Container Security With Clair

Louis DeLosSantos <ldeloss@redhat.com>
Hank Donnay <hdonnay@redhat.com>
Ales Raszka <araszka@redhat.com>

Introduction

What is Clair

- A scalable service for container security.
- Open Source
- Community Developed

Why Clair matters

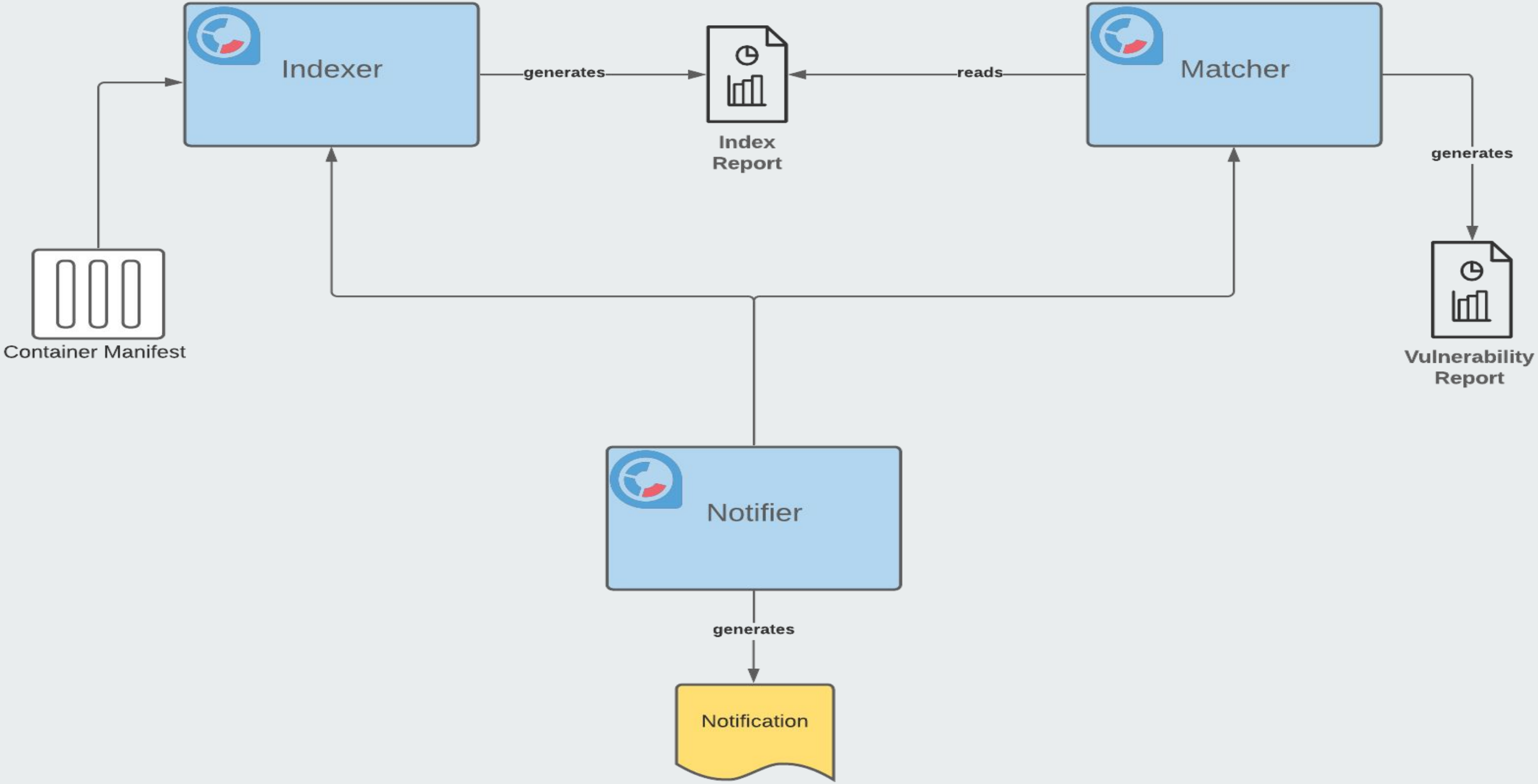
- Containers becoming the default way to ship applications.
- Vulnerable software components come along for the ride.
- Attack vectors now migrate with containers.

What's new in Clair V4

- Manifest focused API
- Content Addressability*
- Updated security advisory data sources
- Native CLI tool
- Updated notifications subsystem
- Language package support (python)
- Redesigned microservice architecture
- Redesigned with scale and performance

How Clair Works

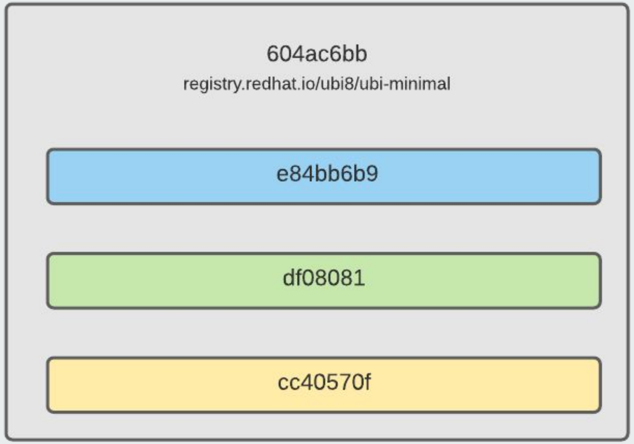
Overview


Indexer
generates
Index Report
reads
Matcher
generates
Vulnerability Report
Container Manifest
Notifier
generates
Notification

Indexing

Extracting the contents of a container

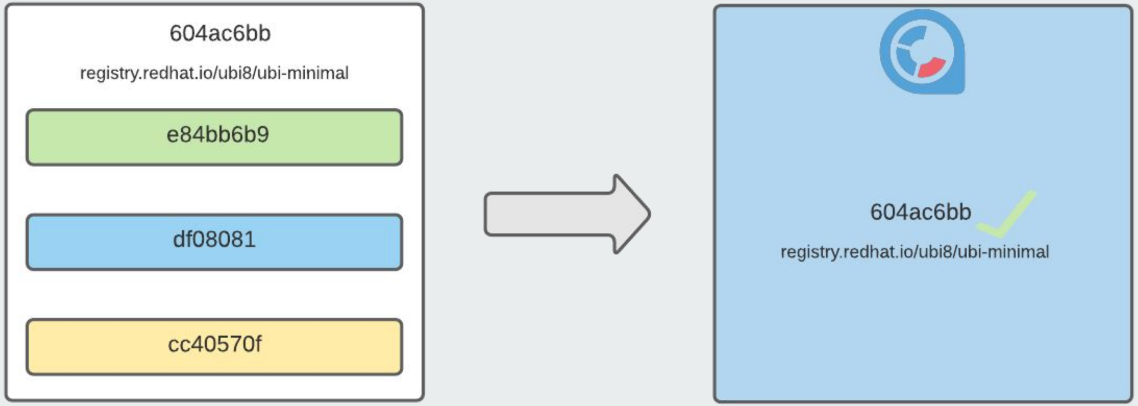
What is a container image



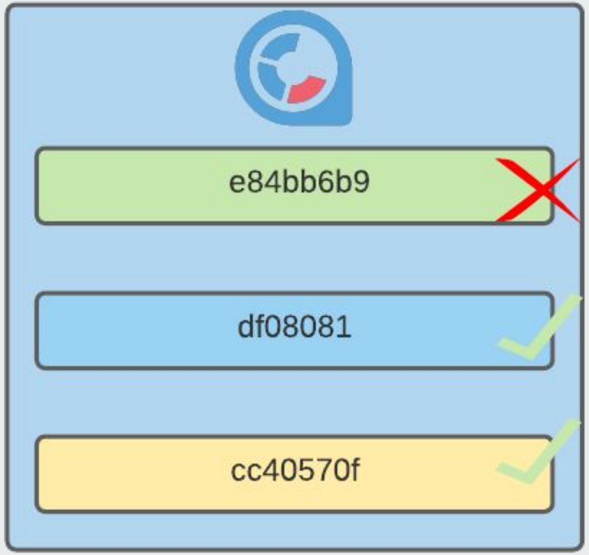
- A content addressable hash identifying the image as a whole
- A series of order dependent layers identified by a content addressable hash
- Each layer is a diff of filesystem contents.

How Indexing Works

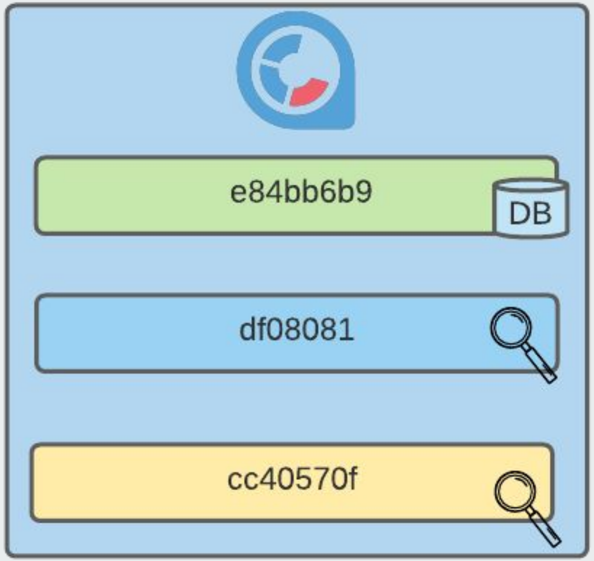
Determine if manifest should be scanned


604ac6bb
registry.redhat.io/ubi8/ubi-minimal
e84bb6b9
df08081
cc40570f
604ac6bb ✓
registry.redhat.io/ubi8/ubi-minimal

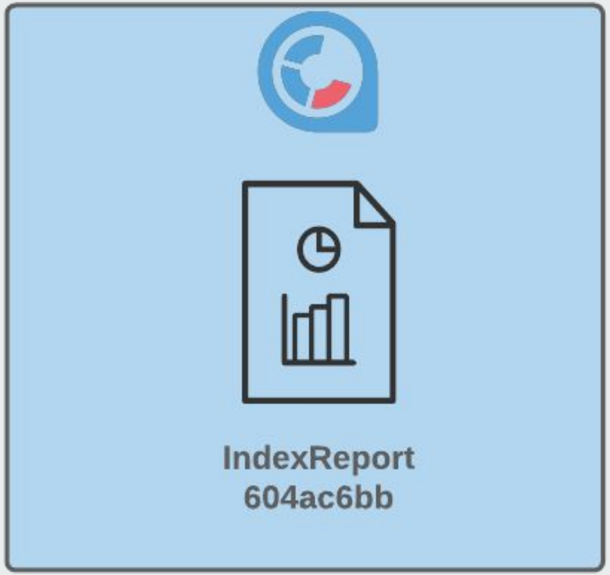
Determine layers to scan



Scan only necessary layers



Generate Index Report



Matching

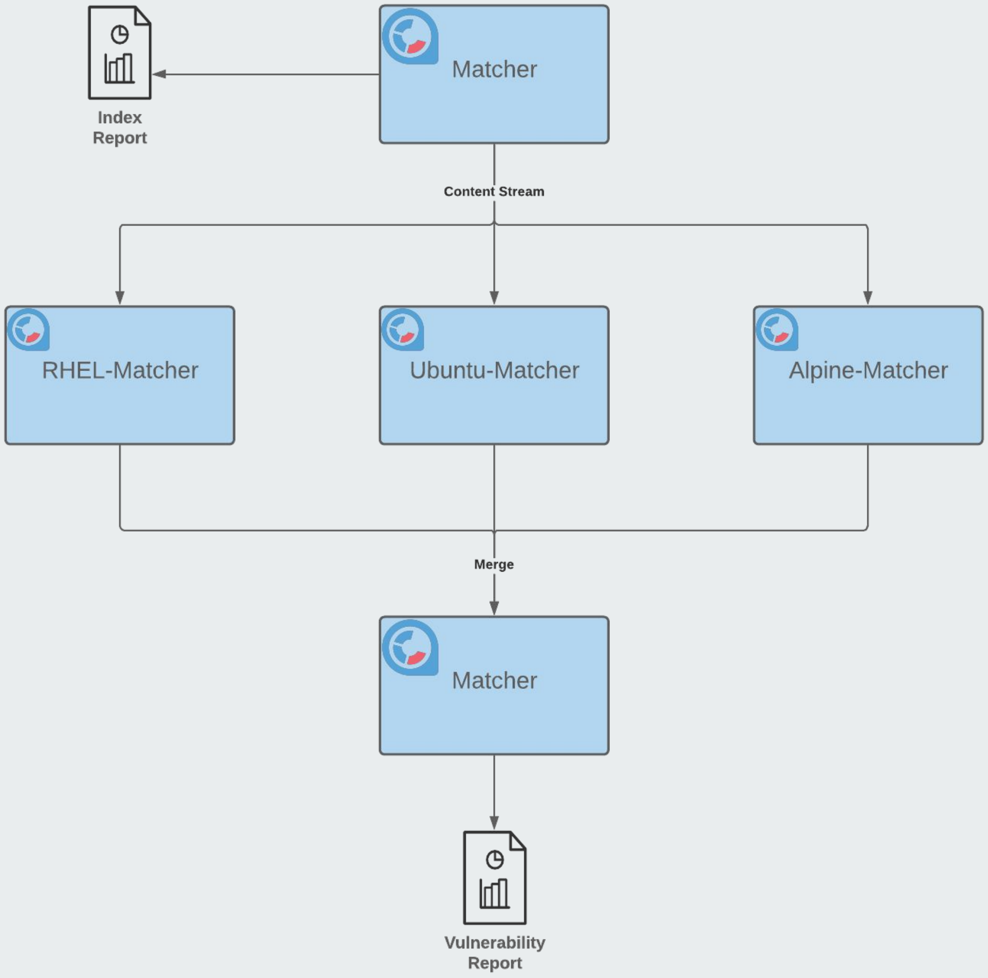
Identifying vulnerable container content

Matchers and Updaters

- Updaters download, parse, and load security advisories into Clair
- Matchers identify vulnerable container contents

How Matching works

- Index Report provided to Matcher Service
- Each Matcher impl created in parallel
- Each matcher reads content stream and flags vulnerable content
- Matcher service merges flagged content into final Vulnerability report



How Updaters work

- Updaters are ran on a configurable interval

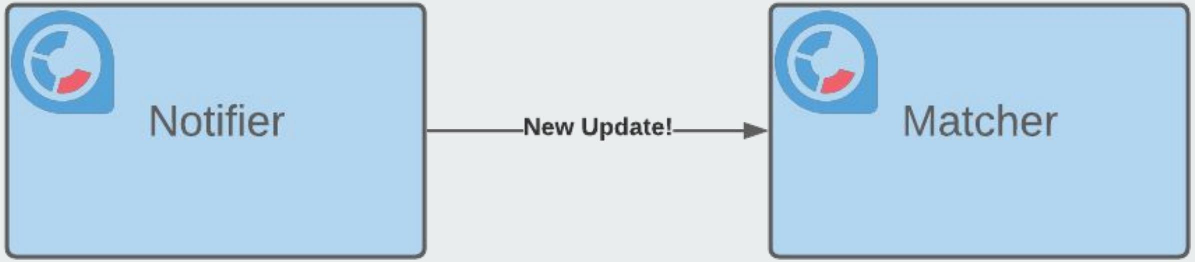
- On each interval, if new content is available it is fetched, parsed, and loaded into Clair

Notifications

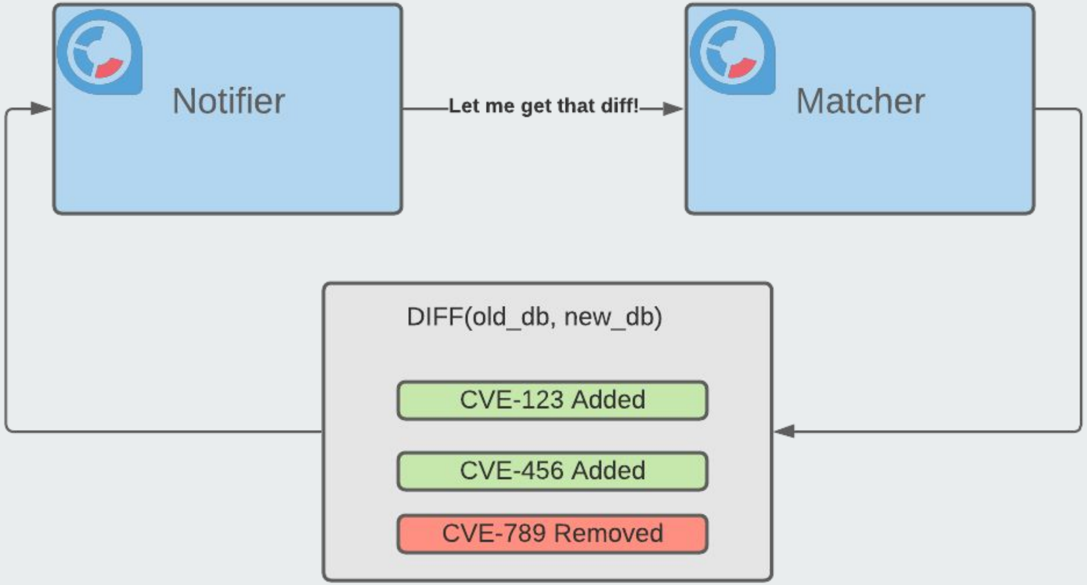
Making clients aware of changes to vulnerable manifests

How Notifications works

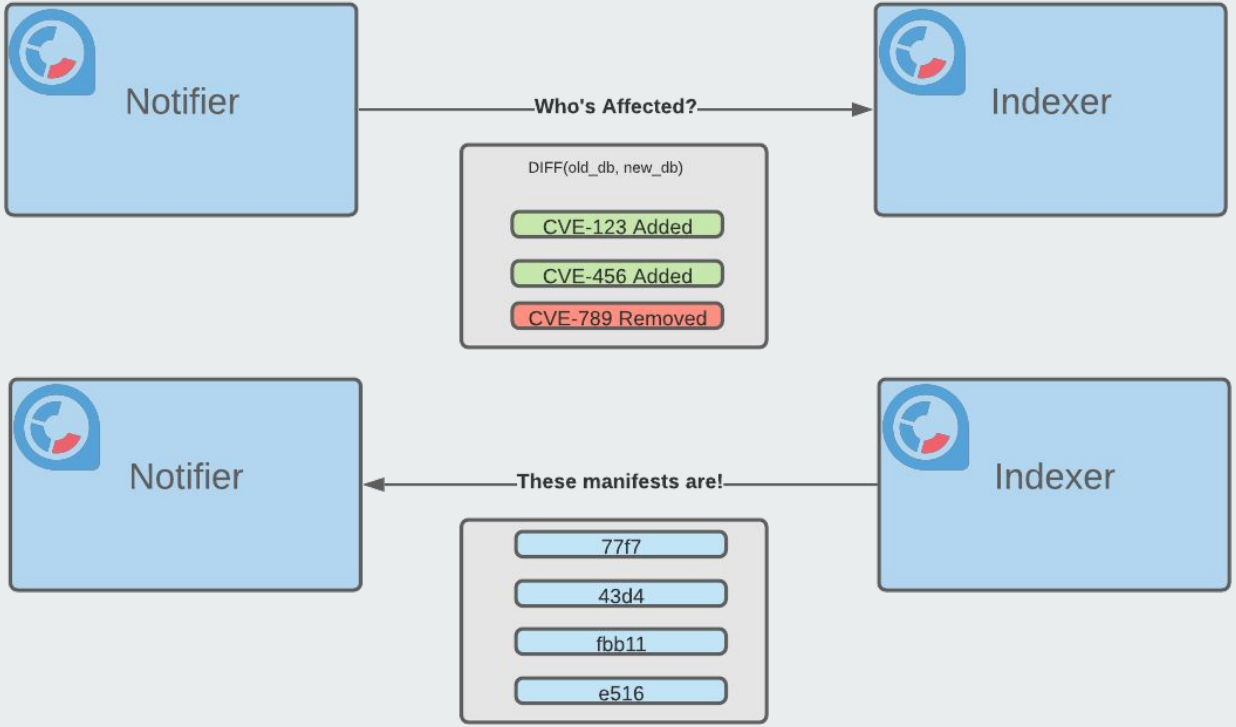
Notifier discovers new update.



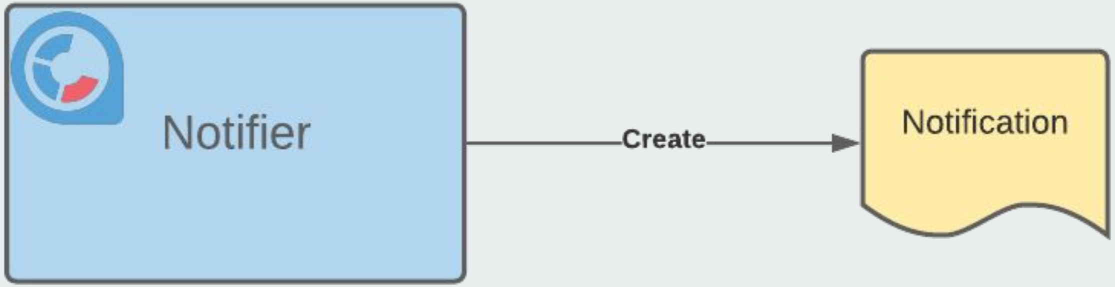
Request Diff



Find Affected Manifests



Generate and deliver notification



Moving from Clair v2 to Clair v4

- Clair V4 API is incompatible with Clair V2
- Images must be re-submitted to Clair V4
- Layer based API deprecated in favor of Manifest driven API

Clair At Red Hat

Quay v3.3 and beyond

- Clair V4 introduced in Quay v3.3
- The default container security tool moving forward
- Clair V2 deprecated as of Quay v3.4

Quay.io

- Clair V4 to power Quay.io end of March 2021

EXD Cloud and Red Hat Container Catalog

- Clair powers the Red Hat Container Catalog
- Clair V4 running at production scale
- Clair security check is part of container release process

Clair v4.1.0 Roadmap

- Expand language package support (Golang, Java, Javascript)
- Kubernetes Operator
- Performance and scale analysis
- Reliability Efforts

Contributing

- Community Development Meetings
  - Biweekly meetings starting March 2021
- Clair mailing list
  - https://groups.google.com/g/clair-dev
- Contact
  - Louis Delossantos <ldelossa@redhat.com>
  - Hank Donnay <hdonnay@redhat.com>
  - Ales Raszka <araszka@redhat.com>
  - Quay team <quay-devel@redhat.com>