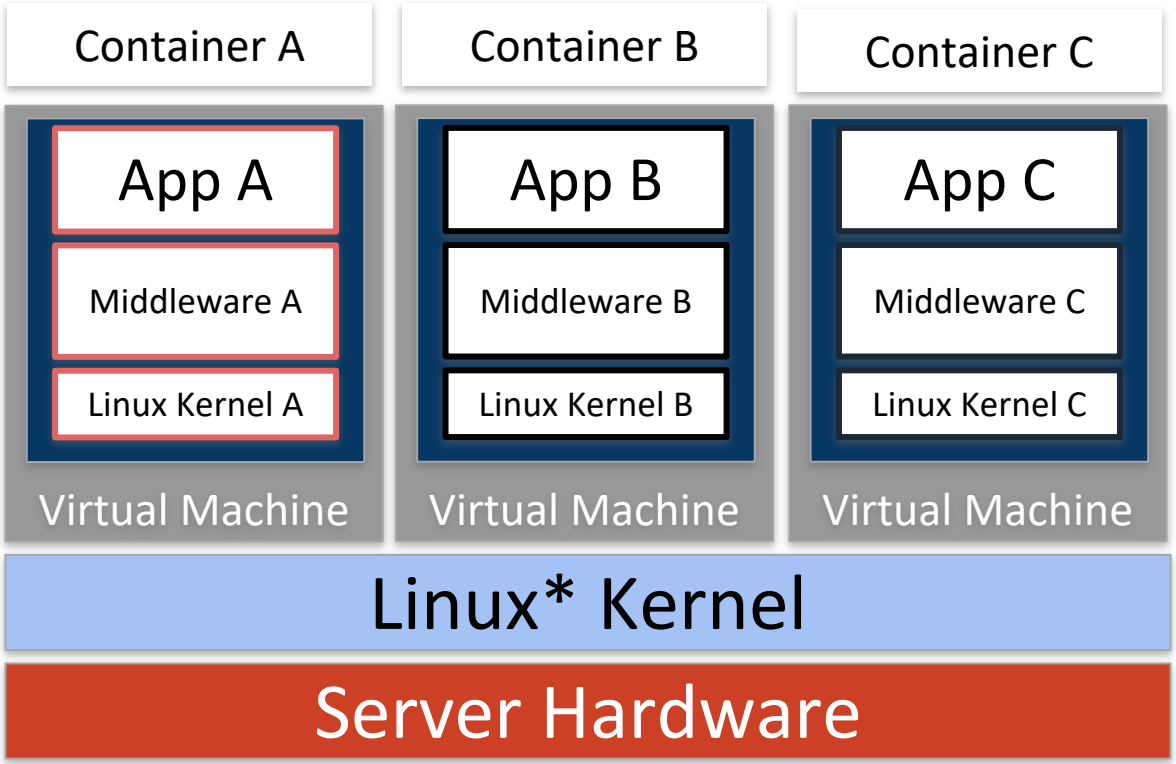
OMG

@raravena80
@jejb_

Kata Containers

@raravena80
@jejb_

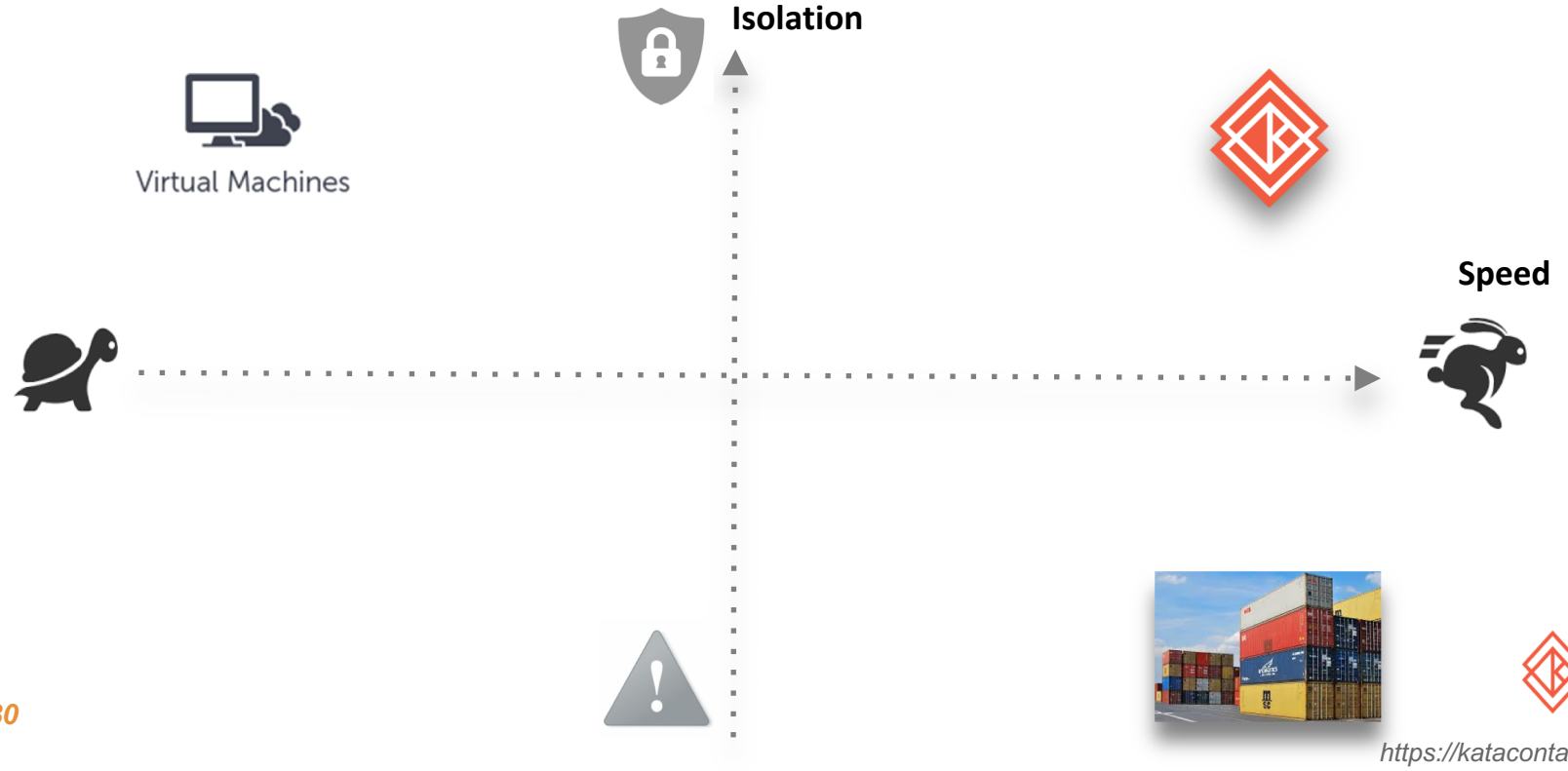
Kata Containers



@raravena80
@jejb_

https://katacontainers.io

Speed vs Security



https://katacontainers.io

@raravena80
@jejb_

Kata Requirements

Machine	• Bare Metal • Nested Virtualization
OS	• Linux
Public Cloud	• GCP - Nested • Azure - Nested • AWS - i3.metal instances
Private Cloud	• Openstack • Bare-metal & Nested Virt Providers
Platform	• Kubernetes - CRIO/Containerd • Docker

@raravena80
@jejb_

Kata Installation

Requirements	• $ kata-runtime kata-check
Kubernetes	• github.com/egernst/kata-deploy • Kata Installation guide
Docker	• Kata Installation guide

@raravena80
@jejb_

Containerd Shim V2

@raravena80
@jejb_

Containerd Shim V2

Problem

- Kata-shim for every containerd-shim

Solution

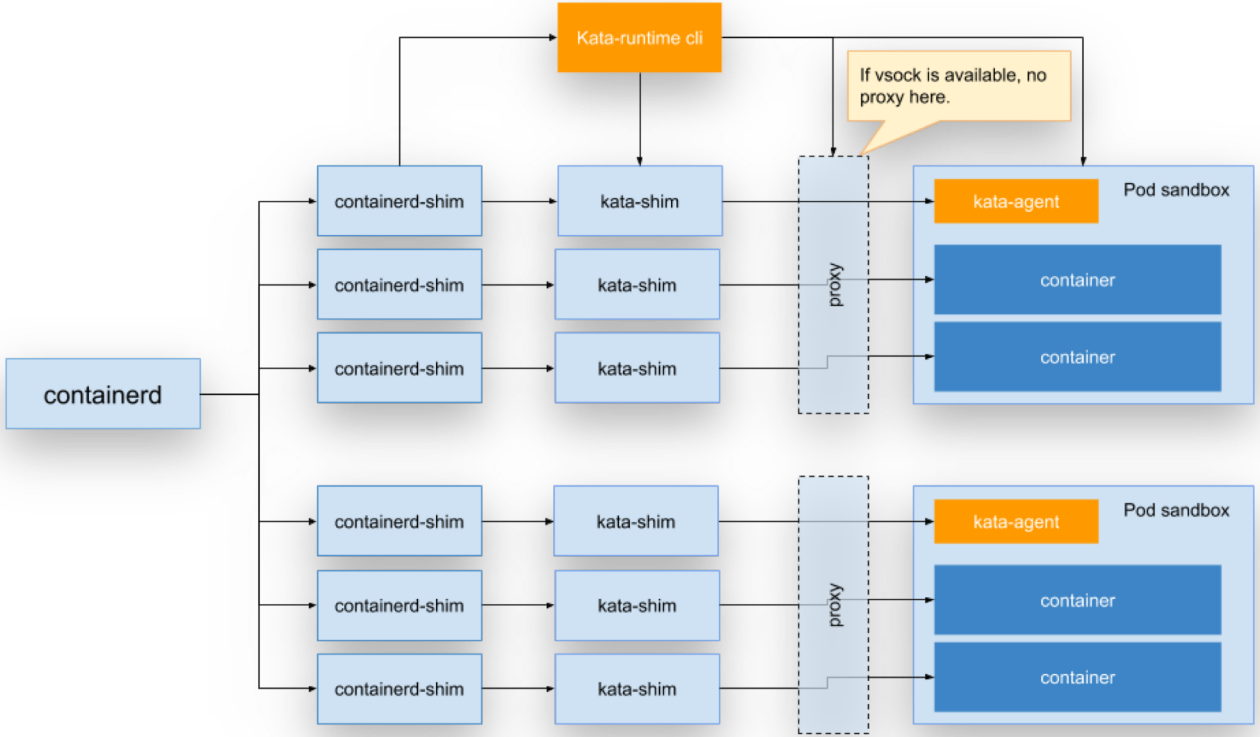
- Containerd-shim per pod
- Containerd handles I/O (stdout)
- No Kata-shim anymore
- No Kata-proxy with vsock support

Kata CLI

- No need to implement a runC compatible CLI

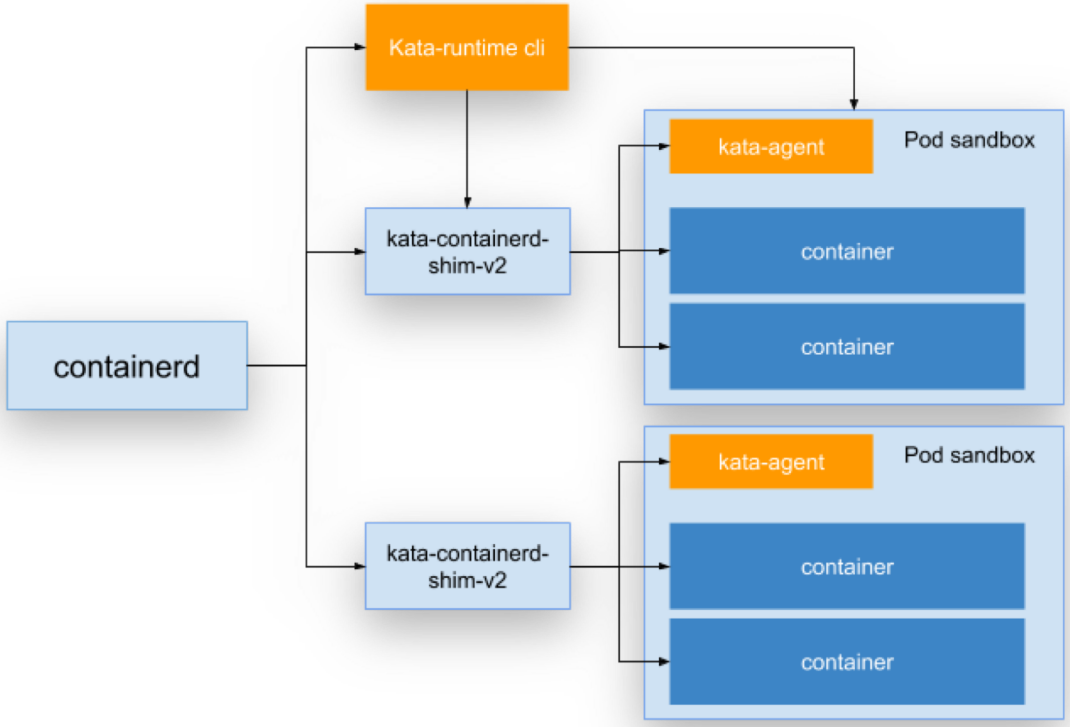
@raravena80
@jejb_

Containerd Shim V2 - Before



@raravena80
@jejb_

Containerd Shim V2 - After

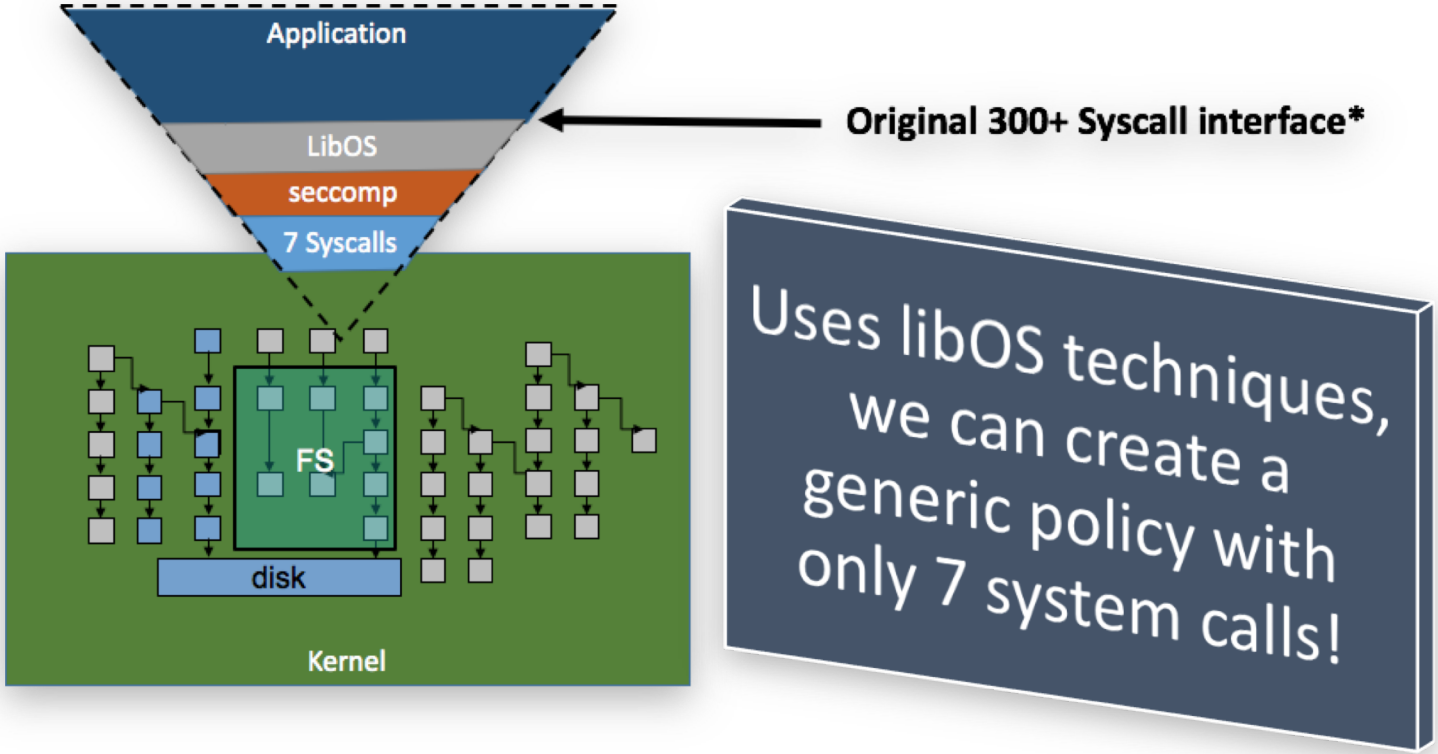


@raravena80
@jejb_

Nabla Containers

@raravena80
@jejb_

Nabla Containers



@raravena80
@jejb_

Inside Nabla

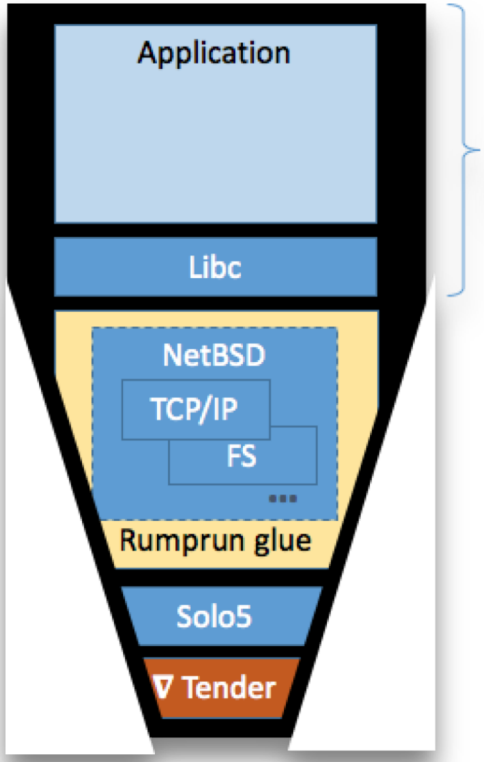
Unmodified user code (e.g., Node.js, redis, nginx, etc.)

Rumprun library OS

- Unmodified NetBSD code + some glue
- Runs on thin Solo5 unikernel interface

Nabla tender

- Setup of seccomp policy
- Translates Solo5 calls to system calls



@raravena80
@jejb_

Nabla Requirements

Machine	• Bare Metal • Regular Virtualization
OS	• Linux
Public Cloud	• Anything
Private Cloud	• Openstack • Bare-metal & Regular Virt Providers
Platform	• Kubernetes - CRIO/Containerd • Docker

@raravena80
@jejb_

Nabla Installation

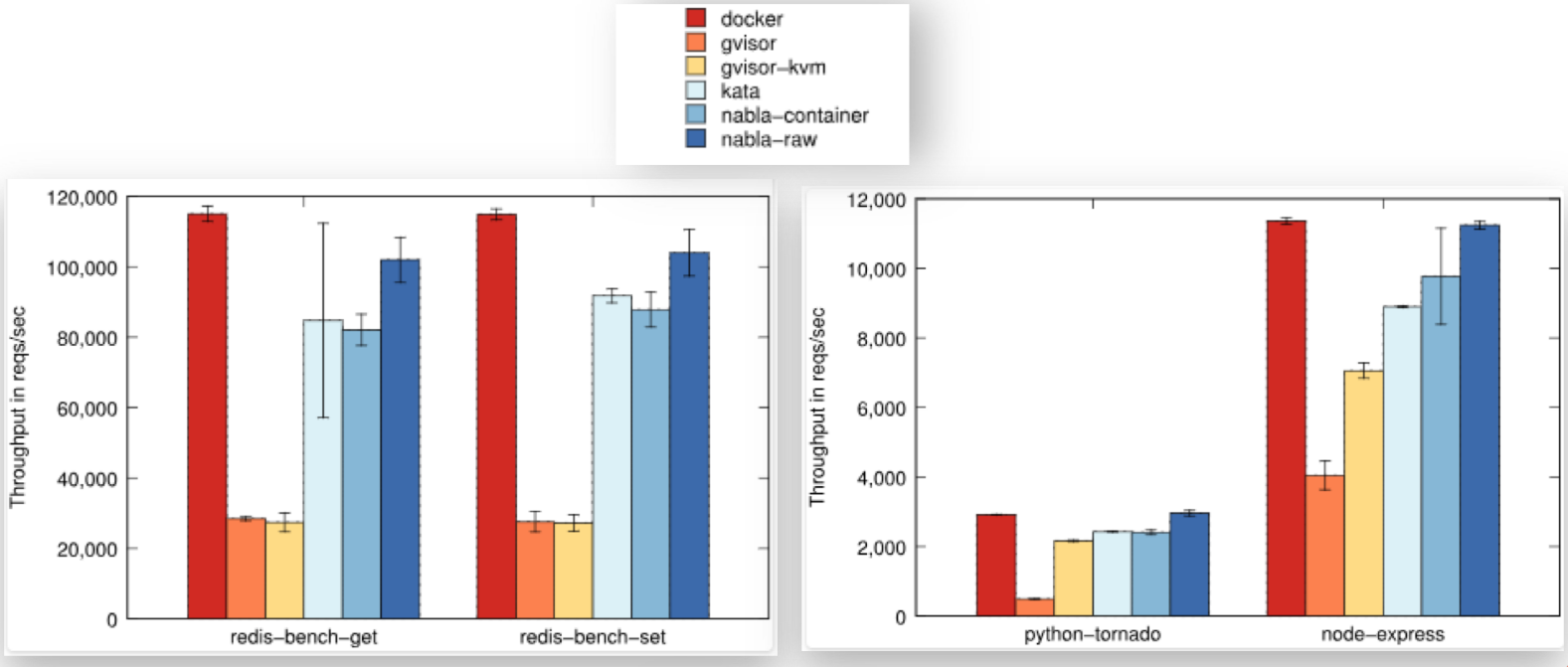
Requirements	• Build runnc • Pull node, go, python, etc image
Kubernetes	• Containerd or CRIO • RuntimeClass or annotation
Docker	• docker run --rm -p 8080:8080 --runtime=runnc nablact/node-express-nabla

@raravena80
@jejb_

Side by Side

@raravena80
@jejb_

Metrics



Throughput

@raravena80
@jejb_

https://blog.hansenpartnership.com/measuring-the-horizontal-attack-profile-of-nabla-containers/

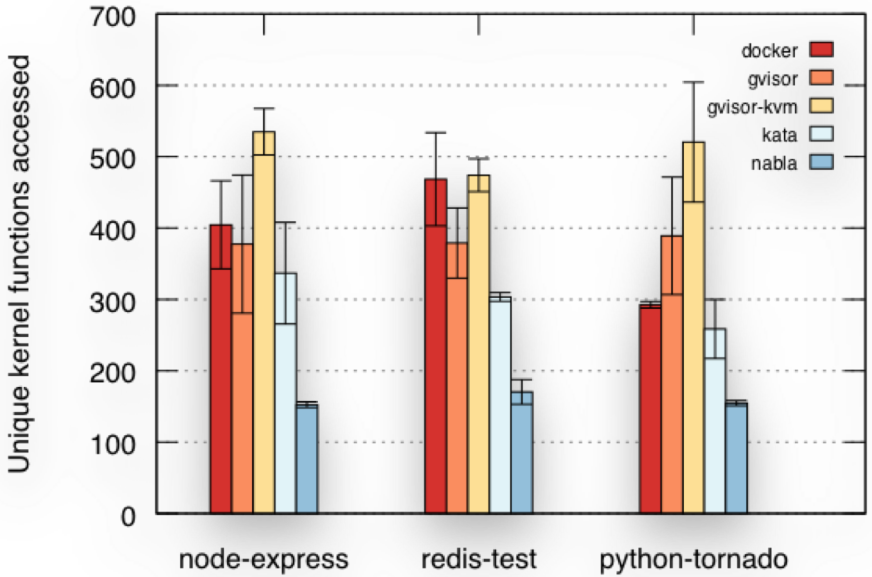
HAP

What?	• Horizontal Attack Profile
Who?	• IBM Research
Method	• Bug density of the Linux Kernel code • Multiply it by the amount of unique code traversed
How?	• Ftrace (function trace)
Runtimes	• Kata • Nabla • Docker • gVisor

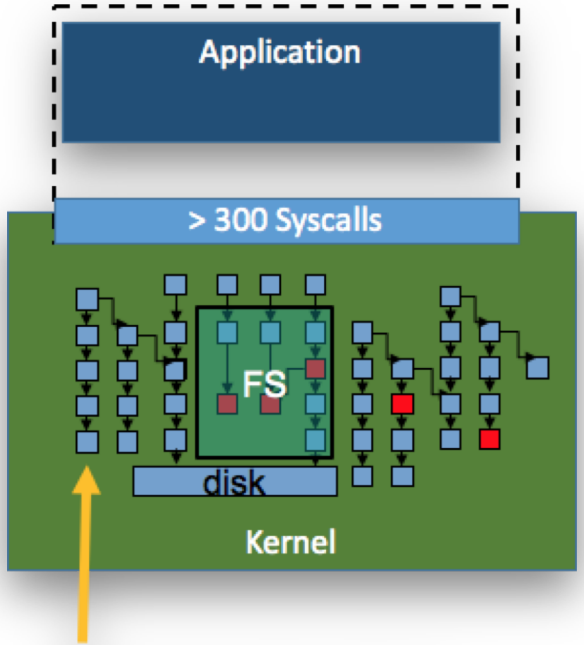
@raravena80
@jejb_

https://blog.hansenpartnership.com/measuring-the-horizontal-attack-profile-of-nabla-containers/

Ftrace Measurements



Lower is better



Measuring number of boxes Touched.

@raravena80
@jejb_

https://blog.hansenpartnership.com/measuring-the-horizontal-attack-profile-of-nabla-containers/

CVE-2018-10840

Filesystem Vulnerability

Kata is vulnerable with 9p

A simple VM isn't enough for full isolation

Consider interface choices

@raravena80
@jejb_

https://nabla-containers.github.io/2018/11/28/fs/

Kata & Nabla

Kata

- VM Isolation
- All syscalls with no mod
- Can run any workloads
- Needs bare metal or nested virt
- Established community
- Any Linux workload compatible out of the box

Nabla

- Solo 5 Unikernel
- 7 syscalls
- Specific workload builds
- Can run on any server
- Growing community
- Compatible with NetBSD workloads

@raravena80
@jejb_

https://katacontainers.io https://nabla-containers.github.io/

Kubernetes

@raravena80
@jejb_

Kubernetes

How?	• CRIO  • Containerd  • RuntimeClass w/K8s 1.12 or later
Kata Requirements	• Bare Metal • Nested Virtualization • QEMU/NEMU
Nabla Requirements	• Bare Metal or VM

@raravena80
@jejb_

RuntimeClass

"

The EC2 for K8s

@raravena80
@jejb_

RuntimeClass

What?

- Use different runtimes in K8s
- K8s 1.12 (alpha)

How?

- RuntimeClass Feature gate
- Install CRD
- Configure Containerd/CRI
- runtimeClassName spec in Pod

@raravena80
@jejb_

https://kubernetes.io/docs/concepts/containers/runtime-class/

RuntimeClass

```yaml
# RuntimeClass is defined in the node.k8s.io API group
apiVersion: node.k8s.io/v1alpha1
kind: RuntimeClass
metadata:
  # RuntimeClass is a non-namespaced Resource
  # The name the RuntimeClass will be referenced by
  name: myclass
# The name of the corresponding CRI configuration
spec:
  runtimeHandler: myconfiguration
```

@raravena80
@jejb_

Credit: https://kubernetes.io/docs/concepts/containers/runtime-class/

RuntimeClass - PodSpec

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  runtimeClassName: myclass
# ...
```

@raravena80
@jejb_

Credit: https://kubernetes.io/docs/concepts/containers/runtime-class/

Containerd Config

```
[plugins.cri.containerd.runtimes.kata-runtime]
      runtime_type = "io.containerd.runc.v1"
      [plugins.cri.containerd.runtimes.runc.options]
        NoPivotRoot = false
        NoNewKeyring = false
        ShimCgroup = ""
        IoUid = 0
        IoGid = 0
        BinaryName = "/opt/kata/bin/kata-runtime"
        Root = ""
        CriuPath = ""
        SystemdCgroup = false
```

@raravena80
@jejb_

Dynamic Runtime Class

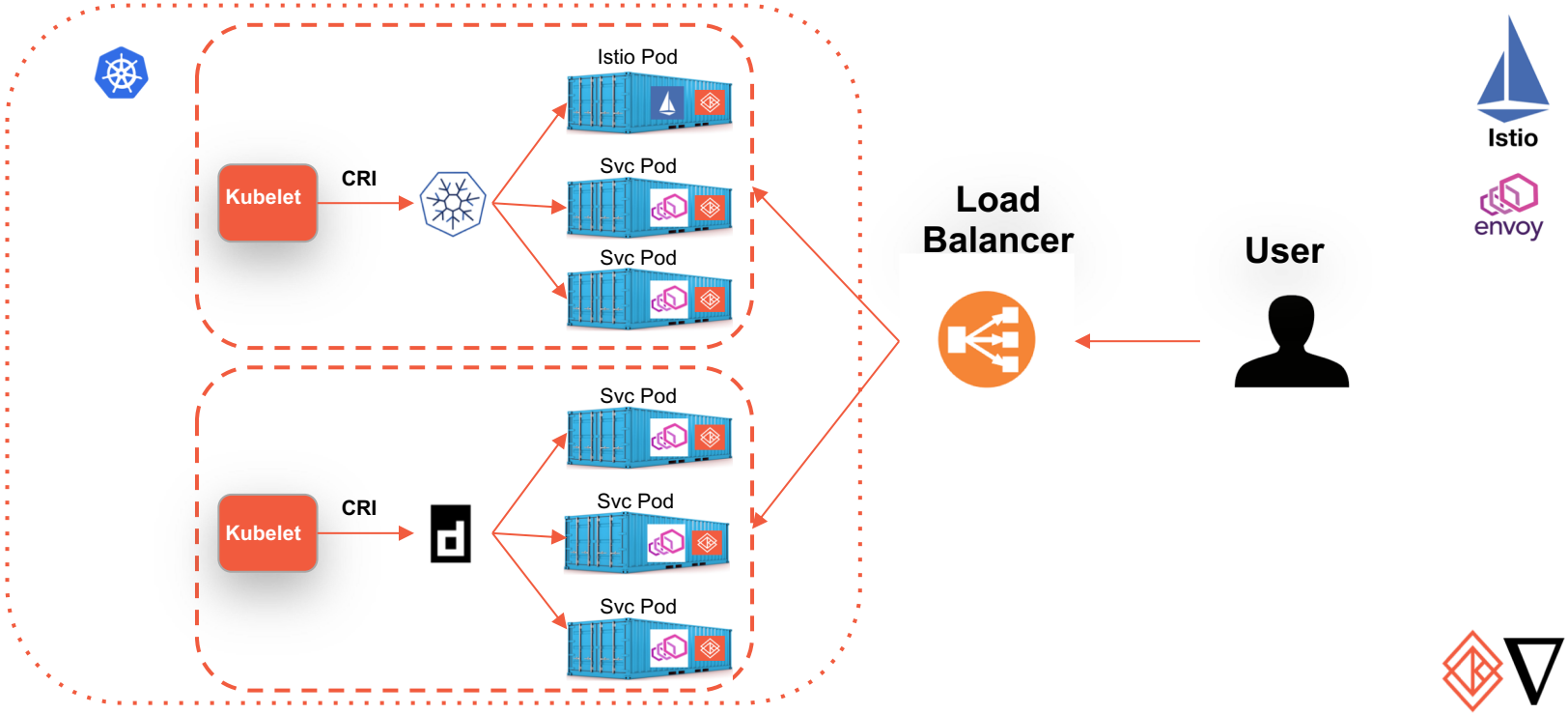

```
service CRIRuntimePlugin {
        // ListAndWatch returns a stream of List of RuntimeHandler
        // Whenever the status of CRI Runtime, the status of Container Runtime
        // or the Config file is changed, ListAndWatch returns the new list
        // $ kubectl get runtimeclass
        rpc ListAndWatch(Empty) returns (stream ListAndWatchResponse) {}
}
```

@raravena80
@jejb_

Workloads

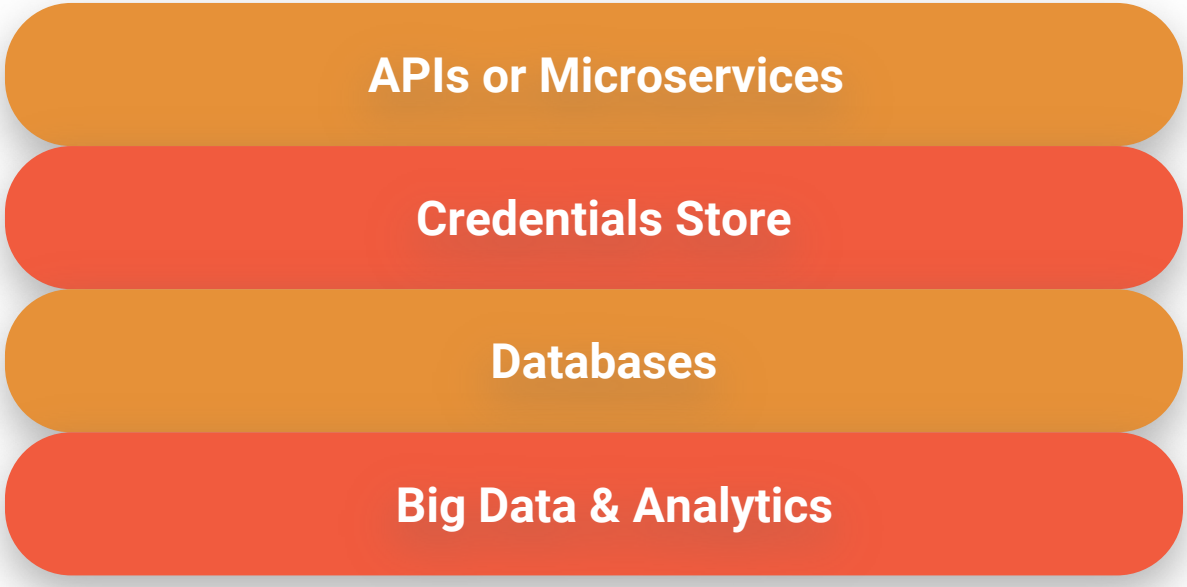
@raravena80
@jejb_

Kubernetes + Service Mesh



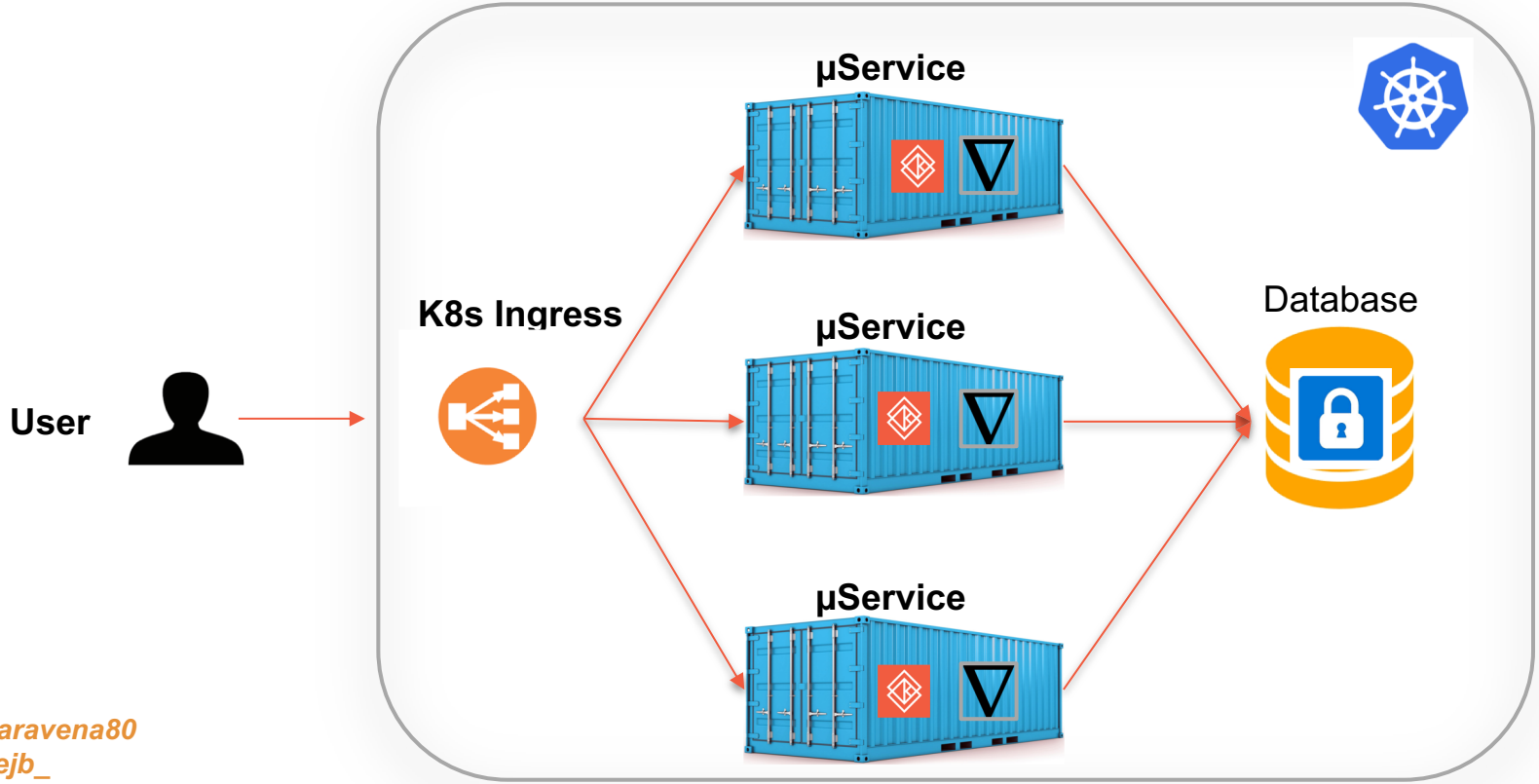
@raravena80
@jejb_

Sample Workloads



@raravena80
@jejb_

Microservices



@raravena80
@jejb_

Credentials Store

User

TLS

Vault Active

Vault Standby

Vault Standby

TLS

TLS

TLS

Storage

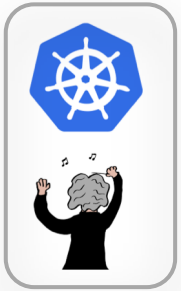
Amazon S3

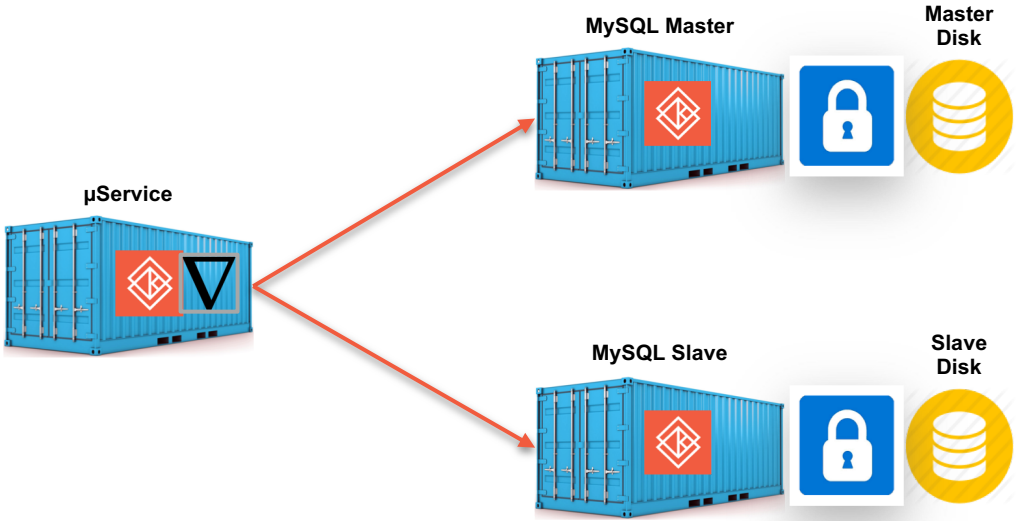
etcd

Vault

@raravena80
@jejb_

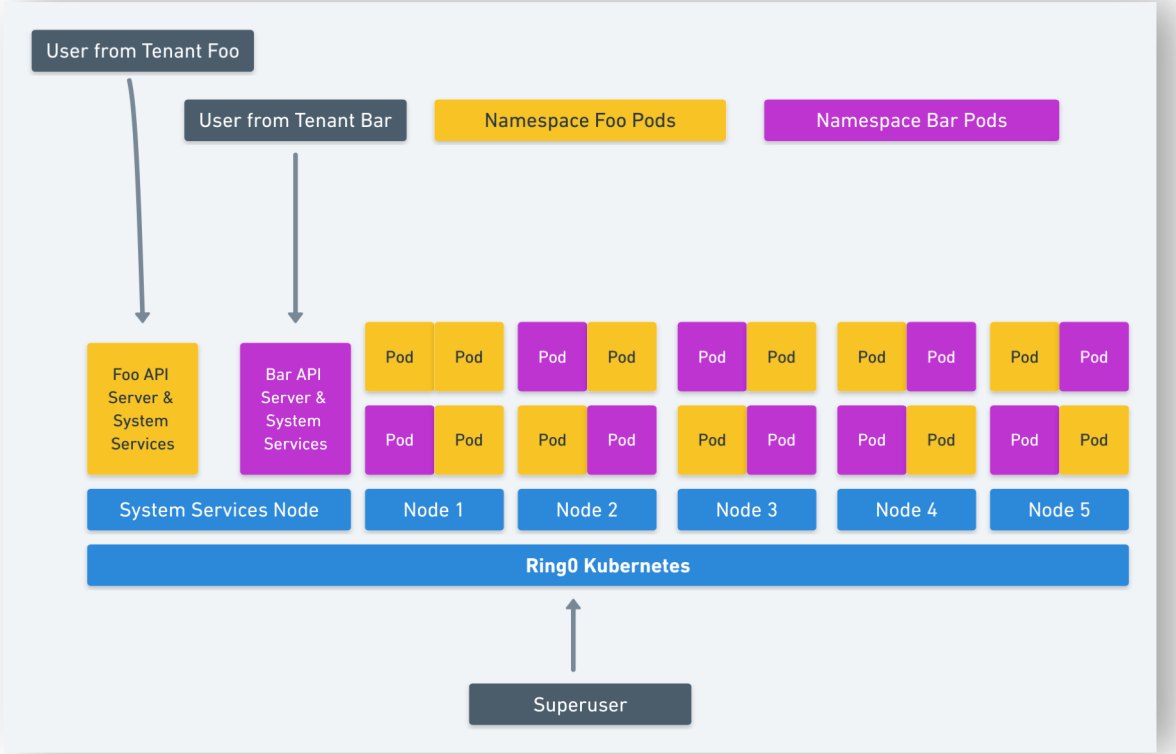
Relational DBs





@raravena80
@jejb_

Multitenancy



@raravena80
@jejb_

Credit: https://blog.jessfraz.com/post/hard-multi-tenancy-in-kubernetes/

Demo

Kata & Nabla Demo

Kata and Nabla Containers

Kubernetes

RuntimeClass

Show Both Runtimes

@raravena80
@jejb_

Future

@raravena80
@jejb_

Kata Containers - Future

Hypervisors	• Hyper-V • VMware • Xen
Public Cloud	• AWS Nested Virtualization • More Bare Metal Offerings • GKE, AKS, ACS (?)
Features	• Firecracker support • Hotplug networking, cpus

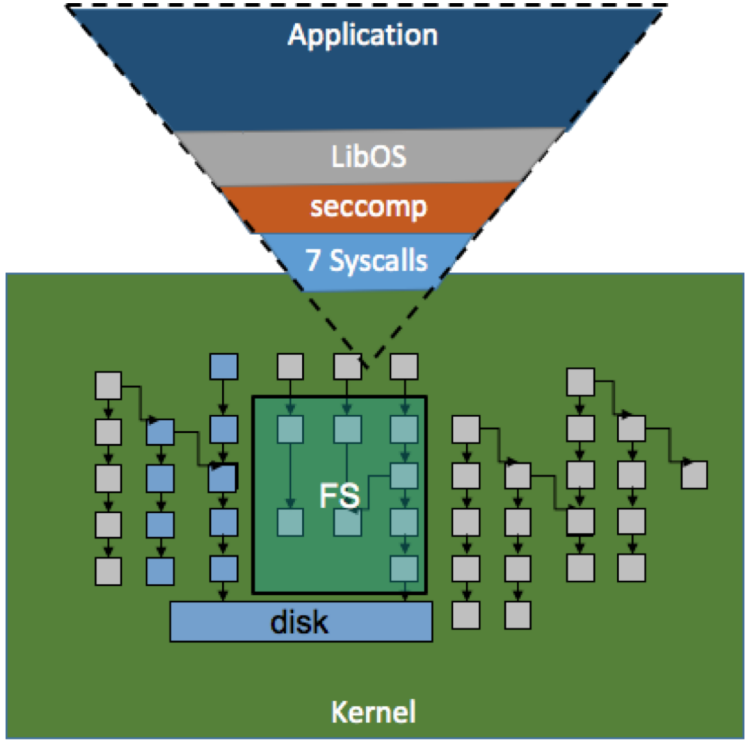
@raravena80
@jejb_

Nabla Containers - Future

Workloads	• Kubernetes integration • More languages and apps
Public Cloud	• GKE, AKS, ACS (?)
Features	• Tooling around rumprun image builds • Buildless containers

@raravena80
@jejb_

Nabla Containers - Future



@raravena80
@jejb_

Future Workloads

NFV	• Multus • https://github.com/intel/multus-cni
Edge Computing	• 5G • IoT
AI/HPC	• Tensorflow / Kubeflow • GPUs • ML model training

@raravena80
@jejb_

A Word on Serverless

How?

- Serverless frameworks

Frameworks

- Riff
- Serverless
- Nuclio
- Fission
- Dispatch
- AWS Chalice

Middleware & Event Managers

- Knative (GKE)
- Event Gateway

@raravena80
@jejb_

https://landscape.cncf.io/landscape=serverless

Resources

Kata Containers	https://katacontainers.io
Nabla Containers	https://nabla-containers.github.io/
RuntimeClass	https://kubernetes.io/docs/concepts/containers/runtime-class/
Multitenancy	https://github.com/kubernetes/community/tree/master/wg-multitenancy
K8s Security Context	https://kubernetes.io/docs/tasks/configure-pod-container/security-context/
Istio Soft Multi-tenancy	https://istio.io/blog/2018/soft-multitenancy/

@raravena80
@jejb_

Thank you!

@raravena80
@jejb_