Linux Plumbers Conference 2024

Contribution ID: **176** Type: **not specified**

Immutable process tags for container tracking

Thursday 19 September 2024 15:35 (15 minutes)

Containers are a user space fiction, there is no single container concept within the Linux kernel and what set of components constitutes a container isn't something we expect everyone to agree on any time soon (if ever).

That said, we've seen many ask for ways to easily figure out whether a process belongs to a container, if so, which one, who/what's responsible for it, …

Some of the existing tools like ps/top rely on some clever parsing of the cgroups used by the process to figure out what container they may belong to. Others walk the entire process tree and keep track of what spawned a particular process tree.

But again, there is no guarantee that a particular container implementation will use cgroups, or will use an easily detectable parent process for the container's tree.

A few years ago, there was a proposal by José Bollo to introduce the concept of process tags (ptags) as a LSM.

While this apparently didn't really go anywhere, the general idea is interesting and would likely be a good generic solution to solve this recurring problem.

Effectively having support for key/value storage of data as part of a process with a number of restrictions on top of it to make it safe and useful:
- Tags are copied to children at clone time
- A tag can never be removed
- The value of a tag can never be altered
- Tags can only be set by root (of namespace) or the process' owner
- Tags are owned by whoever created them
- The number of tags and the length of their value is restricted

This session will attempt to answer:
- Does this solve the usual "what container is this?" question
- Any other use cases for this mechanism?
- Is there something already present today we could rely on instead?
- Are there specific concerns around security or performance of something like this?

Primary author: GRABER, Stéphane (Zabbly)

Presenter: GRABER, Stéphane (Zabbly)

Session Classification: Containers and checkpoint/restore MC

Track Classification: Containers and checkpoint/restore MC