



Connect

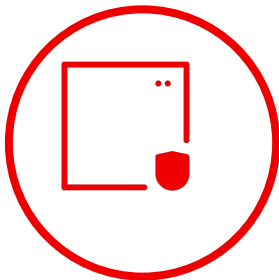
Trust no one : Software pipelines for secure software development

Mark Roberts

Principal Solution Architect, Red Hat, UK

Developers struggle to keep code compliant given high cognitive load

Identify & remediate
malicious **code**



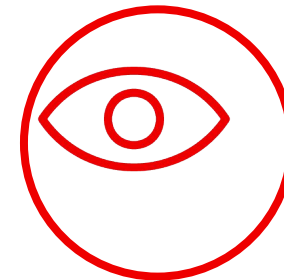
Confidence in
software integrity

Safeguard **build**
systems



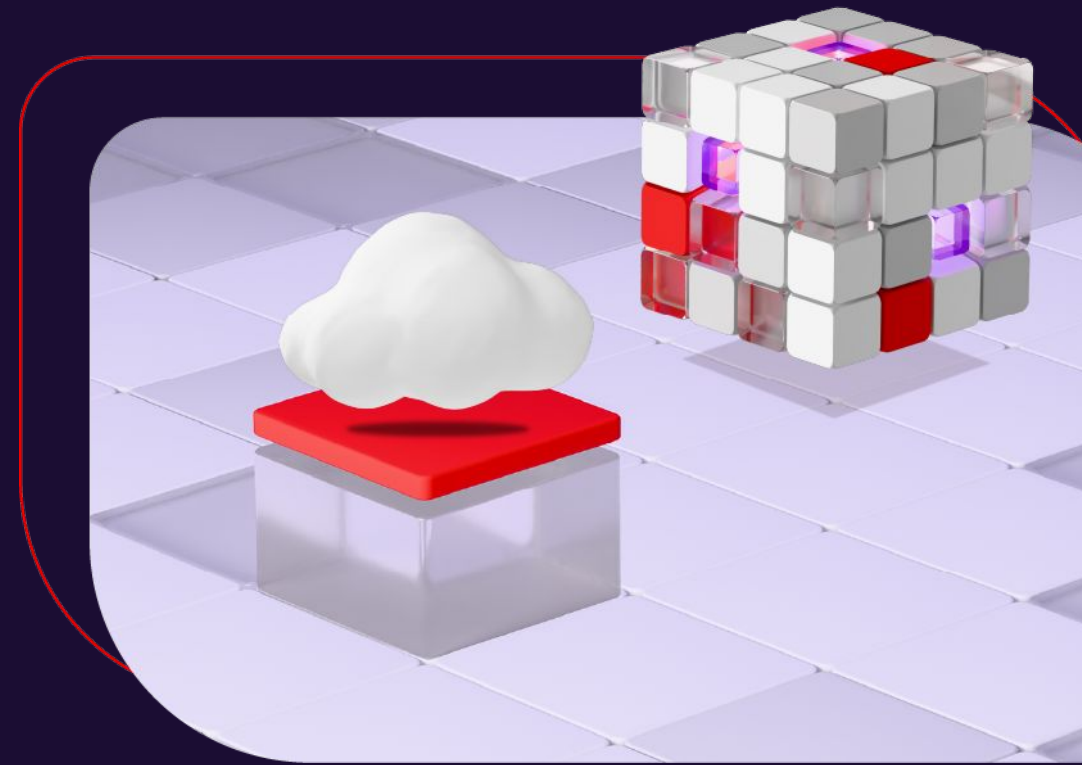
Adherence to
regulation/compliance

Continuously **monitor**
applications at runtime



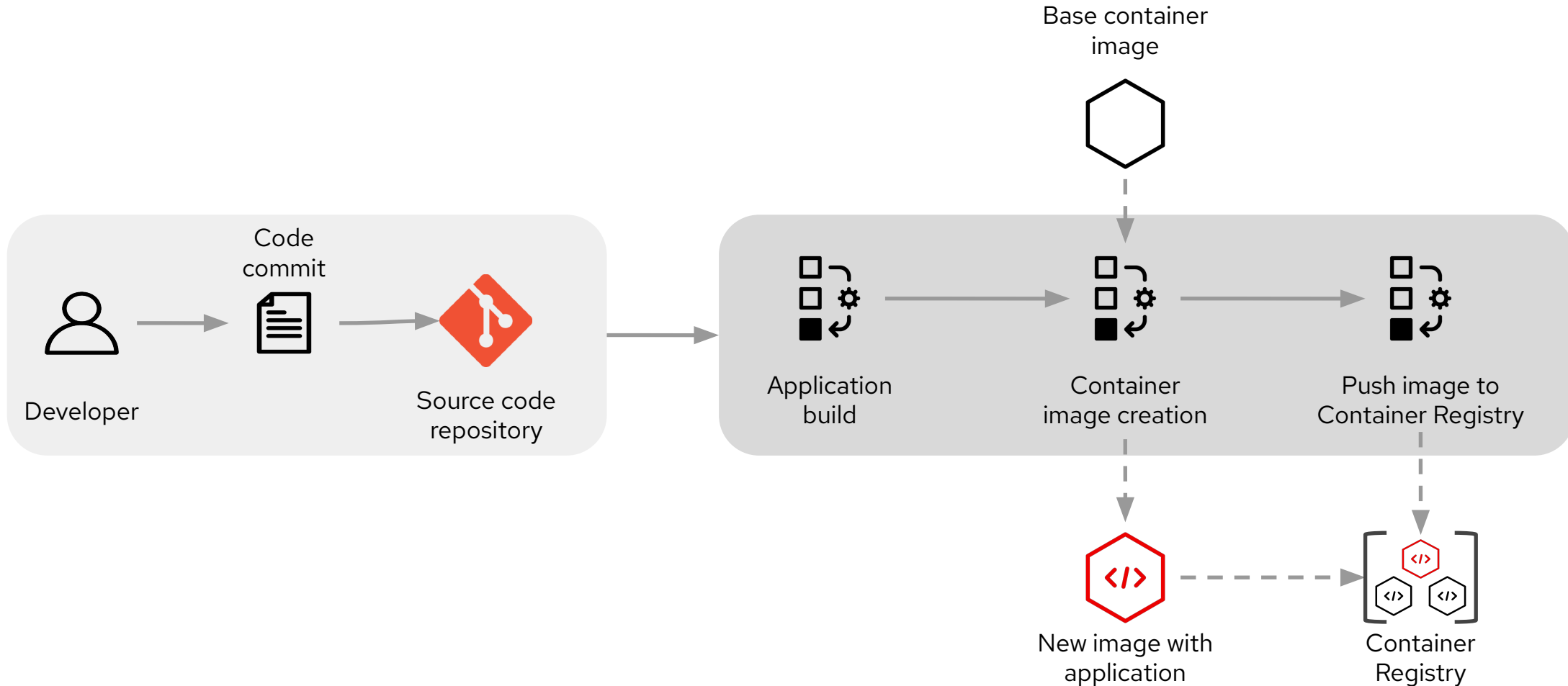
Enhanced trust with
customers/stakeholders

Code, build, deploy &
monitor to create a
Trusted Software
Supply Chain



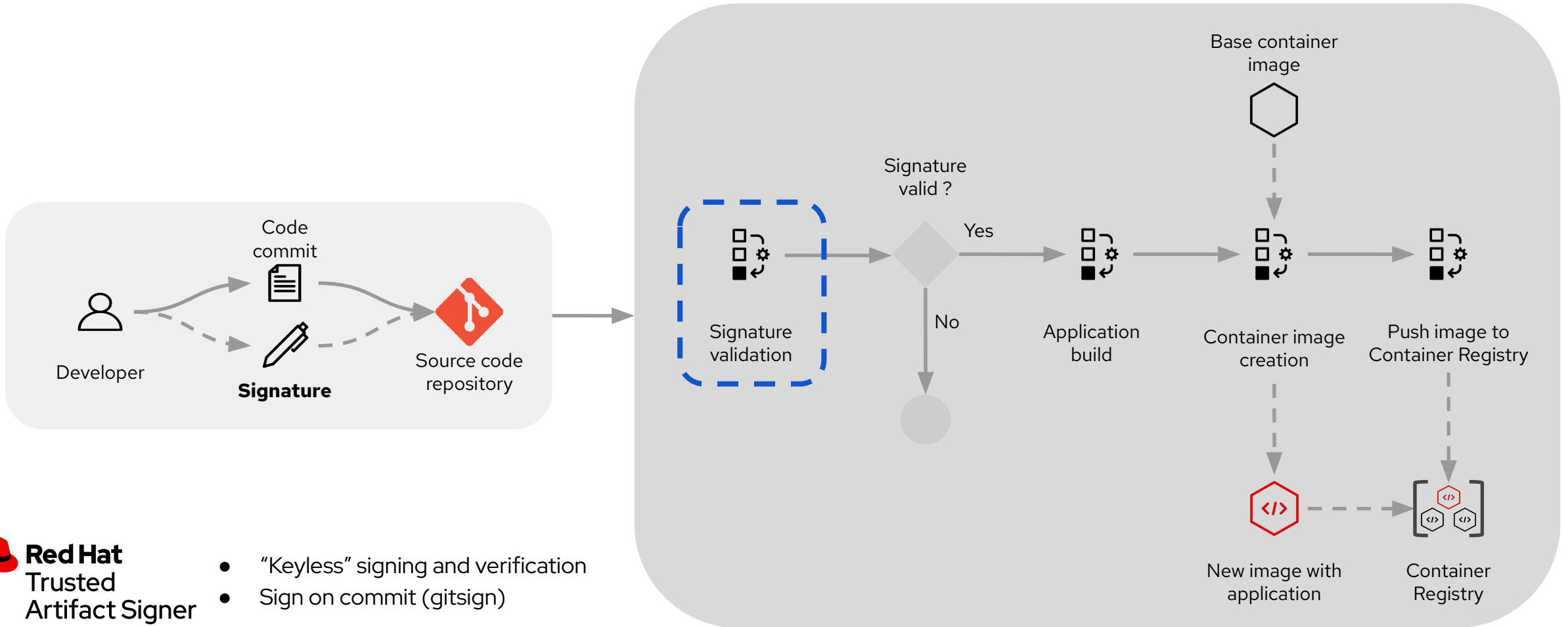
Securing the software supply chain

A simple build process



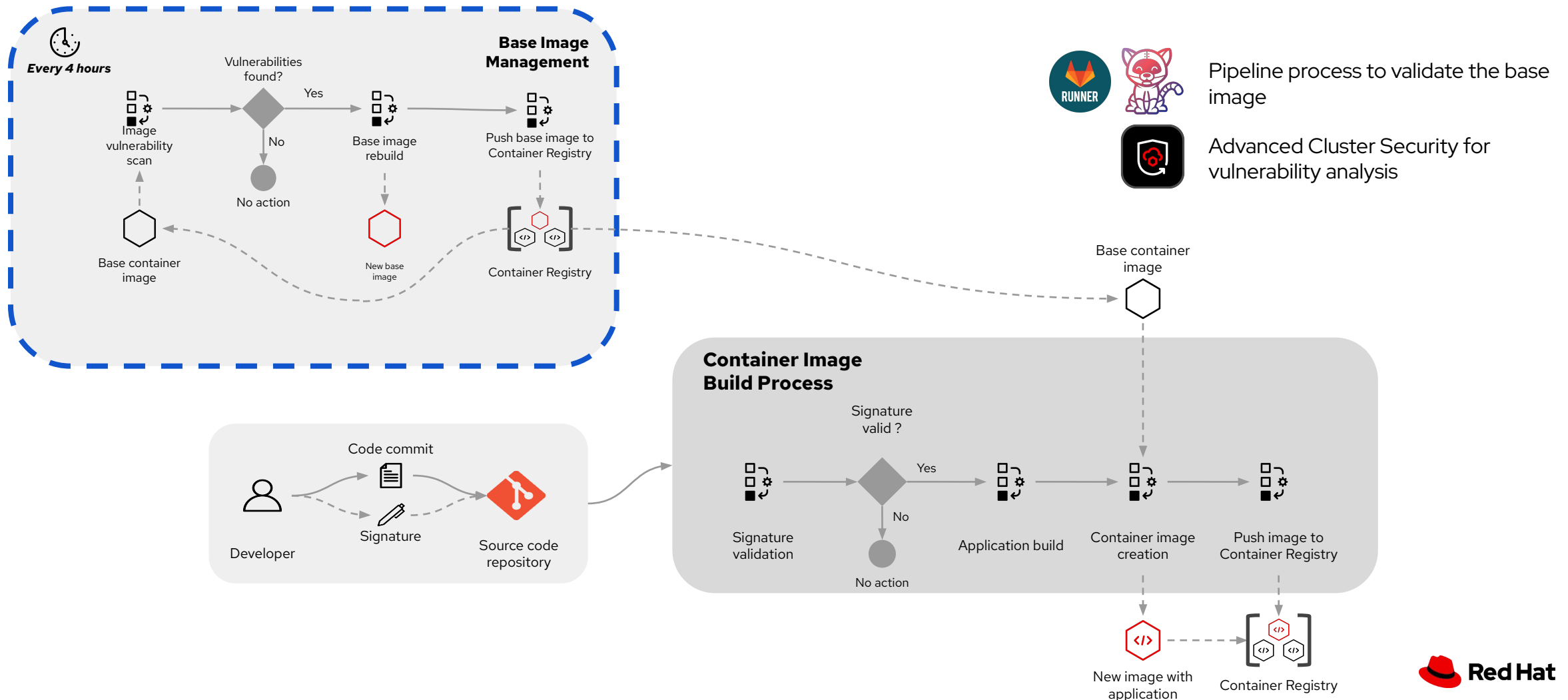
Securing the software supply chain

Signature validation - ensure commits are acceptable and genuine



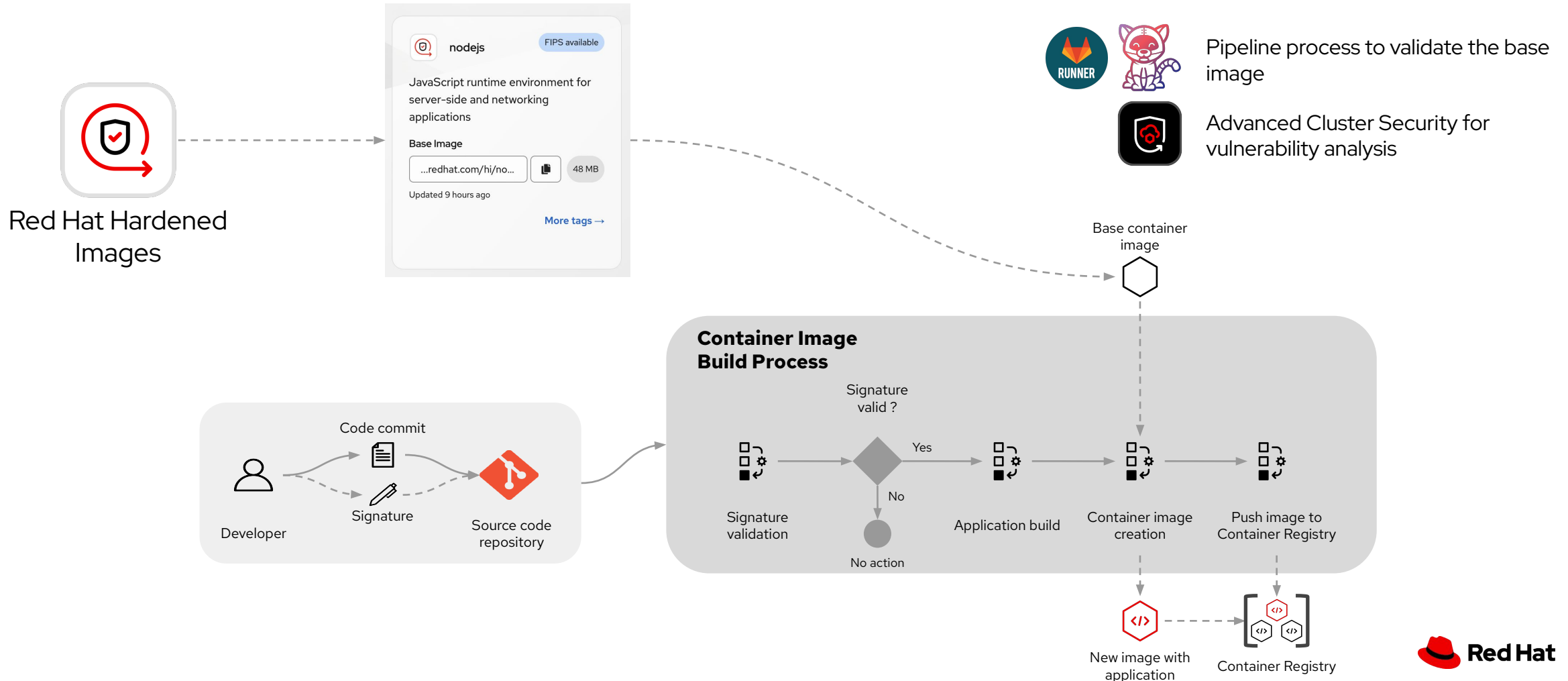
Securing the software supply chain

Ensure the base image is vulnerability free



Securing the software supply chain

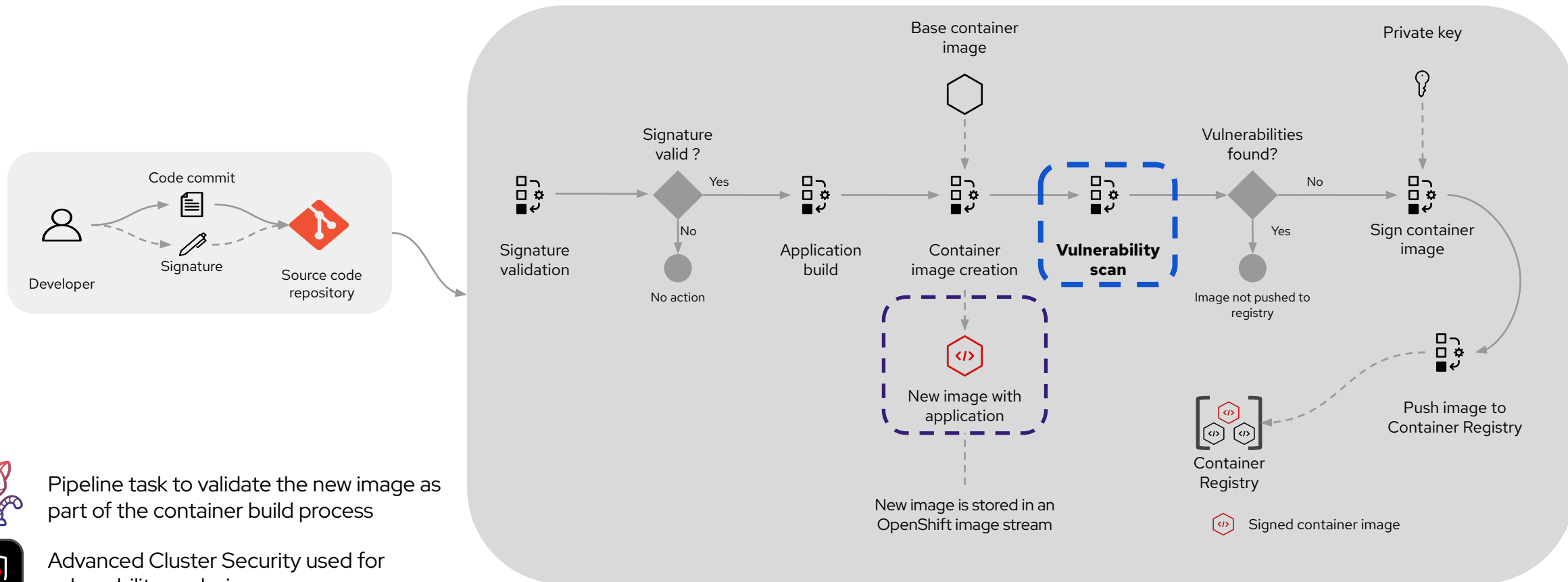
Don't manage the base image



Securing the software supply chain

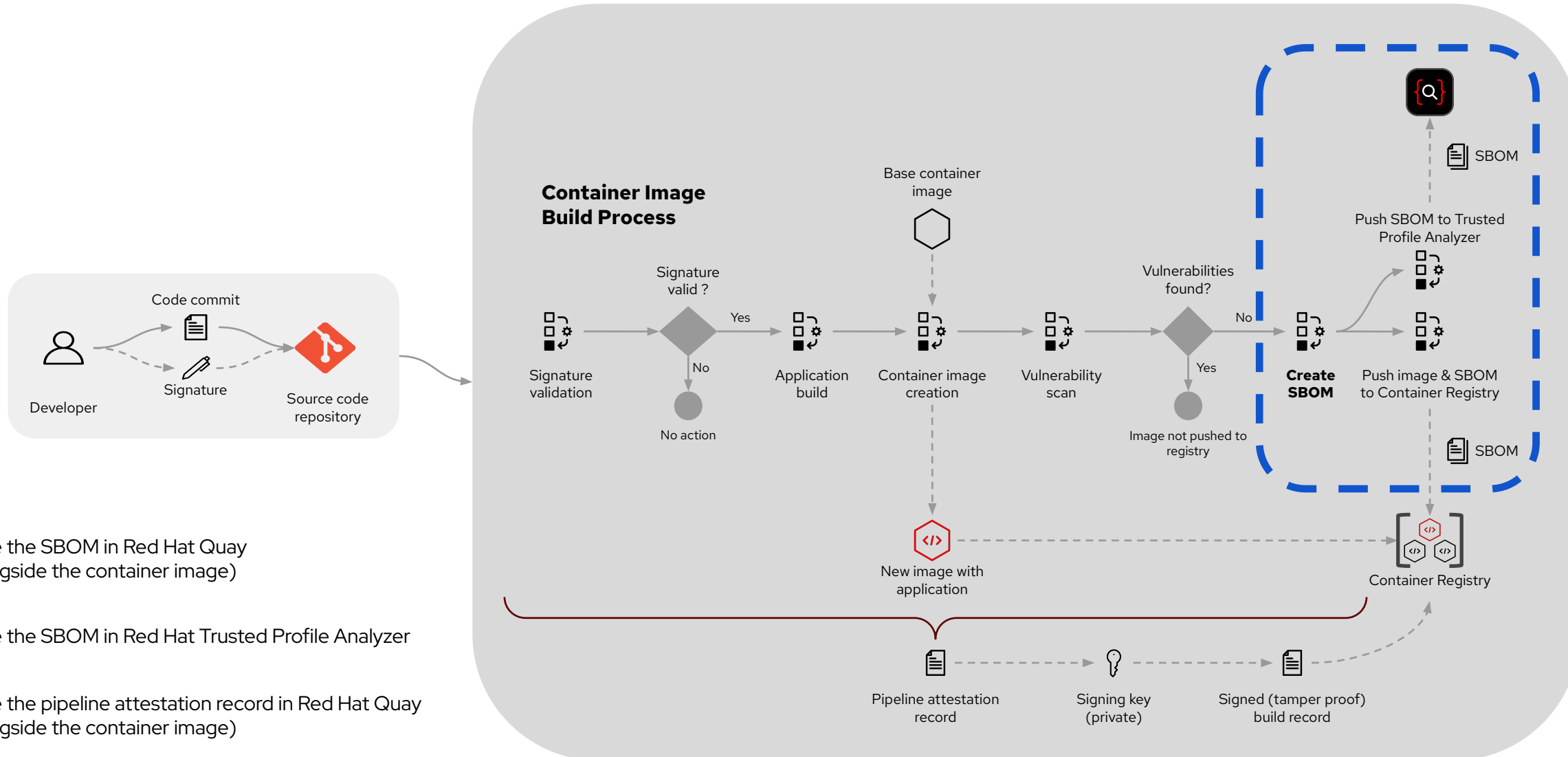
Ensure the new image has no vulnerabilities introduced by the build process

- Image is stored in a temporary OpenShift image stream to allow time for vulnerability scan
- If clean it is then pushed to the enterprise container repository



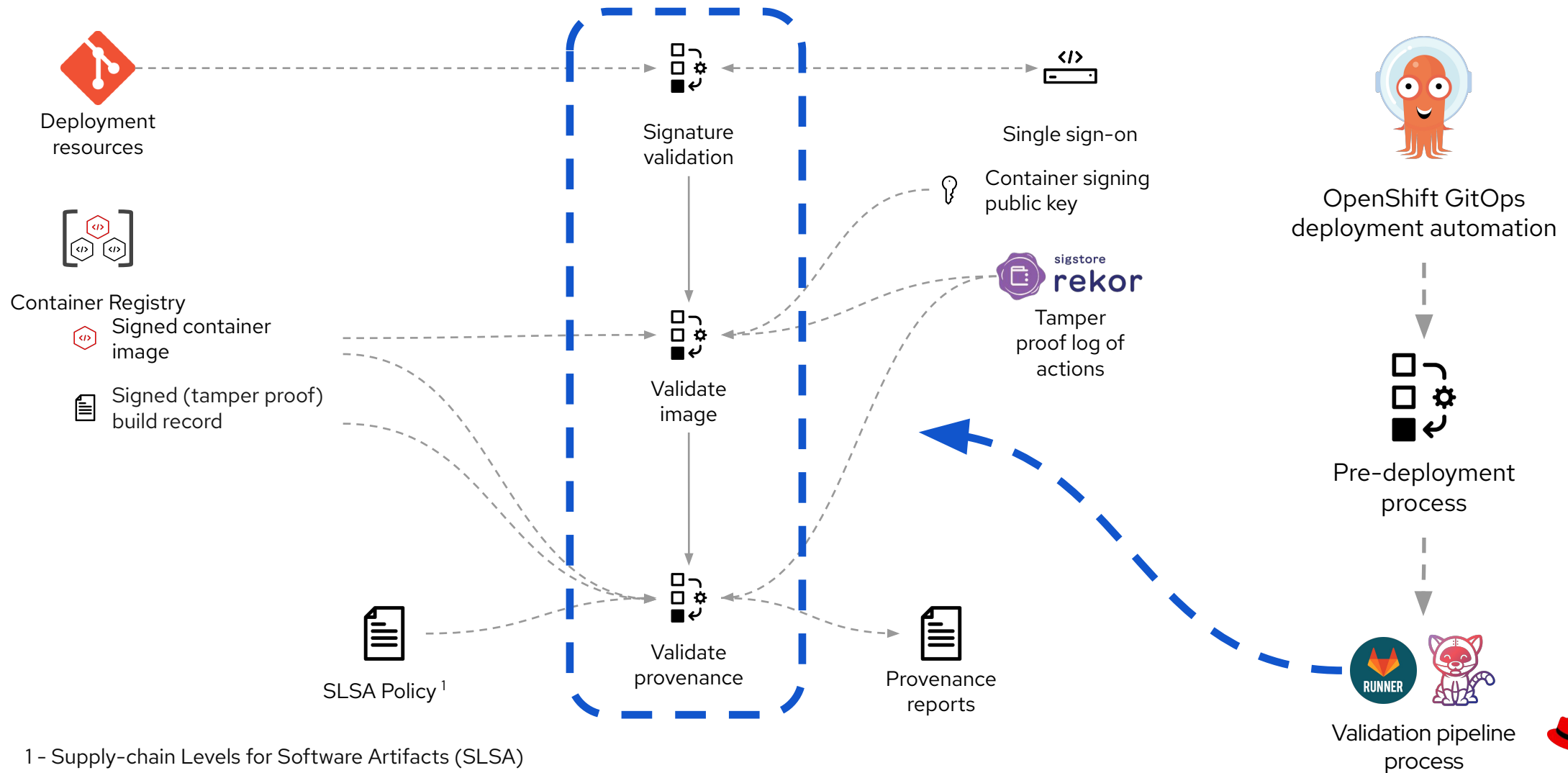
Securing the software supply chain

Creating and storing a Software-bill-of-materials



Securing the software supply chain

Pre-deployment validation



SLSA Provenance validation

- Where did the artifacts come from ?
- What sources were used ?
- How was it built ?
 - Validate that GitLab Runners was the build platform
 - Validate that a vulnerability scan was performed against a specific ACS platform
 - Validate that required source code analysis was done
 - Validate that the deployment resources have been reviewed for policy violations
 - Examine and validate SBOMs
 - Did a task run between the clone and compile stage?

```
{
  "_type": "https://in-toto.io/Statement/v0.1",
  "predicateType": "https://slsa.dev/provenance/v0.2",
  "predicate": {
    "builder": {
      "id": "<URI>"
    },
    "buildType": "<URI>",
    "parameters": { },
    "environment": { }
  },
  "buildConfig": { },
  "metadata": {
    "buildInvocationId": "<STRING>",
    "buildStartedOn": "<TIMESTAMP>",
    "buildFinishedOn": "<TIMESTAMP>",
    "completeness": {
      "parameters": true/false,
      "environment": true/false,
      "materials": true/false
    },
    "reproducible": true/false
  }
}
```

Securing the software supply chain

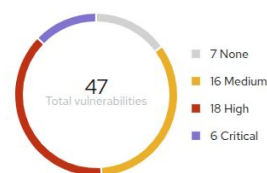
Give the development team the right tools for the task

RHEL-9.0 SPDX

To generate a report, SBOM content will be processed by Red Hat's Dependency Analytics Service. The SBOM data will be deleted from the Dependency Analytics Service once the report has been generated. Contact rhtpa-support@redhat.com if you have any questions or concerns.

[Download](#)

[Info](#) [Packages](#) [Vulnerabilities](#) [Dependency Analytics Report](#)



Name
RHEL-9.0

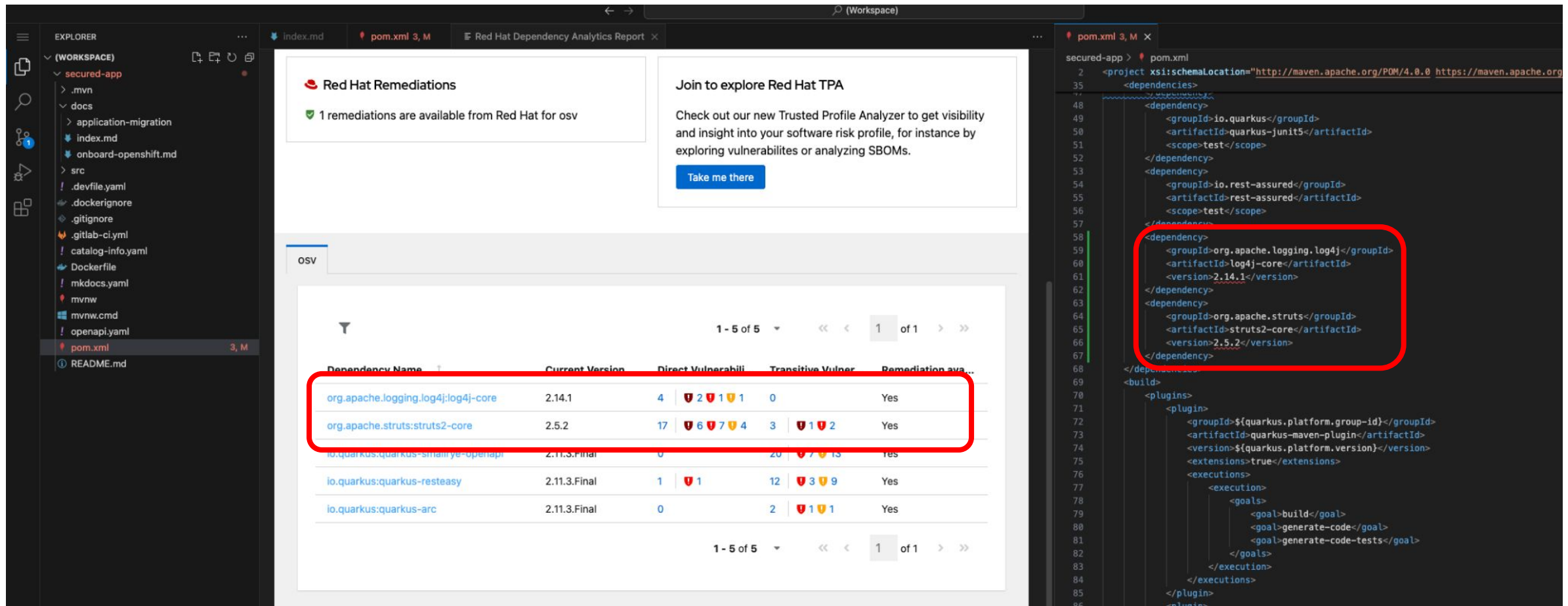
Creation date
Apr 01, 2025

1 - 10 of 47					
1 of 5					
Id ↑	Description	CVSS ↓	Affected dependencies ↓	Published ↓	Updated ↓
CVE-2017-15133	A denial of service flaw was found in miekg-dns before 1.0.4. A remote attacker could use carefully timed TCP packets to block the DNS server from accepting new connections.	High 75/10	1	Jan 29, 2018	Sep 16, 2024
CVE-2017-16028	react-native-meteor-oauth is a library for OAuth2 login to a Meteor server in React Native. The oauth Random Token is generated using a non-cryptographically strong RNG (Math.random()).	0.0/10	1	Jun 04, 2018	Sep 17, 2024
CVE-2017-16137	The debug module is vulnerable to regular expression denial of service when untrusted user input is passed into the o formatter. It takes around 50k characters to block for 2 seconds making this a...	Medium 5.3/10	3	Jun 07, 2018	Sep 16, 2024
Show more					

Securing the software supply chain

Give the development team the right tools for the task

- Dependency analysis information available directly from the editor such as VS Code



The screenshot displays the Red Hat Dependency Analytics Report integrated into the VS Code editor. The report shows a table of dependencies with their current versions and vulnerability status. Two dependencies are highlighted with a red box: `org.apache.logging.log4j:log4j-core` (version 2.14.1) and `org.apache.struts:struts2-core` (version 2.5.2). Both are marked as having direct vulnerabilities. The right pane shows the corresponding `pom.xml` file, with the same two dependencies highlighted by a red box, demonstrating the direct link between the code and the analysis report.

Red Hat Remediations

1 remediations are available from Red Hat for osv

Join to explore Red Hat TPA

Check out our new Trusted Profile Analyzer to get visibility and insight into your software risk profile, for instance by exploring vulnerabilities or analyzing SBOMs.

[Take me there](#)

OSV

Dependency Name	Current Version	Direct Vulnerability	Transitive Vulnerability	Remediation available
<code>org.apache.logging.log4j:log4j-core</code>	2.14.1	4	0	Yes
<code>org.apache.struts:struts2-core</code>	2.5.2	17	3	Yes
<code>io.quarkus:quarkus-smallrye-openapi</code>	2.11.3.Final	0	20	Yes
<code>io.quarkus:quarkus-resteasy</code>	2.11.3.Final	1	12	Yes
<code>io.quarkus:quarkus-arc</code>	2.11.3.Final	0	2	Yes

```

2 <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org
35 <dependencies>
48 <dependency>
49 <groupId>io.quarkus</groupId>
50 <artifactId>quarkus-junit5</artifactId>
51 <scope>test</scope>
52 </dependency>
53 <dependency>
54 <groupId>io.rest-assured</groupId>
55 <artifactId>rest-assured</artifactId>
56 <scope>test</scope>
57 </dependency>
58 <dependency>
59 <groupId>org.apache.logging.log4j</groupId>
60 <artifactId>log4j-core</artifactId>
61 <version>2.14.1</version>
62 </dependency>
63 <dependency>
64 <groupId>org.apache.struts</groupId>
65 <artifactId>struts2-core</artifactId>
66 <version>2.5.2</version>
67 </dependency>
68 </dependencies>
69 <build>
70 <plugins>
71 <plugin>
72 <groupId>${quarkus.platform.group-id}</groupId>
73 <artifactId>quarkus-maven-plugin</artifactId>
74 <version>${quarkus.platform.version}</version>
75 <extensions>true</extensions>
76 <executions>
77 <execution>
78 <goals>
79 <goal>build</goal>
80 <goal>generate-code</goal>
81 <goal>generate-code-tests</goal>
82 </goals>
83 </execution>
84 </executions>
85 </plugin>
86 </plugins>
  
```

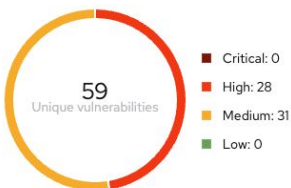
Assess open source license implications

⚠ Red Hat overview of security issues

Vendor Issues

Below is a list of dependencies affected with CVE.

rhtpa/osv-github

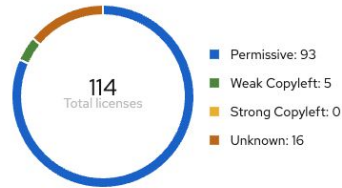


rhtpa/redhat-csaf



License Summary

deps.dev

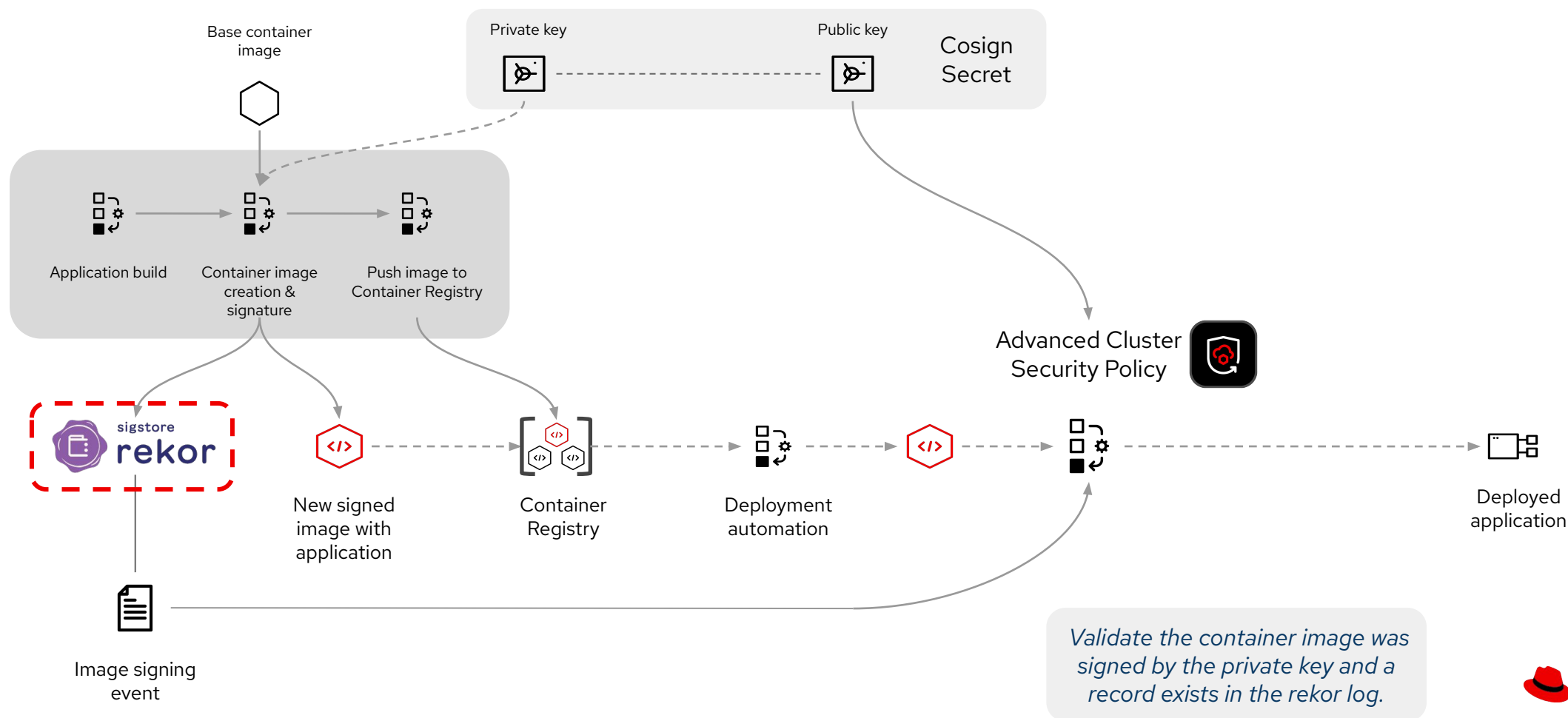


- rhtpa/osv-github
 - Github security advisory formatted in the OSV schema
- rhtpa/redhat-csaf
 - Red Hat security advisories
 - Common Security Advisory Framework (CSAF) & Vulnerability Exploitability Exchange (VEX) data
 - Contextual information - fixed, known affected or known not affected
- License Summary
 - Permissive - Freely use, modify and distribute
 - Weak copyleft - Requirement to open-source modifications made to the library
 - Strong copyleft - **Your entire application source must be made open-source**
 - Unknown - Investigate !
 - (Other conditions and rules apply - read the details)



Deploy only validated container images

Secure the production environment



Deploy only validated container images

Advanced Cluster Security Policy

Rules
Chain criteria with boolean logic.

1 Rule 1

Require image signature
Image must be signed by a trusted signer:

Selected 1 trusted image signer Select

— and —

Drop a policy field inside

Add trusted image signers to policy criteria

Select trusted image signers from the table below.

☐ Name	Verification methods
☒ Cosign-rekor	Cosign
☐ Red Hat	Cosign

Save Cancel

Cosign-rekor

Integration name ^{*}

Cosign-rekor

> Cosign public keys

> Cosign certificates

▼ Transparency log

☒ Enable transparency log validation
Validate the inclusion of the signature in a transparency log.
Required when signatures contain short-lived certificates as they are issued by Fulcio.

Rekor URL

http://rekor-server.trusted-artifact-signer.svc.cluster.local

The URL under which the Rekor transparency log is available. Defaults to the public Rekor instance of Sigstore.
Required for online confirmation of the inclusion into the transparency log.

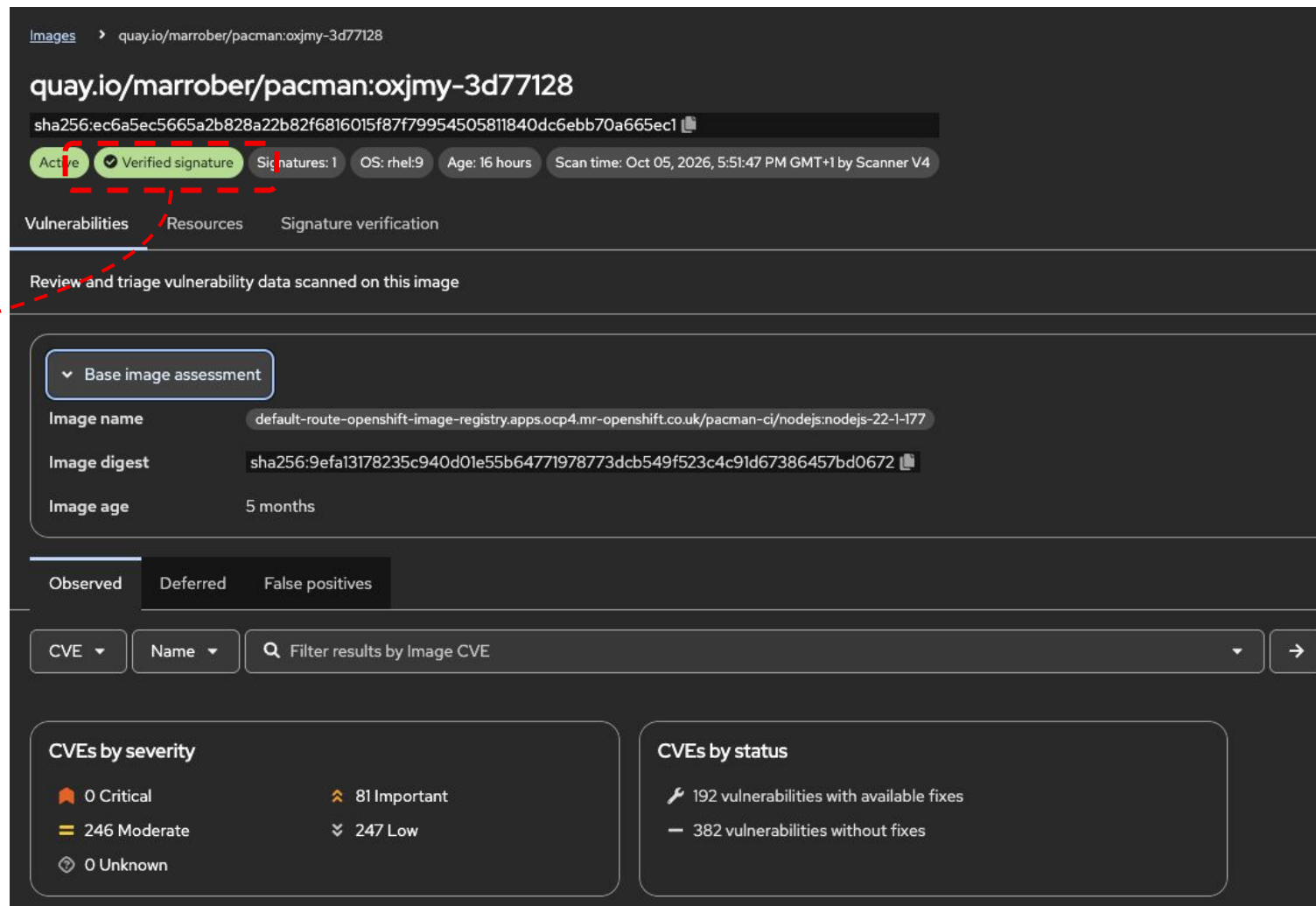
☐ Validate in offline mode
Force offline validation of the signature proof of inclusion into the transparency log. Do not fall back to request confirmation from the transparency log over network.

Rekor public key

-----BEGIN PUBLIC KEY-----

ACS view of the image

Vulnerabilities and evidence of signing



Images > quay.io/marrober/pacman:oxjmy-3d77128

quay.io/marrober/pacman:oxjmy-3d77128

sha256:ec6a5ec5665a2b828a22b82f6816015f87f79954505811840dc6ebb70a665ec1

Active Verified signature Signatures: 1 OS: rhel:9 Age: 16 hours Scan time: Oct 05, 2026, 5:51:47 PM GMT+1 by Scanner V4

Vulnerabilities Resources Signature verification

Review and triage vulnerability data scanned on this image

Base image assessment

Image name default-route-openshift-image-registry.apps.ocp4.mr-openshift.co.uk/pacman-ci/nodejs:nodejs-22-1-177

Image digest sha256:9efa13178235c940d01e55b64771978773dcb549f523c4c91d67386457bd0672

Image age 5 months

Observed Deferred False positives

CVE Name Filter results by Image CVE

CVEs by severity

- 0 Critical
- 246 Moderate
- 0 Unknown
- 81 Important
- 247 Low

CVEs by status

- 192 vulnerabilities with available fixes
- 382 vulnerabilities without fixes

Verified image references

Cosign-rekor

- quay.io/marrober/pacman:oxjmy-3d77128

Build with security focused CI/CD workflows

Efficiently track and trace security issues to remain industry compliant

Project: secured-app-dev ▾

PipelineRuns ▸ PipelineRun details

PLR Ob0de904-696c-4de5-88a8-e964ff0568c5 Succeeded Actions ▾

[Details](#) [YAML](#) [Parameters](#) [Logs](#) [Events](#) [Output](#) [TaskRuns](#)

PipelineRun details

Name
Ob0de904-696c-4de5-88a8-e964ff0568c5

Namespace
NS secured-app-dev

Labels Edit

backstage.io/kubernetes-id=secured-app rht-gitops.com/janus-argocd=secured-app-dev-build tekton.dev/pipeline=secured-app-build
triggers.tekton.dev/eventlistener=secured-app-el triggers.tekton.dev/trigger=commit-trig
triggers.tekton.dev/triggers-eventid=c23e021e-3415-4077-81ff-36f0a70582e1 type=pipeline

Annotations
[2 annotations](#)

Status
 Succeeded

Vulnerabilities
 Critical 3 High 40 Medium 139 Low 90

Pipeline
PL secured-app-build

Download SBOM
cosign download sbom quay-xjcj4.apps.cluster-xjcj4.xjcj4.sandbox1665.opentlc.com/quayadmin/secured-app
[Install Cosign](#)

SBOM
[View SBOM](#)

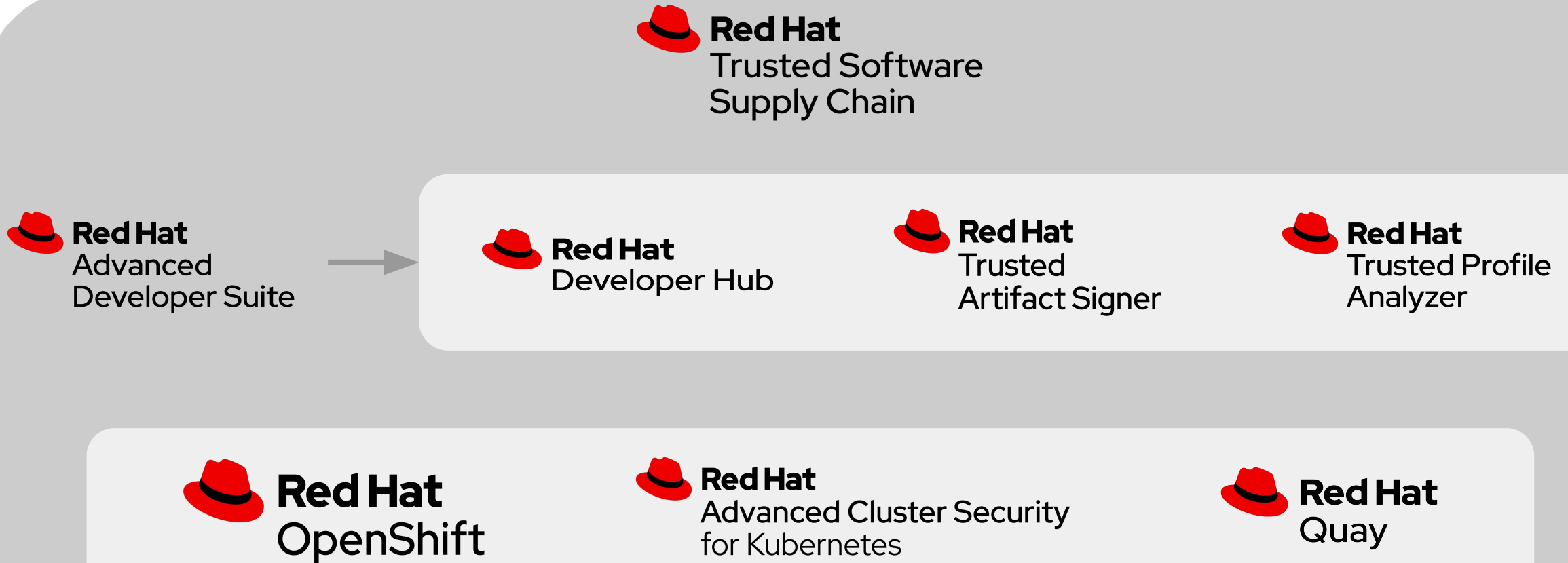
Start time
 Jun 4, 2025, 1:54 PM

Securing the software supply chain

Runtime analysis for out of compliance apps and communication

- Monitor application execution – use of dnf, curl or wget, for example
- Create and validate network policy for microservice isolation
- Notify security teams of violations as they happen
- Validate compliance of clusters, namespaces and individual deployments against standards

Red Hat Advanced Developer Suite





Connect

Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



x.com/RedHat

