Forensic Analysis of Container Checkpoints

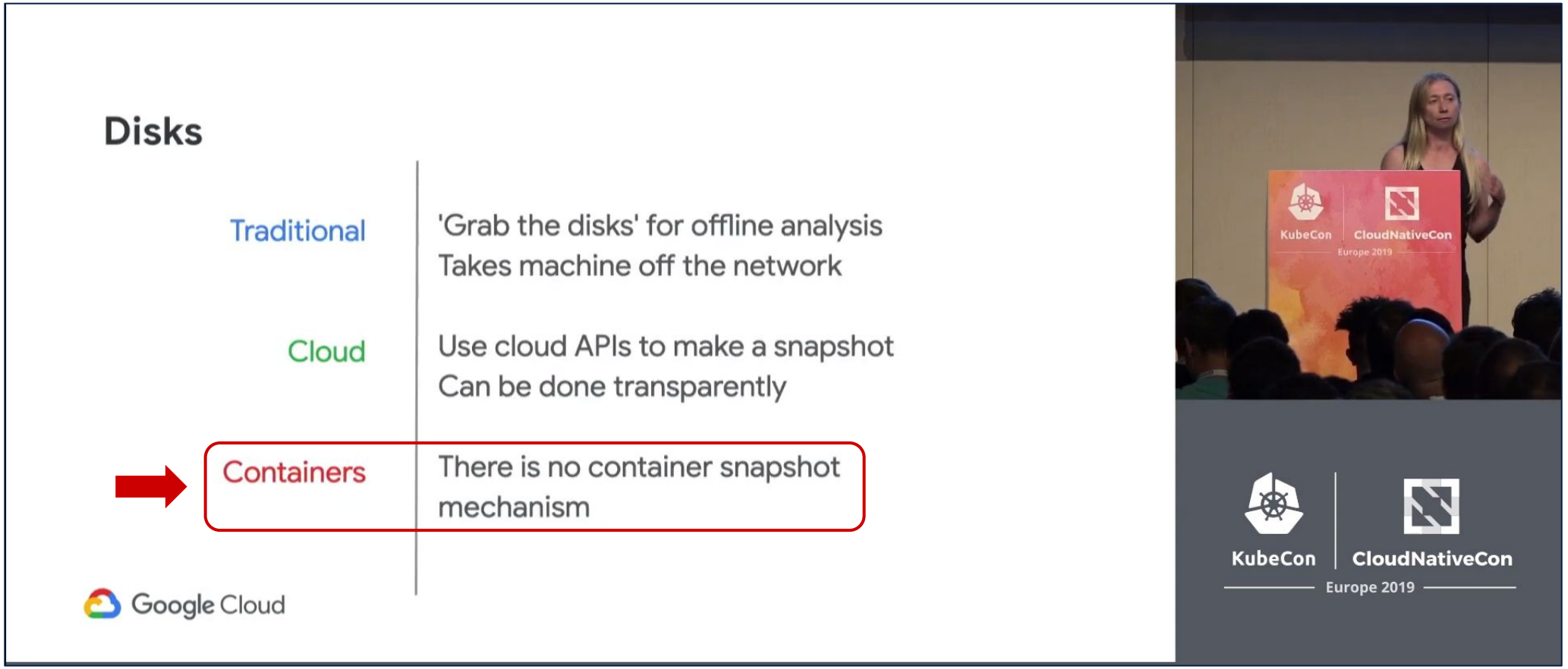
Radostin Stoyanov - PhD Student, Scientific Computing Group

Collaboration with Adrian Reber, Senior Principal Software Engineer

Supervisor: Prof. Wes Armour

KubeCon Europe 2019

KubeCon Europe 2019



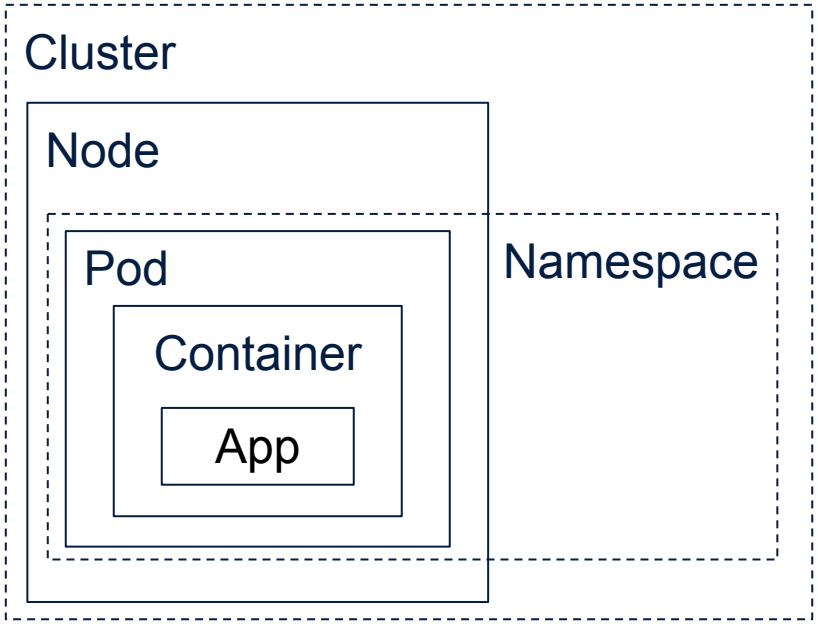
Container Forensics: What to Do When Your Cluster is a Cluster

Maya Kaczorowski & Ann Wallace, Google

Overview

- Security Boundaries & Threat Model
- Container Checkpointing
- Forensic Analysis
- Limitations & Future work

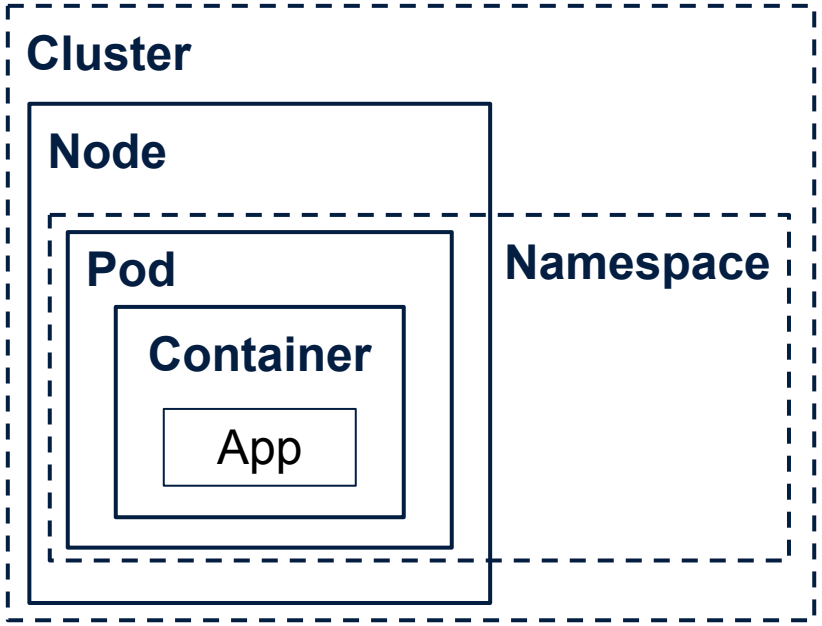
Kubernetes Security Boundaries



Areas of concern:

- Configurable components
- Applications running in Kubernetes

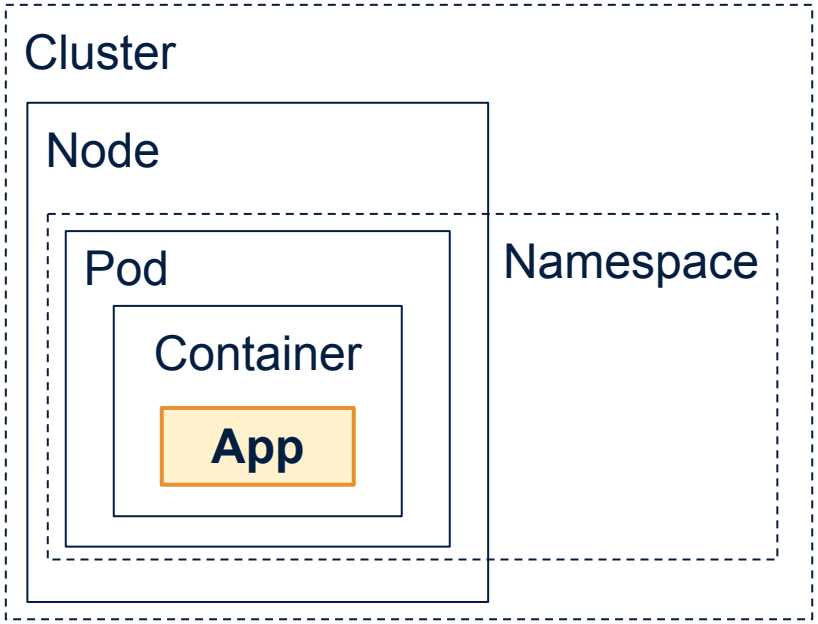
Kubernetes Security Boundaries



Areas of concern:

- Configurable components
  - **Cluster** (kube-apiserver)
  - **Nodes** (kubelet, kube-proxy)
  - **Namespaces** (role-based access control)
  - **Pods** (security context)
  - **Containers** (Linux namespaces, mounts)

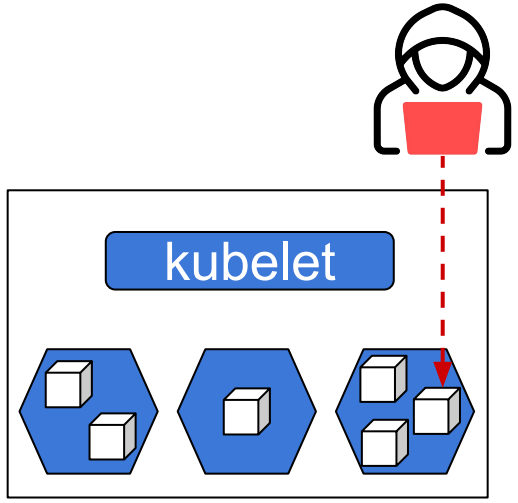
Kubernetes Security Boundaries



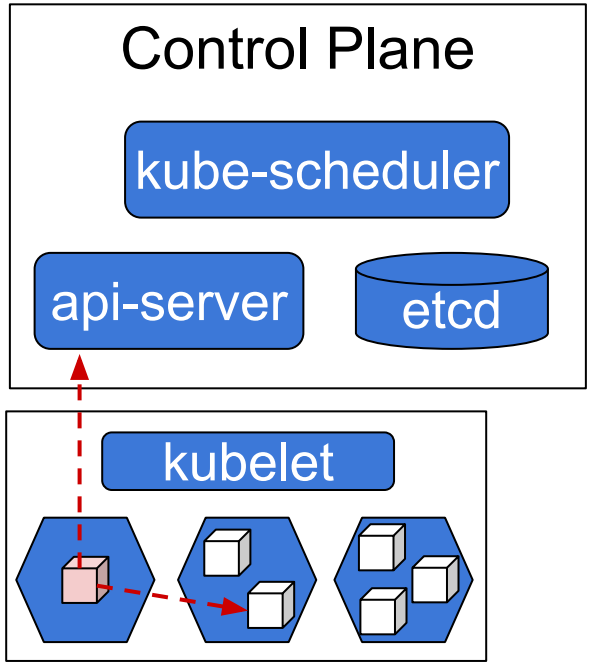
Areas of concern:

- Applications running in Kubernetes
  - **Network namespaces** (NAT)
  - **Configuration** (DNS, proxy)
  - **Secrets** (passwords, tokens)
  - **Application data** (HTTP requests, user data)

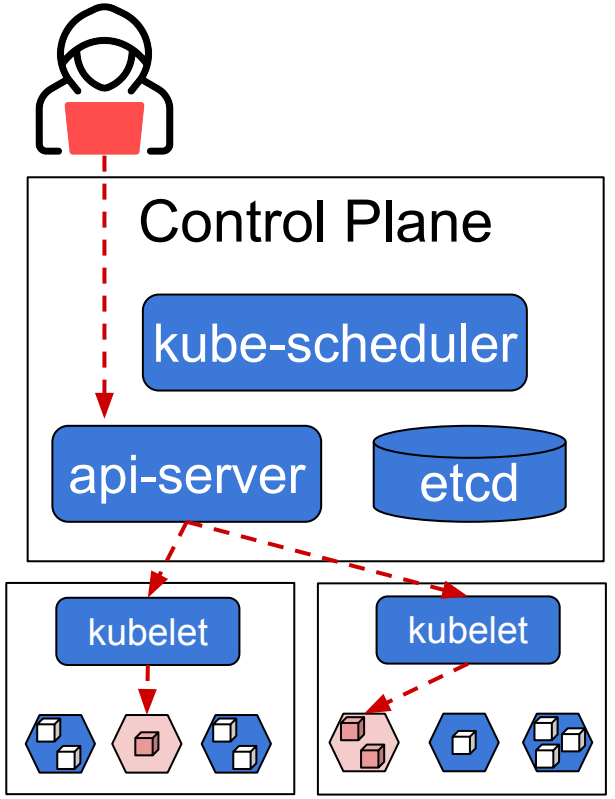
Threat Model



External Attacker



Malicious Container

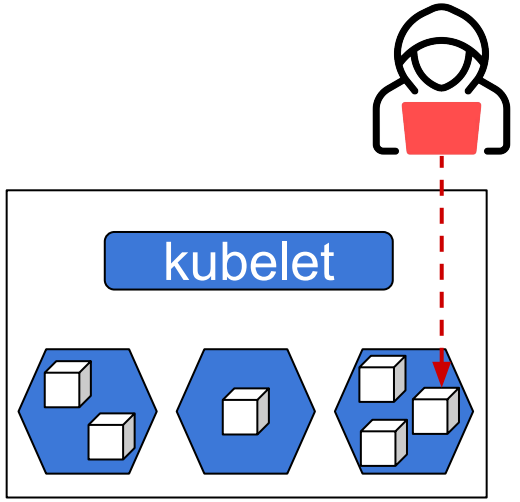


Malicious User
(stolen credentials)

Threat Model

External Attacker

- Risk
  - Access to an application running over the network
- Security Controls
  - Using TLS for all API traffic
  - API authentication & authorization

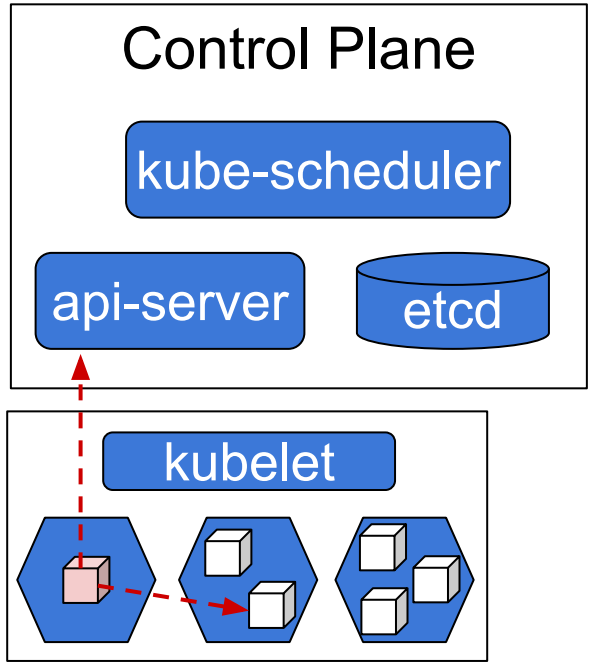


Threat Model

Malicious Container

- Risk
  - Access to container expanding to the cluster (privilege escalation)

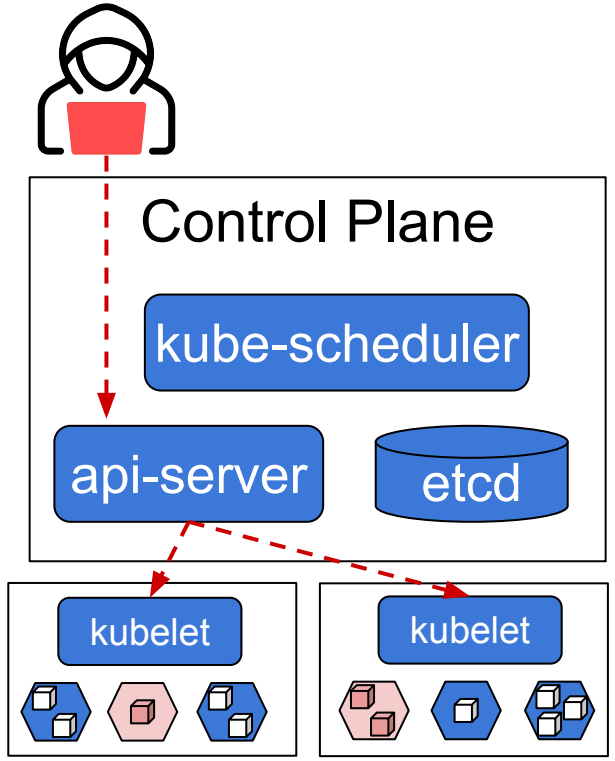
- Security Controls
  - Controlling what privileges containers run with
  - Controlling access to the kubelet
  - Preventing loading unwanted kernel modules
  - Restricting network & API access



Threat Model

Malicious User

- Risk
  - Valid (stolen) credentials & network access
- Security Controls
  - Role-based access control
  - Principle of least privilege
  - Resource quotas for tenant workloads
  - Network isolation



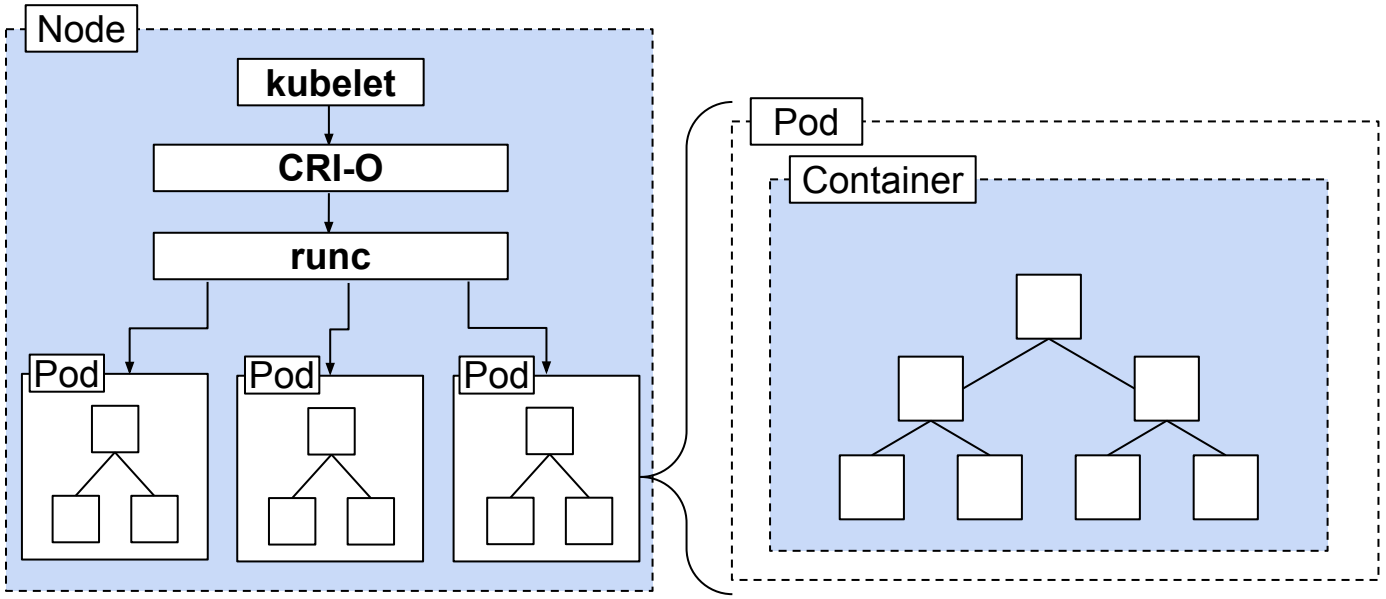
Problem Statement

- Real-time monitoring tools lack the capability to capture the runtime state of containers to perform comprehensive forensic analysis
- Container checkpointing is needed to capture and preserve the state of containers at a specific point in time
- Advanced analysis tools are needed for incident response capabilities

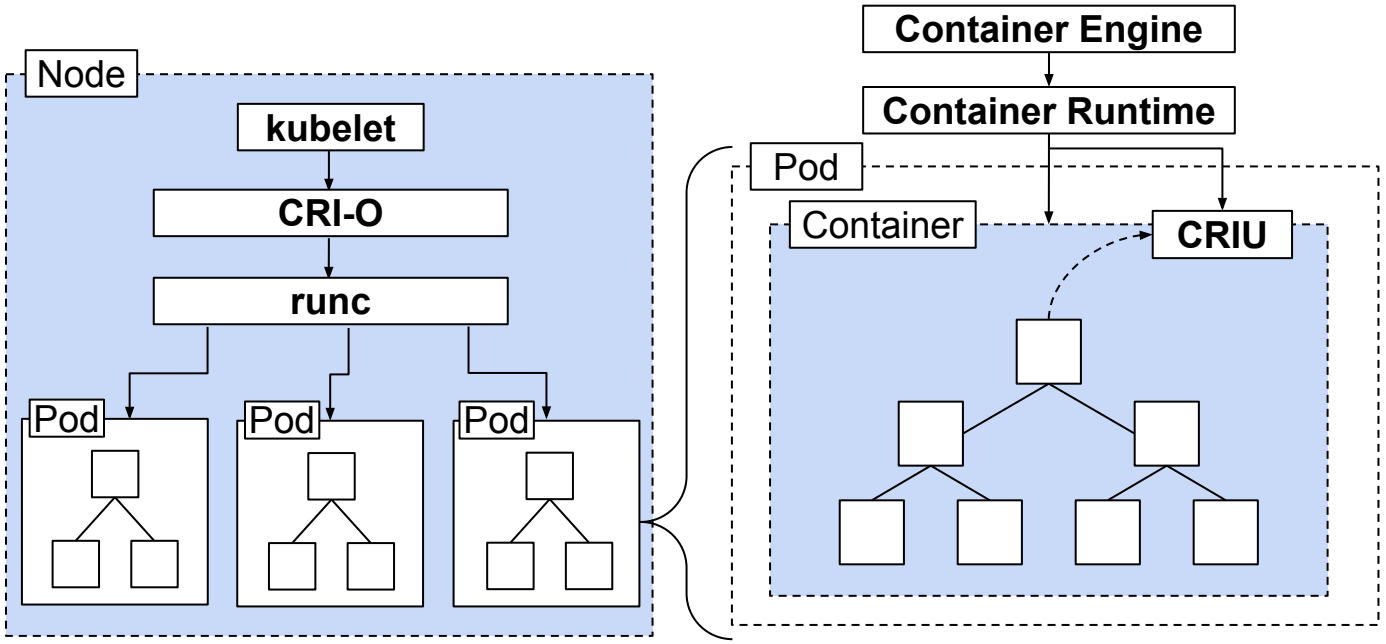
Container Checkpointing in Kubernetes

How to enable and use container checkpointing?

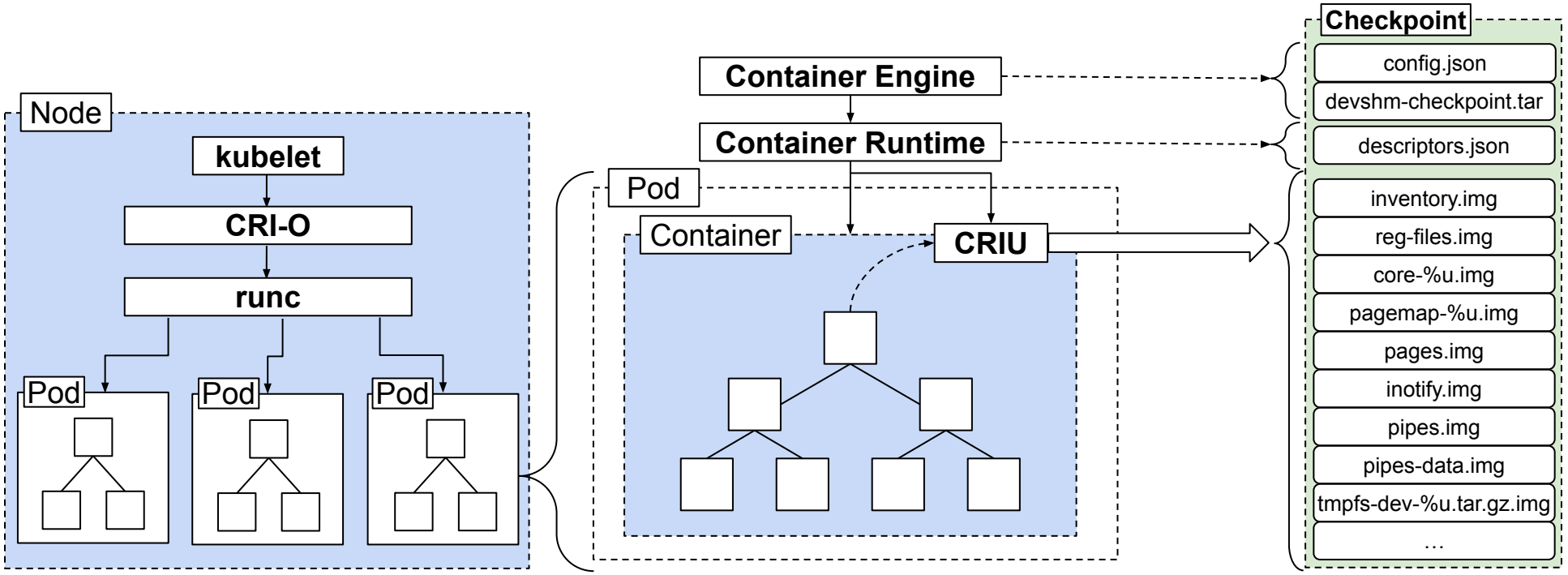
Container Checkpointing



Container Checkpointing



Container Checkpointing



How to enable checkpointing?

1. **CRIU** installed on every kubelet node
2. **CRI-O** v1.25 or newer
   - **containerd** support is currently under review (#6965)
3. Start CRI-O with **`--enable-criu-support=true`**
4. Enable **`ContainerCheckpoint`** feature gate

How to use checkpointing?

Kubelet HTTP POST request

```
curl --cert /var/lib/kubelet/pki/kubelet-client-current.pem \
     --key /var/lib/kubelet/pki/kubelet-client-current.pem \
     -X POST \
     "https://${HOST}:10250/checkpoint/${NAMESPACE}/${POD}/${CONTAINER}"
```

How to use checkpointing?

- Checkpoint file name format

```
checkpoint-${POD}_${NAMESPACE}-${CONTAINER}-${TIMESTAMP}.tar
```

- Default directory: **`/var/lib/kubelet/checkpoints`**

- Note: This format may change in future versions:
  - Partitioned checkpoints into subdirectories: #115888

DEPARTMENT OF ENGINEERING SCIENCE

UNIVERSITY OF OXFORD

Demo

+

cri-o

+

Limitations

Limitations

- Kubernetes Secrets
- Obfuscating Attack Behavior
- Causing a Checkpoint Failure

Kubernetes Secrets

- Security risk
  - If are not adequately protected, can lead to potential security breaches

- Should secrets be captured within container snapshots?
  - Live migration & Fault tolerance: Yes
  - Scaling applications & Fast start-up: No

Obfuscating Attack Behavior

- Obfuscation Techniques
  - Adding irrelevant data
  - Performing legitimate activities
  - Mimicking the behavior of trusted processes
- Deceptive Techniques
  - Camouflaging malicious activities
  - Evading behavioral analysis

Causing a Checkpoint Failure

An attacker can prevent container checkpointing by causing CRIU to fail

- Unsupported system calls (eBPF, ptrace)
- Unsupported file descriptors / sockets (SCTP)
- Nested namespaces
- Others (https://criu.org/What_cannot_be_checkpointed)

Future Work

Intrusion Detection & Prevention

How do container checkpoints help detect and prevent cyber attacks?

Enhancing Intrusion Detection

- Improved visibility of monitoring tools
- Signature-based detection
- Container restart policy (*restart-from-checkpoint*)

Analysing Compromised Applications

Potential attack scenarios:

- SQL Injection
- CMD Execution
- Local / Remote File Inclusion

Analysing Malicious Containers

- Real-Time Monitoring
- Anomaly Detection
- Security Policy Enforcement
- Incident Response and Forensics

+

Google Summer of Code

Google Summer of Code

Prajwal S N
@snprajwal

Behouba Manassé
@behouba

Summary & Questions

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

https://kubernetes.io/blog/2023/03/10/forensic-container-analysis/

https://access.redhat.com/solutions/7008477