bpfman CFPs Tracking document

Kubecon NA 2024 Materials

https://events.linuxfoundation.org/kubecon-cloudnativecon-north-america/
Event: November 12-15, 2024 - Salt Lake City, Utah
CFP Due: **Sunday, June 9 at 11:59 pm Mountain Daylight Time (UTC-6)**

- Level: (Any, Beginner, Intermediate, Advanced) Intermediate
- Session Format: Tutorial (90 mins - 1-5 speakers)
    - End to end writing a basic eBPF program with current best practices and packaging + deploying to kubernetes with bpfman.

Title:

- How anyone can build, package, and deploy eBPF applications in Kubernetes with bpfman
- ???

Track:
(https://events.linuxfoundation.org/kubecon-cloudnativecon-north-america/program/cfp/#suggested-topics)

- Emerging + Advanced?
- Connectivity?

Session format:
Session Presentation (35 minutes - 1-2 speakers)

Level: (Any, Beginner, Intermediate, Advanced)

- Intermediate?

Description (1000 characters):

What is the talk ABOUT?

- How to write a basic eBPF program with current best practices (CORE, organization, etc)
- How to build + package the program (with multi-arch support) in a signed OCI image with bpfman

- How to deploy on K8s by writing application manifests (daemonset etc for the userspace portion)
- How to write an `*Program` manifest to describe the eBPF program
- How to debug + ensure the application is functioning

WHO is this for?

- Everyone (hook is all devs can use eBPF in kubernetes)

Why should they attend it? What are you PROMISING them to learn, what they get in exchange?

- Folks will leave this tutorial with full stack knowledge on how to deploy a low level kernel feature in a high level container orchestration platform.

Are you wanting to install an eBPF program on a Kubernetes cluster but don't know where to begin? We are here to help! Whether you are starting from scratch or already managing multiple eBPF programs through your application, bpfman integration can make lifecycle management much easier. bpfman is a CNCF candidate project that can help you develop and deploy your eBPF program. It doesn't replace existing tools like Cillium bpf2go and libbpf, but works alongside them, enabling you to deploy your eBPF program in Kubernetes without escalated pod privileges.

In this talk, we will help you write an introductory eBPF program, pointing out some of the best practices to use in the process. Next, teach you how to build and package the program in a signed OCI image which can then deploy it in a Kubernetes cluster. We will also show you how to ensure that the program is running and some tricks and tools to use to help debug when it's not.

So if you are interested in using eBPF, come listen to what we have to say!
(1004)

Benefits to Ecosystem (No character limit):

As the number of eBPF programs deployed in Kubernetes continues to grow, so does the inherent risk for the stability, security, and overall health of clusters far and wide. Many CNCF projects such as Cilium, Datadog, Calico and Pixie, have already pioneered the use of this exciting new technology. However, when it comes to the deployment and management of their eBPF programs, they often operate in complete silos. Within these silos is a proliferation of privileged daemonsets, overlapping functionality, and figurative “stomping of each-others toes” when attaching programs to various kernel hook-points. In order to mitigate the continued proliferation of these issues we need to spread awareness throughout our community and start working on effective solutions. The bpfman project is dedicated to helping solve some of these problems by providing an easy and safe way for everyone to load eBPF programs into a cluster. With this talk we hope to take viewers through the end to end journey of writing eBPF programs and deploying them to Kubernetes with the help of bpfman. bpfman is in the queue for CNCF and should be voted on in the June 11 meeting. So we hope to be a CNCF Project by the time KubeCon rolls around.

RustConf 2024 Materials: SUBMITTED

https://rustconf.com/
Event: September 10-13, 2024 - Montreal
CFP Due: **April 25 @ midnight PDT: Call-for-Proposals Closes**

- The bpfman Design journey?
    - Damon Architecture + Struct design
    - In memory database conversion (Sled-DB)
    - Damonless Architecture, library conversion
- Rust + Ebpf + Kubernetes?
    - A focus on all the buzzwords and how rust played into it?
- Rust Zero to Hero a design arc involving

Title: Three Rewrites Later: Following the Design Arc of bpfman, an eBPF Manager

Description (Enter a 1-2 sentence description of your session. You'll have the chance to share more details below.):

Bpfman is a Rust-based eBPF management library which started its life as a long-running daemon. Join us as we dive into the transition, analyze the pros and cons of various rust design strategies, and explain why we settled on the architecture used today.

Details (Include any pertinent details such as outlines, outcomes or intended audience.):

The open source project Aya paved the way for Rustaceans to have native access to the linux kernel's eBPF subsystem while also allowing for the creation of eBPF programs in Rust. bpfman builds on the Aya project to provide users a set of higher-level abstractions for managing eBPF programs and also tackles the integration of the technology with Kubernetes. The project started its life organized as a simple binary crate and was initially hacked together with little focus on the "correct" design or code layout. As time progressed and complexity increased, the internal organization of the project became significantly more Rusty. A rigid line was drawn between the core internal API and the external GRPC-based API which allowed for strict structure-type safety and a more organized project with better overall readability. Following the initial overhaul, a new use case of on-disk memory persistence pushed the community to integrate with a Rust-based in-memory database, Sled, again resulting in a major internal rewrite which fundamentally changed how internal state and typing was handled. At the same time, security concerns regarding the use of a long-running daemon resulted in a final design overhaul to convert the simple binary crate into a library crate, forcing the community to reanalyze the proliferation of asynchronous code and change the project's high-level code structure.

Whether you are a seasoned Rust programmer or a newbie, this talk will help you by providing an overview of Rust basics, while also reflecting on how we could have done better earlier in the project's lifecycle to avoid all those code rewrites.

Pitch (Explain why this session should be considered and what makes you qualified to present on the topic. This will be visible during the anonymous review, so refrain from personally identifying information as much as possible.)

Two years ago I was a Kubernetes-focused golang programmer on the cusp of entering into the wide world of Rust. Little did I know those first few months would result in a journey involving one heck of a struggle against the Rust compiler, with many late nights spent trying to understand topics ranging from simple lifetimes, project layout, and even a foray into tokio and asynchronous Rust programming. Today I am a maintainer on both the bpfman and Aya projects, and while I still have much to learn about Rust-Lang, I have become a true Rustacean, with a passion for Rust which outweighs any other programming language. This session should be considered because it follows a unique design journey with Rust-Lang in a real-world open source project and is given from the perspective of a maintainer with a passion for the language.

eBPF Summit 2023 Materials

Session Type: 20 minute presentation

Potential areas of focus:

- Benefits to the application developer
    - General overview
    - Bpfd simplifies the development of Kubernetes apps
        - Solves problems you'd need to solve yourself
        - Developer productivity
    - Solves security problems
    - Helps with multiprogram cooperation
        - Works great if everyone uses it.
        - If they don't then spell out what non-participating programs need to do to work together.
    - Visibility helps with debugging inevitable problems.
    - Could we do a Blixt use case by then?
- Benefits to system administrators
    - Ability to deploy multiple eBPF-based apps
    - Security
    - Visibility

Title: Quelling the storm of eBPF in Kubernetes with bpfd”

Level: Intermediate

Abstract:

The use of eBPF in Kubernetes applications has been growing rapidly due to the revolutionary capabilities it enables as demonstrated by CNCF projects such as Cilium, Datadog, Calico and Pixie. However, eBPF in Kubernetes also poses a number of new challenges for developers and administrators alike. These challenges include program lifecycle management issues, the widespread use of privileged pods, the lack of bpf-subsystem visibility, and problems with program cooperation. bpfd (https://bpfd.dev/) is a new open-source project designed to simplify the deployment and management of eBPF programs, ultimately providing a secure and visible way for everyone in Kubernetes to utilize this exciting new technology.

In this talk we will give an overview of bpfd, including some of the exciting new features from our most recent release. We will then give a live demo of bpfd to show how it solves some of these common challenges.

Additional links:
https://bpfd.dev/
Bpfd Presented to sig-node: https://youtu.be/H9vnLgvTLvo?si=y6e3R9imvPX2Ne_2&t=1506

Kubecon NA 2023 Materials

Track: **??**
Notes:

- Focus on multi tiered bpfd value add
    - Program co-operation
    - Built in Security
    - Observability
    - Fine Grained versioning

BPFD -> https://github.com/bpfd-dev/bpfd

Kubecon Abstract/ Contributor Summit?: 1000 Character Limit

Title Options: (REVIEWERS PLEASE LEAVE RECs)

~~- Preventing eBPF Murder Mysteries with bpfd~~
~~- **Slay eBPF + Kuberentes dragons like a true knight with bpfd**~~
~~- Bpfd: a novel way to deploy and manage eBPF programs~~
~~- **Bpfd the key to escaping eBPF + Kubernetes dungeons**~~
~~- **The key to escaping the eBPF dungeons deep within your Kubernetes cluster**~~
~~- Quelling the storm of eBPF in Kubernetes with Bpfd~~
~~- **Here be dragons: eBPF on Kubernetes**~~
- **Survive eBPF deployment with bpfd**

Developing eBPF programs is challenging within itself, but what about deploying those applications in a shared environment such as a Kubernetes cluster? As more applications begin to utilize eBPF, the lack of cooperation, wide use of critical permissions and zero cluster-wide visibility can lead to a “perfect storm” of issues which are almost impossible to diagnose.

In this talk we’ll discuss our journey to achieve a better standard for deploying and managing the lifecycle of eBPF programs on Kubernetes. We will demonstrate how bpfd, a system daemon and Kubernetes operator for managing eBPF programs can be used to deploy, observe and secure those programs on all your cluster nodes. Attendees will see how things can go wrong when multiple networking programs are contending for the same resources and learn how bpfd can be used to diagnose and fix these types of issues.

Benefits to ecosystem

As the number of eBPF programs deployed in Kubernetes continues to grow, so does the inherent risk for the stability, security, and overall health of clusters far and wide. Many CNCF projects such as Cilium, Datadog, Calico and Pixie, have already pioneered the use of this exciting new technology. However, when it comes to the deployment and management of their eBPF programs, they often operate in complete silos. Within these silos is a proliferation of privileged daemonsets, overlapping functionality, and figurative “stomping of each-others toes” when attaching programs to various kernel hook-points. In order to mitigate the continued proliferation of these issues we need to spread awareness throughout our community and start working on effective solutions. The bpfd project is dedicated to helping solve some of these problems by providing an easy and safe way for everyone to load eBPF programs into a cluster. With this talk we hope to share our project and experiences to help spread awareness while also growing a community that is focused on tackling the problems associated with deploying eBPF at scale on Kubernetes.