



KubeCon



CloudNativeCon

Europe 2023





KubeCon



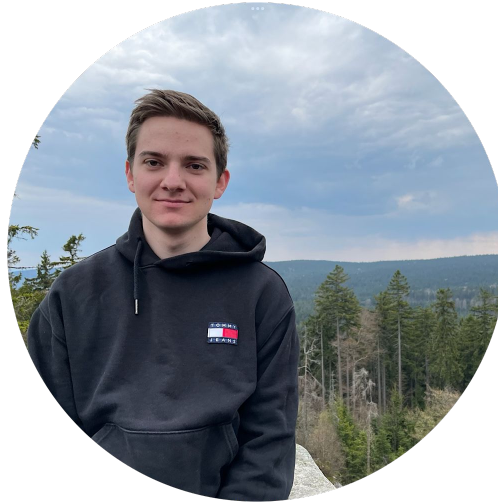
CloudNativeCon

Europe 2023

Running Non-root Made Easy

Luboslav Pivarč, Red Hat





L'uboslav Pivarč - @xpivarc
Software Engineer Red Hat

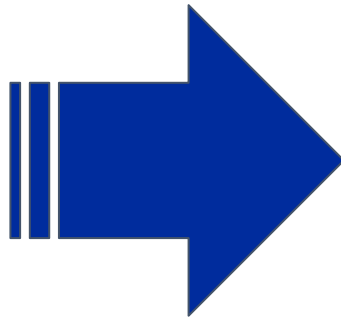
- Kubernetes extension that allows running traditional VM workloads natively side by side with Container workloads
- Runs VMs inside Pods as regular workloads
- Transitioned to non-root Pods



- Principle of least privilege
- Well known way to prevent CVEs
- Pod security standards & other auditing tools
- Security compliance
- Root in the container is root on the Host
- Kubernetes doesn't support user namespaces yet

The goal


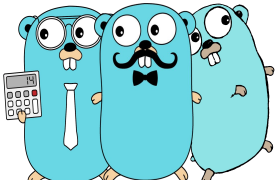

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx-container
    securityContext:
      runAsUser: 0
```



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx-container
    securityContext:
      runAsUser: 1000
```

Access control

UID : GID

USER : GROUP : OTHERS(world)

rwX : rwX : rwX

- The first thing that goes wrong
- You either consume already existing images or you build them
 - Depending on it you have different level of control
 - Always able to adjust to your needs
- There are many tools for building images
 - You can reuse package managers
 - Compile from the source


```
^C[kubecon@fedora kubecon]$ kubectl logs nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: can not modify /etc/nginx/conf.d/default.conf (read-only
file system?)
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/04/15 13:35:40 [warn] 1#1: the "user" directive makes sense only if the master process runs
with super-user privileges, ignored in /etc/nginx/nginx.conf:2
nginx: [warn] the "user" directive makes sense only if the master process runs with super-user
privileges, ignored in /etc/nginx/nginx.conf:2
2023/04/15 13:35:40 [emerg] 1#1: mkdir() "/var/cache/nginx/client_temp" failed (13: Permission
denied)
nginx: [emerg] mkdir() "/var/cache/nginx/client_temp" failed (13: Permission denied)
```

Images - Solution #1

Strace - Used to trace system calls and signals. It can be used to determine what paths you need to adjust.

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - image: <custom-image-with-strace>-nginx
    name: nginx-container
    command:
    - strace
    args:
    - nginx
    securityContext:
      runAsUser: 0
```

```
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 5
mkdir("/var/cache/nginx/client_temp", 0700) = 0
stat("/var/cache/nginx/client_temp", {st_mode=S_IFDIR|0700, st_size=4096, ...}) = 0
```

Images - Solution #2

- EmptyDir - Empty volume bound to Pod lifecycle

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  volumes:
    - name: nginx-cache
      emptyDir:
        sizeLimit: 1Gi
  containers:
    - name: nginx-container
      volumeMounts:
        - mountPath: /var/cache/nginx
          name: nginx-cache
      ...
```

- This can't be used for directories which already has some content!

Images - Tip #1

- Use tools for inspecting image layers such as <https://github.com/wagoodman/dive>
- This reduces the turnaround time and helps you to understand file system layout

Layers			Current Layer Contents			
Cmp	Size	Command	Permission	UID:GID	Size	Filetree
	80 MB	FROM c2308daad2569cf	drwxr-xr-x	0:0	5.3 MB	bin
	62 MB	set -x && addgroup --system --gid 101 nginx &&	-rwxr-xr-x	0:0	1.2 MB	bash
	1.6 kB	#(nop) COPY file:7b307b62e82255f040c9812421a30090bf9abf	-rwxr-xr-x	0:0	44 kB	cat
	2.1 kB	#(nop) COPY file:5c18272734349488bd0c94ec8d382c872c1a0a	-rwxr-xr-x	0:0	73 kB	chgrp
	1.3 kB	#(nop) COPY file:abbcbf84dc17ee4454b6b2e3cf914be88e02cf	-rwxr-xr-x	0:0	64 kB	chmod
	4.6 kB	#(nop) COPY file:e57eef017a414ca793499729d80a7b9075790c	-rwxr-xr-x	0:0	73 kB	chown
	21 MB	apt update && apt install -y strace	-rwxr-xr-x	0:0	151 kB	cp
	2.3 MB	apt install -y libcap2-bin	-rwxr-xr-x	0:0	126 kB	dash
	1.4 MB	setcap 'cap_net_bind_service+eip' /usr/sbin/nginx	-rwxr-xr-x	0:0	114 kB	date
	0 B	mkdir /mydir	-rwxr-xr-x	0:0	81 kB	dd
	12 B	echo "Hello World" >> /mydir/myfile	-rwxr-xr-x	0:0	94 kB	df
Layer Details			-rwxr-xr-x	0:0	147 kB	dir
Tags: (unavailable)			-rwxr-xr-x	0:0	84 kB	dmesg
Id: c2308daad2569cff493c06e55a1b43c5a613c012752001f2b4f3a1126c4			-rwxrwxrwx	0:0	0 B	dnsdomainname → hostname
Digest: sha256:ed7b0ef3bf5bbec74379c3ae3d5339e666a314223e863c70644f			-rwxrwxrwx	0:0	0 B	domainname → hostname
Command: #(nop) ADD file:11b1acca3f68b5c5787e292ff8dbdd114964a7272bf3519ab07			-rwxr-xr-x	0:0	40 kB	echo
Image Details			-rwxr-xr-x	0:0	28 B	egrep
Image name: quay.io/lpivarc/nginx-strace:v4			-rwxr-xr-x	0:0	40 kB	false
Total image size: 167 MB			-rwxr-xr-x	0:0	28 B	fgrep
Potential wasted space: 10 MB			-rwxr-xr-x	0:0	69 kB	findmnt
Image efficiency score: 95 %			-rwxr-xr-x	0:0	203 kB	grep
Count Total Space Path			-rwxr-xr-x	0:0	2.3 kB	gunzip
^C Quit Tab Switch view ^F Filter ^L Show layer changes ^A Show aggregated changes			-rwxr-xr-x	0:0	6.4 kB	gzexe
			-rwxr-xr-x	0:0	98 kB	gzip
			-rwxr-xr-x	0:0	23 kB	hostname
			-rwxr-xr-x	0:0	73 kB	ln
			-rwxr-xr-x	0:0	57 kB	login
			-rwxr-xr-x	0:0	147 kB	ls

Images - Tip #2

- Build efficient images - DOCKERFILE COPY --chown

```
FROM fedora
COPY sample.txt sample.txt
RUN chown 1000:1000 sample.txt
```

2337864017 VS 1264117582

```
FROM fedora
COPY --chown=1000:1000 sample.txt sample.txt
```



```
nginx: [warn] the "user" directive makes sense only if the master
process runs with super-user privileges, ignored in /etc/nginx/n
ginx.conf:2
2023/04/16 11:04:28 [emerg] 1#1: bind() to 0.0.0.0:80 failed (13:
Permission denied)
nginx: [emerg] bind() to 0.0.0.0:80 failed (13: Permission denied
)
```

Capabilities

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  volumes:
  - name: nginx-cache
    emptyDir:
      sizeLimit: 1Gi
  containers:
  - name: nginx-container
    securityContext:
      runAsUser: 1000
      capabilities:
        add:
        - "NET_BIND_SERVICE"
  ...
```

- Capability - “Ability to execute a specified course of actions or to achieve certain outcomes”
- Capabilities are reason why we distinguish two types of processes
 - Unprivileged
 - Privileged

- There are sets of capabilities that a process can have
 - **Permitted** - Limiting superset of capabilities that can be in effective set
 - **Effective** - Set which indicates what capabilities a process have
 - Inheritable
 - Bounding
 - Ambient

Programmatically adjusting capability sets

- Application needs to be aware of capabilities
- The steps:
 1. Find out what capabilities the application requires
 2. Look at the Effective set
 3. Ask Kernel for missing capabilities in Effective set
 4. Continue with application logic

File capabilities

- Requires modification of the image
- We need to set the capability on the application's binary
- `setcap (8)`

Capabilities - Future

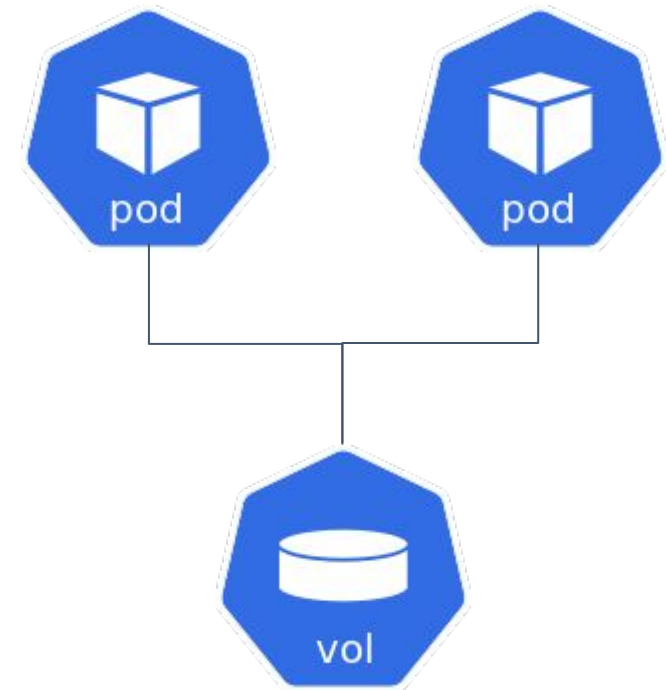
- [KEP 2763: Ambient capabilities in Kubernetes](#)
- [Issue #56374: Kubernetes should configure the ambient capability set](#)
- [CRI API updated](#)

- The device inherits user & group from the host
- The device is accessible only if the container user matches the host assigned
- [Kubernetes blog by Mikko Ylinen](#)
- CRI-O/Containerd needs to be adjusted

```
[crio.runtime]  
device_ownership_from_security_context = true
```

Persistence Volumes & Claims

- The provisioned filesystem doesn't have standard permissions
- Two Pods sharing volume needs to run with same uid & gid
- fsGroup - Recursive change to the specified group + use of Setgid bit



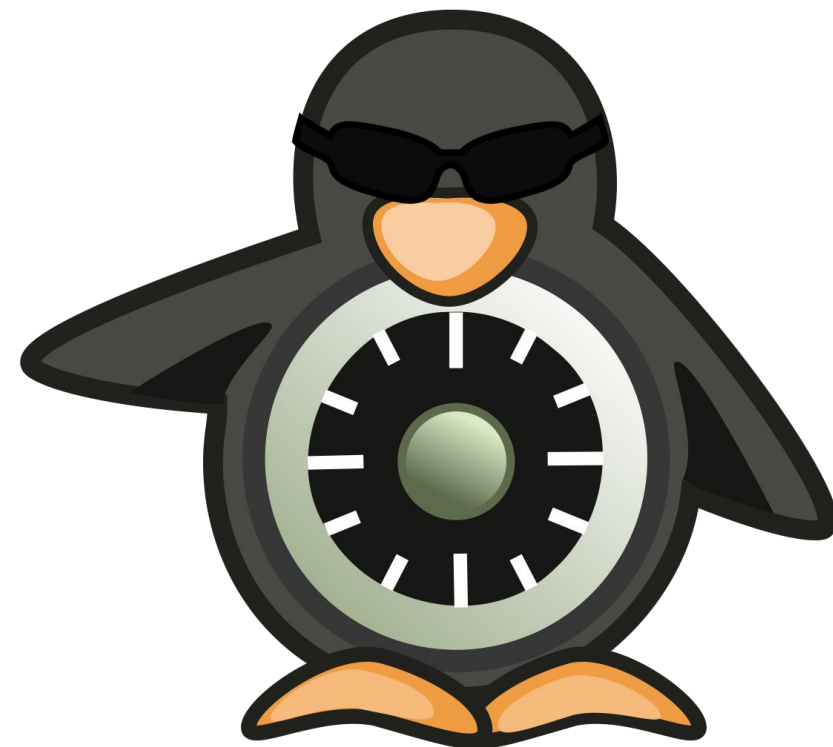
Persistence Volumes & Claims

- What about block volumes?
 - fsGroup = only filesystem
- Block volumes are devices, therefore same solution as for devices applies

SELinux, do not “setenforce 0”

- Be aware of what SELinux label is set on binaries inside a layer
- Sharing a filesystem between Pods? You need to match labels and categories (MCS)

`system_u:object_r:container_file_t:s0:c208,c620`





KubeCon



CloudNativeCon

Europe 2023

Thank you for your attention!



KubeCon



CloudNativeCon

Europe 2023



**Thank you to our Session Recording
Sponsor:**

