KubeCon | CloudNativeCon

North America 2023

KubeCon | CloudNativeCon

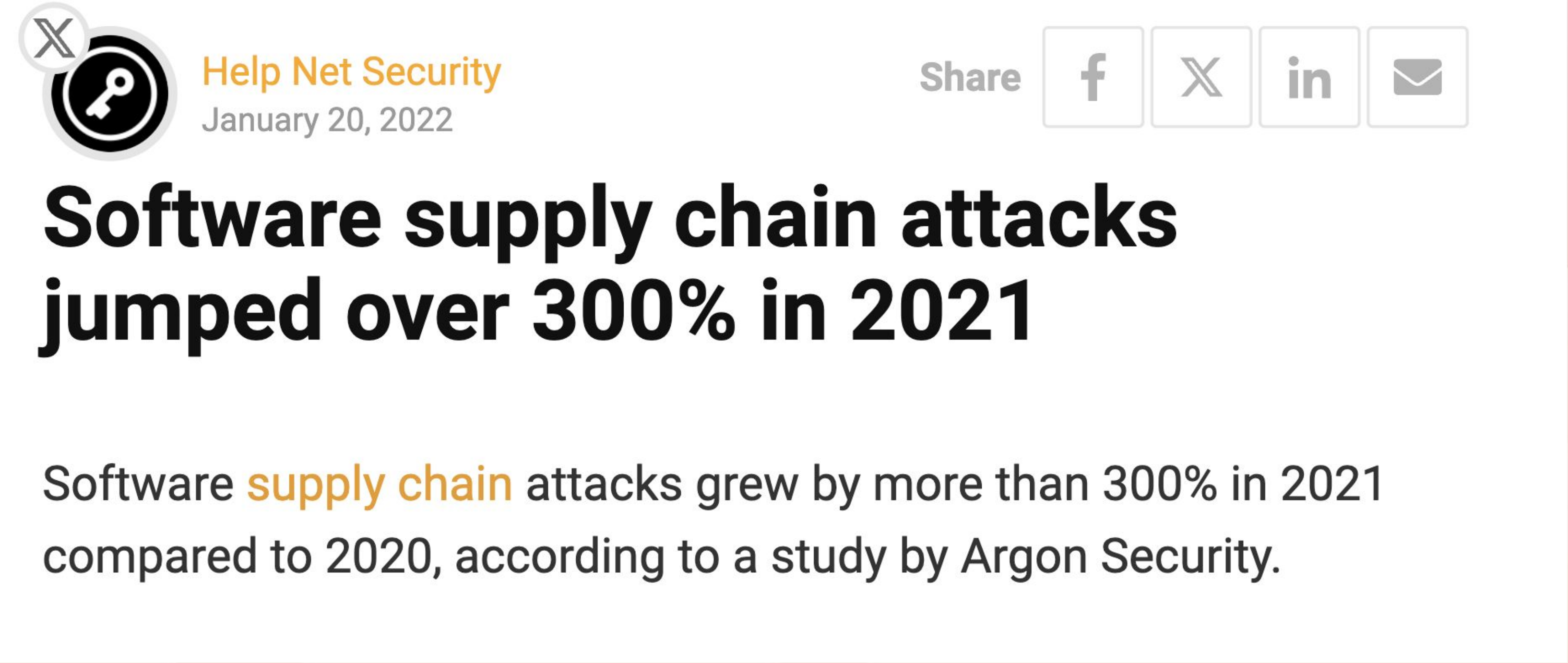
North America 2023

Eraser: Cleaning up Vulnerable Images from Kubernetes Nodes

Ashna Mehrotra, Peter Engelbert

Why Eraser?

1. Eliminate risk of spinning up vulnerable images



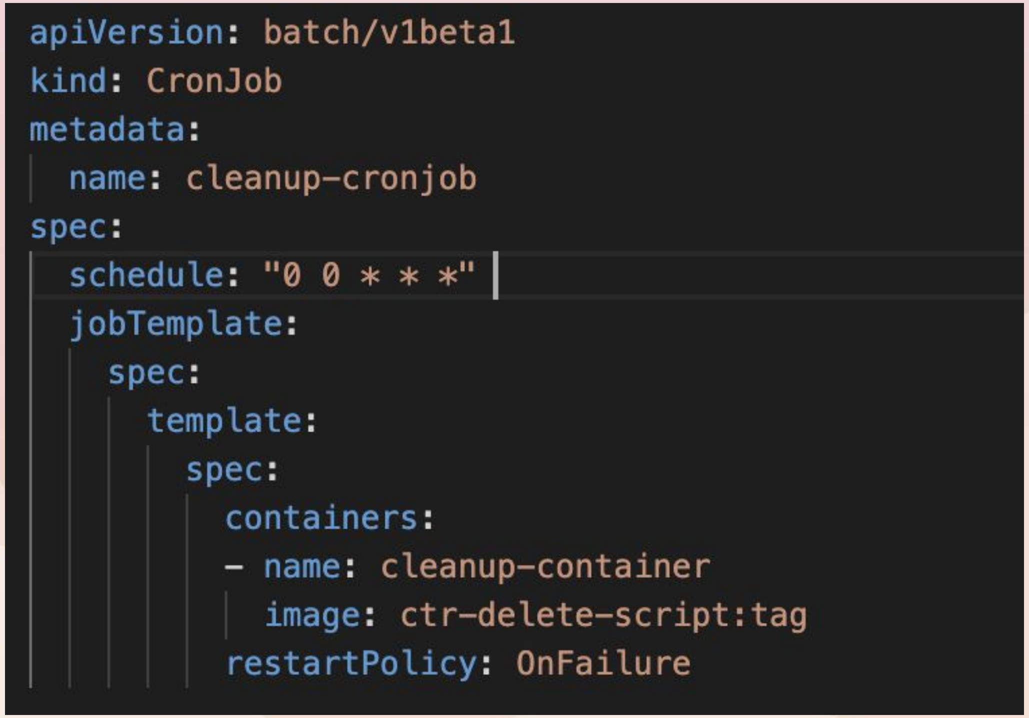
Help Net Security
January 20, 2022

Software supply chain attacks jumped over 300% in 2021

Software supply chain attacks grew by more than 300% in 2021 compared to 2020, according to a study by Argon Security.

2. Eliminate alerts for non-compliant images

3. Conserve developer time

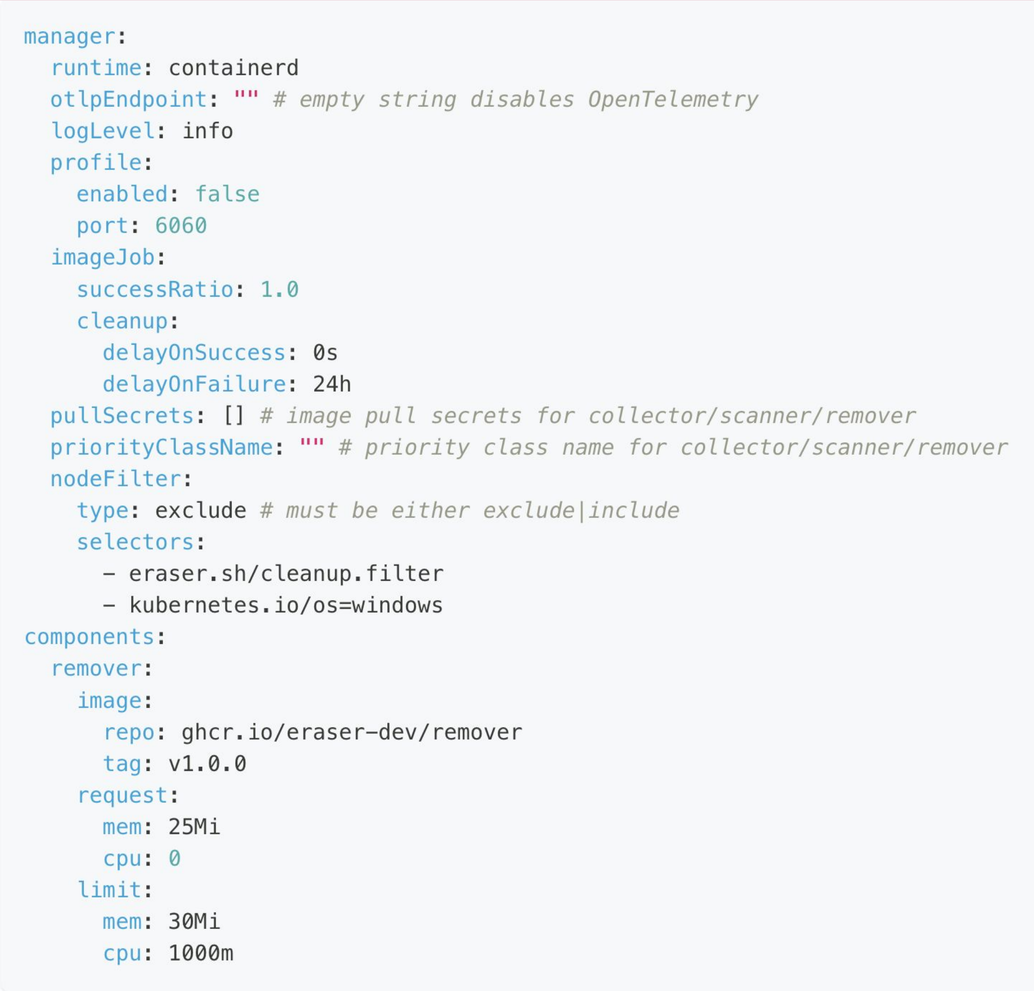


```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: cleanup-cronjob
spec:
  schedule: "0 0 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: cleanup-container
            image: ctr-delete-script:tag
          restartPolicy: OnFailure
```

4. Current Kubernetes garbage collection is inefficient

85% ⟷ 80%

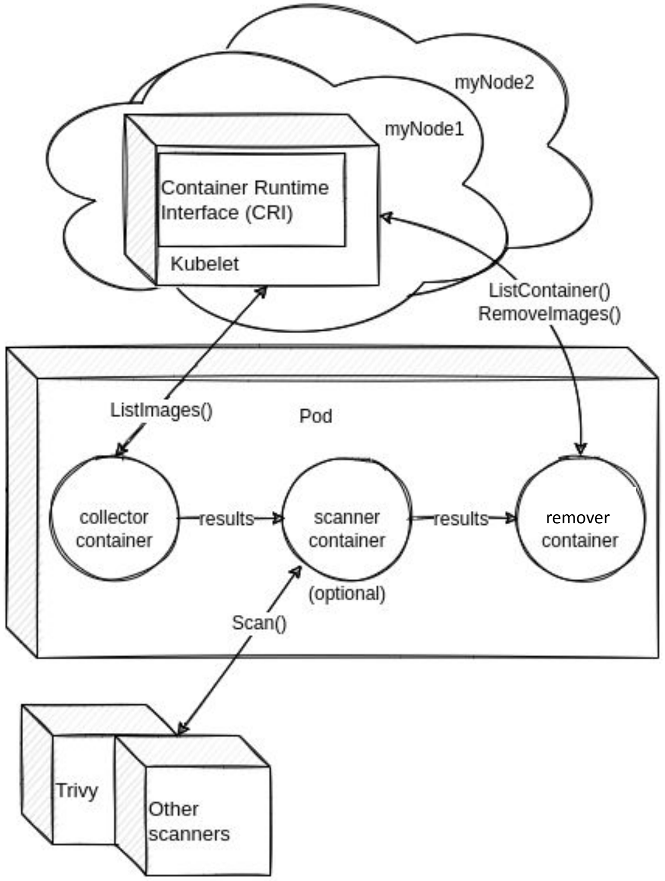
5. Customization features



```yaml
manager:
  runtime: containerd
  otlpEndpoint: "" # empty string disables OpenTelemetry
  logLevel: info
  profile:
    enabled: false
    port: 6060
  imageJob:
    successRatio: 1.0
    cleanup:
      delayOnSuccess: 0s
      delayOnFailure: 24h
  pullSecrets: [] # image pull secrets for collector/scanner/remover
  priorityClassName: "" # priority class name for collector/scanner/remover
  nodeFilter:
    type: exclude # must be either exclude|include
    selectors:
      - eraser.sh/cleanup.filter
      - kubernetes.io/os=windows
components:
  remover:
    image:
      repo: ghcr.io/eraser-dev/remover
      tag: v1.0.0
    request:
      mem: 25Mi
      cpu: 0
    limit:
      mem: 30Mi
      cpu: 1000m
```

ERASER

Architecture



?

?

?

Architecture

Three important questions

1. What images are present on this node?

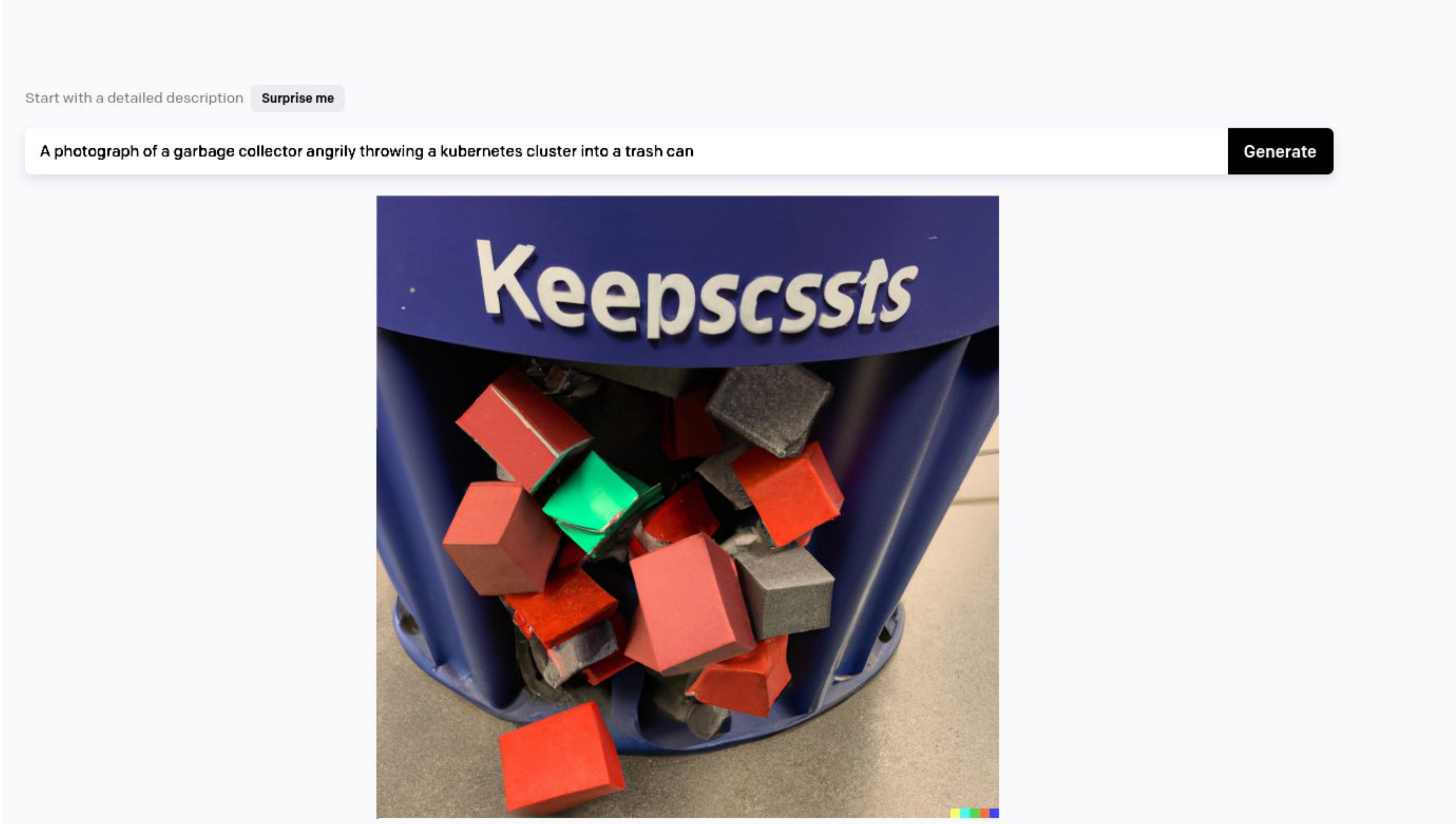
Architecture

Three important questions

1. What images are present on this node?

2. Of those images, which are *not* tied to a container that is currently running?

Architecture



A photograph of a garbage collector angrily throwing a kubernetes cluster into a trash can

Architecture

Three important questions

1. What images are present on this node?
2. Of those images, which are *not* tied to a container that is currently running?
3. Of *those* images, which contain a known CVE?

Architecture

Three important questions

1. What images are present on this node?

2. Of those images, which are *not* tied to a container that is currently running?

3. Of *those* images, which contain a known CVE?

1

“nginx:latest@sha256:0d60ba9…”
“busybox:latest@sha256:02391…”
“alpine:3.7.3@sha256:9225145…”
“alpine:latest@sha256:48d9183…”
“ubuntu:22.04@sha256:c9cf959…”

Architecture

Three important questions

1. What images are present on this node?
2. Of those images, which are *not* tied to a container that is currently running?
3. Of *those* images, which contain a known CVE?

2

“nginx:latest@sha256:0d60ba9…”

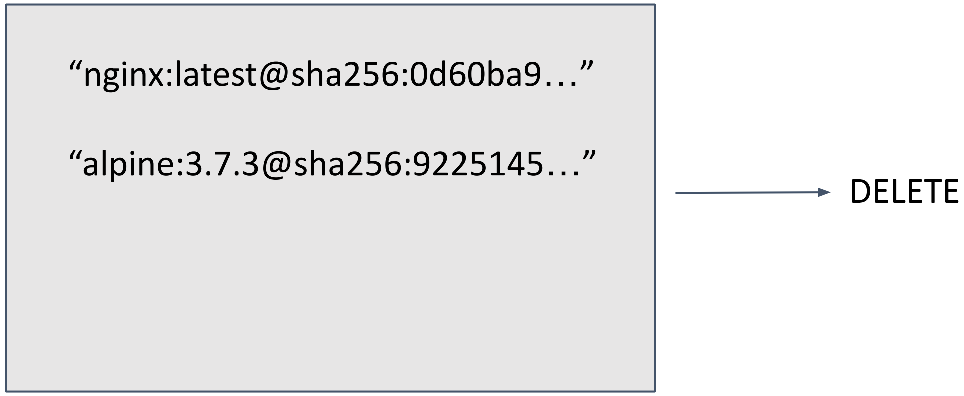
“alpine:3.7.3@sha256:9225145…”
“alpine:latest@sha256:48d9183…”

Architecture

Three important questions

1. What images are present on this node?
2. Of those images, which are *not* tied to a container that is currently running?
3. Of *those* images, which contain a known CVE?

3



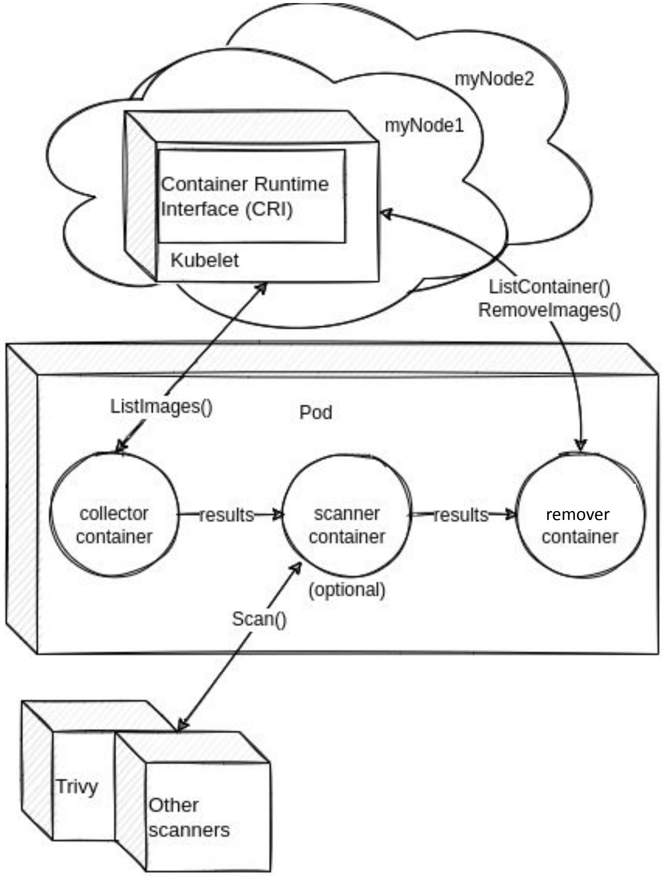
“nginx:latest@sha256:0d60ba9…”

“alpine:3.7.3@sha256:9225145…”

DELETE

Architecture

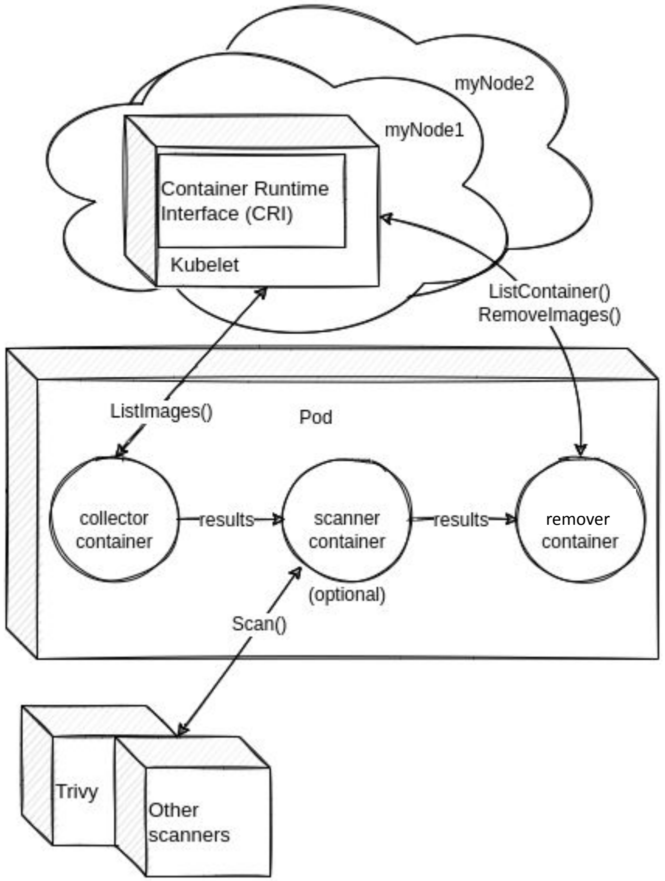
Three important questions



Architecture

Three important questions

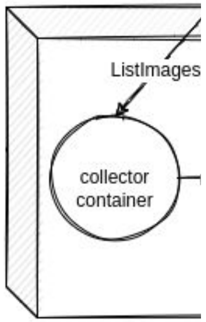
1. What images are present on this node?
2. Of those images, which are *not* tied to a container that is currently running?
3. Of *those* images, which contain a known CVE?



Architecture

Three important questions

1. What images are present on this node?
2. Of those images, which are *not* tied to a container that is currently running?

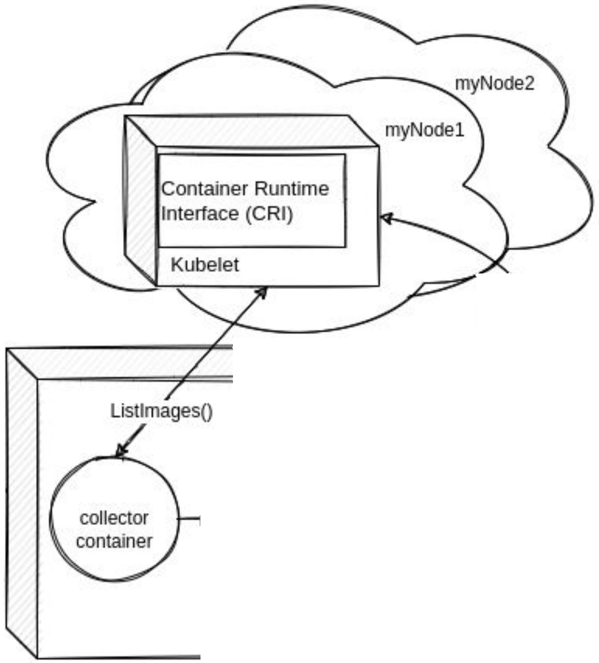


Architecture

Three important questions

1. What images are present on this node?

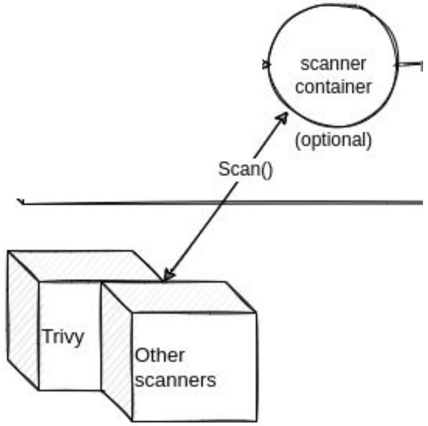
2. Of those images, which are *not* tied to a container that is currently running?



Architecture

Three important questions

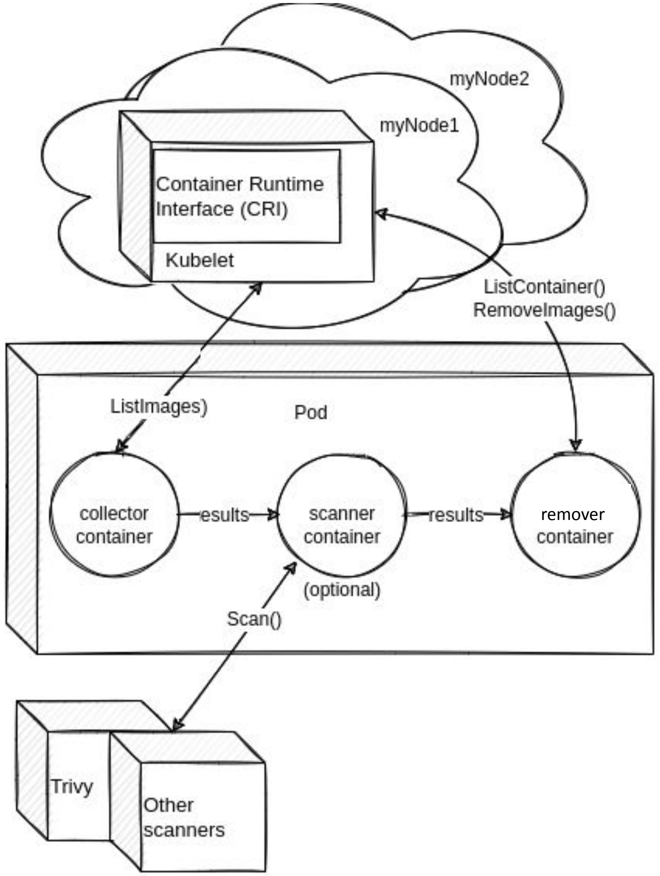
3. Of *those* images, which contain a known CVE?



Architecture

Three important questions

1. What images are present on this node?
2. Of those images, which are *not* tied to a container that is currently running?
3. Of *those* images, which contain a known CVE?



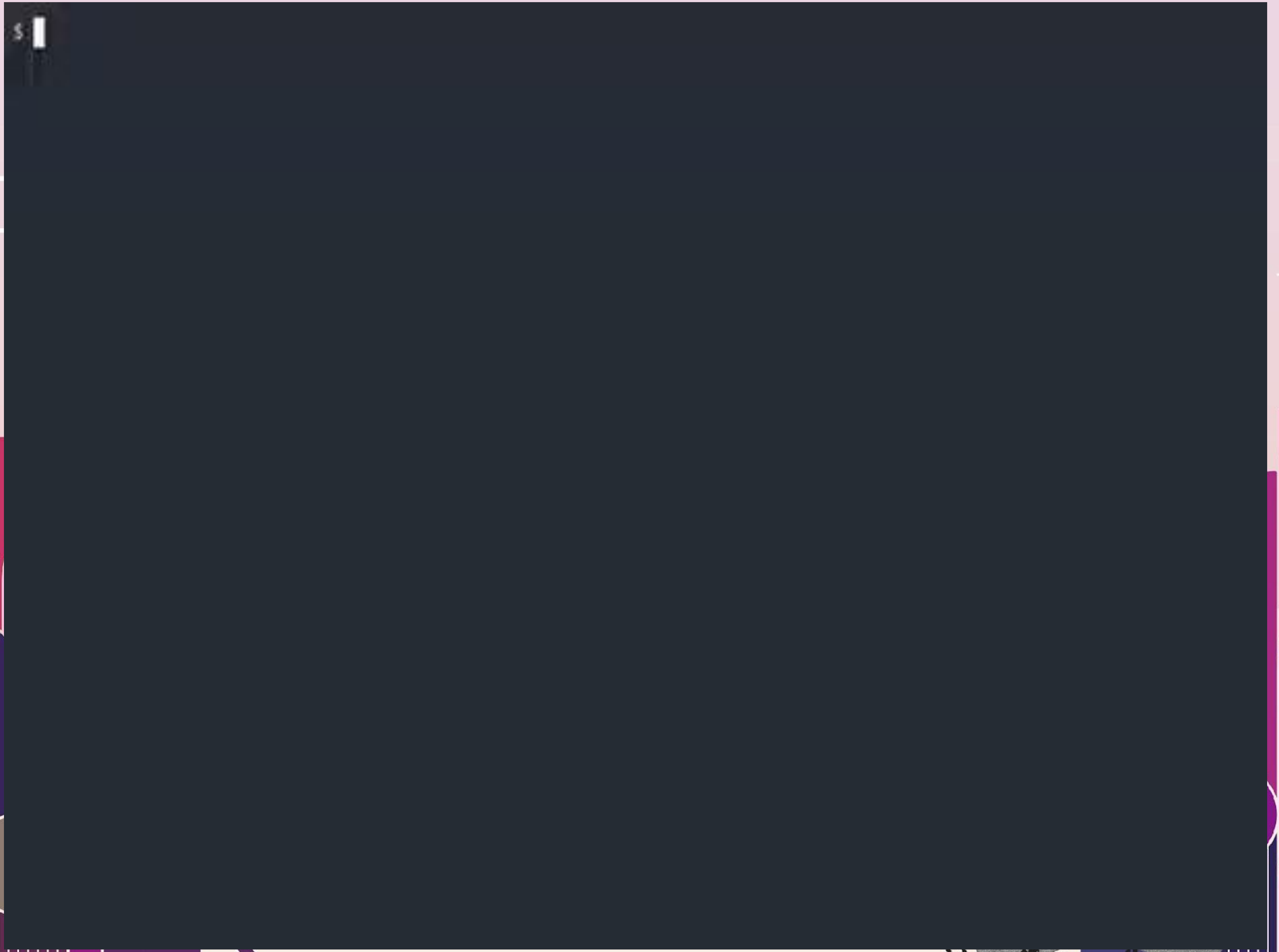
Demo 1

Cleaning up Images on Demand

```
✓ pme@thinkpad : .../go/eraser (demo)
$ 
```

Demo 2

Clean up Images Periodically



```
$
```

Future Work

- Surface vulnerable images
- Stagger the load on the cluster by running jobs in waves
- Support for CRI “Pinned” images
- Custom locations for the runtime socket
- Wider Adoption

How to Get Involved

- Contributing Guide: https://github.com/eraser-dev/eraser#contributing
- Slack Channel: https://kubernetes.slack.com/archives/C03Q8KV8YQ4
- GitHub Repo: https://github.com/eraser-dev/eraser
- Scanner Template Repo: https://github.com/eraser-dev/eraser-scanner-template/

Thank You!

Session QR Codes will be
sent via email before the event

**Please scan the QR Code above
to leave feedback on this session**