CRI-O: Look Ma, No Pause

Peter Hunt
Mrunal Patel

What is Pause?

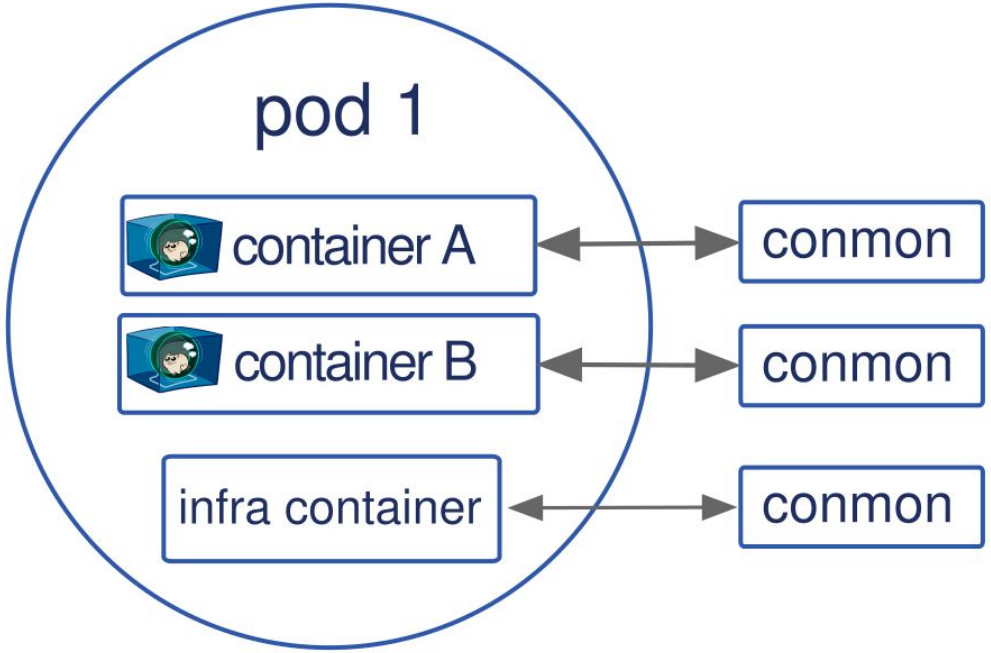
- If you have ever run docker ps on a kubernetes node, you would have seen a mysterious pause image for every single one of your pods

- A container that sits and sleeps (in most cases)
  - yes, that's really it

```
~ # docker ps
CONTAINER ID        IMAGE                                  COMMAND                  CREATED             STATUS              PORTS               NAMES
22e75302b212        6dc8ef8287d3                           "/dnsmasq-nanny -v=2…"   About a minute ago  Up About a minute                       k8s_dnsmasq_kube-dns-58bc79b99f-7v6rx_kube-system_72f6ac64-3ff9-4a08-964b-f205e9225ec6_1
68d0f4a16d1a        k8s.gcr.io/kubernetes-dashboard-amd64  "/dashboard --insecu…"   3 minutes ago       Up 2 minutes                            k8s_kubernetes-dashboard_kubernetes-dashboard-864d864f44-dbxql_kube-system_7abbe45d-c813-47e
4-b300-f09aa7f779ae_0
bb9fa857fb46        4b2e93f0133d                           "/sidecar --v=2 --lo…"   3 minutes ago       Up 3 minutes                            k8s_sidecar_kube-dns-58bc79b99f-7v6rx_kube-system_72f6ac64-3ff9-4a08-964b-f205e9225ec6_0
6a4dc8ed229a        55a3c5209c5e                           "/kube-dns --domain=…"   3 minutes ago       Up 3 minutes                            k8s_kubedns_kube-dns-58bc79b99f-7v6rx_kube-system_72f6ac64-3ff9-4a08-964b-f205e9225ec6_0
e8e7f204370e        k8s.gcr.io/pause:3.2                   "/pause"                 3 minutes ago       Up 3 minutes                            k8s_POD_kube-dns-58bc79b99f-7v6rx_kube-system_72f6ac64-3ff9-4a08-964b-f205e9225ec6_0
0f6a0e6e033d        k8s.gcr.io/pause:3.2                   "/pause"                 3 minutes ago       Up 3 minutes                            k8s_POD_kubernetes-dashboard-864d864f44-dbxql_kube-system_7abbe45d-c813-47e4-b300-f09aa7f779
ae_0
```

What is Pause?

- Holds pod namespaces and is contained within the pod cgroup
- It is referred to as the pod infra container



History of Kubernetes Pods

- kubernetes started by using docker as the container runtime
  - This was way before the CRI was introduced to support other runtimes

- kubernetes needs one IP per pod. How do you do that with docker?
  - Start a container
  - Join the network namespace of that container and assign it an IP
  - Start other containers in the pod that join the network namespace of the first container
  - That’s how the pause container came to be

History of Kubernetes Pods

- Soon after, support for sharing other namespaces like ipc was added to the pause container
  - It was renamed from netContainer to infraContainer in the code base
  - https://github.com/kubernetes/kubernetes/pull/3817/files

- Eventually, pid namespace sharing was added when pause's job became a little more involved than just holding the namespaces
  - It was also responsible for reaping processes in the pod

Why Drop the Pause?

- Reason #1: Uses up space for binary and container

Per node: not bad

```
 # podman images
REPOSITORY          TAG   IMAGE ID       CREATED        SIZE
k8s.gcr.io/pause    3.2   80d28bedfe5d   5 months ago   688 kB
```

Per pod: not great

```
 $ sudo pmap 9184
9184:   /pause
0000000000400000      4K r---- pause
0000000000401000    496K r-x-- pause
000000000047d000    144K r---- pause
000000000004a1000     28K rw--- pause
000000000004a8000      4K rw---   [ anon ]
0000000000678000    140K rw---   [ anon ]
00007ffd9922e000    132K rw---   [ stack ]
00007ffd99328000     16K r----   [ anon ]
00007ffd9932c000      8K r-x--   [ anon ]
ffffffffff600000      4K r-x--   [ anon ]
 total              976K
```

Why Drop the Pause?

- Reason #2: takes time to create, mount, and start

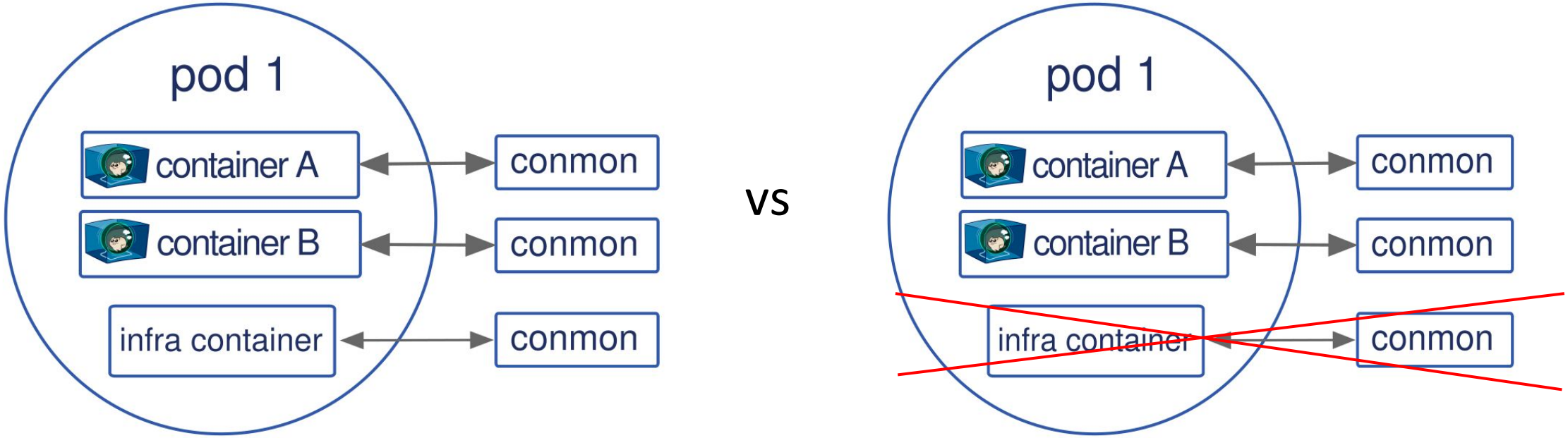
```
[root@localhost cri-o]
 # # first with pause
[root@localhost cri-o]
 # time crictl runp ~/pod_yamls/011c06cc-76f3-4a68-ae94-b42c1f26ef13.json
9e162d8f2dfcfd0b8dfe2934c47532dc170765829a24229cdec52bd87bf617a3

real    0m1.752s
user    0m0.012s
sys     0m0.010s
[root@localhost cri-o]
 # # now without pause
[root@localhost cri-o]
 # time crictl runp ~/pod_yamls/011c06cc-76f3-4a68-ae94-b42c1f26ef13.json
460d64c3cd335c0ecc12403c2e0766754704b9763b09183d201bc3bf090eaf93

real    0m1.684s
user    0m0.006s
sys     0m0.008s
```

Why drop the pause?

- Reason #3: Process management overhead
  - More code in setup and cleanup
  - Tracking why the pause container died (OOM killed?)
  - No process ---> No process management!!!



How to drop the pause?

- Step 1: Find a way to keep the namespaces without a process
  - Linux supports bind mounting namespaces

```
[root@localhost cri-o]
 # unshare --net
[root@localhost cri-o]
 # mount --bind /proc/self/ns/net /var/run/net1
[root@localhost cri-o]
 #
```

```
[root@localhost cri-o]
 # unshare --net=/var/run/net1
[root@localhost cri-o]
 #
```

How to drop the pause?

- Step 2: Apply sysctls for the pod
  - Linux sysctls are namespaced, so we must apply them to our pinned namespaces
  - Examples:
  - NET namespace
    - net.ipv4.ip_forward
    - net.bridge.bridge-nf-call-iptables
  - IPC namespace
    - fs.mqueue.msg_max
    - fs.mqueue.queues_max

How to drop the pause?

KubeCon CloudNativeCon Europe 2020 Virtual

- Step 3: Make pod containers use these namespaces
  - runc config.json points to these bind mounted namespaces

```
"namespaces": [
    {
        "type": "pid"
    },
    {
        "type": "network",
        "path": "/proc/864303/ns/net"
    },
    {
        "type": "ipc",
        "path": "/proc/864303/ns/ipc"
    },
    {
        "type": "uts",
        "path": "/proc/864303/ns/uts"
    },
    {
        "type": "mount"
    }
],
```

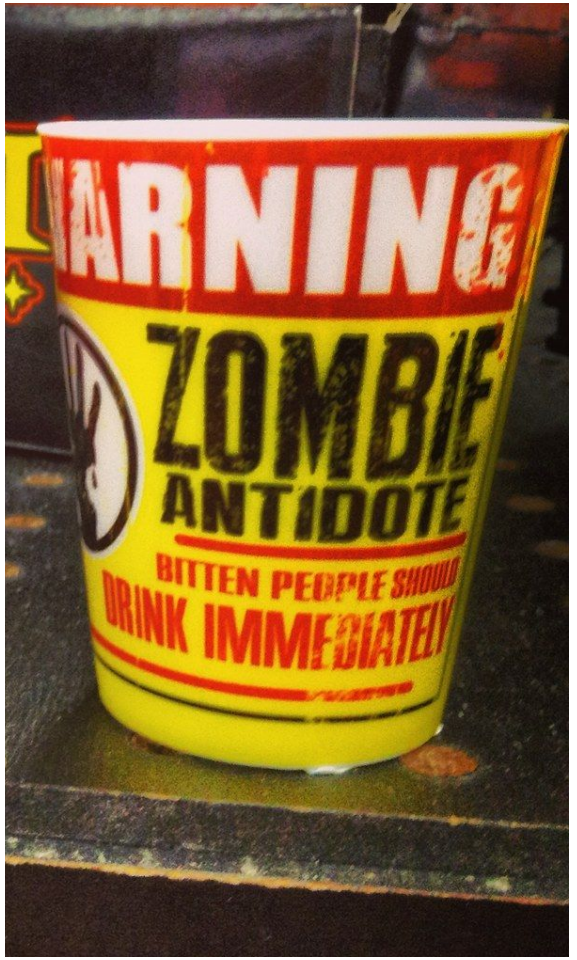
```
"namespaces": [
    {
        "type": "pid"
    },
    {
        "type": "network",
        "path": "/var/run/netns/ee3d680b-dbb7-4bb5-82f3-9f8952a7a88e"
    },
    {
        "type": "ipc",
        "path": "/var/run/ipcns/ee3d680b-dbb7-4bb5-82f3-9f8952a7a88e"
    },
    {
        "type": "uts",
        "path": "/var/run/utsns/ee3d680b-dbb7-4bb5-82f3-9f8952a7a88e"
    },
    {
        "type": "mount"
    }
],
```

Introducing pinNS

- https://github.com/cri-o/cri-o/tree/master/pinns
- Inspired by `ip netns ...` commands, and containernetworking ns package
- exec'ed from CRI-O
  - C plays nicer with namespaces than Go
- CRI-O takes the mounted namespaces from pinns, and hands them to the container
- boom, pause dropped
  - ..mostly

PID Namespaces

- PID 1 is responsible for reaping children
  - For this case (pod level pid namespace) we keep the pause container
- What about have a container process be pid 1?
  - would need to ensure ordering
  - … which the pause container already does
- most have private PID namespaces anyway



CRI-O Journey to Pinned NS

Functionality change

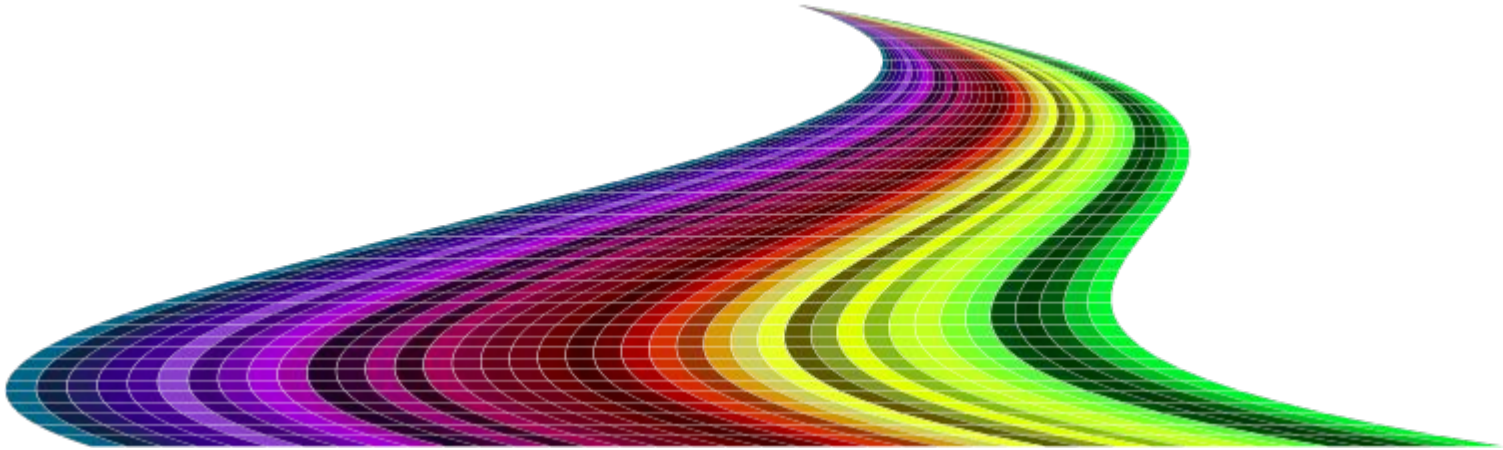
- Configure pods with namespaces in proc
- For Kata, preconfigure network namespace
  - manage_network_ns_lifecycle (CRI-O 1.0)
- For security, configure (most) other namespaces
  - manage_ns_lifecycle (CRI-O 1.17)
- For performance, drop the pause
  - drop_infra (CRI-O 1.19)

CRI-O Sandbox Option

- With Pause
- Kata VM
- Pinned Namespaces
- Dropped Pause

Roadmap

- Experimental support targeted for CRI-O 1.19
- Podman will also use pinns in upcoming releases
- Eventually, pinned namespaces and dropped pause will be the default



https://www.openclipart.org/detail/220006/multi-colored-brick-road

Demo

https://www.openclipart.org/image/2000px/svg_to_png/2814/valessiobrito-Plate-Computer-Privacy.png

Performance Comparison


```
# where JSONDIR has 100 unique pod JSONs

function main() {
   for json in $(ls $JSONDIR); do
         crictl runp $JSONDIR/$json
   done
   wait
   crictl rmp -fa
}
main
```

Performance Comparison

KubeCon CloudNativeCon Europe 2020 Virtual

```
# drop pause
===> multitime results
1: -q /bin/bash -c ./lots_of_pods.bash
            Mean    Std.Dev.Min      Median      Max
real        54.386  1.121   52.143   54.264  56.780
user        1.140   0.038   1.090    1.135   1.227
sys         0.689   0.055   0.626    0.681   0.804

# keep pause
 # multitime -q -n 10 /bin/bash -c ./lots_of_pods.bash
===> multitime results
1: -q /bin/bash -c ./lots_of_pods.bash
            Mean    Std.Dev.Min      Median      Max
real        95.549  1.160   94.252   95.395  98.416
user        1.175   0.046   1.136    1.153   1.293
sys         0.715   0.033   0.652    0.717   0.772
```


KubeCon
CloudNativeCon
Europe 2020
Virtual
HELM
KEEP CLOUD NATIVE
CONNECTED