Checkpointing and Rollback-Recovery of Distributed Applications in Kubernetes

Europe Research Interest Group Meeting - April 13, 2023

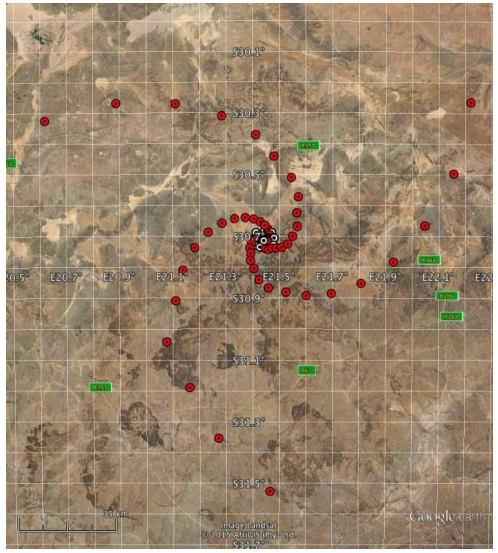
Radostin Stoyanov

DPhil Candidate, Scientific Computing Group

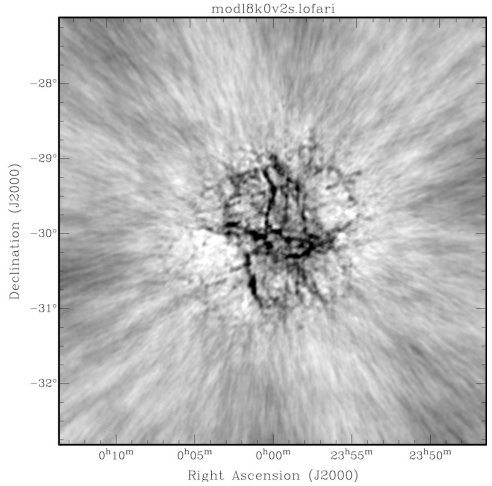
Supervisor: Prof. Wes Armour

Square Kilometer Array

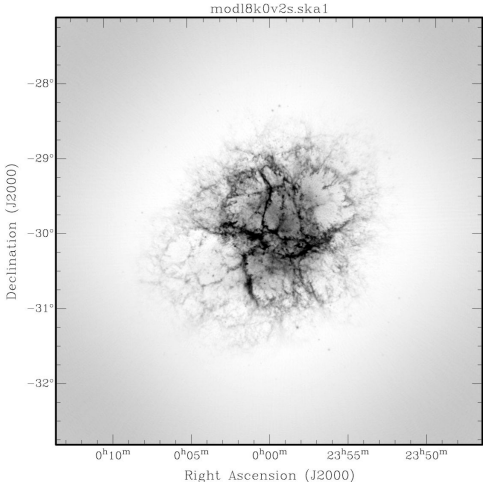
SKA1-Low (Australia)



SKA1-Mid (South Africa)



a) LOFAR-INTL[1] Snapshot



b) SKA-Low Snapshot

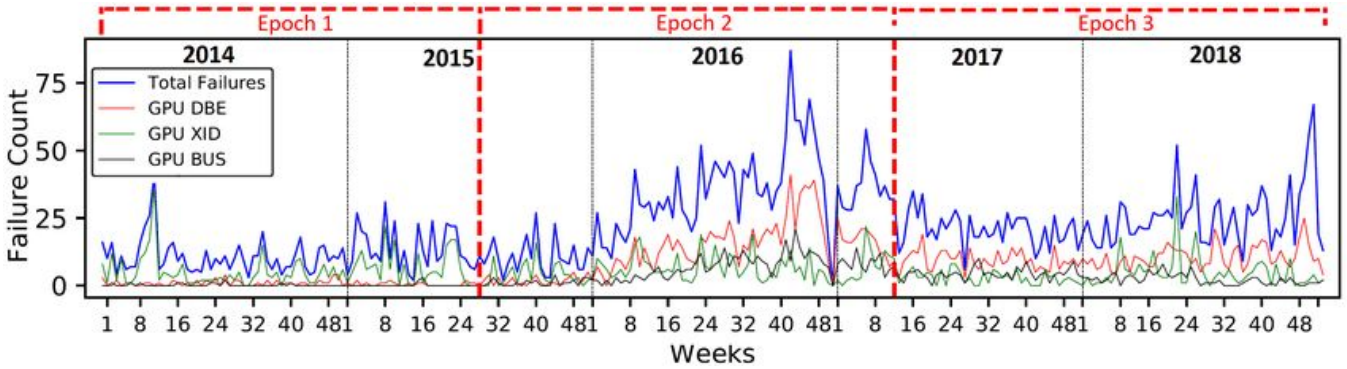
Philip Diamond. SKA Community Briefing. 2017.

1 LOFAR (LOw Frequency ARray) - a radio telescope working at the lowest frequencies accessible from Earth.

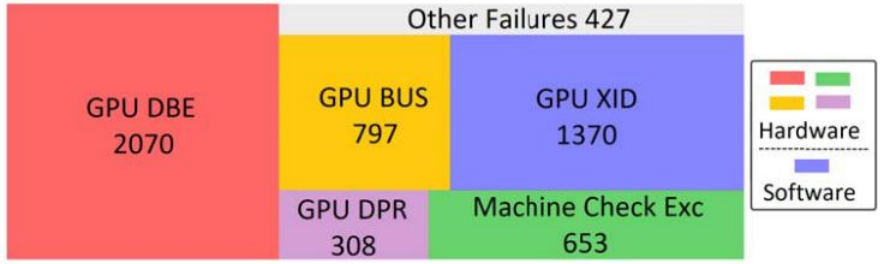
Five-Year Failure Record of a Leadership-Class Supercomputer

Titan supercomputer, a Cray XK7 system located at the Oak Ridge Leadership Computing Facility (OLCF)

- 18,688 nodes (total of 299,008 CPU cores)
- Combined elements provide 17.59 petaFLOPs
- Each node has:
  - 32 GB of main memory
  - AMD 16-core Opteron CPU
  - NVIDIA Tesla K20 GPU



Categorization of failures over a 5 year (2014 - 2018) reliability record:



1. GPU-related errors dominate the record
2. Most failures only affect a single node
3. Software failures bring down several nodes

Elvis Rojas et al. "Analyzing a Five-Year Failure Record of a Leadership-Class Supercomputer". International Symposium on Computer Architecture and High Performance Computing. 2019

Checkpoint/Restore in Userspace

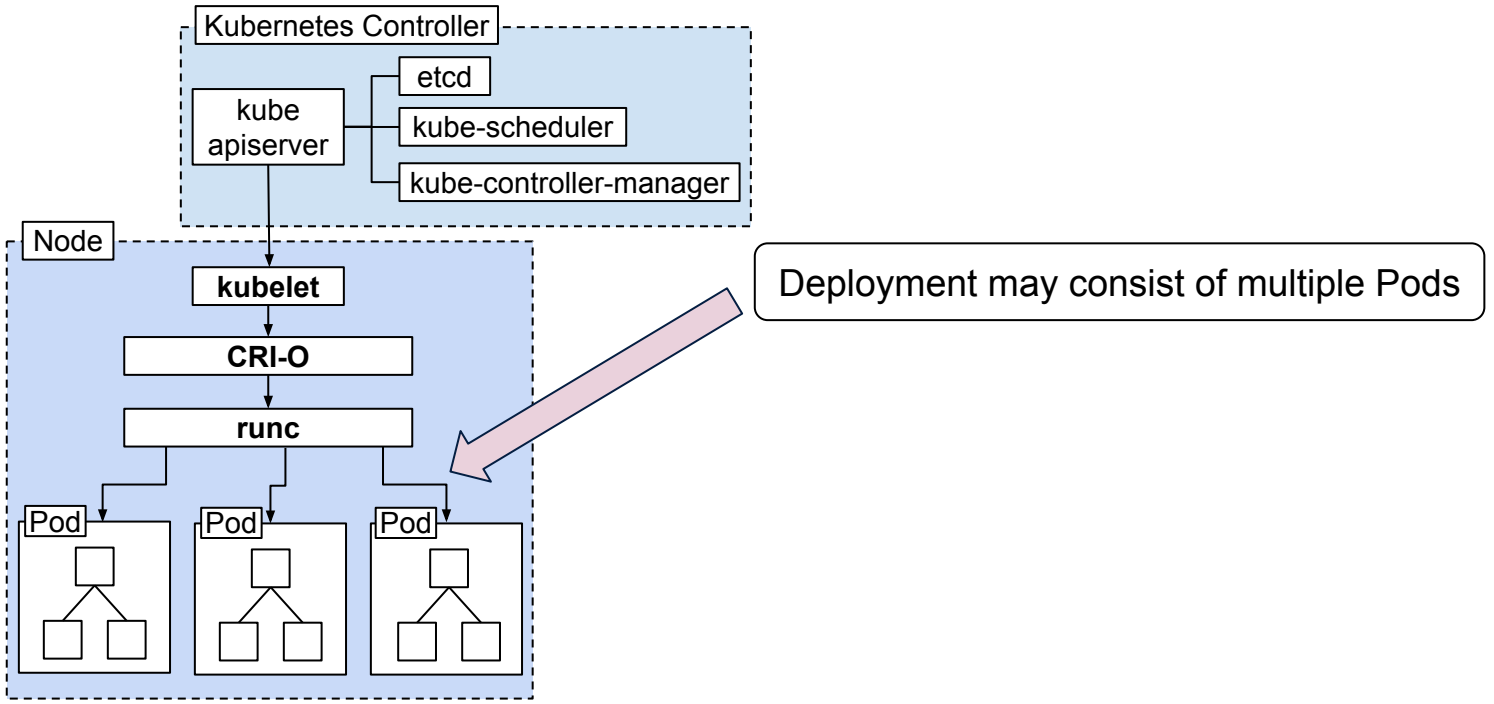
- Open-source tool for checkpoint/restore of Linux processes
- Entirely in userspace
- Integrated with:
  - runc / crun / youki
  - Docker / Podman / LXD
  - CRI-O / containerd
  - Kubernetes

Container Checkpointing in Kubernetes

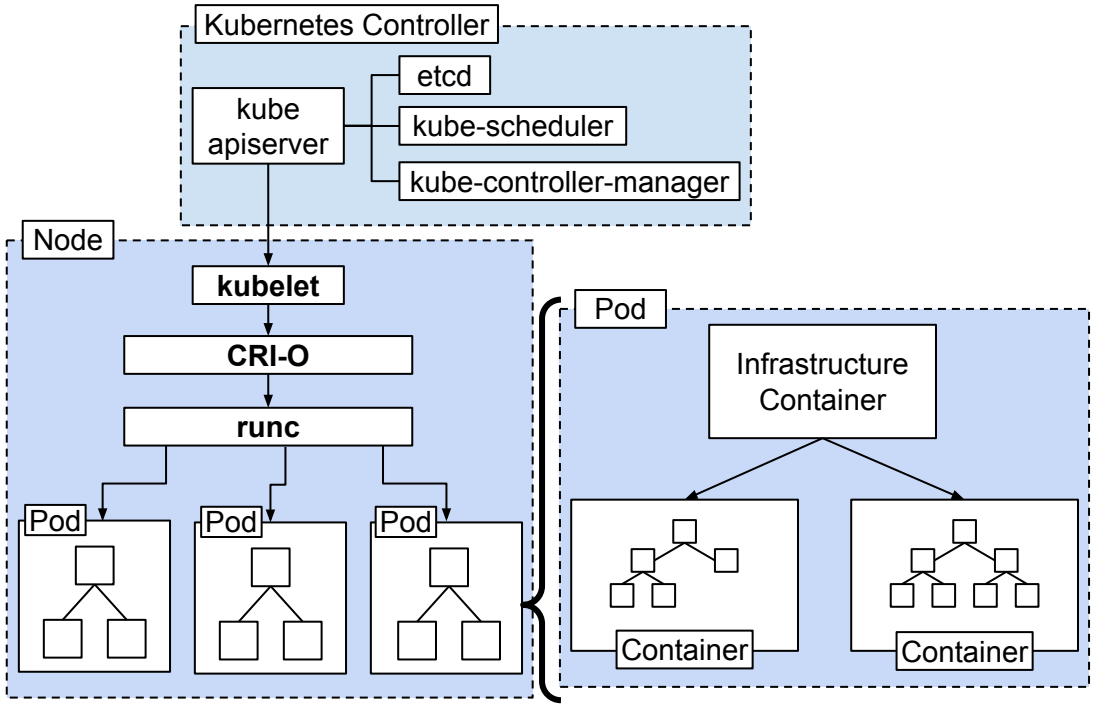
An alpha feature introduced with Kubernetes v1.25

https://kubernetes.io/blog/2022/12/05/forensic-container-checkpointing-alpha

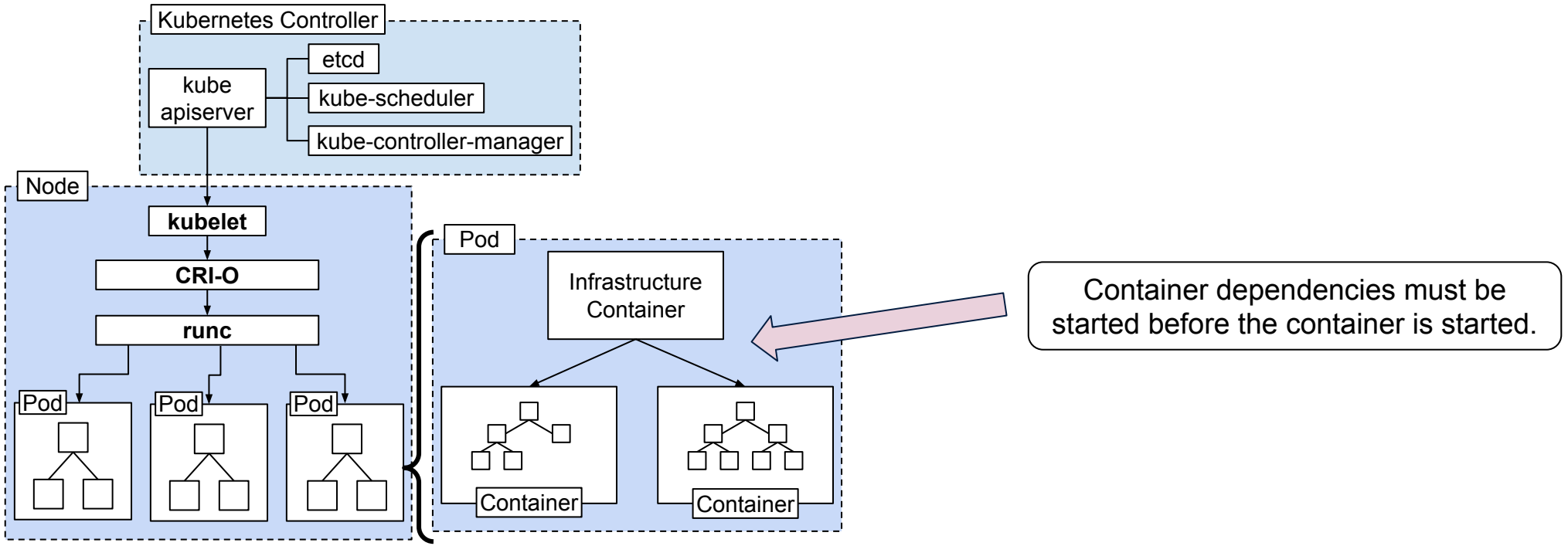
Container Checkpointing in Kubernetes



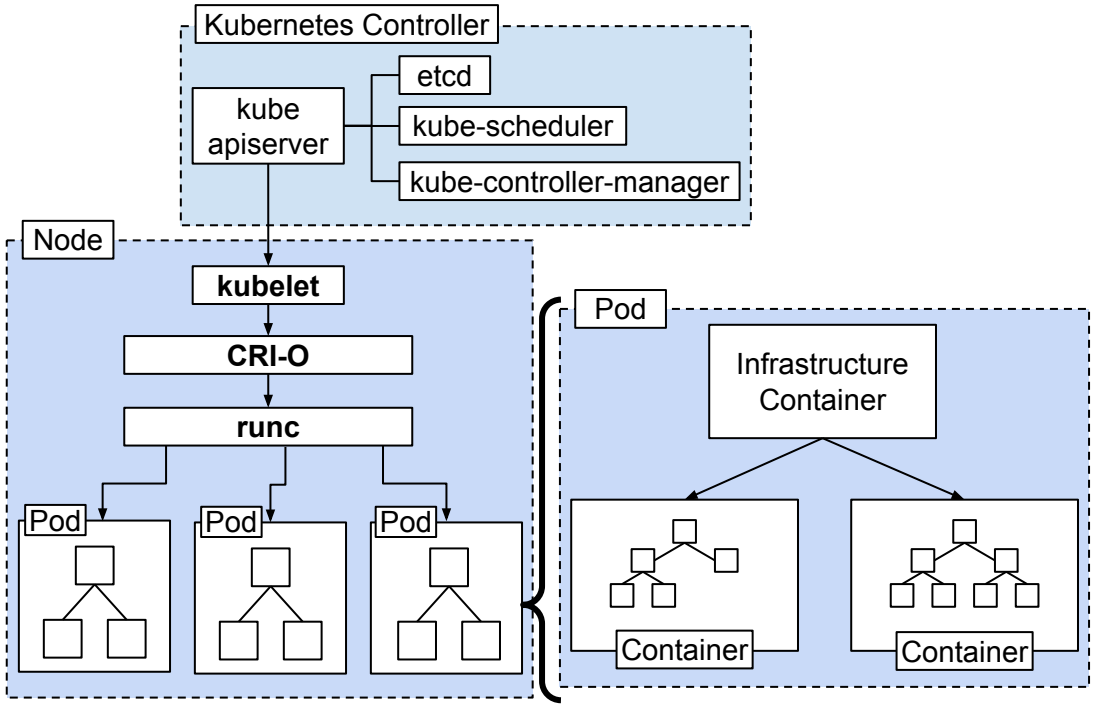
Container Checkpointing in Kubernetes



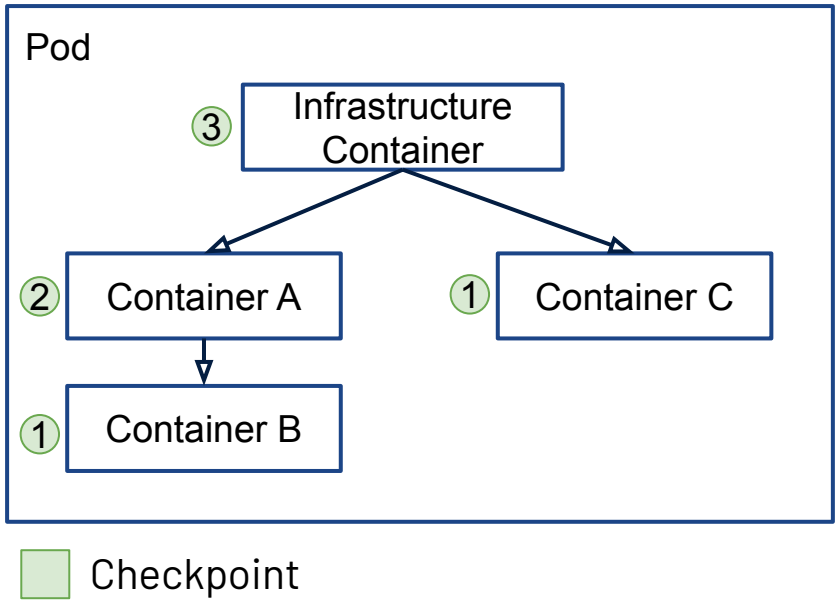
Container Checkpointing in Kubernetes



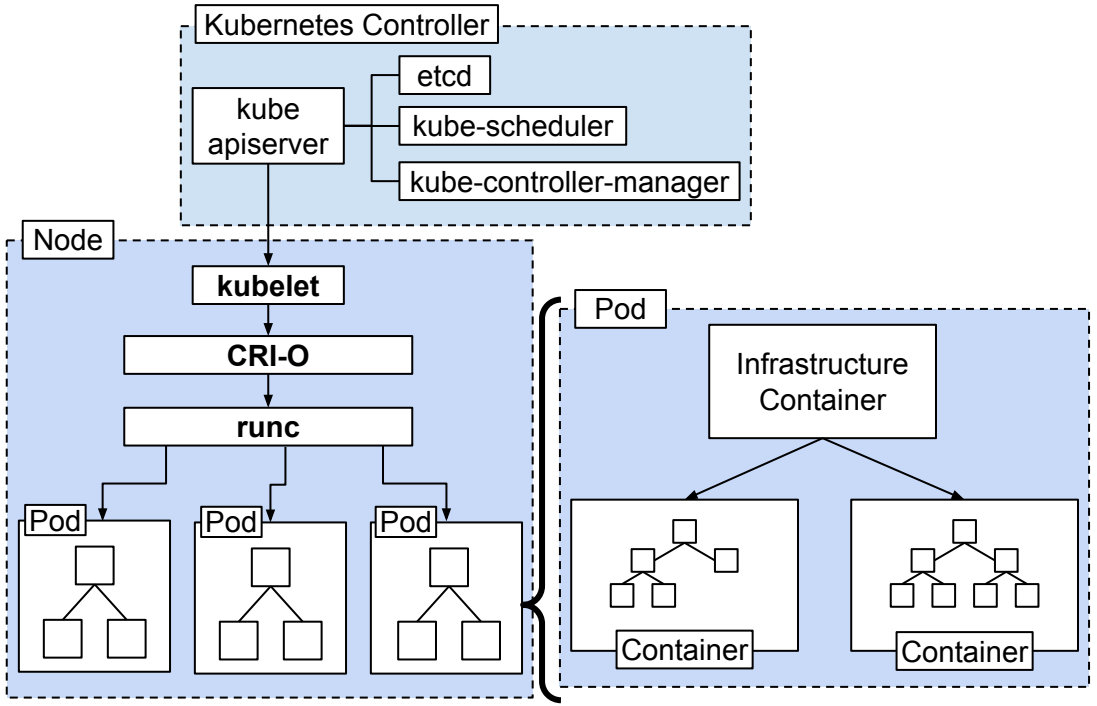
Container Checkpointing in Kubernetes



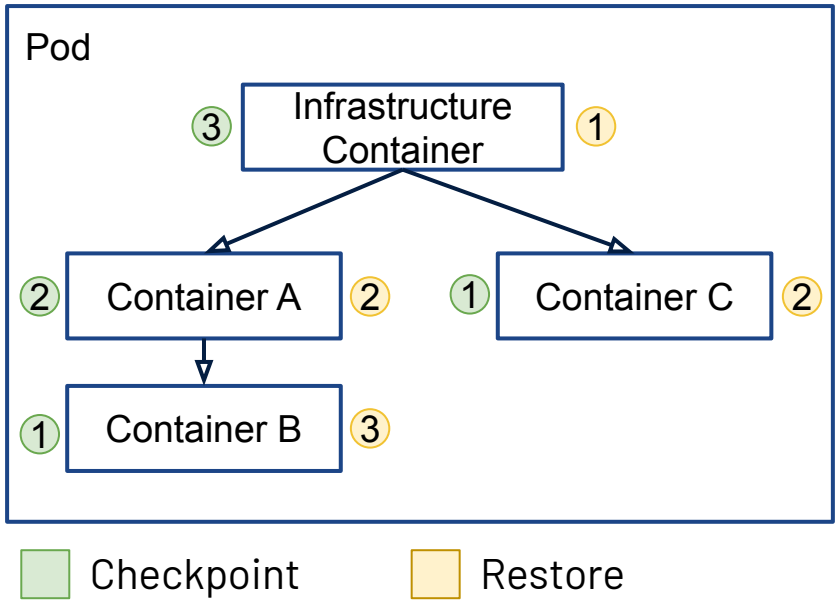
Container dependencies



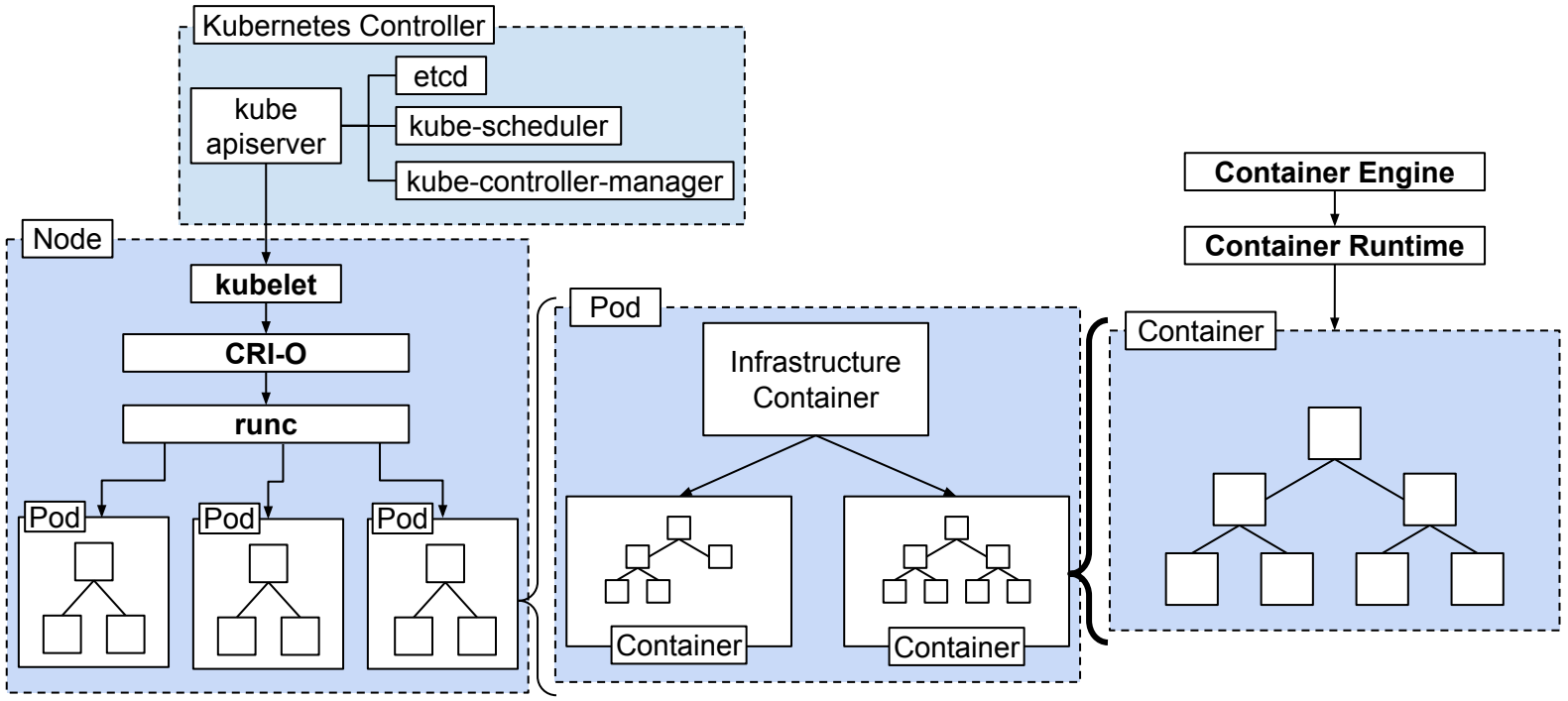
Container Checkpointing in Kubernetes



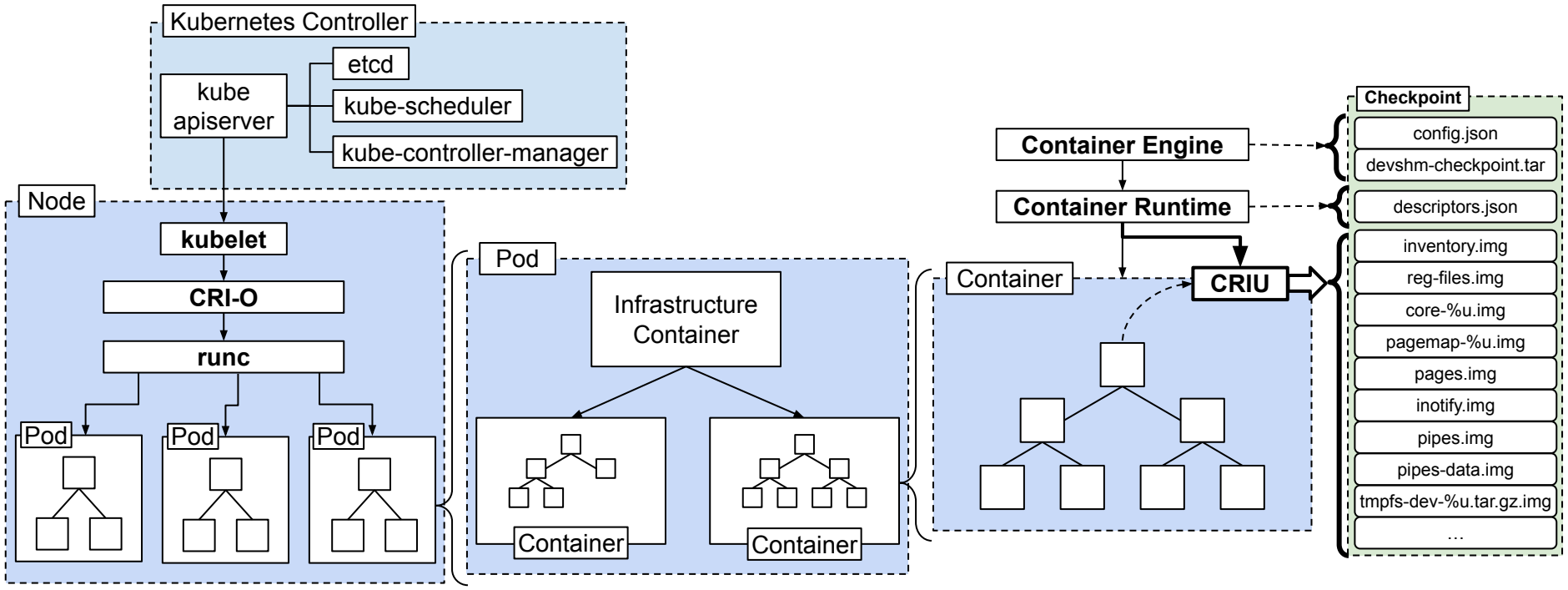
Container dependencies



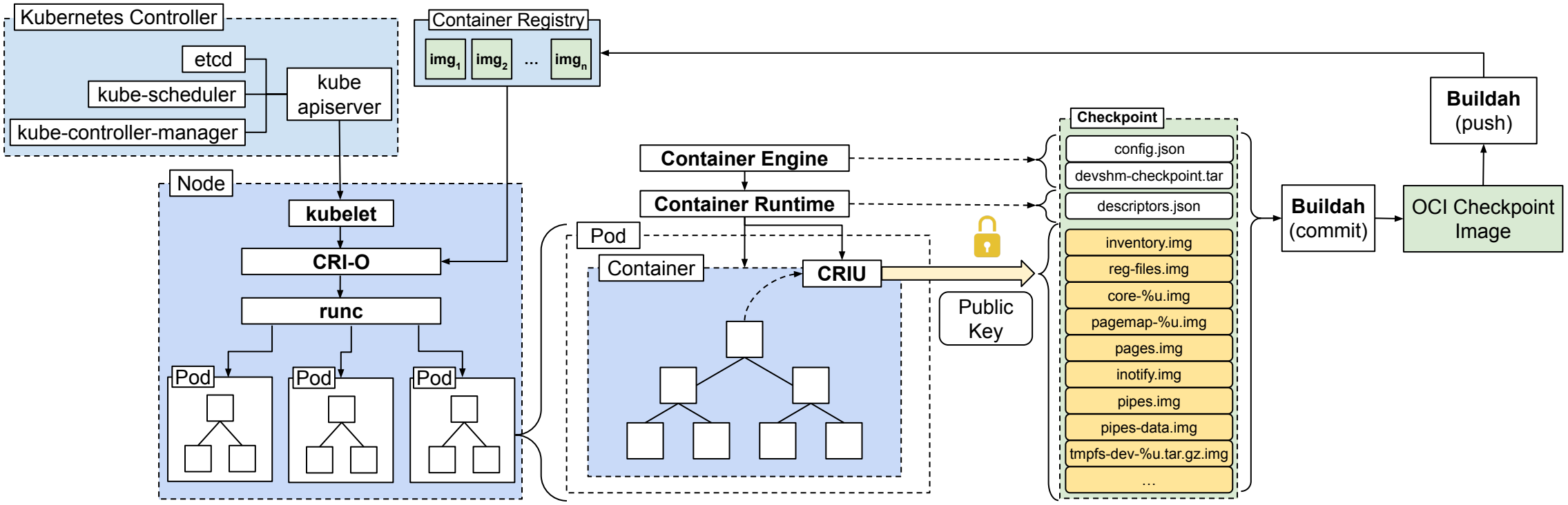
Container Checkpointing in Kubernetes



Container Checkpointing in Kubernetes



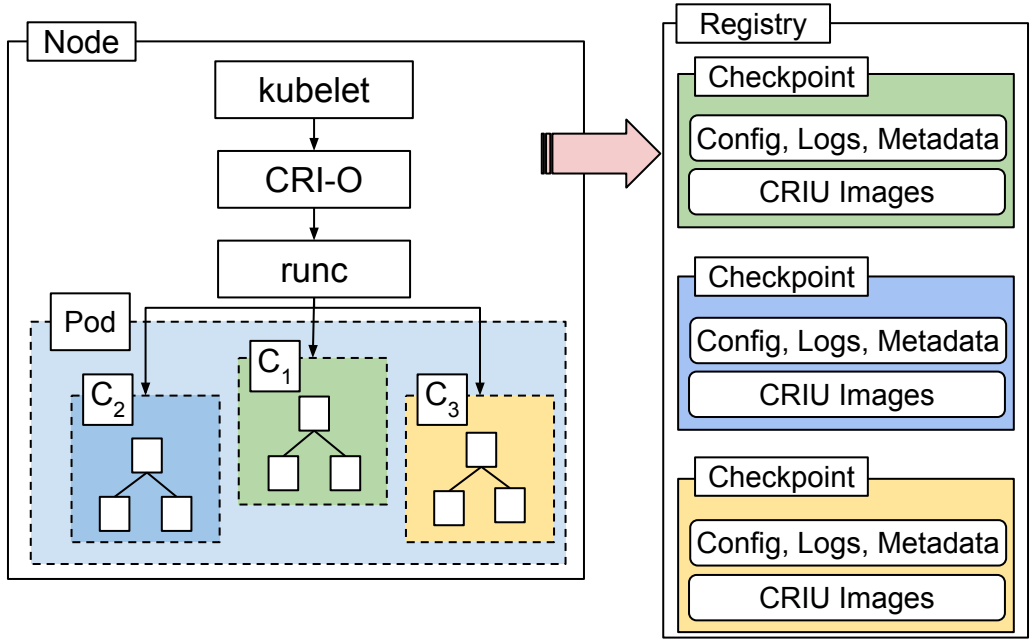
Container Checkpointing in Kubernetes



Checkpoint Images

Basic units of deployment using standard image distribution infrastructure.

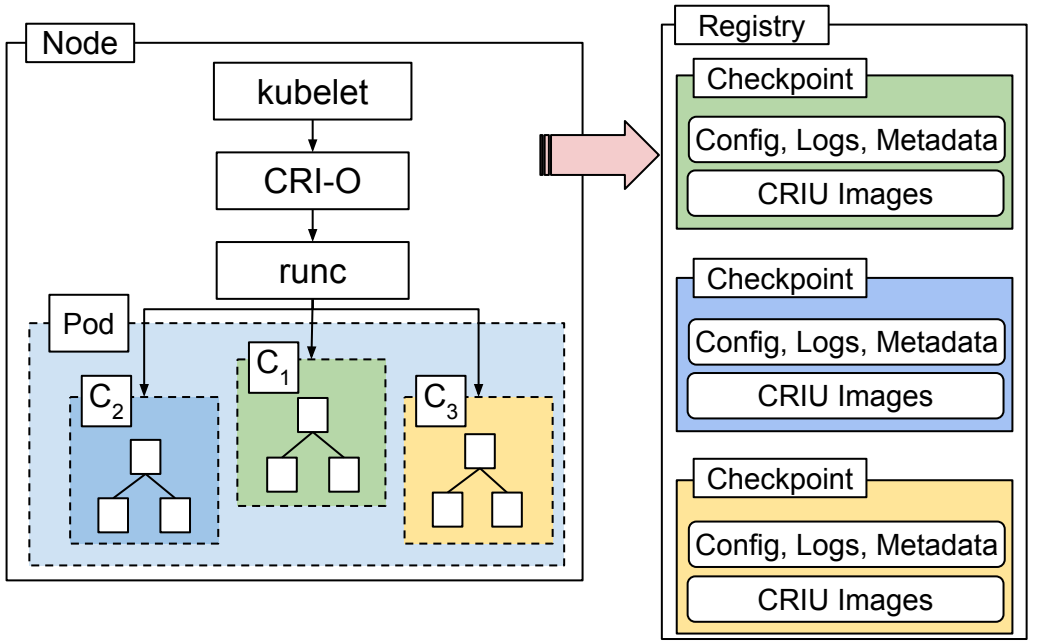
Checkpoint Images



Mar, 2022. Radostin Stoyanov. Add support for checkpoint image. https://github.com/containers/podman/pull/13505
Aug, 2022. Adrian Reber. Support checkpointing to OCI image. https://github.com/cri-o/cri-o/pull/6181
Nov, 2022. Radostin Stoyanov. Add support for checkpoint images with 'podman run'. https://github.com/containers/podman/pull/16569
Dec, 2022. Adrian Reber. Forensic container checkpointing in Kubernetes. https://kubernetes.io/blog/2022/12/05/forensic-container-checkpointing-alpha/

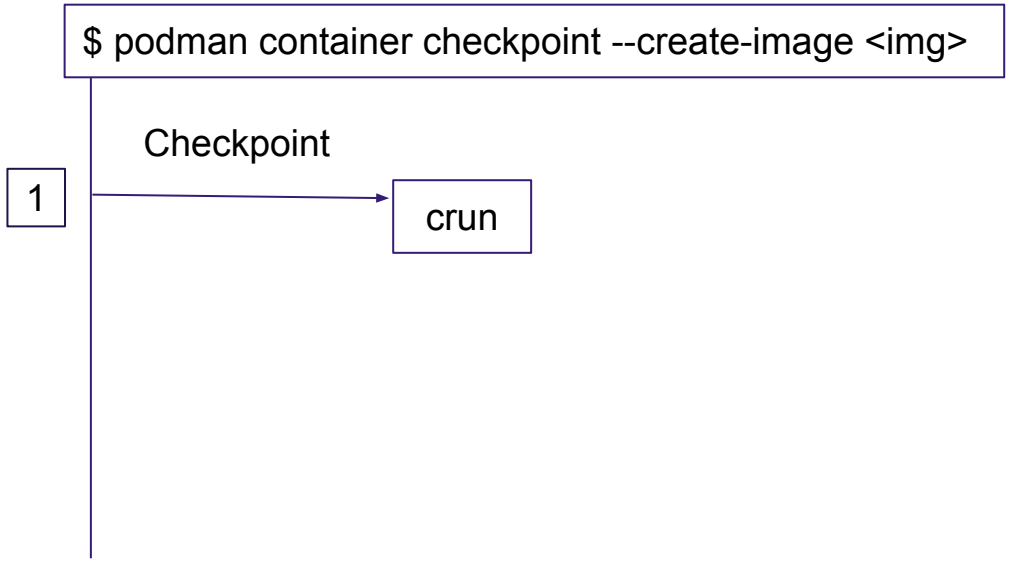
Checkpoint Images

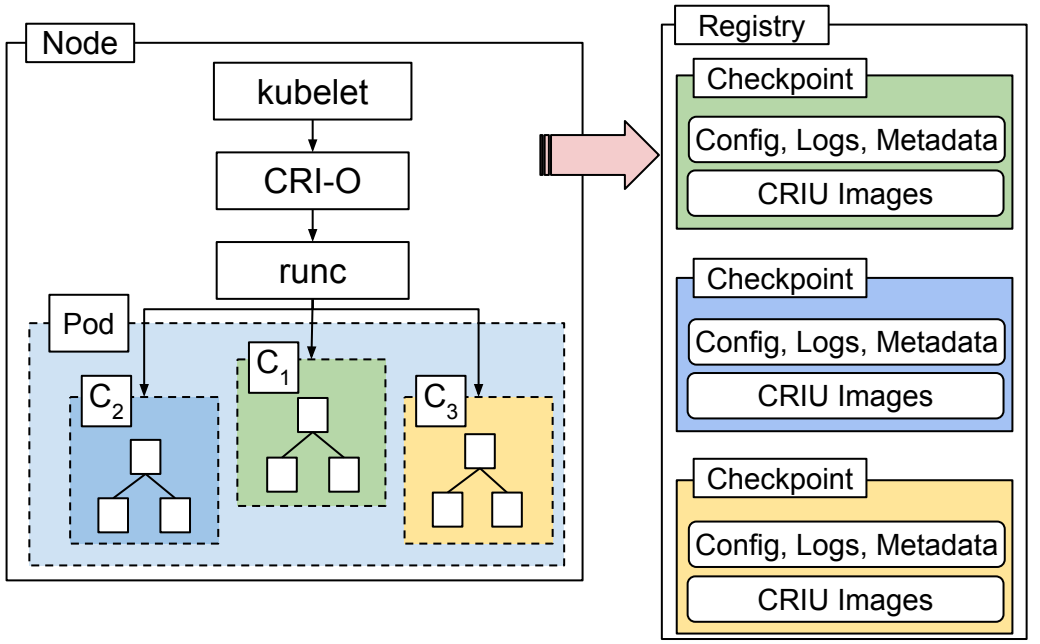
```
$ podman container checkpoint --create-image <img>
```



Mar, 2022. Radostin Stoyanov. Add support for checkpoint image. https://github.com/containers/podman/pull/13505
Aug, 2022. Adrian Reber. Support checkpointing to OCI image. https://github.com/cri-o/cri-o/pull/6181
Nov, 2022. Radostin Stoyanov. Add support for checkpoint images with 'podman run'. https://github.com/containers/podman/pull/16569
Dec, 2022. Adrian Reber. Forensic container checkpointing in Kubernetes. https://kubernetes.io/blog/2022/12/05/forensic-container-checkpointing-alpha/

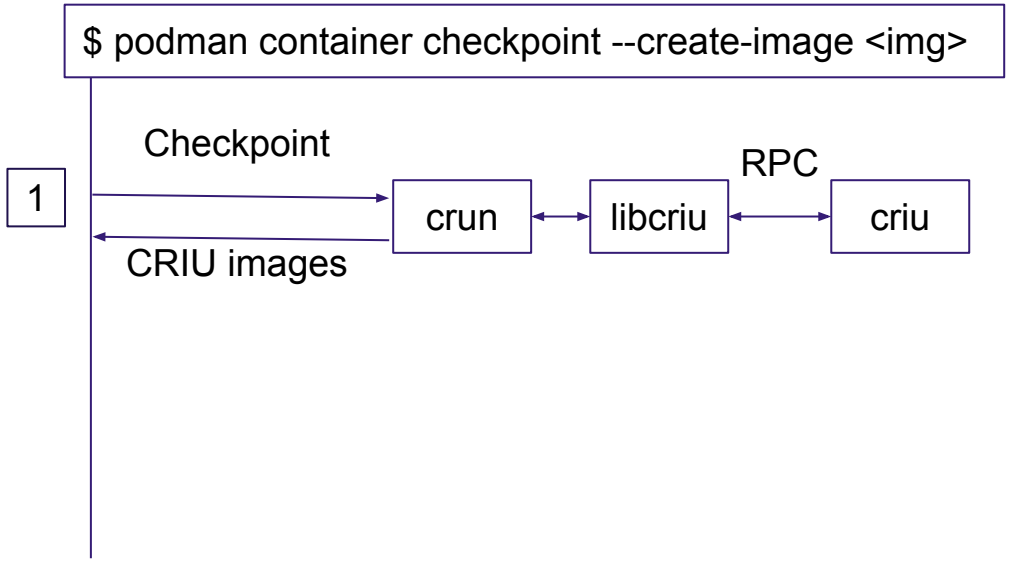
Checkpoint Images

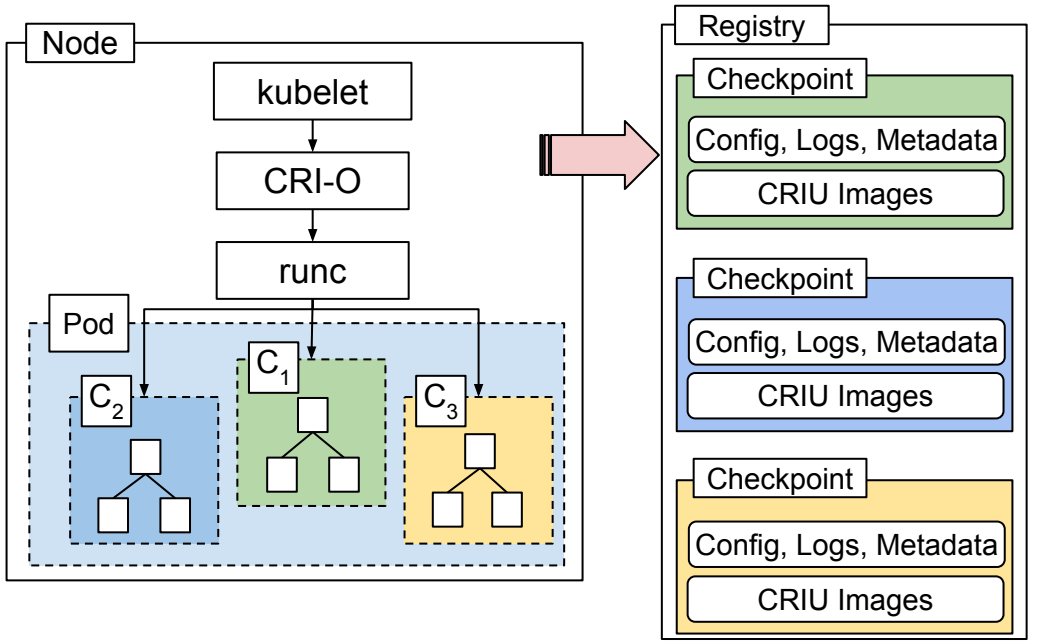




Mar, 2022. Radostin Stoyanov. Add support for checkpoint image. https://github.com/containers/podman/pull/13505
Aug, 2022. Adrian Reber. Support checkpointing to OCI image. https://github.com/cri-o/cri-o/pull/6181
Nov, 2022. Radostin Stoyanov. Add support for checkpoint images with 'podman run'. https://github.com/containers/podman/pull/16569
Dec, 2022. Adrian Reber. Forensic container checkpointing in Kubernetes. https://kubernetes.io/blog/2022/12/05/forensic-container-checkpointing-alpha/

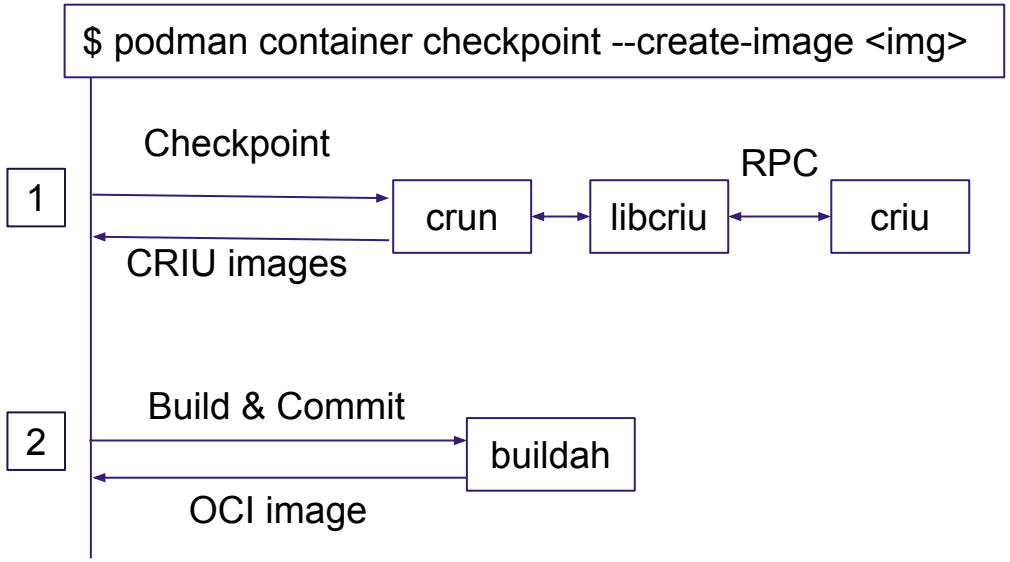
Checkpoint Images

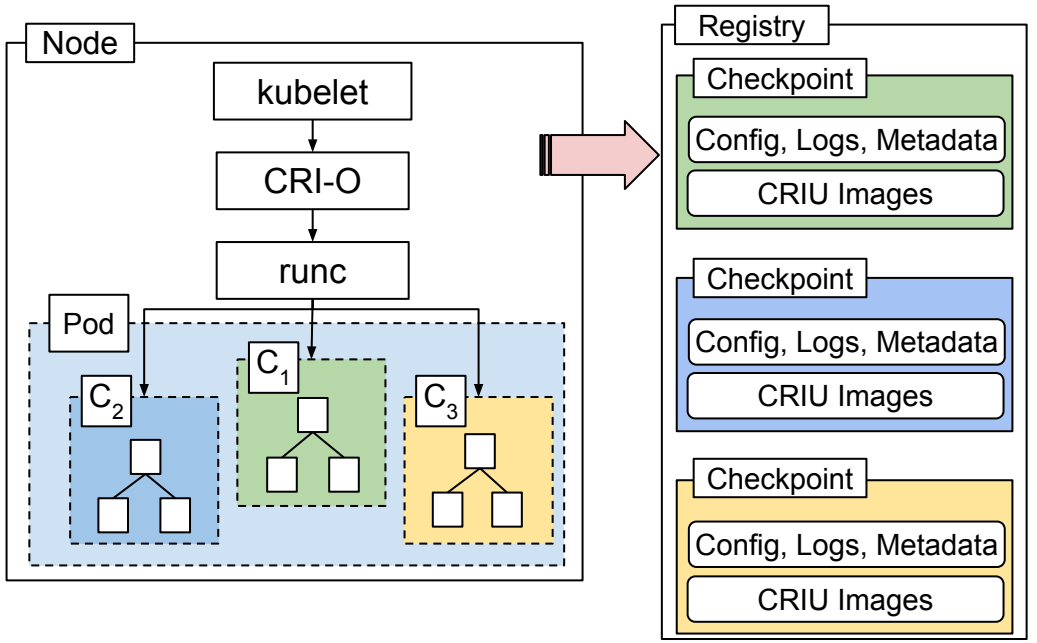




Mar, 2022. Radostin Stoyanov. Add support for checkpoint image. https://github.com/containers/podman/pull/13505
Aug, 2022. Adrian Reber. Support checkpointing to OCI image. https://github.com/cri-o/cri-o/pull/6181
Nov, 2022. Radostin Stoyanov. Add support for checkpoint images with 'podman run'. https://github.com/containers/podman/pull/16569
Dec, 2022. Adrian Reber. Forensic container checkpointing in Kubernetes. https://kubernetes.io/blog/2022/12/05/forensic-container-checkpointing-alpha/

Checkpoint Images





Mar, 2022. Radostin Stoyanov. Add support for checkpoint image. https://github.com/containers/podman/pull/13505
Aug, 2022. Adrian Reber. Support checkpointing to OCI image. https://github.com/cri-o/cri-o/pull/6181
Nov, 2022. Radostin Stoyanov. Add support for checkpoint images with 'podman run'. https://github.com/containers/podman/pull/16569
Dec, 2022. Adrian Reber. Forensic container checkpointing in Kubernetes. https://kubernetes.io/blog/2022/12/05/forensic-container-checkpointing-alpha/

Coordinated Checkpointing

Maintaining the consistency of distributed computations in the presence of failures.

Coordinated Checkpointing Model

- **Two-phase** algorithm based on Koo–Toueg [1]
- Checkpoints: **tentative** (local) and **permanent** (global)
- In case of a failure, rollback to the last permanent checkpoint

1 R. Koo and S. Toueg, Checkpointing and rollback-recovery for distributed systems, IEEE Transactions on Software Engineering, 13 (1) 1987

Coordinated Checkpointing Model

Phase 1

1: Create a *tentative* checkpoint for each container.

2: Notify the coordinator whether it succeeded.

3: If all tentative checkpoints are successful, continue to ***Phase 2***.
Otherwise, discard tentative checkpoints and go back to ***Step 1***.

Coordinated Checkpointing Model

Phase 2

1: All tentative checkpoints are made permanent.

2: Transfer local checkpoints to coordinator.

3: Create OCI checkpoint images.

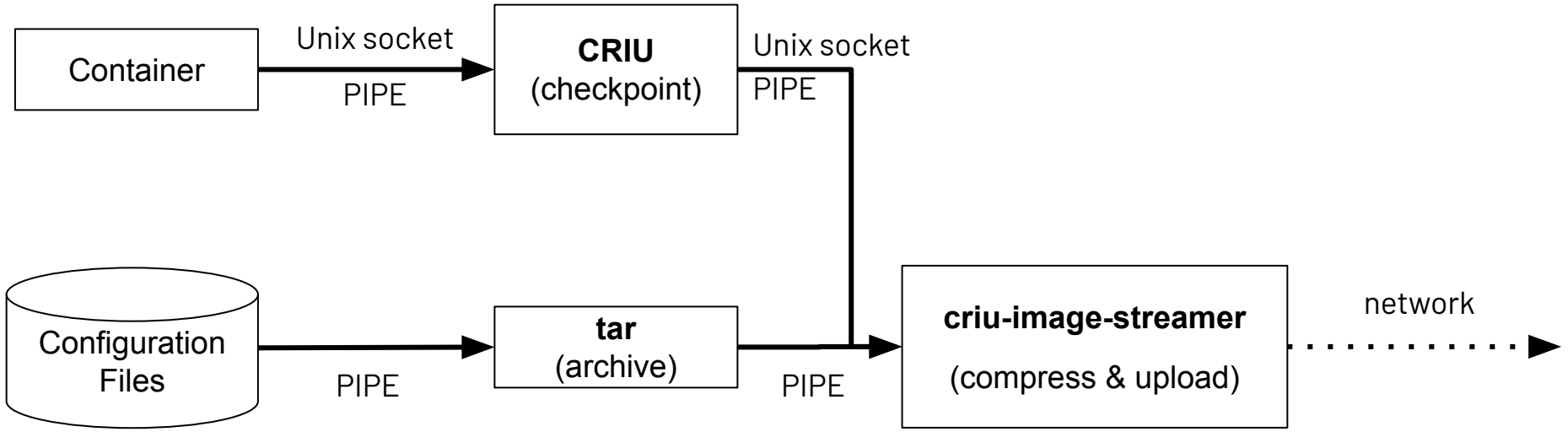
4: Upload images to registry.

CRIU Coordinator

- Per container CRIU configuration file [1]
- An external executable running as an action script [2]
- Checkpoint synchronization at *pre-dump* or *pre-stream* [3]
- Environment variables config [4]:
  - Container/Pod ID
  - Dependencies
  - Address & Port
  - Log

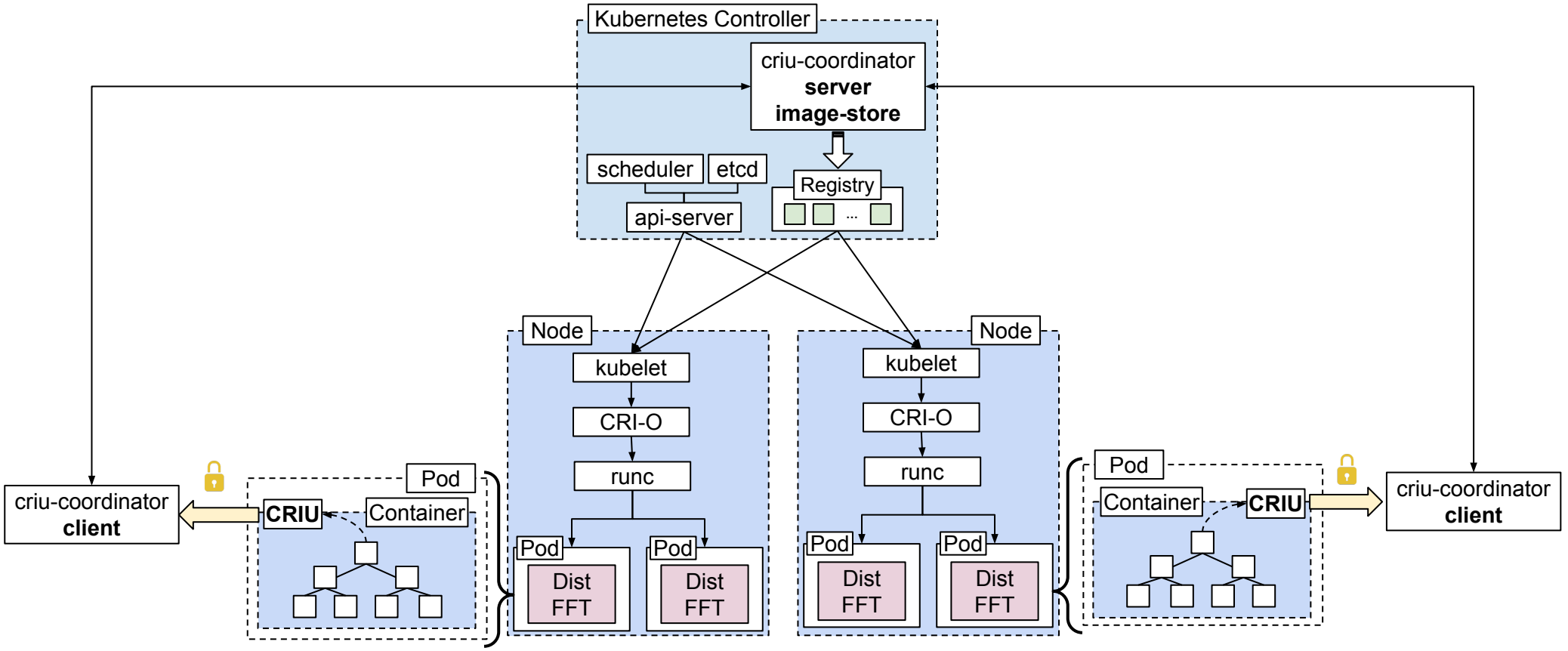
1 https://github.com/opencontainers/runc/blob/main/docs/checkpoint-restore.md
2 https://criu.org/Action_scripts
3 action-scripts: add pre-stream hook (https://github.com/checkpoint-restore/criu/pull/2149)
4 config: add –action-env cli option (https://github.com/checkpoint-restore/criu/pull/2136)

Checkpoint Streaming



Nicolas Viennot (Two Sigma). *Fast checkpointing with criu-image-streamer*. Linux Plumbers Conference. 2020
Radostin Stoyanov, et. al., *Secure Image-less Container Migration*. Linux Plumbers Conference. 2019
Rodrigo Bruno, et. al., *GC-assisted JVM Live Migration for Java Server Applications*. Middleware '16

Coordinated Checkpointing Model

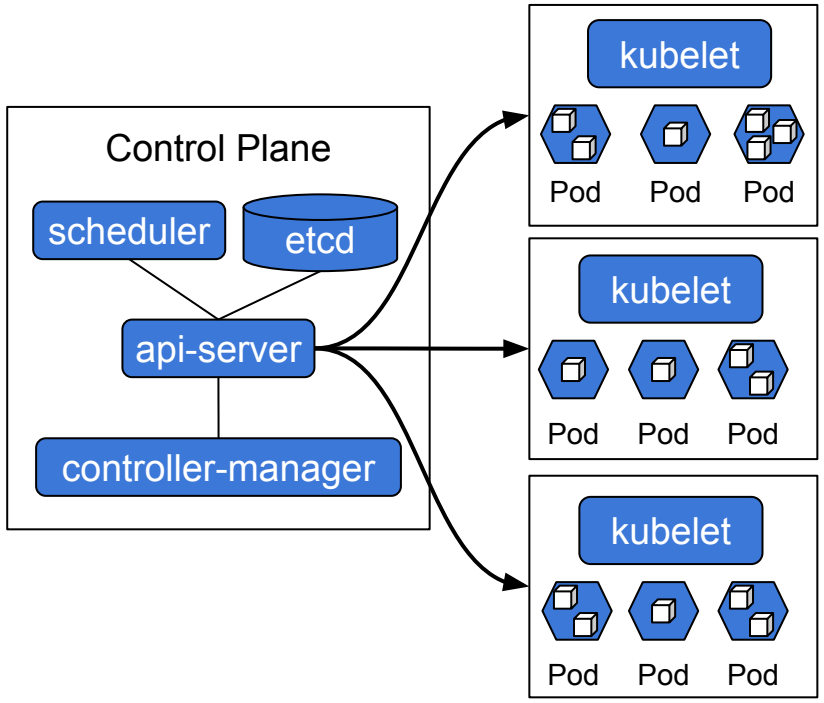


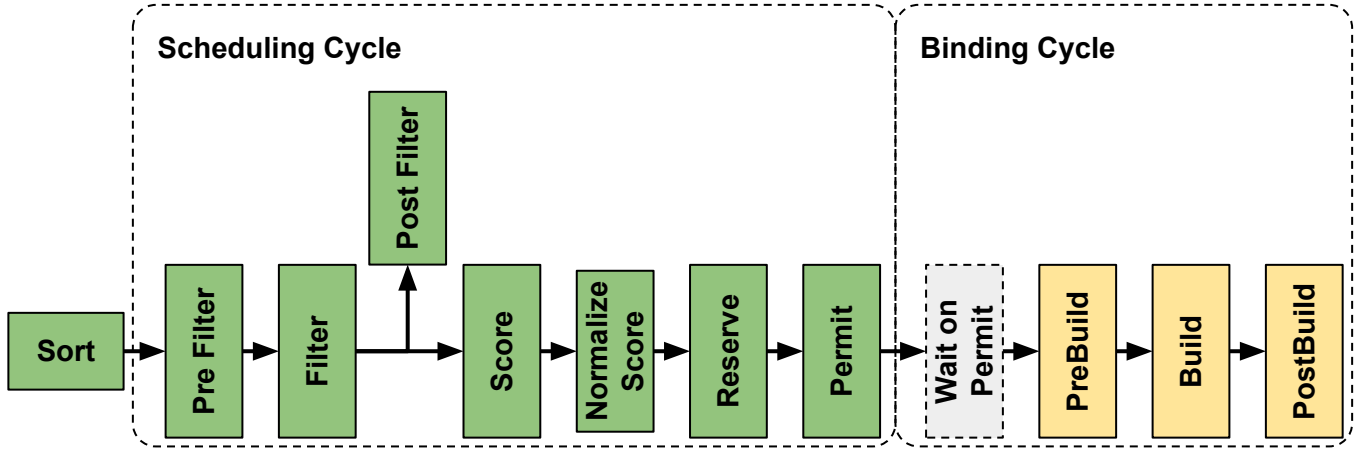
Global Checkpoint

Ensuring consistency and correctness of distributed applications.

Global Checkpoint

Preemptive Scheduling

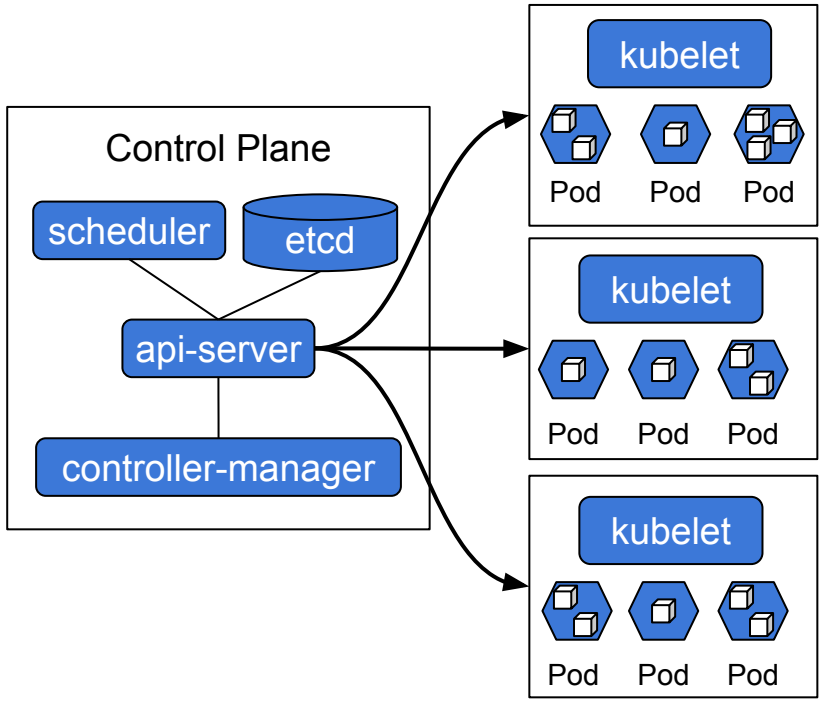




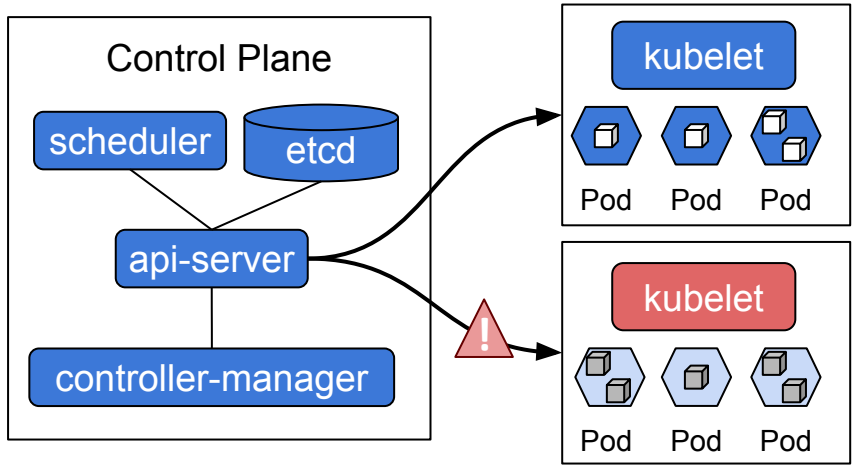
https://kubernetes.io/docs/concepts/scheduling-eviction/scheduling-framework/

Global Checkpoint

Preemptive Scheduling

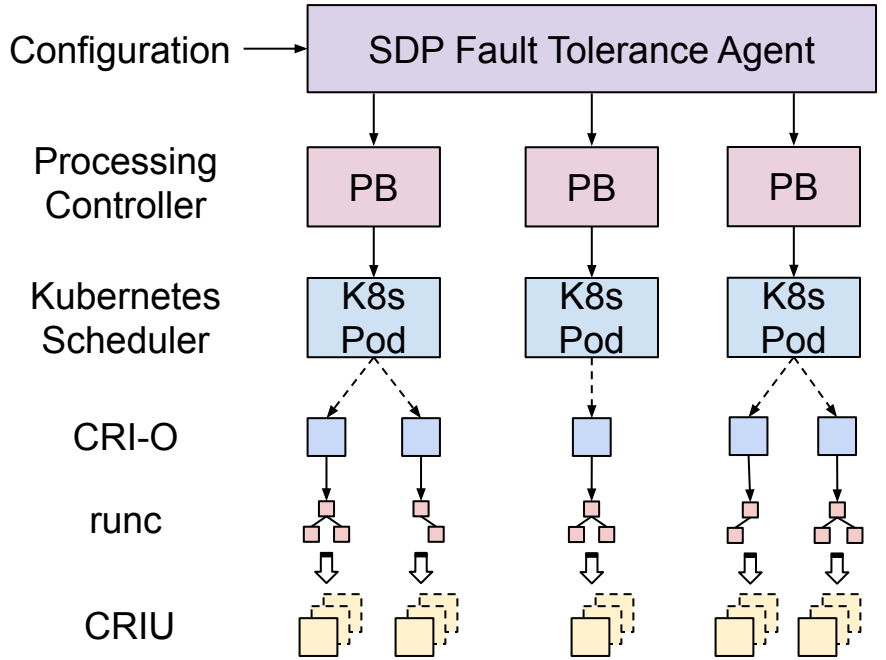


Application Fault Tolerance



Application Fault Tolerance

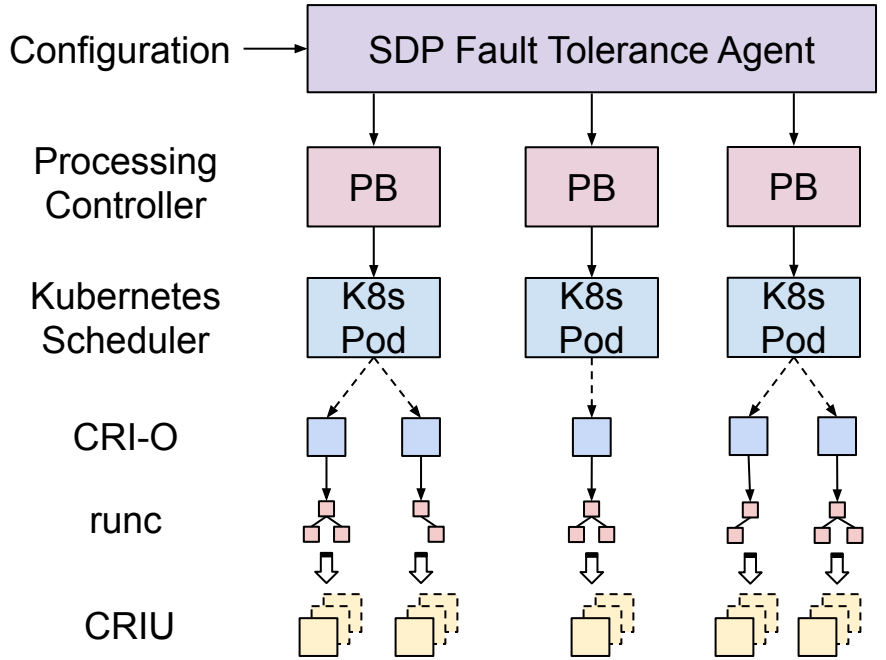
```
root@ska-sdp-console-0:/app# ska-sdp --help
Command line utility for interacting with SKA Science Data Processor (SDP).

Usage:
    ska-sdp COMMAND [options] [SDP_OBJECT] [<args>...]
    ska-sdp COMMAND (-h|--help)
    ska-sdp (-h|--help)

SDP Objects:
    pb            Interact with processing blocks
    script        Interact with available processing script definitions
    deployment    Interact with deployments
    eb            Interact with execution blocks
    controller    Interact with Tango controller device
    subarray      Interact with Tango subarray device

Commands:
    checkpoint    A fault-tolerance mechanism periodically saving the runtime
                  state of execution block
    list          List information of object from the Configuration DB
    get | watch   Print all the information (i.e. value) of a key in the Config DB
    create        Create a new, raw key-value pair in the Config DB;
                  Run a processing script; Create a deployment
    update        Update a raw key value from CLI
    edit          Edit a raw key value from text editor
    delete        Delete a single key or all keys within a path from the Config DB
    end           Stop/Cancel execution block
    import        Import processing script definitions from file or URL
```



https://gitlab.com/ska-telescope/sdp/ska-sdp-config.git

Application Fault Tolerance

```
root@ska-sdp-console-0:/app# ska-sdp checkpoint --help
A fault-tolerance mechanism that periodically saves the runtime state
of processing block(s) to persistent storage. In the case of failure,
the processing block(s) will be restarted from the most recent
checkpoint and resume processing.

Usage:
    ska-sdp checkpoint (-a |--all) list
    ska-sdp checkpoint [options] interval [<interval>]
    ska-sdp checkpoint [options] storage [<location>]
    ska-sdp checkpoint (-h|--help)

Arguments:
    <interval>  Checkpoint interval in milliseconds.
    <location>  The location where checkpoint snapshots are made durable.
```

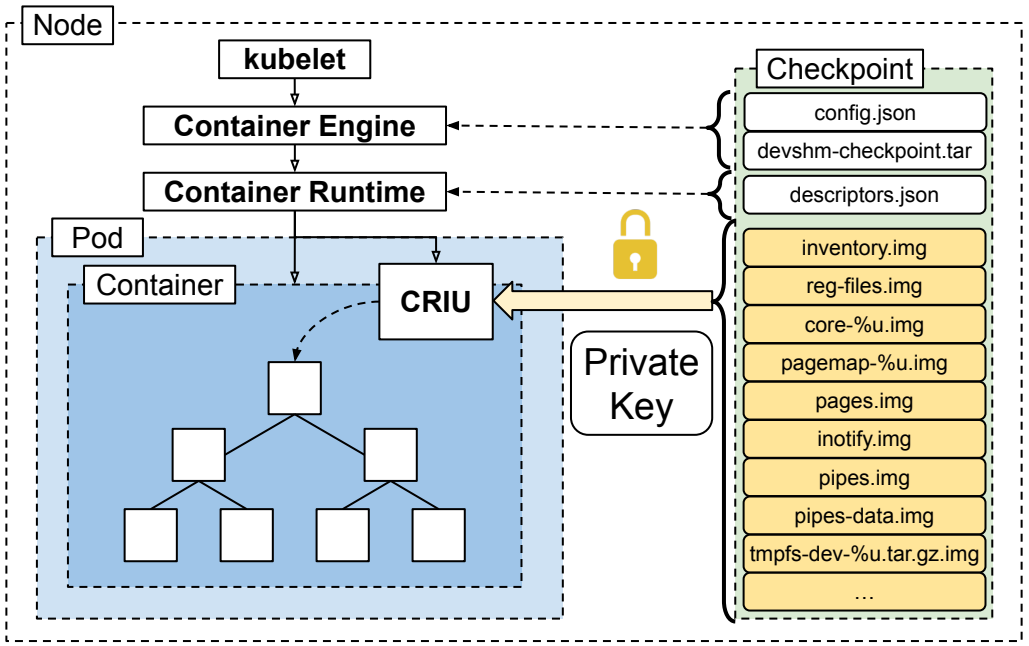


https://gitlab.com/ska-telescope/sdp/ska-sdp-config.git

Rollback Recovery

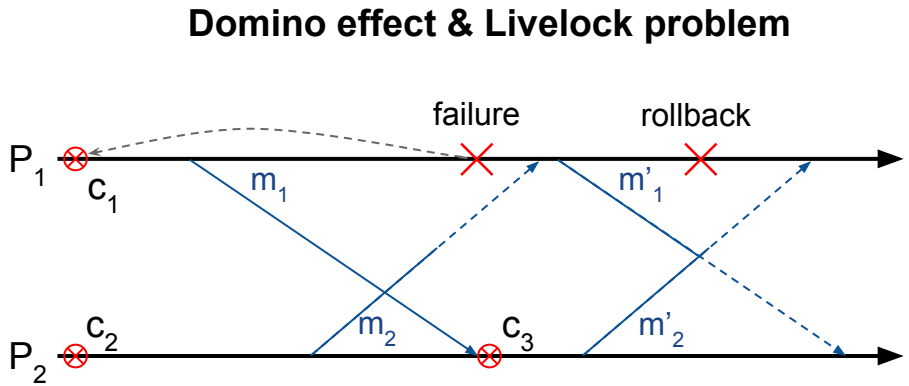
Recovering from failures and continue to operate correctly.

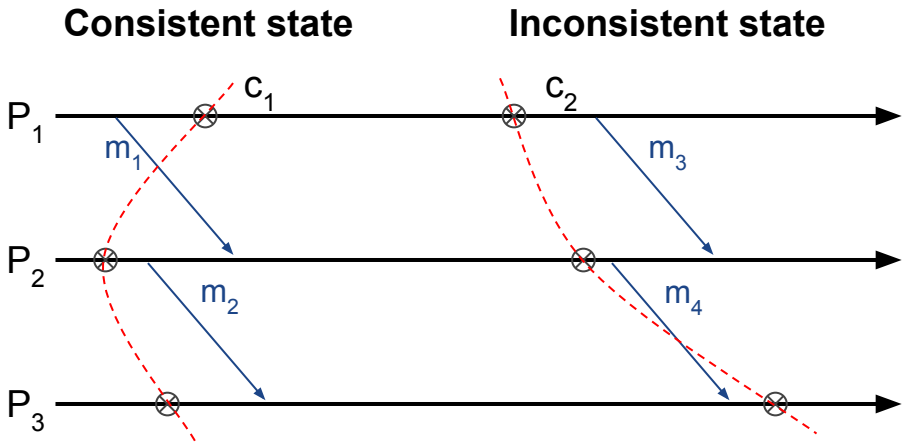
Rollback Recovery

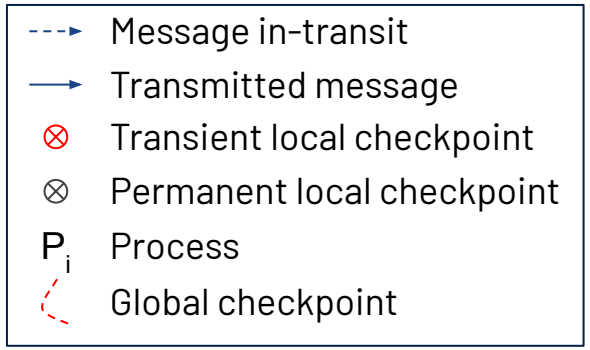



```
[radostin@thinkpad:~] 2 $ criu --help | grep "\-\-tls"
  --tls-cacert FILE      trust certificates signed only by this CA
  --tls-cacrl FILE       path to CA certificate revocation list file
  --tls-cert FILE        path to TLS certificate file
  --tls-key FILE         path to TLS private key file
  --tls                  use TLS to secure remote connection
  --tls-no-cn-verify     do not verify common name in server certificate

[radostin@thinkpad:~] $ sudo cat /etc/criu/runc.conf
tls
tls-cacert=/etc/pki/CA/cacert.pem
tls-cert=/etc/pki/criu/servercert.pem
tls-key=/etc/pki/criu/private/serverkey.pem


$ criu dump --tls -t <PID> -D /var/lib/containers/storage/<container ID>

$ criu restore --tls -D /var/lib/containers/storage/<container ID>
```

Rollback Recovery







Rollback Recovery

Phase 1:

1: Create containers from checkpoint images.

2: Restore synchronization at *pre-restore* hook.

3: Notify the coordinator whether it succeeded.

4: If all containers are successfully restored, continue to **Phase 2**.

Otherwise, remove containers and go back to **Step 1**.

Rollback Recovery

Phase 2:

1: Restart synchronization at *pre-resume* hook.

2: Notify the coordinator with result.

Rollback Recovery

Correctness

- Either *all* or *none* of the containers have permanent checkpoints.
- All containers resume execution from a consistent state.
- During recovery, execution is resumed only when all containers have been successfully restored.

Summary & Questions

https://criu.org/

https://github.com/checkpoint-restore/criu