Agenda

Background
Use cases
Technical Details
Future
Checkpoint/Restore in Userspace

CRIU
Multiple Integrations Exist
Container Live Migration

OpenVZ
Container Live Migration

Borg
Container Live Migration

LXC/LXD
Container Live Migration

Docker
Container Live Migration

Podman
Container Live Migration

CRI-O
Container Live Migration

Kubernetes

https://github.com/kubernetes/kubernetes/issues/3949
Forensic Container Checkpointing

https://github.com/kubernetes/enhancements/pull/3264
https://github.com/kubernetes/kubernetes/pull/104907
https://kubernetes.io/blog/2022/12/05/forensic-container-checkpointing-alpha/
Use Cases
Reboot and Save State
Container
Quick Startup
Container Live Migration
Source

Container

Destination
Spot instances
CRIU
First Step: Checkpointing
Seize Process Using

ptrace()
Collect Details From

/proc/<PID>/*
Parasite Code
Parasite Code

Injected into the process
Parasite Code

Daemon waiting for commands
Parasite Code

Removed after usage
Original
Process
Code
To Be
Checkpointed
Checkpointing Finished

All relevant information written
Second/Last Step: Restoring
Read Checkpoint Images
clone() For Each PID/TID

clon3() with Linux 5.5
CRIU Morphs Itself

Open and position file descriptors
CRIU Morphs Itself
Map memory pages
CRIU Morphs Itself

Load security settings
CRIU Morphs Itself

Jump into restored process
What’s next?
kubectl checkpoint
Pod Checkpoint/Restore
kubectl migrate
Scheduler Integration
Checkpoint Image Standard

https://github.com/opencontainers/image-spec/issues/962
Summary

- CRIU can checkpoint and restore containers
- Integrated in different containers engines
- Used in production
- Reboot into new kernel without losing container state
- Start multiple copies
- Migrate running containers
- Spot instances
- Forensic container checkpointing (KEP #2008)
- Restore via `create` and `start`
Thank you