golang

Debugging and Observing Containers Like a Pro

October 25, 2022 | Milan, Italy

Jose Blanquicet

Senior Software Engineer

Microsoft

How do you debug a containerized (and distributed) application?

Agenda

- Quick introduction to **eBPF**
- Identify difficulties **using BCC (eBPF-based) and standard Linux tools** to debug container issues
- Introducing **Local Gadget**
- Try **Go packages** to debug and observe containers
- Debug container issues **using Local Gadget**
- **The future** of Local Gadget (Roadmap)

Introduction

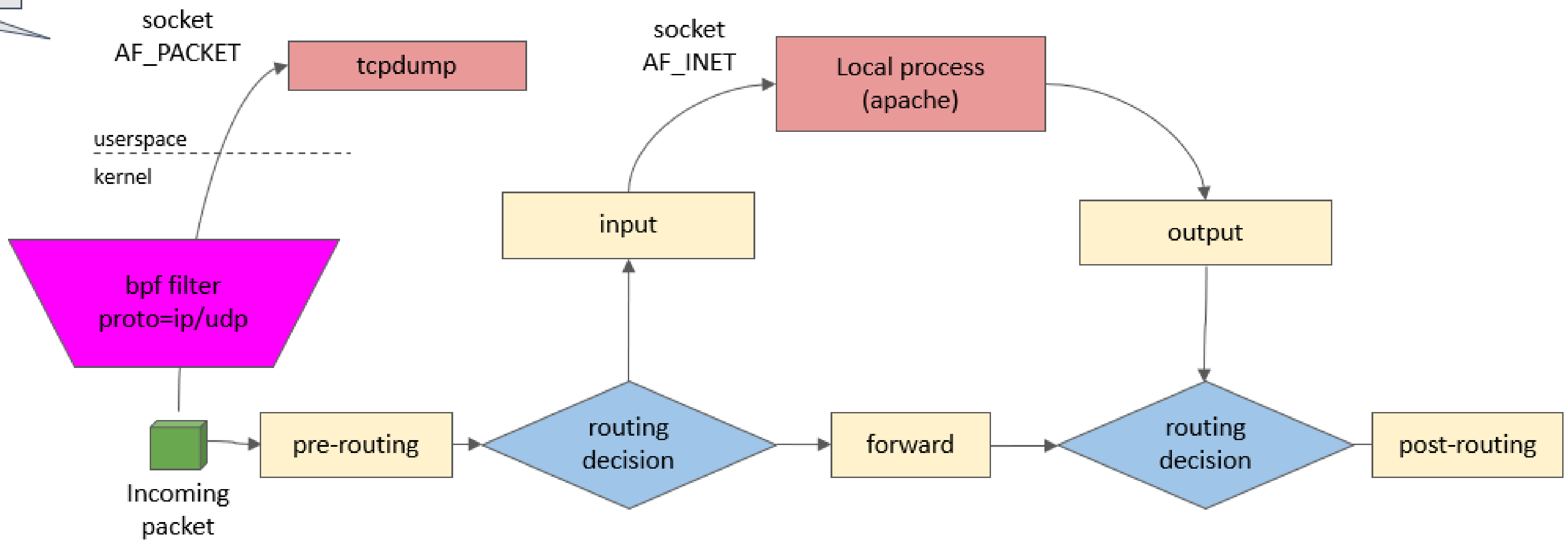
eBPF

https://ebpf.io/: eBPF - Introduction, Tutorials & Community Resources

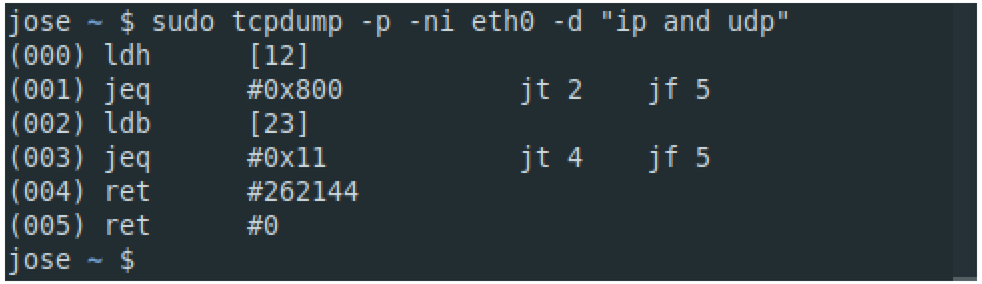
Intro classic BPF

Have you ever used tcpdump (classic Berkeley Packet Filter)?

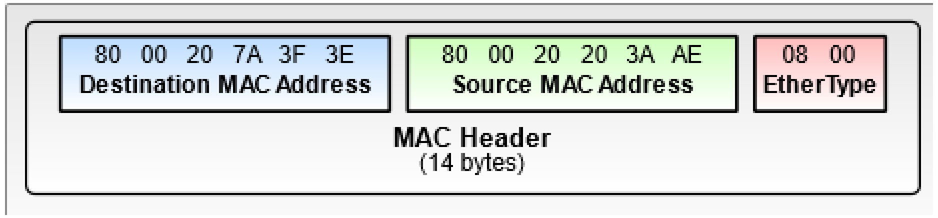
“raw” AF_PACKET socket: receiving headers + payload

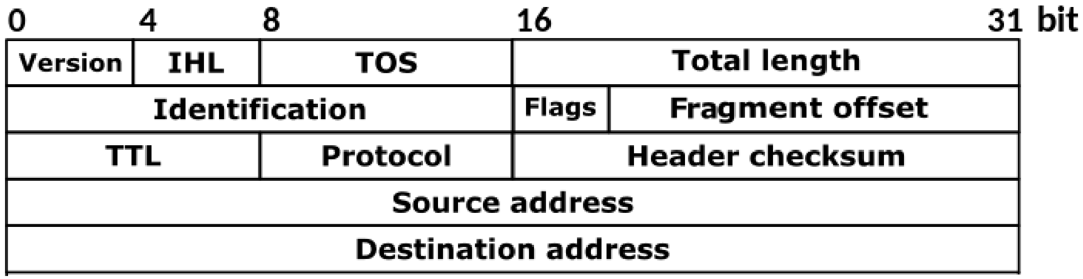


Intro classic BPF (2)



```
jose ~ $ sudo tcpdump -p -ni eth0 -d "ip and udp"
(000) ldh      [12]
(001) jeq      #0x800           jt 2    jf 5
(002) ldb      [23]
(003) jeq      #0x11            jt 4    jf 5
(004) ret      #262144
(005) ret      #0
jose ~ $
```





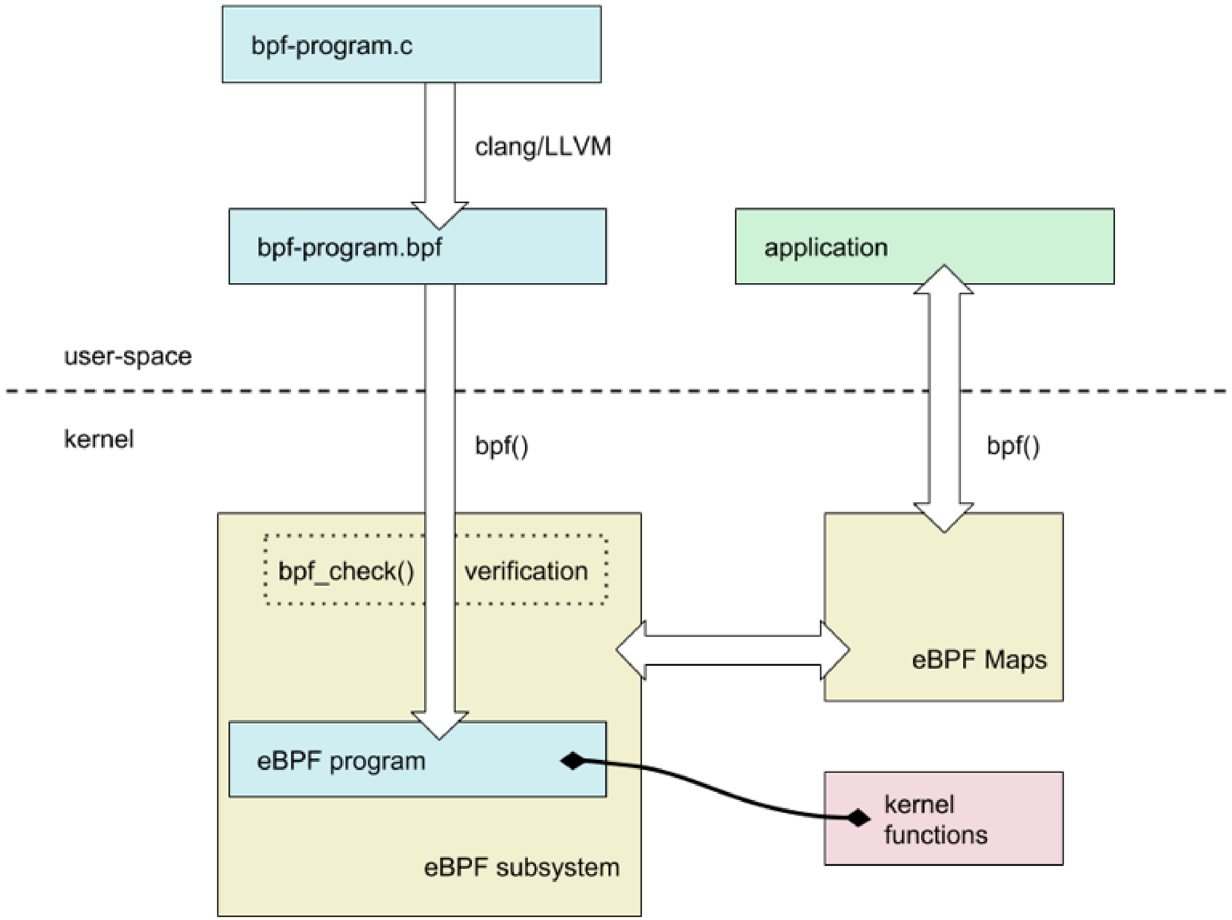
https://commons.wikimedia.org/wiki/File:Ethernet_Type_II_Frame_format.svg

https://commons.wikimedia.org/wiki/File:IPv4_Packet-en.svg

eBPF

- BPF was **extended** in 2013 with some new features that make it more powerful:
  - More registries, eBPF maps, helpers, etc.
- More use cases:
  - Tracing
  - Networking
  - Security
- More about eBPF: https://ebpf.io/

eBPF - The whole picture



Why eBPF?

- Brings flexibility to the kernel
    - We don't need to wait for a new kernel release to implement a new feature
- It's efficient
    - Just-in-Time (JIT) compiler makes the performance overhead low
- It's safe
    - User provided code can be running in a "sandbox" environment in the kernel

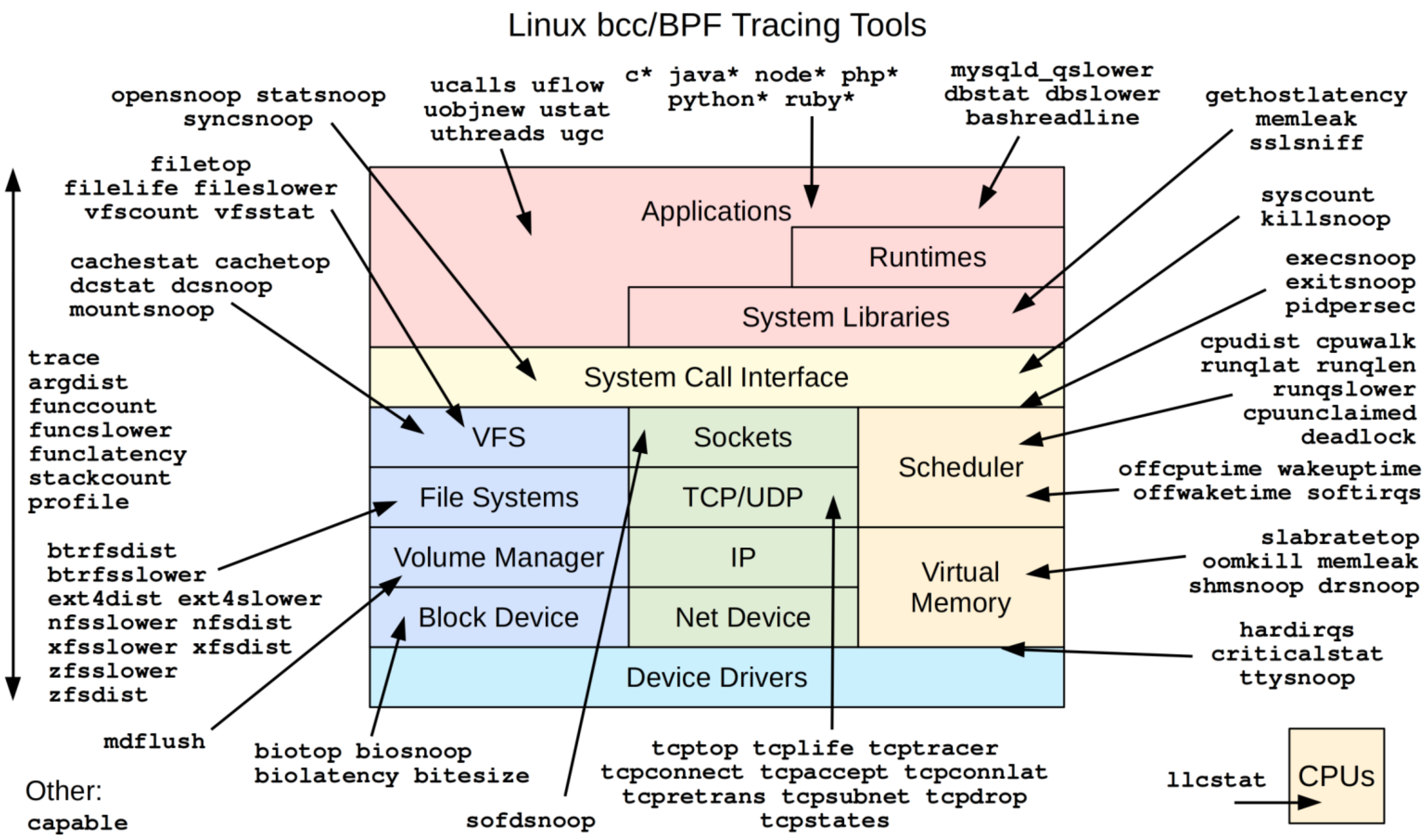
*“JavaScript is to the web browsers
as eBPF is to the Linux kernel”*

ebpf-go (Go library)

- Pure-Go library to handle eBPF objects (maps, programs, link, etc.)
  - Doesn’t depend on CGO
- Mainly maintained by Cilium and Cloudflare
- Packages
  - cmd/bpf2go: allows compiling and embedding eBPF programs written in C within Go code
  - link: allows attaching eBPF to various hooks
  - perf: allows reading from A PERF_EVENT_ARRAY
  - ringbuf: allows reading from a BPF_MAP_TYPE_RINGBUF map

https://github.com/cilium/ebpf: ebpf-go library

BCC (eBPF-based) tools



https://github.com/iovisor/bcc#tools 2019

https://github.com/iovisor/bcc iovisor/bcc: Tools for BPF-based Linux IO analysis, networking, monitoring, and more

Demo #1: Debug container issues **using BCC and standard Linux tools**

https://github.com/iovisor/bcc iovisor/bcc: Tools for BPF-based Linux IO analysis, networking, monitoring, and more

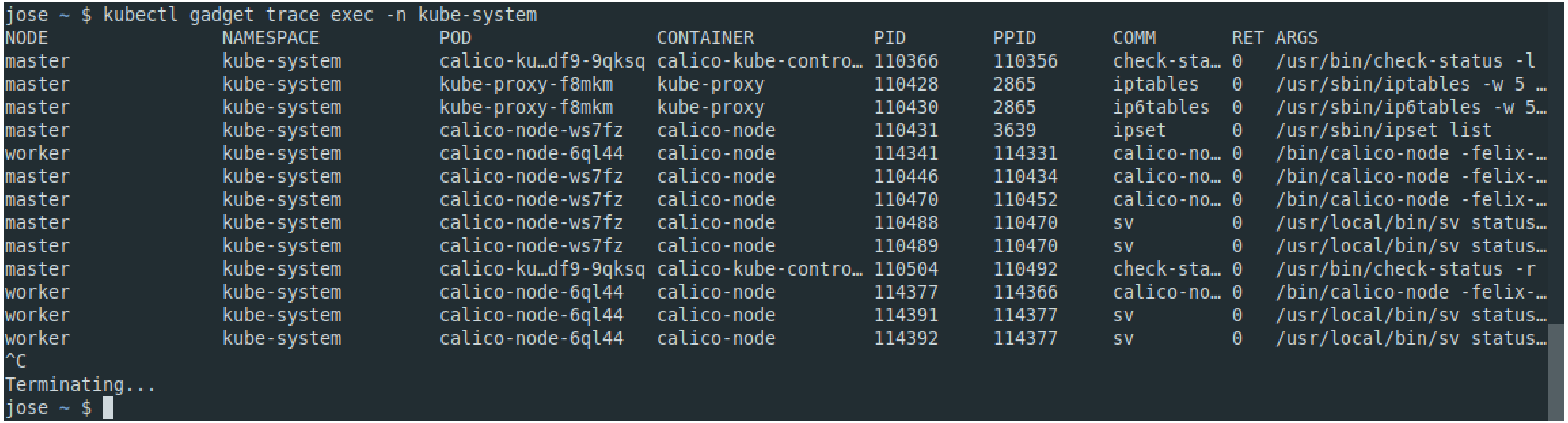
Demo #1: What issues did we find?

- Need to **manually** retrieve container information (PID1, namespaces, etc.).
- **Extracting/Filtering** the data of interest is difficult.
- Switching between Linux namespaces to run tools in the **correct context**.

Inspektor Gadget

https://www.inspektor-gadget.io/docs/latest

Inspektor Gadget



```
jose ~ $ kubectl gadget trace exec -n kube-system
NODE                NAMESPACE           POD                 CONTAINER           PID        PPID       COMM       RET ARGS
master              kube-system         calico-ku…df9-9qksq calico-kube-contro… 110366     110356     check-sta… 0   /usr/bin/check-status -l
master              kube-system         kube-proxy-f8mkm    kube-proxy          110428     2865       iptables   0   /usr/sbin/iptables -w 5 …
master              kube-system         kube-proxy-f8mkm    kube-proxy          110430     2865       ip6tables  0   /usr/sbin/ip6tables -w 5…
master              kube-system         calico-node-ws7fz   calico-node         110431     3639       ipset      0   /usr/sbin/ipset list
worker              kube-system         calico-node-6ql44   calico-node         114341     114331     calico-no… 0   /bin/calico-node -felix-…
master              kube-system         calico-node-ws7fz   calico-node         110446     110434     calico-no… 0   /bin/calico-node -felix-…
master              kube-system         calico-node-ws7fz   calico-node         110470     110452     calico-no… 0   /bin/calico-node -felix-…
master              kube-system         calico-node-ws7fz   calico-node         110488     110470     sv         0   /usr/local/bin/sv status…
master              kube-system         calico-node-ws7fz   calico-node         110489     110470     sv         0   /usr/local/bin/sv status…
master              kube-system         calico-ku…df9-9qksq calico-kube-contro… 110504     110492     check-sta… 0   /usr/bin/check-status -r
worker              kube-system         calico-node-6ql44   calico-node         114377     114366     calico-no… 0   /bin/calico-node -felix-…
worker              kube-system         calico-node-6ql44   calico-node         114391     114377     sv         0   /usr/local/bin/sv status…
worker              kube-system         calico-node-6ql44   calico-node         114392     114377     sv         0   /usr/local/bin/sv status…
^C
Terminating...
jose ~ $
```

What about these use-cases?

- The Kubernetes API server is down.
- Working outside Kubernetes environment.
- You are implementing a tool that needs to get insights from the node:
  - Include the local-gadget binary in your container image, and your app simply execs local-gadget (JSON format).
  - If your app is in **Go**, you can run our tracers using the packages we created.

Local Gadget

https://www.inspektor-gadget.io/docs/latest/local-gadget/

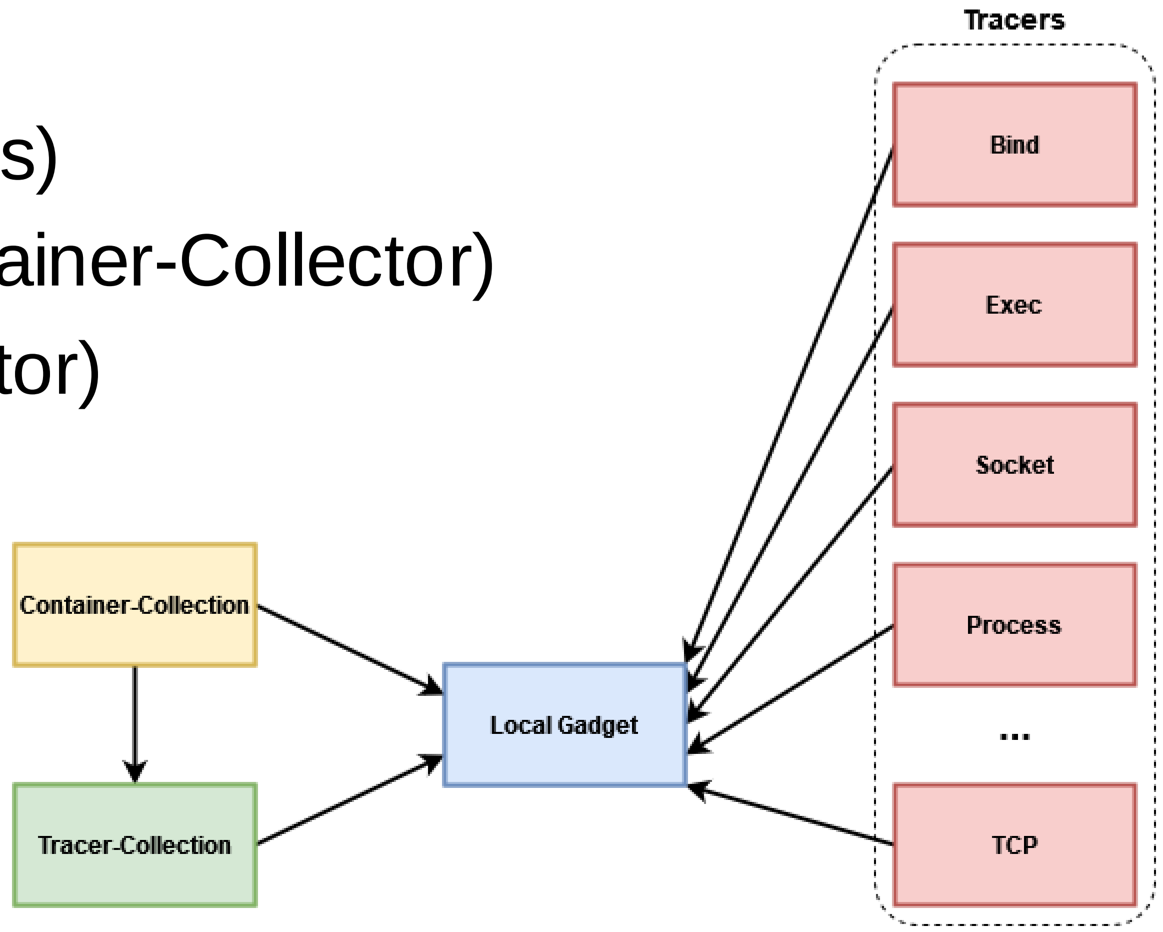
Local Gadget: What is it?

- It is a **single binary (statically linked)**.
- Allows you to trace local containers using **eBPF**.
- Enriches events with **Kubernetes metadata**.
- Can be used for trace Kubernetes and **non**-Kubernetes containers.
- Available tools (or “gadgets”): Some based on **BCC tools** (e.g., trace bind, exec, open events), as well as some developed by our team for other use cases (e.g., snapshot processes and sockets, trace DNS events, audit seccomp policies).

Local Gadget: Architecture

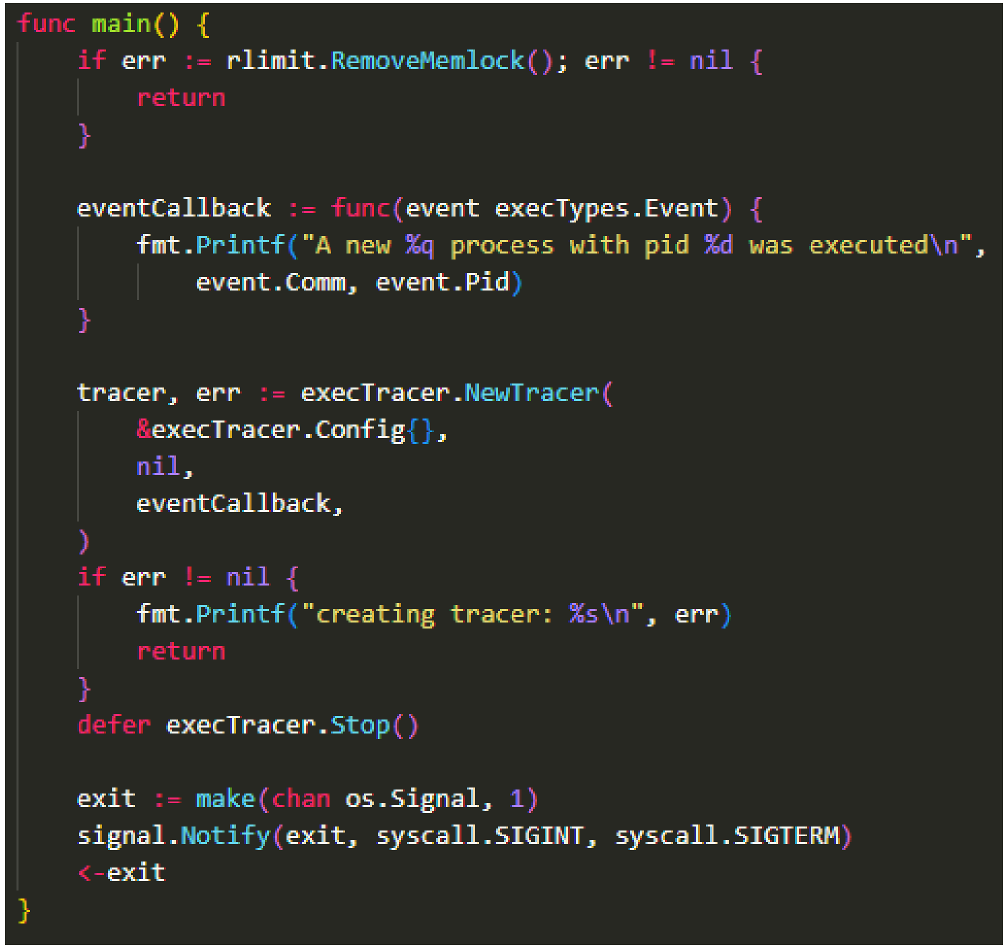
Three main tasks:

- Collect insights (Tracers)
- Data enrichment (Container-Collector)
- Filtering (Tracer-Collector)

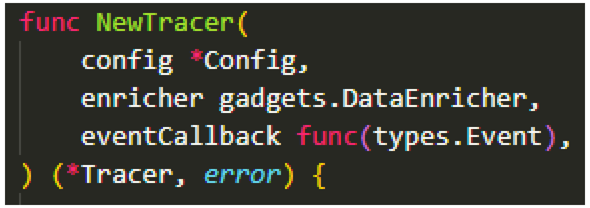


Collect insights (Tracers)

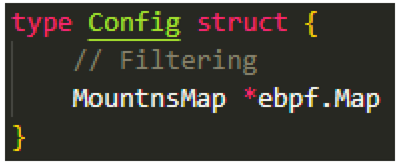
We wrote the control plane in Go, so that it can be easily used/integrated:



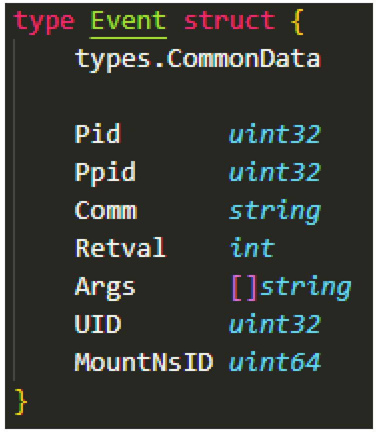
```go
func main() {
    if err := rlimit.RemoveMemlock(); err != nil {
        return
    }

    eventCallback := func(event execTypes.Event) {
        fmt.Printf("A new %q process with pid %d was executed\n",
            event.Comm, event.Pid)
    }

    tracer, err := execTracer.NewTracer(
        &execTracer.Config{},
        nil,
        eventCallback,
    )
    if err != nil {
        fmt.Printf("creating tracer: %s\n", err)
        return
    }
    defer execTracer.Stop()

    exit := make(chan os.Signal, 1)
    signal.Notify(exit, syscall.SIGINT, syscall.SIGTERM)
    <-exit
}
```



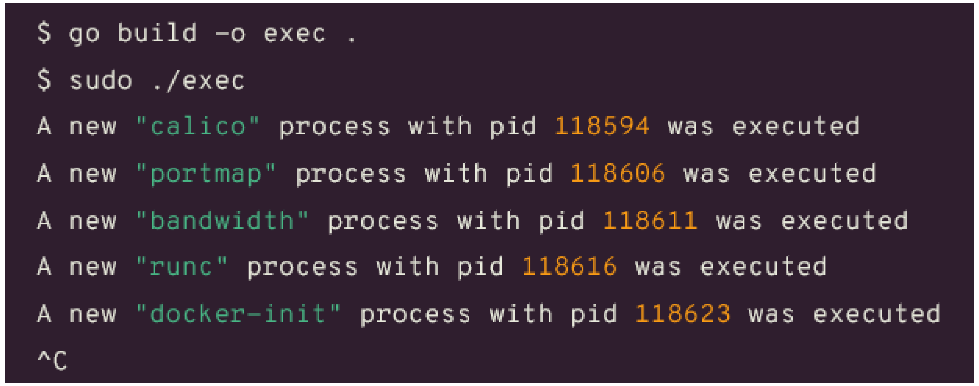
```go
func NewTracer(
    config *Config,
    enricher gadgets.DataEnricher,
    eventCallback func(types.Event),
) (*Tracer, error) {
```



```go
type Config struct {
    // Filtering
    MountnsMap *ebpf.Map
}
```



```go
type Event struct {
    types.CommonData

    Pid       uint32
    Ppid      uint32
    Comm      string
    Retval    int
    Args      []string
    UID       uint32
    MountNsID uint64
}
```



```
$ go build -o exec .
$ sudo ./exec
A new "calico" process with pid 118594 was executed
A new "portmap" process with pid 118606 was executed
A new "bandwidth" process with pid 118611 was executed
A new "runc" process with pid 118616 was executed
A new "docker-init" process with pid 118623 was executed
^C
```

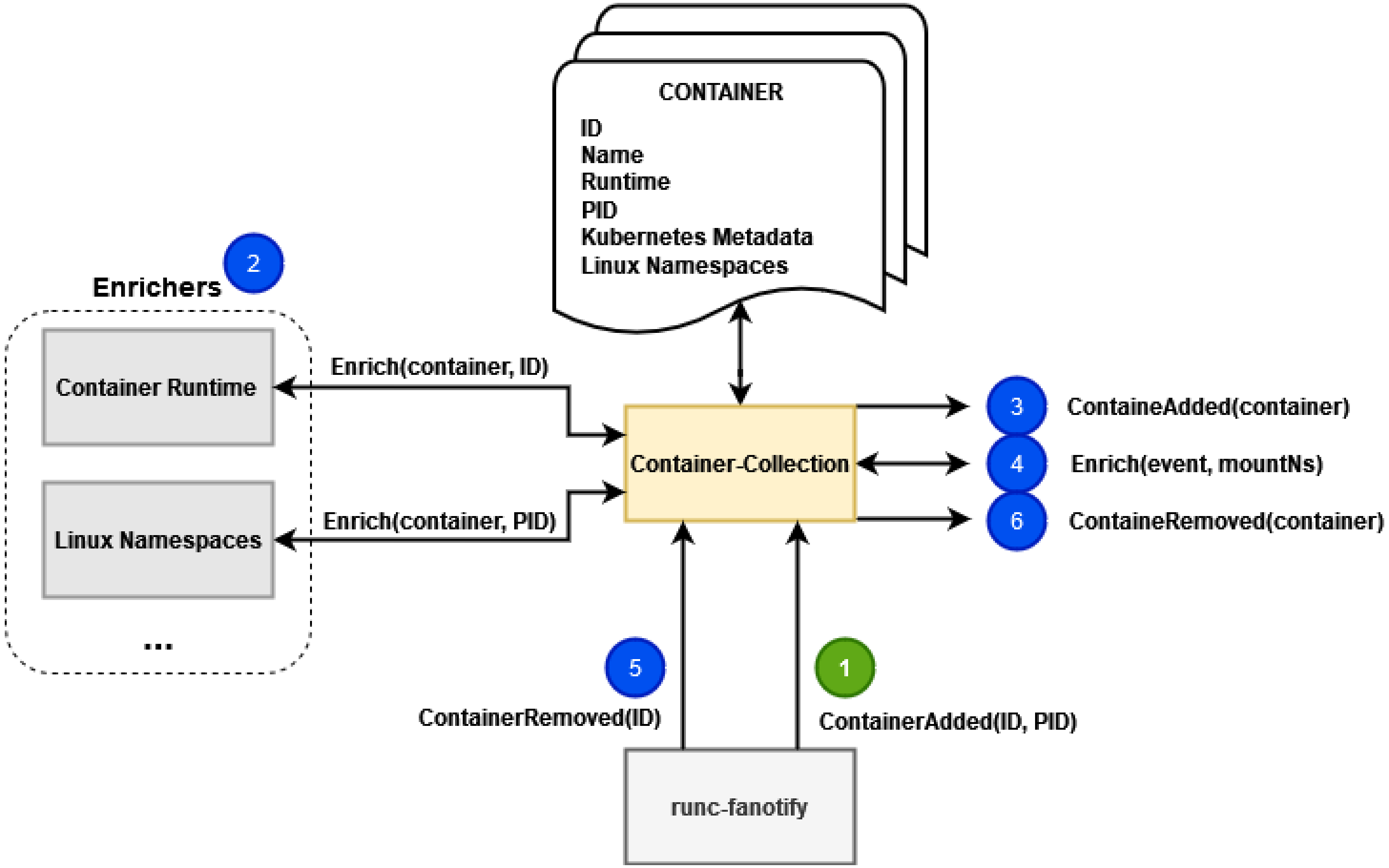
https://github.com/kinvolk/inspektor-gadget/tree/main/examples/gadgets/basic/trace/exec

Data enrichment (Container-Collection)

- **Enriches** events.
- Notifies about **container creation/deletion**.
- Get Kubernetes info

  from the

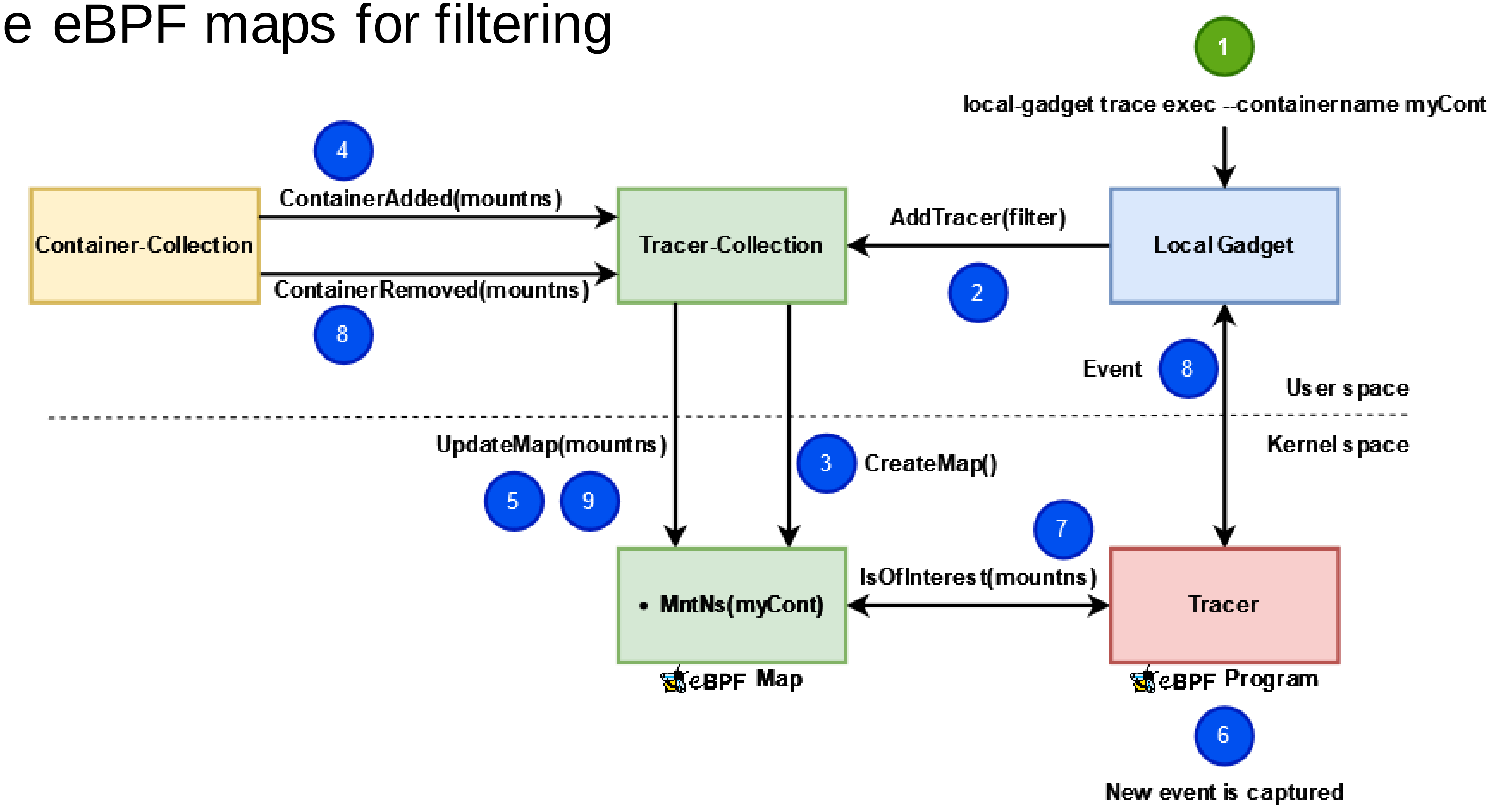
  Container Runtime.



https://github.com/kinvolk/inspektor-gadget/tree/main/examples/container-collection

Filtering (Tracer-Collection)

Manage eBPF maps for filtering



https://github.com/kinvolk/inspektor-gadget/tree/main/examples/gadgets/withfilter/trace/exec

Local-Gadget: Internal modules

- Do you want to know more about these components?
  - Blog: https://www.inspektor-gadget.io/blog
  - Examples: https://github.com/kinvolk/inspektor-gadget/tree/main/examples

Demo #2: Debug container issues **using Local Gadget**

Notes from Local Gadget demo

- **No manual steps** for filtering.
- Don’t lose any event at container **startup**.
- Enrichment of **Kubernetes** metadata.
- Debug Kubernetes containers **even if the API server is down.**

The future of Local Gadget

- Support **filtering by Kubernetes resources**: --k8s-namespace, --k8s-pod, --k8s-container.
- Support **non-Kubernetes** containers created by other container runtimes.
- Continue adding **more and more gadgets** … Is there a use-case where you think Local Gadget could be useful? **Reach us!**

Get involved!

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

Questions?

Thanks!

Jose Blanquicet

- GitHub: blanquicet
- Twitter: @jose_blanquicet
- Email: josebl@microsoft.com

Let’s connect!