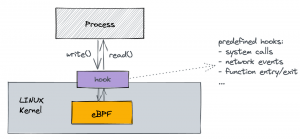
eBPF DOCUMENTAZIONE

**What is eBPF?**

[eBPF](https://ebpf.io/what-is-ebpf) stands for extended Berkeley Packet Filter. The Linux kernel has been around for a long time, however, it is not easy to modify or extend the kernel unless you know how to patch it. If you are familiar with [Kubernetes’s custom resources](https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/) or Envoy filters, you understand how important it is to build extensions based on your specific scenarios. What eBPF provides to the Linux kernel is the extensibility to enable developers to program the Linux kernel to quickly build intelligent or feature-rich functions based on their business needs. eBPF programs to the Linux kernel are similar to what web assembly modules are to Envoy. They allow developers to extend the kernel easily and run their eBPF code as sandboxed programs in the kernel without changing the kernel source code or loading kernel modules.



PROGRAMMING eBPF PROGRAM WITH BEE

BumbleBee helps to build, run and distribute eBPF programs using OCI images. It allows you to focus on writing eBPF code, while taking care of the user space components - automatically exposing your data as metrics or logs.

Bumblebee simplifies building eBPF tools and allows you to package, distribute, and run them anywhere.Just focus on the eBPF portion of your code and BumbleBee automates away the boilerpart, including the userspace code.

Geeting the eBPF tool chain “just right” is hard. BumbleBee automates the build process and lets you to focus on the code. BumbleBee packages your eBPF code as an OCI image so you can distribute it across your infrastructure.

With BumbleBee, you focus on your eBPF code and run it anywhere. BumbleBee also builds the userspace code and can expose the eBPF maps as logs, metrics, and histograms. BumbleBee leverages BTP introspection to know what types to display.

1. First thing to do is to choose the programming language.
2. Now that we have selected the language to use, we will be prompted to select the type of program you want to create. As eBPF enables you to write programs that can hook into essentially any kernel functionality, there are several "types" of programs you can create.
3. Next you will be asked for the type of global map you would like to use. Maps are the instrument through which eBPF user space, and kernel space programs are able to communicate with each other.
4. After deciding on a map type, you will be asked to decide on an output format. Normally developing eBPF applications requires writing user space and kernel space code (with bee you only need to develop the kernel space code, and then bee can automatically handle and output the data from your eBPF maps).

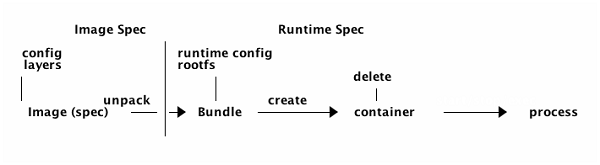
OCI SPECIFICATION -> Principio del CompileOnce-RunAnywhere

<https://alibaba-cloud.medium.com/open-container-initiative-oci-specifications-375b96658f55>

OCI (Open Container Initiative) is an industry collaborated effort to define open containers specifications regarding container image format and runtime.

OCI has two specs, an Image spec and a Runtime spec.

The diagram illustrates what they cover and how they interact.



An OCI image will be downloaded from somewhere and then it will be unpacked into an OCI Runtime filesystem bundle. From that point, the OCI Runtime Bundle will be run by an OCI Runtime. The Runtime Specification defines how to run a “filesystem bundle”.