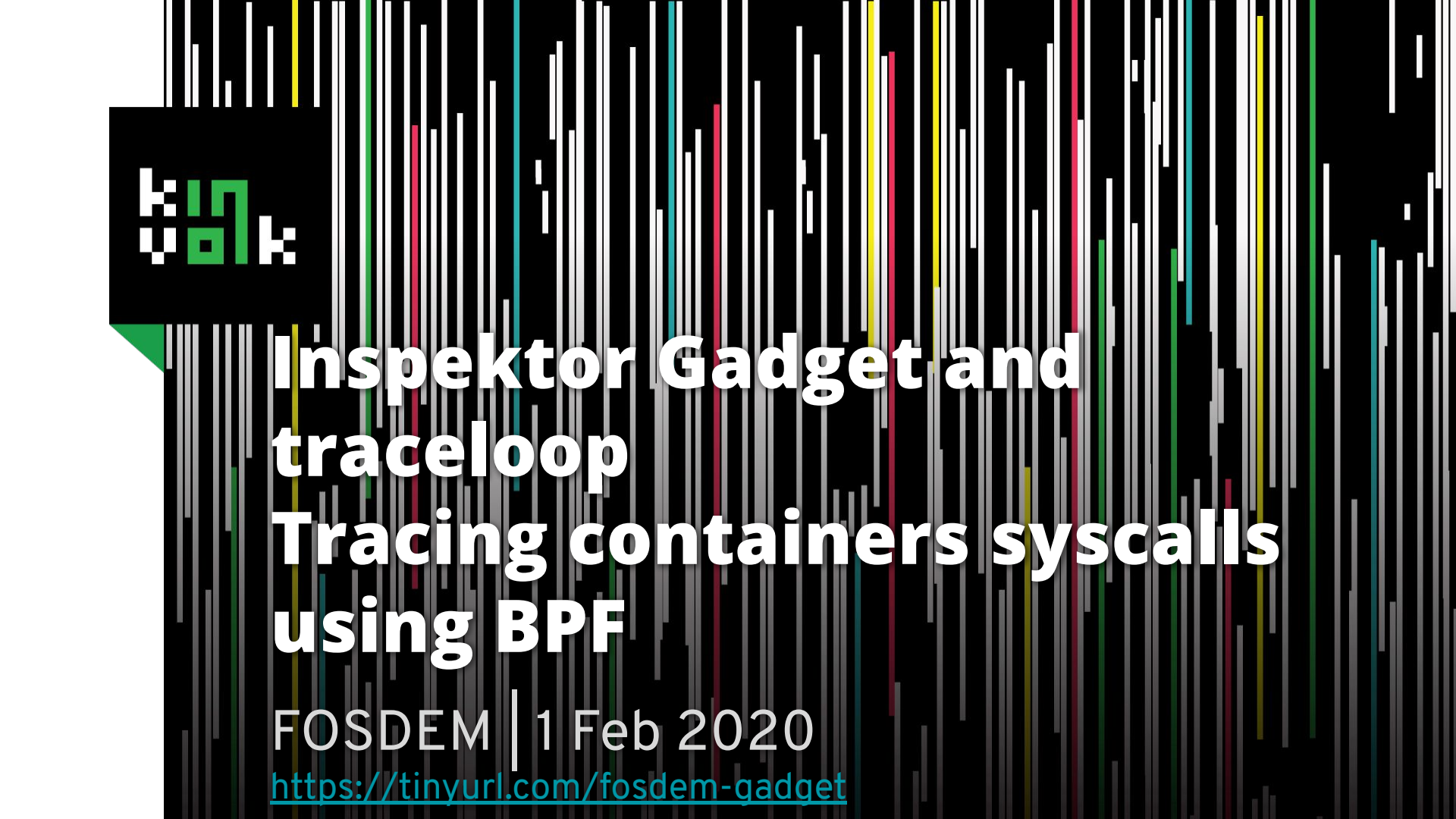


Inspektor Gadget and traceloop
Tracing containers syscalls using BPF

FOSDEM | 1 Feb 2020

https://tinyurl.com/fosdem-gadget

Hi, I'm Alban

Alban Crequy
CTO, Kinvolk

Github: **alban**
Twitter: **albcr**
Email: **alban@kinvolk.io**

Kinvolk

🚆 Driving Kubernetes Forward

Engineering products + support services for Kubernetes, containers, process management and Linux user-space + kernel

Blog: **kinvolk.io/blog**
Github: **kinvolk**
Twitter: **kinvolkio**
Email: **hello@kinvolk.io**

strace

Kubernetes

BPF

Traceloop

Tracing system calls in cgroups using BPF and overwritable ring buffers

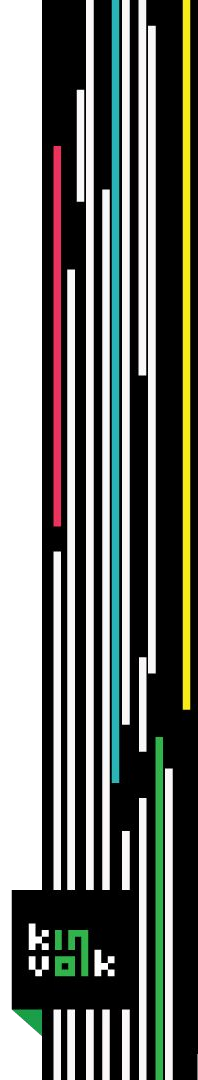
https://github.com/kinvolk/traceloop

Inspektor Gadget

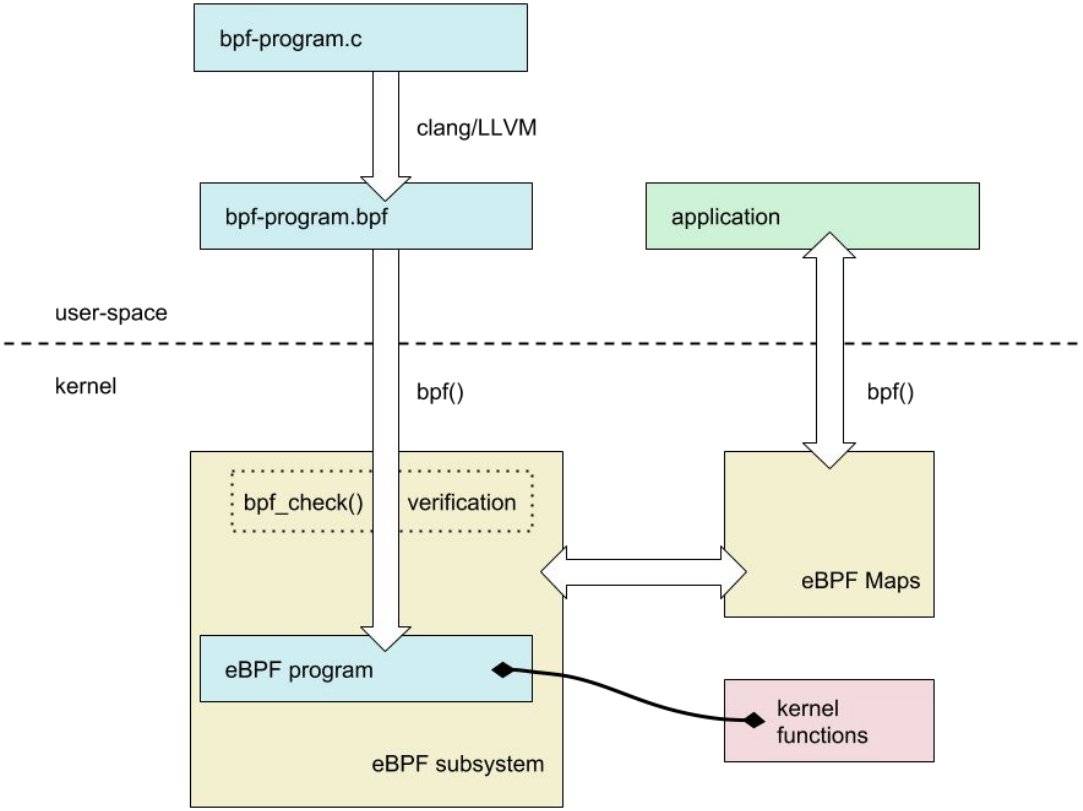
Collection of gadgets for developers of Kubernetes applications

https://github.com/kinvolk/inspektor-gadget

Kubernetes Slack: #inspektor-gadget



BPF in a nutshell



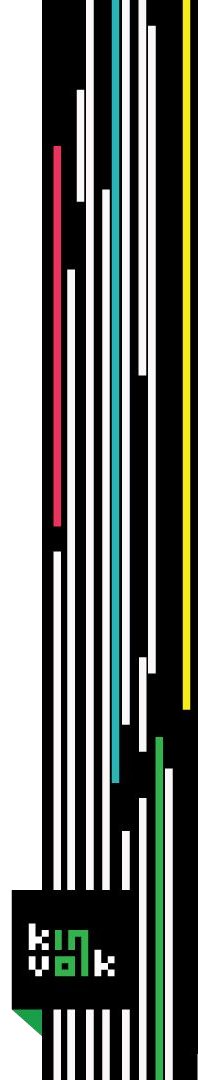
Debugging with “strace” on Kubernetes

- Strace is slow
  - cannot be used for all pods on prod

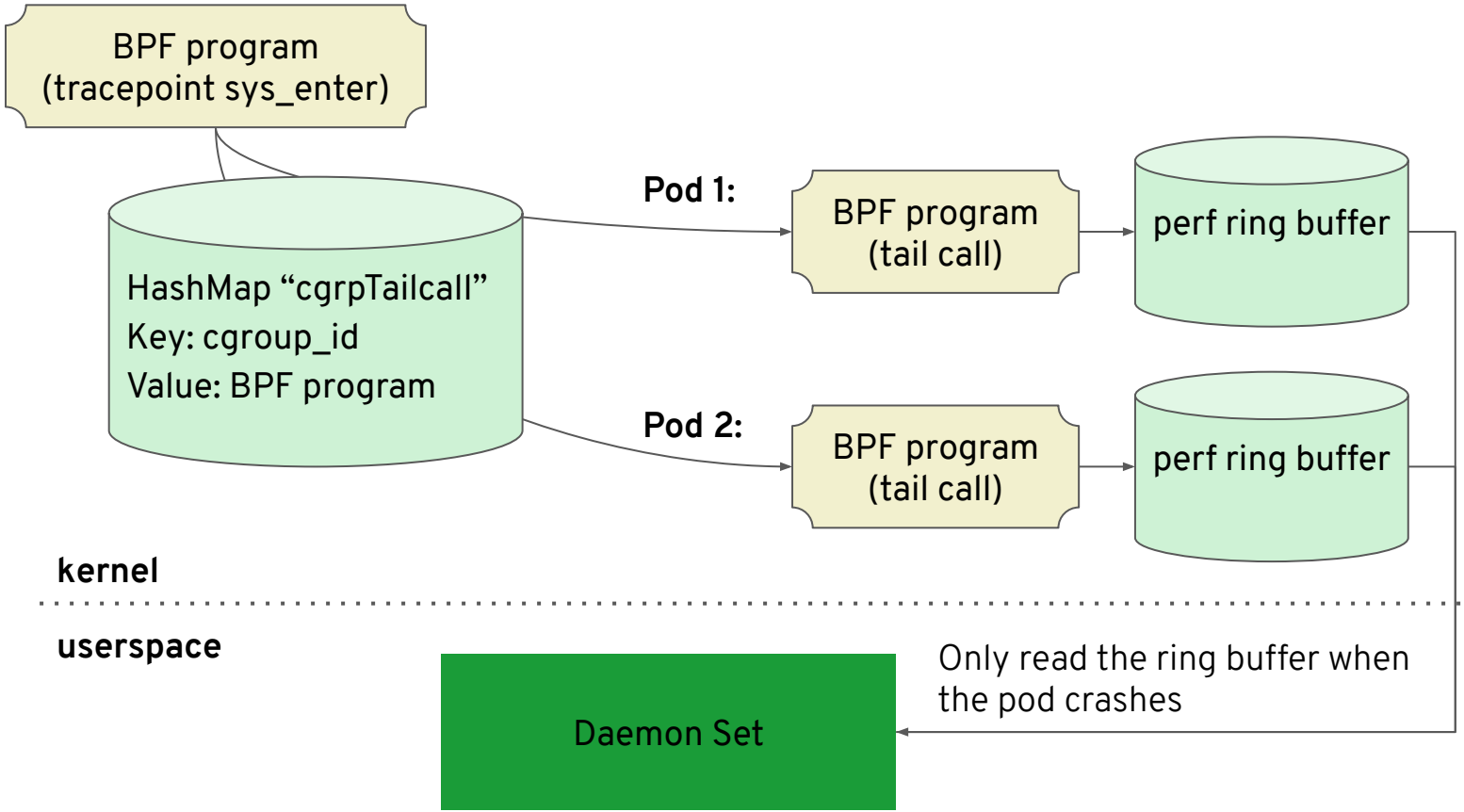
- We need to know what’s going to crash
  - And start strace just before
  - Problem with unreproducible crashes
- Idea: “flight recorder”
  - Capture syscalls with BPF instead of strace
  - Send the events to a per-pod ring buffer
  - Only read the ring buffer when the pod crashed

Comparing strace and traceloop

	strace	**traceloop**
Capture method	ptrace	BPF on tracepoints
Granularity	process	cgroup
Speed	slow	fast
Reliability	Synchronous Cannot lose events	Asynchronous Can lose events Can fail to read buffers (EFAULT)

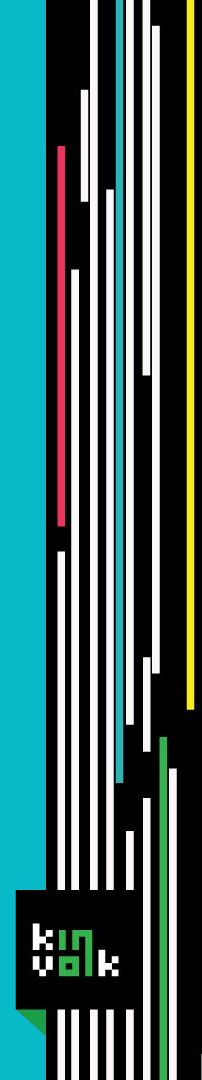


Debugging with “strace” on Kubernetes



DEMO

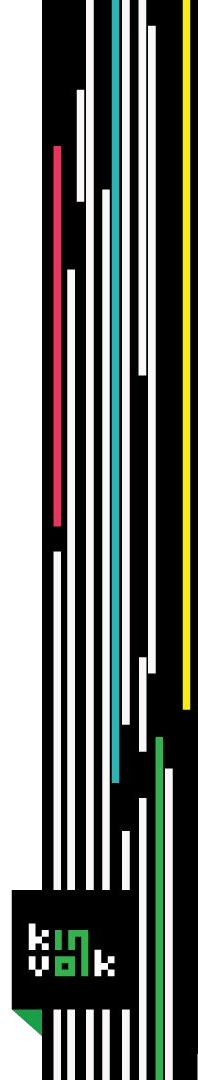
traceloop



Adapting BPF tracing tools to Kubernetes

What do we need for Kubernetes?

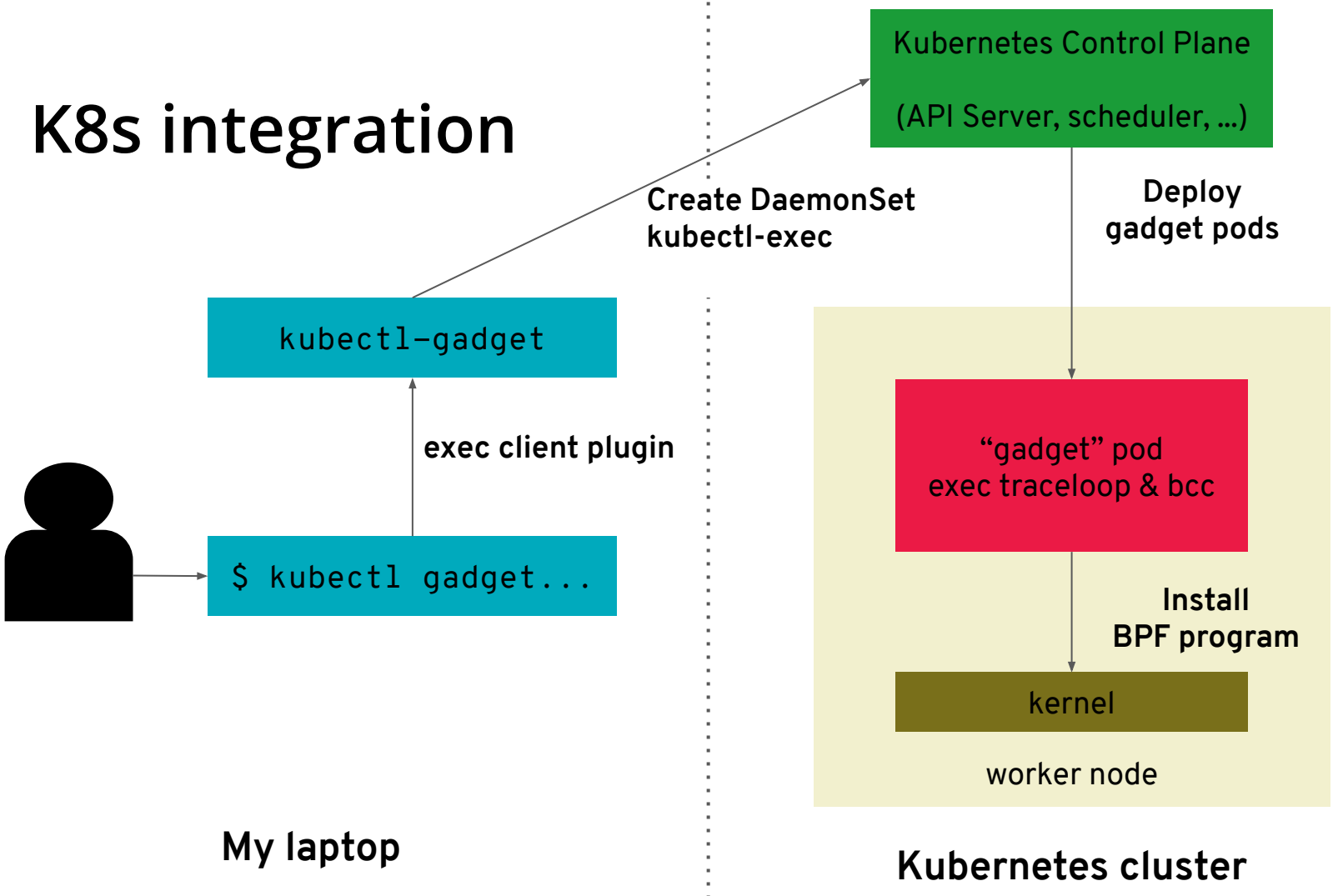
- Granularity of tracing: your pod
  - Pids are not useful when we don't know which container it is
  - We don't want to trace all the system processes on a node
- Aggregation
  - Using Kubernetes labels
- kubectl-like UX experience
  - Developers should not need to SSH
  - Developers should not need to deploy a pod + kubectl-exec for each tracing



Tracing tools for Kubernetes

Linux tracing tool	Kubernetes tracing tool
bpftrace https://github.com/iovisor/bpftrace	kubectl trace https://github.com/iovisor/kubectl-trace
BPF Compiler Collection (BCC) https://github.com/iovisor/bcc	Inspektor Gadget https://github.com/kinvolk/inspektor-gadget
 traceloop https://github.com/kinvolk/traceloop	

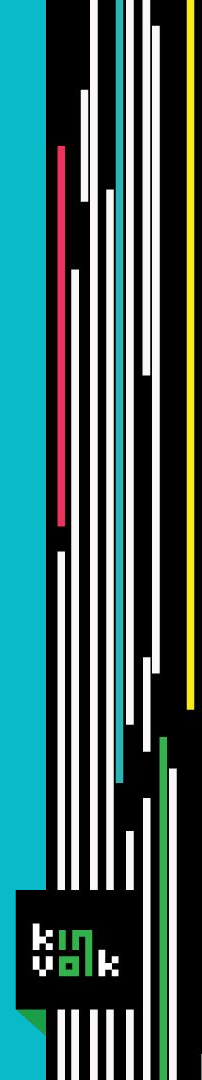
K8s integration



DEMO

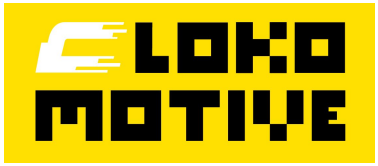
Inspektor Gadget +traceloop

Stopgaps in traceloop



Inspektor Gadget + traceloop

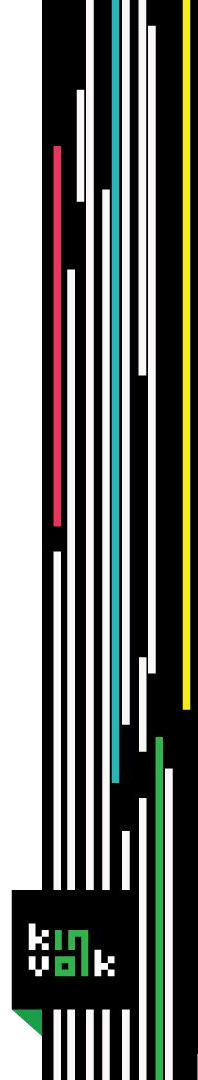
- Works on:
  - Kinvolk's Flatcar Container Linux + Lokomotive
  - Minikube (Linux 4.14)
  - GKE (Linux 4.14)



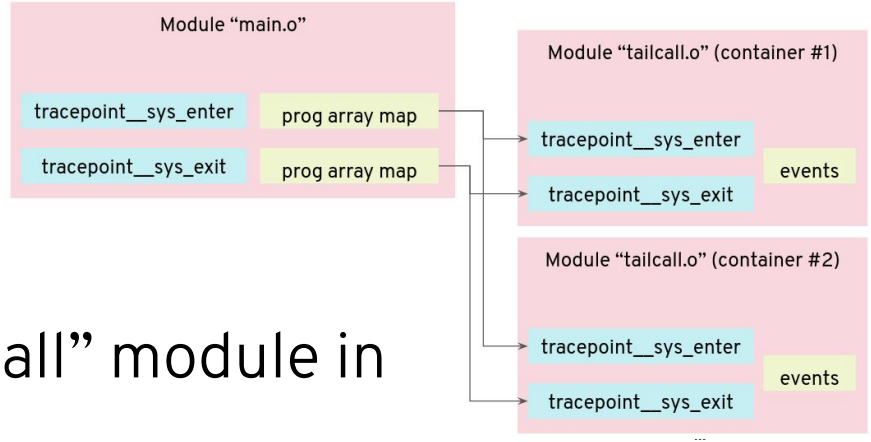
- Without:
  - Linux >= 4.18 (for bpf_get_current_cgroup_id)
  - cgroup-v2
  - runc without using OCI hooks

No cgroup-v2

- bpf_get_current_cgroup_id not available
  - Detect new namespaces:
    struct task_struct -> struct nsproxy -> struct uts_namespace -> inode
  - Find out struct offsets at startup to support several kernel versions without recompiling the BPF program



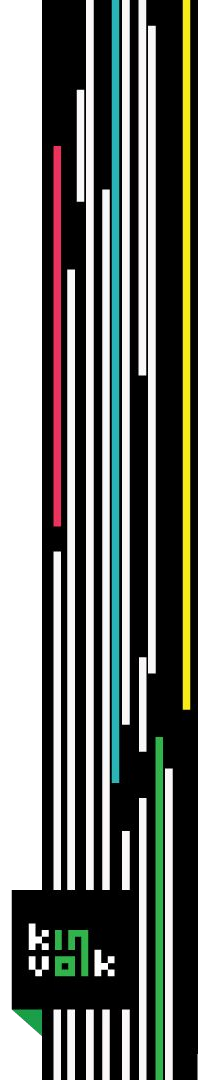
No OCI hooks



- Cannot add a new “tailcall” module in the PreStart OCI hook
- Cannot directly use the Kubernetes API
  - That would be too late to get the early syscalls

No OCI hooks

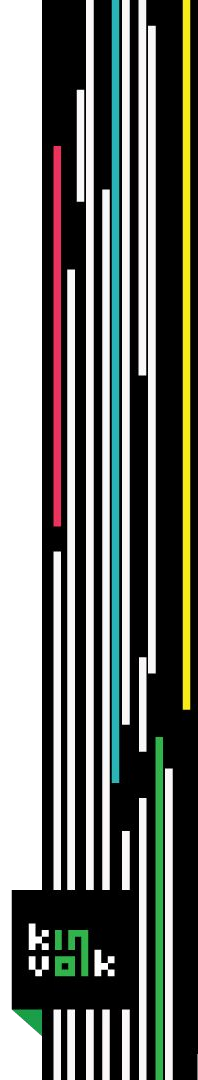
- Add a pool of “tailcall” modules for future containers
- When detecting a new container from BPF, plug the prog map array from BPF
- Reconcile with containers from the Kubernetes API



Other gadgets

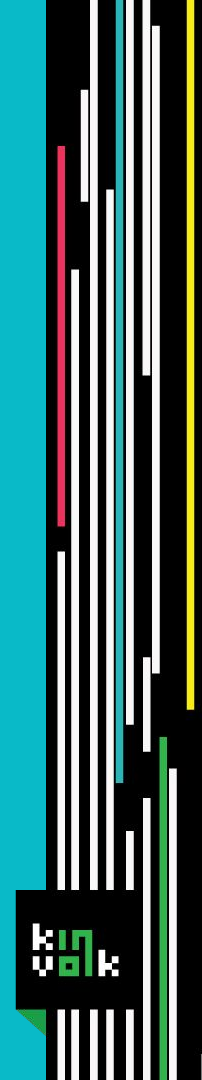
Use cases

- Debugging your app
  - ✔️ traceloop
  - ✔️ opensnoop, execsnoop
  - ❌ WIP: tcptop
- Help writing Kubernetes network policies
  - ❌ TODO (tcpconnect)
- Help writing Kubernetes PSP
  - ❌ WIP: capabilities



DEMO

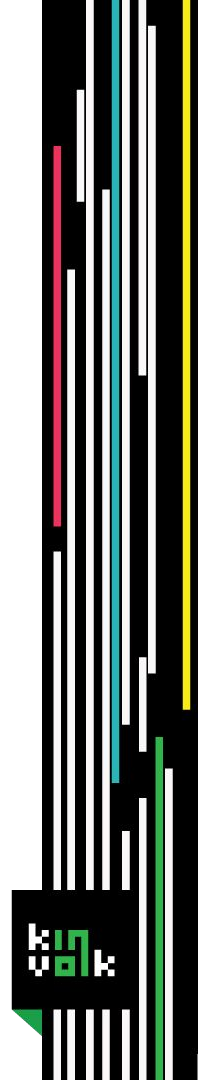
Inspektor Gadget
+ execsnoop, opensnoop



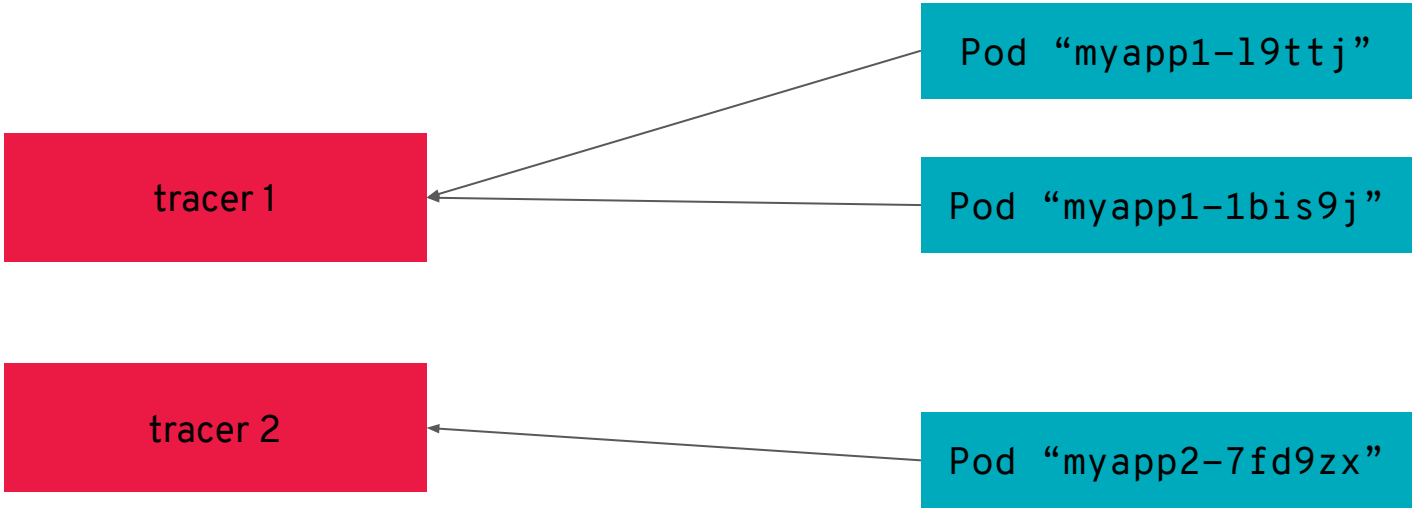
Gadget Tracer Manager

Selecting containers

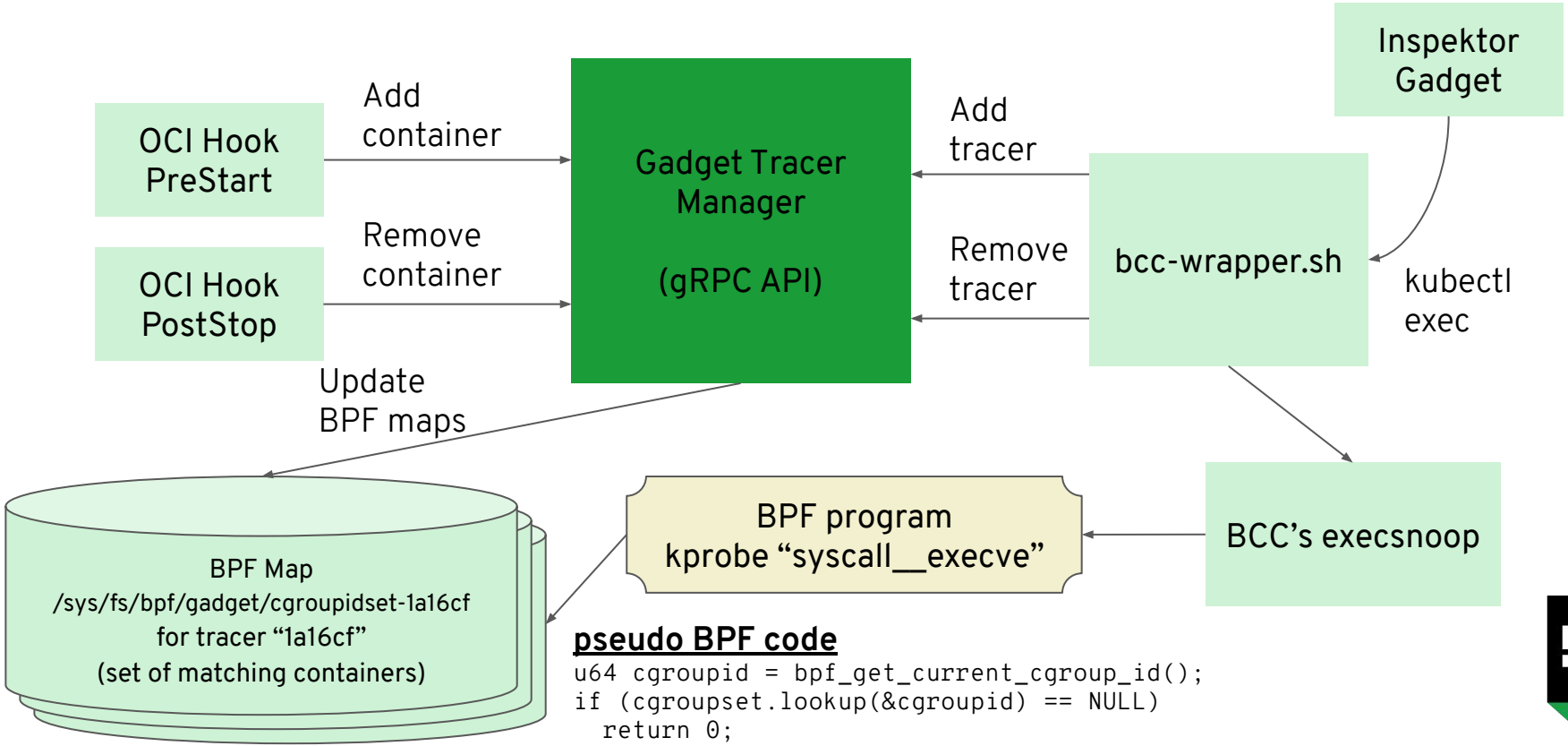
```
$ kubectl gadget execsnoop \
      --label k8s-app=myapp1,tier=bar \
      --namespace default \
      --podname myapp1-l9ttj \
      --node ip-10-0-12-31 \
      --containerindex 0
```



Pods & tracers come and go



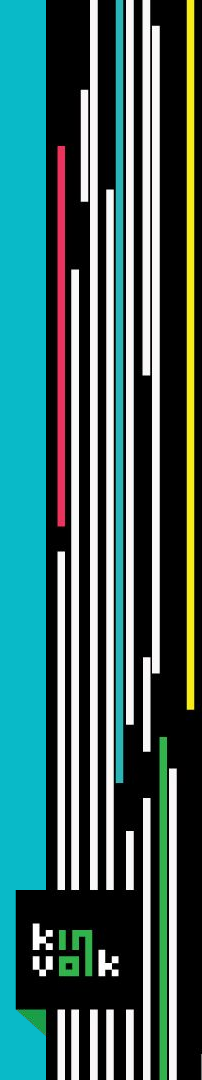
Keeping track of containers & tracers



pseudo BPF code

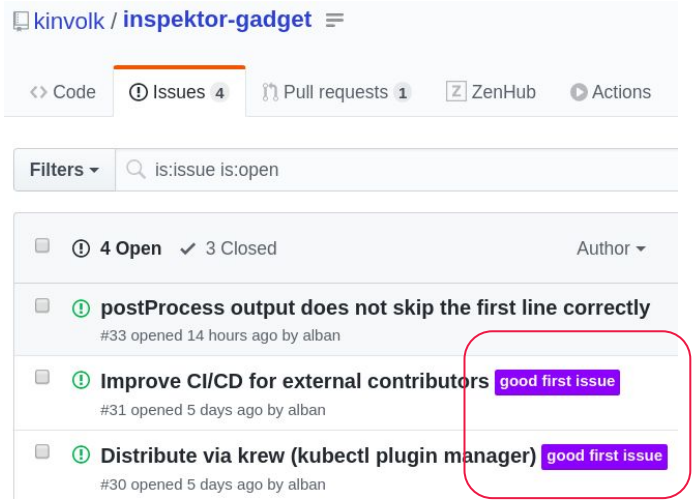
```
u64 cgroupid = bpf_get_current_cgroup_id();
if (cgroupset.lookup(&cgroupid) == NULL)
  return 0;
```

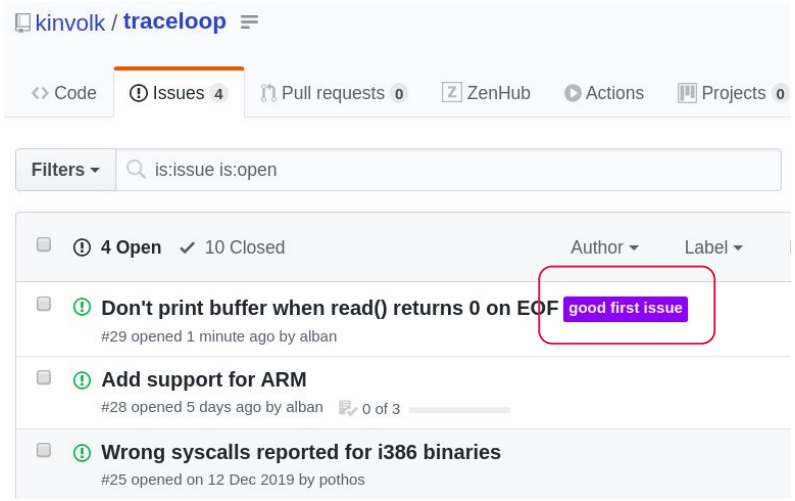
Contribute



How to contribute

- Join the Kubernetes Slack #inspektor-gadget
- GitHub issues with label “good first issue”





Thank you!

Alban Crequy

Github: **alban**
Twitter: **albcr**
Email: **alban@kinvolk.io**

Kinvolk

Blog: **kinvolk.io/blog**
Github: **kinvolk**
Twitter: **kinvolkio**
Email: **hello@kinvolk.io**

Kubernetes Slack: #inspektor-gadget
Slides: https://tinyurl.com/fosdem-gadget