# Debugging Container Workloads with Inspektor Gadget

Qasim Sarfraz

# Hello!

## Qasim Sarfraz

Software Engineer @ Microsoft

- I am from Lahore, Pakistan
- Currently, based in Hamburg, Germany
- I work with on OSS, Containers, Kubernetes
- I'm focused on Inspektor Gadget, kubectl aks and CoreDNS header plugin.
- I'm available at https://mqasimsarfraz.com

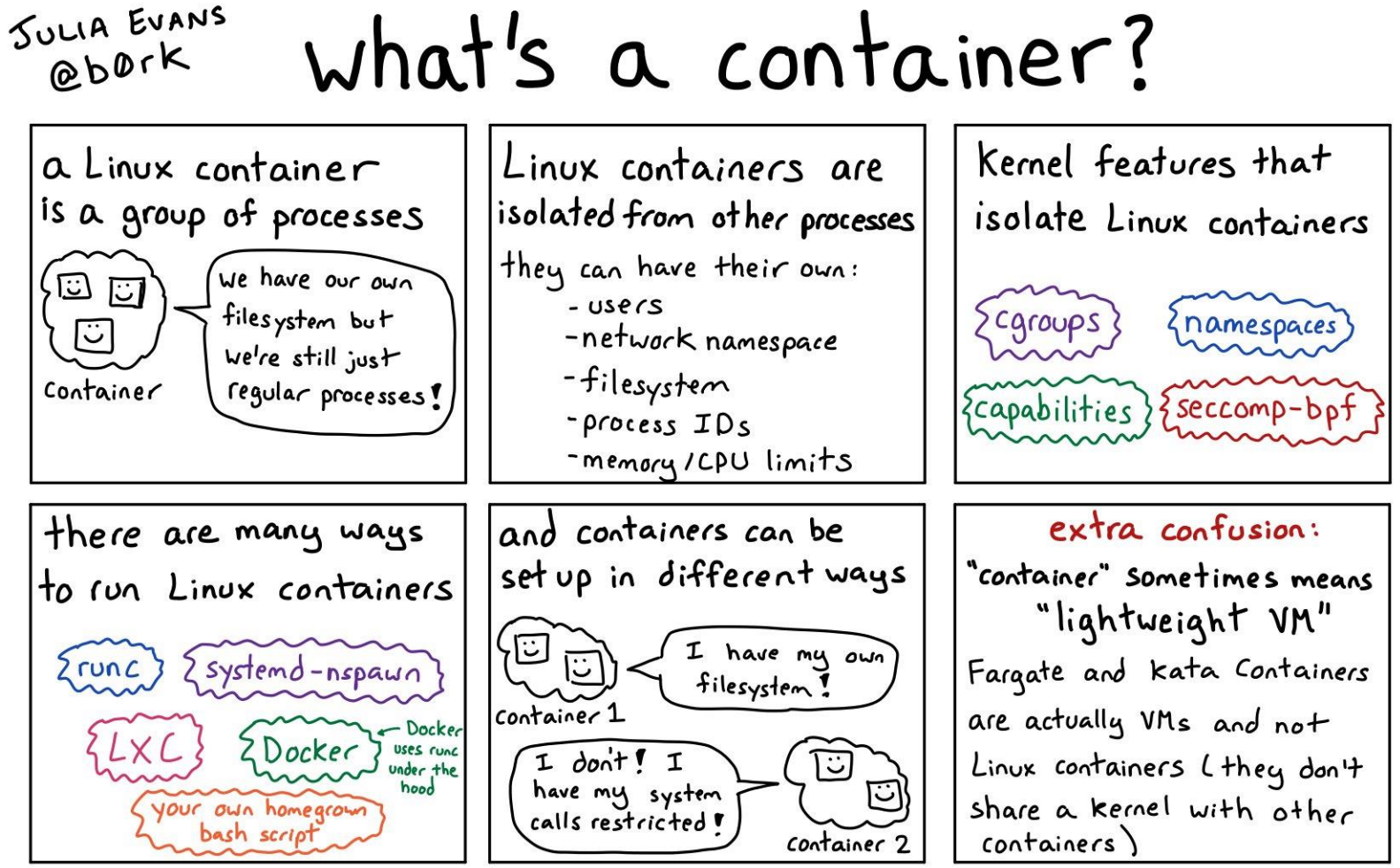
# Agenda

1. Challenges with Container Debugging
2. Inspektor Gadget as a **tool**
3. Inspektor Gadget as a **framework**
4. Contributing

# Challenges with Container Debugging

A journey from a process to a pod

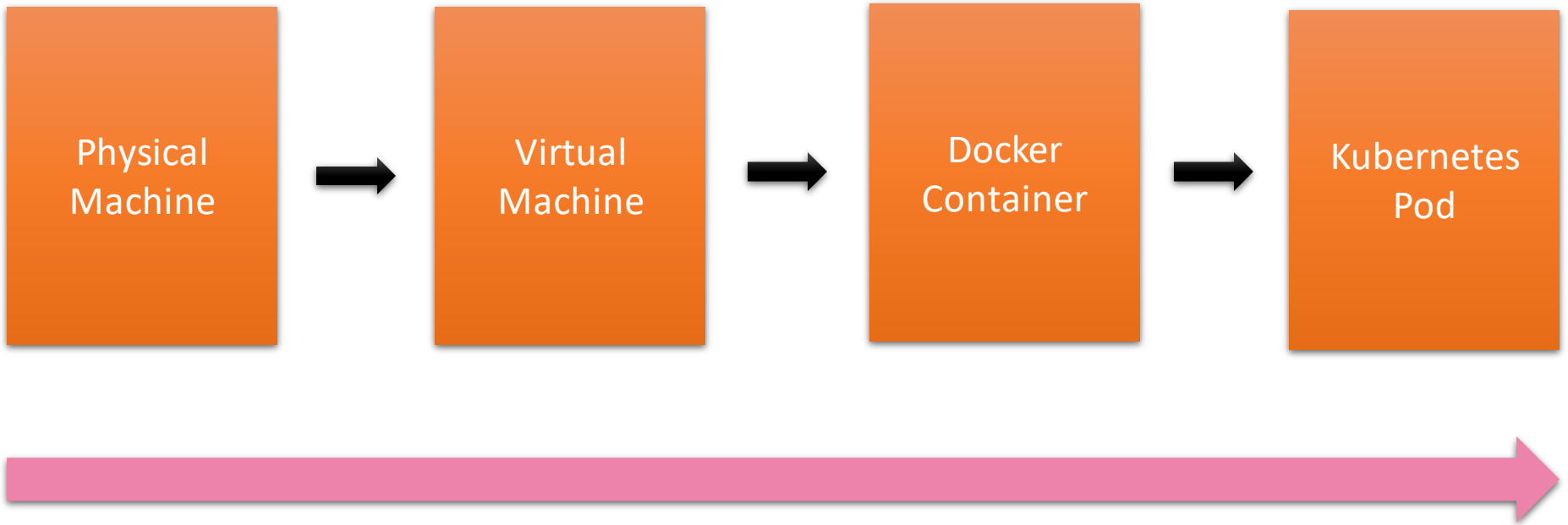
# Challenges with Container Debugging




https://twitter.com/b0rk/status/1225445956734390273

# Challenges with Container Debugging

The way we deploy our application workloads has changed a lot in the past years



# Challenges with Container Debugging

INSPEKTOR GADGET

Did the tools used to debug application workloads also evolved?

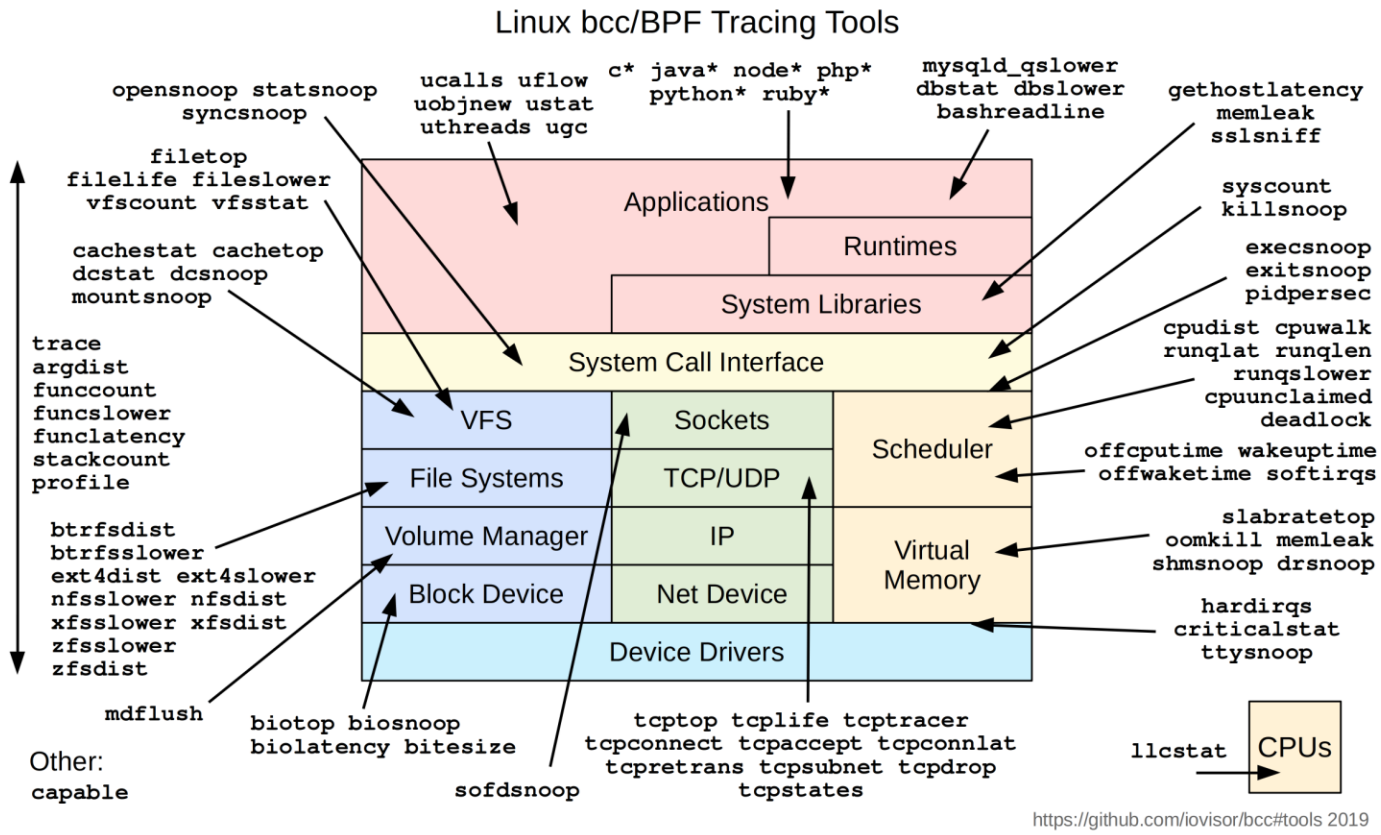
```
→ top -p 1

top - 11:23:00 up 5 days,  1:35,  1 user,  load average: 0,22, 0,28, 0,39
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3,3 us,  0,8 sy,  0,0 ni, 95,0 id,  0,8 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :  64009,1 total,  31811,8 free,   8216,7 used,  23980,6 buff/cache
MiB Swap:  16384,0 total,  16384,0 free,      0,0 used.  54125,1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0  168988  14244   8276 S   0,0   0,0   0:10.88 systemd
```

```
→ kubectl top pod -n kube-system coredns-6d4b75cb6d-hwt7m
NAME                       CPU(cores)   MEMORY(bytes)
coredns-6d4b75cb6d-hwt7m   4m           12Mi
```

How about tracing a system call from a process?

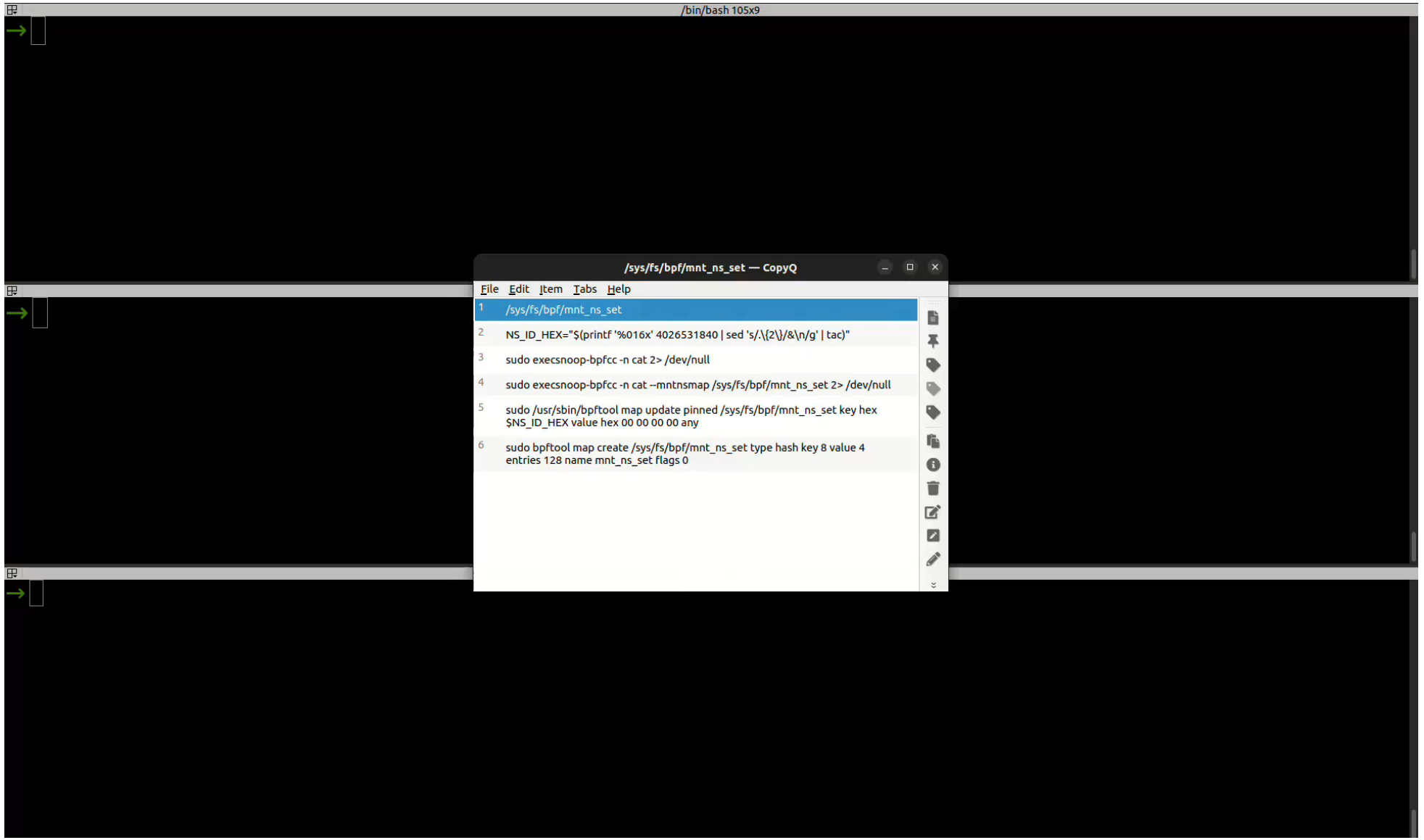
# Challenges with Container Debugging

- We can use [BPF Compiler Collection](#) (BCC)

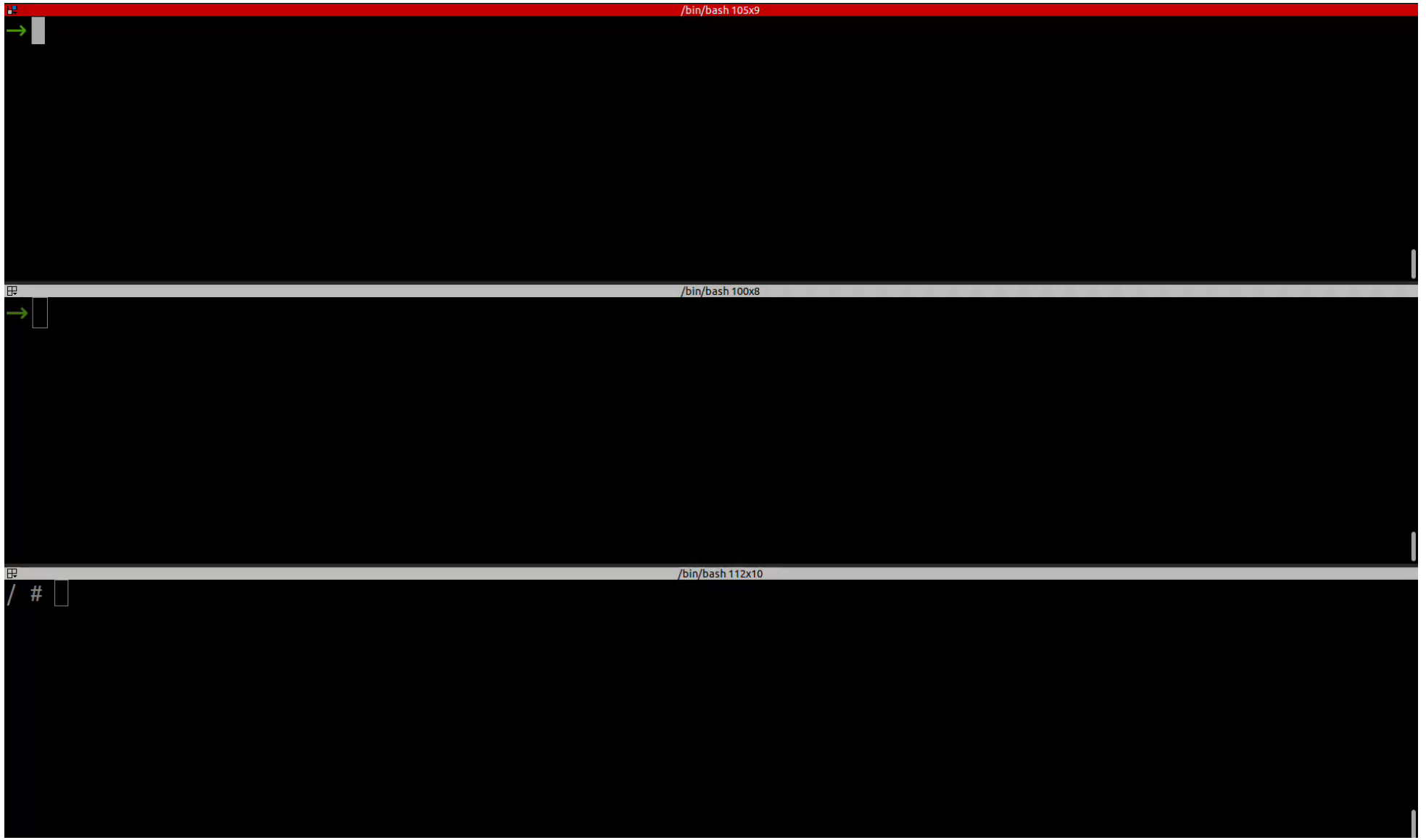


# Challenges with Container Debugging

INSPEKTOR GADGET



# Challenges with Container Debugging




https://github.com/iovisor/bcc/blob/master/docs/special_filtering.md

# Challenges with Container Debugging

INSPEKTOR GADGET

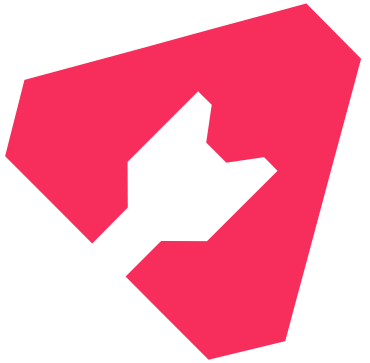
- Painful debugging experience.
- Lack of support of:
  - Container enrichment
  - Container filtering
- Distribution/Packaging Problem?

# Inspektor Gadget as a tool

https://www.inspektor-gadget.io

# Inspektor Gadget as a tool

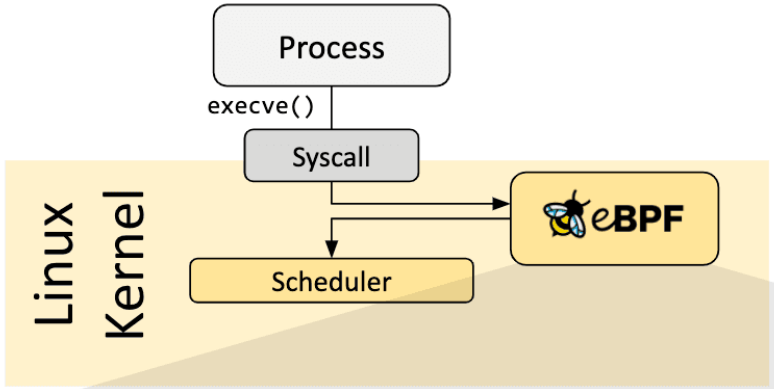
- A collection of **eBPF-based** gadgets to debug and inspect Kubernetes apps and resources
- It is a **CNCF** sandbox project
- Available as a krew plugin
  - **`kubectl krew install gadget`**
  - **`kubectl gadget deploy`**
- Also available as:
  - A CLI tool for Linux hosts called **ig**
  - A container image for **`kubectl debug`** ...



INSPEKTOR
GADGET

# What is eBPF?

What JavaScript is to the browser, eBPF is to the Linux kernel




```c
int syscall__ret_execve(struct pt_regs *ctx)
{
        struct comm_event event = {
                .pid = bpf_get_current_pid_tgid() >> 32,
                .type = TYPE_RETURN,
        };

        bpf_get_current_comm(&event.comm, sizeof(event.comm));
        comm_events.perf_submit(ctx, &event, sizeof(event));

        return 0;
}
```

https://ebpf.io/what-is-ebpf/

# Demo

# Can we improve it?

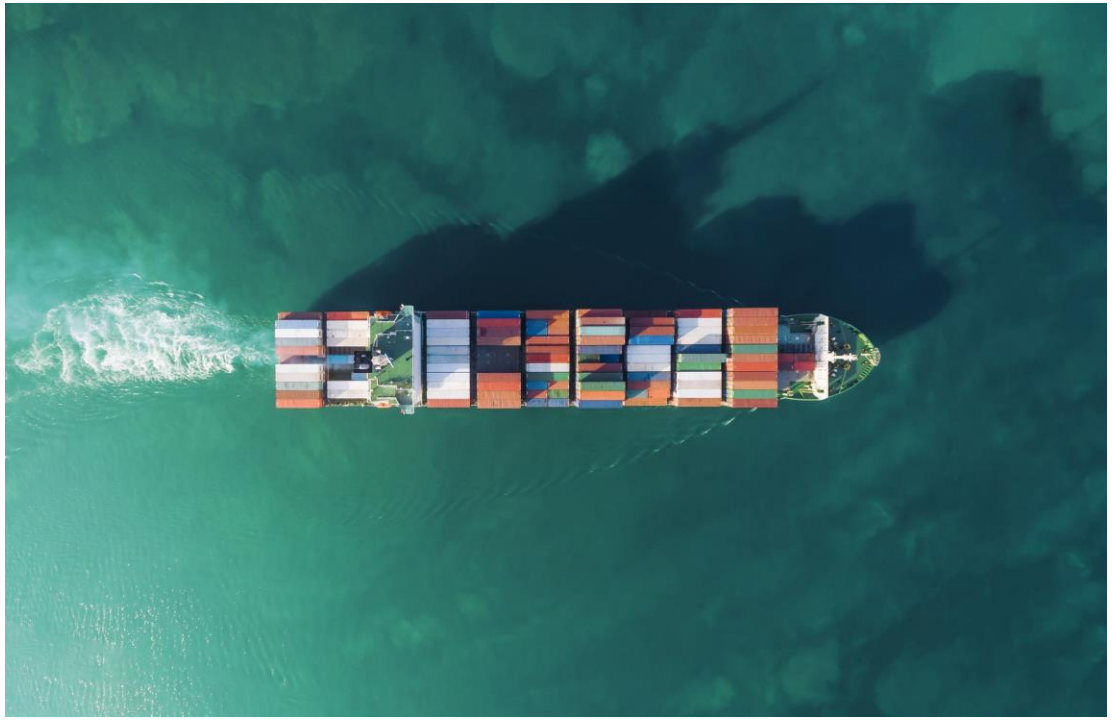
- Painful debugging experience. ✓
- Lack of support of:
  - Container enrichment ✓
  - Container filtering
- Distribution/Packaging Problem ?

# Inspektor Gadget as a framework

https://www.inspektor-gadget.io

# Inspektor Gadget as a framework

- Decouple the idea of gadgets from Inspektor Gadget
- Gadgets packaged as OCI images and Published to OCI registries
- Use docker like interface to **build, push** and **run** them



# Demo

# Interested in Contributing?

- We live on Github
- Use CONTRIBUTING.md as a starting point
- Look for issues with label: "good first issue"
- Reach us out at Kubernetes slack: #inspektor-gadget


# Thank you