

# Infrastructure Security 2.0

Jonathan Pulsifer



# \$ whoami

- Infrastructure Security Engineer @ Shopify
- Certified Kubernetes Administrator
- [twitter.com/JonPulsifer](https://twitter.com/JonPulsifer)
- [github.com/JonPulsifer](https://github.com/JonPulsifer)

## Previously

- Team Lead at CFNOC
- Network Defense Instructor at CFSCE
- SANS Mentor / Co-instructor (GCIA, GSEC)



## Jonathan Pulsifer

@JonPulsifer

Find me dropping container capabilities and working on 🐳 security @Shopify || IT guy for @LawNeedsFem || CKA, GCIA, GSEC #kubernetes #cloudnative #treatyoself

📍 Ottawa, ON

# Containers



# What Containers Are **NOT**

- chroots
- bsd jails
- solaris zones
- virtual machines



# What Containers Are **NOT**

- chroots
- bsd jails
- solaris zones
- virtual machines
- **REAL**

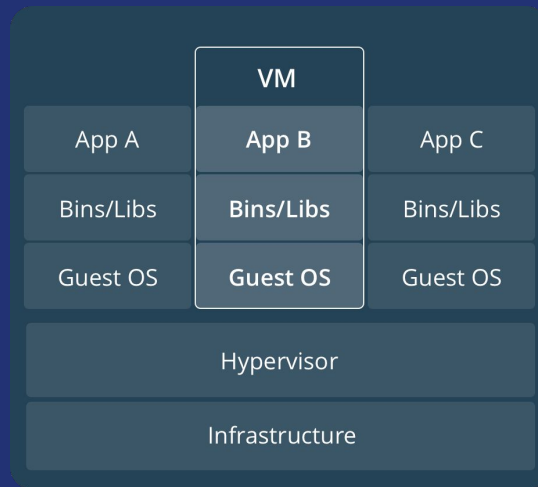
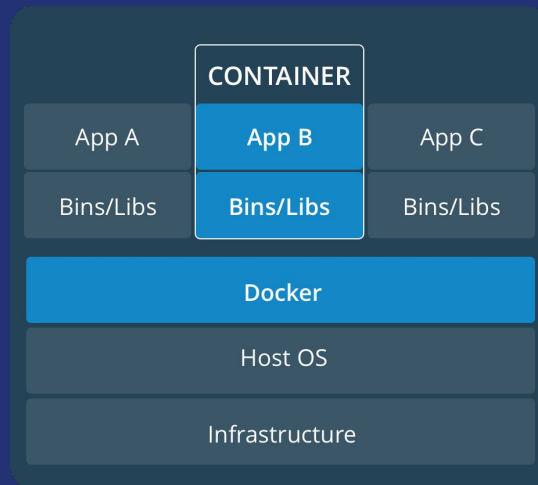


So, what is a **container**?



# OS Level Virtualization

- Also called **containerization**
- Packages code and dependencies together
- Virtualizes the OS not the hardware
- Kernel allows multiple isolated user-space instances (called containers!)



# Linux Namespaces

Namespace	Isolates
<b>PID</b>	Process IDs
<b>Mount</b>	Filesystem mount points
<b>IPC</b>	Messaging queues
<b>Network</b>	Network interfaces
<b>UTS</b>	Hostname and domain name
<b>User</b>	User and group IDs
<b>Cgroup</b>	What resources a process can use



# Control Groups

Cgroup	tl;dr
<b>CPU</b>	Provides access to the CPU by “CPU shares”
<b>Memory</b>	Sets limits on memory use and generates usage reports
<b>PIDS</b>	Number of processes that may be created
<b>Blkio</b>	Limits I/O to and from block devices (HDD, SSD, USB etc)
<b>CPUSet</b>	Use these particular CPUs
<b>Devices</b>	Allows or denies access to devices
<b>Net_Prio</b>	Dynamically set the priority of network traffic

How do we **build** them?

# Docker Images

REPOSITORY	TAG	IMAGE ID	SIZE
alpine	3.6	37eec16f1872	3.97MB
ubuntu	artful	579580072367	93.8MB
debian	stretch	874e27b628fd	100MB
centos	7	196e0ce0c9fb	197MB
busybox	latest	54511612f1c4	1.13MB

# Dockerfile

```
FROM alpine:3.6
```

```
RUN apk add --no-cache snort
```

```
ENTRYPOINT ["/usr/bin/snort"]
```

# Dockerfile

```
FROM alpine:3.6
```

```
RUN addgroup -S snort \  
&& adduser -SG snort snort
```

```
RUN apk upgrade --no-cache \  
&& apk add --no-cache snort
```

```
ENTRYPOINT ["/usr/bin/snort"]  
CMD ["-u", "snort", "-g", "snort"]
```

## Builder Stats

6,000

average builds per weekday

330,000

images in GCR

# PIPA

- Buildpack, Dockerfile, or custom build pipelines
- Kubernetes template validation
- **Container Audits:**
  - does this image run as root?
  - does this image contain any vulnerable packages?
  - other container attestations

Shopify/friendly-ghost (production) builder  
git@github.com:Shopify/friendly-ghost.git



171 Builds

0 Running

0 Scheduled

New Build

Pipeline Settings

Merge pull request #111 from Shopify/edgescale/enable-auto-tls

Build #163 | master | 3e43842

Passed in 2m 49s



Pipeline Setup

Trigger validation build

buildpack - Build Contain...



Grafeas + Kritis



Jonathan Pulsifer

Created Mon 25th Sep at 10:43 AM

Triggered from Webhook

Rebuild



Pipeline Setup | pipa setup

9 seconds

pipa-agent-production-596776426-21wsc



Trigger validation build | pipa wrapper /buildkite/validations/k8s/run.sh

18 seconds

pipa-agent-production-596776426-6jg77



buildpack - Build Container | pipa build -x --push -- /buildkite/pipelines/buildpack/... 2 minutes, 13 seconds | pipa-agent-production-596776426-p1brl

Log

Artifacts

Agent

Environment

```
+ Expand groups - Collapse groups [Delete] [Download] [Follow]

1 ▶ Running global environment hook 0s
3 ▶ 🚀 Setting up Package Cloud Environment 3s
4 ▶ Applying environment changes 0s
8 ▶ Running global pre-checkout hook 0s
10 ▶ Preparing build directory 0s
13 ▶ Running global checkout hook 1s
46 ▶ Running global command hook 0s
48 ▶ Starting build 0s
49 ▶ Creating dummy DB containers 0s
50 ▶ Downloading cache 5s
55 ▶ buildpack 🚀 1m 31s
416 ▶ Image layers 0s
432 ▶ Pruning cache 0s
433 ▶ Uploading cache 8s
438 ▶ Deleting local cache copy 1s
439 ▶ Deleting dummy DB containers 2s
441 ▶ Pushing to registry 17s
468 ▶ Applying environment changes 0s
470 ▶ Running global post-command hook 0s
472 ▶ Cleaning up stage 4s
497 ▶ Cleanup complete 0s
498 ▶ Running global pre-exit hook 0s
```

Back to top



Grafeas + Kritis | /buildkite/grafeas/kritis

7 seconds

pipa-agent-production-596776426-p1brl





## Grafeas

- <https://github.com/Grafeas/Grafeas>
- Central source of truth for software component metadata
- my.regist.ry/image@sha256:hash as key for containers
- Container notes produced at build
- See GCP's or Shopify's Engineering blog for more

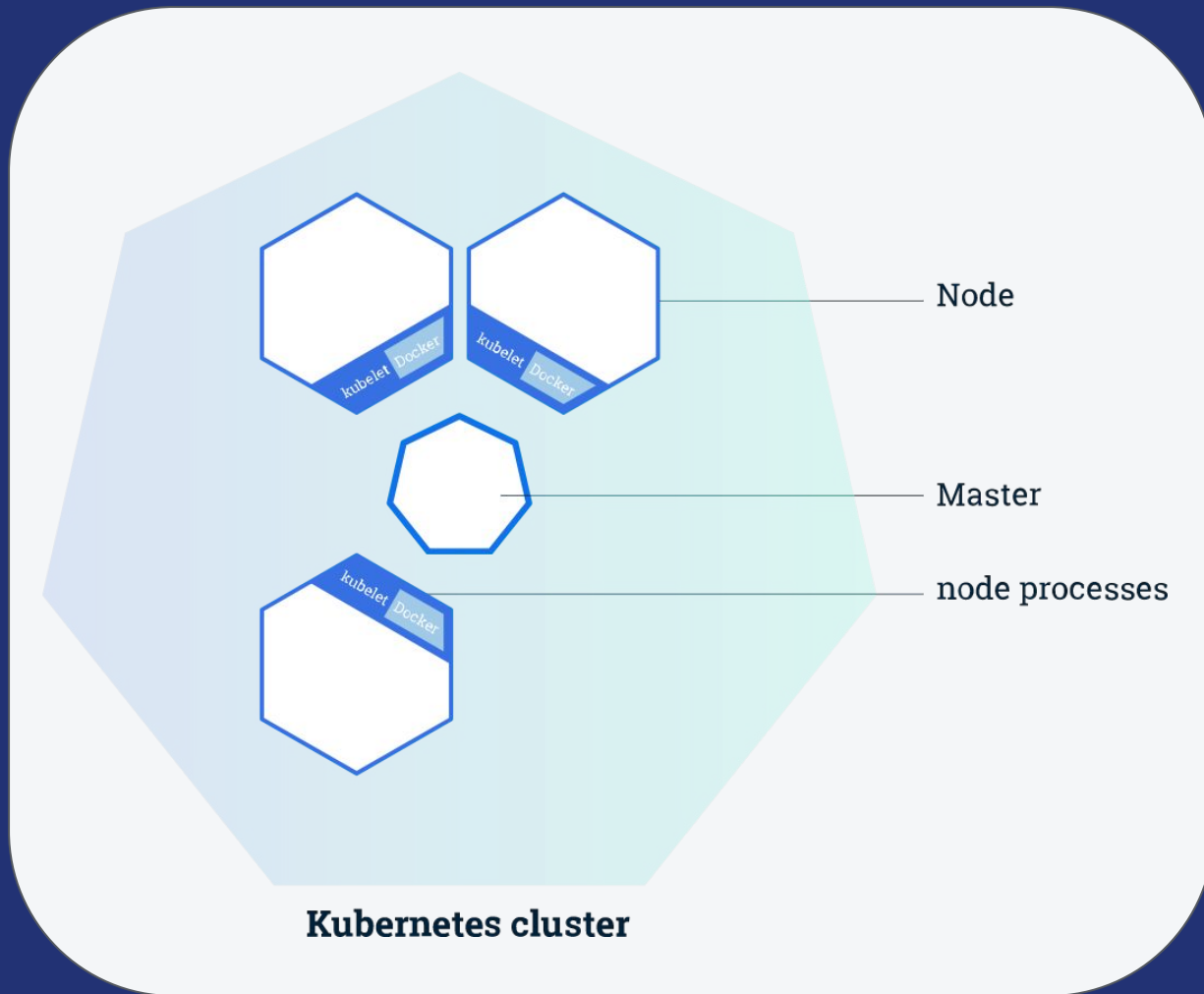
## Kritis

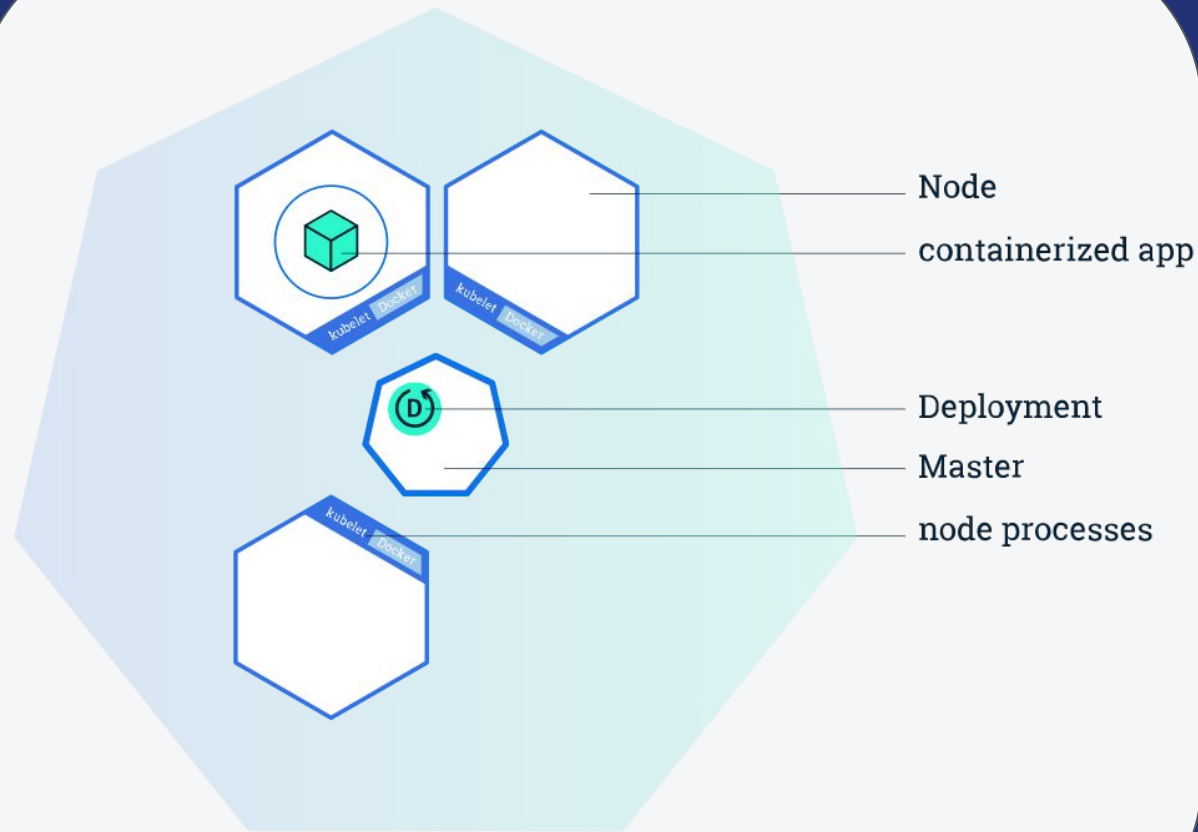
- Use metadata stored in Grafeas to create policies
- Real-time enforcement of policies on Kubernetes

```
1  {
2    "createTime": "2017-09-14T04:34:47.777125Z",
3    "kind": "PACKAGE_VULNERABILITY",
4    "name": "projects/myproject/occurrences/randomID",
5    "noteName": "providers/myscanner/notes/CVE-2017-13036",
6    "resourceUrl": "https://gcr.io/myproject/image@sha256:hash",
7    "updateTime": "2017-09-14T04:34:47.777125Z",
8    "vulnerabilityDetails": {
9      "cvssScore": 7.5,
10     "packageIssue": [
11       {
12         "affectedLocation": {
13           "cpeUri": "cpe:/o:canonical:ubuntu_linux:16.04",
14           "package": "tcpdump",
15           "version": {
16             "name": "4.9.0",
17             "revision": "1ubuntu1-ubuntu16.04.1"
18           }
19         },
20         "fixedLocation": {
21           "cpeUri": "cpe:/o:canonical:ubuntu_linux:16.04",
22           "package": "tcpdump",
23           "version": {
24             "name": "4.9.2",
25             "revision": "0ubuntu0.16.04.1"
26           }
27         },
28         "severityName": "LOW"
29       }
30     ],
31     "severity": "HIGH"
32   }
33 }
```

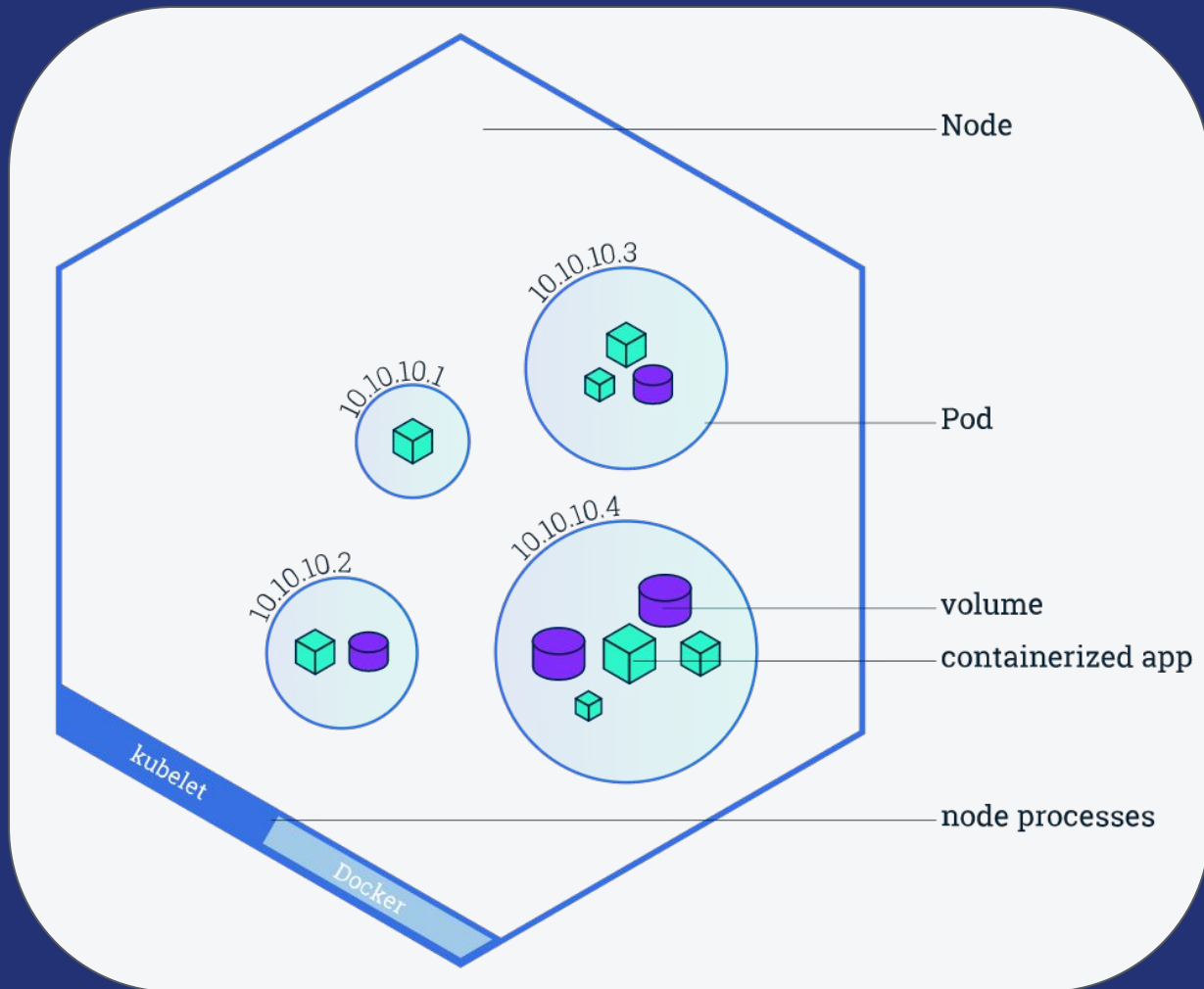
How do we **deploy** containers?

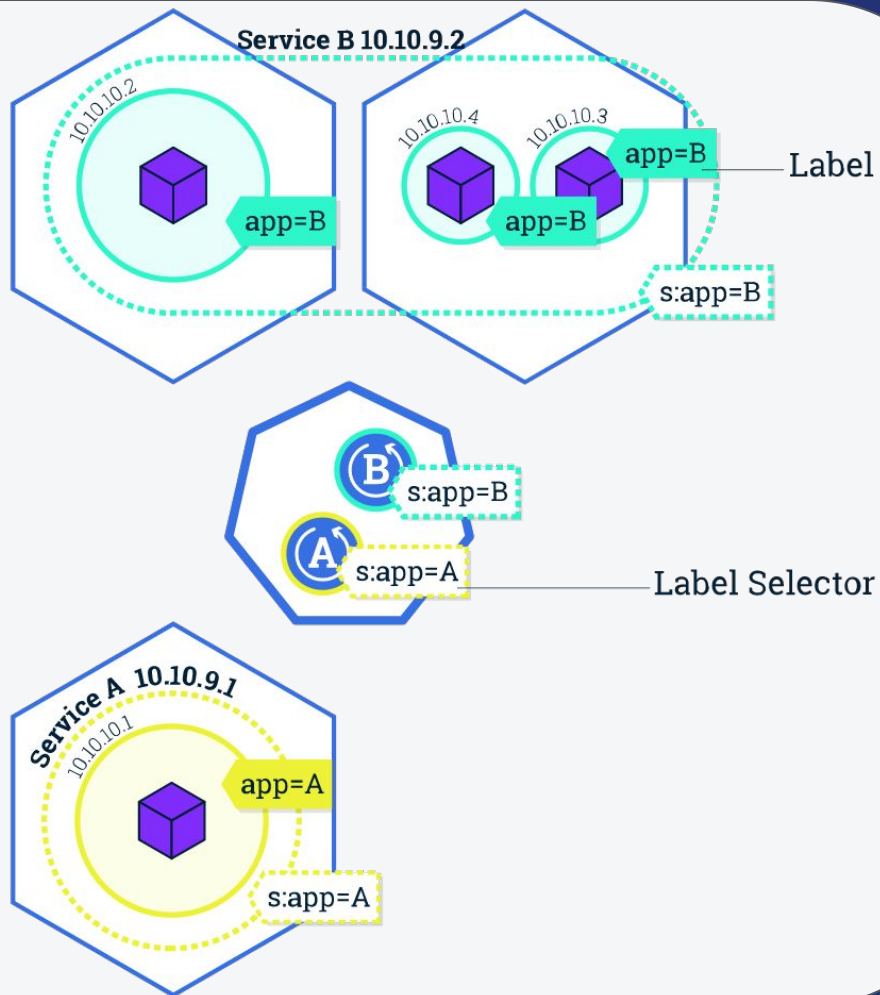






**Kubernetes Cluster**







# jonpulsifer/debian:sudo

FROM debian:stretch

RUN useradd jawn

RUN apt -yq update \  
 && apt -yq upgrade \  
 && apt -yq install sudo

RUN echo "jawn ALL=(ALL) ALL" >> /etc/sudoers \  
 && echo "jawn:password" | chpasswd

USER jawn

# Kubernetes Manifests

```
apiVersion: v1
kind: Pod
metadata:
  name: sudo
  labels:
    app: sudo
    env: staging
spec:
  containers:
  - name: sudo
    image: jonpulsifer/debian:sudo
    command: ["tail", "-f", "/dev/null"]
    securityContext:
      allowPrivilegeEscalation: false
```

## allowPrivilegeEscalation: false

```
~/infrasec/sudo on master ☁ secure-the-cloud/cloudlab/lab
```

```
> kubectl create -f pod.yaml  
pod "sudo" created
```

```
~/infrasec/sudo on master ☁ secure-the-cloud/cloudlab/lab
```

```
> kubectl get pods --show-labels
```

NAME	READY	STATUS	RESTARTS	AGE	LABELS
sudo	1/1	Running	0	8s	app=sudo,env=staging

```
~/infrasec/sudo on master ☁ secure-the-cloud/cloudlab/lab
```

```
> kubectl exec -i --tty sudo /bin/bash
```

```
jawn@sudo:/$ uname -a
```

```
Linux sudo 4.4.64+ #1 SMP Wed Aug 30 20:27:36 PDT 2017 x86_64 GNU/Linux
```

```
jawn@sudo:/$ sudo
```

```
sudo: effective uid is not 0, is /usr/bin/sudo on a file system with the  
'nosuid' option set or an NFS file system without root privileges?
```

```
jawn@sudo:/$ # :(  
#
```

# Pod Security Policies

```
apiVersion: extensions/v1beta1
kind: PodSecurityPolicy
metadata:
  name: restricted
spec:
  allowPrivilegeEscalation: false
  allowedHostPaths: ["/var/log"]
  defaultAllowPrivilegeEscalation: false
  hostIPC: false
  hostNetwork: false
  hostPID: false
  privileged: false
  readOnlyRootFilesystem: true
  runAsUser:
    rule: MustRunAsNonRoot
  volumes: ["secret", "downwardAPI", "configMap"]
```

requiredDropCapabilities:

- AUDIT\_WRITE
- CHOWN
- DAC\_OVERRIDE
- FOWNER
- FSETID
- KILL
- MKNOD
- NET\_BIND\_SERVICE
- NET\_RAW
- SETFCAP
- SETGID
- SETUID
- SETPCAP
- SYS\_CHROOT

# AppArmor

```
#include <tunables/global>

profile deny-write flags=(attach_disconnected) {
    #include <abstractions/base>

    file,

    # Deny all file writes.
    deny /** w,
}
```

# AppArmor + Kubernetes

```
apiVersion: v1
kind: Pod
metadata:
  name: apparmor
  annotations:
    container.apparmor.security.beta.kubernetes.io/sleeper: localhost/deny-write
spec:
  containers:
  - name: sleeper
    image: busybox
    command: ["sh", "-c", "echo 'sleepy time :)' && sleep 3600"]
```

# seccomp

```
"seccomp": {  
  "defaultAction": "SCMP_ACT_ALLOW",  
  "architectures": [  
    "SCMP_ARCH_X86_64"  
  ],  
  "syscalls": [  
    {  
      "names": [  
        "getcwd",  
        "chmod"  
      ],  
      "action": "SCMP_ACT_ERRNO"  
    }  
  ]  
}
```

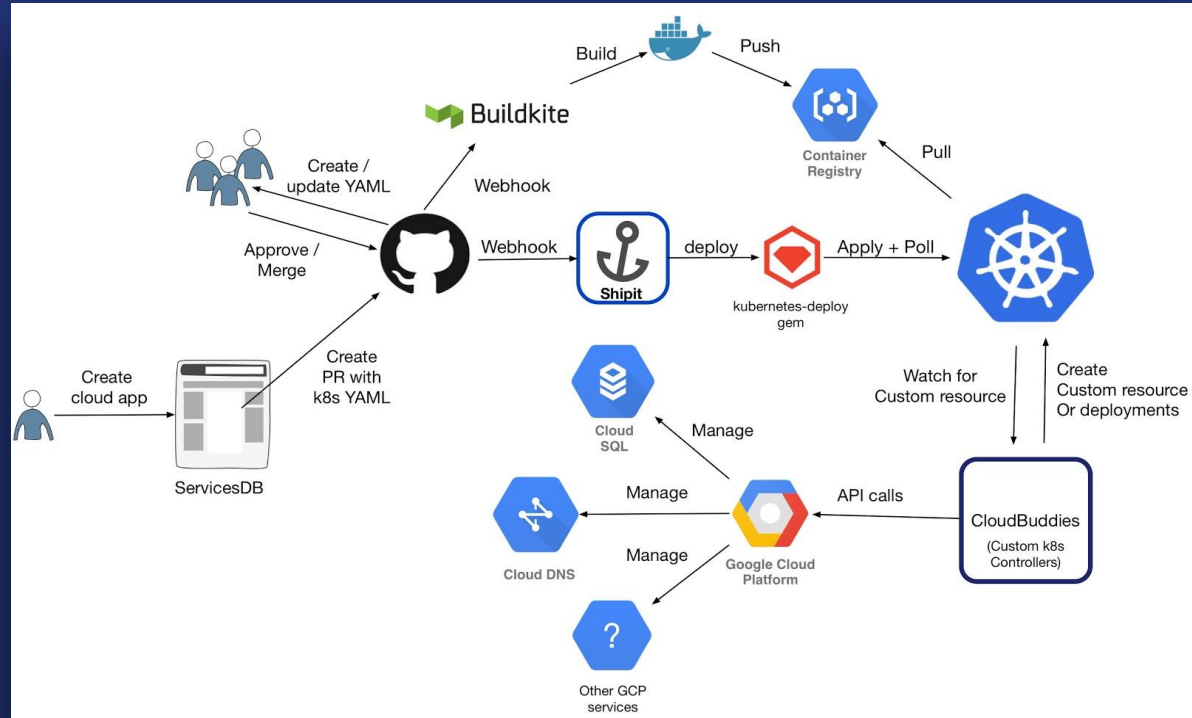


# seccomp + Kubernetes

```
apiVersion: v1
kind: Pod
metadata:
  name: seccomp
  annotations:
    container.seccomp.security.alpha.kubernetes.io/persistence: runtime/default
spec:
  containers:
  - name: persistence
    image: alpine:3.6
    command: ["tail", "-f", "/dev/null"]
```

**So what?**

# Cloud Platform Architecture



# Thanks!

@JonPulsifer

