Infrastructure Security 2.0

Jonathan Pulsifer

shopify

$ whoami

- Infrastructure Security Engineer @ Shopify
- Certified Kubernetes Administrator
- twitter.com/JonPulsifer
- 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

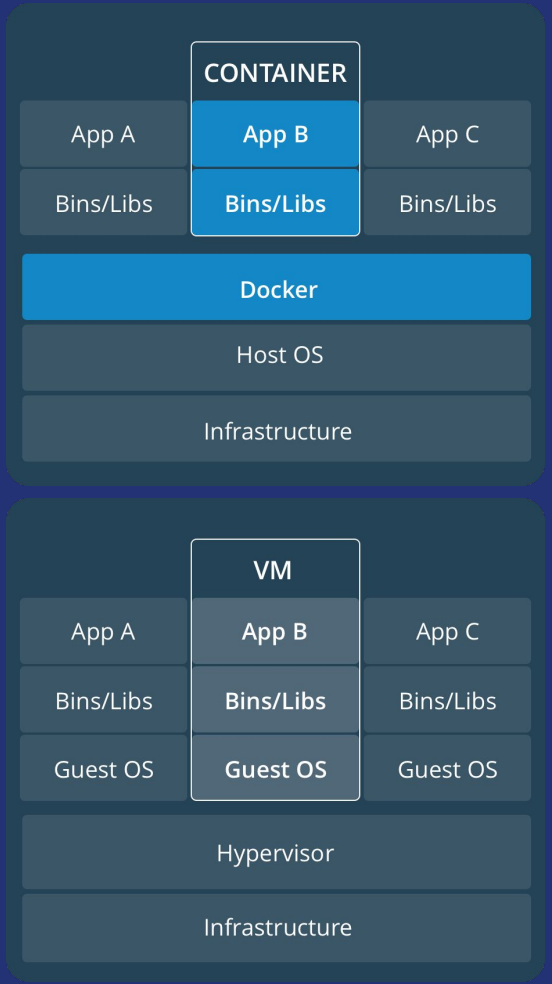
What Containers Are NOT

- 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!)



https://www.docker.com/what-container

Linux Namespaces

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

Control Groups

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

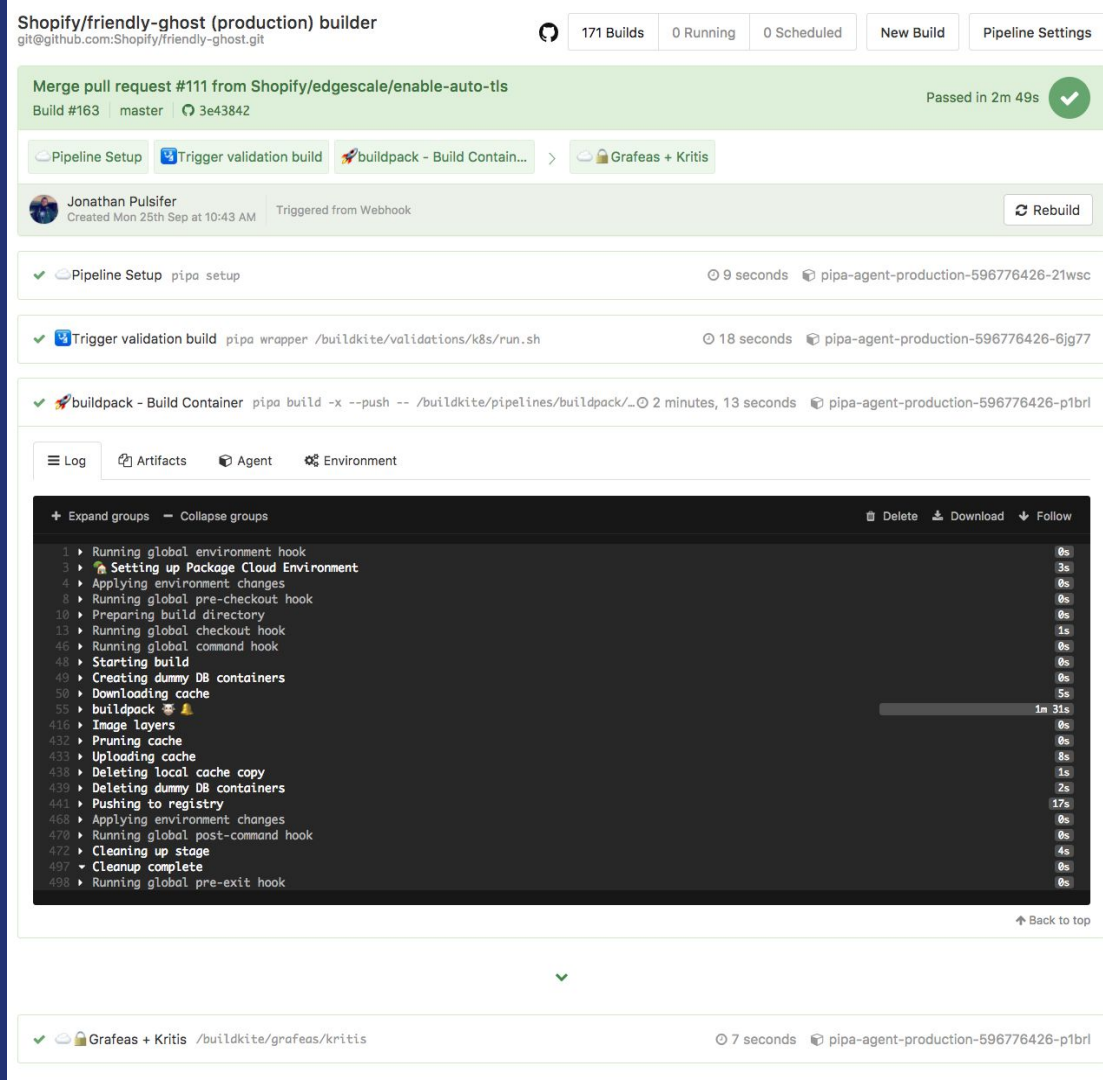
average builds per weekday

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



Grafeas

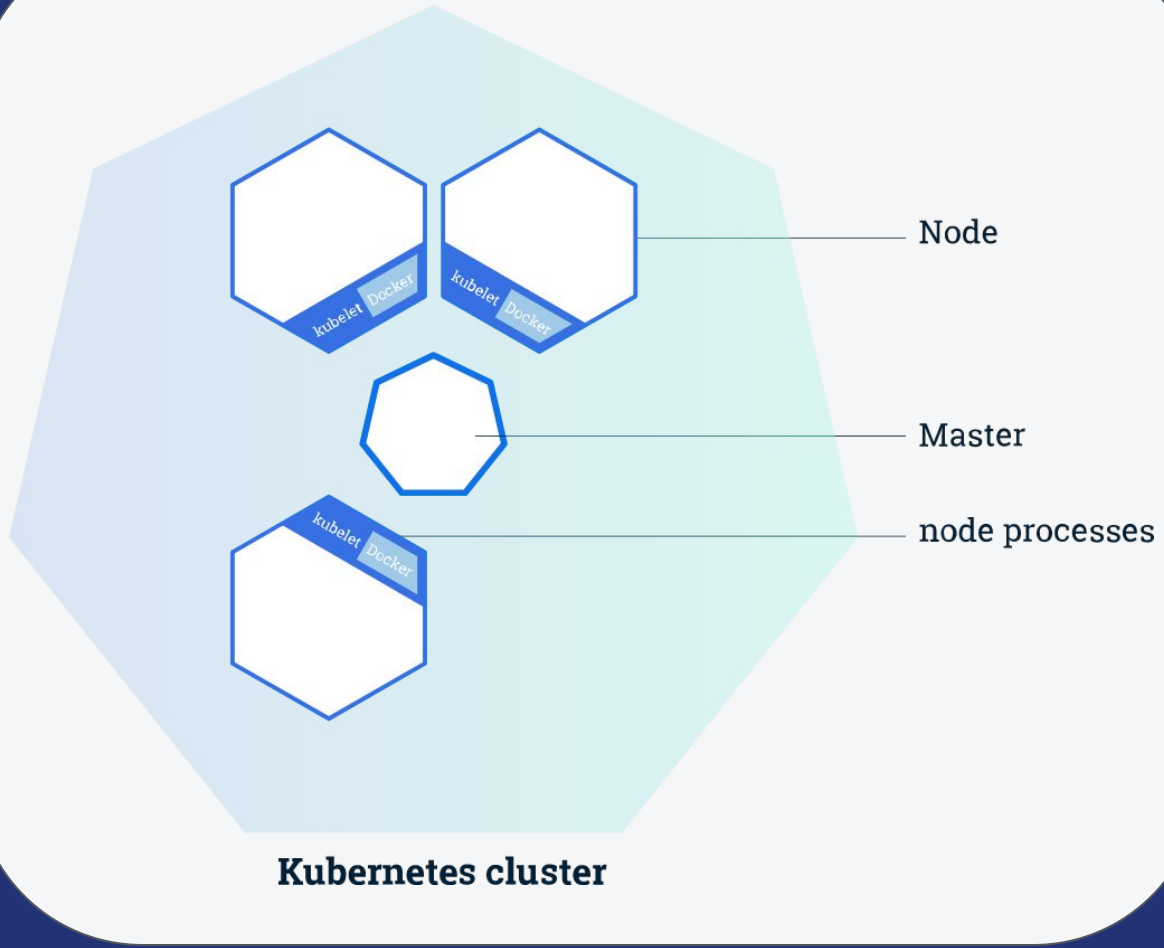
- https://github.com/Grafeas/Grafeas
- Central source of truth for software component metadata
- my.regist.ry/image@sha256:hash as key for containers
- Container notes produced at build
- See GCP's or Shopify's Engineering blog for more

Kritis

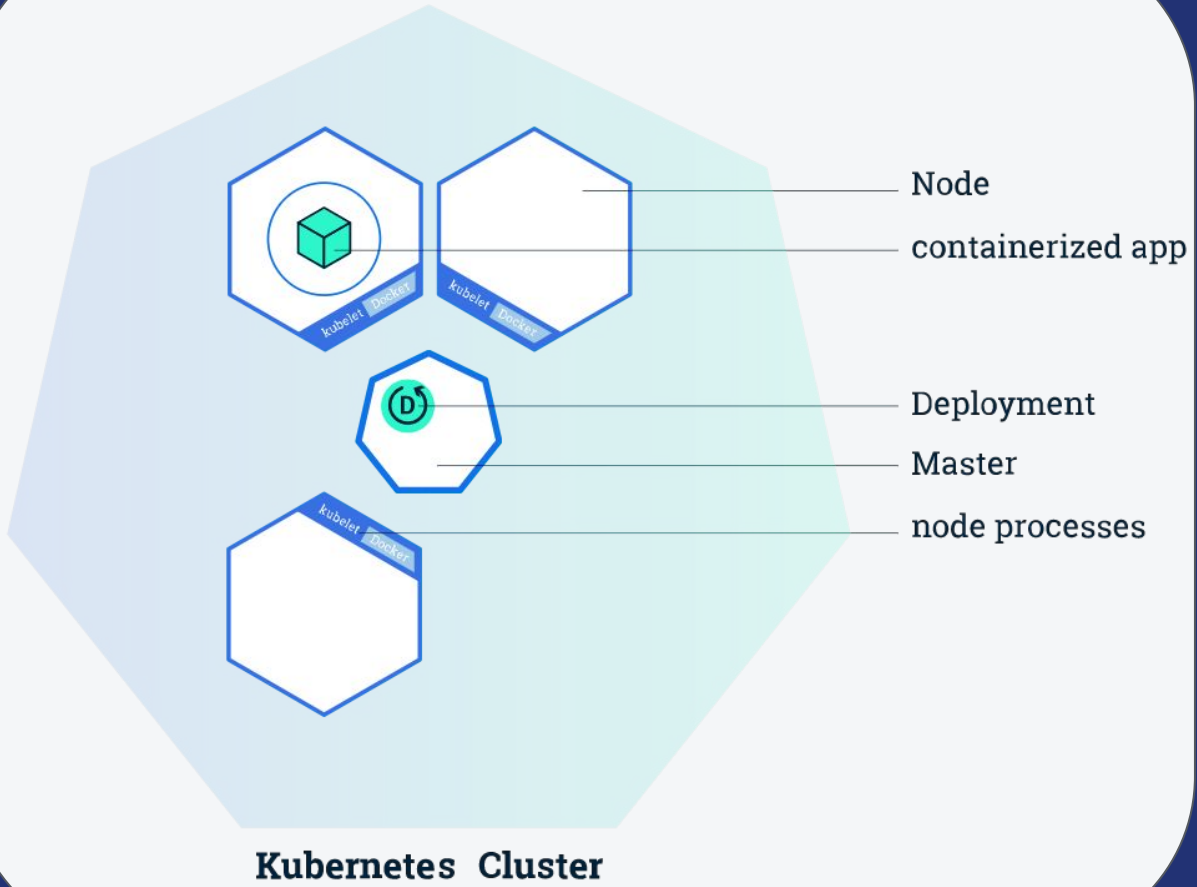
- Use metadata stored in Grafeas to create policies
- Real-time enforcement of policies on Kubernetes


```
{
    "createTime": "2017-09-14T04:34:47.777125Z",
    "kind": "PACKAGE_VULNERABILITY",
    "name": "projects/myproject/occurrences/randomID",
    "noteName": "providers/myscanner/notes/CVE-2017-13036",
    "resourceUrl": "https://gcr.io/myproject/image@sha256:hash",
    "updateTime": "2017-09-14T04:34:47.777125Z",
    "vulnerabilityDetails": {
        "cvssScore": 7.5,
        "packageIssue": [
            {
                "affectedLocation": {
                    "cpeUri": "cpe:/o:canonical:ubuntu_linux:16.04",
                    "package": "tcpdump",
                    "version": {
                        "name": "4.9.0",
                        "revision": "1ubuntu1~ubuntu16.04.1"
                    }
                },
                "fixedLocation": {
                    "cpeUri": "cpe:/o:canonical:ubuntu_linux:16.04",
                    "package": "tcpdump",
                    "version": {
                        "name": "4.9.2",
                        "revision": "0ubuntu0.16.04.1"
                    }
                },
                "severityName": "LOW"
            }
        ],
        "severity": "HIGH"
    }
}
```

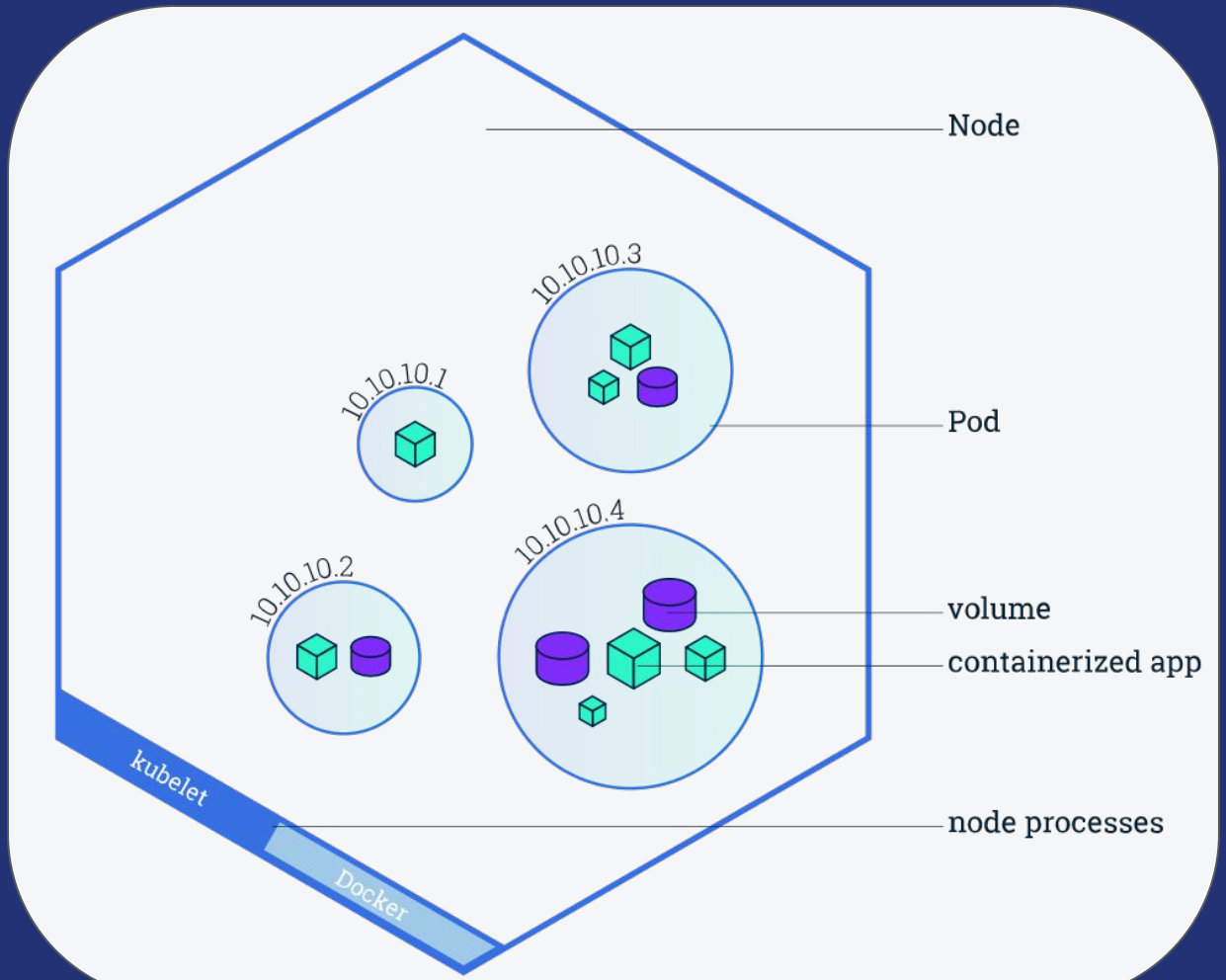
How do we deploy containers?



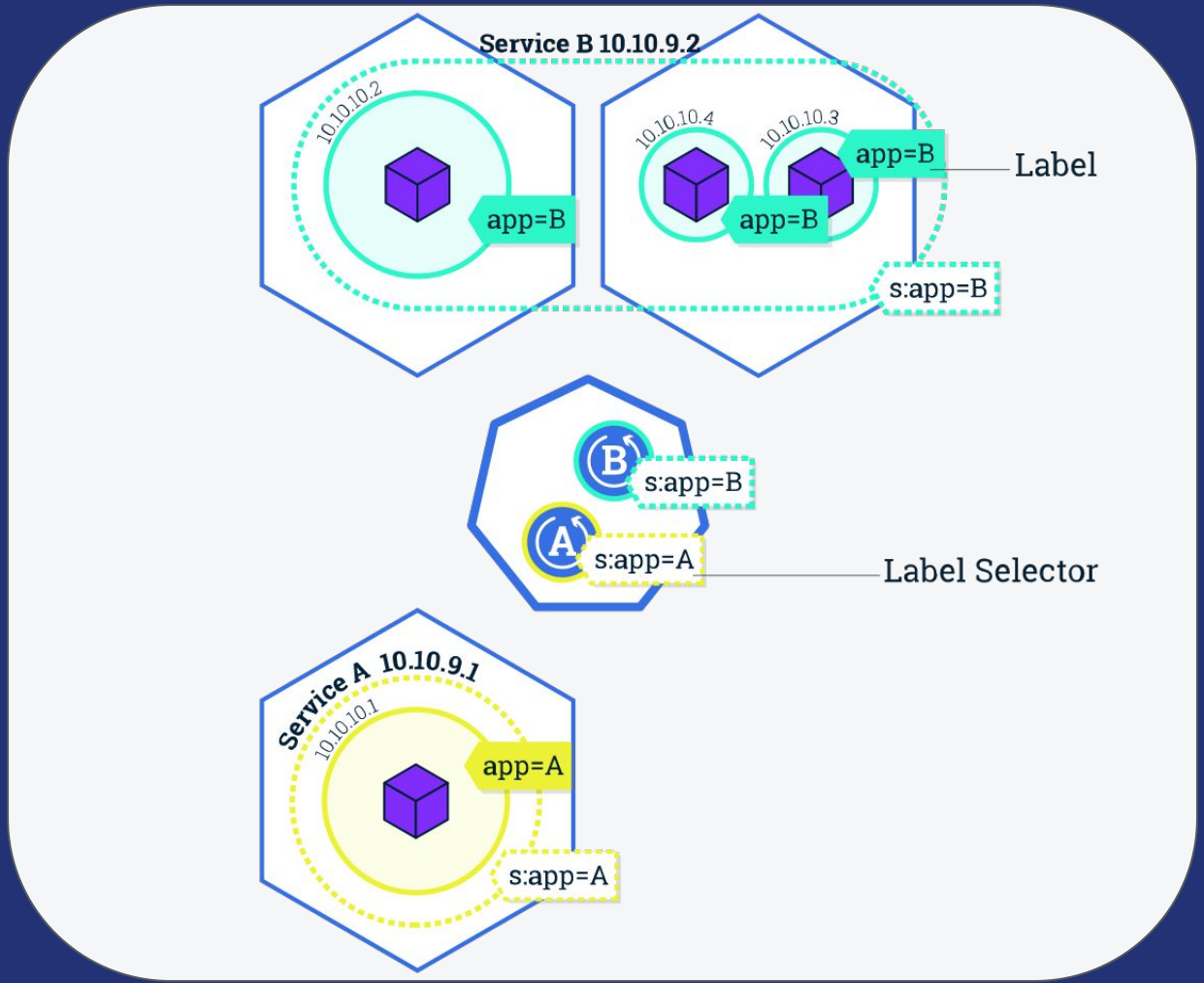
https://kubernetes.io/docs/tutorials/kubernetes-basics/



https://kubernetes.io/docs/tutorials/kubernetes-basics/



https://kubernetes.io/docs/tutorials/kubernetes-basics/



https://kubernetes.io/docs/tutorials/kubernetes-basics/

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

```yaml
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

```yaml
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"]
```

```yaml
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

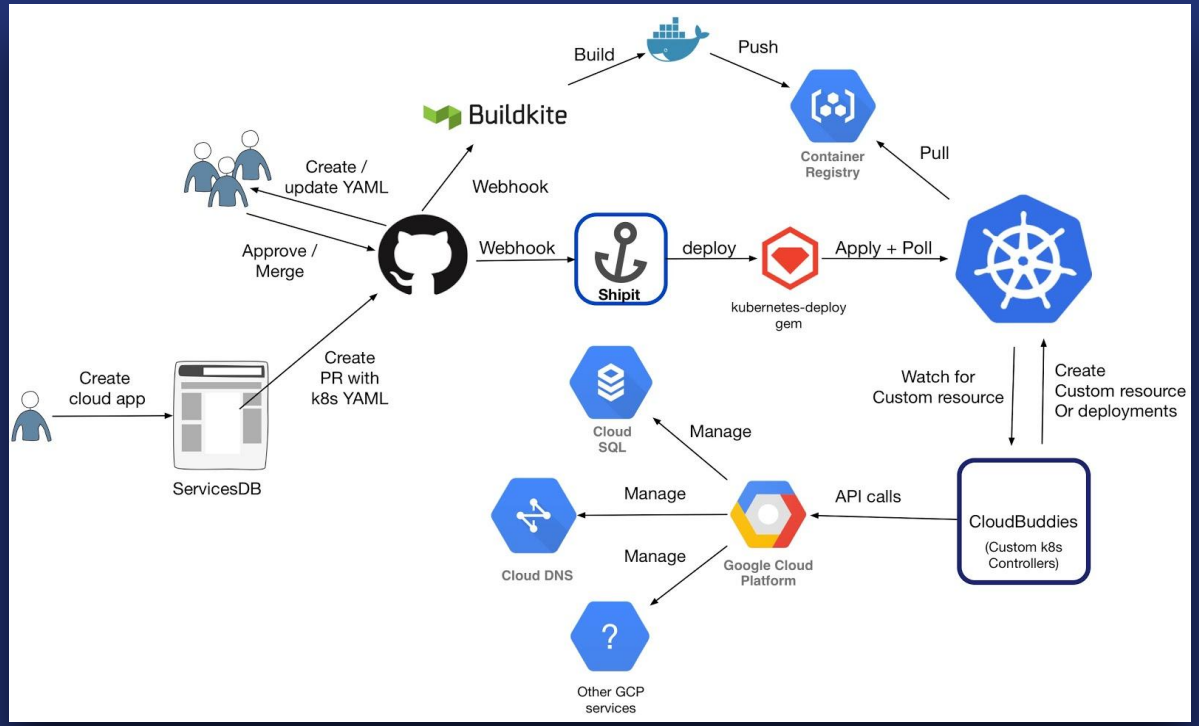
```
"seccomp": {
    "defaultAction": "SCMP_ACT_ALLOW",
    "architectures": [
        "SCMP_ARCH_X86_64"
    ],
    "syscalls": [
        {
            "names": [
                "getcwd",
                "chmod"
            ],
            "action": "SCMP_ACT_ERRNO"
        }
    ]
}
```

seccomp + Kubernetes

```yaml
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