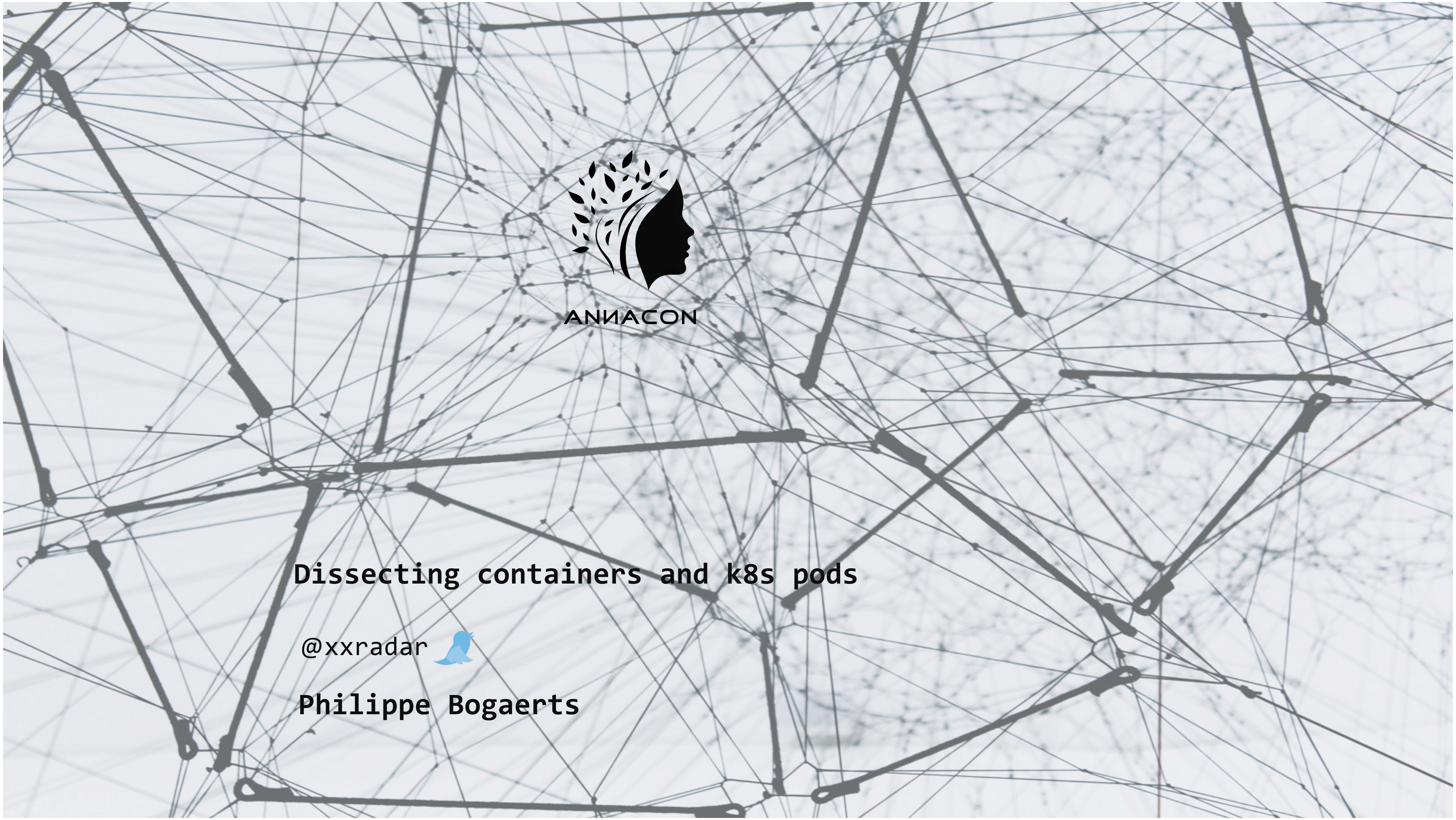


ANNACON

# Dissecting containers and k8s pods

@xxradar

**Philippe Bogaerts**

# What about today’s talk?

It’s all about exploring how container and pods do their magic.

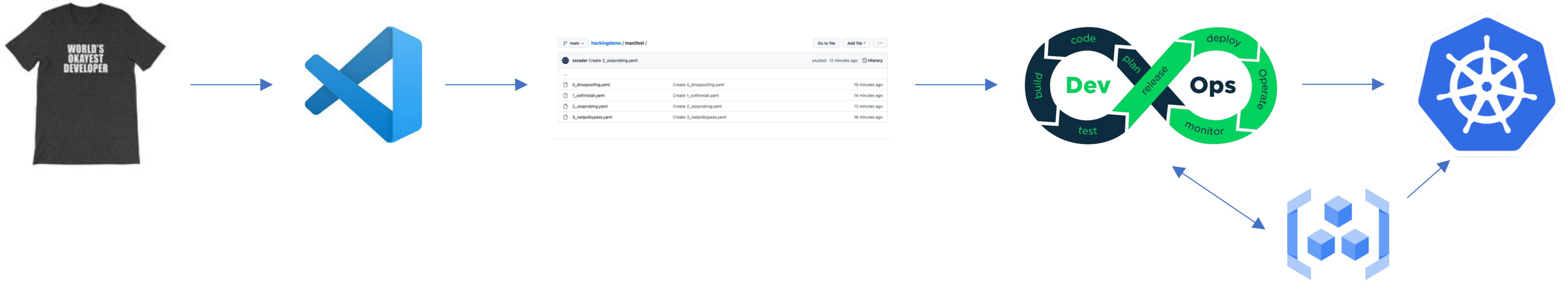
What is a container actually?

How does a container relate to a Kubernetes pod?

Can we better assess the risk when we know things work?

Why do things go terribly wrong?

# From code to prod



main hackingdemo / manifest /    Go to file    Add file    ···

xxradar Create 2_xssprobing.yaml    a4a38d5 12 minutes ago    History

| | | |
|---|---|---|
| .. | | |
| 0_dnsspoofing.yaml | Create 0_dnsspoofing.yaml | 15 minutes ago |
| 1_softinstall.yaml | Create 1_softinstall.yaml | 14 minutes ago |
| 2_xssprobing.yaml | Create 2_xssprobing.yaml | 12 minutes ago |
| 3_netpolbypass.yaml | Create 3_netpolbypass.yaml | 16 minutes ago |

Public Cloud    Internet    Data Center    SaaS    Edge Compute

# # whoami

- Public Cloud Consultant System Engineer EMEA FORTINET
- Co-founder and co-organizer https://brucon.org
- Training and pen-testing https://kubiosec.tech/

Breaking Stuff as a Hobby | Cloud Native Stuff | DevSecOps | Network and Application security | Container and K8S security | K8s Networking | Security Advocate & Research | Low and slow BBQ | Cocktails

https://www.linkedin.com/in/philippebogaerts/ @xxradar

[opinions expressed are solely my own]

# Why are containers so popular ?

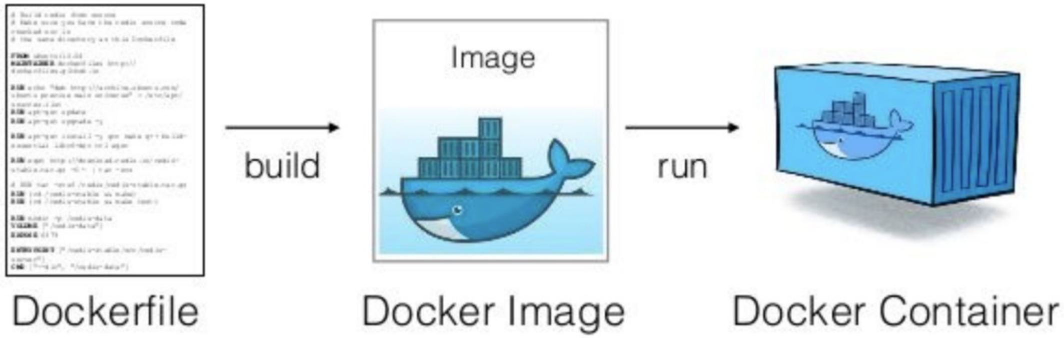
During a Wednesday back in 2016 in SFO during booth duty …
Containers ?? Don't know anything about it … what am I doing here ??

The next Friday evening in SFO airport while waiting for a plane back home,
I googled 'docker', installed docker on my MacBook and "build, ship and run" my first container …
and then I boarded the plane …

# What is Cloud Native?

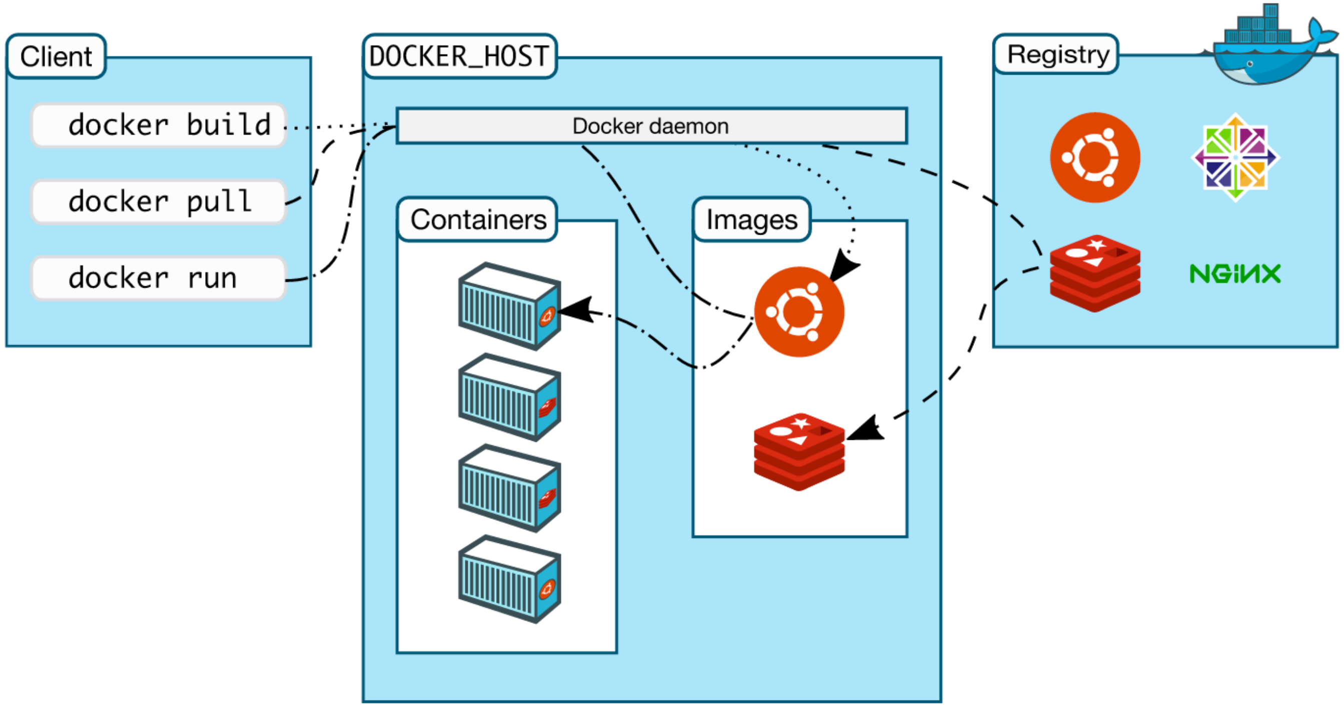
Cloud native is the software approach of building, deploying, and managing modern applications in cloud computing environments. Modern companies want to build highly scalable, flexible, and resilient applications that they can update quickly to meet customer demands.

# Docker made running containers easy !



## Build, ship and run

# Docker basics



# How are containers build ?

```
$ cat Dockerfile

FROM ubuntu:latest

RUN apt-get update && apt-get install -y openssl
RUN apt-get -y install ca-certificates

USER xxradar

WORKDIR /scripts

COPY tlssan_scan.sh tlssan_scan.sh

ENTRYPOINT ["/scripts/tlssan_scan.sh"]
```

# How are containers build (2)?

```
$ cat Dockerfile

# Base Alpine Linux based image with OpenJDK JRE only
FROM openjdk:8-jre-alpine

# copy application WAR (with libraries inside)
COPY target/spring-boot-*.war /app.war

# specify default command
CMD ["/usr/bin/java", "-jar", "-Dspring.profiles.active=test", "/app.war"]
```

# Image vulnerabilities

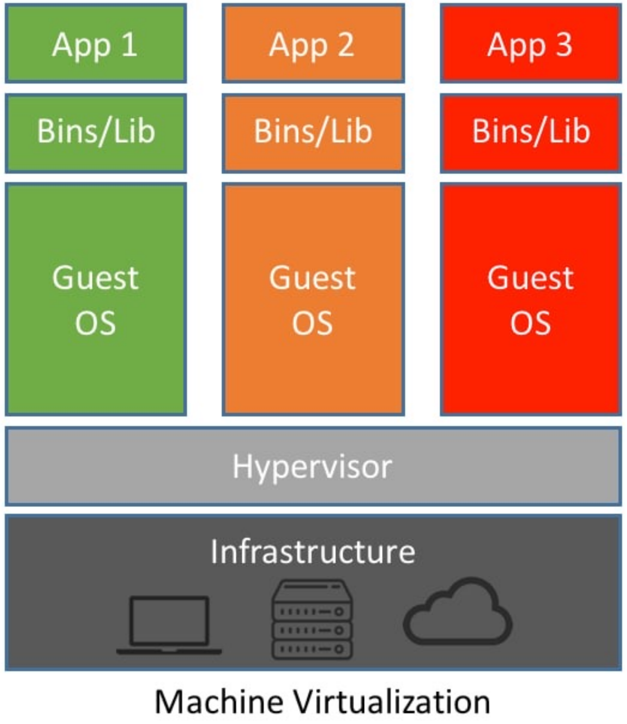
```
$ trivy image openjdk:8-jre-alpine | grep  -i total
Total: 216 (UNKNOWN: 0, LOW: 106, MEDIUM: 79, HIGH: 27, CRITICAL: 4)

$ trivy image openjdk:11 | grep -i total
Total: 389 (UNKNOWN: 0, LOW: 146, MEDIUM: 98, HIGH: 118, CRITICAL: 27)
```

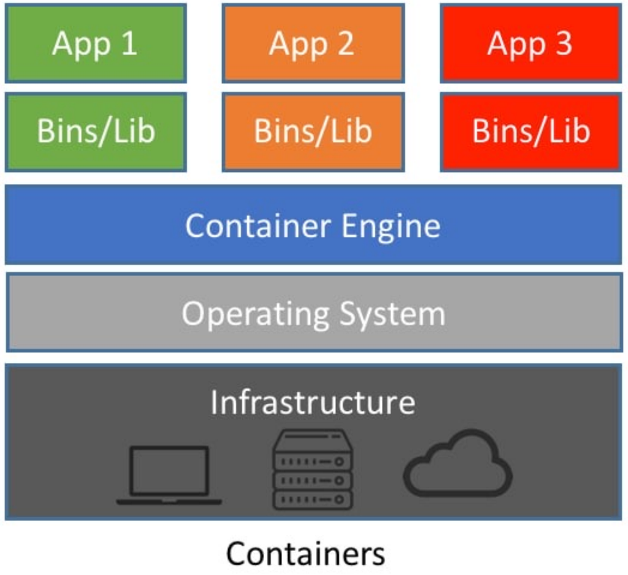
# D3m0 0#01

Building and running a container

# Containers vs. VM



Machine Virtualization

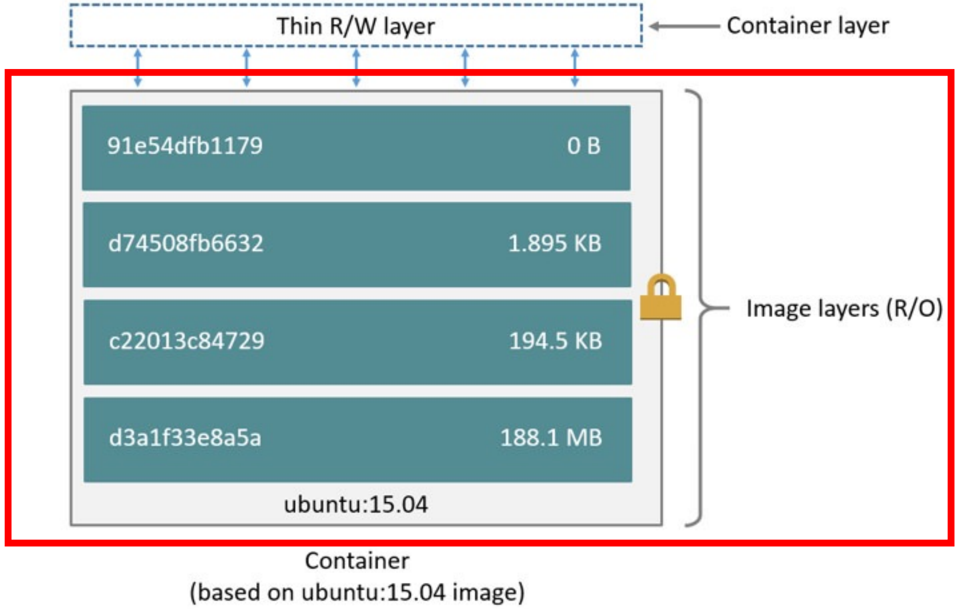


Containers

# Container runtimes

- The container runtime is the low-level component that creates and runs containers.
  - Containerd
  - CRI-O
  - Docker Engine
  - Mirantis Container Runtime
  - Podman
  - …
- Not all runtimes can be used in K8S

# Container Image (OCI specification)



```
/var/lib/docker
/var/lib/docker/aufs/diff/1b06661d...57x30604ee2b/app
/var/lib/docker/overlay2/4ca4af…0aa38d941a045fdb7d/diff/tmp
```

# What makes containers a container?

- Linux namespaces
- Control groups
- Linux capabilities

# Linux namespaces

- Control group
  - isolates the root directory
- IPC
  - isolates inter process communication
- Network
  - isolates the network stack
- Mount
  - isolates mount points
- Process ID (PID)
  - isolates process IDs
- User ID
  - isolates User and Group IDs
- UTS
  - isolates hostnames and domain names
- Time

# Linux Capabilities

- Two categories of processes
  - privileged
    - bypass all kernel permission checks
    - effective user ID is 0, referred to as superuser or root
  - unprivileged
    - subject to full permission checking based on the process’s credentials

- Linux divides the privileges traditionally associated with superuser into distinct units, known as *capabilities*, which can be independently enabled and disabled.

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

# Capabilities allowed by default

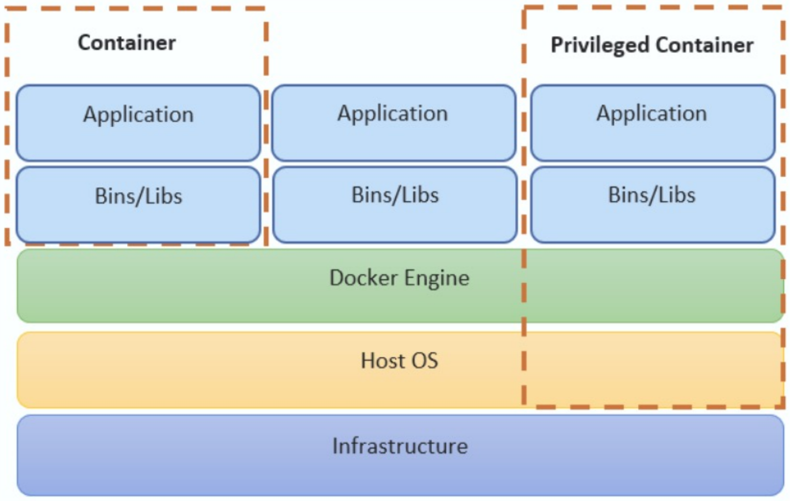
| Capability Key | Capability |
|---|---|
| AUDIT_WRITE | Write records to kernel auditing log. |
| CHOWN | Make arbitrary changes to file UIDs and GIDs (see chown(2)). |
| DAC_OVERRIDE | Bypass file read, write, and execute permission checks. |
| FOWNER | Bypass permission checks on operations that normally require the file system UID of the process to match the UID of the file. |
| FSETID | Don’t clear set-user-ID and set-group-ID permission bits when a file is modified. |
| KILL | Bypass permission checks for sending signals. |
| MKNOD | Create special files using mknod(2). |
| NET_BIND_SERVICE | Bind a socket to internet domain privileged ports (port numbers less than 1024). |
| NET_RAW | Use RAW and PACKET sockets. |
| SETFCAP | Set file capabilities. |
| SETGID | Make arbitrary manipulations of process GIDs and supplementary GID list. |
| SETPCAP | Modify process capabilities. |
| SETUID | Make arbitrary manipulations of process UIDs. |
| SYS_CHROOT | Use chroot(2), change root directory. |

# Capabilities not granted by default

| Capability Key | Capability Description |
|---|---|
| AUDIT_CONTROL | Enable and disable kernel auditing; change auditing filter rules; retrieve auditing status and filtering rules. |
| AUDIT_READ | Allow reading the audit log via multicast netlink socket. |
| BLOCK_SUSPEND | Allow preventing system suspends. |
| BPF | Allow creating BPF maps, loading BPF Type Format (BTF) data, retrieve JITed code of BPF programs, and more. |
| CHECKPOINT_RESTORE | Allow checkpoint/restore related operations. Introduced in kernel 5.9. |
| DAC_READ_SEARCH | Bypass file read permission checks and directory read and execute permission checks. |
| IPC_LOCK | Lock memory (mlock(2), mlockall(2), mmap(2), shmctl(2)). |
| IPC_OWNER | Bypass permission checks for operations on System V IPC objects. |
| LEASE | Establish leases on arbitrary files (see fcntl(2)). |
| LINUX_IMMUTABLE | Set the FS_APPEND_FL and FS_IMMUTABLE_FL i-node flags. |
| MAC_ADMIN | Allow MAC configuration or state changes. Implemented for the Smack LSM. |
| MAC_OVERRIDE | Override Mandatory Access Control (MAC). Implemented for the Smack Linux Security Module (LSM). |
| NET_ADMIN | Perform various network-related operations. |
| NET_BROADCAST | Make socket broadcasts and listen to multicasts. |
| PERFMON | Allow system performance and observability privileged operations using perf_events, i915_perf and other kernel subsystems |
| SYS_ADMIN | Perform a range of system administration operations. |
| SYS_BOOT | Use reboot(2) and kexec_load(2), reboot and load a new kernel for later execution. |
| SYS_MODULE | Load and unload kernel modules. |
| SYS_NICE | Raise process nice value (nice(2), setpriority(2)) and change the nice value for arbitrary processes. |
| SYS_PACCT | Use acct(2), switch process accounting on or off. |
| SYS_PTRACE | Trace arbitrary processes using ptrace(2). |
| SYS_RAWIO | Perform I/O port operations (iopl(2) and ioperm(2)). |
| SYS_RESOURCE | Override resource Limits. |
| SYS_TIME | Set system clock (settimeofday(2), stime(2), adjtimex(2)); set real-time (hardware) clock. |
| SYS_TTY_CONFIG | Use vhangup(2); employ various privileged ioctl(2) operations on virtual terminals. |
| SYSLOG | Perform privileged syslog(2) operations. |
| WAKE_ALARM | Trigger something that will wake up the system. |

# Privileged containers

- The --privileged flag gives all capabilities to the container, and it also lifts all the limitations enforced by the device cgroup controller



# Uncommon ?

- DIND – docker in docker
  - https://hub.docker.com/_/docker
- Tracee
  - https://github.com/aquasecurity/tracee
- Portainer
  - https://docs.portainer.io/v/ce-2.11/start/install/server/docker/linux
- Traefik
  - https://hub.docker.com/_/traefik

# D3m0 0#02

Privileged container – stealing secrets

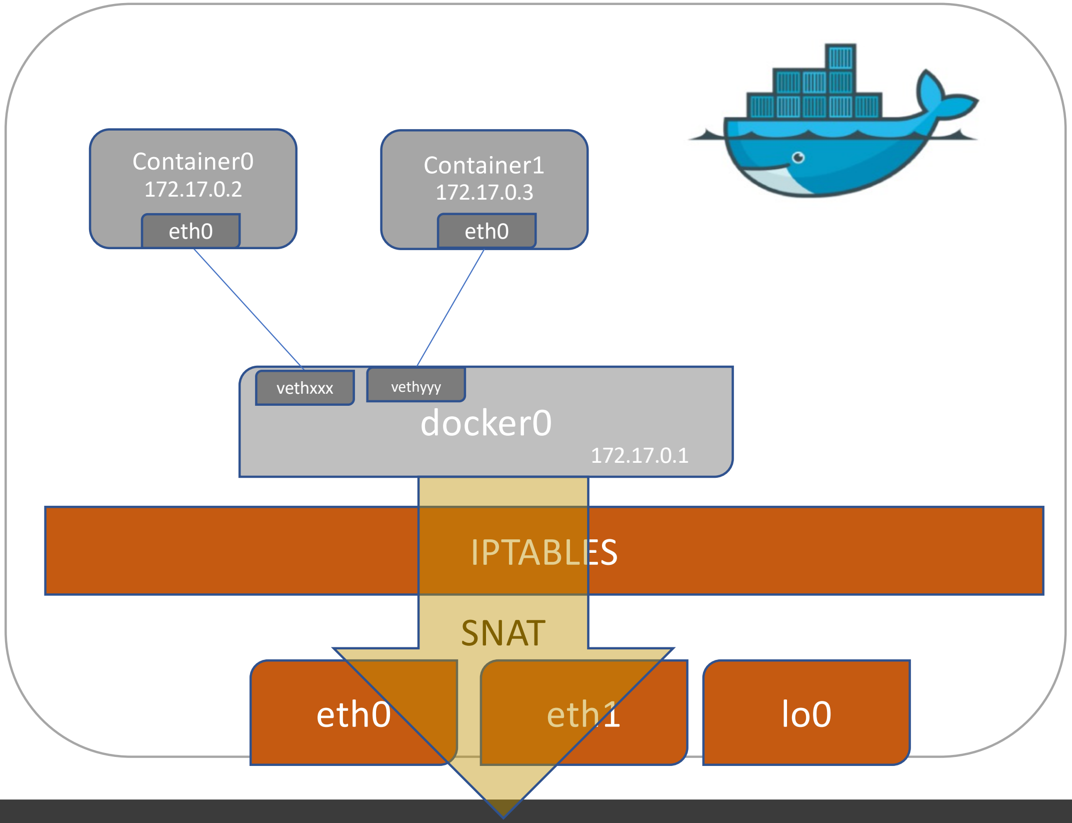
# Mounting volumes

- Volumes can be mounted in pods
  - Persistent storage

- Don’t mount critical paths
  - docker socket
  - /
  - ... (ex. log directories)

[https://github.com/xxradar/a_hackers_view/blob/master/examples/gaining_root/readme.md](https://github.com/xxradar/a_hackers_view/blob/master/examples/gaining_root/readme.md)

# Default Bridge Networking



# Docker networking

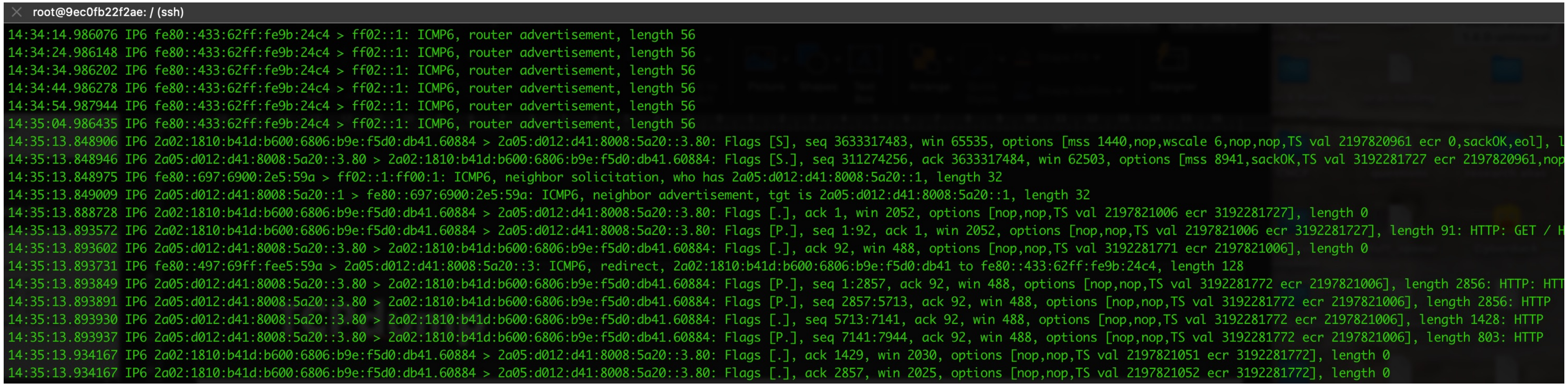
- Default bridge
- Non-default bridge
- MACVLAN
- IPVLAN
- --net=host
- --net=container:id
- Overlay (swarm)

```
docker network create --ipv6 -d ipvlan \
    -o parent=ens5 \
    --subnet 2a05:d012:d41:8008:5a20::/80 \
    --ip-range 2a05:d012:d41:8008:5a20::/96 ip6vlan
```

Tip: https://xxradar.medium.com/docker-pentester-series-1-macvlan-be4bca3062f2

# Troubleshooting w/ TCPdump

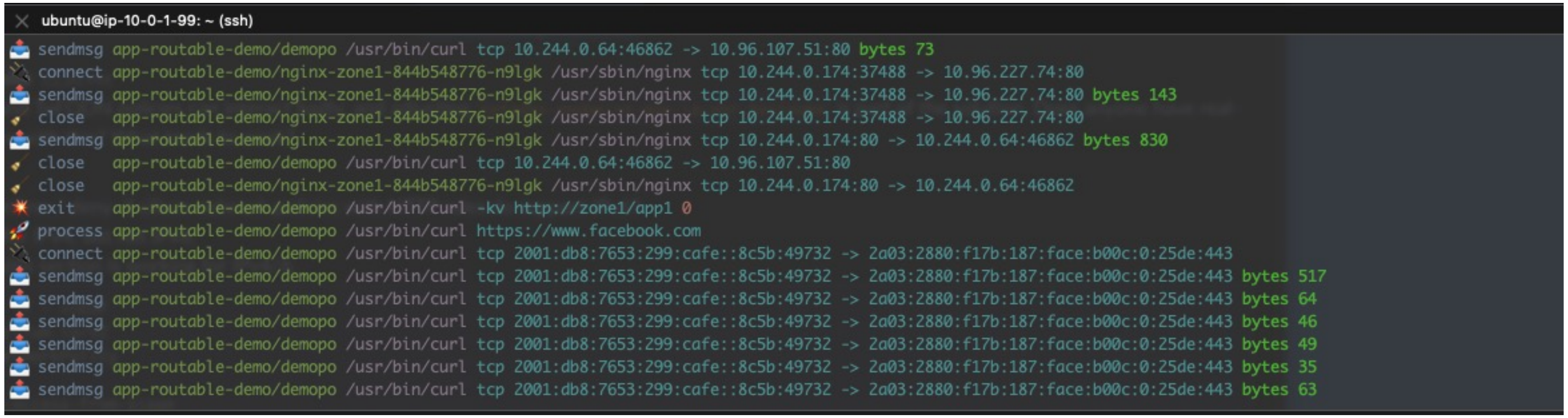
```
docker run -it --net=container:www3 xxradar/hackon tcpdump -n
```



```
14:34:14.986076 IP6 fe80::433:62ff:fe9b:24c4 > ff02::1: ICMP6, router advertisement, length 56
14:34:24.986148 IP6 fe80::433:62ff:fe9b:24c4 > ff02::1: ICMP6, router advertisement, length 56
14:34:34.986202 IP6 fe80::433:62ff:fe9b:24c4 > ff02::1: ICMP6, router advertisement, length 56
14:34:44.986278 IP6 fe80::433:62ff:fe9b:24c4 > ff02::1: ICMP6, router advertisement, length 56
14:34:54.987944 IP6 fe80::433:62ff:fe9b:24c4 > ff02::1: ICMP6, router advertisement, length 56
14:35:04.986435 IP6 fe80::433:62ff:fe9b:24c4 > ff02::1: ICMP6, router advertisement, length 56
14:35:13.848906 IP6 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884 > 2a05:d012:d41:8008:5a20::3.80: Flags [S], seq 3633317483, win 65535, options [mss 1440,nop,wscale 6,nop,nop,TS val 2197820961 ecr 0,sackOK,eol], l
14:35:13.848946 IP6 2a05:d012:d41:8008:5a20::3.80 > 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884: Flags [S.], seq 311274256, ack 3633317484, win 62503, options [mss 8941,sackOK,TS val 3192281727 ecr 2197820961,nop
14:35:13.848975 IP6 fe80::697:6900:2e5:59a > ff02::1:ff00:1: ICMP6, neighbor solicitation, who has 2a05:d012:d41:8008:5a20::1, length 32
14:35:13.849009 IP6 2a05:d012:d41:8008:5a20::1 > fe80::697:6900:2e5:59a: ICMP6, neighbor advertisement, tgt is 2a05:d012:d41:8008:5a20::1, length 32
14:35:13.888728 IP6 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884 > 2a05:d012:d41:8008:5a20::3.80: Flags [.], ack 1, win 2052, options [nop,nop,TS val 2197821006 ecr 3192281727], length 0
14:35:13.893572 IP6 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884 > 2a05:d012:d41:8008:5a20::3.80: Flags [P.], seq 1:92, ack 1, win 2052, options [nop,nop,TS val 2197821006 ecr 3192281727], length 91: HTTP: GET / H
14:35:13.893602 IP6 2a05:d012:d41:8008:5a20::3.80 > 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884: Flags [.], ack 92, win 488, options [nop,nop,TS val 3192281771 ecr 2197821006], length 0
14:35:13.893731 IP6 fe80::497:69ff:fee5:59a > 2a05:d012:d41:8008:5a20::3: ICMP6, redirect, 2a02:1810:b41d:b600:6806:b9e:f5d0:db41 to fe80::433:62ff:fe9b:24c4, length 128
14:35:13.893849 IP6 2a05:d012:d41:8008:5a20::3.80 > 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884: Flags [P.], seq 1:2857, ack 92, win 488, options [nop,nop,TS val 3192281772 ecr 2197821006], length 2856: HTTP: HTT
14:35:13.893891 IP6 2a05:d012:d41:8008:5a20::3.80 > 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884: Flags [P.], seq 2857:5713, ack 92, win 488, options [nop,nop,TS val 3192281772 ecr 2197821006], length 2856: HTTP
14:35:13.893930 IP6 2a05:d012:d41:8008:5a20::3.80 > 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884: Flags [.], seq 5713:7141, ack 92, win 488, options [nop,nop,TS val 3192281772 ecr 2197821006], length 1428: HTTP
14:35:13.893937 IP6 2a05:d012:d41:8008:5a20::3.80 > 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884: Flags [P.], seq 7141:7944, ack 92, win 488, options [nop,nop,TS val 3192281772 ecr 2197821006], length 803: HTTP
14:35:13.934167 IP6 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884 > 2a05:d012:d41:8008:5a20::3.80: Flags [.], ack 1429, win 2030, options [nop,nop,TS val 2197821051 ecr 3192281772], length 0
14:35:13.934167 IP6 2a02:1810:b41d:b600:6806:b9e:f5d0:db41.60884 > 2a05:d012:d41:8008:5a20::3.80: Flags [.], ack 2857, win 2025, options [nop,nop,TS val 2197821052 ecr 3192281772], length 0
```

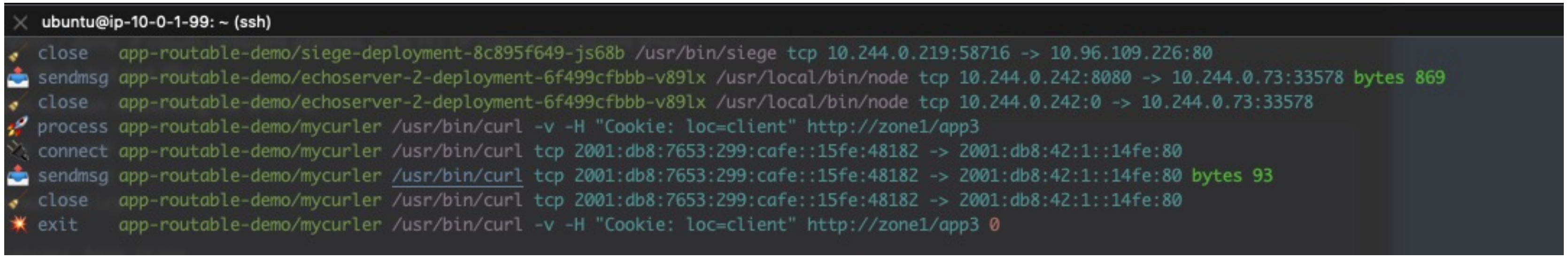
# Runtime security and monitoring

- Tetragon



ubuntu@ip-10-0-1-99: ~ (ssh)

```
📤 sendmsg app-routable-demo/demopo /usr/bin/curl tcp 10.244.0.64:46862 -> 10.96.107.51:80 bytes 73
🔌 connect app-routable-demo/nginx-zone1-844b548776-n9lgk /usr/sbin/nginx tcp 10.244.0.174:37488 -> 10.96.227.74:80
📤 sendmsg app-routable-demo/nginx-zone1-844b548776-n9lgk /usr/sbin/nginx tcp 10.244.0.174:37488 -> 10.96.227.74:80 bytes 143
🧹 close   app-routable-demo/nginx-zone1-844b548776-n9lgk /usr/sbin/nginx tcp 10.244.0.174:37488 -> 10.96.227.74:80
📤 sendmsg app-routable-demo/nginx-zone1-844b548776-n9lgk /usr/sbin/nginx tcp 10.244.0.174:80 -> 10.244.0.64:46862 bytes 830
🧹 close   app-routable-demo/demopo /usr/bin/curl tcp 10.244.0.64:46862 -> 10.96.107.51:80
🧹 close   app-routable-demo/nginx-zone1-844b548776-n9lgk /usr/sbin/nginx tcp 10.244.0.174:80 -> 10.244.0.64:46862
💥 exit    app-routable-demo/demopo /usr/bin/curl -kv http://zone1/app1 0
🚀 process app-routable-demo/demopo /usr/bin/curl https://www.facebook.com
🔌 connect app-routable-demo/demopo /usr/bin/curl tcp 2001:db8:7653:299:cafe::8c5b:49732 -> 2a03:2880:f17b:187:face:b00c:0:25de:443
📤 sendmsg app-routable-demo/demopo /usr/bin/curl tcp 2001:db8:7653:299:cafe::8c5b:49732 -> 2a03:2880:f17b:187:face:b00c:0:25de:443 bytes 517
📤 sendmsg app-routable-demo/demopo /usr/bin/curl tcp 2001:db8:7653:299:cafe::8c5b:49732 -> 2a03:2880:f17b:187:face:b00c:0:25de:443 bytes 64
📤 sendmsg app-routable-demo/demopo /usr/bin/curl tcp 2001:db8:7653:299:cafe::8c5b:49732 -> 2a03:2880:f17b:187:face:b00c:0:25de:443 bytes 46
📤 sendmsg app-routable-demo/demopo /usr/bin/curl tcp 2001:db8:7653:299:cafe::8c5b:49732 -> 2a03:2880:f17b:187:face:b00c:0:25de:443 bytes 49
📤 sendmsg app-routable-demo/demopo /usr/bin/curl tcp 2001:db8:7653:299:cafe::8c5b:49732 -> 2a03:2880:f17b:187:face:b00c:0:25de:443 bytes 35
📤 sendmsg app-routable-demo/demopo /usr/bin/curl tcp 2001:db8:7653:299:cafe::8c5b:49732 -> 2a03:2880:f17b:187:face:b00c:0:25de:443 bytes 63
```



ubuntu@ip-10-0-1-99: ~ (ssh)

```
🧹 close   app-routable-demo/siege-deployment-8c895f649-js68b /usr/bin/siege tcp 10.244.0.219:58716 -> 10.96.109.226:80
📤 sendmsg app-routable-demo/echoserver-2-deployment-6f499cfbbb-v89lx /usr/local/bin/node tcp 10.244.0.242:8080 -> 10.244.0.73:33578 bytes 869
🧹 close   app-routable-demo/echoserver-2-deployment-6f499cfbbb-v89lx /usr/local/bin/node tcp 10.244.0.242:0 -> 10.244.0.73:33578
🚀 process app-routable-demo/mycurler /usr/bin/curl -v -H "Cookie: loc=client" http://zone1/app3
🔌 connect app-routable-demo/mycurler /usr/bin/curl tcp 2001:db8:7653:299:cafe::15fe:48182 -> 2001:db8:42:1::14fe:80
📤 sendmsg app-routable-demo/mycurler /usr/bin/curl tcp 2001:db8:7653:299:cafe::15fe:48182 -> 2001:db8:42:1::14fe:80 bytes 93
🧹 close   app-routable-demo/mycurler /usr/bin/curl tcp 2001:db8:7653:299:cafe::15fe:48182 -> 2001:db8:42:1::14fe:80
💥 exit    app-routable-demo/mycurler /usr/bin/curl -v -H "Cookie: loc=client" http://zone1/app3 0
```

# D3m0 0#03

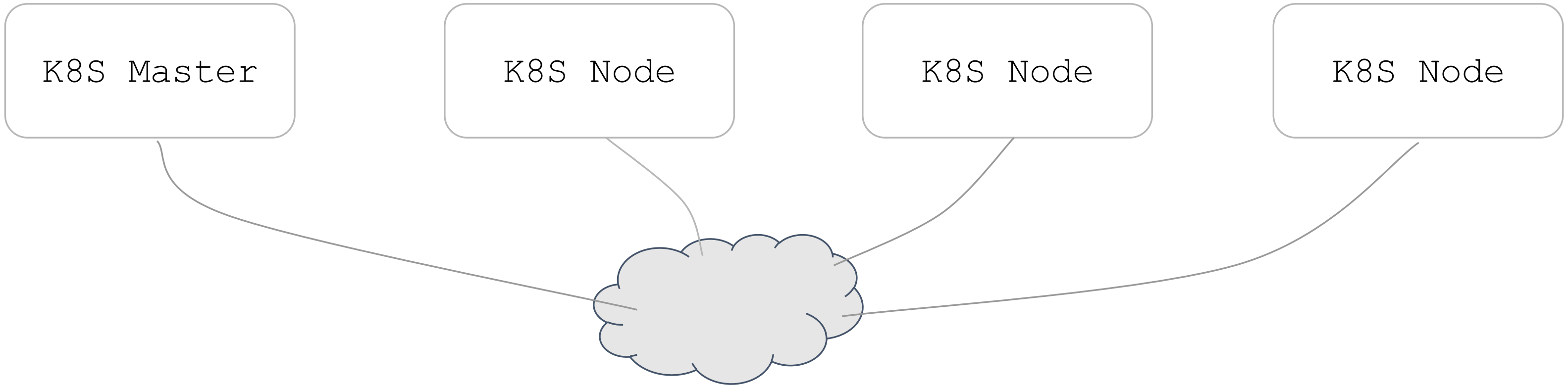
Insecure mounts

# D3m0 0#04

CICD

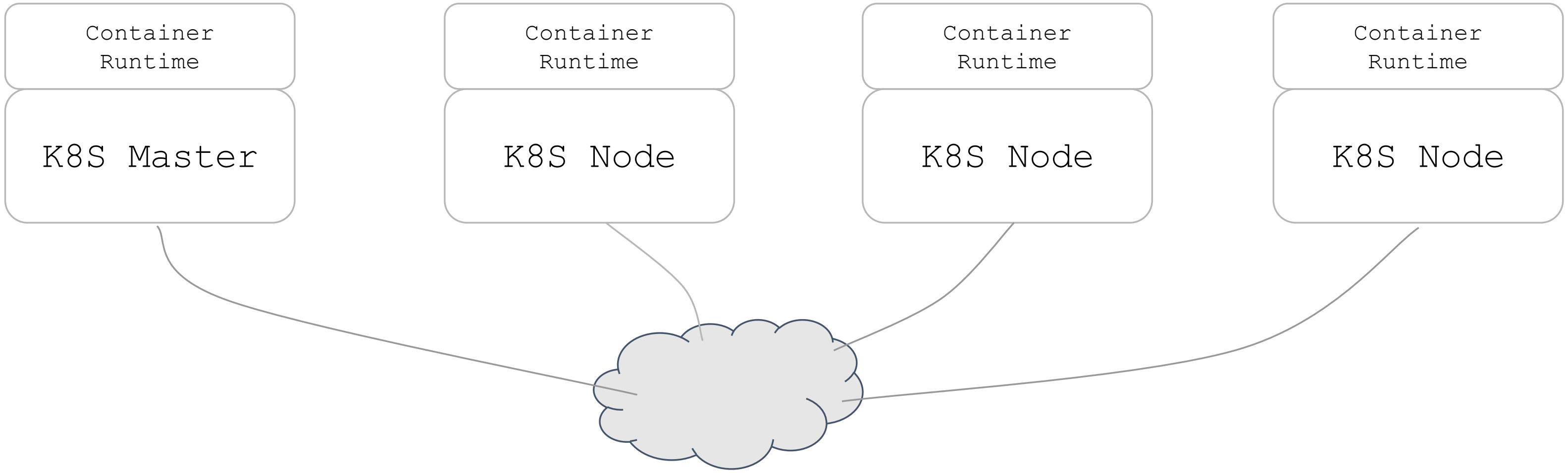
# Kubernetes - Nodes

- Hardware or VM
- Master node(s) & Worker nodes



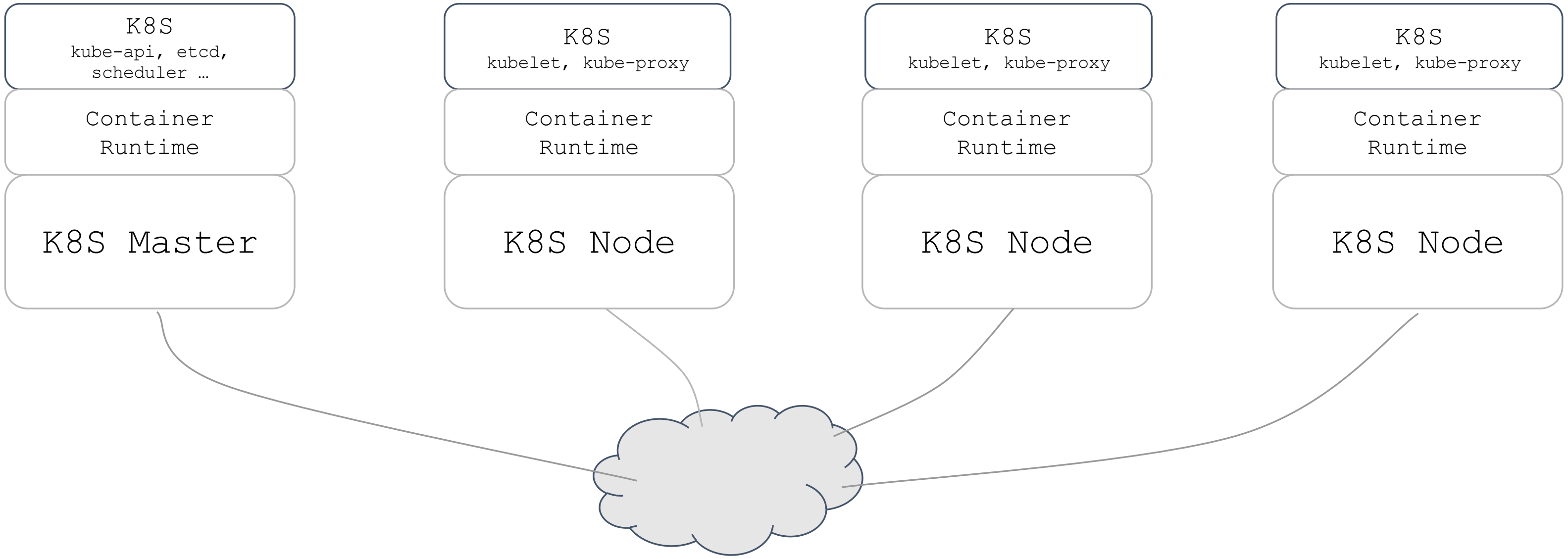
# Kubernetes – Container Runtime

- Container runtimes
  - CRI-O
  - Containerd
  - …



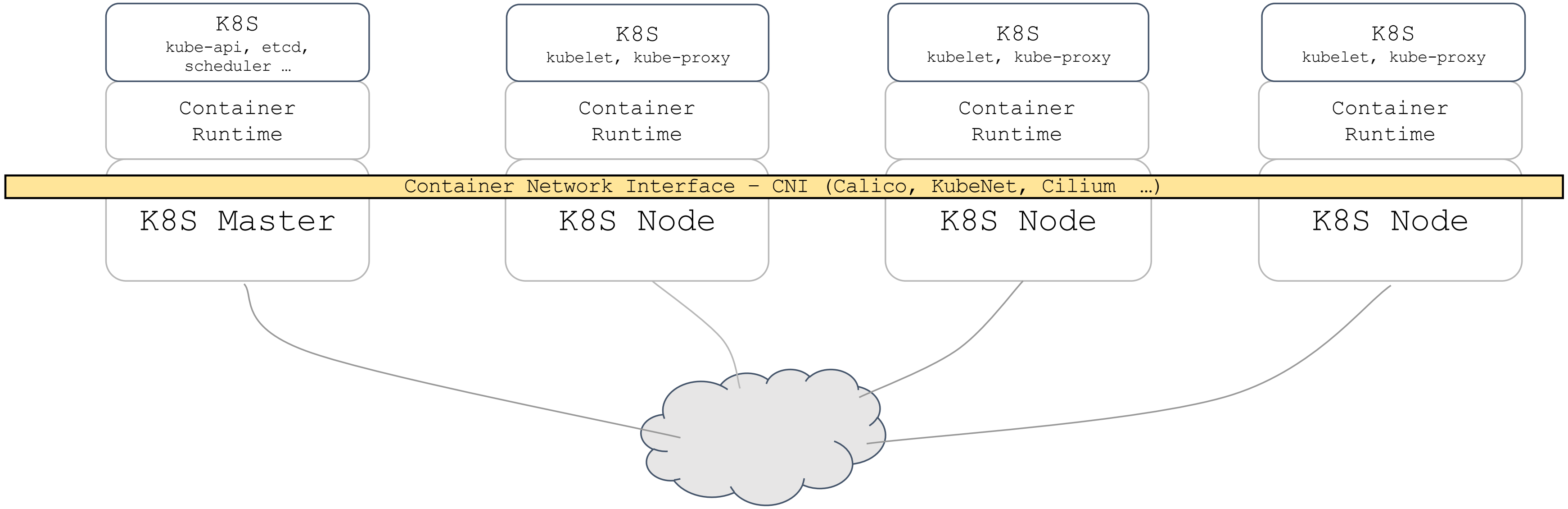
# Kubernetes – Control Plane

- K8S components are typically binaries or pods that communicate over the network using the host network IP address

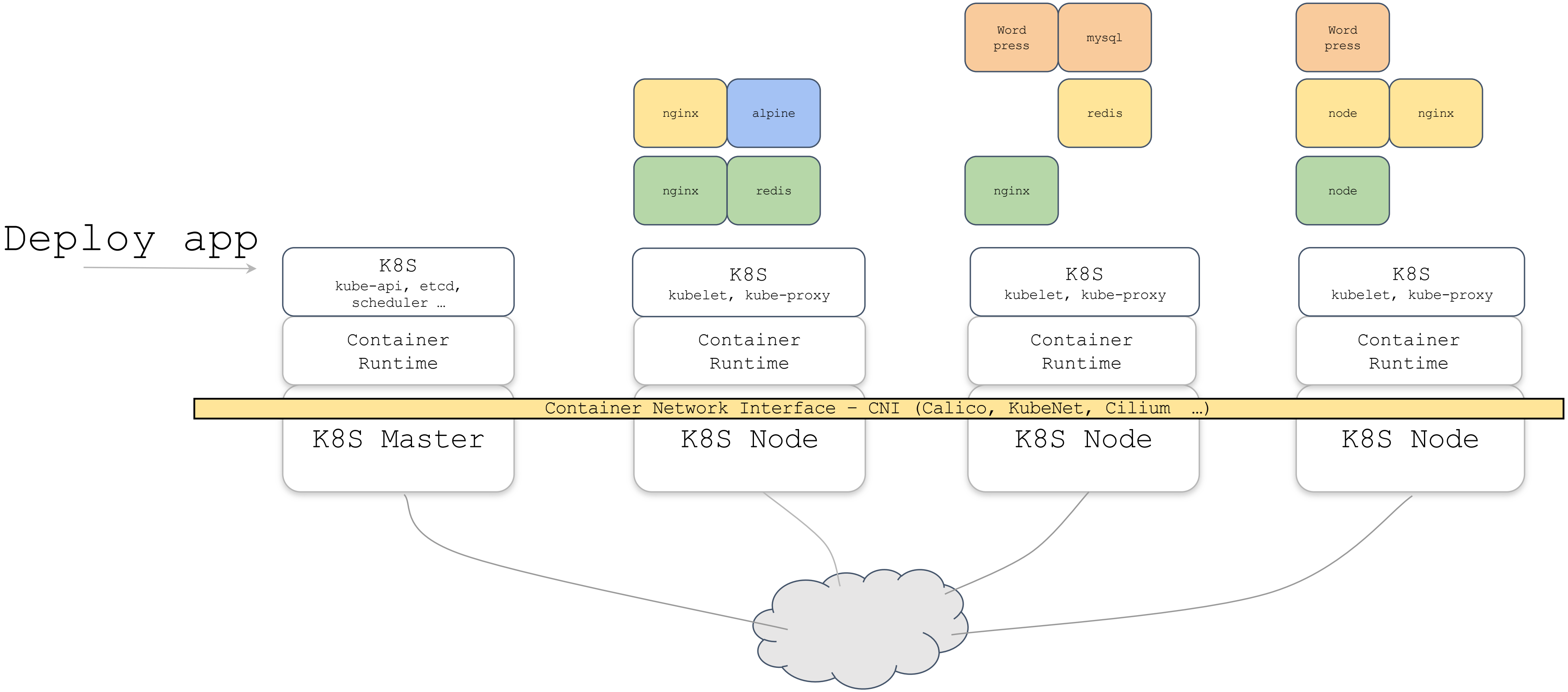


# CNI - Container Network Interface

- K8S worloads (ex. Pods) need to communicate using IP networking. The networking, IPAM, routing … is handled by the CNI (and not K8S)

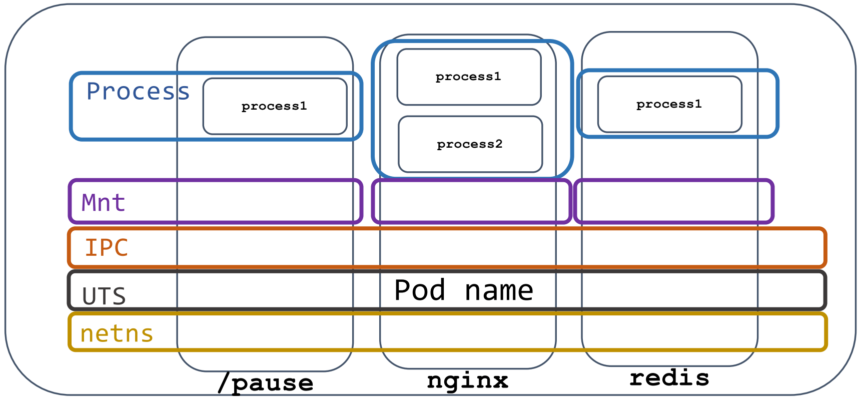


# Kubernetes – Basic principles

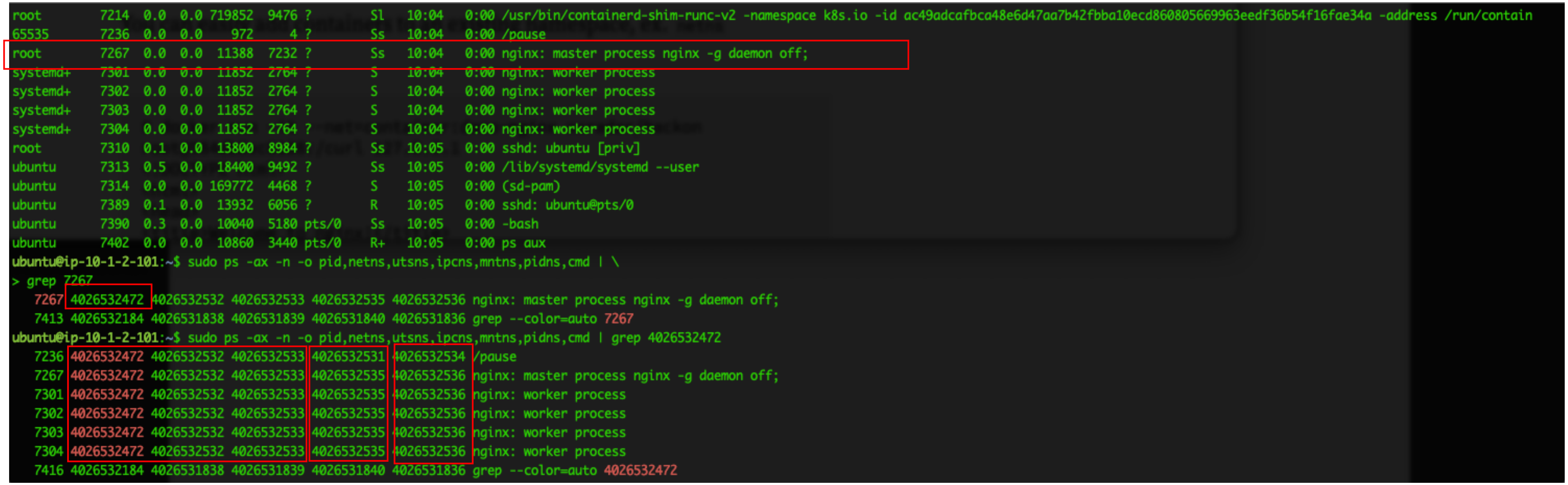


# What is a pod?

- a collection of one or more containers
- the smallest unit of a Kubernetes application

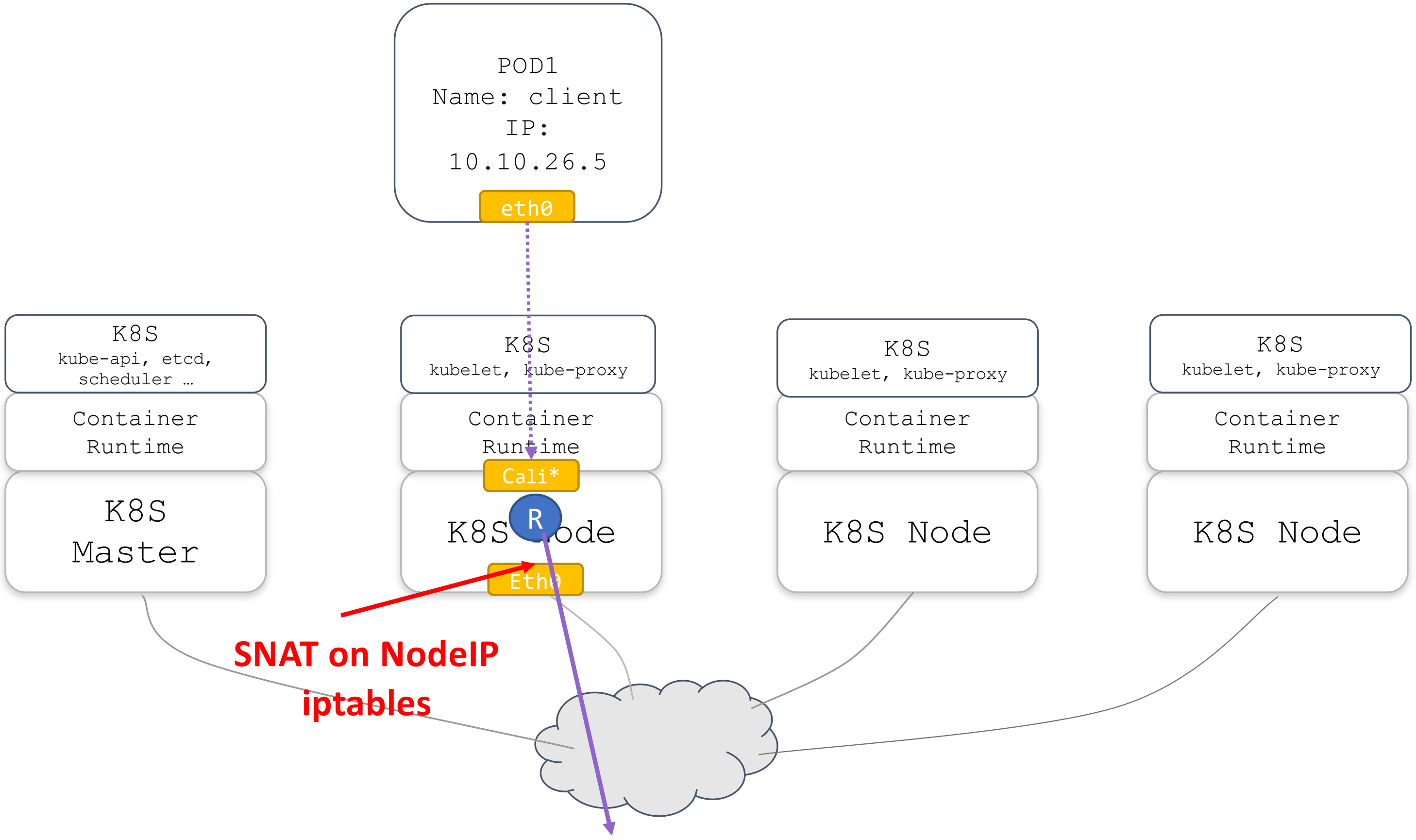


# Example



```
root        7214  0.0  0.0 719852  9476 ?        Sl   10:04   0:00 /usr/bin/containerd-shim-runc-v2 -namespace k8s.io -id ac49adcafbca48e6d47aa7b42fbba10ecd860805669963eedf36b54f16fae34a -address /run/contain
65535       7236  0.0  0.0    972     4 ?        Ss   10:04   0:00 /pause
root        7267  0.0  0.0  11388  7232 ?        Ss   10:04   0:00 nginx: master process nginx -g daemon off;
systemd+    7301  0.0  0.0  11852  2764 ?        S    10:04   0:00 nginx: worker process
systemd+    7302  0.0  0.0  11852  2764 ?        S    10:04   0:00 nginx: worker process
systemd+    7303  0.0  0.0  11852  2764 ?        S    10:04   0:00 nginx: worker process
systemd+    7304  0.0  0.0  11852  2764 ?        S    10:04   0:00 nginx: worker process
root        7310  0.1  0.0  13800  8984 ?        Ss   10:05   0:00 sshd: ubuntu [priv]
ubuntu      7313  0.5  0.0  18400  9492 ?        Ss   10:05   0:00 /lib/systemd/systemd --user
ubuntu      7314  0.0  0.0 169772  4468 ?        S    10:05   0:00 (sd-pam)
ubuntu      7389  0.1  0.0  13932  6056 ?        R    10:05   0:00 sshd: ubuntu@pts/0
ubuntu      7390  0.3  0.0  10040  5180 pts/0    Ss   10:05   0:00 -bash
ubuntu      7402  0.0  0.0  10860  3440 pts/0    R+   10:05   0:00 ps aux
ubuntu@ip-10-1-2-101:~$ sudo ps -ax -n -o pid,netns,utsns,ipcns,mntns,pidns,cmd | \
> grep 7267
   7267 4026532472 4026532532 4026532533 4026532535 4026532536 nginx: master process nginx -g daemon off;
   7413 4026532184 4026531838 4026531839 4026531840 4026531836 grep --color=auto 7267
ubuntu@ip-10-1-2-101:~$ sudo ps -ax -n -o pid,netns,utsns,ipcns,mntns,pidns,cmd | grep 4026532472
   7236 4026532472 4026532532 4026532533 4026532531 4026532534 /pause
   7267 4026532472 4026532532 4026532533 4026532535 4026532536 nginx: master process nginx -g daemon off;
   7301 4026532472 4026532532 4026532533 4026532535 4026532536 nginx: worker process
   7302 4026532472 4026532532 4026532533 4026532535 4026532536 nginx: worker process
   7303 4026532472 4026532532 4026532533 4026532535 4026532536 nginx: worker process
   7304 4026532472 4026532532 4026532533 4026532535 4026532536 nginx: worker process
   7416 4026532184 4026531838 4026531839 4026531840 4026531836 grep --color=auto 4026532472
```

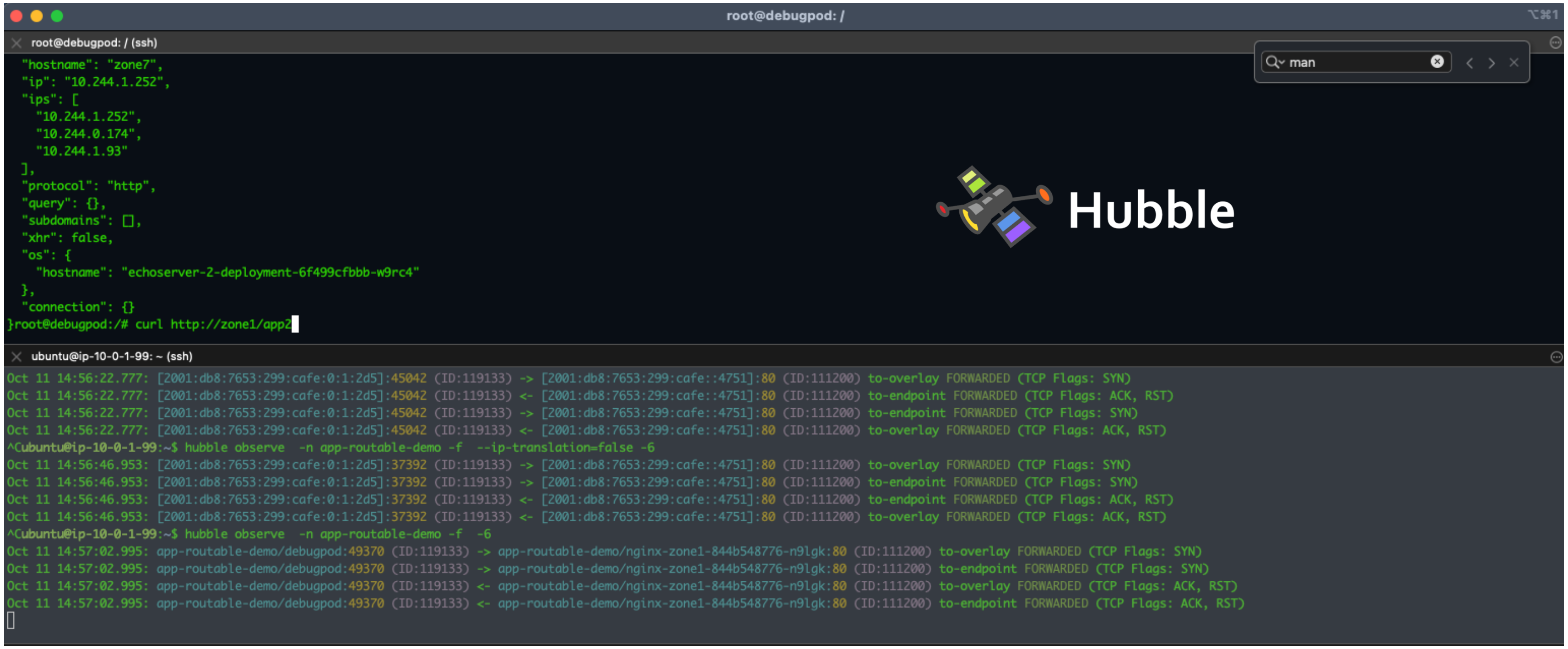
# Cluster egress (SNAT)



SNAT on NodeIP
iptables

# Observability and troubleshooting

- TCPdump
- EBPF

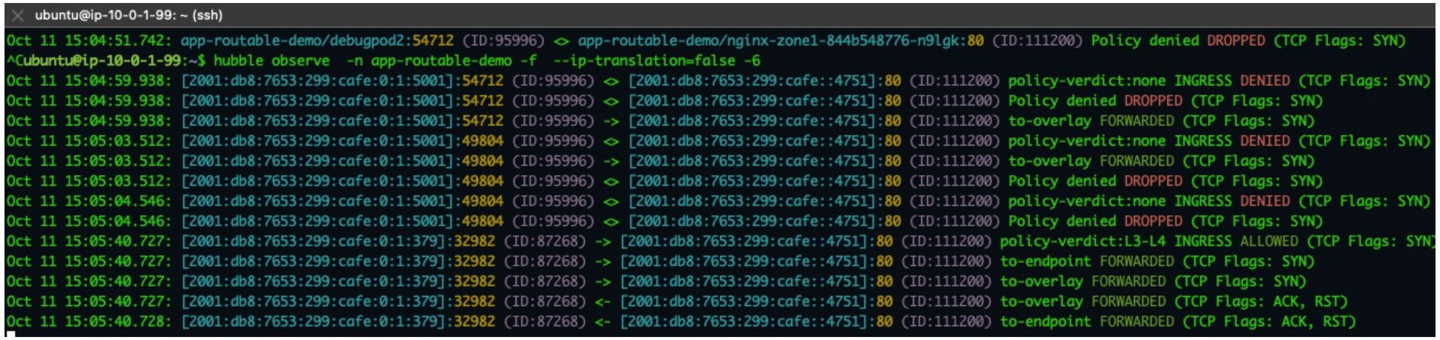


```
  "hostname": "zone7",
  "ip": "10.244.1.252",
  "ips": [
    "10.244.1.252",
    "10.244.0.174",
    "10.244.1.93"
  ],
  "protocol": "http",
  "query": {},
  "subdomains": [],
  "xhr": false,
  "os": {
    "hostname": "echoserver-2-deployment-6f499cfbbb-w9rc4"
  },
  "connection": {}
}root@debugpod:/# curl http://zone1/app2
```

```
Oct 11 14:56:22.777: [2001:db8:7653:299:cafe:0:1:2d5]:45042 (ID:119133) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: SYN)
Oct 11 14:56:22.777: [2001:db8:7653:299:cafe:0:1:2d5]:45042 (ID:119133) <- [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, RST)
Oct 11 14:56:22.777: [2001:db8:7653:299:cafe:0:1:2d5]:45042 (ID:119133) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: SYN)
Oct 11 14:56:22.777: [2001:db8:7653:299:cafe:0:1:2d5]:45042 (ID:119133) <- [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: ACK, RST)
^Cubuntu@ip-10-0-1-99:~$ hubble observe  -n app-routable-demo -f  --ip-translation=false -6
Oct 11 14:56:46.953: [2001:db8:7653:299:cafe:0:1:2d5]:37392 (ID:119133) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: SYN)
Oct 11 14:56:46.953: [2001:db8:7653:299:cafe:0:1:2d5]:37392 (ID:119133) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: SYN)
Oct 11 14:56:46.953: [2001:db8:7653:299:cafe:0:1:2d5]:37392 (ID:119133) <- [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, RST)
Oct 11 14:56:46.953: [2001:db8:7653:299:cafe:0:1:2d5]:37392 (ID:119133) <- [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: ACK, RST)
^Cubuntu@ip-10-0-1-99:~$ hubble observe  -n app-routable-demo -f  -6
Oct 11 14:57:02.995: app-routable-demo/debugpod:49370 (ID:119133) -> app-routable-demo/nginx-zone1-844b548776-n9lgk:80 (ID:111200) to-overlay FORWARDED (TCP Flags: SYN)
Oct 11 14:57:02.995: app-routable-demo/debugpod:49370 (ID:119133) -> app-routable-demo/nginx-zone1-844b548776-n9lgk:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: SYN)
Oct 11 14:57:02.995: app-routable-demo/debugpod:49370 (ID:119133) <- app-routable-demo/nginx-zone1-844b548776-n9lgk:80 (ID:111200) to-overlay FORWARDED (TCP Flags: ACK, RST)
Oct 11 14:57:02.995: app-routable-demo/debugpod:49370 (ID:119133) <- app-routable-demo/nginx-zone1-844b548776-n9lgk:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, RST)
```

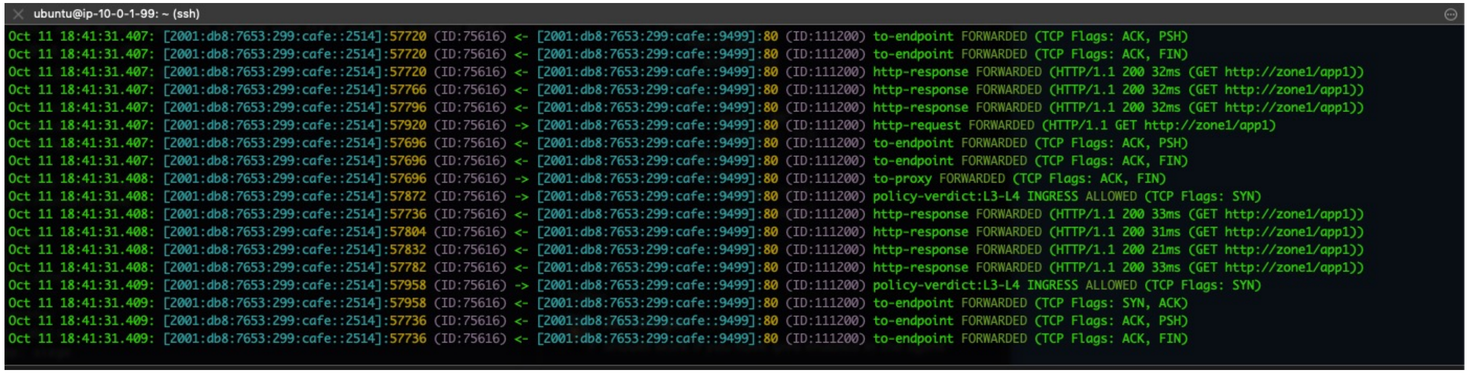
# Network Security Policies

```
kubectl apply -f - <<EOF
apiVersion: cilium.io/v2
kind: CiliumNetworkPolicy
metadata:
  name: allow-access-from-siege
  namespace: app-routable-demo
spec:
  endpointSelector:
    matchLabels:
      app: nginx-zone1
  ingress:
    - fromEndpoints:
        - matchLabels:
            app: siege
      toPorts:
        - ports:
            - port: "80"
              protocol: TCP
EOF
```



ubuntu@ip-10-0-1-99: ~ (ssh)

```
Oct 11 15:04:51.742: app-routable-demo/debugpod2:54712 (ID:95996) <> app-routable-demo/nginx-zone1-844b548776-n9lgk:80 (ID:111200) Policy denied DROPPED (TCP Flags: SYN)
^Cubuntu@ip-10-0-1-99:~$ hubble observe  -n app-routable-demo -f  --ip-translation=false -6
Oct 11 15:04:59.938: [2001:db8:7653:299:cafe:0:1:5001]:54712 (ID:95996) <> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) policy-verdict:none INGRESS DENIED (TCP Flags: SYN)
Oct 11 15:04:59.938: [2001:db8:7653:299:cafe:0:1:5001]:54712 (ID:95996) <> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) Policy denied DROPPED (TCP Flags: SYN)
Oct 11 15:04:59.938: [2001:db8:7653:299:cafe:0:1:5001]:54712 (ID:95996) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: SYN)
Oct 11 15:05:03.512: [2001:db8:7653:299:cafe:0:1:5001]:49804 (ID:95996) <> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) policy-verdict:none INGRESS DENIED (TCP Flags: SYN)
Oct 11 15:05:03.512: [2001:db8:7653:299:cafe:0:1:5001]:49804 (ID:95996) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: SYN)
Oct 11 15:05:03.512: [2001:db8:7653:299:cafe:0:1:5001]:49804 (ID:95996) <> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) Policy denied DROPPED (TCP Flags: SYN)
Oct 11 15:05:04.546: [2001:db8:7653:299:cafe:0:1:5001]:49804 (ID:95996) <> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) policy-verdict:none INGRESS DENIED (TCP Flags: SYN)
Oct 11 15:05:04.546: [2001:db8:7653:299:cafe:0:1:5001]:49804 (ID:95996) <> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) Policy denied DROPPED (TCP Flags: SYN)
Oct 11 15:05:40.727: [2001:db8:7653:299:cafe:0:1:379]:32982 (ID:87268) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) policy-verdict:L3-L4 INGRESS ALLOWED (TCP Flags: SYN)
Oct 11 15:05:40.727: [2001:db8:7653:299:cafe:0:1:379]:32982 (ID:87268) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: SYN)
Oct 11 15:05:40.727: [2001:db8:7653:299:cafe:0:1:379]:32982 (ID:87268) -> [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: SYN)
Oct 11 15:05:40.727: [2001:db8:7653:299:cafe:0:1:379]:32982 (ID:87268) <- [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-overlay FORWARDED (TCP Flags: ACK, RST)
Oct 11 15:05:40.728: [2001:db8:7653:299:cafe:0:1:379]:32982 (ID:87268) <- [2001:db8:7653:299:cafe::4751]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, RST)
```



ubuntu@ip-10-0-1-99: ~ (ssh)

```
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57720 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, PSH)
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57720 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, FIN)
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57720 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-response FORWARDED (HTTP/1.1 200 32ms (GET http://zone1/app1))
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57766 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-response FORWARDED (HTTP/1.1 200 32ms (GET http://zone1/app1))
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57796 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-response FORWARDED (HTTP/1.1 200 32ms (GET http://zone1/app1))
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57920 (ID:75616) -> [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-request FORWARDED (HTTP/1.1 GET http://zone1/app1)
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57696 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, PSH)
Oct 11 18:41:31.407: [2001:db8:7653:299:cafe::2514]:57696 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, FIN)
Oct 11 18:41:31.408: [2001:db8:7653:299:cafe::2514]:57696 (ID:75616) -> [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-proxy FORWARDED (TCP Flags: ACK, FIN)
Oct 11 18:41:31.408: [2001:db8:7653:299:cafe::2514]:57872 (ID:75616) -> [2001:db8:7653:299:cafe::9499]:80 (ID:111200) policy-verdict:L3-L4 INGRESS ALLOWED (TCP Flags: SYN)
Oct 11 18:41:31.408: [2001:db8:7653:299:cafe::2514]:57736 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-response FORWARDED (HTTP/1.1 200 33ms (GET http://zone1/app1))
Oct 11 18:41:31.408: [2001:db8:7653:299:cafe::2514]:57804 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-response FORWARDED (HTTP/1.1 200 31ms (GET http://zone1/app1))
Oct 11 18:41:31.408: [2001:db8:7653:299:cafe::2514]:57832 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-response FORWARDED (HTTP/1.1 200 21ms (GET http://zone1/app1))
Oct 11 18:41:31.408: [2001:db8:7653:299:cafe::2514]:57782 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) http-response FORWARDED (HTTP/1.1 200 33ms (GET http://zone1/app1))
Oct 11 18:41:31.409: [2001:db8:7653:299:cafe::2514]:57958 (ID:75616) -> [2001:db8:7653:299:cafe::9499]:80 (ID:111200) policy-verdict:L3-L4 INGRESS ALLOWED (TCP Flags: SYN)
Oct 11 18:41:31.409: [2001:db8:7653:299:cafe::2514]:57958 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: SYN, ACK)
Oct 11 18:41:31.409: [2001:db8:7653:299:cafe::2514]:57736 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, PSH)
Oct 11 18:41:31.409: [2001:db8:7653:299:cafe::2514]:57736 (ID:75616) <- [2001:db8:7653:299:cafe::9499]:80 (ID:111200) to-endpoint FORWARDED (TCP Flags: ACK, FIN)
```

# D3m0 0#05

K8S backdooring

```
docker run -it --privileged --pid=host debian nsenter -t 1 -m -u -i sh
```

# Questions ?

https://meetups.kubiosec.tech