

bpfman

What is bpfman?

- A suite of tools for deploying, managing and observing eBPF
- Works on Linux
- Works on Kubernetes

What problems does bpfman solve?

- No privileges required for userspace apps/pods (no CAP_BPF)
 - Policing of CAP_BPF granted to apps/pods COMING SOON!
- Daemon-less design like podman, but for eBPF
- Easy packaging and signing of eBPF code using OCI images
- Declarative configuration on Kubernetes (Custom Resources)
- Observability of eBPF usage

No CAP_BPF?



CAP_BPF Policing!

COMING SOON!



Daemon-less (Linux Only)

- Like podman!
- Loads your eBPF code and exits avoiding keeping elevated privileges for a long time

Daemon(set)-less (K8s only)

COMING SOON!



OCI images

- bpfman build creates multi-arch manifests with containing the right eBPF bytecode for the target architecture
- Push images to any OCI-compatible registry
- Use cosign to sign your container images



- bpfman can verify signatures!
 - Finer-grained attestations

COMING SOON!

Declarative Configuration (K8s) - Kernel space

apiVersion: bpfman.io/v1alpha1

kind: ClusterBpfApplication

metadata:

labels:

app.kubernetes.io/name: my-app

name: my-app

spec:

nodeSelector: {}

byteCode:

image:

url: quay.io/example/example-ks:latest

imagePullPolicy: Always

programs:

- name: kprobe_counter
 - type: KProbe

kprobe:

links:

- function: try_to_wake_up
 - offset: 0
- name: tracepoint_kill_recorder

type: TracePoint

tracepoint:

links:

- name: syscalls/sys_enter_kill

Declarative Configuration (K8s) - User space

| apiVersion: apps/v1 | volumeMounts: |
|--|--|
| kind: DaemonSet | - name: my-app-maps |
| metadata: | mountPath: /run/app/maps |
| name: my-app-ds | readOnly: true |
| spec: | volumes: |
| template: | - name: my-app-mapps |
| spec: | csi: |
| containers: | driver: csi.bpfman.io |
| - name: my-app | volumeAttributes: |
| image: quay.io/example/example-us:latest | csi.bpfman.io/program: my_app |
| imagePullPolicy: Always | csi.bpfman.io/maps: kprobe stats man tracepoint stats man |
| capabilities: | |
| drop: [ALL] | |

Observability!

Deployment via our operator COMING SOON!

| 😑 Home > Dashboards > eBPF Subsystem Metrics 🟠 🤹 | | | | | | | | | |
|--|---------------|-------|------------------------|---------------------|------------|-----------|----------------------------|----------------|--|
| Programs | | | Maps | | | | Links | | |
| | 21 | | | 5 | | | 3 | | |
| | | | | Ŭ | | | Ŭ | | |
| Loaded Program Info | | | | | | | | : | |
| name | type | | | load_time | map_ids | | tag | gpl_compatible | |
| | cgroup_device | 1555 | | 2023-08-24 12:04:12 | 0 | | 4114258343915946848 | false | |
| dump_bpf_map | tracing | 316 | | | [65] | | 14284438553703352928 | true | |
| dump_bpf_prog | tracing | 317 | | | [65] | | | true | |
| restrict_filesy | Ism | 43595 | | 2023-11-20 14:06:08 | [24268] | | 12315268691123498641 | true | |
| Loaded Map Info | | | | | | | | | |
| id | name | type | | key_size | value_size | | max_entries | flags | |
| 22 | cgroup_hash | 13 | | | 4 | | 2048 | 0 | |
| 22977 | xdp_stats_map | 6 | | 4 | 16 | | 5 | 0 | |
| 24268 | cgroup_hash | 13 | | | 4 | | 2048 | 0 | |
| 24551 | | 1 | | 8 | 4 | | 64 | 0 | |
| Loaded Link Info | | | | | | | | | |
| prog_id | | | id | | | type | | | |
| 43595 | | 12311 | | | | 2 | | | |
| 316 | | 5 | | | 4 | | | | |
| 317 | | | 6 | | | 4 | | | |
| Verified Instructions Total | | | Translated Bytes Total | | | Total Mem | ory Locked by Bpf Programs | | |
| 16 KiB | | | 22 10 | | | 384 KiB — | | | |
| 12 KiB | | | | | | 320 KiB | | | |
| | | | 24 KIB | | | 256 KIB | | | |
| | | | 16 KIB | | | 128 KiB | | | |
| 4 KIB | | | 8 KiB | | | 64 KiB | | | |
| 10:05 | 10:10 10:15 | | 10:05 | 10:10 | 10:15 | | 0:05 10:10 | 10:15 | |

Want to learn more?



Read the docs!

Get notified of project updates



Sign up to our newsletter!

Want to contribute?



Earn a sticker!

Proudly written in Rust

Interested to learn more about CNCF projects written in Rust?

You might be interested in:







