Red Hat Trusted Software Supply Chain

Matching supply chain resiliency with innovation speed

Ilkka Tengvall
Solutions Architect
ikke@redhat.com
Matrix: #redhat-fin-users:hacklab.fi

Red Hat


When we buy a new car we expect every part supplied to be genuine
...

https://www.youtube.com/watch?v=3Hip60FA6j8


... can we say the same about every
single part of our software ?

Software supply chain attacks: a matter of when, not if

Ransom paid but a mere fraction to the overall downtime and recovery costs of a data breach

742%

average annual increase in software supply chain attacks over the past 3 years[1]

20%

data breaches are due to a compromised software supply chain[2]

78%

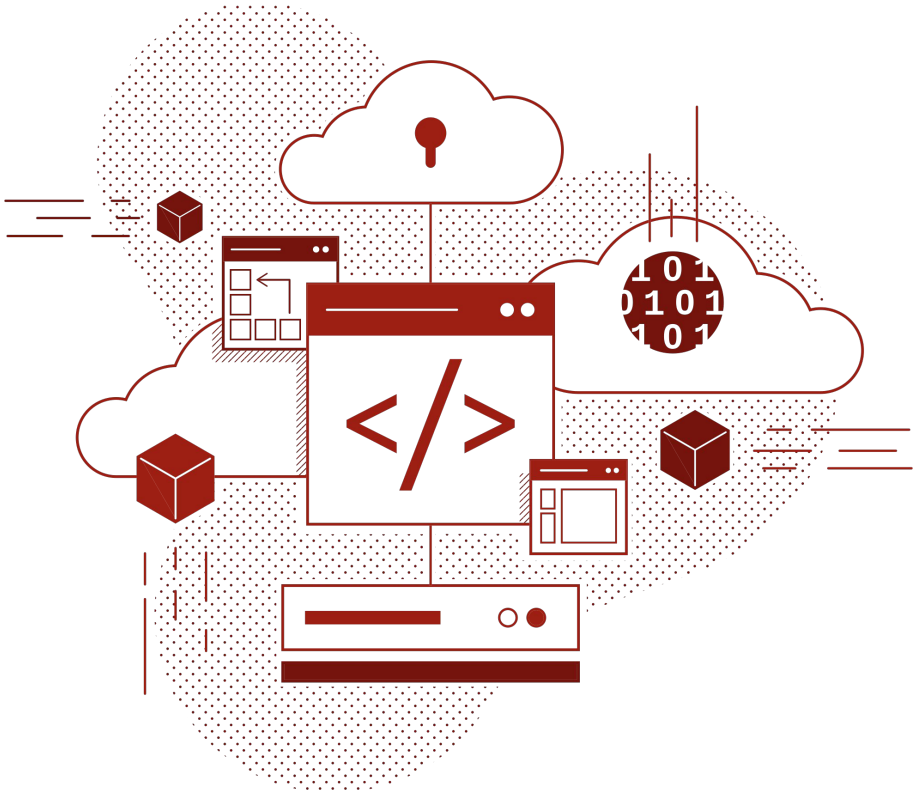
have initiatives to increase collaboration between DevOps and Security teams[3]

92%

say enterprise open source solutions are important as their business accelerates to the open hybrid cloud[4]

[1] State of the Software Supply Chain | [2] Cost of a Data Breach 2022 - IBM Report | [3]State of Kubernetes Security Report 2022 - Red Hat Report | [4]State of Enterprise Open Source 2022 - Red Hat Report

Red Hat



Growing attack surfaces with new, emerging threats daily

Software supply chain security a critical component to securing data, IP and source code

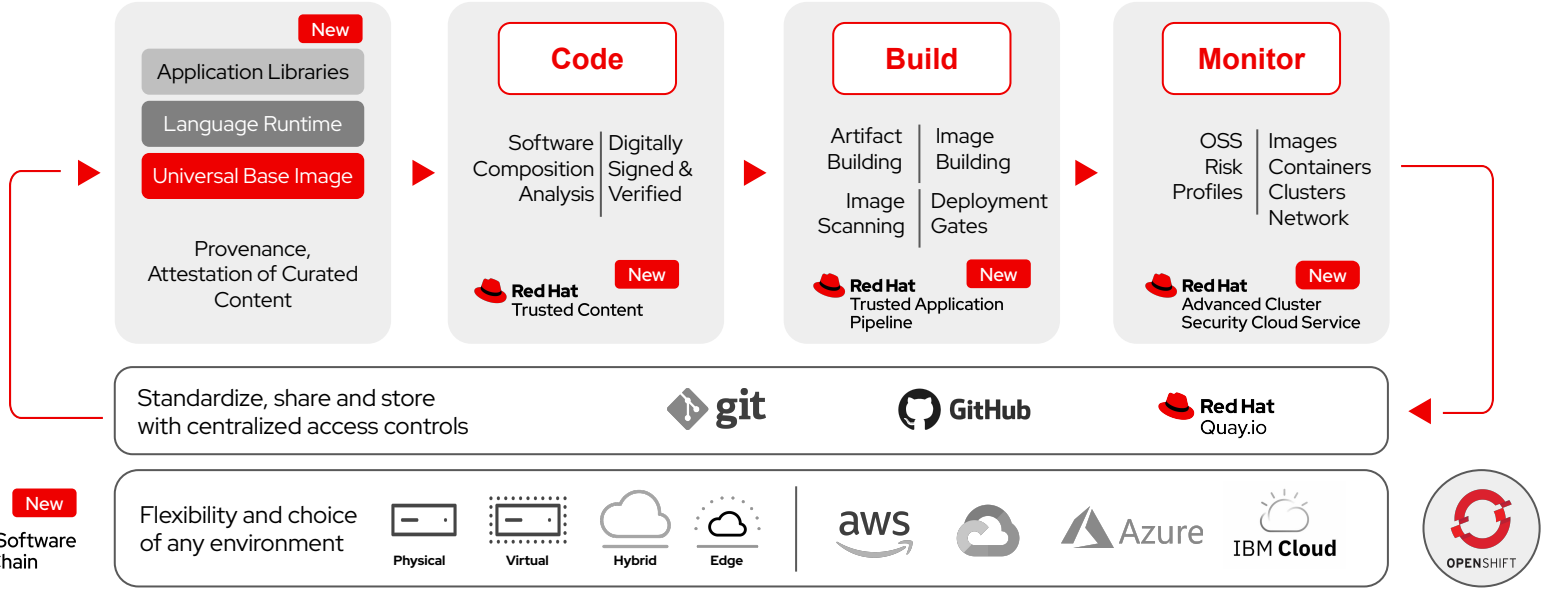
- Stolen Certificates
- Typosquatting Attack
- Dependency Confusion
- Compromised Build Environment
- Malware preinstalled on devices
- Malicious code in firmware

Red Hat: Providing trusted enterprise open source software for 30+ years

- All code is cloned in internal repositories.
- Strong distribution mechanisms with signed packages.
- Strong safeguards against tampering.
- Minimal modifications over product lifetimes protects from unwanted and potentially risky upstream code changes.

Code, build, and monitor to a Trusted Software Supply Chain

*Delivered as a **cloud service** with integrated security guardrails at every phase of the software development lifecycle*

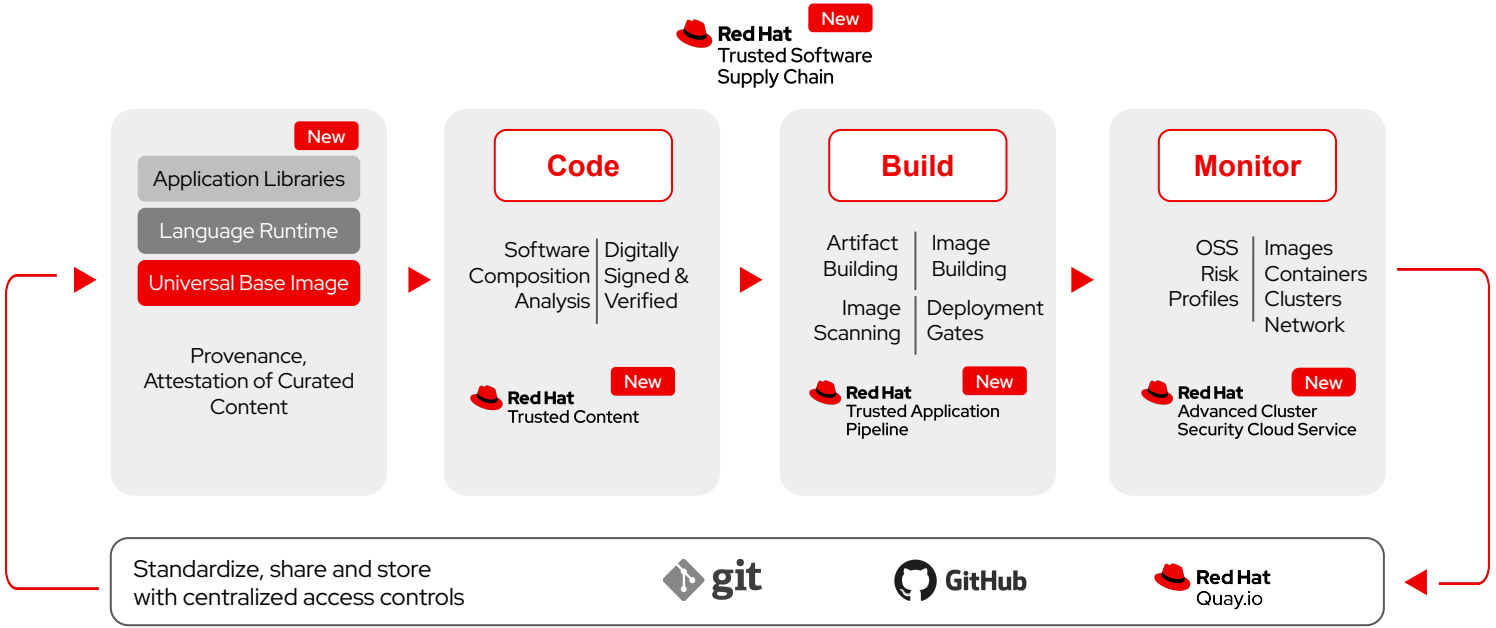


Red Hat Trusted Software Supply Chain New

Red Hat

Code, build, and monitor to a Trusted Software Supply Chain

*Delivered as a **cloud service** with integrated security guardrails at every phase of the software development lifecycle*



Ulkomaat | Australia

Nainen tarjosi vierailleen myrkky-sienipiirasta, nyt häntä epäillään kolmen ihmisen kuolemasta

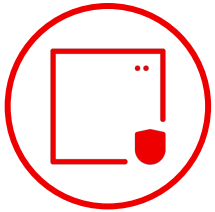
Poliisi tutkii Australiassa kolmen ihmisen kuolemaan johtanutta epäiltyä sienimyrkytystä.

https://www.hs.fi/ulkomaat/art-2000009775653.html

Secure the use of source code and transitive dependencies

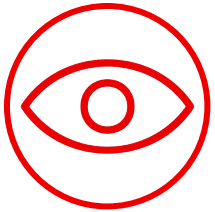
Software supply chain security considerations for the software development lifecycle

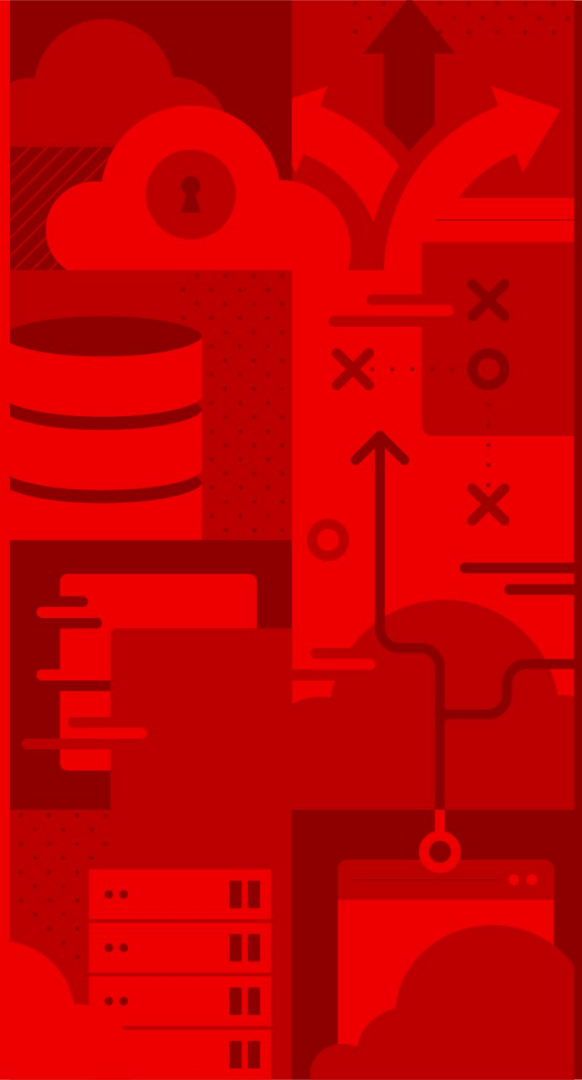
Prevent & identify malicious **code**



Safeguard **build** systems early

Continuously **monitor** security at runtime





Secure your open source code and dependencies early!

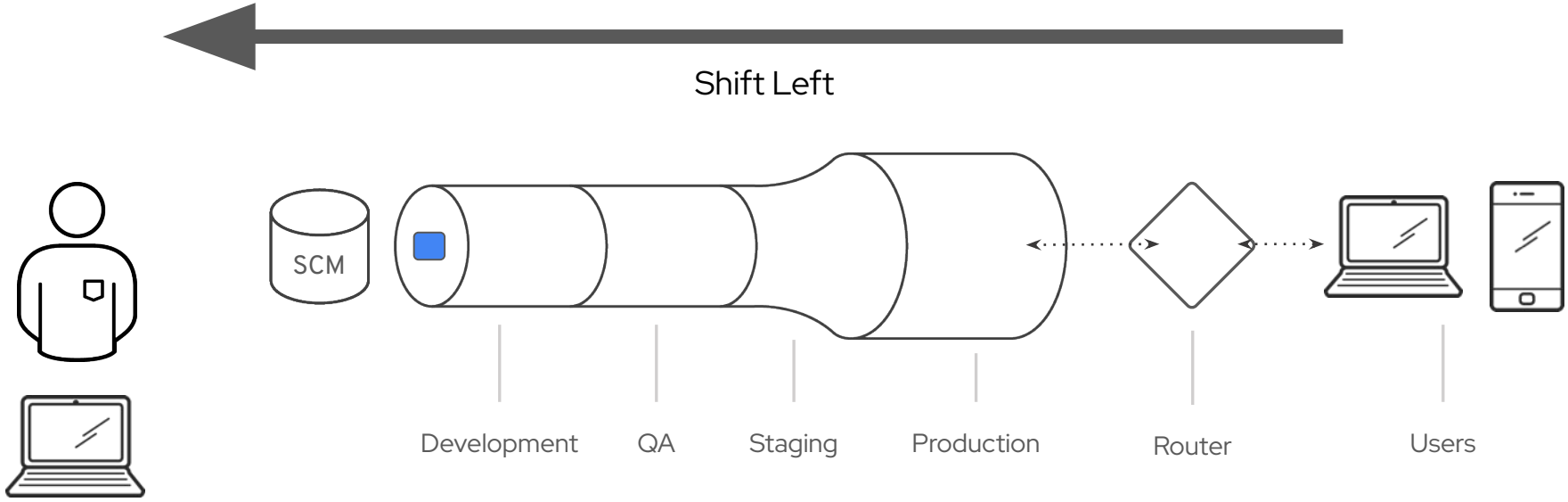
Red Hat

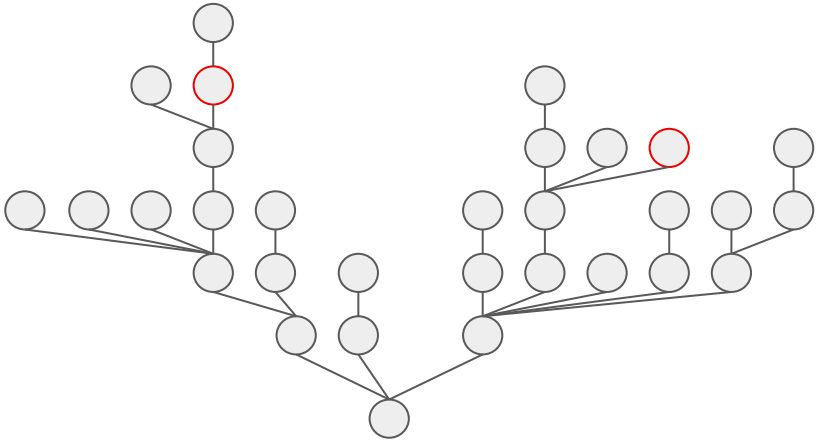
Terminology

Term	Definition
SLSA	Supply Chain Levels for Software Artifacts SLSA is a set of standards and technical controls you can adopt to improve artifact integrity, and build
SAST	Static Application Security Testing Executed at build time as part of the CI
DAST	Dynamic Application Security Testing Often executed on staging clusters
OWASP	Open Web Application Security Project OWASP Top 10
CVE	Common Vulnerability and Exposures
Provenance	Recording of origin, history and who made the changes
Attestation	Authenticated statement (metadata) about a software artifact or collection of software artifacts
Sigstore	Sigstore empowers software developers to securely sign software artifacts such as release files, container images, binaries, bill of material manifests and more. Signing materials are then stored in a tamper-resistant public log.
SBOM	Software Bill of Materials
SPDX, CycloneDX	Competing solutions for the structure of a SBOM. SPDX lead by Linux Fouancation. CycloneDX lead by OWASP.
SCA	Software Composition Analysis

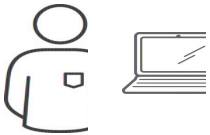
SLSA Levels (https://slsa.dev/spec/v1.0/levels)

Level 1	Level 2	Level 3
Preventing Mistakes Automated Build Process Generated provenance about source, build process, artifact and dependencies	Preventing tampering after the build Generated, signed and verifiable provenance	Preventing tampering during the build Prevent runs from influencing one another, prevent secret material used to sign provenance from being accessible by the end-user's defined steps


Shift Left
SCM
Development
QA
Staging
Production
Router
Users
Developer



I need a HTTP library, JSON parser, database access, Java runtime, Linux OS



pom.xml
package.json
requirements.txt
Dockerfile

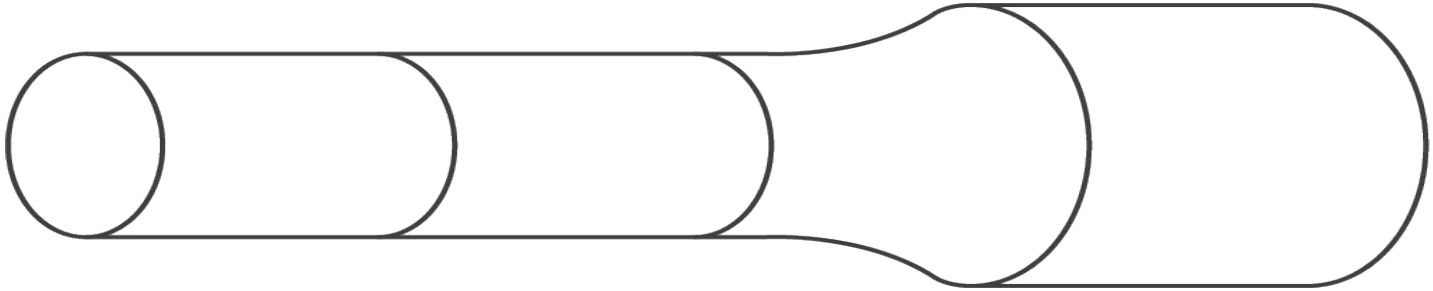
mvn dependency:tree

Spring Boot 2.7.7 Hello World

```
[INFO] com.example:demo:jar:0.0.1-SNAPSHOT
[INFO] +- org.springframework.boot:spring-boot-starter-web:jar:2.7.7:compile
[INFO] |  +- org.springframework.boot:spring-boot-starter:jar:2.7.7:compile
[INFO] |  |  +- org.springframework.boot:spring-boot-starter-logging:jar:2.7.7:compile
[INFO] |  |  |  +- ch.qos.logback:logback-classic:jar:1.2.11:compile
[INFO] |  |  |  |  \- ch.qos.logback:logback-core:jar:1.2.11:compile
[INFO] |  |  |  +- org.apache.logging.log4j:log4j-to-slf4j:jar:2.17.2:compile
[INFO] |  |  |  |  \- org.apache.logging.log4j:log4j-api:jar:2.17.2:compile
[INFO] |  |  |  \- org.slf4j:jul-to-slf4j:jar:1.7.36:compile
[INFO] |  |  +- jakarta.annotation:jakarta.annotation-api:jar:1.3.5:compile
[INFO] |  |  \- org.yaml:snakeyaml:jar:1.30:compile
[INFO] |  +- org.springframework.boot:spring-boot-starter-json:jar:2.7.7:compile
[INFO] |  |  +- com.fasterxml.jackson.core:jackson-databind:jar:2.13.4.2:compile
[INFO] |  |  |  +- com.fasterxml.jackson.core:jackson-annotations:jar:2.13.4:compile
[INFO] |  |  |  \- com.fasterxml.jackson.core:jackson-core:jar:2.13.4:compile
[INFO] |  |  +- com.fasterxml.jackson.datatype:jackson-datatype-jdk8:jar:2.13.4:compile
[INFO] |  |  +- com.fasterxml.jackson.datatype:jackson-datatype-jsr310:jar:2.13.4:compile
[INFO] |  |  \- com.fasterxml.jackson.module:jackson-module-parameter-names:jar:2.13.4:compile
[INFO] |  +- org.springframework.boot:spring-boot-starter-tomcat:jar:2.7.7:compile
[INFO] |  |  +- org.apache.tomcat.embed:tomcat-embed-core:jar:9.0.70:compile
[INFO] |  |  +- org.apache.tomcat.embed:tomcat-embed-el:jar:9.0.70:compile
[INFO] |  |  \- org.apache.tomcat.embed:tomcat-embed-websocket:jar:9.0.70:compile
[INFO] |  +- org.springframework:spring-web:jar:5.3.24:compile
[INFO] |  |  \- org.springframework:spring-beans:jar:5.3.24:compile
[INFO] |  \- org.springframework:spring-webmvc:jar:5.3.24:compile
[INFO] |     +- org.springframework:spring-aop:jar:5.3.24:compile
[INFO] |     +- org.springframework:spring-context:jar:5.3.24:compile
[INFO] |     \- org.springframework:spring-expression:jar:5.3.24:compile
[INFO] +- org.springframework.boot:spring-boot-devtools:jar:2.7.7:compile
[INFO] |  +- org.springframework.boot:spring-boot:jar:2.7.7:compile
[INFO] |  \- org.springframework.boot:spring-boot-autoconfigure:jar:2.7.7:compile
[INFO] \- org.springframework.boot:spring-boot-starter-test:jar:2.7.7:test
[INFO]    +- org.springframework.boot:spring-boot-test:jar:2.7.7:test
[INFO]    +- org.springframework.boot:spring-boot-test-autoconfigure:jar:2.7.7:test
[INFO]    +- com.jayway.jsonpath:json-path:jar:2.7.0:test
[INFO]    |  +- net.minidev:json-smart:jar:2.4.8:test
[INFO]    |  |  \- net.minidev:accessors-smart:jar:2.4.8:test
[INFO]    |  |     \- org.ow2.asm:asm:jar:9.1:test
[INFO]    |  \- org.slf4j:slf4j-api:jar:1.7.36:compile
[INFO]    +- jakarta.xml.bind:jakarta.xml.bind-api:jar:2.3.3:test
[INFO]    |  \- jakarta.activation:jakarta.activation-api:jar:1.2.2:test
[INFO]    +- org.assertj:assertj-core:jar:3.22.0:test
[INFO]    +- org.hamcrest:hamcrest:jar:2.2:test
[INFO]    +- org.junit.jupiter:junit-jupiter:jar:5.8.2:test
[INFO]    |  +- org.junit.jupiter:junit-jupiter-api:jar:5.8.2:test
[INFO]    |  |  +- org.opentest4j:opentest4j:jar:1.2.0:test
[INFO]    |  |  +- org.junit.platform:junit-platform-commons:jar:1.8.2:test
[INFO]    |  |  \- org.apiguardian:apiguardian-api:jar:1.1.2:test
[INFO]    |  +- org.junit.jupiter:junit-jupiter-params:jar:5.8.2:test
[INFO]    |  \- org.junit.jupiter:junit-jupiter-engine:jar:5.8.2:test
[INFO]    |     \- org.junit.platform:junit-platform-engine:jar:1.8.2:test
[INFO]    +- org.mockito:mockito-core:jar:4.5.1:test
[INFO]    |  +- net.bytebuddy:byte-buddy:jar:1.12.20:test
[INFO]    |  +- net.bytebuddy:byte-buddy-agent:jar:1.12.20:test
[INFO]    |  \- org.objenesis:objenesis:jar:3.2:test
[INFO]    +- org.mockito:mockito-junit-jupiter:jar:4.5.1:test
[INFO]    +- org.skyscreamer:jsonassert:jar:1.5.1:test
[INFO]    |  \- com.vaadin.external.google:android-json:jar:0.0.20131108.vaadin1:test
[INFO]    +- org.springframework:spring-core:jar:5.3.24:compile
[INFO]    |  \- org.springframework:spring-jcl:jar:5.3.24:compile
[INFO]    +- org.springframework:spring-test:jar:5.3.24:test
[INFO]    \- org.xmlunit:xmlunit-core:jar:2.9.0:test
[INFO] ------------------------------------------------------------------------
```

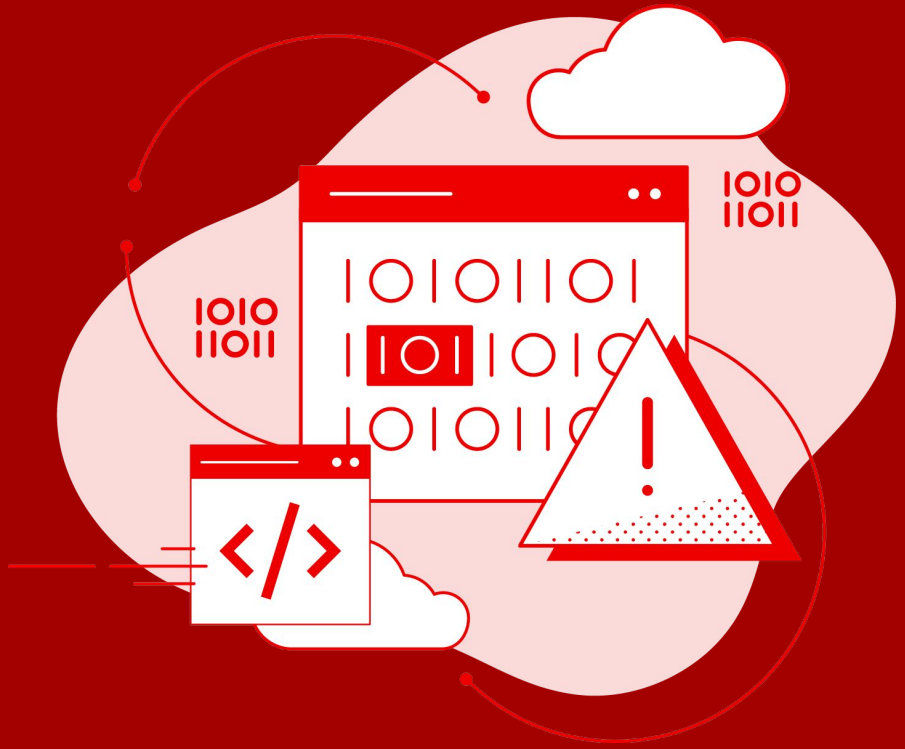
Red Hat

From git push to production



As a managed service, you can be up and running in minutes. Complicated product integrations are handled for you. Upgrades are continuous and seamless.

Deliver **securely-built images** to a registry, deploy applications to the cloud or to your on-prem OpenShift cluster with just a few steps.

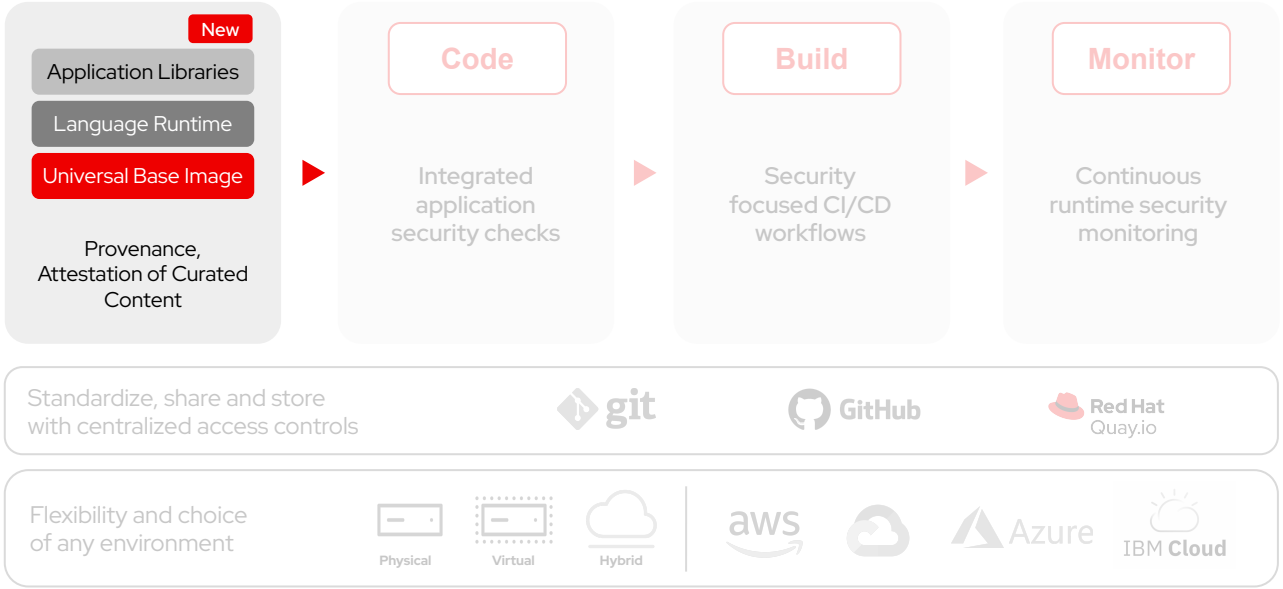


Prevent and identify malicious code

Red Hat

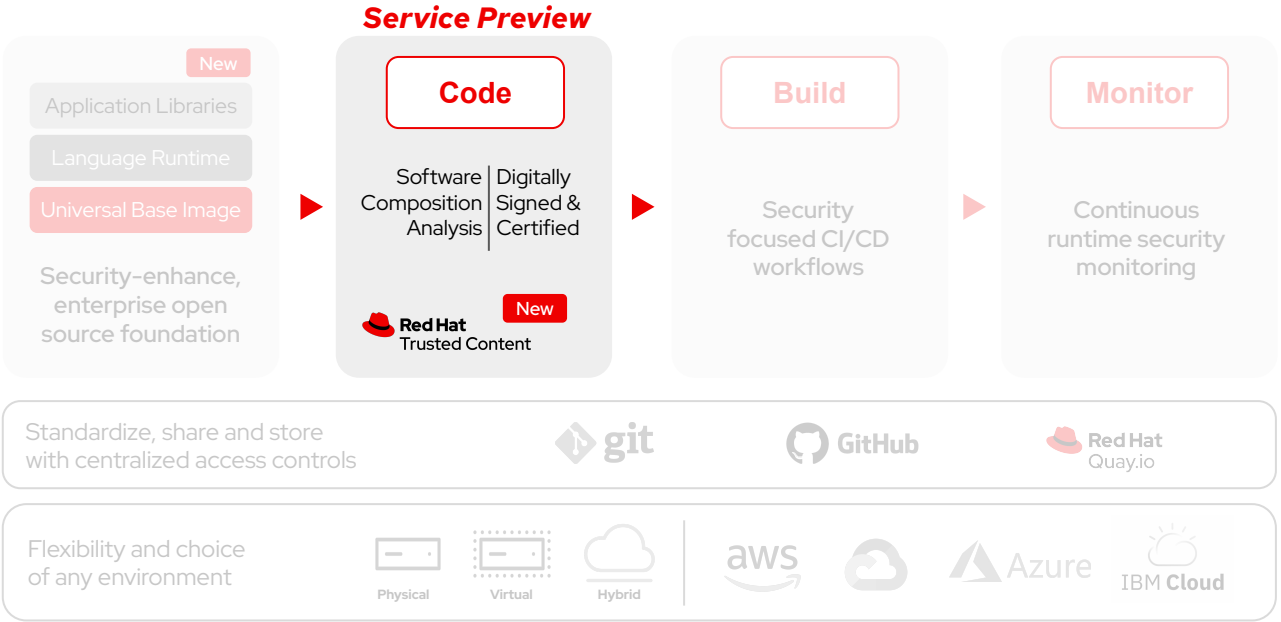
Security-enhanced, enterprise open source foundation

Build on business resilience that keeps pace with innovation velocity

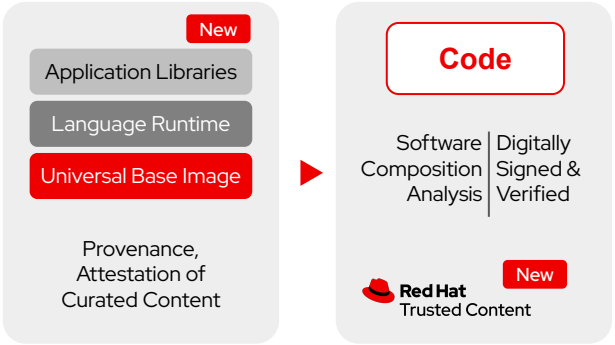


Integrated application security checks

Catch security issues early to keep and grow user trust



Code with integrated application security checks

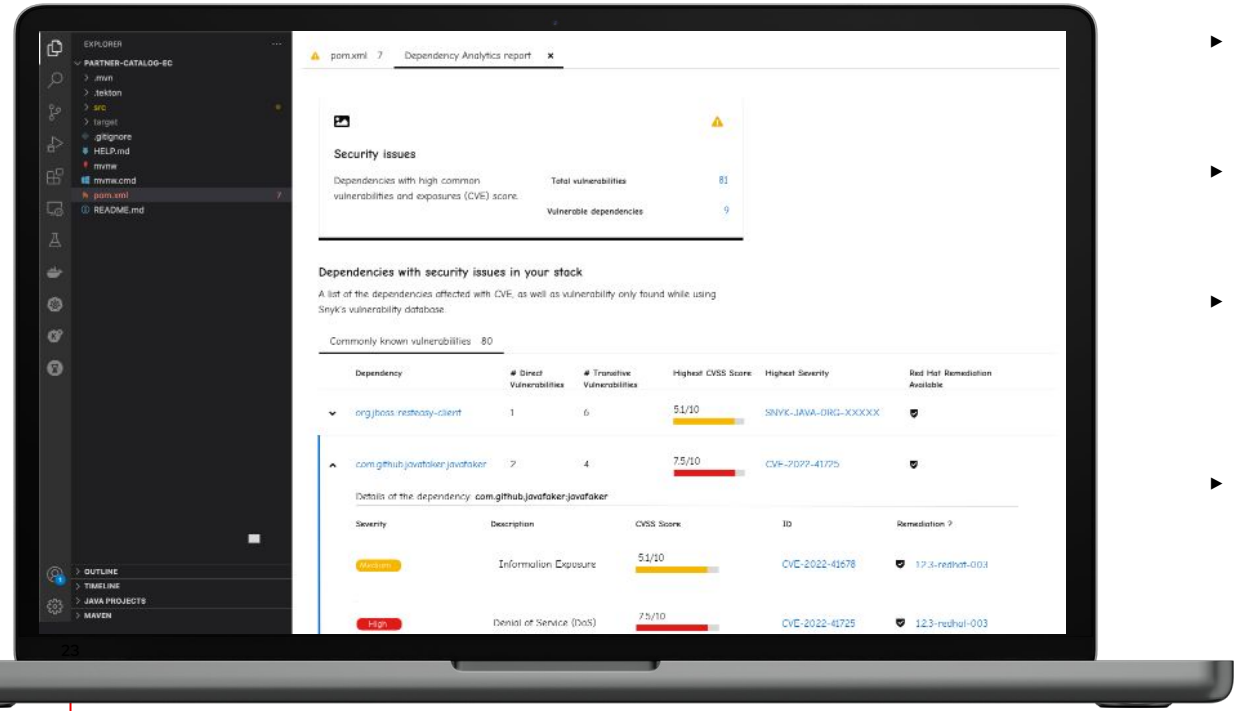


Catch security issues early to keep and grow user trust

- Trusted curated content
- Automated software composition analysis and dependency analytics
- Aggregated view with drill down on security health
- Cryptographic signing and verification

Red Hat

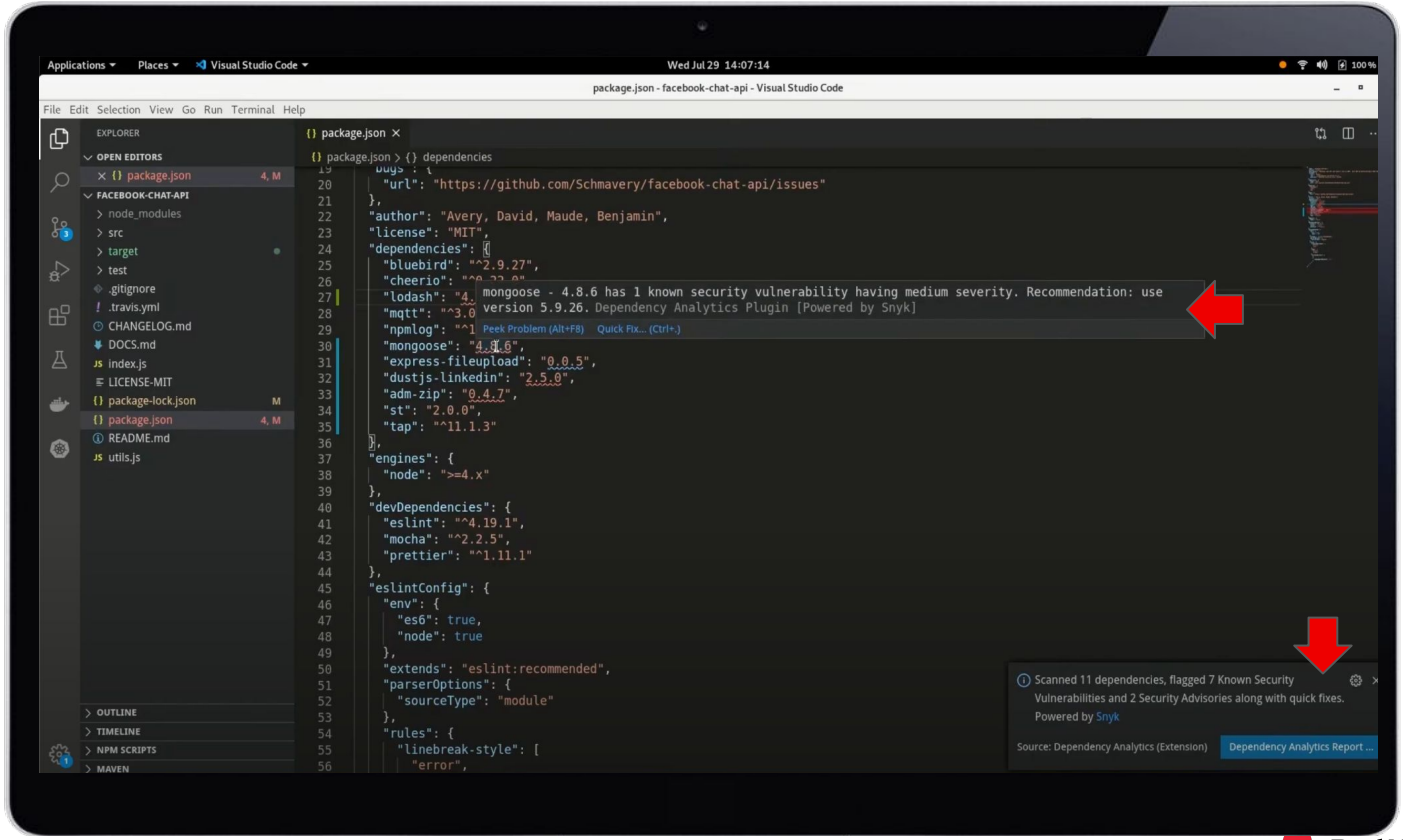
Leverage tried tested trusted curated content with security best practices at code time



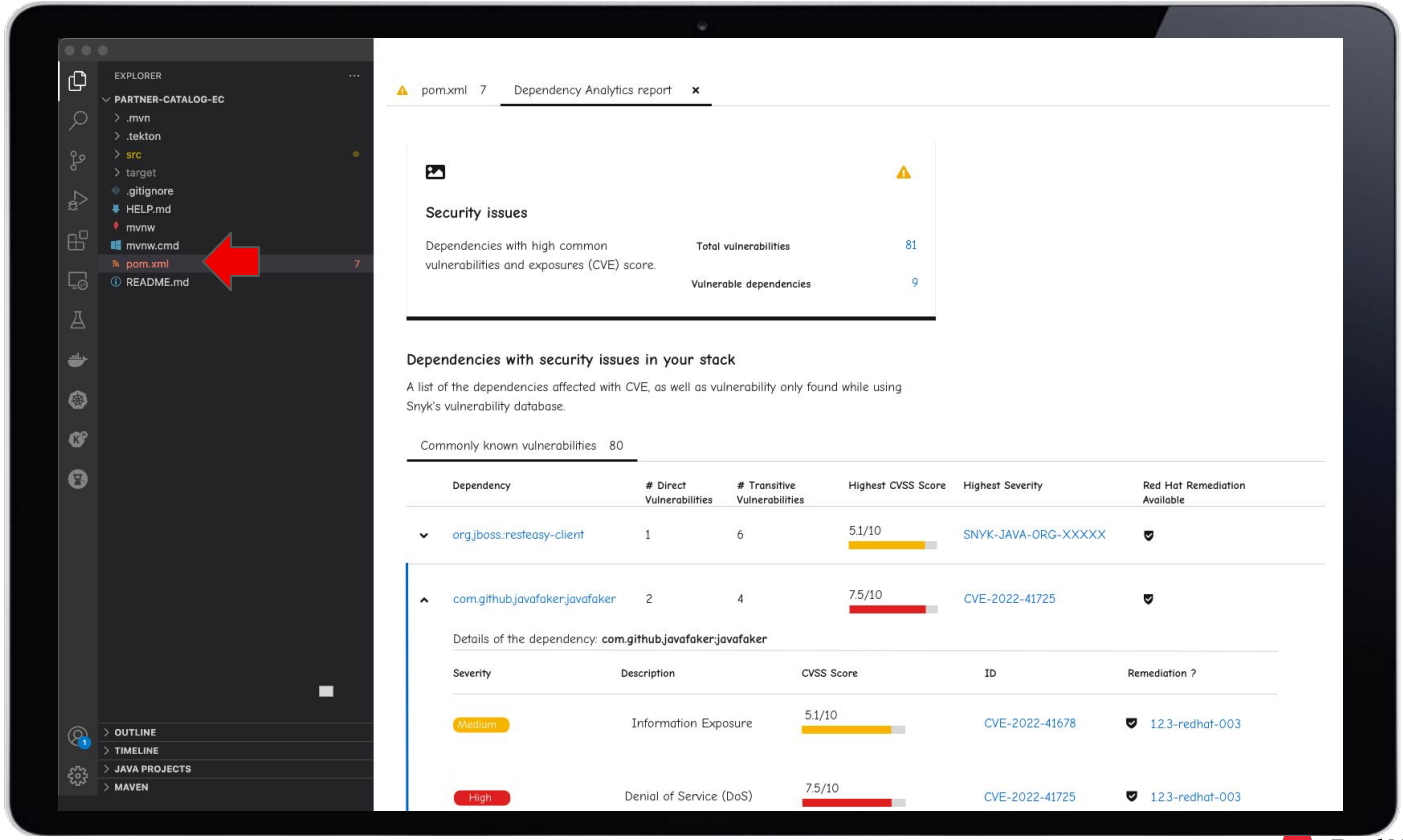
- 30 years providing trusted images and app libraries that are signed, verified
- Automate dependency analytics early with plug-ins to popular IDEs
- Single, shared repository for trusted content, detailed usage information, security issues and recommendations
- Tamper proof code to verify content from an open, immutable ledger

Red Hat

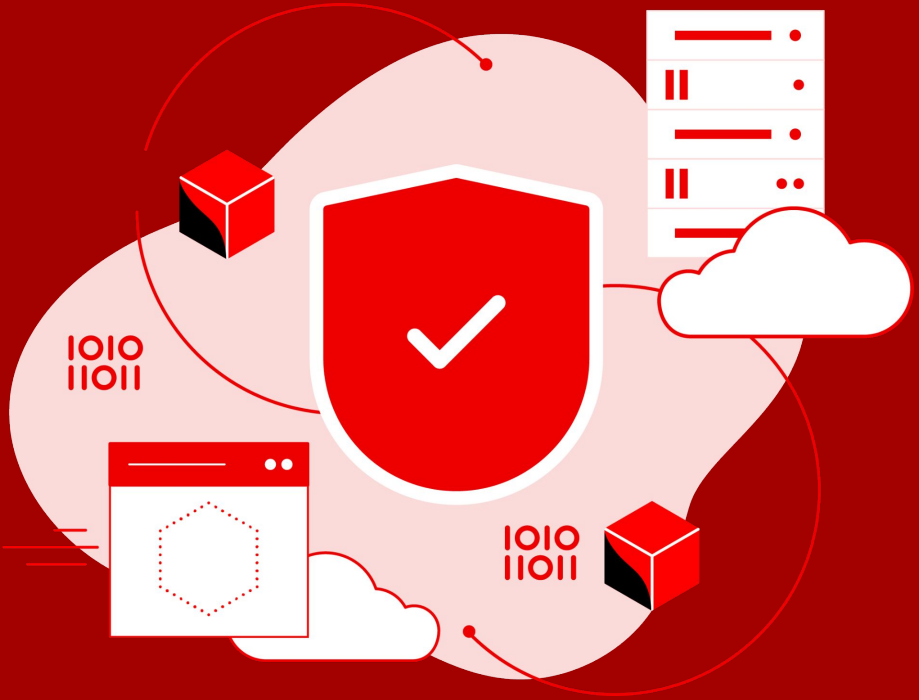
Remedy vulnerabilities with Trusted Content



Analyze and fix security issues from the IDE



TEMO

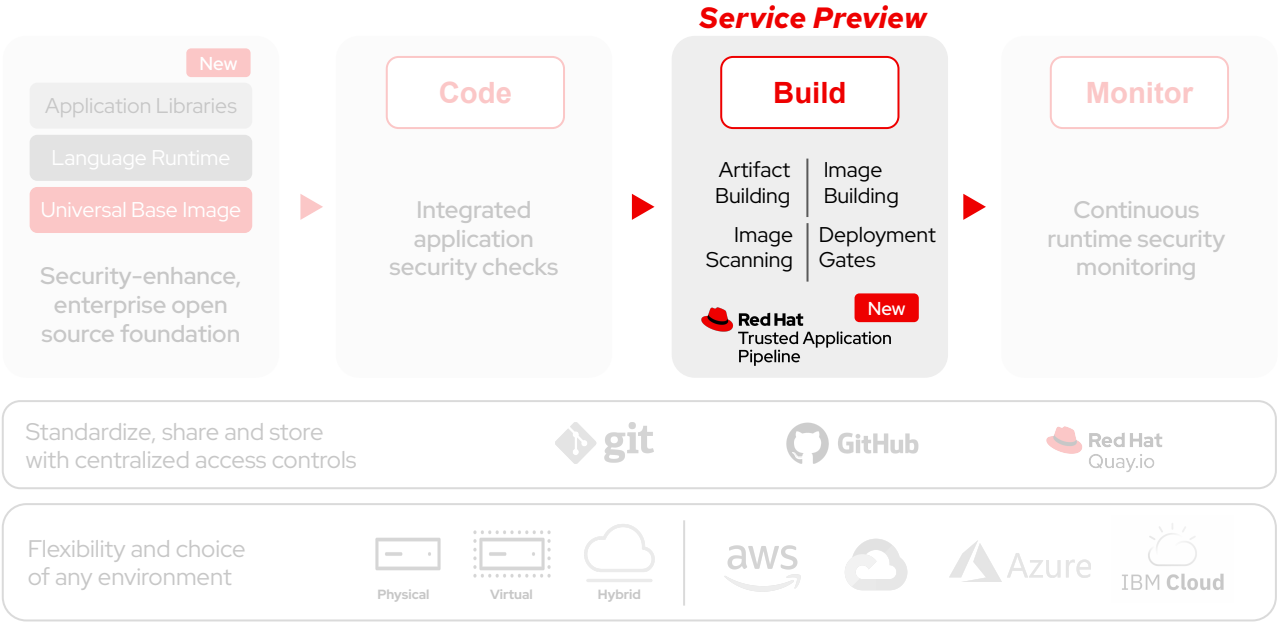


Safeguard build systems early

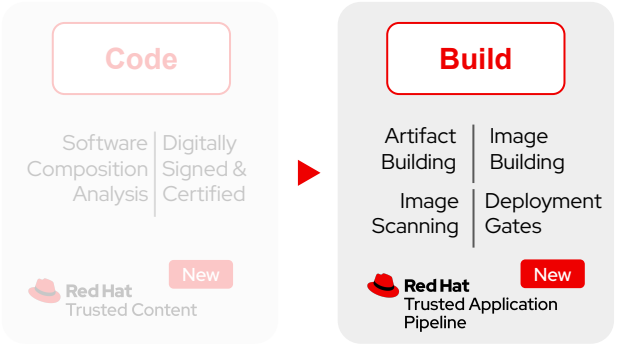
Red Hat

Security focused CI/CD workflows

Improved productivity,efficiency addresses customer needs faster



Build with security focused CI/CD workflows

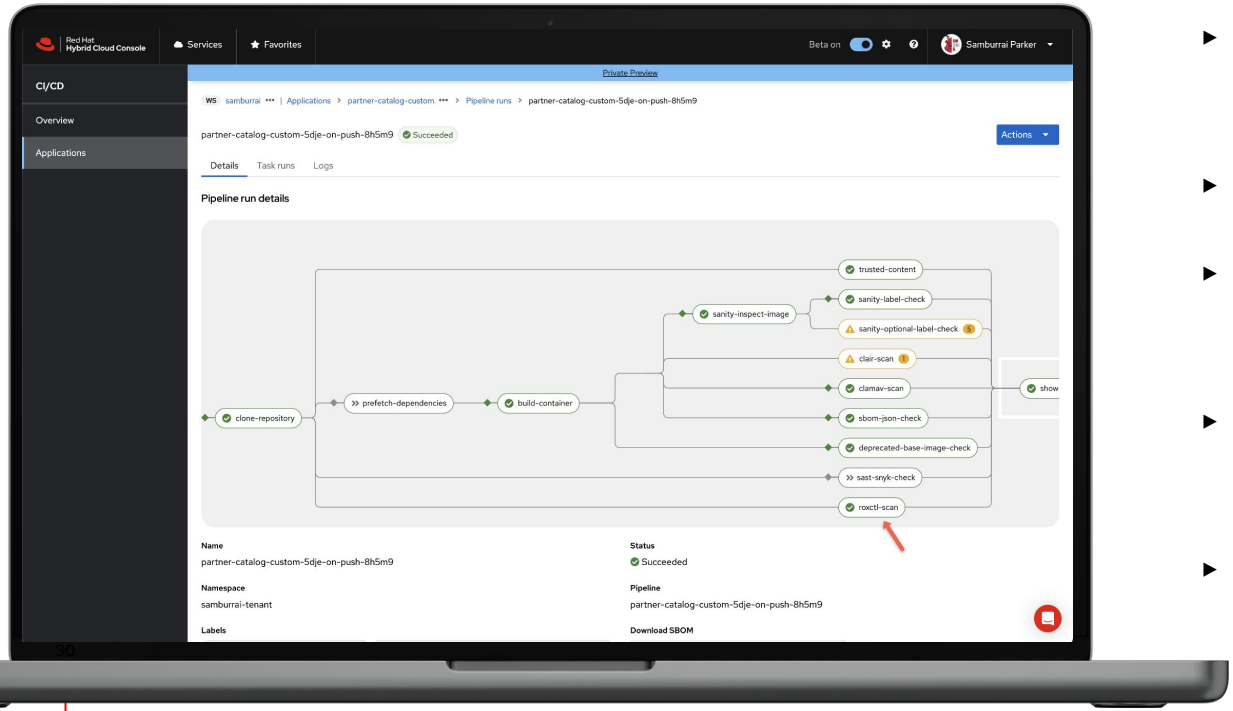


Meet industry compliance while increasing productivity, efficiency

- Integrated security guardrails across pipelines
- Auto-generated Software-Bill-of-Materials (SBOM)
- Attestations and provenance checks
- Deployment based on policies to a declared state
- Continuous image vulnerability scanning

Red Hat

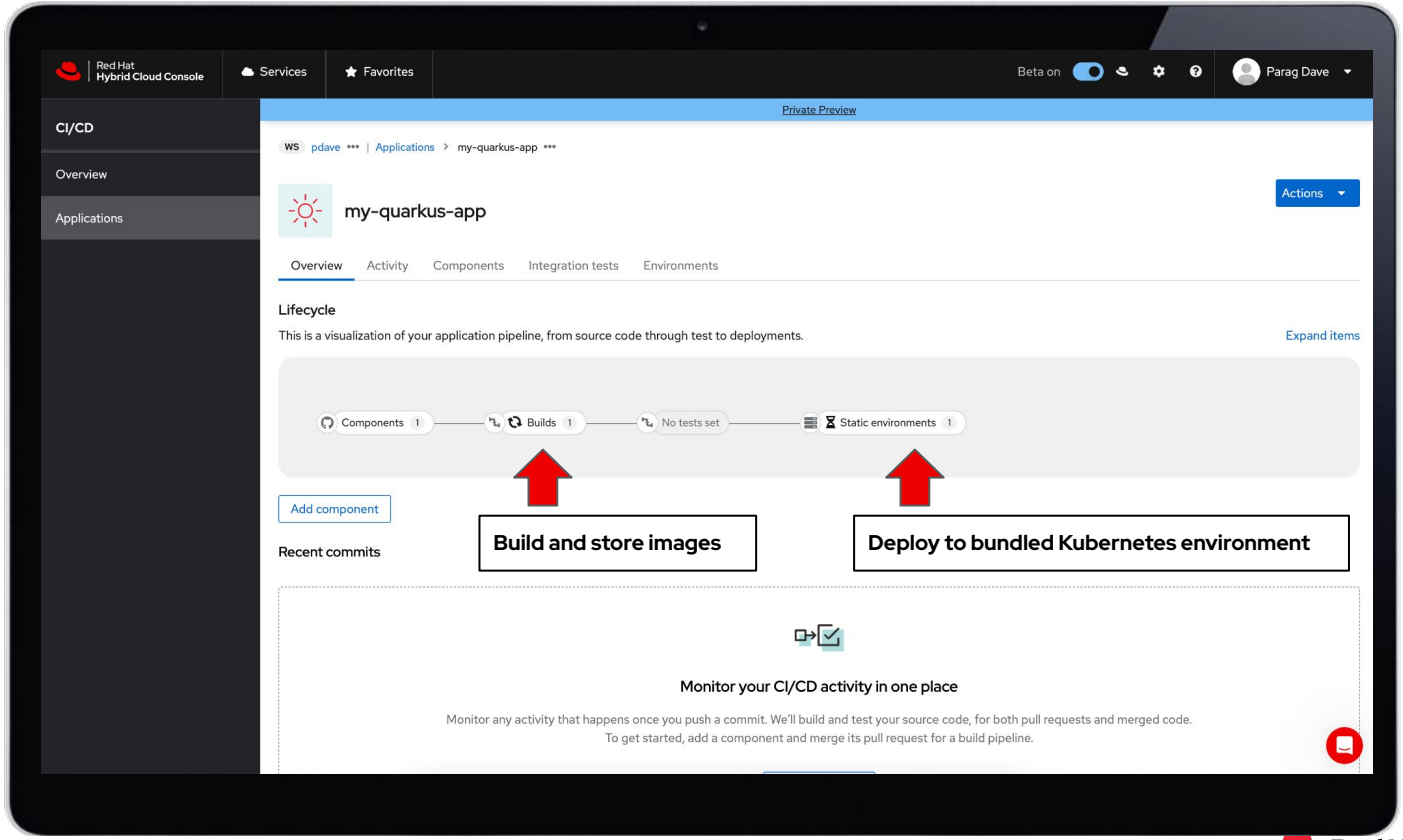
Strengthen the CI/CD pipeline with an automated chain of trust and approval gates



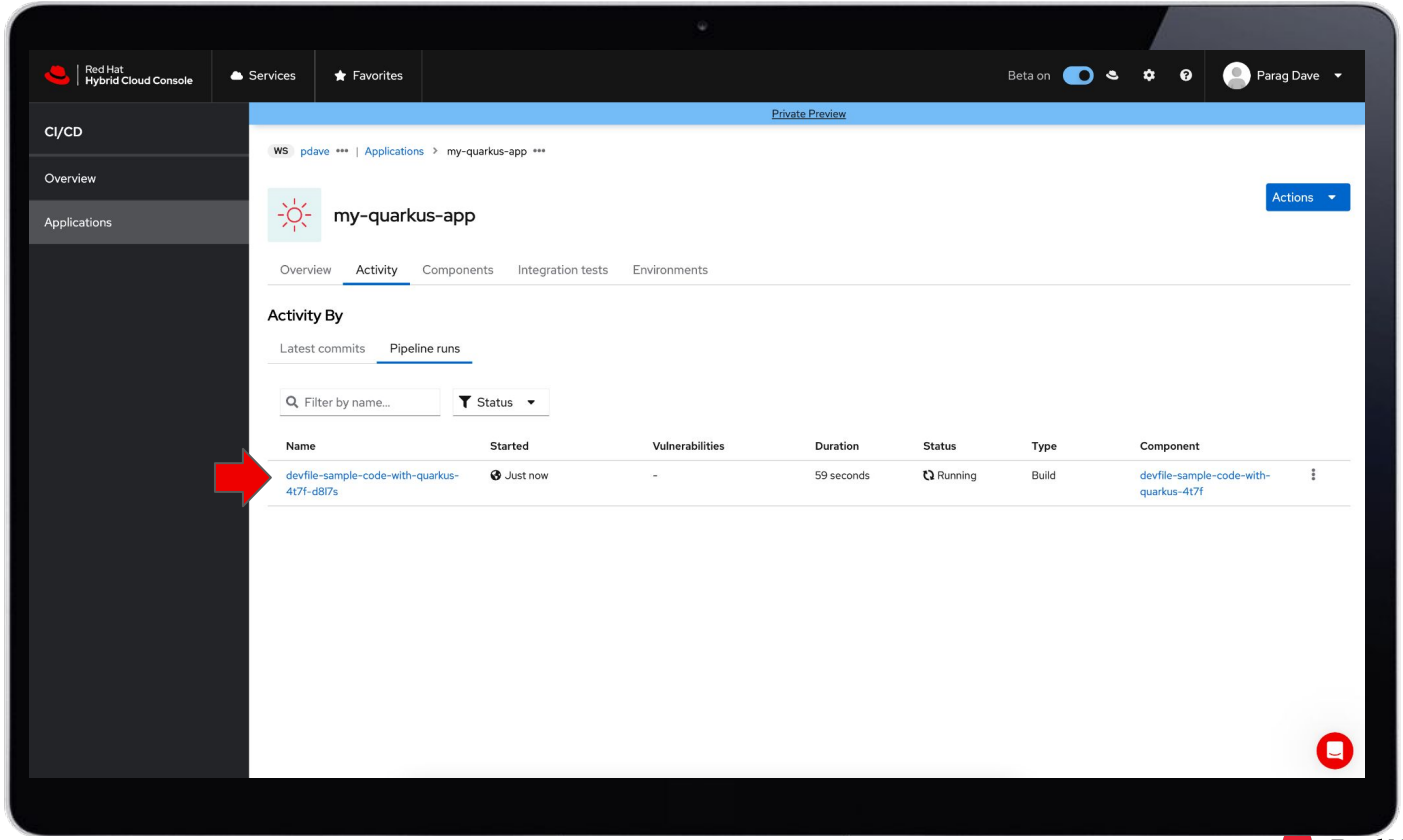
- Ready to use, customizable pipeline definition for hermetic builds
- Auto-generated SBOMs in minutes
- Scan images for vulnerabilities and exposures
- Continuously deploy via enterprise contract's 43 rules
- Pre-integrated security guardrails via Tekton Chains

Red Hat

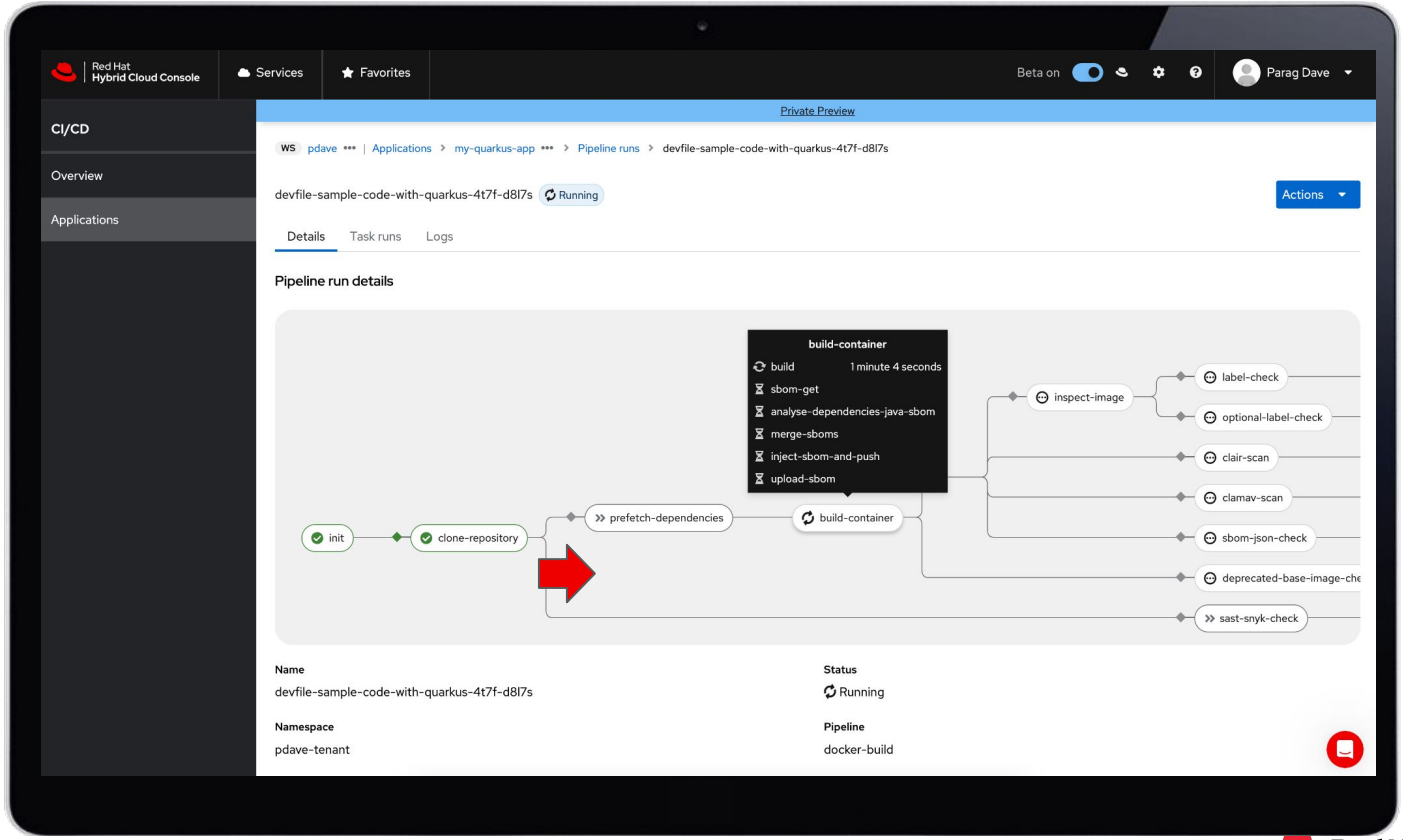
Automatically run default CI/CD pipeline



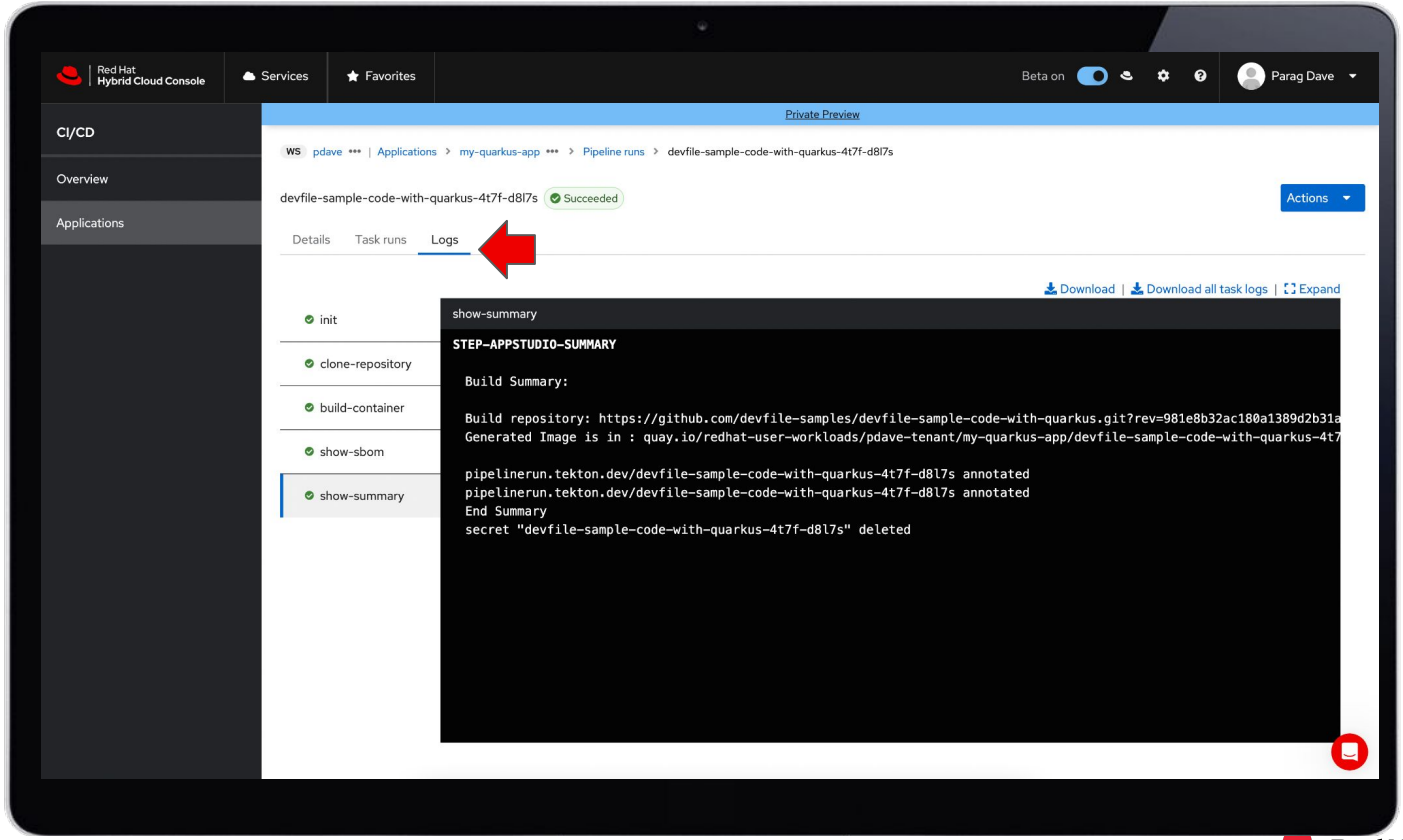
Drill down on pipeline details



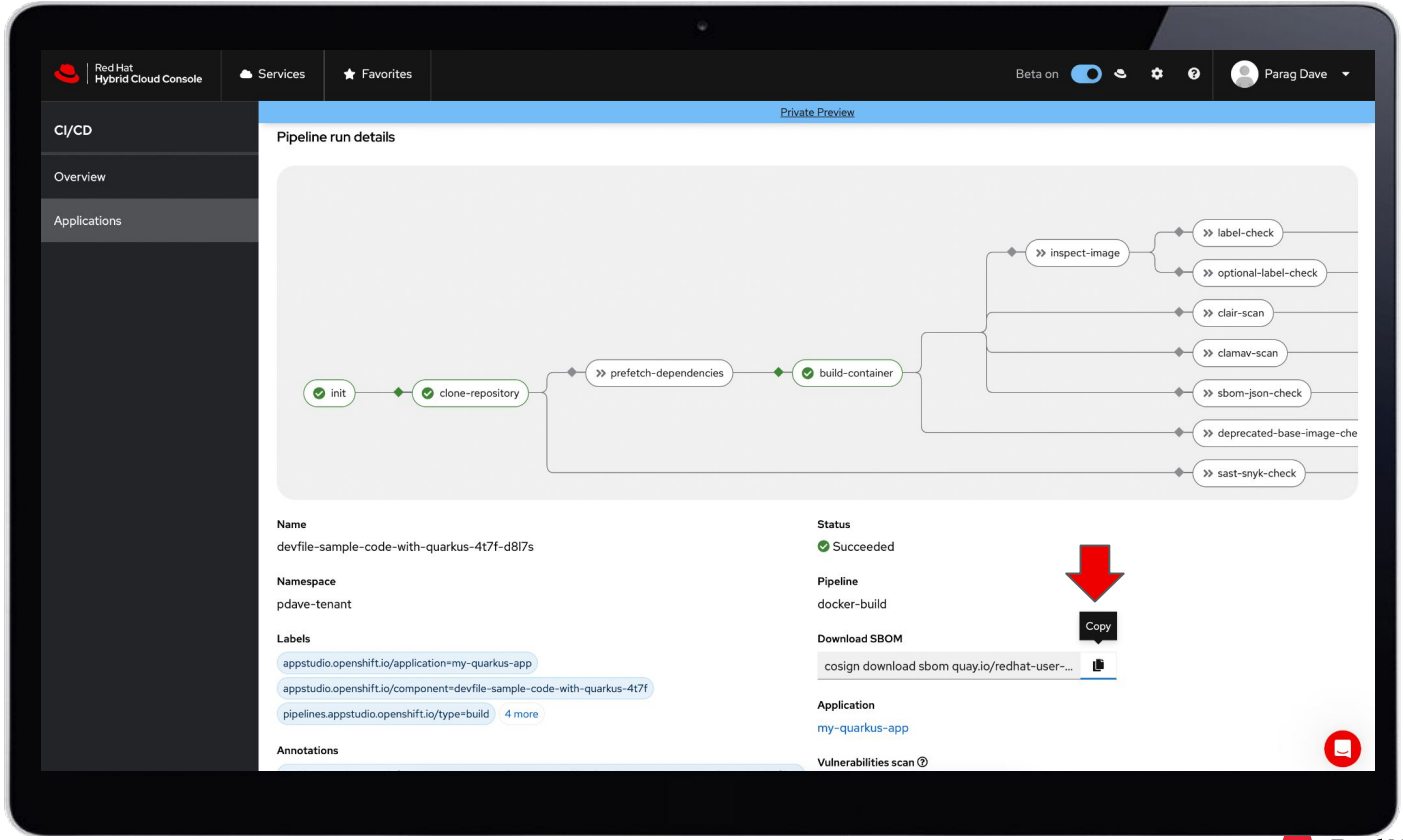
View live pipeline runs in real-time



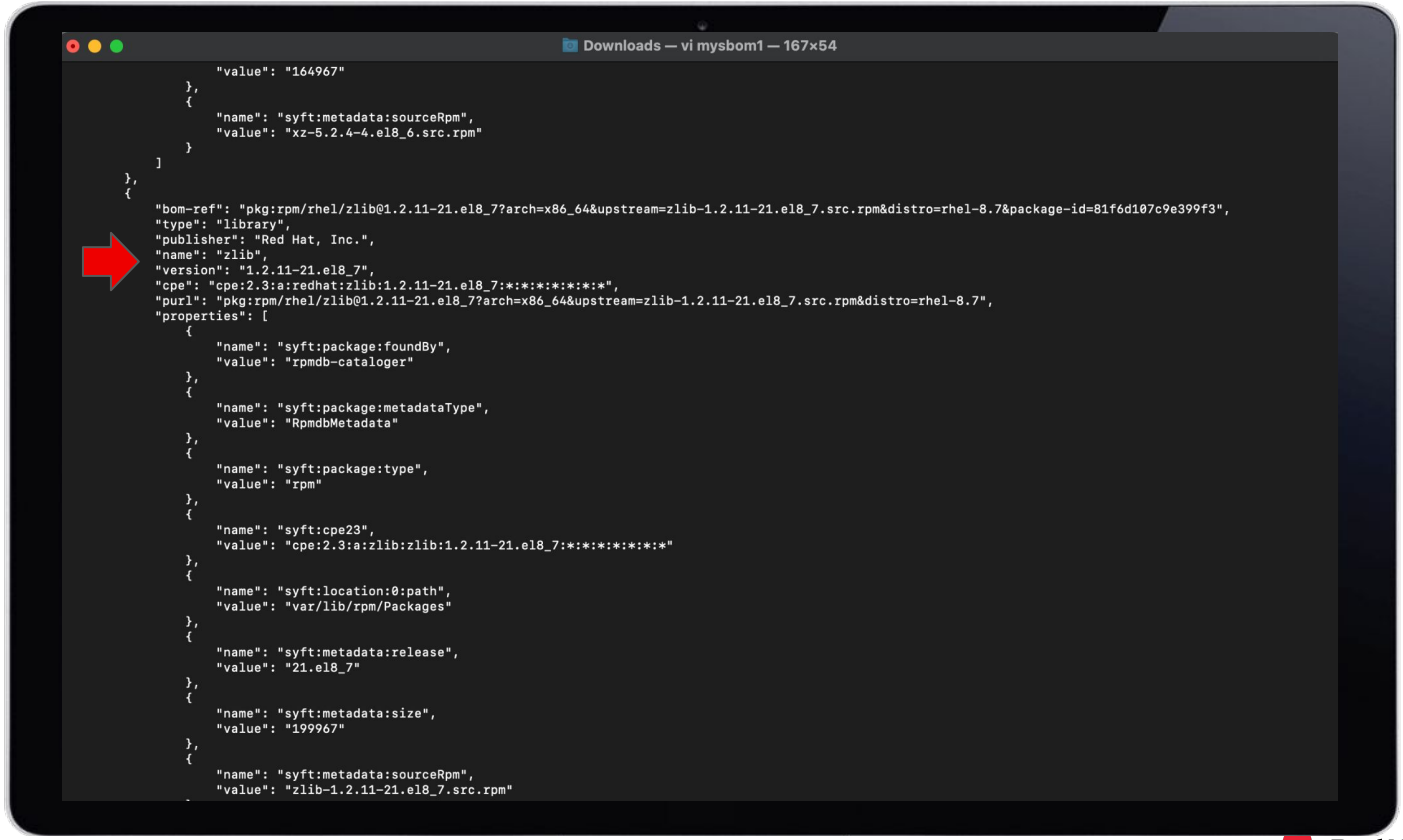
Access pipeline task log



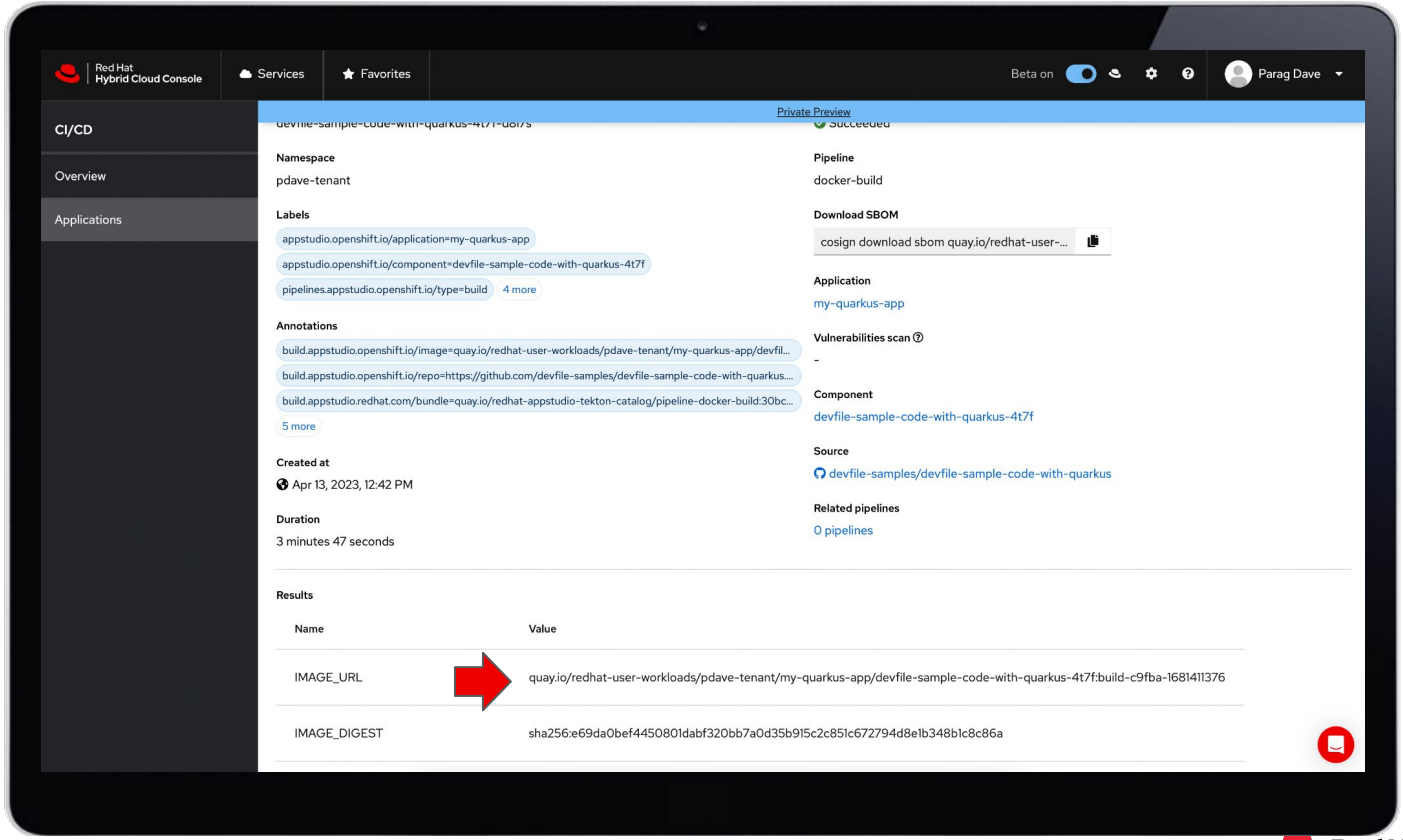
Download and share SBOM for the build



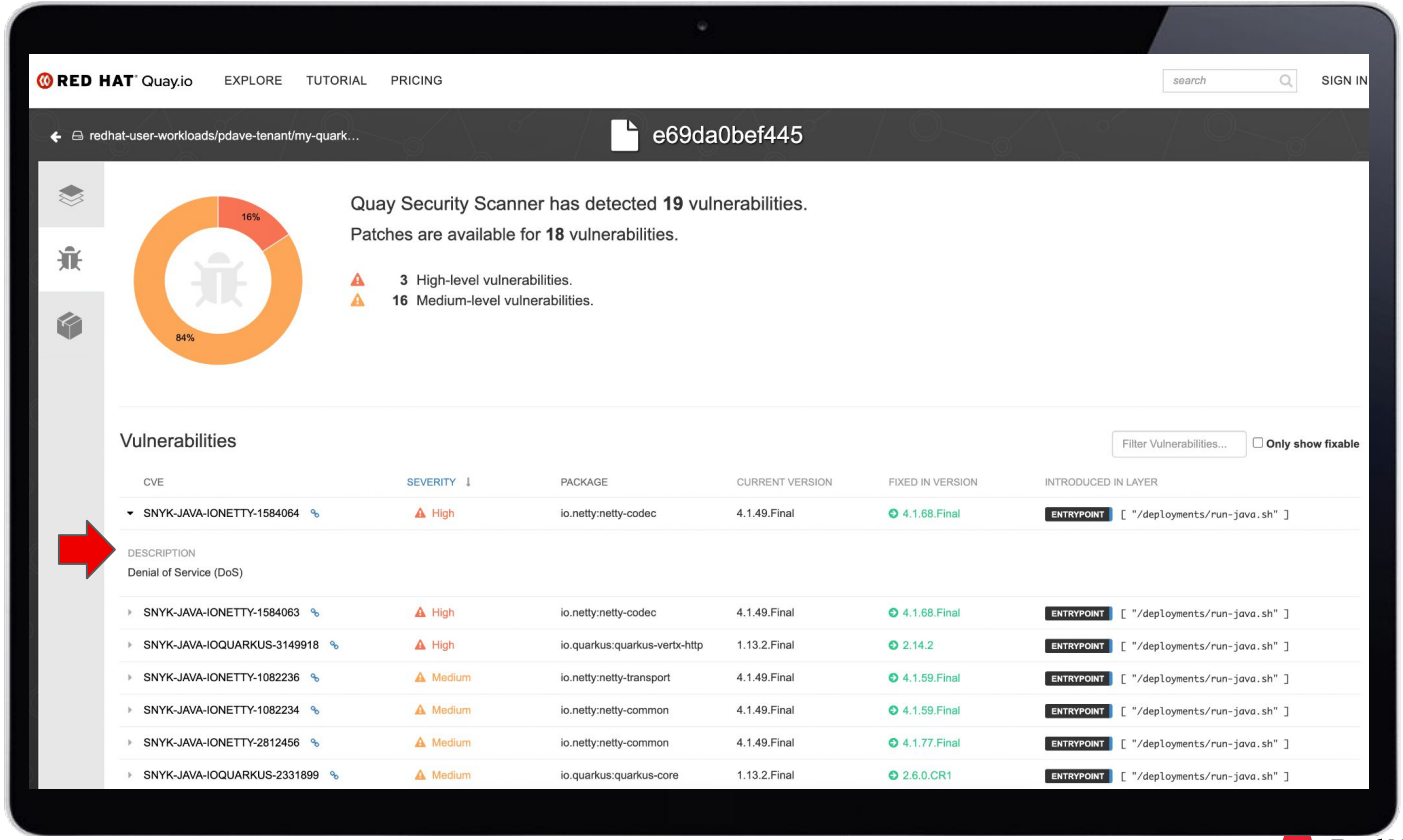
View the SBOM and verify software components



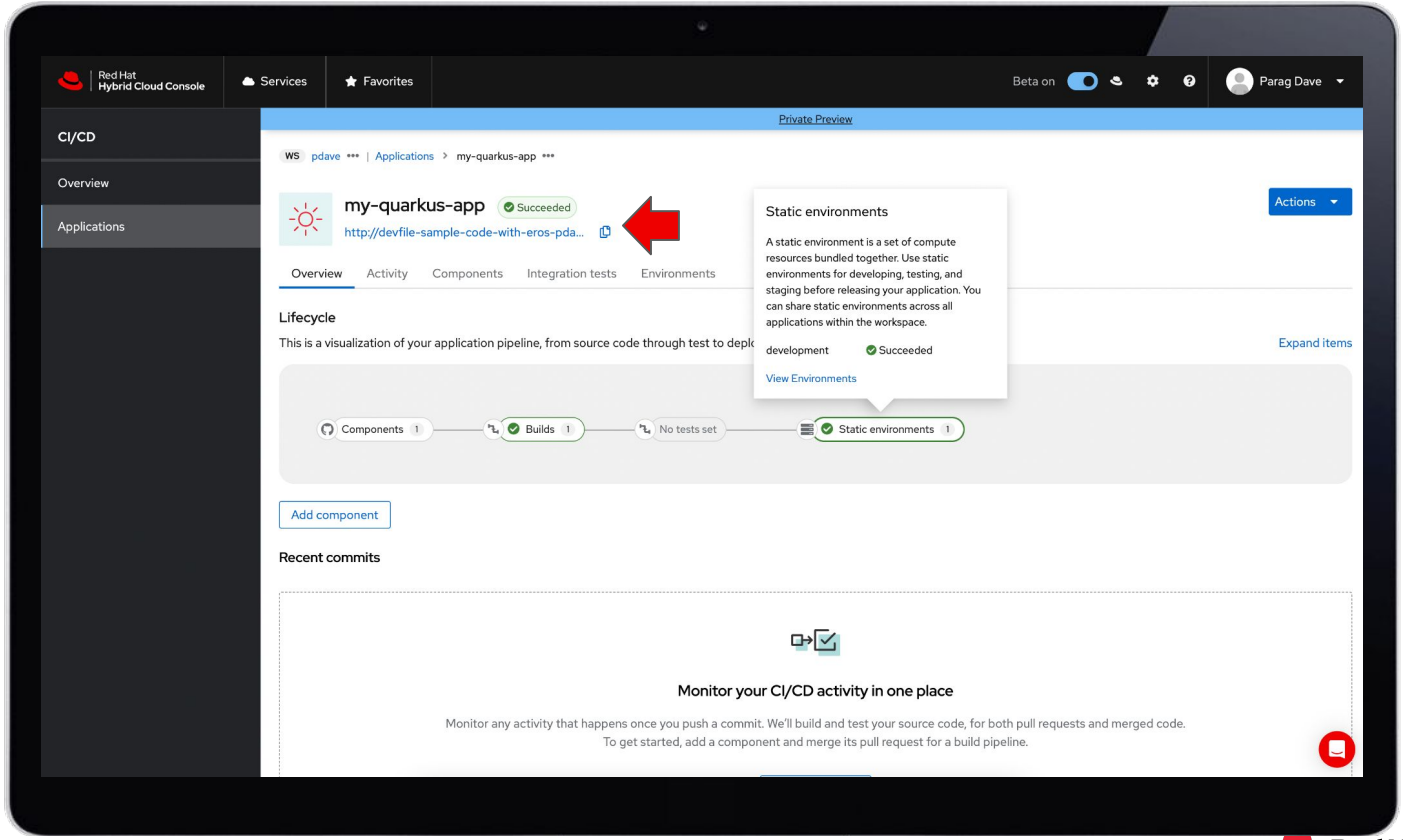
Analyze built image for vulnerabilities



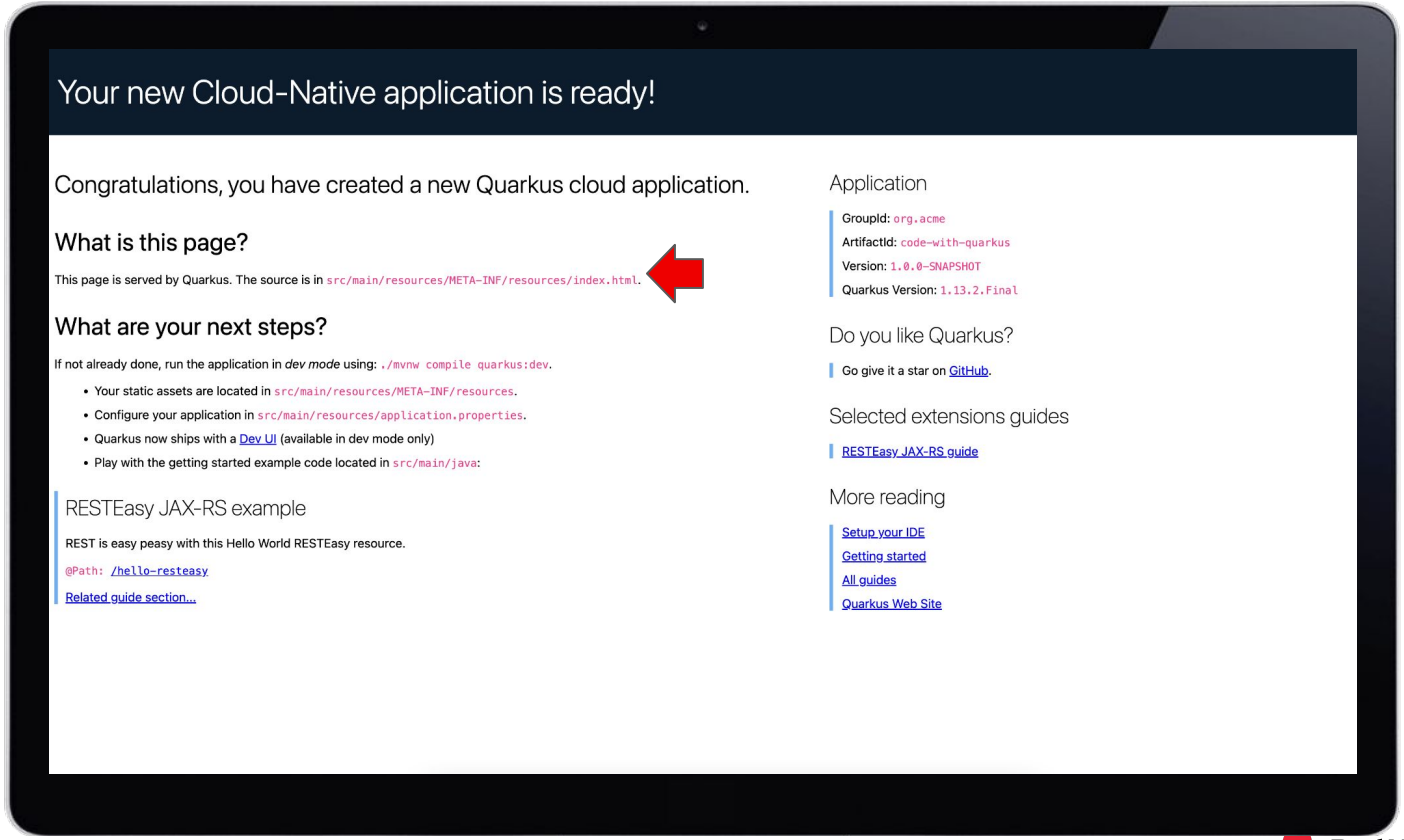
Drill down on details of vulnerabilities detected



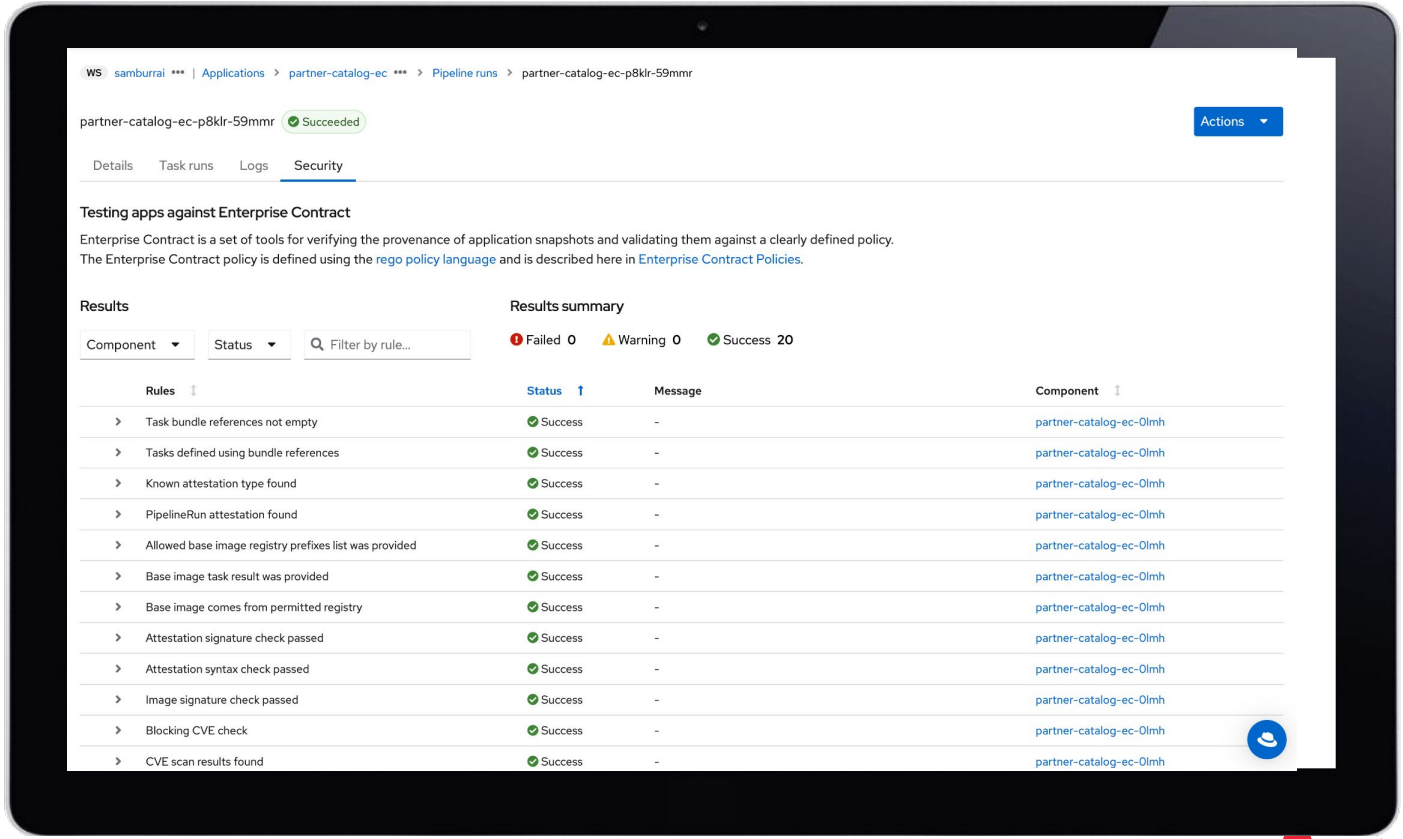
Access application deployed to a bundled Kubernetes environment



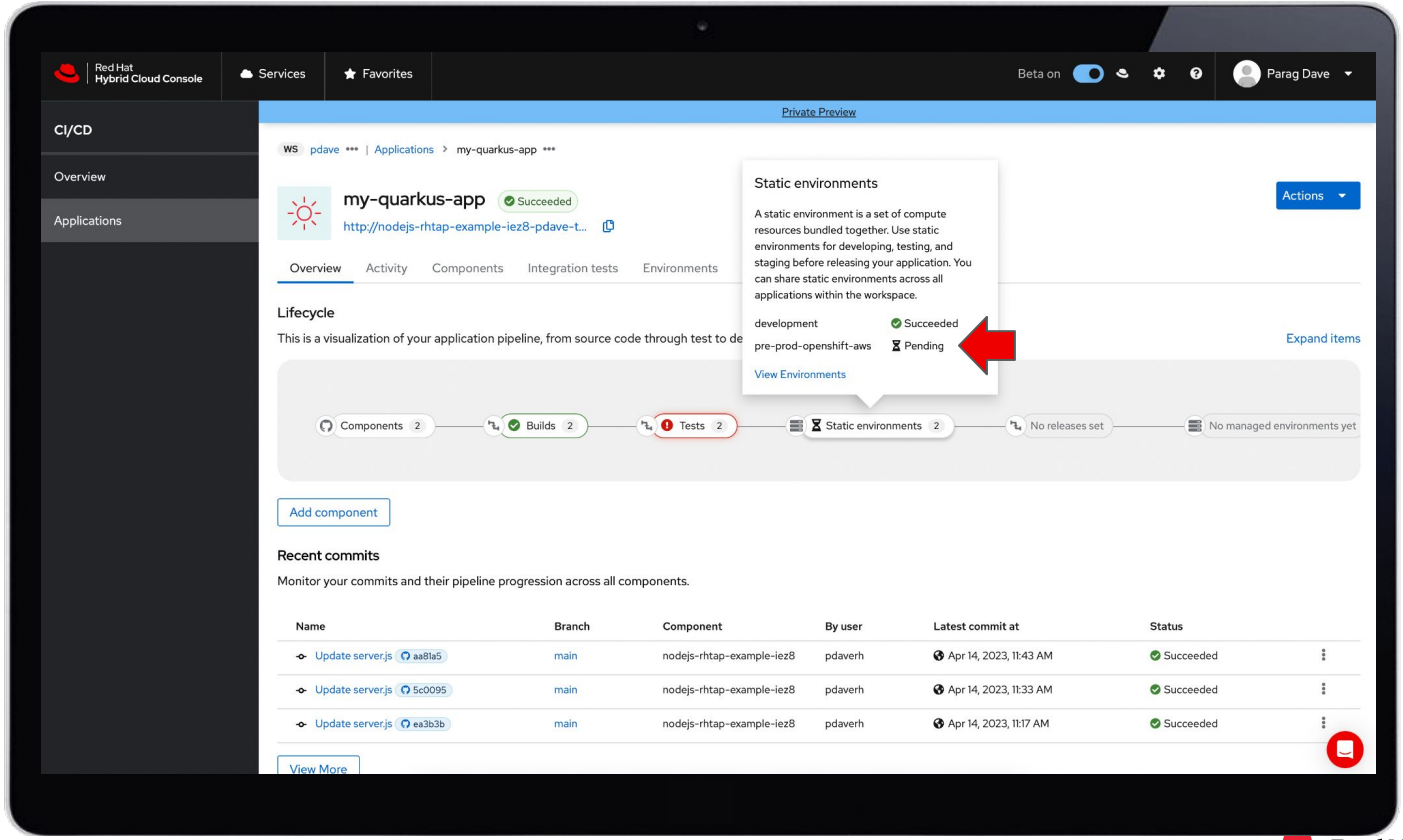
Share and collaborate on the application from a unique URL



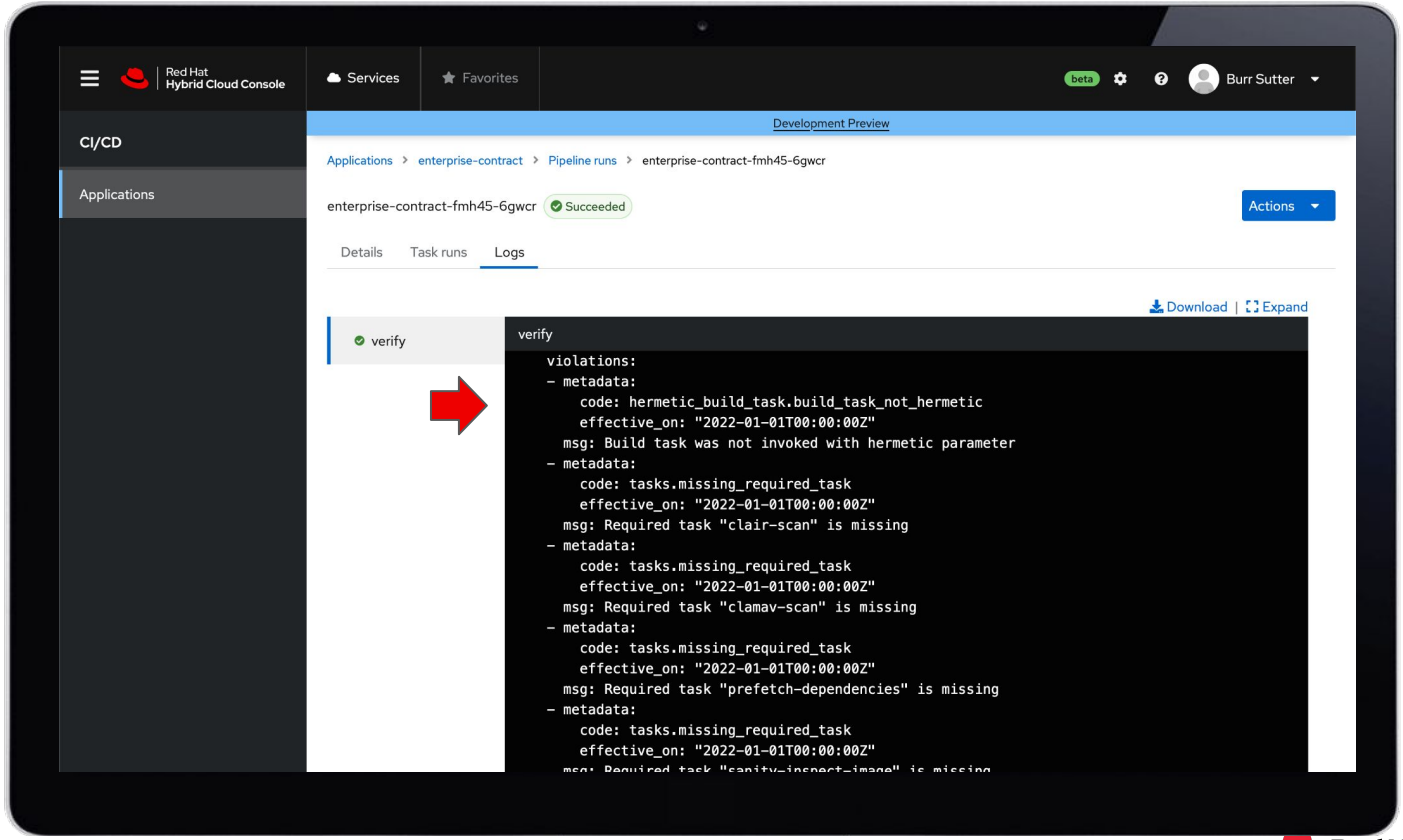
Set approval gates using enterprise contracts available out-of-the-box



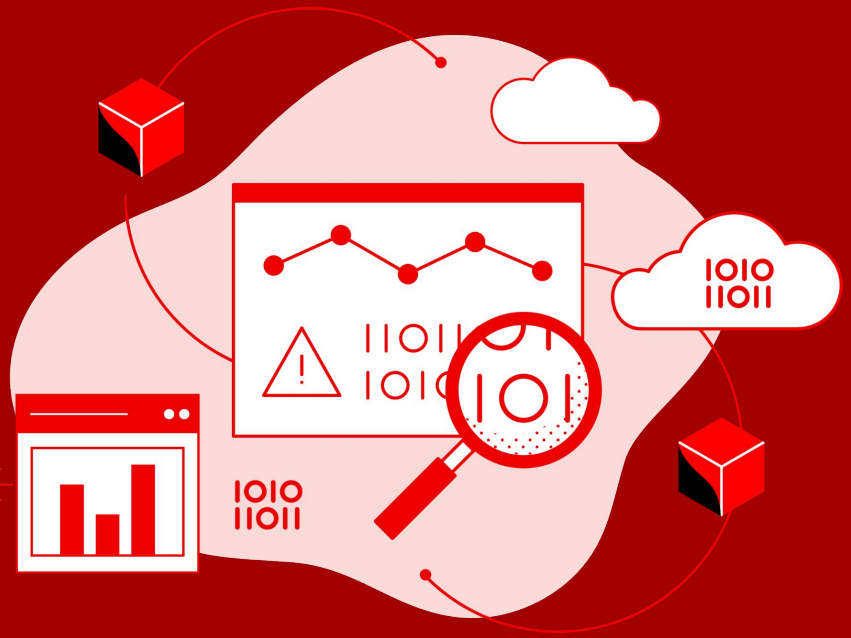
Suspicious build activity is automatically blocked from production



Check details of flagged items in Enterprise Service Contract



TEMO

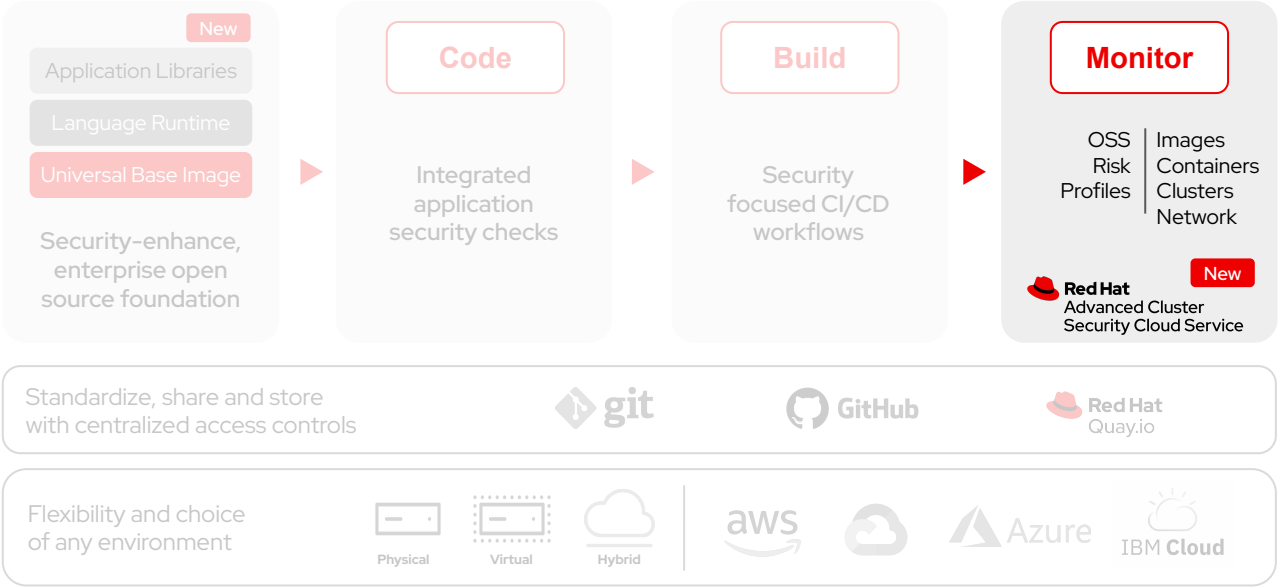


Continuously monitor security at runtime

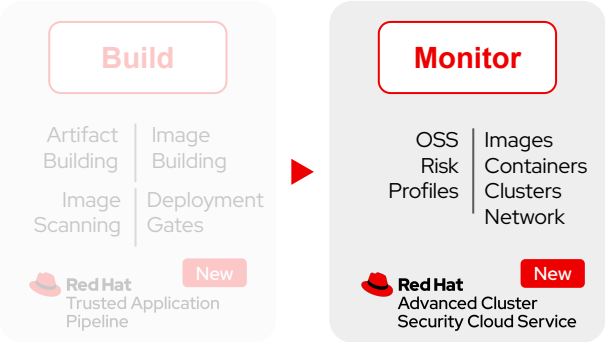
Red Hat

Continuous security monitoring at runtime

Cut down alert noise, fatigue to eliminate production downtimes



Monitor and identify runtime security incidents

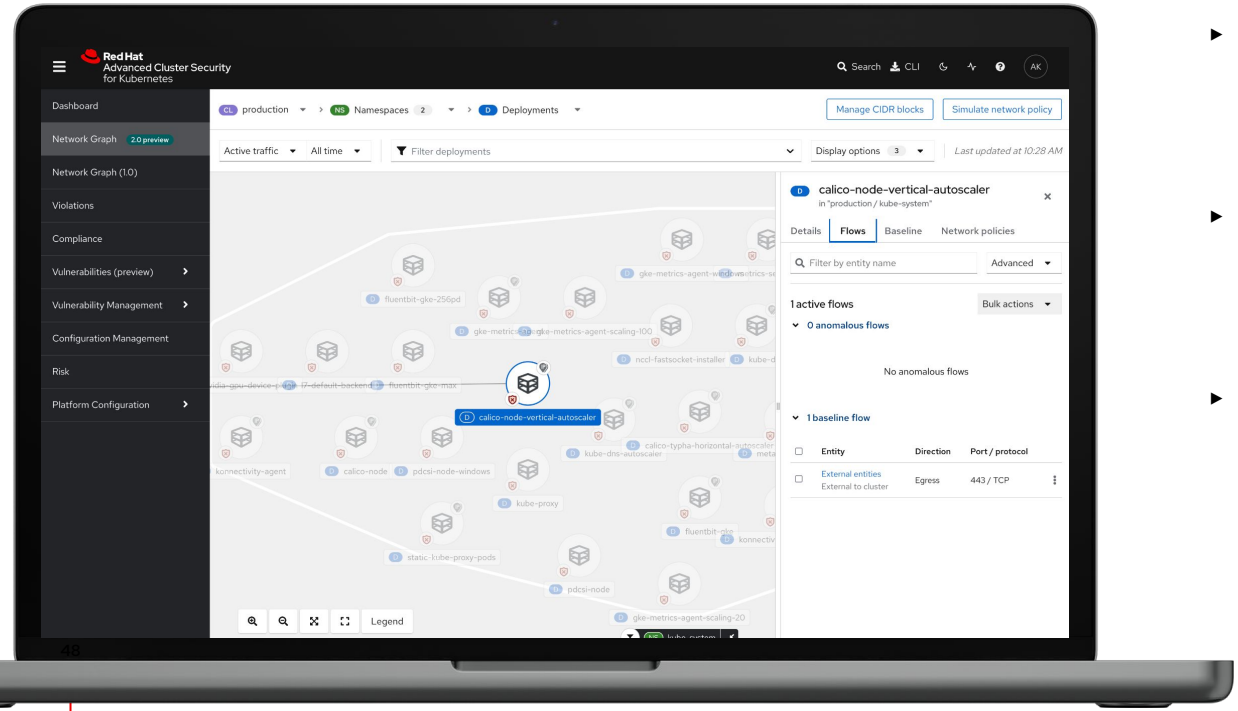


Reduce noise, alert fatigue for shorter time to response

- Continuous improvement from runtime to build
- Detect and respond to suspicious activity
- Runtime vulnerability scanning and management
- Audit for compliance across hundreds of controls
- Expedite incident response to reduce down times

Red Hat

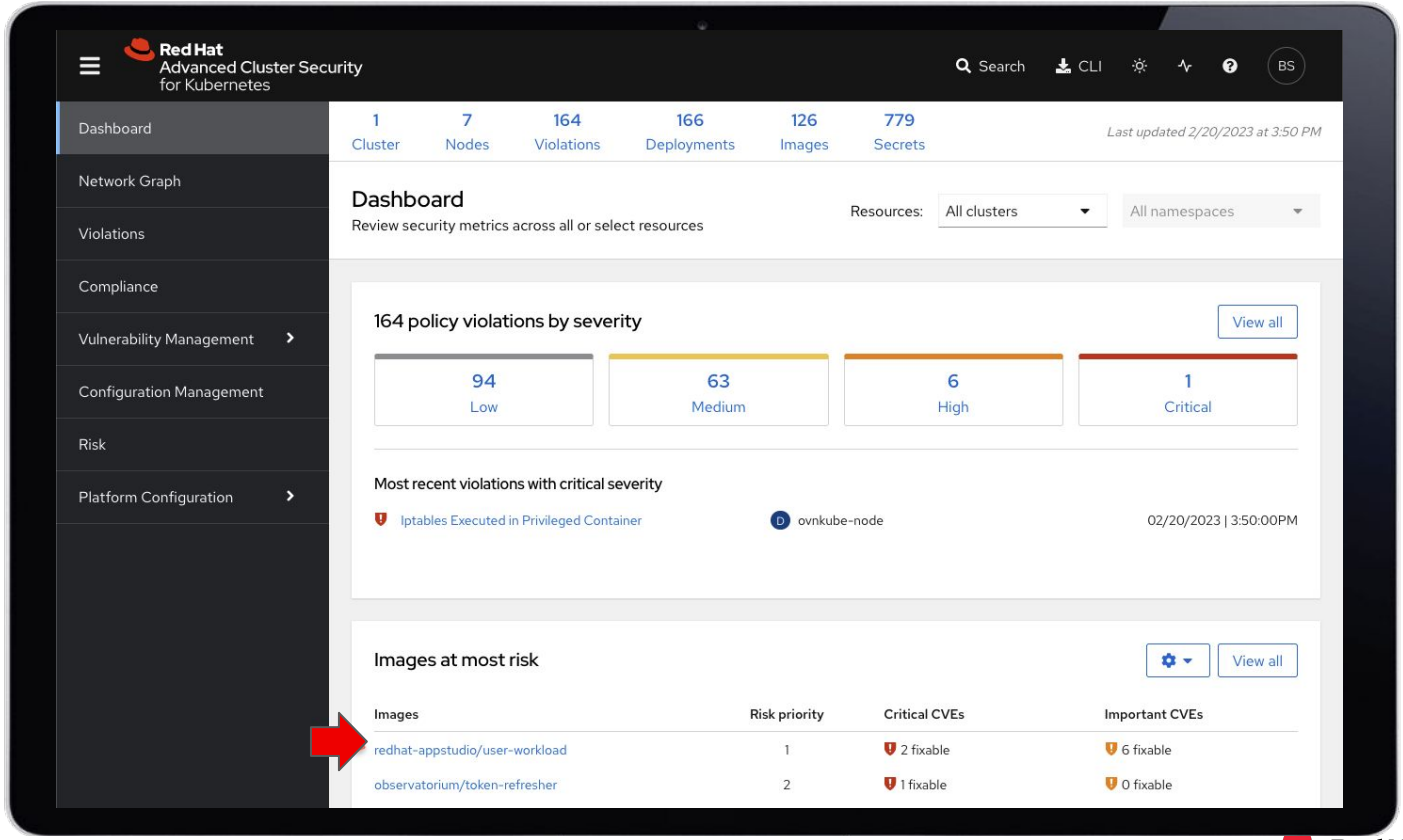
Continuous runtime security and response with minimal false positives



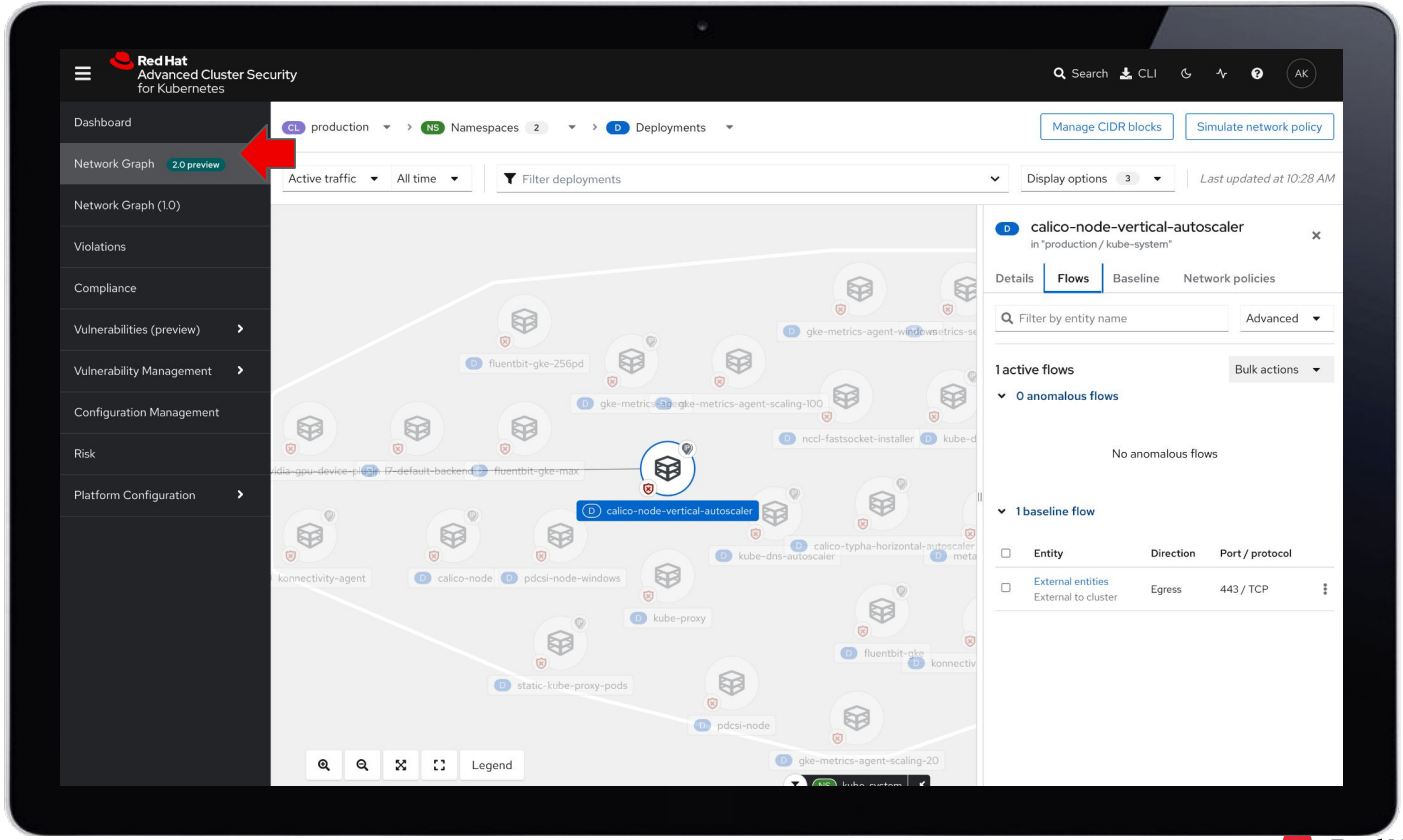
- Prevent high risk workloads from being deployed or running using OOTB deploy-time and runtime policies
- Harden workloads by enforcing network policies in accordance with the principles of least privilege
- Monitor for anomalous behavior indicative of a threat, and configure custom policies and responses, providing feedback to developers

Red Hat

Continuously monitor for anomalous behaviour at runtime

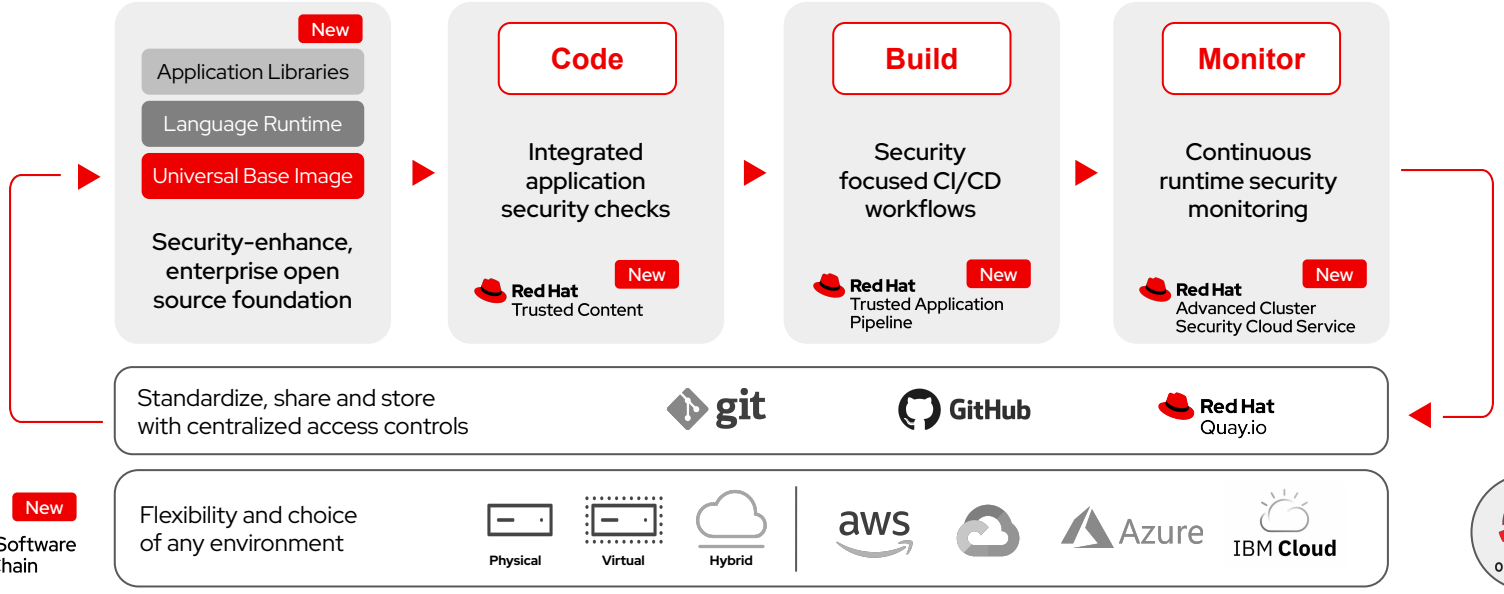


Harden workloads by enforcing network policies



Layered security throughout the stack and lifecycle

Achieve business agility while meeting security requirements



Red Hat
Trusted Software
Supply Chain
New

TEMO

Thank you

Learn more

red.ht/trusted


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat