Container Migration with Kubernetes and CRIU

Adrian Reber
Principal Software Engineer

2021, May 14

https://people.redhat.com/~areber/2021-lpd.pdf

Agenda

Background

Use cases

Details

Demos

Future

Red Hat

Checkpoint/Restore in Userspace

CRIU

Red Hat

Definition:

Container Live Migration

Red Hat

Transfer Running Container

Red Hat

Serialize on Source System

Red Hat

Transfer to Destination System

Red Hat

Background

Red Hat

Kubernetes

Red Hat

Container

Red Hat

Container

namespaces

Red Hat

Container

cgroups

Red Hat

Container

seccomp

Red Hat

Container Runtime

`runc` or `crun`

Red Hat

Container Engine

`CRI-O` or `containerd` or `Podman`

Red Hat

Container Orchestration

Kubernetes

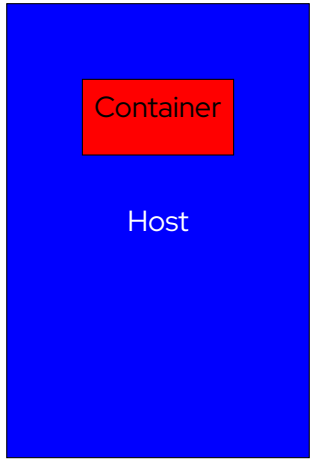
Red Hat

Use Cases

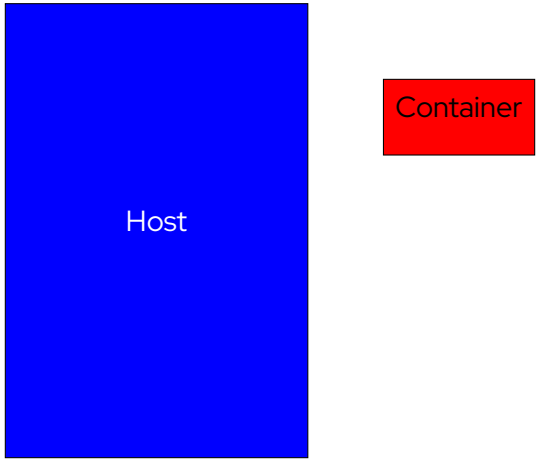
Red Hat

Reboot and Save State

Red Hat

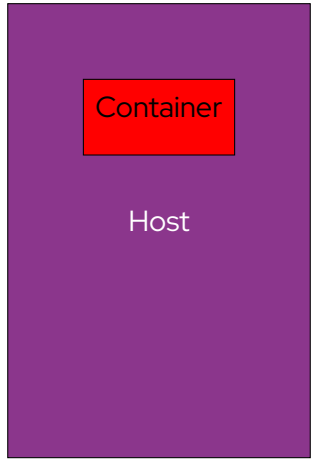


Red Hat



Red Hat

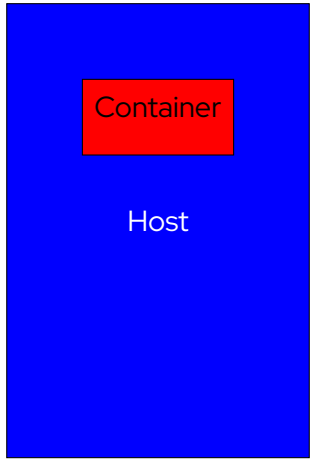
Red Hat



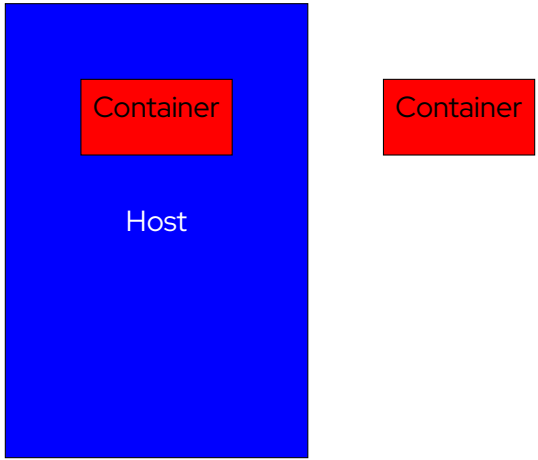
Red Hat

Quick Startup

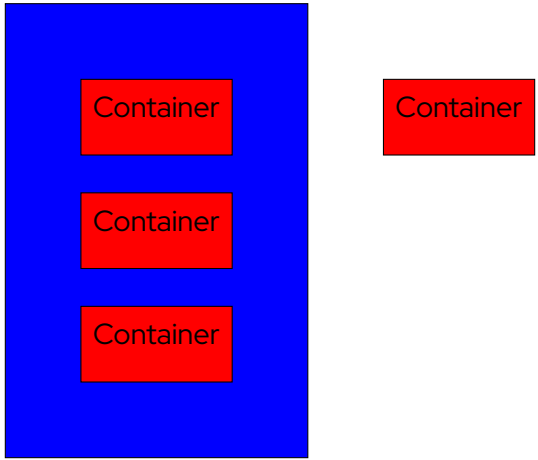
Red Hat



Red Hat



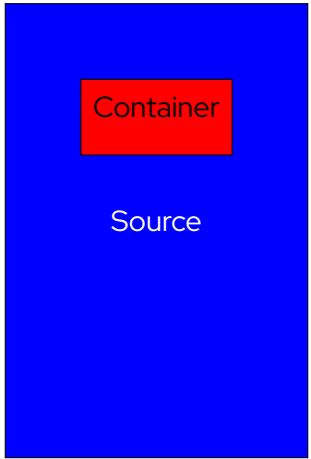
Red Hat

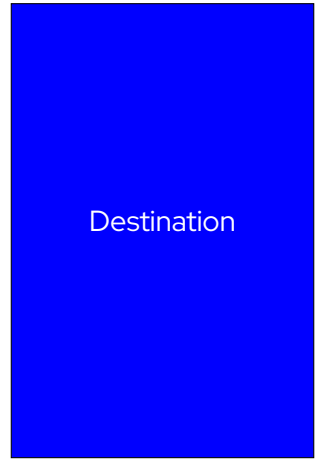


Red Hat

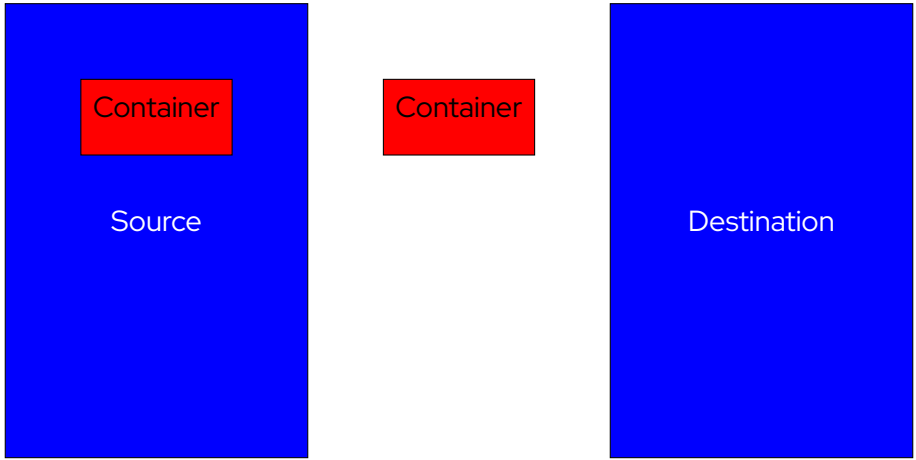
Container Live Migration

Red Hat

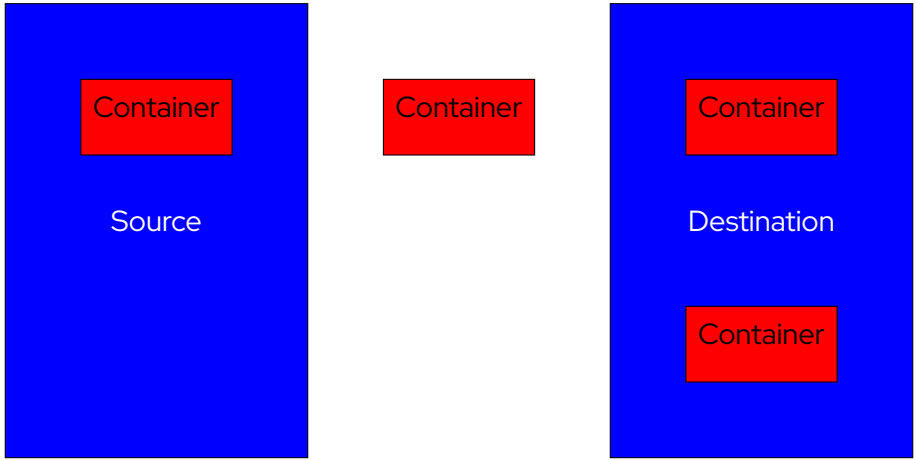




Red Hat



Red Hat


Container
Source
Container
Container
Destination
Container

Red Hat

CRIU

Red Hat

First Step: Checkpointing

Red Hat

Seize Process Using

`ptrace()`

Red Hat

Collect Details From

/proc/<PID>/*

Red Hat

Parasite Code

Red Hat

Parasite Code

Most favorite part

Red Hat

Parasite Code

And the craziest

Red Hat

Parasite Code

Injected into the process

Red Hat

Parasite Code

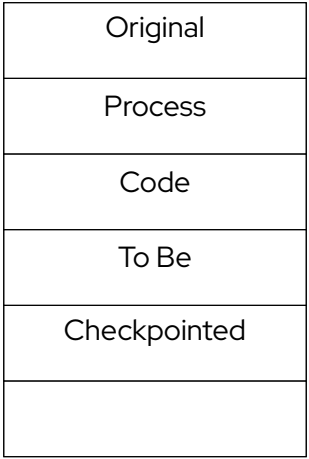
Daemon waiting for commands

Red Hat

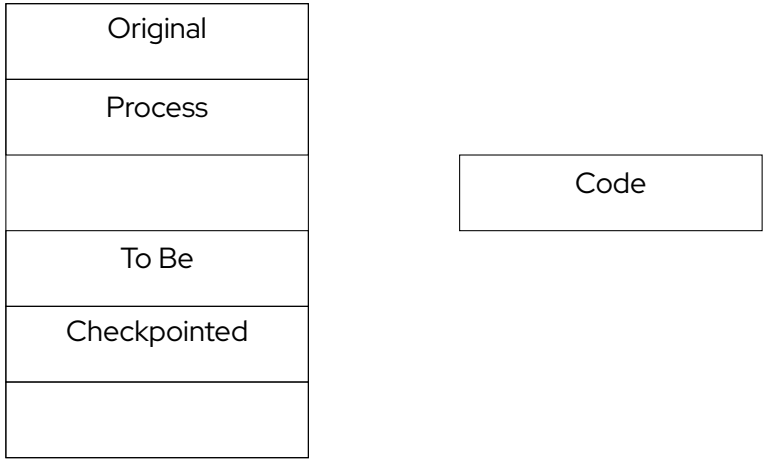
Parasite Code

Removed after usage

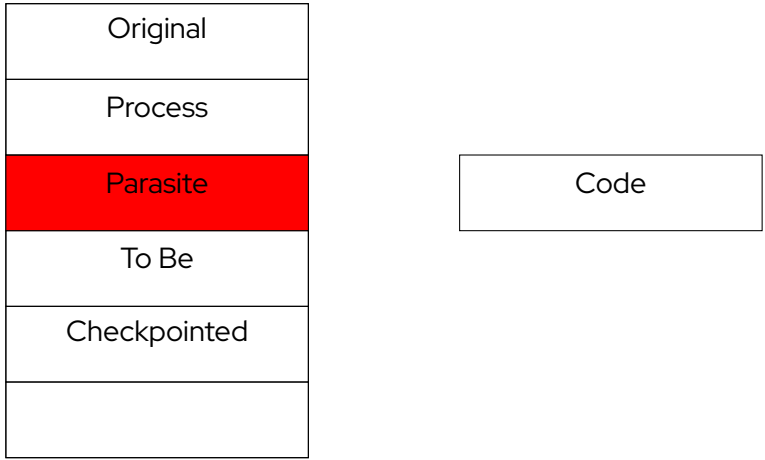
Red Hat



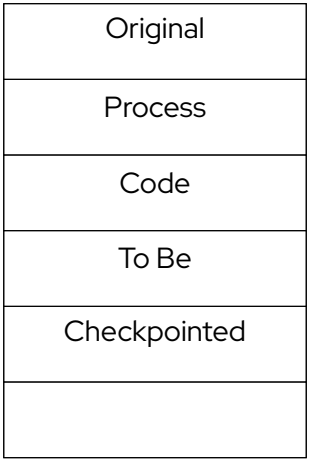
Red Hat



Red Hat



Red Hat



Red Hat

Checkpointing Finished

Red Hat

Checkpointing Finished

All relevant information written

Red Hat

Checkpointing Finished

Target process is killed

Red Hat

Checkpointing Finished

Or continues to run

Red Hat

Second/Last Step: Restoring

Red Hat

Read Checkpoint Images

Red Hat

`clone()` For Each PID/TID

`clone3()` with Linux 5.5

Red Hat

CRIU Morphs Itself

Open and position file descriptors

Red Hat

CRIU Morphs Itself

Map memory pages

Red Hat

CRIU Morphs Itself

Load security settings

Red Hat

CRIU Morphs Itself

Jump into restored process

Red Hat

```
# podman run --rm -d adrianreber/wildfly-hello
699f33eb7fecbc5bbb00400be0aa79c888dbc63a54cac7bd2eed836a57d8a68a
# podman inspect -l --format "{{.NetworkSettings.IPAddress}}"
10.88.0.247
# curl 10.88.0.247:8080/helloworld/
0
# curl 10.88.0.247:8080/helloworld/
1
# podman container checkpoint -l --export=/tmp/chkpt.tar.gz
699f33eb7fecbc5bbb00400be0aa79c888dbc63a54cac7bd2eed836a57d8a68a
# scp /tmp/chkpt.tar.gz rhel08:/tmp
```


Red Hat

```
# podman container restore --import=/tmp/chkpt.tar.gz
699f33eb7fecbc5bbb00400be0aa79c888dbc63a54cac7bd2eed836a57d8a68a
# podman inspect -l --format "{{.NetworkSettings.IPAddress}}"
10.88.0.247
# curl 10.88.0.247:8080/helloworld/
2
# curl 10.88.0.247:8080/helloworld/
3
```


Red Hat

```
# podman container restore --import=/tmp/chkpt.tar.gz -n hello1
d02feeec894d77f66cc82484fe77ae369396a85f6d05594dc156c21e685942dd
# podman container restore --import=/tmp/chkpt.tar.gz -n hello2
735efb4fee6961d3eee069beb28dde5cbc6fc46c1a32a43ecc993d04c02015b2
# podman inspect --format "{{.NetworkSettings.IPAddress}}" hello1
10.88.0.248
# podman inspect --format "{{.NetworkSettings.IPAddress}}" hello2
10.88.0.249
# curl 10.88.0.248:8080/helloworld/
2
# curl 10.88.0.249:8080/helloworld/
2
```


Red Hat

Future:

`kubectl migrate`

Red Hat

Future:

Non-root checkpoint/restore

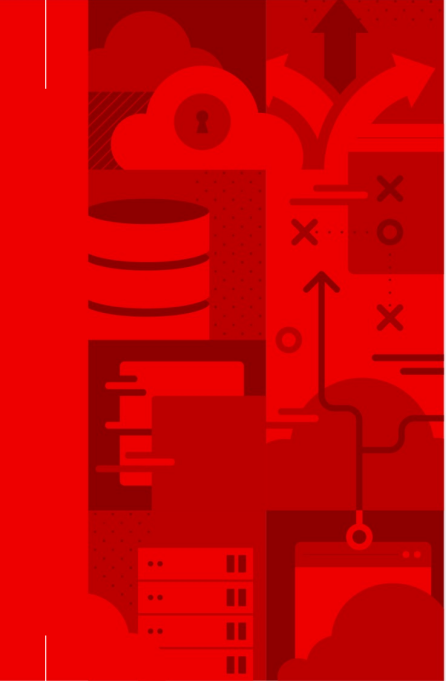
`CAP_CHECKPOINT_RESTORE`

Red Hat

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

Red Hat



Thank you

Red Hat