

Securing your Software Supply Chain

Matching supply chain resiliency with innovation speed



About this session



François Duthilleul
Principal Solution Architect
EMEA Telco team



Focus on customers/partners security questions*

Technical interface to some security agencies**

Member of O-RAN WG11 (Security Work Group)

Leading Edge Security WG within Red Hat

What is software supply chain security?

by François Duthilleul

Software supply chain security combines best practices from risk management and cybersecurity to help protect the software supply chain from potential vulnerabilities. Francois will talk us through all aspects:

- ◆ Why software supply chain security is critical?
- ◆ What are the security risks to the software supply chain?
- ◆ DevSecOps and software security
- ◆ Software supply chain security v.s. application security

* For customers and partners based in EMEA through direct interactions or RFX

** Mostly ANSSI (France) and NCSC (UK)



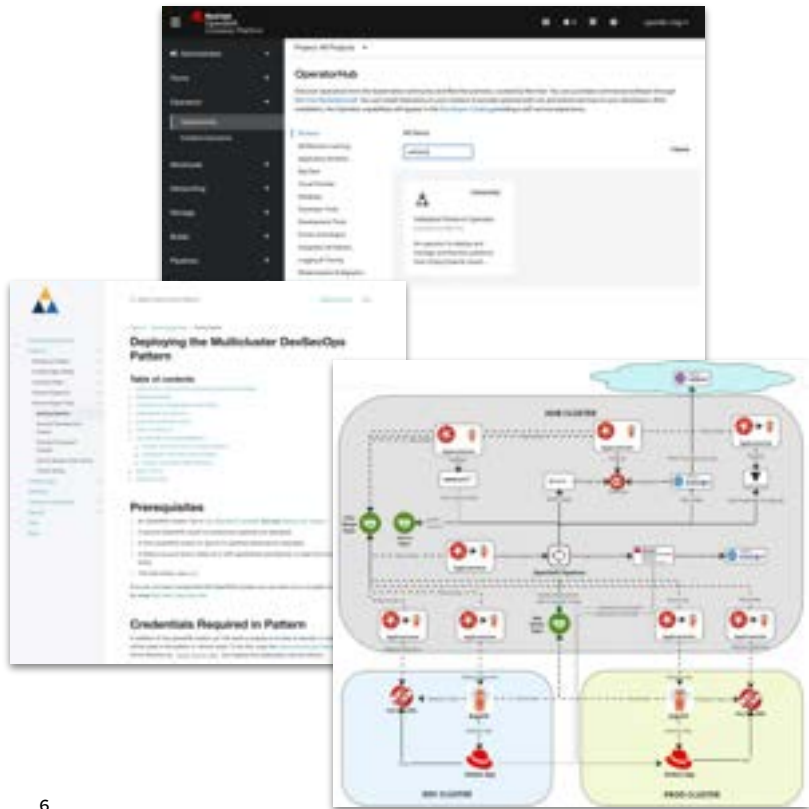


What is SW Supply Chain Security ?

DevSecOps vs SW Supply Chain Security

- ▶ **Both concepts address security in the software development process (SDLC).**
They are closely related but have a different focus area.
- ▶ **DevSecOps** combines the principles of DevOps—which emphasizes collaboration and automation between development and operations teams—with security practices to create a culture of security within the software development life cycle.
- ▶ **SW Supply Chain Security** is to identify and mitigate risks associated with the software supply chain, including the potential for malicious or compromised components. This involves ensuring the integrity, authenticity, and confidentiality of software components, as well as monitoring and managing the dependencies and third-party libraries used in software development.

If you want to know more on these topics...




▶ **DevSecOps**

- [What is DevSecOps ?](#)
- [Validated Pattern "Multicluster DevSecOps"](#)
- <https://red.ht/devsecops>

▶ **Software Supply Chain Security**

- [What is software supply chain security ?](#)
- <https://red.ht/trusted>



Why software
supply chain
security is critical ?



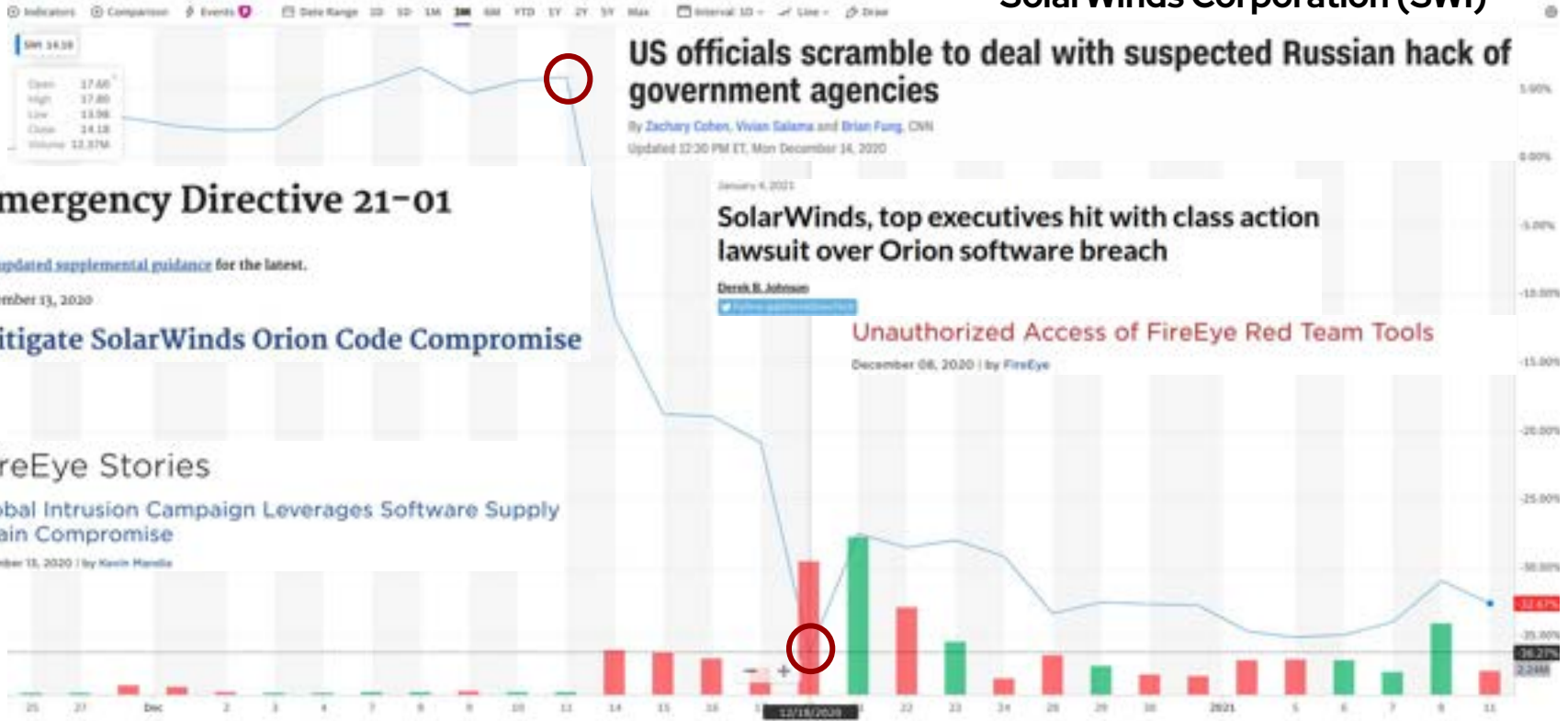
When we buy a car we expect every part supplied to be **genuine** ...



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

Software Supply Chain Security got full attention after the SolarWinds attack...

SolarWinds Corporation (SWI)



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 years¹

20%

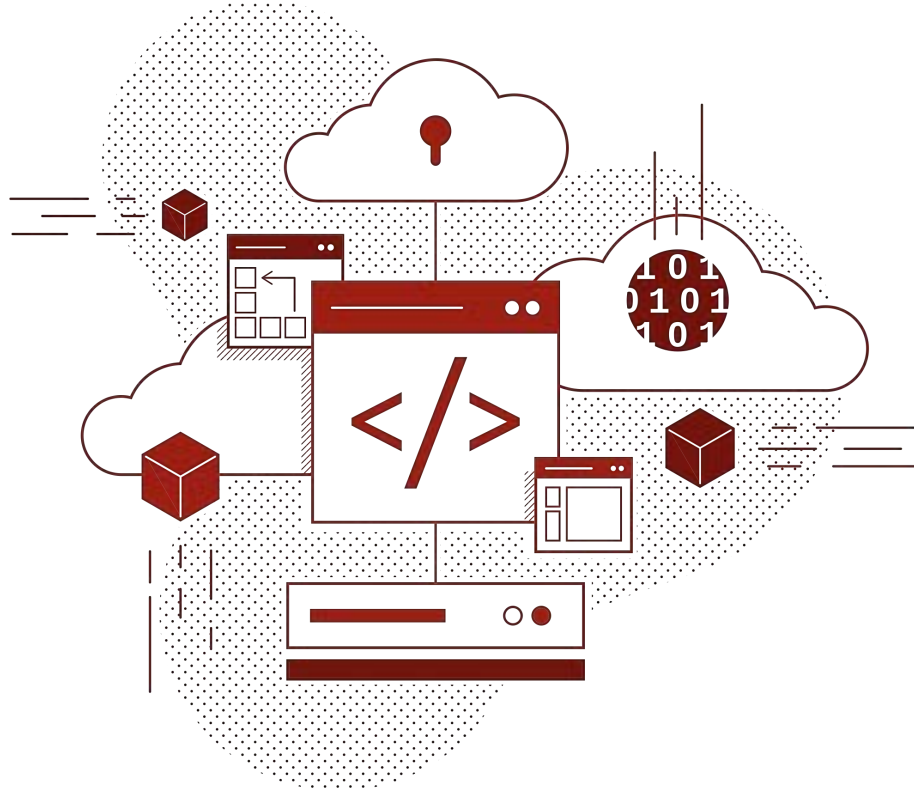
data breaches are due to a
compromised software
supply chain²

1 in 5

data breaches are due to
a software supply chain
compromise³

71%

YoY increase in cost
of average ransom
payment⁴



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



Typosquatting Attack

This attack uses slight misspellings of popular package names to get a victim to install a compromised package. The package is typically a clone of the original one but with additional malicious functionality.

django becomes "diango," "djago," "dajngo," etc.



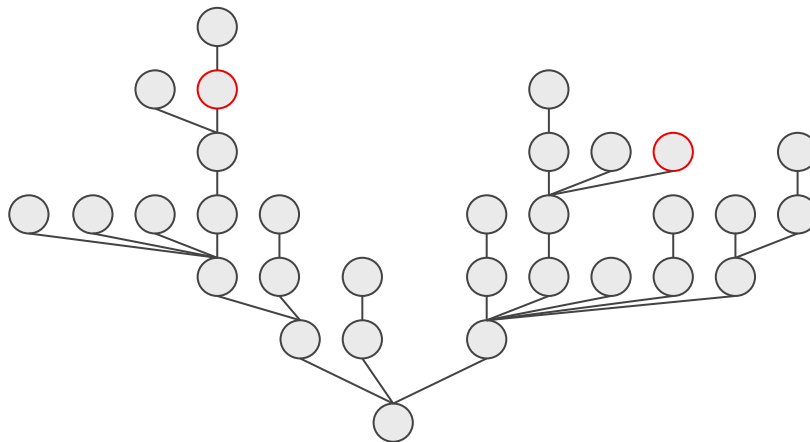
Dependency confusion attack

A software supply chain attack that substitutes malicious third-party code for a legitimate internal software dependency resulting in usage of a compromised package

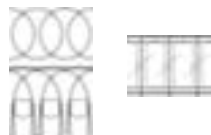
A security researcher* was able to compromise the builds of Apple, Google, Paypal and others by uploading packages to public repositories with the **same name** as internally built packages, but **higher version numbers**.



```
52 - python -m pip install helix-scripts --extra-index-url  
https://dnceng.pkgs.visualstudio.com/public/_packaging/helix-client-prod/pypi/simple
```



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



pom.xml
package.json
requirements.txt
Dockerfile

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:slf4j-to-slf4j:jar:1.7.5:compile
[INFO] | | +- jakarta.annotation:jakarta.annotation-api:jar:1.3.5:compile
[INFO] | | | \- org.yaml:snakeyaml:jar:1.28:compile
[INFO] | +- org.springframework.boot:spring-boot-starter-jooq: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:9.0.78:compile
[INFO] | | +- org.apache.tomcat.embed:tomcat-embed-el:jar:9.0.78:compile
[INFO] | | | \- org.apache.tomcat.embed:tomcat-embed-websocket:jar:9.0.78: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-asm: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:1.7.5:compile
[INFO] | +- jakarta.xml.bind:jakarta.xml.bind-api:jar:2.3.2: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]

```




Compromised Build Environment

This attack alters build to include modified source (not matching source repo)

Webmin: An attacker modified the build infrastructure to use source files not matching source control.

SolarWinds: An attacker compromised the build platform and installed an implant that injected malicious behavior during each build.

your laptop!: is a terrible place to build software....

Governments around the world are raising the bar

Executive Order (EO) 14028

Improving the Nation's Cybersecurity

- ▶ establishes baseline security standards for development of software sold to the government.
- ▶ charges multiple agencies – including NIST [National Institute of Standards and Technology] with enhancing cybersecurity
- ▶ Section 4 directs NIST to "develop guidelines...which are ultimately aimed at U.S. federal agencies but which also are available for industry and others to use

...doing business with U.S. federal agencies will require SSDF [secure software development framework] compliance.





What are the security risks to the software supply chain?

What is SLSA ?

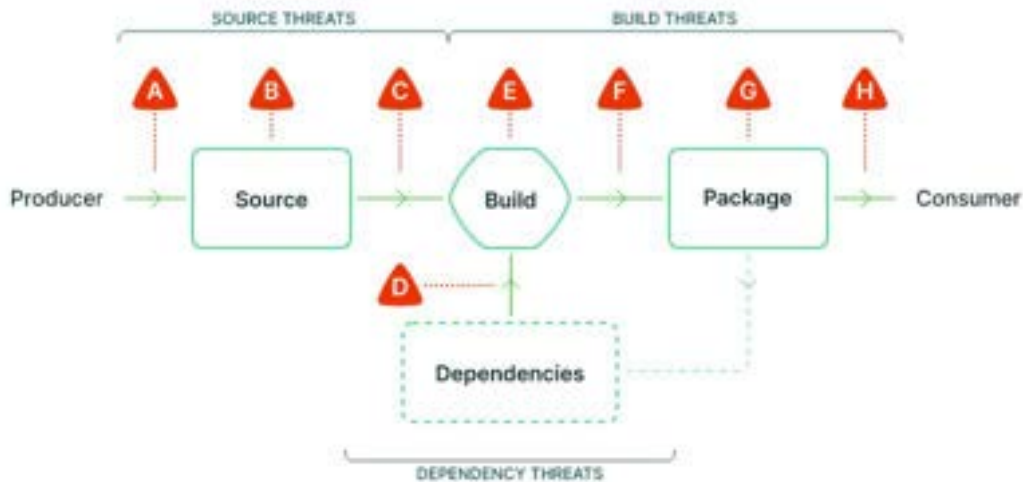


slsa.dev

SLSA = Supply-chain Levels for Software Artifacts

- ▶ Specification for describing and incrementally improving supply chain security
- ▶ Useful both for software producers & consumers
- ▶ Series of levels that describe increasing security guarantees
- ▶ Organized by tracks. Each track covering one area of the SW supply chain
- ▶ SLSA v1.0, first stable release of SLSA

What are the Supply Chain Threats ?



SOURCE THREATS

- A Submit unauthorized change
- B Compromise source repo
- C Build from modified source

DEPENDENCY THREATS

- D Use compromised dependency

BUILD THREATS

- E Compromise build process
- F Upload modified package
- G Compromise package repo
- H Use compromised package

What is SBOM ?



- ▶ SBOM stands for **Software Bill of Materials**
- ▶ List of the ingredients of a software product
- ▶ Available in different format such as CycloneDX¹ and SPDX²
- ▶ SBOM mandated in the Executive Order 14028 “Improving the Nation’s Cybersecurity”
- ▶ Does not inform whether any ingredient is toxic or not



What is a SBOM looking like ?

```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.4",
  "version": 1,
  "metadata": {
    "timestamp": "2023-03-31T06:38:54Z",
    "tools": [
      {
        "vendor": "Quarkus Community",
        "name": "Quarkus Domino Api 0.0.82 SBOM Generator",
        "version": "0.0.82",
        "hashes": [
          {
            "alg": "SHA-512",
            "content":
"70fa276575cceb96325b0f810c2c02a6c5ed7eb6d238a2a62792
658e9dafc1c440dalc9e435bd2afd8004c36dd38leda08c90b9ca
893e857f7edeef317ca6d60"
          }
        ]
      }
    ]
  },
  "component": {
    "group": "com.redhat.quarkus.platform",
    "name": "quarkus-bom",
    "version": "2.13.7.Final-redhat-00003",
    "description": "Red Hat Build of Quarkus - Kubernetes Native
Java stack tailored for OpenJDK HotSpot and GraalVM",
    "licenses": [
      {
        "license": {
          "id": "Apache-2.0"
        }
      }
    ],
    "cpe": "cpe:/a:redhat:quarkus:2.13:el8",
    "purl":
"pkg:maven/com.redhat.quarkus.platform/quarkus-bom@2.13.7.Fi
nal-redhat-00003?type=pom",
    "releaseNotes": [
      {
        "type": "Patch",
        "title": "Red Hat Build of Quarkus 2.13.7.Final",
        "aliases": [
          "RHBO",
          "Quarkus",
          "Fireball"
        ]
      }
    ],
    "properties": [
      {
        "name": "product-name",
        "value": "Red Hat Build of Quarkus"
      },
      {
        "name": "product-version",
        "value": "2.13.7.Final-redhat-00003"
      }
    ],
    "type": "framework"
  },
}
```

```
"components": [
  {
    "publisher": "redhat",
    "group": "com.redhat.quarkus.platform",
    "name": "quarkus-bom",
    "version": "2.13.7.Final-redhat-00003",
    "description": "Quarkus Universe platform aggregates extensions from
Quarkus Core and those developed by the community into a single
compatible and versioned set that application developers can reference
from their applications to align the dependency versions",
    "licenses": [
      {
        "license": {
          "id": "Apache-2.0"
        }
      }
    ],
    "purl":
"pkg:maven/com.redhat.quarkus.platform/quarkus-bom@2.13.7.Final-red
hat-00003?type=pom",
    "externalReferences": [
      {
        "type": "website",
        "url":
"https://github.com/quarkusio/quarkus-platform/quarkus-platform-config"
      },
      {
        "type": "distribution",
        "url": "https://maven.repository.redhat.com/ga/"
      },
      {
        "type": "issue-tracker",
        "url": "https://github.com/quarkusio/quarkus/issues/"
      },
      {
        "type": "vcs",
        "url":
"https://code.engineering.redhat.com/gerrit/quarkusio/quarkus-platform
.git"
      }
    ],
    "properties": [
      {
        "name": "package:type",
        "value": "maven"
      },
      {
        "type": "library",
        "bom-ref":
"pkg:maven/com.redhat.quarkus.platform/quarkus-bom@2.13.7.Final-red
hat-00003?type=pom"
      },
      {
        "publisher": "JBoss by Red Hat",
        "group": "io.quarkiverse.config",
        ...
      }
    ]
  }
]
```

```
"dependencies": [
  {
    "ref":
"i.q.quarkus-bootstrap-runner:2.13.7.Final-redhat-00003
#1",
    "dependsOn": [
      "pkg:maven/io.smallrye.common/smallrye-common-io@1.13.1.redhat-00001?type=jar"
    ]
  },
  {
    "ref":
"pkg:maven/io.smallrye.common/smallrye-common-expression@1.13.1.redhat-00001?type=jar",
    "dependsOn": [
      "pkg:maven/io.smallrye.common/smallrye-common-function@1.13.1.redhat-00001?type=jar"
    ]
  },
  {
    "ref":
"i.s.c.smallrye-config-core:2.12.3.redhat-00001#2",
    "dependsOn": [
      "pkg:maven/io.smallrye.common/smallrye-common-class-loader@1.13.1.redhat-00001?type=jar",
      "pkg:maven/io.smallrye.common/smallrye-common-expression@1.13.1.redhat-00001?type=jar",
      "pkg:maven/io.smallrye.config/smallrye-config-common@2.12.3.redhat-00001?type=jar"
    ]
  },
  ...
]
```



Source: <https://cyclonedx.org/docs/1.4/json/>

What is VEX ?



- ▶ VEX stands for **Vulnerability Exploitability eXchange**
- ▶ Includes the product's status as it relates to a particular vulnerability (Not Affected, Affected, Fixed, Under Investigation)
- ▶ Machine readable vulnerabilities-related advisories
- ▶ Developed by the OASIS CSAF¹ (Common Security Advisory Framework)



What is a VEX looking like ?

```
{
  "document": {
    "category": "csaf_vex",
    "csaf_version": "2.0",
    "lang": "en",
    "notes": [
      {
        "category": "vex_security_data",
        "text": "Red Hat Security",
        "title": "Red Hat Product Security"
      },
      {
        "category": "legal_disclaimer",
        "text": "This content is licensed under the Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original.",
        "title": "Terms of Use"
      }
    ],
    "publisher": {
      "category": "vendor",
      "contact_details": {
        "name": "Red Hat Product Security",
        "namespace": "https://www.redhat.com"
      },
      "title": "Red Hat Security data for CVE-2023-0044",
      "tracking": {
        "current_release_date": "2023-01-04T11:00:00.000Z",
        "generator": {
          "date": "2023-01-04T11:00:00.000Z",
          "engine": {
            "name": "Red Hat SDEngine",
            "version": "3.10.0"
          }
        },
        "id": "CVE-2023-0044",
        "initial_release_date": "2022-07-12T11:00:00.000Z",
        "revision_history": [
          {
            "date": "2023-01-04T11:00:00.000Z",
            "number": "1",
            "summary": "Initial version."
          },
          {
            "date": "2023-03-24T11:00:00.000Z",
            "number": "5",
            "summary": "Last update."
          }
        ],
        "status": "final",
        "version": "1"
      }
    }
  }
}
```

Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). If you distribute this content, or a modified version of it, you must provide attribution to Red Hat Inc. and provide a link to the original.

```
"product_tree": {
  "branches": [
    {
      "branches": [
        {
          "branches": [
            {
              "category": "product_name",
              "name": "Red Hat Build of Quarkus",
              "product": {
                "name": "Red Hat Build of Quarkus",
                "product_id": "8Base-RHBQ-2.13",
                "product_identification_helper": {
                  "cpe": "cpe:/a:redhat:quarkus:2.13:el8"
                }
              }
            }
          ]
        },
        {
          "category": "product_family",
          "name": "Red Hat build of Quarkus (RHBQ)"
        }
      ]
    },
    {
      "branches": [
        {
          "category": "product_version",
          "name": "quarkus-vertx-http",
          "product": {
            "name": "quarkus-vertx-http:2.13.7.Final-redhat-00003",
            "product_id": "quarkus-vertx-http:2.13.7.Final-redhat-00003",
            "product_identification_helper": {
              "purl": "pkg:maven/io.quarkus/quarkus-vertx-http@2.13.7.Final-redhat-00003?type=jar"
            }
          }
        }
      ]
    },
    {
      "category": "vendor",
      "name": "Red Hat"
    }
  ],
  "relationships": [
    {
      "category": "default_component_of",
      "full_product_name": {
        "name": "quarkus-vertx-http:2.13.7.Final-redhat-00003 as a component of Red Hat build of Quarkus (RHBQ)",
        "product_id": "8Base-RHBQ-2.13:quarkus-vertx-http:2.13.7.Final-redhat-00003"
      },
      "product_reference": "quarkus-vertx-http:2.13.7.Final-redhat-00003",
      "relates_to_product_reference": "8Base-RHBQ-2.13"
    }
  ]
}
```

```
"vulnerabilities": [
  {
    "cve": "CVE-2023-0044",
    "cwe": [
      {
        "id": "",
        "name": ""
      }
    ],
    "discovery_date": "2023-01-04T00:00:00Z",
    "ids": [
      {
        "system_name": "Red Hat Bugzilla",
        "text": "https://bugzilla.redhat.com/show_bug.cgi?id=2158081"
      }
    ],
    "notes": [
      {
        "category": "general",
        "text": "The CVSS score(s) listed for this vulnerability do not reflect the associated product's status, and are included for informational purposes to better understand the severity of this vulnerability.",
        "title": "CVSS score applicability"
      },
      {
        "category": "description",
        "text": "A flaw was found in Quarkus. If the Quarkus Form Authentication session cookie Path attribute is set to /, then a cross-site attack may be initiated, which might lead to information disclosure.",
        "title": "Vulnerability description"
      },
      {
        "category": "summary",
        "text": "quarkus-vertx-http: a cross-site attack may be initiated which might lead to the Information Disclosure",
        "title": "Vulnerability summary"
      }
    ],
    "product_status": {
      "fixed": [
        "8Base-RHBQ-2.13:quarkus-vertx-http:2.13.7.Final-redhat-00003"
      ],
      "remediations": [
        {
          "category": "vendor_fix",
          "details": "For details on how to apply this update, which includes the changes described in this advisory, refer to:\nhttps://access.redhat.com/articles/11258",
          "product_ids": [
            "8Base-RHBQ-2.13:quarkus-vertx-http:2.13.7.Final-redhat-00003"
          ]
        }
      ]
    }
  }
]
```

LEGEND

Text in green: Elements required by the profile "CSAF Base"
Text in red: Additional required elements to satisfy the profile "VEX"

```
"references": [
  {
    "category": "self",
    "summary": "https://access.redhat.com/errata/RHSA-2023:0758",
    "url": "https://access.redhat.com/errata/RHSA-2023:0758"
  },
  {
    "category": "external",
    "summary": "https://access.redhat.com/security/updates/classification/#moderate",
    "url": "https://access.redhat.com/security/updates/classification/#moderate"
  },
  {
    "category": "external",
    "summary": "https://access.redhat.com/jbosnetwork/restricted/listSoftware.html?product=redhat.quarkus&downloadType=distributions&version=2.13.7",
    "url": "https://access.redhat.com/jbosnetwork/restricted/listSoftware.html?product=redhat.quarkus&downloadType=distributions&version=2.13.7"
  },
  {
    "category": "self",
    "summary": "RHSA-2023:0758 CSAF Advisory",
    "url": "https://access.redhat.com/security/data/csaf/v2/advisories/2023/rhsa-2023:0758.json"
  },
  {
    "scores": [
      {
        "cvss_v3": {
          "attackComplexity": "LOW",
          "attackVector": "NETWORK",
          "availabilityImpact": "NONE",
          "baseScore": 5.3,
          "baseSeverity": "LOW",
          "confidentialityImpact": "LOW",
          "integrityImpact": "NONE",
          "privilegesRequired": "NONE",
          "scope": "UNCHANGED",
          "userInteraction": "NONE",
          "vectorString": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N",
          "version": "3.1"
        },
        "products": [
          "8Base-RHBQ-2.13:quarkus-jdbc-postgresql:2.13.7.Final-redhat-00003"
        ]
      }
    ]
  }
]
```

What is an Attestation ? What is Provenance ?



- ▶ An attestation is an authenticated statement (metadata) about a software artifact or collection of software artifacts.
- ▶ For example, an attestation might state exactly how an artifact was produced, including the build command that was run and all of its dependencies (as in the case of SLSA Provenance).
- ▶ Provenance focuses on documenting the origin and history of an artifact to enhance transparency and traceability.



SLSA v1.0*

single track (build) with three levels

Build L1: Provenance exists

Preventing Mistakes

Build platform automatically generates provenance

Software producer distributes provenance to consumers

What is provenance?

pro·v·e·n·ance: It's the verifiable information about software artifacts describing where, when and how something was produced.

It's the play-by-play of what happened during your build.

In the container world, we can think:

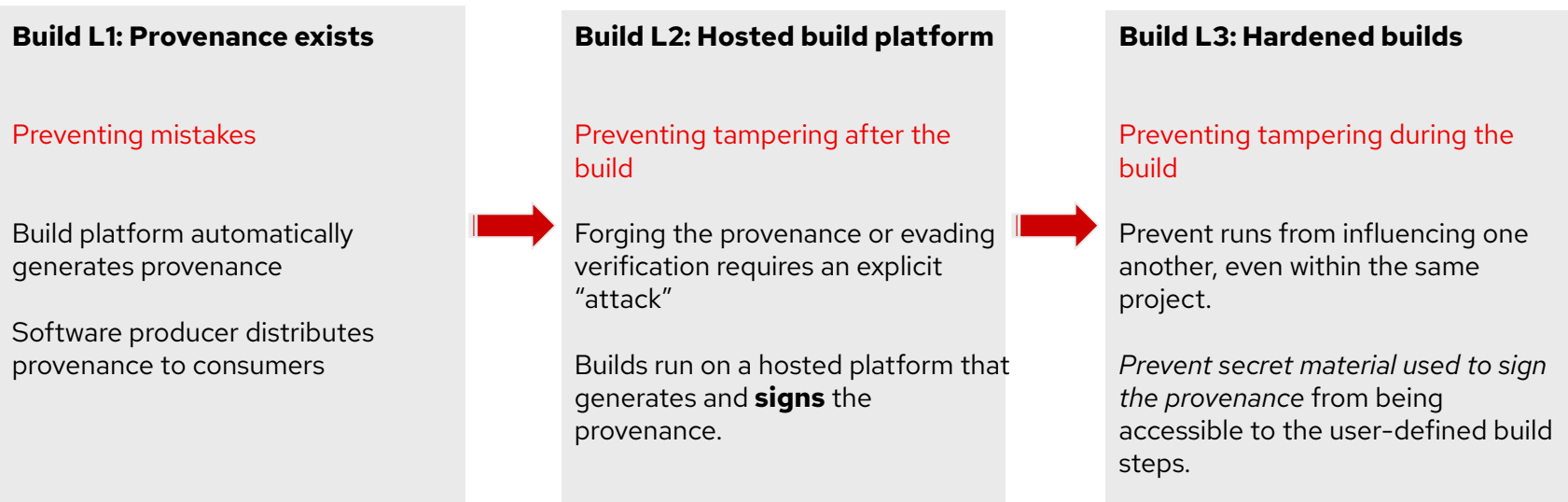
dockerfile = "**what I want to happen**" -> `pip install requests`

and provenance = "**what actually happened**"

installed requests==2.31.0 from PyPI

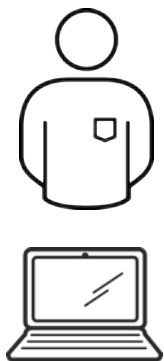
SLSA v1.0*

single track (build) with three levels

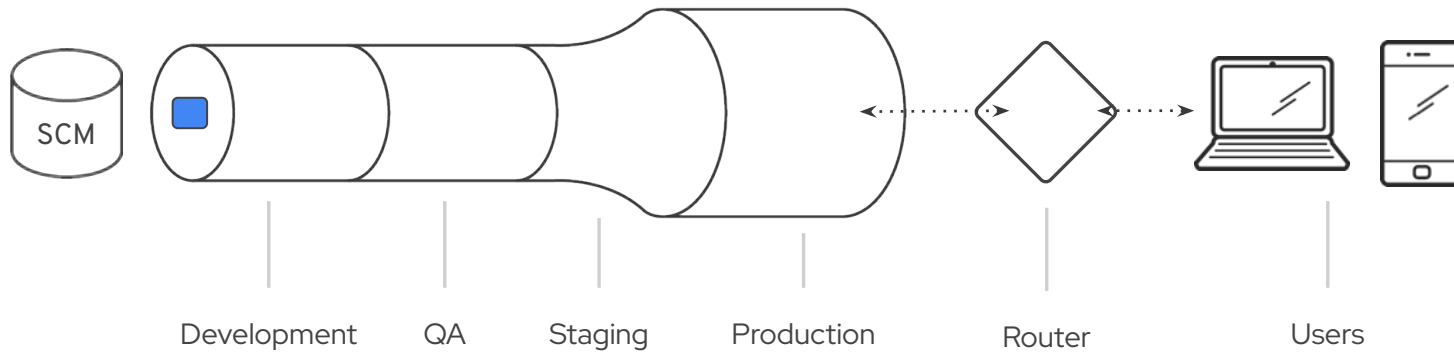


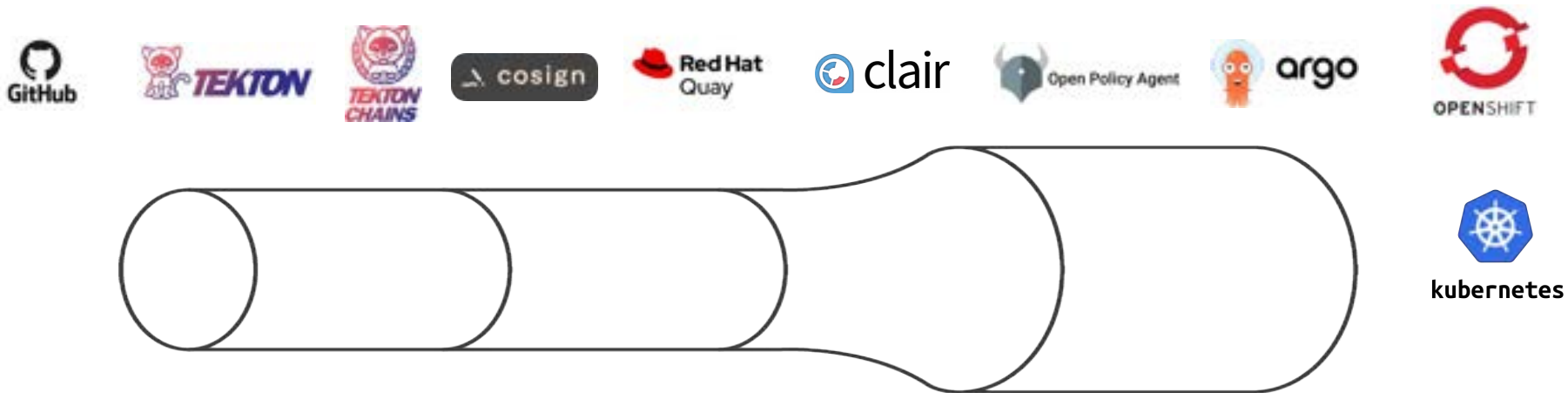


Shift Left



Developer





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.



Red Hat Trusted Software Supply Chain

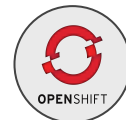
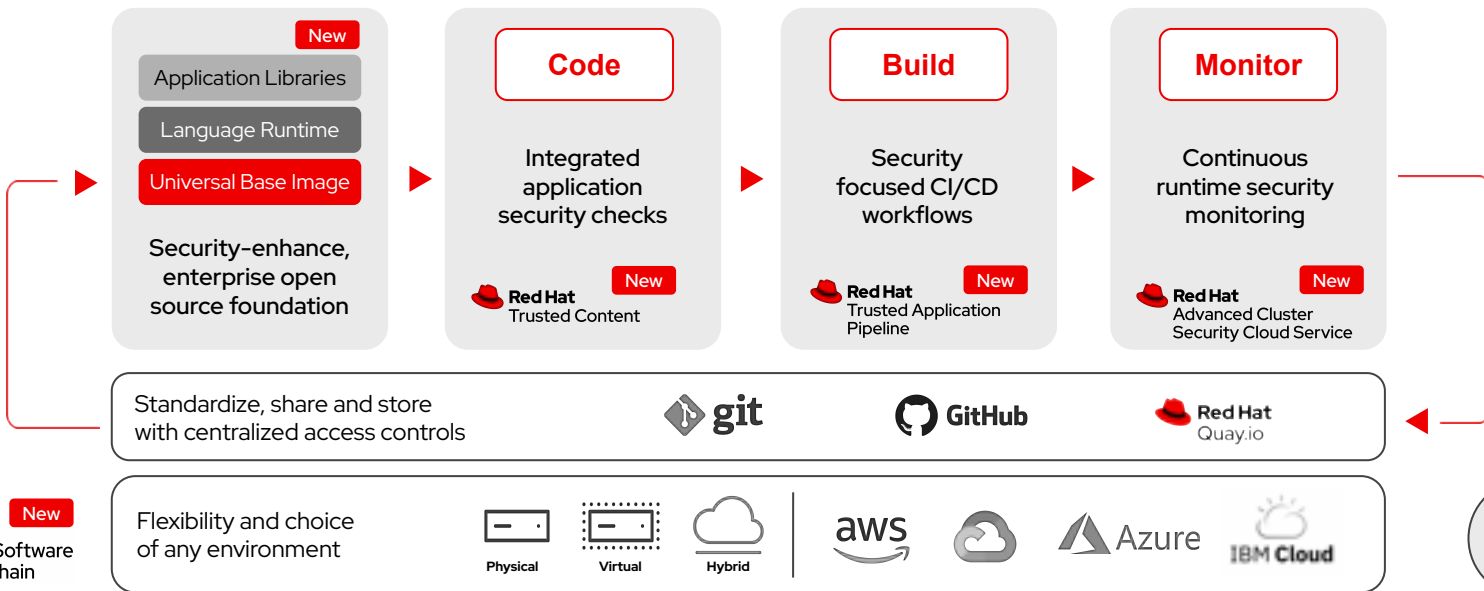
Red Hat: Providing trusted enterprise open source software for 30+ years



- ▶ 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

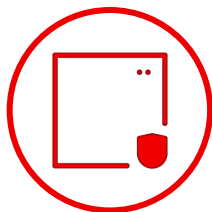
Delivered as a **cloud service** with integrated security guardrails at every phase of the software development lifecycle



Secure the use of source code and transitive dependencies

Software supply chain security considerations for the software development lifecycle

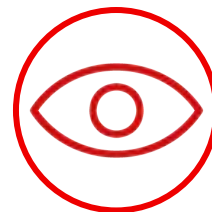
Prevent & identify
malicious **code**



Safeguard **build**
systems early



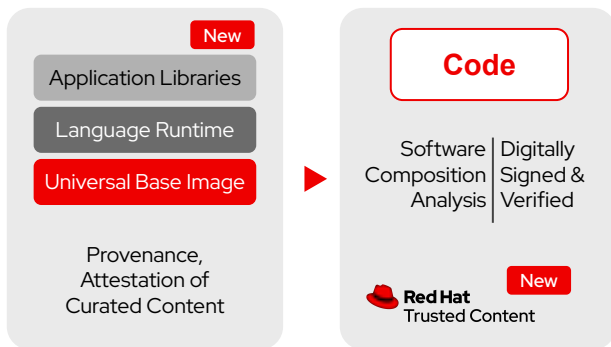
Continuously **monitor**
security at runtime





*Prevent and
identify malicious
code*

Code with integrated application security checks



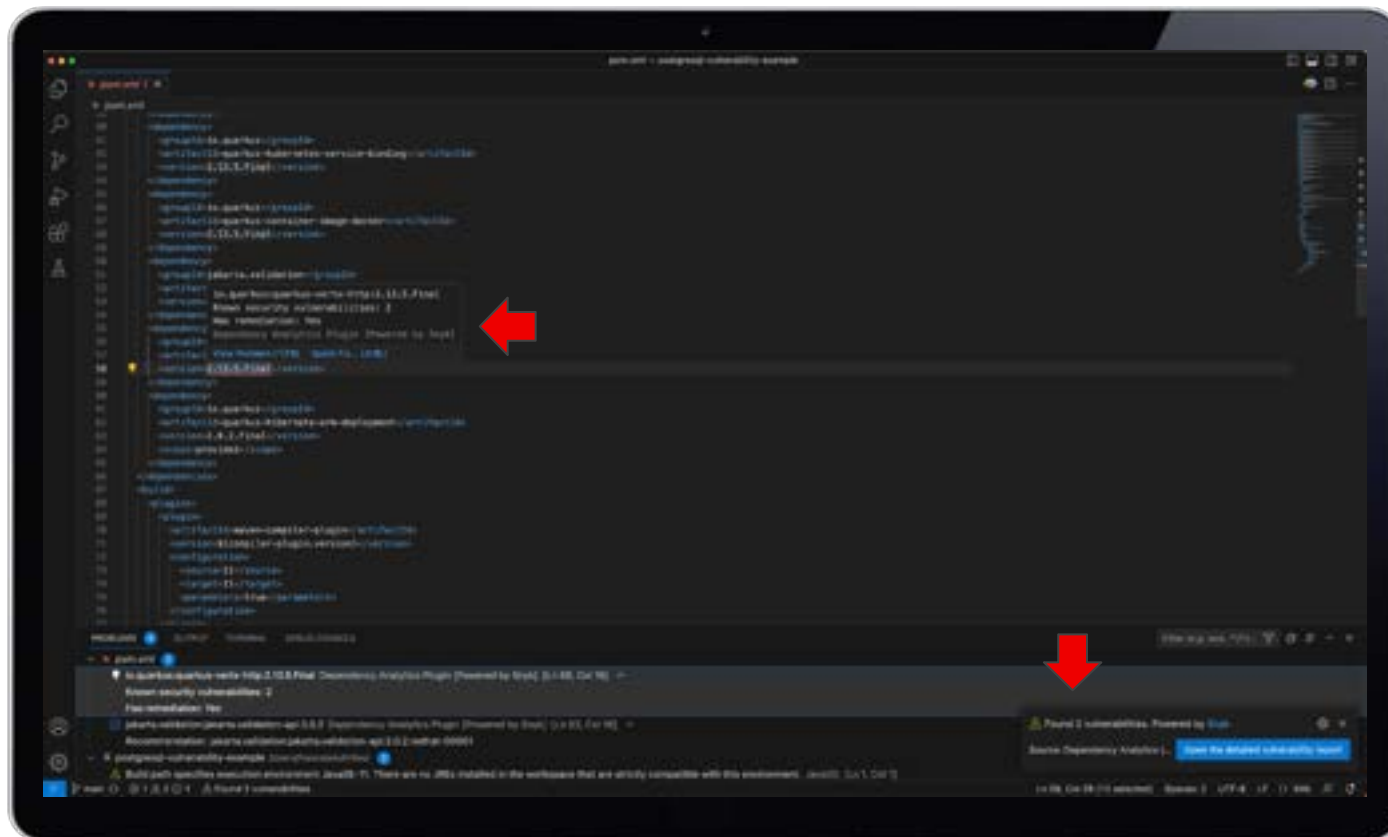
Catch security issues early to
keep and grow user trust

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

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

Remedy vulnerabilities with Trusted Content



Analyze and fix security issues from the IDE

The screenshot displays a software development IDE with a 'Dependency Analysis Report' open. A red arrow points to the 'Dependency Analysis Report' entry in the left-hand sidebar. The main window shows a 'Security Issues' section with the following text:

Below is a list of dependencies affected with CVE, as well as vulnerability only found using Snyk's vulnerability databases.

Dependencies with security issues in your stack.

Dependencies with high common vulnerabilities and exposures (CVE) score.

- Total Vulnerabilities: 10
- Vulnerable Dependencies: 5

Commonly Known Vulnerabilities

#	Dependencies	# Direct	# Transitive	Highest CVSS	Highest Severity	Red Hat remediation available
#1	io.quarkus:quarkus-internal-test-deployment	0	6	9.8/10	CVE-2023-0344	
#2	io.quarkus:quarkus-vertx-http	1	3	7.5/10	CVE-2023-0344	1 Direct

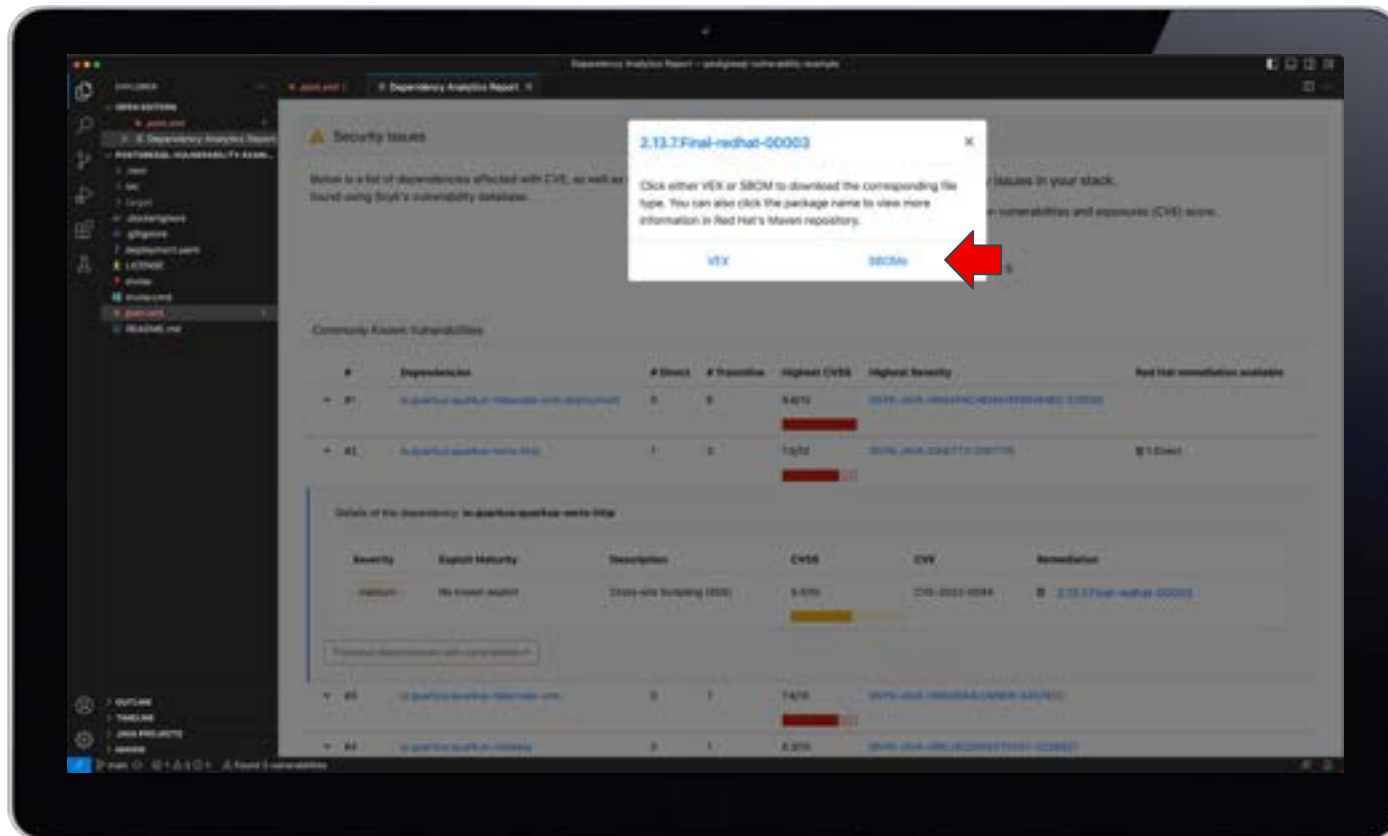
Details of the dependency [io.quarkus:quarkus-vertx-http](#)

Severity	Exploit Maturity	Description	CVSS	CVE	Remediation
Medium	No known exploit	Cross-site Scripting (XSS)	6.4/10	CVE-2023-0344	2.15.7 final medium OOBCE

Transitive Dependencies with vulnerability

#3	io.quarkus:quarkus-internal-test	0	7	7.4/10	CVE-2023-0344
#4	io.quarkus:quarkus-vertx	0	1	5.3/10	CVE-2023-0344

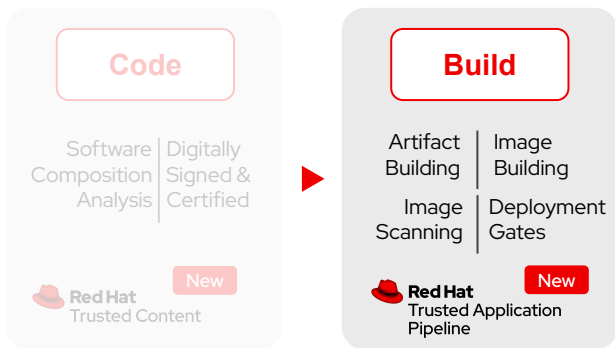
Download the SBOM and VEX files or View more info in the Red Hat repository





Safeguard build systems early

Build with security focused CI/CD workflows



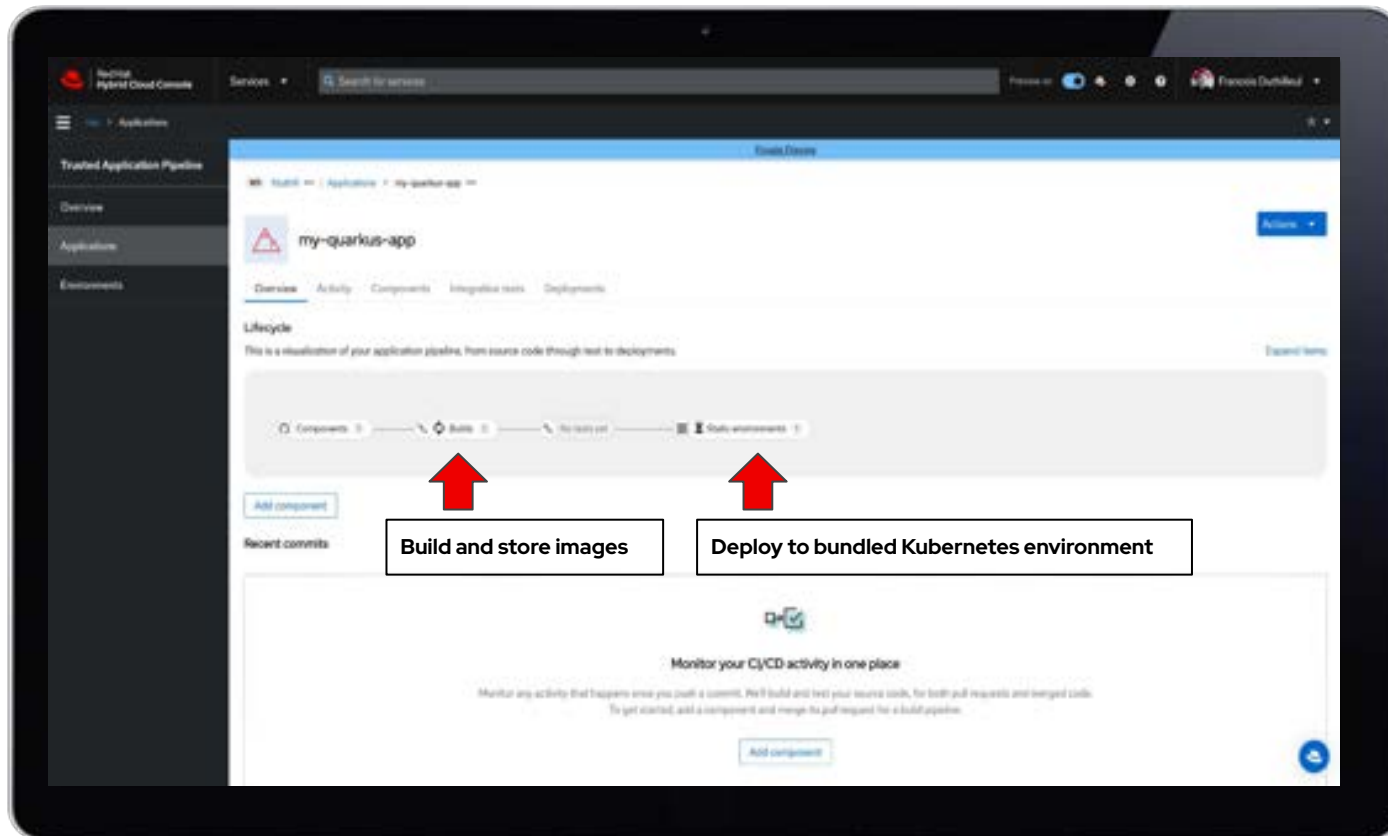
Meet industry compliance while increasing productivity, efficiency

- ▶ 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

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

Automatically run default CI/CD pipeline

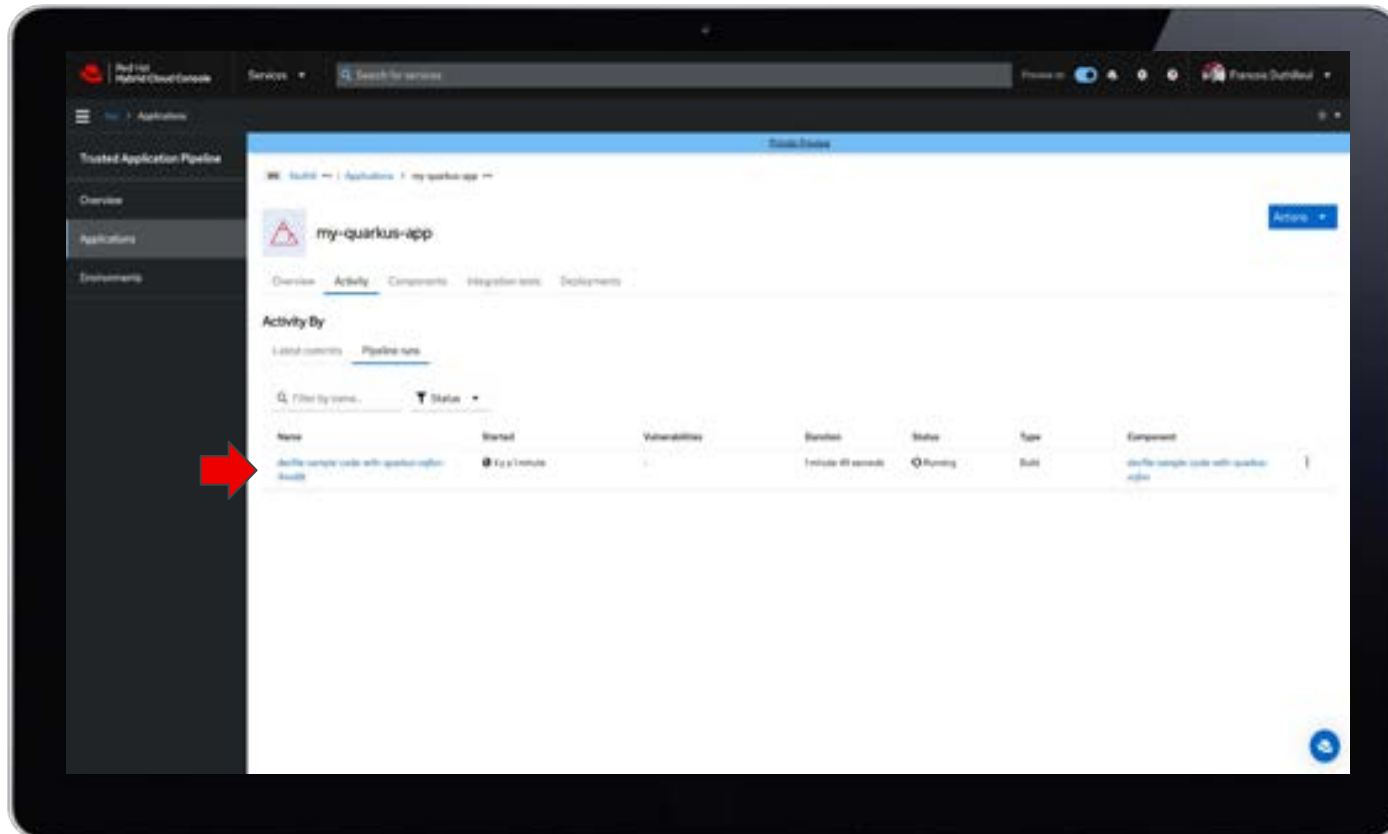


The screenshot displays the Red Hat Hybrid Cloud Console interface for a service named 'my-quarkus-app'. The left sidebar shows navigation options: 'Trusted Application Pipelines', 'Overview', 'Applications', and 'Environments'. The main content area is titled 'Create DevOps' and shows the application's lifecycle. A horizontal pipeline diagram consists of four stages: 'Components', 'Build', 'Test and lint', and 'Deploy to environments'. Two red arrows point from text boxes below to the 'Build' and 'Deploy to environments' stages. The text boxes contain the following descriptions:

- Build and store images** (pointing to the 'Build' stage)
- Deploy to bundled Kubernetes environment** (pointing to the 'Deploy to environments' stage)

Below the pipeline diagram, there is a section for 'Recent commits' and a 'Monitor your CI/CD activity in one place' section with a description: 'Monitor any activity that happens once you push a commit. We'll build and test your source code, for both pull requests and merged code. To get started, add a component and merge its pull request for a build pipeline.' An 'Add component' button is visible at the bottom of this section.

Drill down on pipeline details



View live pipeline runs in real-time

The screenshot displays the Red Hat OpenShift Console interface. The main content area shows the details of a pipeline run for the application 'my-quarkus-app'. The pipeline is titled 'devfile sample code with quarkus-nginx-04884' and is currently in a 'Running' state. The pipeline run details section shows a flowchart of the pipeline steps. A red arrow points to the 'build container' step, which is currently running. The 'build container' step is shown with a list of tasks: 'build', 'install-git', 'analyze-dependencies-pure-shell', 'merge-branches', 'test-shell-and-pull', and 'upload-shell'. The pipeline flow starts with 'test', followed by 'clone repository', 'inherited-dependencies', and 'build container'. From 'build container', the pipeline branches into several parallel tasks: 'inspect image', 'linter-check', 'optional-lint-check', 'cli-test', 'clean-run', 'show-pull-check', 'download-base-image-check', and 'in-pull-request'. These tasks converge into 'show success' and 'show done' steps.

Pipeline run details

build container in seconds

- build
- install-git
- analyze-dependencies-pure-shell
- merge-branches
- test-shell-and-pull
- upload-shell

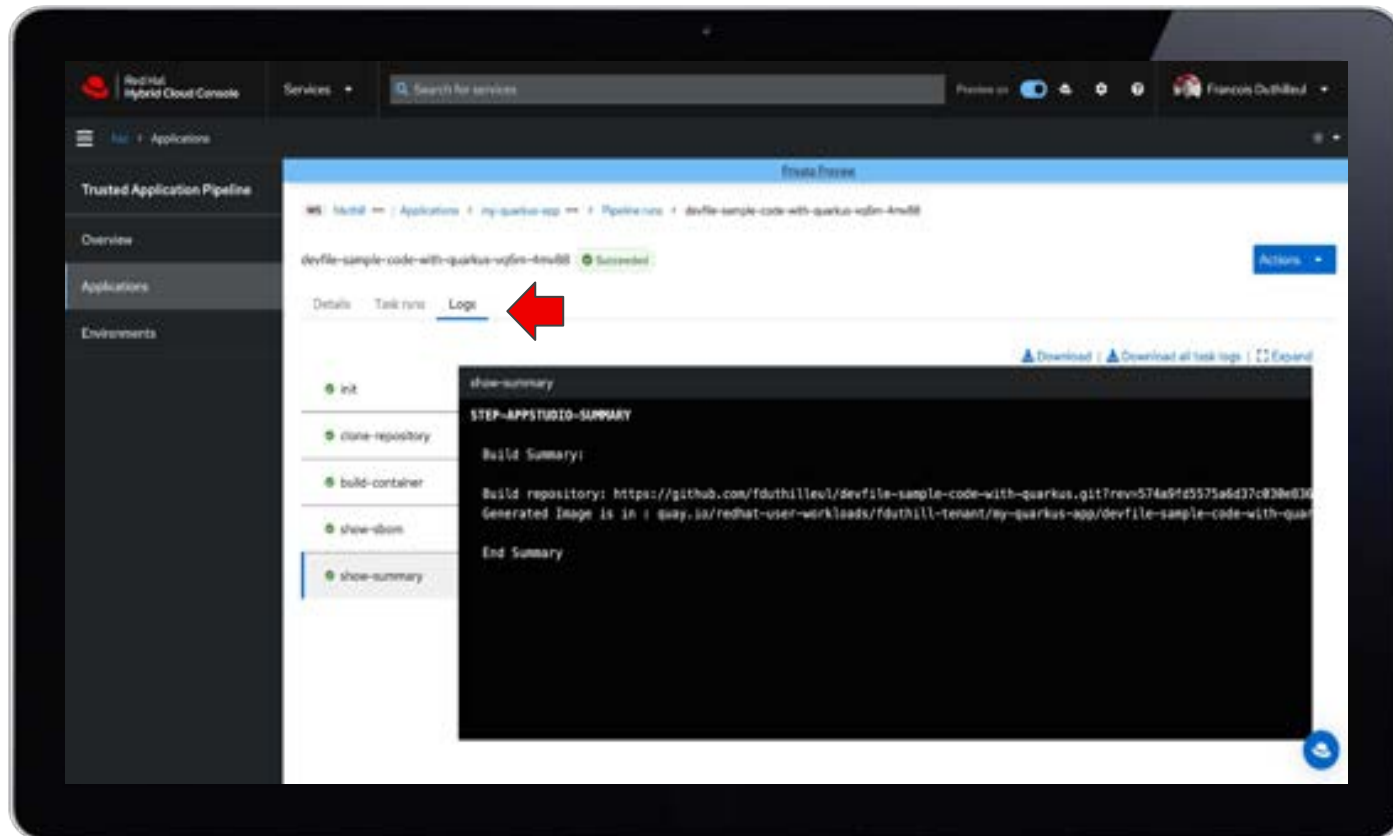
test → **clone repository** → **inherited-dependencies** → **build container**

inspect image → **linter-check** → **optional-lint-check** → **cli-test** → **clean-run** → **show-pull-check** → **download-base-image-check** → **in-pull-request**

show success → **show done**

Name: devfile sample code with quarkus-nginx-04884
Namespace: default-tenant
Labels: app=devfile-sample-code-with-quarkus-nginx, app=devfile-sample-code-with-quarkus-nginx
Status: Running
Pipeline: devfile-build
Application: my-quarkus-app

Access pipeline task log



Download and share SBOM for the build

The screenshot displays the Red Hat Pipeline Console interface for a build. The main area shows a flowchart of the pipeline stages, including 'clone repository', 'git fetch dependencies', 'build container', and several 'run' stages. The build status is 'Successful'. A red arrow points to the 'Download SBOM' link in the 'Build Summary' section. Another red arrow points to the 'View SBOM' link in the 'SBOM' section.

Name: javafile-sample-code-with-quarkus-native-knative

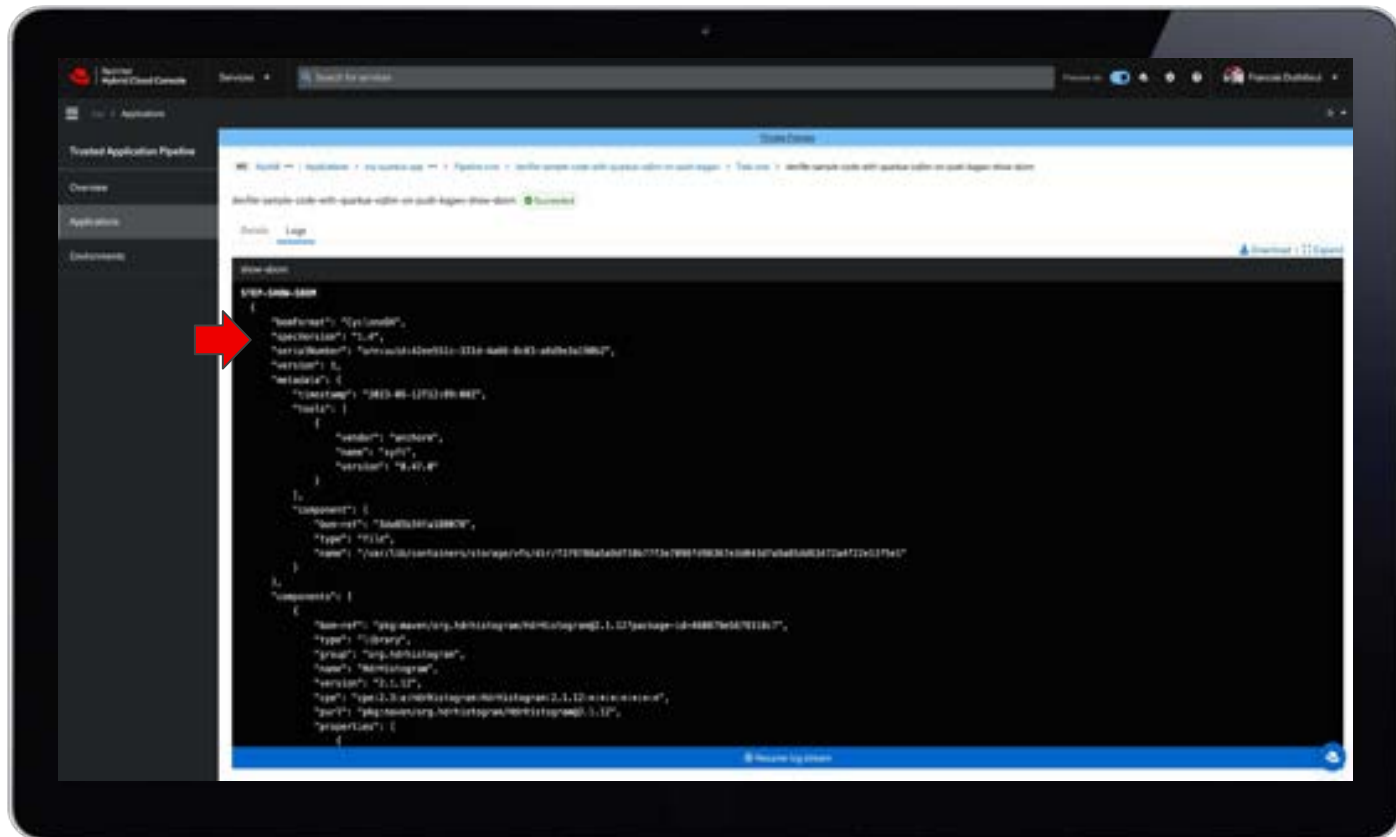
Status: Successful

Pipeline: Arkan-Build

Download SBOM: [Download SBOM](#)

SBOM: [View SBOM](#)

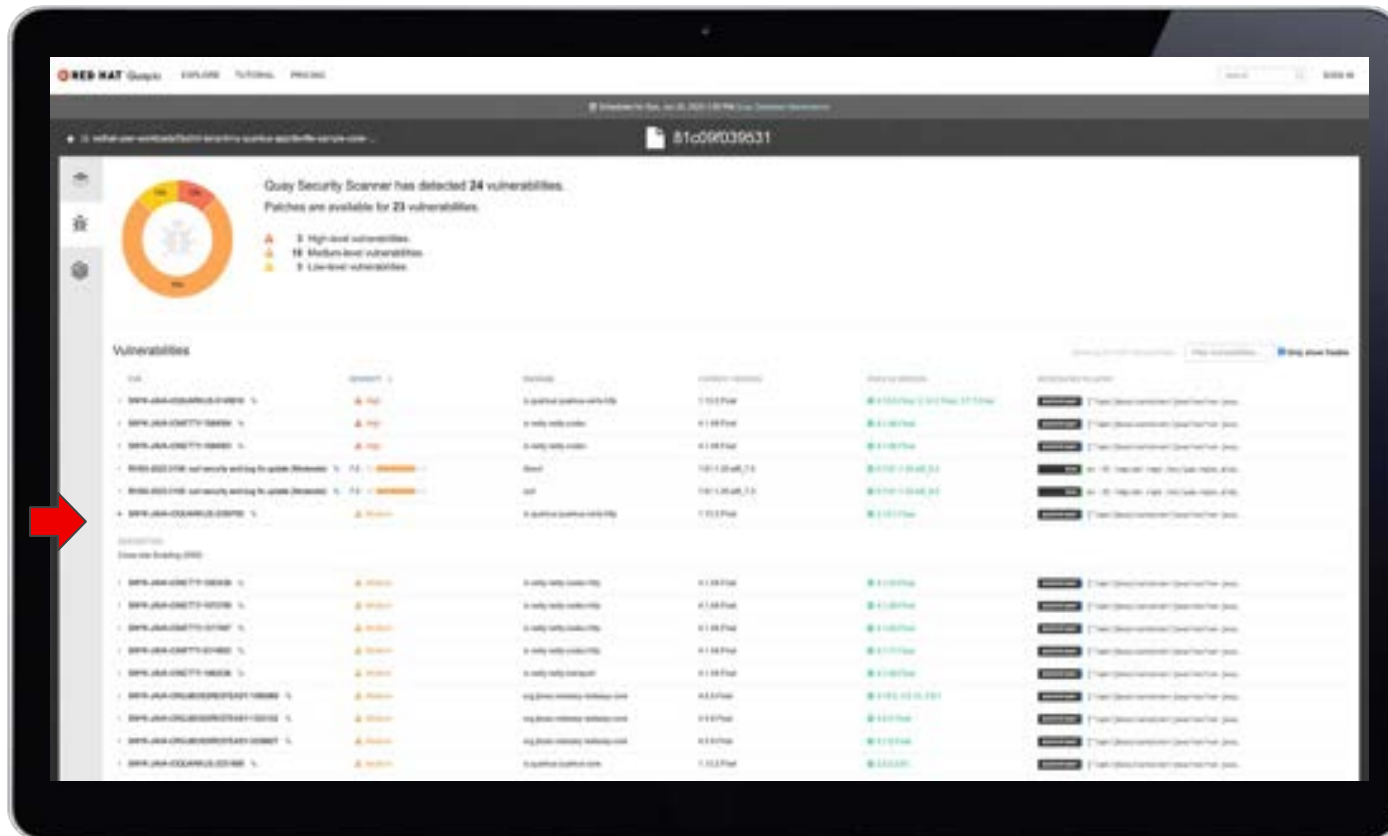
View the SBOM and verify software components



Viewing the artifacts and the attestation using cosign

```
francoisduthilleul@fduthill-mac ~ % cosign tree quay.io/redhat-user-workloads/fduthill-tenant/my-quarkus-app/
devfile-sample-code-with-quarkus-vq6m:1cf979badbcc16fd3c4d631a4b48cadf87cd06f0
📦 Supply Chain Security Related artifacts for an image: quay.io/redhat-user-workloads/fduthill-tenant/my-quarkus-app/devfile-sample-code-with-quarkus-vq6m:1cf979badbcc16fd3c4d631a4b48cadf87cd06f0
├── 📄 Attestations for an image tag: quay.io/redhat-user-workloads/fduthill-tenant/my-quarkus-app/devfile-sample-code-with-quarkus-vq6m:sha256-24bc730fb6f8663fc79c3c2e00a8fb8ce441e5ef629ae9c6f84d904acc4b58e9.att
│   ├── 🍷 sha256:035540dcef86e4d91c1aec4f3a2ec92ea02320491105394536448a31fdacfc5
│   ├── 🍷 sha256:0305c4208ec1460037a9daa8927dcbd5f5b0bc6ab45d945987d0646d3b68f262
│   └── 🍷 sha256:82c0524c5ab41712446923f1fa16adf32c53a417fc5aaeff1ce1a3d141a8cd3c
├── 📄 Signatures for an image tag: quay.io/redhat-user-workloads/fduthill-tenant/my-quarkus-app/devfile-sample-code-with-quarkus-vq6m:sha256-24bc730fb6f8663fc79c3c2e00a8fb8ce441e5ef629ae9c6f84d904acc4b58e9.sig
│   └── 🍷 sha256:c9a27f269be779f6c0635cbd3c89af8fb31d892bcb6f022a82f7e9f356629e6d
├── 📄 SBOMs for an image tag: quay.io/redhat-user-workloads/fduthill-tenant/my-quarkus-app/devfile-sample-code-with-quarkus-vq6m:sha256-24bc730fb6f8663fc79c3c2e00a8fb8ce441e5ef629ae9c6f84d904acc4b58e9.sbom
│   └── 🍷 sha256:c8c68fdd8ecf1815e5e60f6cc834cfa649c9111b6a73415272658da090f4991f
francoisduthilleul@fduthill-mac ~ % cosign download attestation quay.io/redhat-user-workloads/fduthill-tenant/my-quarkus-app/devfile-sample-code-with-quarkus-vq6m:1cf979badbcc16fd3c4d631a4b48cadf87cd06f0 | jq '.payload | @base64d|fromjson' > attestation.json
francoisduthilleul@fduthill-mac ~ %
```


Drill down on details of vulnerabilities detected

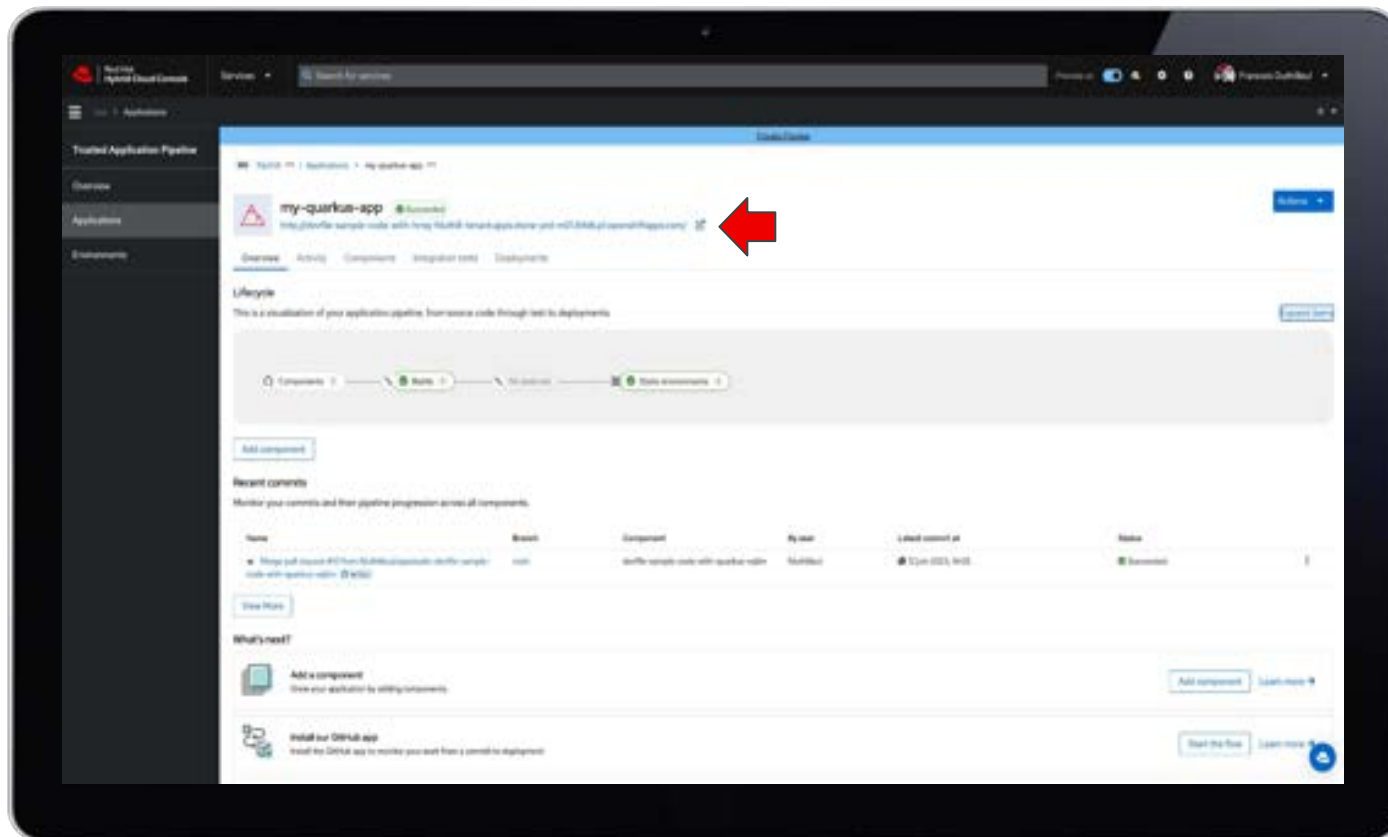


The screenshot displays the Red Hat Quay Security Scanner interface. At the top, it indicates that 24 vulnerabilities were detected, with patches available for 23 of them. A donut chart shows the distribution: 3 High-severity, 19 Medium-severity, and 2 Low-severity vulnerabilities.

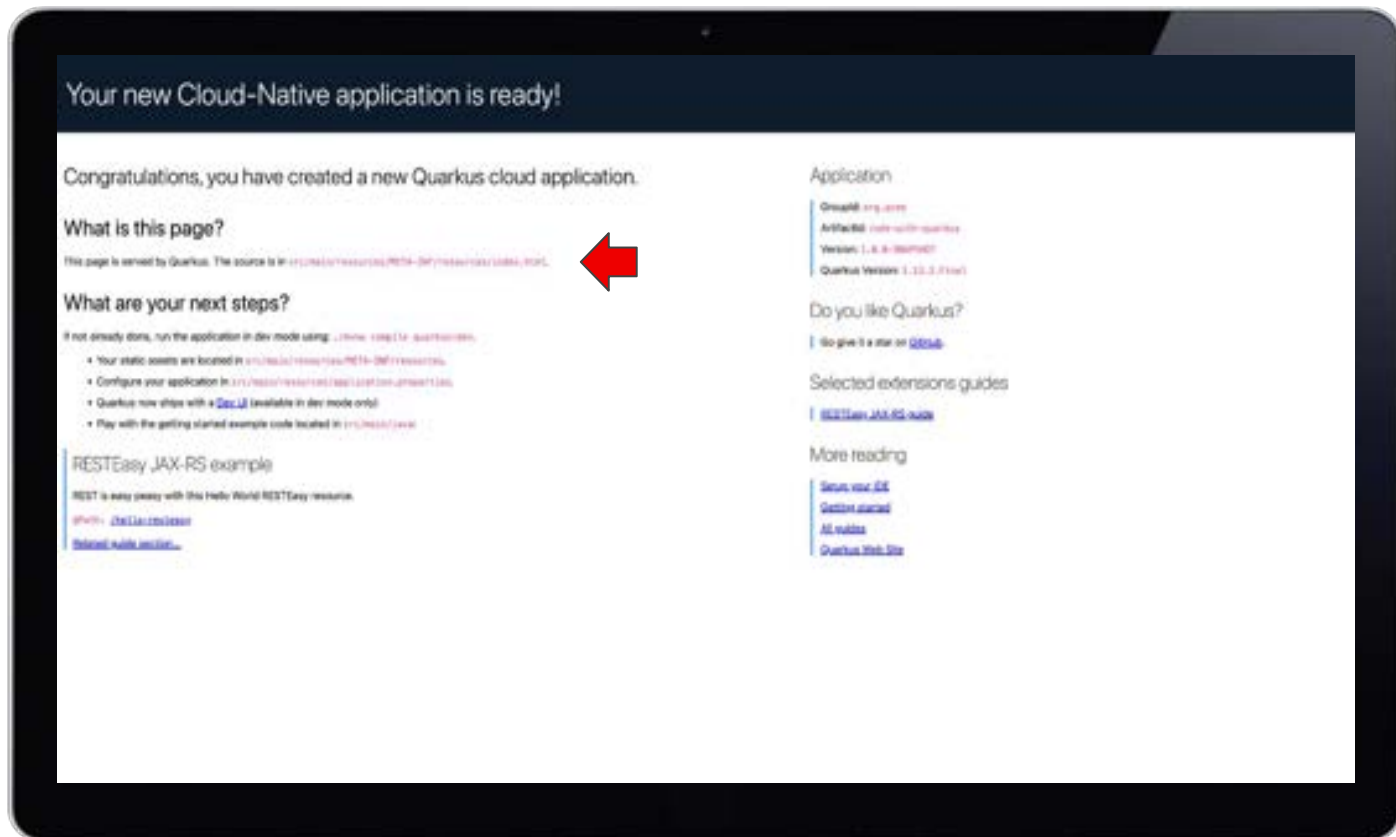
Vulnerabilities

ID	Severity	Package	Version Installed	Version Pinned	CVSS Score	Actions
1 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
2 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
3 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
4 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
5 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
6 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
7 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
8 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
9 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
10 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
11 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
12 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
13 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
14 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
15 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
16 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
17 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
18 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
19 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
20 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
21 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
22 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
23 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch
24 CVE-2019-0554 (CVE-2019-0554) %	High	golang.org/x/crypto	0.0.0-20190501000502-666e06488886	0.0.0-20190501000502-666e06488886	9.8	Details Patch

Access application deployed to a bundled Kubernetes environment



Share and collaborate on the application from a unique URL



Verifying artifacts

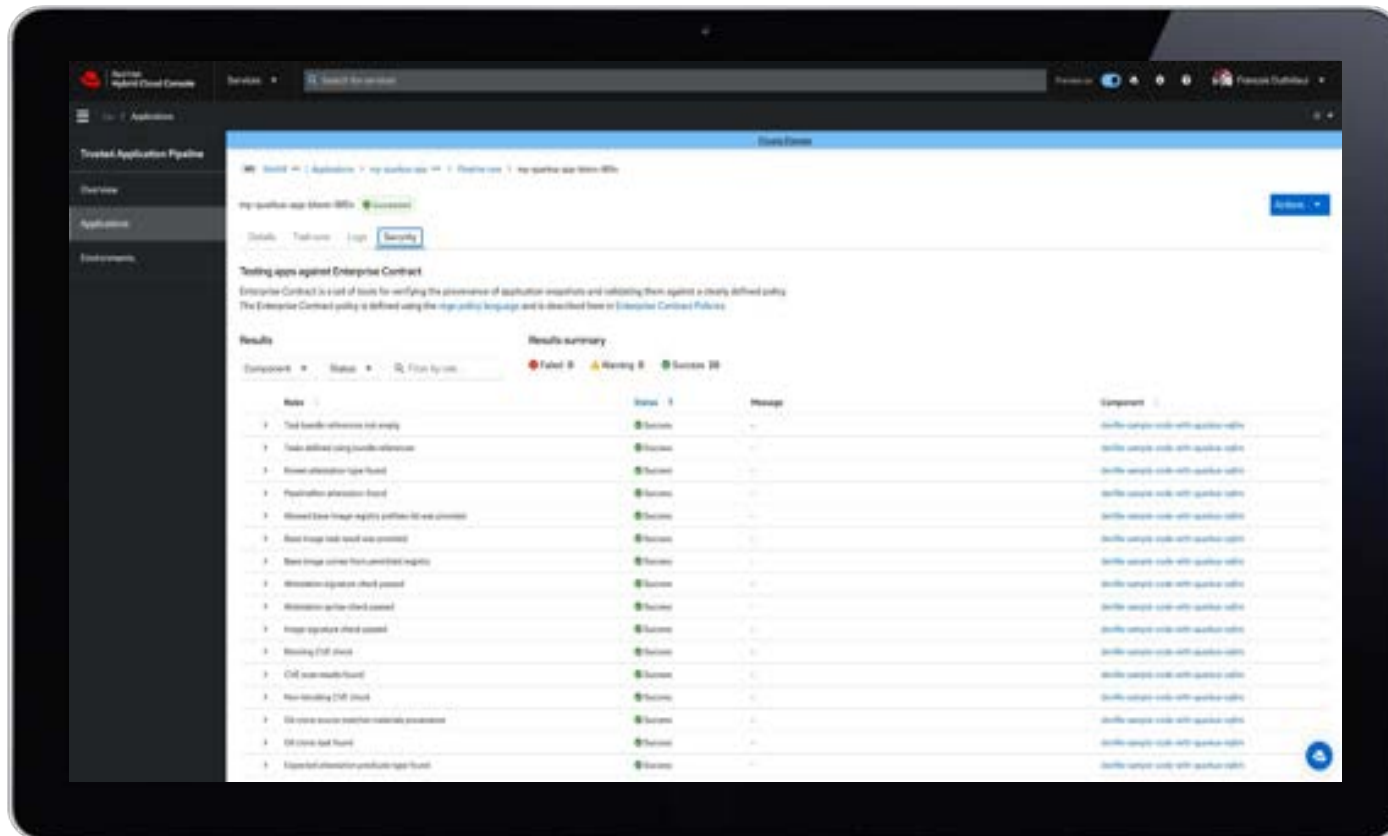


The **Enterprise Contract** is an artifact verifier and customizable policy checker designed to be easily integrated with CI/CD workflows.

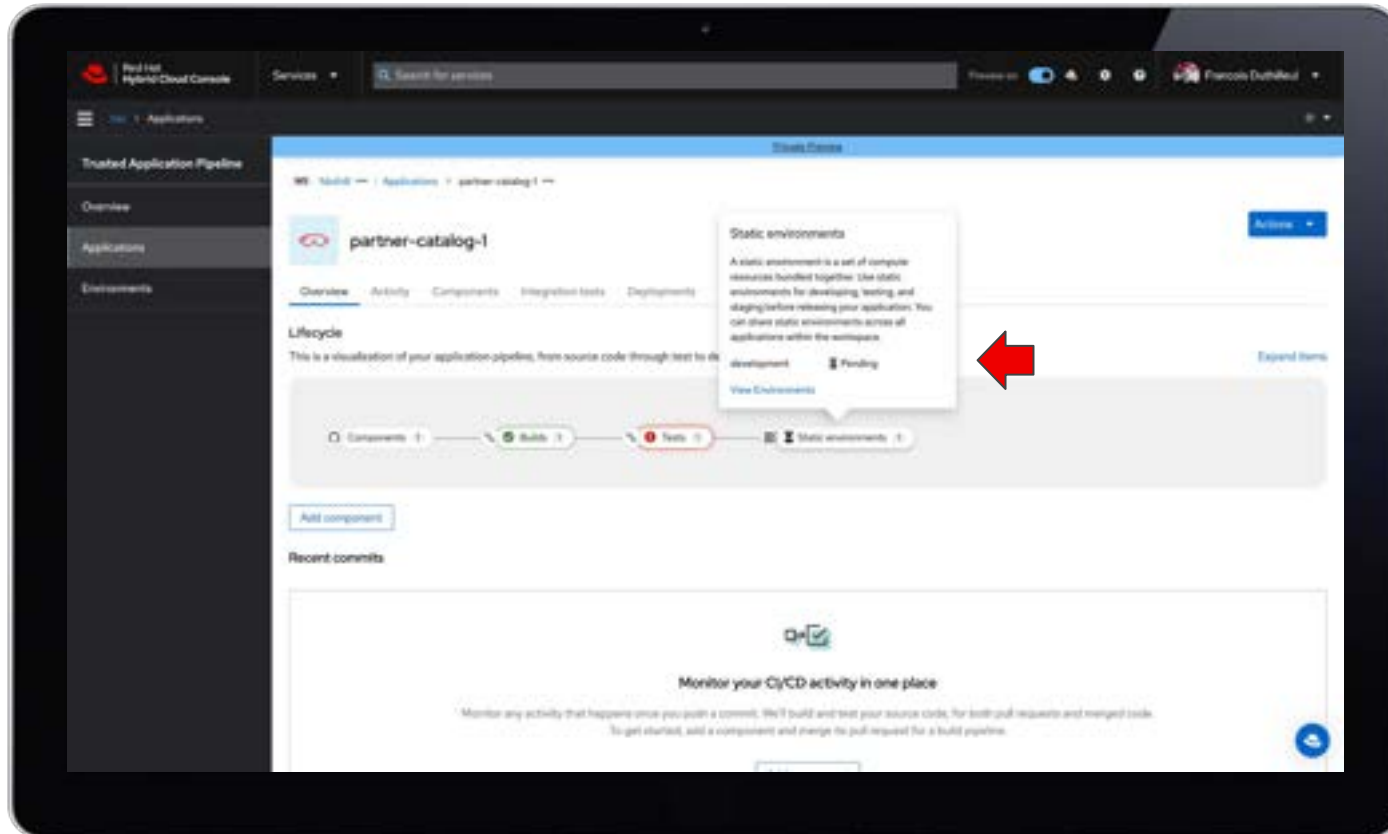
It was built to power the new Red Hat Trusted Application Pipeline and should be general enough to enhance supply chain security in other CI/CD systems.

Learn more at <https://enterprisecontract.dev/>

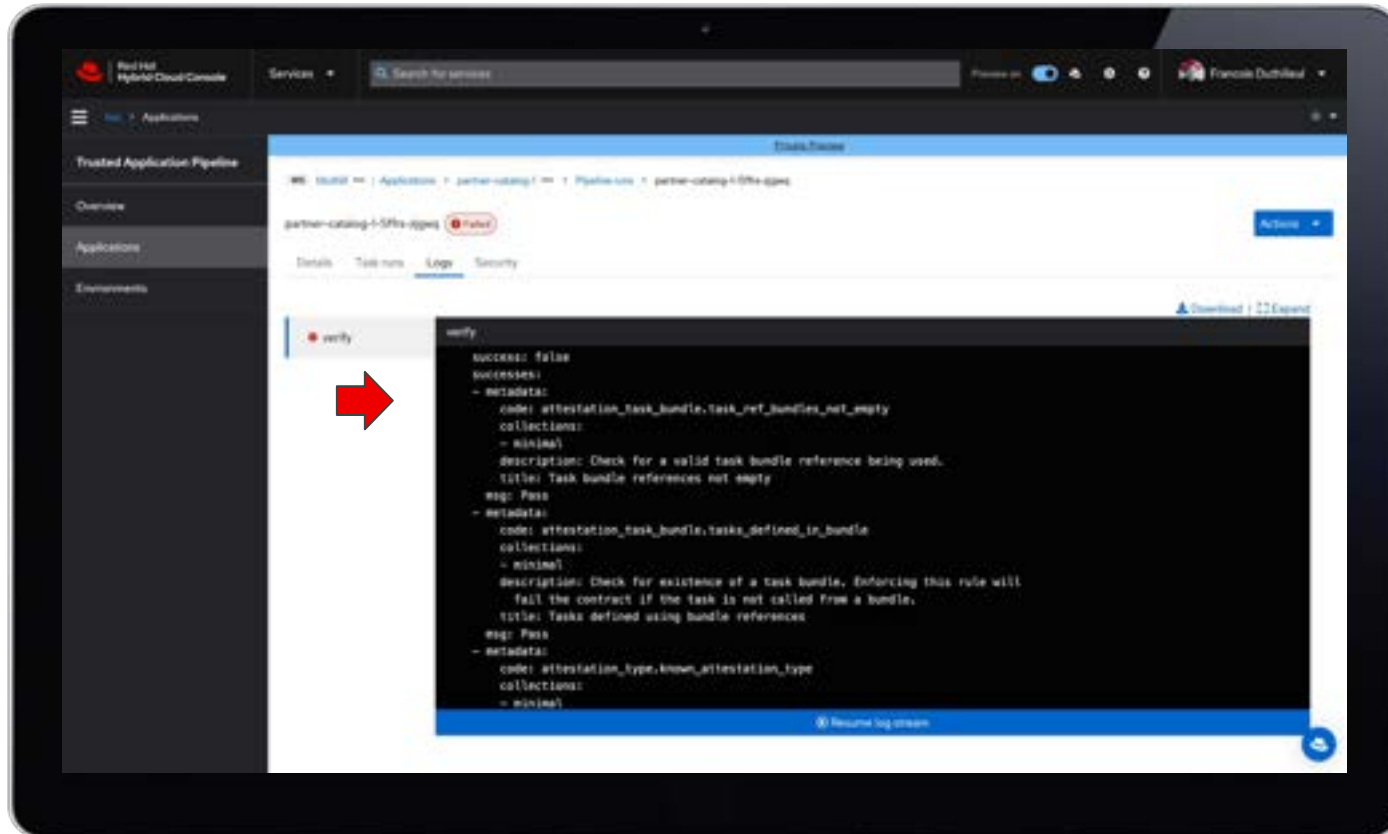
Set approval gates using enterprise contracts available out-of-the-box



Suspicious build activity is automatically blocked from production



Check details of flagged items in Enterprise Service Contract



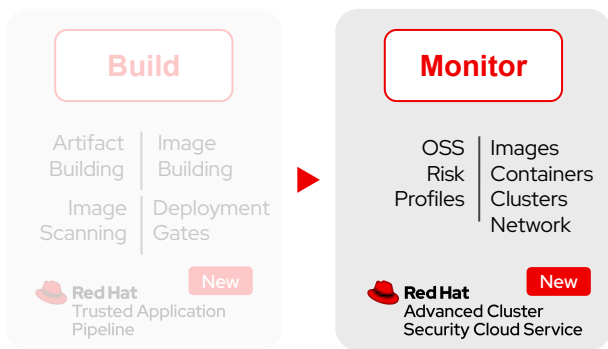
The screenshot displays the Red Hat Hybrid Cloud Console interface. The left sidebar shows navigation options: Applications, Trusted Application Pipeline, Overview, Applications, and Environments. The main content area shows a pipeline named 'partner-catalog-1-5f7e-036e'. A task named 'verify' is highlighted with a red arrow, indicating a failure. The task details are shown in a modal window, displaying the following JSON output:

```
verify
success: false
successes:
- metadata:
  code: attestation_task_bundle.task_ref_bundles_not_empty
  collections:
  - minimal
  description: Check for a valid task bundle reference being used.
  title: Task bundle references not empty
  msg: Pass
- metadata:
  code: attestation_task_bundle.tasks_defined_in_bundle
  collections:
  - minimal
  description: Check for existence of a task bundle, enforcing this rule will
    fail the contract if the task is not called from a bundle.
  title: Tasks defined using bundle references
  msg: Pass
- metadata:
  code: attestation_type.known_attestation_type
  collections:
  - minimal
```



*Continuously
monitor security
at runtime*

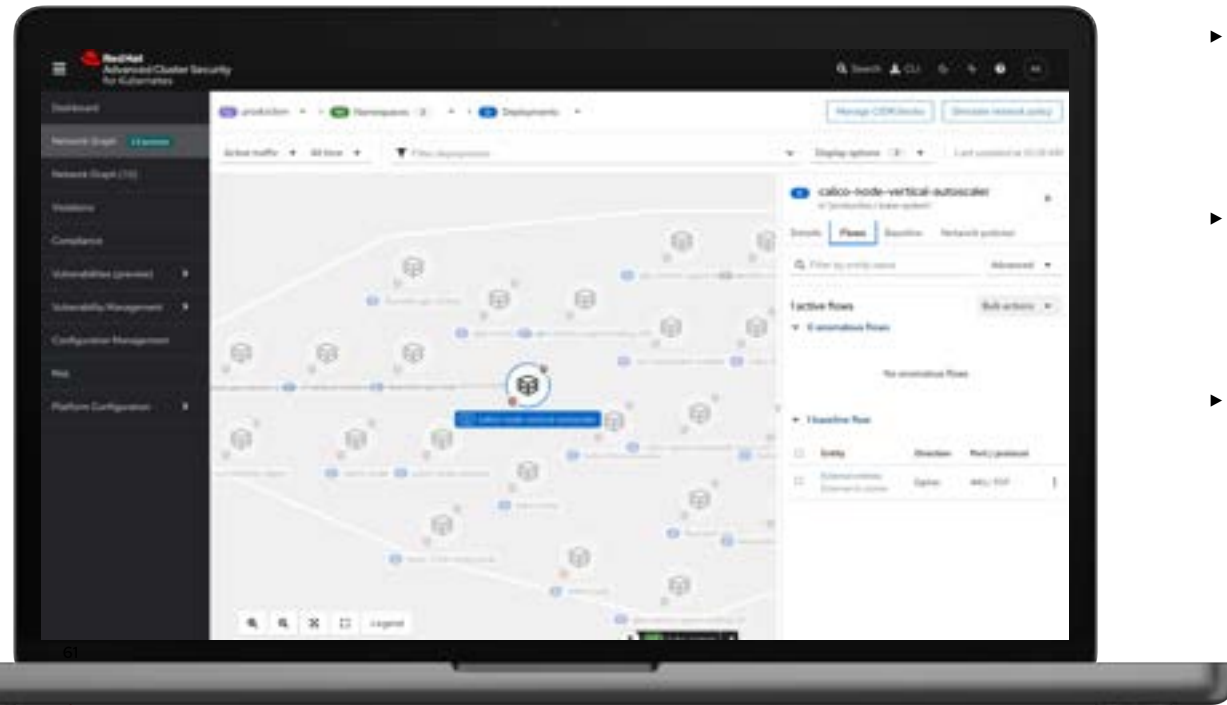
Monitor and identify runtime security incidents



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

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

Continuously monitor for anomalous behaviour at runtime

The screenshot displays the Red Hat Advanced Cluster Security for Kubernetes dashboard. The top navigation bar includes the Red Hat logo, the product name, a search bar, and user information (CLI). The main content area is titled 'Dashboard' and provides an overview of security metrics across all clusters and namespaces. A summary bar shows 1 Cluster, 7 Nodes, 164 Violations, 166 Deployments, 126 Images, and 779 Secrets, with a last update time of 03/02/2023 at 3:30 PM.

The '164 policy violations by severity' section features a bar chart with the following data:

Severity	Count
Low	94
Medium	63
High	6
Critical	1

The 'Most recent violations with critical severity' section lists a single violation: 'iptables Executed in Privileged Container' on the 'evrkube-node' node, dated 03/20/2023 at 3:50:00 PM.

The 'Images at most risk' section contains a table with the following data:

Images	Risk priority	Critical CVEs	Important CVEs
redhat-openshift/ocp-workload	1	2 fixable	6 fixable
observator-lum/token-refresher	2	1 fixable	0 fixable

A red arrow points to the 'Images' column header in the table.

Harden workloads by enforcing network policies

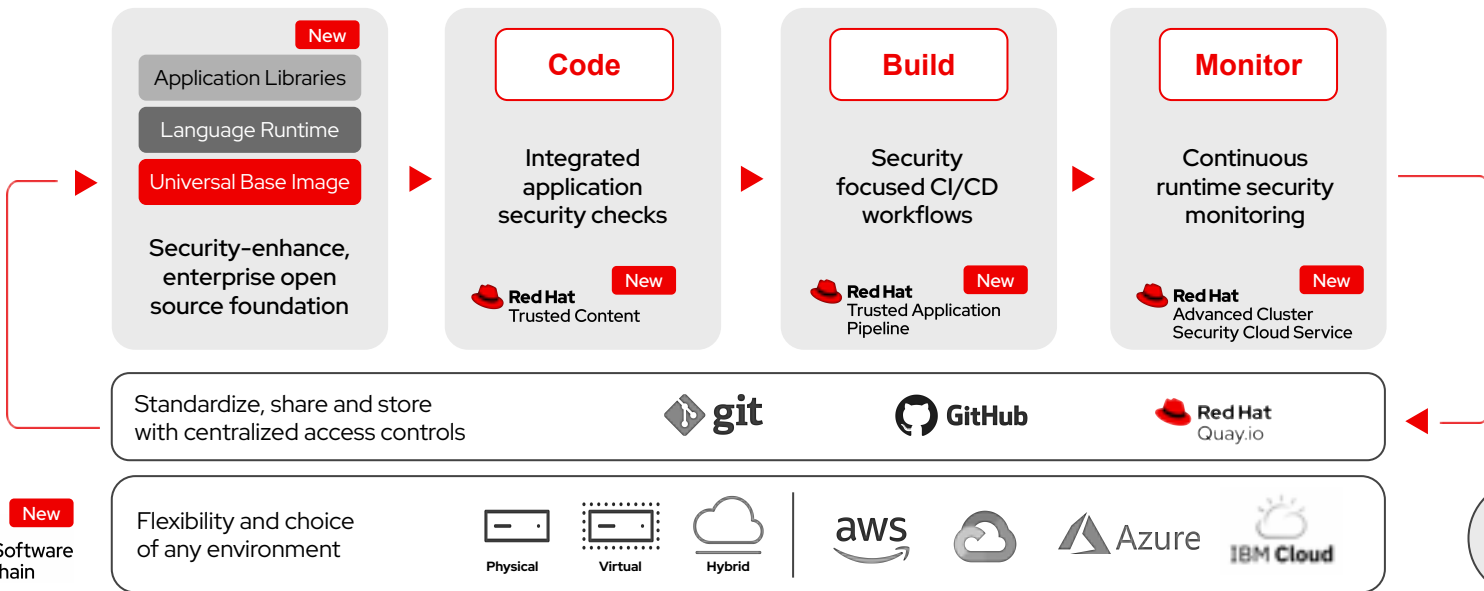
The screenshot displays the Red Hat Advanced Cluster Security for Kubernetes dashboard. The interface is divided into several sections:

- Left Sidebar:** A navigation menu with options: Dashboard, Network Graph (highlighted with a red arrow), Network Graph (2), Violations, Compliance, Vulnerabilities (preview), Vulnerability Management, Configuration Management, Risks, and Platform Configuration.
- Top Header:** Includes the Red Hat logo, "Advanced Cluster Security for Kubernetes", a search bar, and navigation icons.
- Breadcrumbs:** Shows the current path: production > namespaces (3) > Deployments.
- Main Content Area:** A network graph showing various pods and their connections. A red arrow points to the "Network Graph" menu item. A specific pod, "calico-node-vertical-autoscaler", is highlighted with a blue box.
- Right Panel:** Details for the selected pod, including tabs for "Details", "Flows", "Baseline", and "Network policies". It includes a search bar for "Filter by entity name" and a section for "Active flows" showing "0 anomalous flows" and "1 baseline flow".

Entity	Direction	Port/protocol
External entities External to cluster	Egress	443/TCP

Layered security throughout the stack and lifecycle

Achieve business agility while meeting security requirements



Thank you

Learn more

red.ht/trusted

 linkedin.com/company/red-hat

 youtube.com/user/RedHatVideos

 facebook.com/redhatinc

 twitter.com/RedHat



Some 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 Foundation. CycloneDX lead by OWASP.
SCA	Software Composition Analysis