



KubeCon



CloudNativeCon

Europe 2019



KubeCon



CloudNativeCon

Europe 2019

Back to the future with eBPF



@beatrizmrg, Beatriz Martínez Rubio

Supporting the Cloud Native World

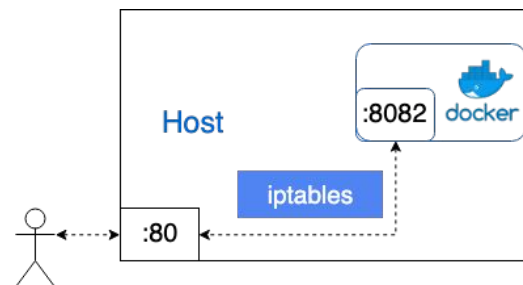
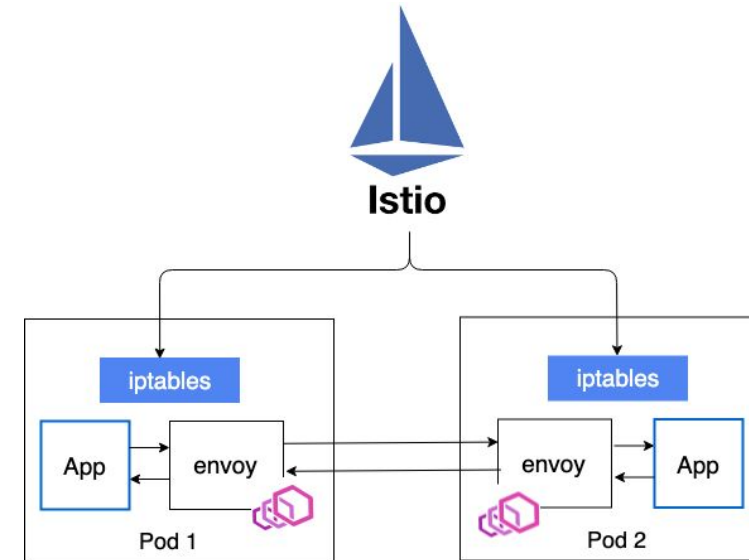
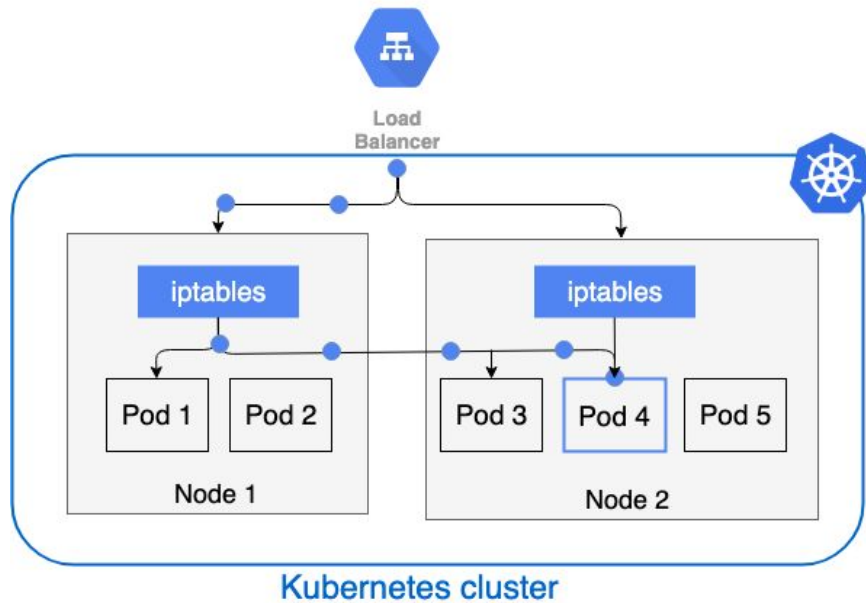


KubeCon



CloudNativeCon

Europe 2019



BPF (1992)

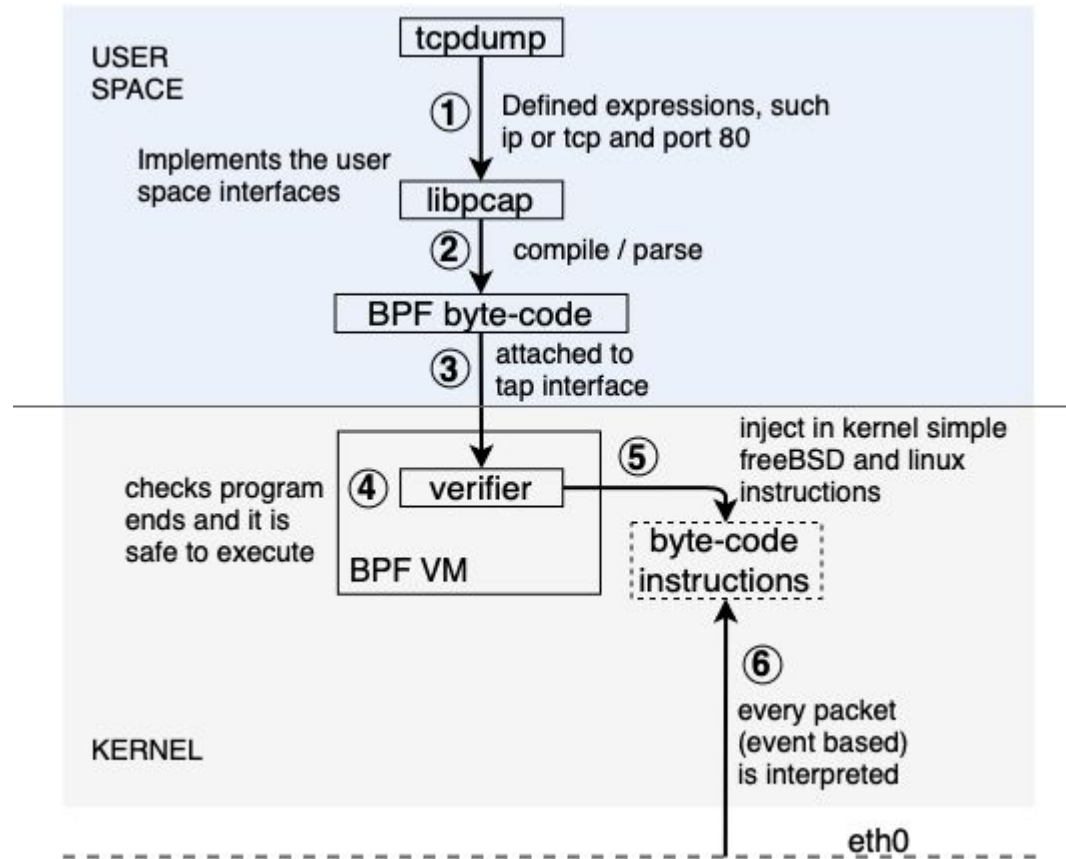


KubeCon



CloudNativeCon

Europe 2019



```
$ tcpdump -p -ni en0 -d "ip or tcp and port 80"
(000) ldh      [12]
(001) jeq      #0x800          jt 2    jf 12
(002) ldb      [23]
(003) jeq      #0x6           jt 4    jf 12
(004) ldh      [20]
(005) jset     #0x1fff        jt 12   jf 6
(006) ldx      4*([14]&0xf)
(007) ldh      [x + 14]
(008) jeq      #0x50          jt 11   jf 9
(009) ldh      [x + 16]
(010) jeq      #0x50          jt 11   jf 12
(011) ret      #262144
(012) ret      #0
```

eBPF: dynamic Linux kernel

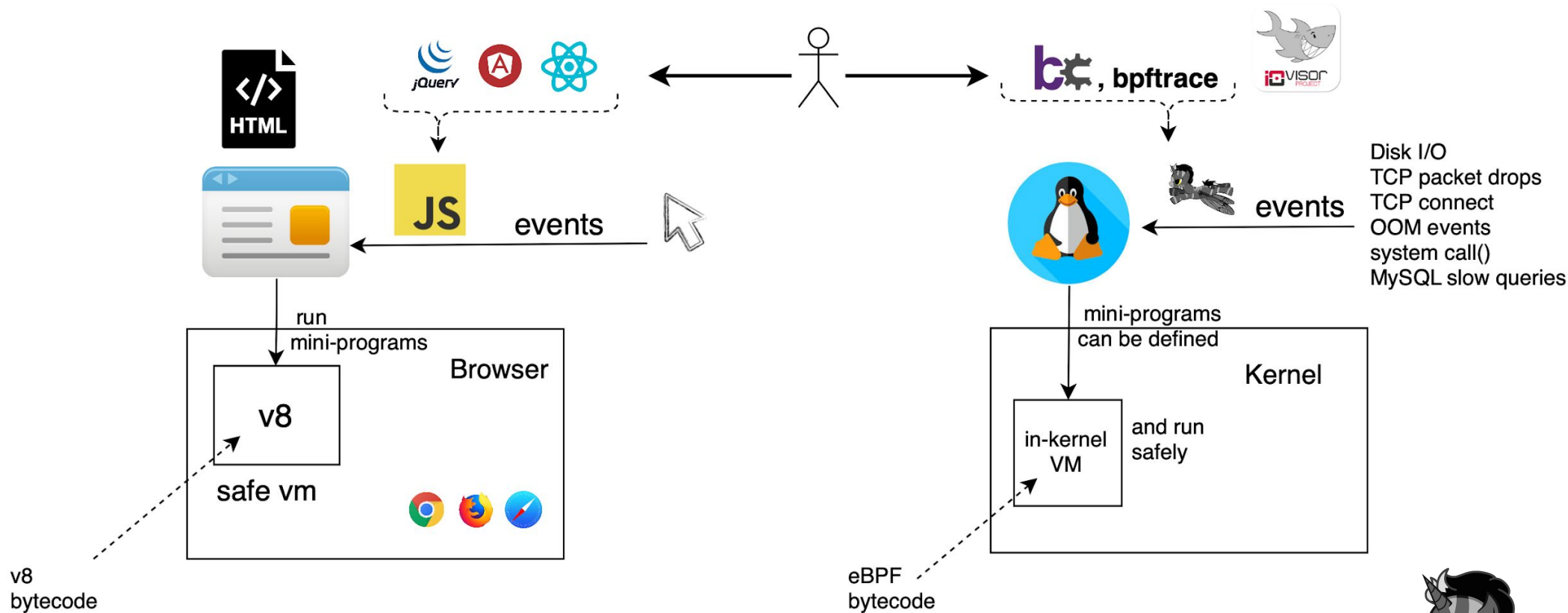


KubeCon



CloudNativeCon

Europe 2019



Inspired on Brendan Gregg's Blog example

Writing eBPF programs



KubeCon



CloudNativeCon

Europe 2019

```
#define ETH_LEN 14

struct dns_hdr_t
{
    uint16_t id;
    uint16_t flags;
    uint16_t qdcount;
    uint16_t ancount;
    uint16_t nscount;
    uint16_t arcount;
} BPF_PACKET_HEADER;

struct dns_query_flags_t
{
    uint16_t qtype;
    uint16_t qclass;
} BPF_PACKET_HEADER;
```

BPF bytecode	Brutal
C	Hard
perf	Hard
bcc	Moderate
bpfftrace	Easy
ply	Easy

Used in prod:
Netflix, Facebook



```
(000) ldh    [12]
(001) jeq    #0x800
jt 2     jf 5
(002) ldb    [23]
(003) jeq    #0x11
jt 4     jf 5
(004) ret    #65535
(005) ret    #0
```

one-liners

Syscall counts by process

```
bpfftrace -e 'tracepoint:raw_syscalls:sys_enter { @[comm] = count(); }'
```

Testing eBPF



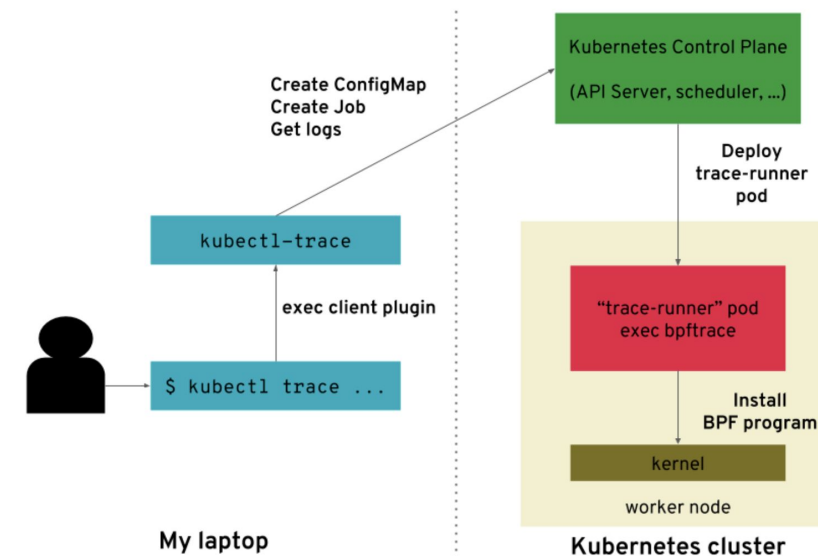
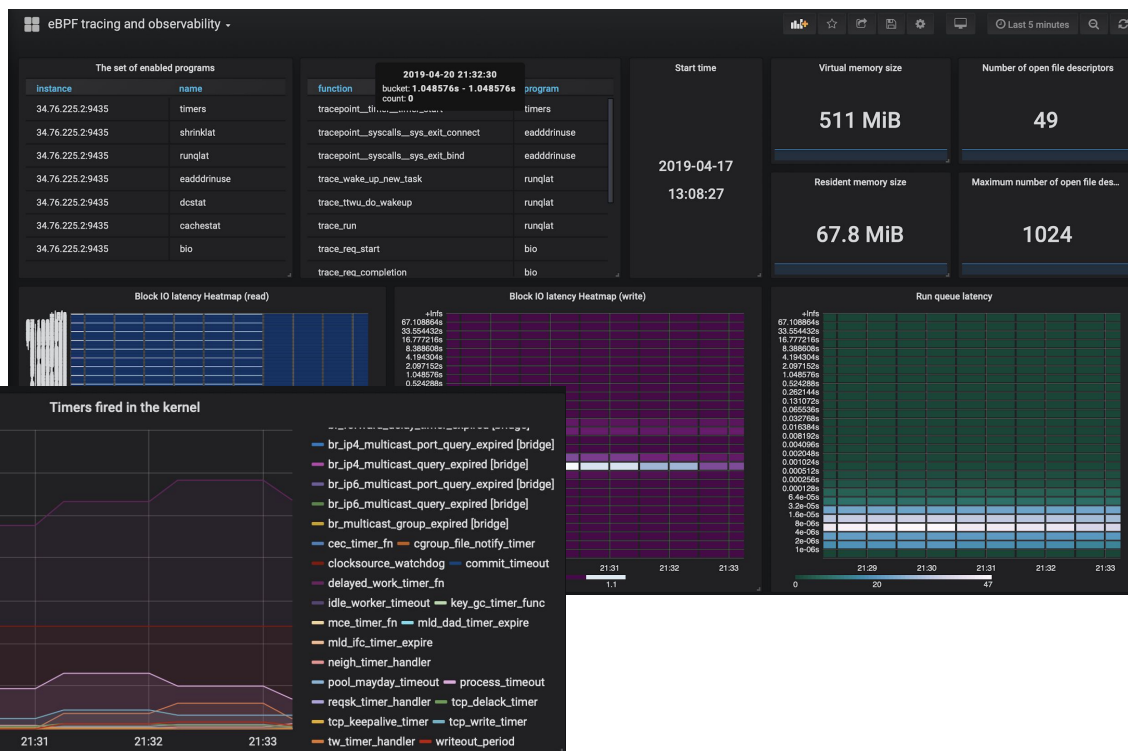
KubeCon



CloudNativeCon

Europe 2019

ebpf-exporter (bcc)



Real world examples



KubeCon



CloudNativeCon

Europe 2019

Use cases:

- Networking
- Firewalls
- Security
- Tracing
- Device Drivers



RED HAT BLOG

Introduction to eBPF in Red Hat Enterprise Linux 7

January 7, 2019 | Stanislav Kozina

[< Back to all posts](#)

Tags: [Platform](#)

The recent release of Red Hat Enterprise Linux 7.6 enables extended Berkeley Packet Filter (eBPF) in-kernel virtual machine which can be used for system tracing. In this blog we introduce the basic concept of this technology and few example use cases. We also present some of the existing tooling built on top of eBPF.

SHARE

Search a



Netflix

Performance profiling and tracing



Sysdig

eBPF instrumentation for high performance system calls tracing



Weaveworks

Trace TCP events



AWS Firecracker

Using Seccomp BPF to restrict system calls.



Facebook

eBPF-based load balancer with DDoS



Cloudflare

DDoS and Observability



Cilium

Powerful and efficient networking, security and load-balancing at L3-L7.



Redhat

RHEL 7.6 enables extended eBPF in-kernel VM



KubeCon



CloudNativeCon

Europe 2019

 [@beatrizmrg](https://twitter.com/beatrizmrg), Beatriz Martínez Rubio