컨테이너 및 쿠버네티스 보안에 대한 계층화된 접근 방식

빌드, 배포, 실행의 전 과정에서 컨테이너의 보안 유지

목차

소개

컨테이너는 애플리케이션과 그 종속 요소를 단일 이미지로 패키징하여 개발에서 테스트, 프로덕션에 이르는 전 과정에서 활용할 수 있다는 장점 때문에 광범위하게 사용되고 있습니다. 컨테이너를 사용하면 물리 서버, VM(가상 머신), 프라이빗 클라우드 또는 퍼블릭 클라우드와 같은 다양한 배포 대상과 환경에 걸쳐 일관성을 쉽게 유지할 수 있습니다. IT 팀은 컨테이너를 통해 비즈니스 민첩성을 갖춘 애플리케이션을 더 쉽게 개발하고 관리할 수 있습니다.

- **애플리케이션:** 개발자는 컨테이너를 사용하여 애플리케이션과 그 종속 요소를 더 쉽게 단일 유닛으로 구축하고 이관할 수 있습니다. 컨테이너는 배포하는 데 몇 초밖에 걸리지 않습니다. 컨테이너화된 환경에서 소프트웨어 빌드 프로세스는 전체 라이프 사이클 중 애플리케이션 코드가 필요한 런타임 라이브러리에 통합되는 단계입니다.
- **인프라:** 컨테이너는 공유 Linux® OS 커널의 샌드박스화된 애플리케이션 프로세스를 말합니다. 컨테이너는 가상 머신보다 더 컴팩트하고 가벼우며 덜 복잡합니다. 또한 온프레미스에서 퍼블릭 클라우드 플랫폼에 이르는 다양한 환경 간에 이식이 가능합니다.

쿠버네티스는 기업용 컨테이너 오케스트레이션 플랫폼입니다. 많은 조직에서 컨테이너를 사용하여 필수 서비스를 실행하는 오늘날, 컨테이너의 보안 유지는 무엇보다 중요한 문제입니다. 이 문서에서는 컨테이너화된 애플리케이션을 위한 핵심 보안 요소에 대해 설명합니다.

컨테이너와 쿠버네티스의 포괄적 보안: 계층 및 라이프사이클

컨테이너 보안은 실행 중인 Linux 프로세스의 보안과 매우 유사합니다. 컨테이너를 배포하고 실행하기에 앞서, 솔루션 스택의 계층 전반에 걸친 보안뿐만 아니라 애플리케이션 및 컨테이너 라이프 사이클 전반에 걸친 보안도 고려해야 합니다. 보안은 IT 라이프사이클 전체에 걸쳐 통합된 프로세스이자 새로 등장하는 위협과 솔루션에도 폭넓게 대응할 수 있는 지속적인 프로세스여야 한다는 점을 명심하십시오. 그림 1은 컨테이너 보안에 대한 포괄적인 접근 방식을 나타낸 것입니다.

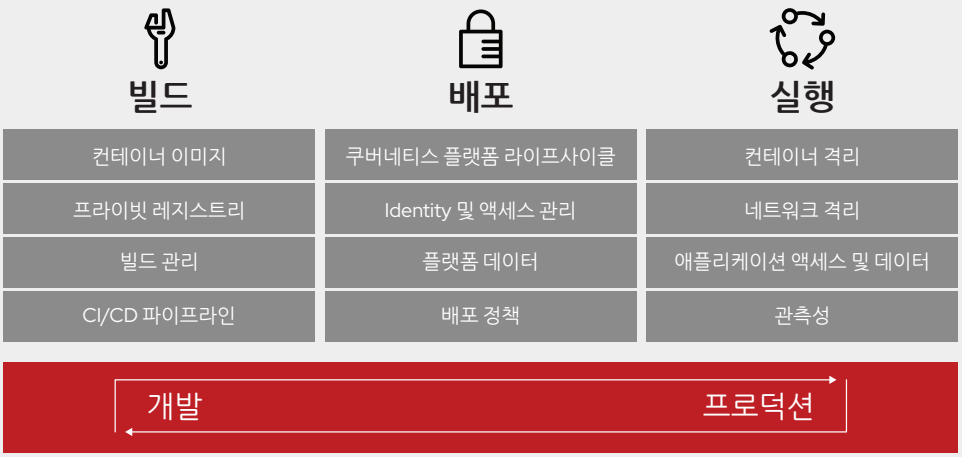

그림 1. 통합 컨테이너 보안에는 빌드, 배포, 런타임, 적절한 시점에 전문 툴을 이용해 보안을 확장하는 작업이 포함됩니다.

개발자는 컨테이너를 사용하여 애플리케이션과 그 종속 요소를 더 쉽게 단일 유닛으로 구축하고 이관할 수 있습니다. 또한 컨테이너를 사용하면 공유 호스트에서 멀티테넌트 애플리케이션을 배포할 수 있으므로 서버 활용률을 극대화하기도 용이합니다. 여러 애플리케이션을 단일 호스트에서 손쉽게 배포하고 필요에 따라 개별 컨테이너를 늘리거나 종료할 수 있습니다. 전통적인 가상화와는 달리 하이퍼바이저 없이 각 VM에서 게스트 운영 체제를 관리할 수 있습니다. 컨테이너는 하드웨어가 아니라 애플리케이션 프로세스를 가상화합니다.

물론 애플리케이션이 단일 컨테이너로 제공되는 사례는 거의 없습니다. 단순한 애플리케이션이라도 프런트엔드, 백엔드, 데이터베이스가 있는 것이 보통입니다. 그리고 마이크로서비스 기반의 현대적인 애플리케이션을 컨테이너로 배포하는 경우 여러 개의 컨테이너를 배포하게 되는데, 때로는 동일한 호스트에 배포하고 때로는 그림 2와 같이 여러 호스트 또는 노드에 분산하여 배포합니다.

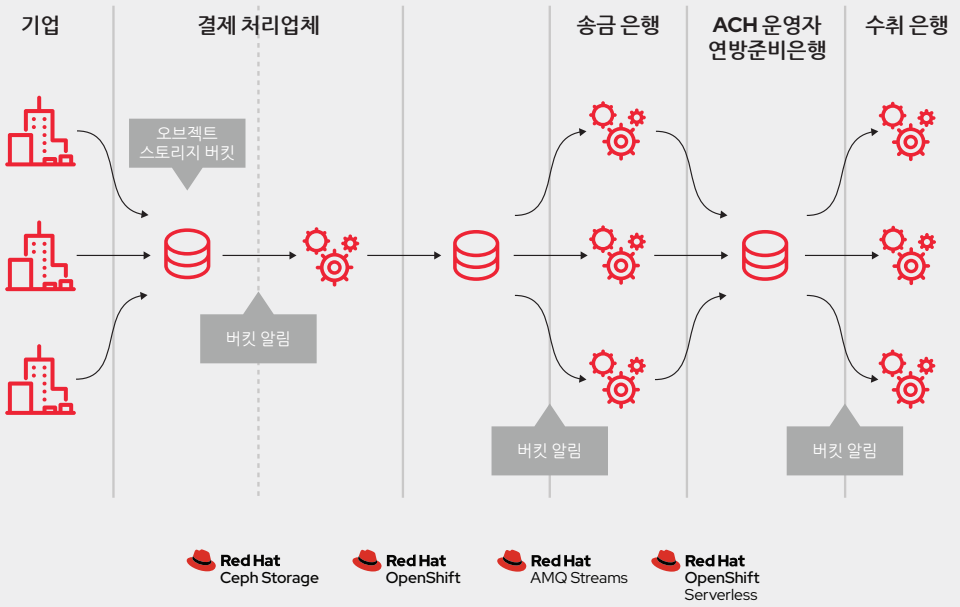

그림 2. 컨테이너 환경은 복잡하고 여러 노드에 분산될 수 있습니다.

대규모 컨테이너 배포를 관리할 때는 다음 사항을 고려해야 합니다.

- 어떤 컨테이너를 어떤 호스트에 배포해야 하는가?
- 어떤 호스트의 용량이 더 큰가?
- 서로 액세스해야 하는 컨테이너는 무엇이고 서로 어떻게 인식할 수 있는가?
- 네트워크 및 스토리지와 같은 공유 리소스에 대한 액세스 및 관리를 어떻게 제어할 것인가?
- 컨테이너 상태를 어떻게 모니터링할 것인가?
- 수요를 충족하기 위해 애플리케이션 용량을 어떻게 자동 확장할 것인가?
- 개발자 셀프 서비스를 지원하는 동시에 보안 요구 사항을 충족하려면 어떻게 해야 하는가?

자체 컨테이너 관리 환경을 구축할 수도 있지만, 그러려면 개별 구성 요소를 통합하고 관리하는 데 시간이 걸립니다. 아니면 관리 및 보안 기능이 내장된 컨테이너 플랫폼을 배포하는 방법도 있습니다. 이 접근 방식을 선택하면 인프라를 재구성하는 대신 비즈니스 가치가 있는 애플리케이션을 빌드하는 데 집중할 수 있습니다.

Red Hat® OpenShift® Container Platform은 컨테이너화된 애플리케이션을 빌드하고 규모를 조정할 수 있는 하이브리드 클라우드 방식의 일관된 엔터프라이즈급 쿠버네티스 플랫폼을 제공합니다. 쿠버네티스를 조직 전체에서 사용하려면 애플리케이션 자체에 보안을 구축하고 컨테이너 배포 보안을 관리하기 위한 정책을 자동으로 작성하는 기능과 컨테이너 런타임을 보호하는 기능 등 추가적인 보안 기능이 필요합니다.

애플리케이션 자체에 보안 구축

클라우드 네이티브 배포에서는 애플리케이션 자체에 보안을 구축하는 과정이 매우 중요합니다. 컨테이너화된 애플리케이션의 보안을 유지하려면 다음 사항을 준수해야 합니다.

1. 신뢰할 수 있는 컨테이너 콘텐츠 사용
2. 엔터프라이즈 컨테이너 레지스트리 사용
3. 컨테이너 빌드 제어 및 자동화
4. 애플리케이션 파이프라인에 보안 통합

1. 신뢰할 수 있는 컨테이너 콘텐츠 사용

보안 관리에서 중요한 것은 컨테이너에 포함된 요소입니다. 오늘날 애플리케이션과 인프라는 즉시 사용 가능한 요소로 구성되었습니다. 이들 중 대부분은 Linux 운영 체제, Apache Web Server, Red Hat JBoss® Enterprise Application Platform, PostgreSQL 및 Node.js와 같은 오픈소스 패키지입니다. 이러한 패키지의 컨테이너화된 버전도 제공되므로 자체적으로 빌드하지 않아도 됩니다. 그러나 외부 소스에서 다운로드하는 여느 코드와 마찬가지로 패키지의 출처가 어디인지, 누가 빌드했는지, 패키지 내에 악성 코드가 들어 있는지 식별해야 합니다. 다음과 같이 자문해 보십시오.

- 컨테이너의 콘텐츠가 인프라에 악영향을 미칠 수 있는가?
- 애플리케이션 계층에 알려진 취약점이 있는가?
- 컨테이너의 런타임 및 OS 계층이 최신 상태인가?
- 컨테이너가 얼마나 자주 업데이트되는가? 업데이트되는 시기는 어떻게 알 수 있는가?

Red Hat은 지난 수년간 Red Hat Enterprise Linux 및 당사 포트폴리오를 통해 신뢰할 수 있는 Linux 콘텐츠를 패키징하고 제공해 왔으며 최근에는 이렇게 신뢰할 수 있는 콘텐츠를 패키징하여 Linux 컨테이너로 제공하고 있습니다. Red Hat Universal Base Image를 도입하면 Open Container Initiative(OCI)를 준수하는 Linux 컨테이너를 어디에서 실행하든 Red Hat 컨테이너의 더욱 우수한 안정성, 보안 및 성능을 누릴 수 있습니다. 따라서 Red Hat Universal Base Image에 컨테이너화된 애플리케이션을 빌드한 다음 선택한 컨테이너 레지스트리로 푸시하고 공유할 수 있습니다.

Red Hat은 Red Hat Ecosystem Catalog를 통해 다양한 개발언어 런타임, 미들웨어, 데이터베이스를 위한 다수의 인증된 이미지와 오퍼레이터 등도 제공합니다. Red Hat 인증 컨테이너 및 오퍼레이터는 베어 메탈에서 VM, 클라우드에 이르기까지 Red Hat Enterprise Linux가 실행되는 모든 곳에서 실행되며 Red Hat과 파트너의 지원이 뒷받침됩니다.

Red Hat

Red Hat은 자체 제공하는 이미지의 상태를 지속적으로 모니터링합니다. 컨테이너 상태 지수(Container Health Index)를 통해 각 컨테이너 이미지의 '등급'이 공개되며, 이를 통해 프로덕션 시스템의 요구 사항을 충족하려면 컨테이너 이미지를 어떻게 큐레이션, 소비, 평가해야 하는지 알 수 있습니다. 컨테이너의 등급은 적용되지 않은 보안 정오표가 컨테이너를 구성하는 모든 요소에 미치는 영향과 그 기간을 고려해 매겨집니다. 이를 통해 보안 전문가는 물론 비전문가도 이해할 수 있는 종합적인 컨테이너 안전성 등급을 제공합니다.

Red Hat은 runc CVE-2019-5736, MDS CVE-2019-11091, VHOST-NET CVE-2019-14835에 대한 수정 등 보안 업데이트를 릴리스할 때 컨테이너 이미지를 다시 빌드하여 퍼블릭 레지스트리에 푸시합니다. Red Hat 보안 권고는 인증된 컨테이너 이미지에서 새롭게 발견된 문제를 알린 후 업데이트된 이미지로 안내하므로 사용자는 해당 이미지를 사용하는 모든 애플리케이션을 업데이트할 수 있습니다.

Red Hat에서 제공하지 않는 콘텐츠가 필요한 경우도 있습니다. 다른 소스로부터 획득한 컨테이너 이미지를 사용할 때 알려진 취약점에 대한 최신 정보를 파악하려면 취약성 데이터베이스가 지속적으로 업데이트되는 컨테이너 스캔 툴을 사용하는 것이 좋습니다. 알려진 취약점 목록은 계속해서 바뀌고 있기 때문에, Red Hat이 Red Hat 컨테이너 이미지를 관리하는 것과 같이 사용자 역시 자신들의 이미지를 최초 다운로드 시 해당 이미지내 콘텐츠에 대해 검사해야 하며 승인된 이미지와 배포된 이미지 모두에 대해 시간 경과에 따른 취약성 상태를 계속 추적해야 합니다.

2. 엔터프라이즈 컨테이너 레지스트리를 사용해 컨테이너 이미지에 대한 액세스 보안 강화

물론 IT 팀에서는 다운로드한 퍼블릭 컨테이너 이미지 위에 콘텐츠를 배치하여 컨테이너를 빌드합니다. 따라서 다운로드한 컨테이너 이미지와 내부에서 빌드한 이미지에 대한 액세스 및 이관을 다른 유형의 바이너리와 동일한 방식으로 관리해야 합니다. 다수의 프라이빗 레지스트리가 컨테이너 이미지의 스토리지를 지원하며 레지스트리에 저장되어 있는 컨테이너 이미지에 대해 자동화된 정책을 지원하는 프라이빗 레지스트리를 선택하는 것을 권장합니다.

Red Hat OpenShift에는 컨테이너 이미지를 관리할 수 있는 기본적인 기능을 제공하는 프라이빗 레지스트리가 포함되어 있습니다. Red Hat OpenShift 레지스트리는 역할 기반 액세스 제어(RBAC)를 제공하므로 특정 컨테이너 이미지의 풀 앤 푸시(Pull and push)를 수행할 수 있는 권한을 누구에게 부여할지 정할 수 있습니다. Red Hat OpenShift는 JFrog의 Artifactory, Sonatype Nexus와 같이 기업이 현재 사용하고 있는 다른 프라이빗 레지스트리와의 통합도 지원합니다.

Red Hat Quay는 독립형 엔터프라이즈 레지스트리로 제공됩니다. Red Hat Quay는 지리적 복제 및 빌드 이미지 트리거와 같은 여러 부가적인 엔터프라이즈 기능을 제공합니다.

Clair 프로젝트는 Red Hat Quay 보안 스캐너를 구동하여 Red Hat Quay 내 모든 이미지의 취약점을 탐지하는 오픈소스 엔진입니다. Red Hat OpenShift Container Security Operator는 Red Hat Quay와 통합되어 OpenShift 콘솔에서 배포한 이미지의 알려진 취약점을 클러스터 전체에 걸쳐 볼 수 있는 기능을 제공합니다.

3. 컨테이너 이미지 빌드 제어 및 자동화

이 빌드 프로세스를 관리하는 것이 소프트웨어 스택 보안의 핵심입니다. "한 번 빌드하여 모든 곳에 배포"한다는 철학을 바탕으로 빌드 프로세스에서 생성된 제품이 프로덕션 환경에 배포된 것과 정확히 일치하도록 할 수 있습니다. 컨테이너의 불변성을 유지하는 것도 중요합니다. 바꿔 말하자면 실행 중인 컨테이너를 패치하는 대신에 다시 빌드하여 재배포해야 합니다.

Red Hat OpenShift는 사용자 정의 이미지의 보안을 강화하기 위한 한 가지 방법으로 외부 이벤트에 근거하여 빌드를 자동화하는 기능을 다수 제공합니다.

- Red Hat Quay 트리거는 GitHub 푸시, BitBucket 푸시, GitLab 푸시, 웹후크와 같은 외부 이벤트에서 Dockerfile의 리포지토리 빌드를 생성할 수 있는 메커니즘을 제공합니다.
- S2I(Source-to-Image)는 소스 코드와 기본 이미지를 결합하기 위한 오픈소스 프레임워크입니다(그림 3). 개발팀과 운영팀은 재사용 가능한 빌드 환경에서 S2I를 사용하여 쉽게 협업할 수 있습니다. 개발자가 S2I에 따라 git에 코드를 커밋하면 Red Hat OpenShift는 다음을 수행할 수 있습니다.
  - 코드 리포지토리의 웹후크나 그 외 자동화된 CI(지속적인 통합) 프로세스를 통해 S2I 기본 이미지와 새롭게 커밋된 코드 등 제공되는 아티팩트에서 새로운 이미지 자동 구성을 트리거
  - 테스트를 위해 새로 빌드된 이미지를 자동으로 배포
  - 테스트된 이미지를 프로덕션 상태로 이관하고 지속적인 통합 및 배포(CI/CD) 프로세스를 통해 새 이미지를 자동으로 배포합니다.

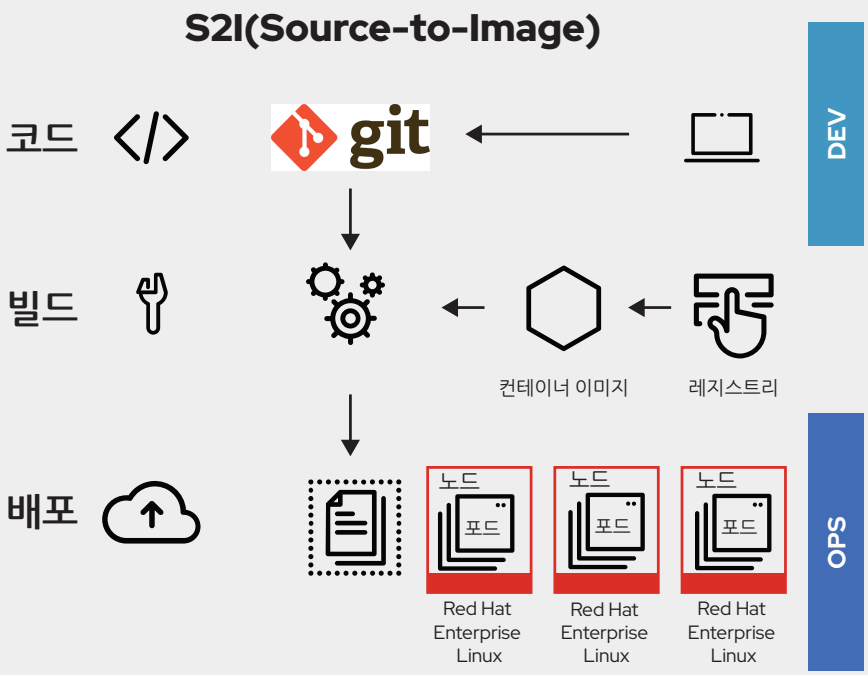

그림 3. S2I는 소스 코드에서 재현 및 즉시 실행 가능한 컨테이너 이미지를 빌드하기 위한 툴킷이자 워크플로우입니다.

- Red Hat OpenShift 이미지 스트림을 사용해 클러스터에 배포된 외부 이미지의 변경 사항을 감시할 수 있습니다. 이미지 스트림은 빌드나 배포, 작업, 복제 컨트롤러, 복제본 세트와 같이 Red Hat OpenShift에서 제공되는 모든 네이티브 리소스와 함께 사용할 수 있습니다. 이미지 스트림을 감시함으로써 빌드와 배포 시 새 이미지가 추가되거나 수정될 때 알림을 받고 빌드나 배포를 각기 자동으로 시작하는 방식으로 대응할 수 있습니다.

  예를 들어 기본, 미들웨어, 애플리케이션이라는 세 가지 컨테이너 이미지 계층을 사용해 빌드된 애플리케이션이 있다고 가정해 보겠습니다. 기본 이미지에서 문제가 발견되어 이 이미지를 Red Hat이 다시 빌드하여 Red Hat Ecosystem Catalog로 푸시합니다. 이미지 스트림이 활성화되어 있으면 Red Hat OpenShift는 이미지가 변경되었을 때 이를 감지할 수 있습니다. 이 이미지에 의존하며 트리거가 정의된 빌드의 경우 Red Hat OpenShift는 애플리케이션 이미지를 자동으로 다시 빌드하여 고정된 기본 이미지를 통합합니다.

  빌드가 완료되면 업데이트된 사용자 정의 이미지가 Red Hat OpenShift의 내부 레지스트리로 푸시됩니다. Red Hat OpenShift는 내부 레지스트리의 이미지 변경을 즉시 감지하며 트리거가 정의된 애플리케이션에 대해 업데이트된 이미지를 자동으로 배포합니다. 이를 통해 프로덕션에서 실행되는 코드가 가장 최근에 업데이트된 이미지와 항상 동일하게 됩니다. 이 모든 기능이 함께 작동하여 보안 기능을 CI/CD 프로세스 및 파이프라인에 통합합니다.

4. 애플리케이션 파이프라인에 보안 통합

Red Hat OpenShift에는 컨테이너(서버리스 포함)를 위해 작동하는 차세대 쿠버네티스 CI/CD 파이프라인인 CI용 Jenkins와 Tekton의 통합 인스턴스가 포함되어 있습니다. Red Hat OpenShift에는 자체 빌드 또는 프라이빗 이미지 레지스트리 등의 CI/CD 툴을 통합하는 데 사용할 수 있는 다양한 RESTful API도 포함되어 있습니다.

애플리케이션 보안의 모범 사례는 레지스트리, 통합 개발 환경(IDE), CI/CD 툴 등의 파이프라인에 자동화된 보안 테스트를 통합하는 것입니다.

레지스트리: 컨테이너 이미지는 프라이빗 컨테이너 레지스트리에서 스캐닝할 수 있고 스캐닝되어야 합니다. 취약점이 발견되었을 때 Red Hat Quay를 Clair 보안 스캐너와 함께 사용해 개발자에게 알려줄 수 있습니다. OpenShift Container Security Operator는 Red Hat Quay와 통합되어 OpenShift 콘솔에서 배포한 이미지의 알려진 취약점을 클러스터 전체에 걸쳐 볼 수 있는 기능을 제공합니다. 또는 Red Hat Ecosystem Catalog에서 여러 개의 타사 인증 컨테이너 스캐닝 솔루션을 찾아볼 수 있습니다.

IDE: Red Hat Dependency Analytics 통합 개발 환경(IDE) 플러그인은 코드를 IDE에 처음 도입할 때 취약성 경고와 함께 프로젝트 종속성에 대한 문제 해결 방법을 조언합니다.

CI/CD: 스캐너를 CI와 통합하여 알려진 취약점을 실시간으로 검사할 수 있습니다. 스캐너는 컨테이너의 오픈소스 패키지 목록을 작성하고, 이미 알려진 취약점을 알려주고, 앞서 스캐닝된 패키지에서 새로운 취약점이 발견되면 이를 알려줍니다.

또한 보안 스캔을 통해 문제가 있다고 감지된 빌드에 플래그를 지정하는 정책을 CI 프로세스에 포함해야 합니다. 이렇게 해야 IT 팀이 적절한 조치로 문제를 최대한 빨리 해결할 수 있습니다.

끝으로 사용자 정의 방식으로 빌드된 컨테이너에 서명하여 빌드 후 배포하는 과정에서 컨테이너가 조작되지 않게 하는 것이 좋습니다.

배포: 배포의 구성, 보안 및 컴플라이언스 관리

배포의 보안을 효과적으로 유지하는 방법에는 배포 정책 자동화뿐 아니라 쿠버네티스 플랫폼도 포함됩니다. Red Hat OpenShift에는 다음과 같은 뛰어난 기능이 포함되어 있습니다.

1. 플랫폼 구성 및 라이프사이클 관리
2. Identity 및 액세스 관리
3. 플랫폼 데이터 및 연결된 스토리지 보안
4. 배포 정책

5. 플랫폼 구성 및 라이프사이클 관리

2019년 여름에 발표된 Cloud Native Computing Foundation(CNCF) 쿠버네티스 보안 감사에서는 쿠버네티스에 대한 가장 큰 보안 위협은 쿠버네티스 구성 요소를 구성하고 강화하기가 복잡하다는 점이라는 결론을 내렸습니다. Red Hat OpenShift는 쿠버네티스 오퍼레이터를 사용해 이 문제를 해결합니다.

오퍼레이터는 쿠버네티스 네이티브 애플리케이션을 패키징, 배포, 관리하는 방법입니다. 오퍼레이터는 애플리케이션 관리에 필요한 애플리케이션별 로직으로 쿠버네티스 애플리케이션 프로그래밍 인터페이스(API)를 확장할 수 있는 사용자 정의 컨트롤러의 역할을 합니다. 각 Red Hat OpenShift 플랫폼 오퍼레이터에 래핑되어 있으며 OpenShift에 대한 자동 구성, 모니터링, 관리 기능을 제공합니다. 개별 오퍼레이터는 API 서버, 소프트웨어 정의 네트워크(SDN)와 같은 구성 요소를 직접 구성하는 반면, 클러스터 버전 오퍼레이터는 플랫폼 전반에 걸쳐 여러 개의 오퍼레이터를 관리합니다. 오퍼레이터를 통해 커널에서 스택의 상위 서비스에 이르기까지 업데이트를 비롯한 클러스터 관리를 자동화할 수 있습니다.

컨테이너 플랫폼이 제공하는 핵심 가치 중 하나는 개발자 셀프 서비스를 지원한다는 것입니다. 이를 통해 개발팀은 승인된 계층에 빌드된 애플리케이션을 더 쉽고 빠르게 제공할 수 있습니다. 셀프 서비스 포털은 팀이 협업을 증진함과 동시에 보안을 유지하는 데 충분한 통제 권한을 부여합니다. Operator Lifecycle Manager(OLM)는 Red Hat OpenShift 클러스터 사용자가 애플리케이션 활성화에 필요한 서비스를 배포할 수 있는 오퍼레이터를 찾아 사용하도록 지원하는 프레임워크를 제공합니다. 사용자는 OLM을 이용해 역할 기반 액세스 제어를 설치 및 업그레이드하고 가용 오퍼레이터에 할당할 수 있습니다.

컴플라이언스를 지원하기 위해 Red Hat OpenShift는 컴플라이언스 프레임워크가 요구하는 기술적 제어 기능으로 플랫폼의 컴플라이언스를 자동화하는 컴플라이언스 오퍼레이터를 제공합니다. Red Hat OpenShift 관리자는 컴플라이언스 오퍼레이터를 통해 원하는 클러스터 컴플라이언스 상태를 설명하고, 격차를 대략 확인하고 이를 해결할 수도 있습니다. 컴플라이언스 오퍼레이터는 클러스터를 실행 중인 노드를 포함한 모든 플랫폼의 컴플라이언스를 평가합니다. 클러스터 노드에서 정기적으로 파일 무결성 점검을 실행할 수 있도록 파일 무결성 오퍼레이터도 제공합니다.

6. Identity 및 액세스 관리

개발자와 관리자가 공동으로 사용할 수 있는 기능이 쿠버네티스에 풍부하게 갖춰져 있지만 강력한 Identity 관리 및 RBAC는 컨테이너 플랫폼을 이루는 매우 중요한 요소입니다. 쿠버네티스 API는 대규모 컨테이너 관리를 자동화하는 핵심 요소입니다. 예를 들어 API는 포드 및 서비스 구성 및 배포 등의 요청을 시작하고 검증하는 데 사용됩니다.

API 인증 및 허가는 컨테이너 플랫폼 보안에 매우 중요합니다. API 서버는 액세스의 중심 지점으로 가장 높은 수준의 보안 감사를 받아야 합니다. Red Hat OpenShift 컨트롤 플레인에는 클러스터 인증 오퍼레이터를 통한 기본 인증이 포함되어 있습니다. 개발자, 관리자, 서비스 계정은 자신을 API에 인증할 수 있는 OAuth 액세스

토큰을 받습니다. 관리자는 사용자가 토큰을 받기 전에 인증할 수 있도록 선택한 Identity 공급자를 클러스터에 맞게 구성할 수 있습니다. LDAP(Lightweight Directory Access Protocol) 디렉터리를 포함해 9개의 Identity 공급자가 지원됩니다.

정교한 RBAC는 Red Hat OpenShift에서 기본적으로 지원됩니다. RBAC 객체에 따라 사용자가 클러스터 내에서 특정 작업을 수행할 수 있는지 여부가 결정됩니다. 클러스터 관리자는 클러스터 역할 및 바인딩을 사용해 OpenShift 클러스터와 클러스터 내 프로젝트에 맞게 액세스 수준을 제어할 수 있습니다.

7. 플랫폼 데이터 보안

Red Hat OpenShift는 전송 중인 데이터의 보안을 위해 기본적으로 쿠버네티스를 강화하며 유휴 데이터 보안을 위한 옵션도 있습니다.

Red Hat OpenShift는 다음과 같은 방법으로 전송 중인 플랫폼 데이터를 보호합니다.

- 상호 통신하는 모든 컨테이너 플랫폼 구성 요소에 대해 https를 경유해 전송 중인 데이터를 암호화
- 컨트롤 플레인과의 모든 통신 내용을 TLS(Transport Layer Security)를 통해 전송
- API 서버에 대한 액세스가 X.509 인증서 또는 토큰을 기반으로 이루어지도록 보장
- 프로젝트 할당량을 사용해 악성 토큰으로 인한 피해의 범위 제한
- etcd를 자체 인증 기관(CA) 및 인증서로 구성. (쿠버네티스에서 etcd는 영구 마스터 상태를 저장하고, 다른 구성 요소는 etcd의 변경 사항을 감시하여 자신을 지정된 상태로 전환합니다.)
- 플랫폼 인증서를 자동으로 교체

Red Hat OpenShift는 다음과 같은 방법으로 유휴 플랫폼 데이터를 보호합니다.

- 보안 강화를 위해 선택적으로 Red Hat Enterprise Linux CoreOS 디스크 및 etcd 데이터 저장소를 암호화
- Red Hat OpenShift에 연방 정보 처리 표준(FIPS) 준비 상태 제공 FIPS 140-2는 암호화 모듈을 승인하는 데 사용되는 미국 정부의 보안 표준입니다. Red Hat Enterprise Linux CoreOS가 FIPS 모드로 부팅되면 Red Hat OpenShift 플랫폼 구성 요소는 Red Hat Enterprise Linux 암호화 모듈을 호출합니다.

컨테이너는 스테이트리스(Stateless) 및 스테이트풀(Stateful) 애플리케이션 모두에 유용합니다. Red Hat OpenShift는 일회성 스토리지와 영구 스토리지를 둘 다 지원합니다. 컨테이너에 연결되어 있는 스토리지를 보호하는 것은 보안 스테이트풀(Stateful) 서비스의 핵심 요소입니다. Red Hat OpenShift는 네트워크 파일 시스템(NFS), Amazon Web Services(AWS) Elastic Block Stores(EBS), Google Compute Engine(GCE) 영구 디스크, Azure 디스크, iSCSI, Cinder 등 여러 가지 스토리지 유형을 지원합니다.

또한 Red Hat OpenShift Container Storage는 Red Hat OpenShift Container Platform과 통합되는 영구 소프트웨어 정의 스토리지로서 Red Hat OpenShift Container Platform에 최적화되어 있습니다. OpenShift Container Storage는 암호화, 복제, 하이브리드 멀티클라우드 전반에 걸친 가용성과 같은 기능이 필요한 클라우드 네이티브 애플리케이션을 위한 고확장성 영구 스토리지를 제공합니다.

- **PV(퍼시스턴트 볼륨)**는 리소스 제공업체가 지원하는 어떤 방식으로든 호스트에 마운트할 수 있습니다. 제공업체들은 서로 다른 기능을 제공하며 각 PV의 액세스 모드는 해당 볼륨이 지원하는 특정 모드로 설정됩니다. 예를 들어 NFS에서 여러 번의 읽기/쓰기 클라이언트가 지원될 수 있지만 특정 NFS PV는 서버에서 읽기 전용으로 내보낼 수 있습니다. 각 PV에는 해당 PV에 대해 허용할 기능과 관련된 개별 액세스 모드들을 가집니다. 이들은 ReadWriteOnce, ReadOnlyMany, ReadWriteMany 등입니다.

- **공유 스토리지**(예: NFS, Ceph, Gluster)의 경우 공유 스토리지 퍼시스턴트 볼륨(PV)에 해당 그룹 ID(gid)를 PV 리소스의 주석으로 등록할 수 있습니다. 포드에서 PV를 요청하면 주석이 지정된 gid가 포드의 보조 그룹에 추가되고, 공유 스토리지의 콘텐츠에 액세스할 수 있는 권한이 포드에 부여됩니다.
- **블록 스토리지**(예: EBS, GCE Persistent Disks, iSCSI)의 경우 컨테이너 플랫폼은 권한이 부여되지 않은 포드에 대해 마운트된 볼륨의 루트를 보호하기 위해 SELinux 기능을 사용할 수 있으며, 마운트된 볼륨은 여기에 연결된 컨테이너에 속하게 되고 이 컨테이너에 의해서만 확인이 가능하게 됩니다.

물론, 현재 사용하는 스토리지 솔루션에서 제공되는 보안 기능도 활용해야 합니다.

8. 정책 기반 배포 자동화

보안 관점에서 본다면 강력한 보안에는 컨테이너 및 클러스터 배포를 관리하는 데 사용할 수 있는 자동화된 정책이 포함되어야 합니다.

- 정책 기반 컨테이너 배포

이미지가 특정 이미지 레지스트리에서 풀링되도록 허용하거나 허용하지 않도록 Red Hat OpenShift 클러스터를 구성할 수 있습니다. 모범 사례는 프로덕션 클러스터가 프라이빗 레지스트리에서 이미지가 배포되는 것만 허용하게 하는 것입니다.

Red Hat OpenShift의 보안 컨텍스트 제한 조건(SCC) 권한 컨트롤러 플러그인은 시스템에서 포드를 사용하기 위해 필요한 일련의 포드 실행 조건을 정의합니다. **보안 컨텍스트 제한 조건**은 권한 축소 기능을 기본적으로 제공합니다. 이 중요한 기능은 여전히 모범 사례에 해당합니다. Red Hat OpenShift 보안 컨텍스트 제한 조건(SCC)은 권한 있는 컨테이너가 OpenShift 작업자 노드에서 실행되지 않도록 기본적으로 보장합니다. 호스트 네트워크 및 호스트 프로세스 ID에 대한 액세스는 기본적으로 거부됩니다.

필요한 권한을 가진 사용자는 기본 SCC 정책을 조정하여 허용 범위를 넓힐 수 있습니다.

Red Had Advanced Cluster Management for Kubernetes는 개방형 표준을 사용하는 **고급 애플리케이션 라이프사이클 관리**를 제공합니다. 이를 통해 기존 CI/CD 파이프라인 및 거버넌스 제어로 통합되는 배치 정책을 사용해 애플리케이션을 배포합니다.

- 정책 기반 멀티클러스터 관리

여러 개의 클러스터를 배포하면 여러 가용성 영역에서 애플리케이션 고가용성을 제공하거나 Amazon Web Services(AWS), Google Cloud, Microsoft Azure 등 여러 클라우드 제공업체에 걸친 배포 또는 마이그레이션을 공통 관리할 수 있는 기능을 제공하는 데 도움이 됩니다. 여러 클러스터를 관리하는 경우 오케스트레이션 툴이 다양한 배포 인스턴스 전반에 필요한 보안을 제공해야 합니다. 항상 그렇듯 구성, 인증, 권한 부여가 핵심 요소이긴 하지만, 애플리케이션이 실행되는 위치에 관계없이 데이터를 애플리케이션에 안전하게 전달하고 클러스터에 전반에 걸쳐 애플리케이션 정책을 관리하는 기능도 중요합니다. Red Had Advanced Cluster Management for Kubernetes가 제공하는 기능은 다음과 같습니다.

- **멀티클러스터 라이프사이클 관리**: 쿠버네티스 클러스터를 안정적으로, 일관성 있게, 규모에 따라 생성, 업데이트, 폐기할 수 있습니다.
- **정책 기반 거버넌스 위험 및 컴플라이언스**: 정책을 이용해 업계 기업 표준에 따라 보안 제어의 일관성을 자동으로 구성하고 유지할 수 있습니다. 한 개 이상의 관리형 클러스터에 적용할 컴플라이언스 정책을 지정할 수도 있습니다.

실행 중인 애플리케이션 보호

인프라에 관계없이 애플리케이션 보안을 유지하는 것이 매우 중요합니다. 컨테이너화된 애플리케이션의 보안을 유지하는 데 필요한 사항은 다음과 같습니다.

1. 컨테이너 격리
2. 애플리케이션 및 네트워크 격리
3. 애플리케이션 액세스 보안
4. 관측성

9. 컨테이너 격리

운영팀이 컨테이너 패키징 및 오케스트레이션 기술을 최대한 활용하려면 컨테이너를 실행하기에 적합한 환경이 필요합니다. 즉 운영팀은 컨테이너 내부로부터 호스트 커널을 보호하고 컨테이너 간의 보안을 유지하는 등 컨테이너 격리를 통해 보호할 수 있는 OS(운영 체제)가 필요합니다.

컨테이너는 격리 및 리소스 제한 기능을 갖춘 Linux 프로세스로서, 공유 호스트 커널에서 샌드박스형 애플리케이션을 실행할 수 있게 합니다. 컨테이너 보안을 위한 접근 방식은 Linux에서 실행 중인 프로세스를 위한 접근 방식과 동일합니다.

NIST 특별 간행물 800-190에서는 보안 강화를 위해 컨테이너 최적화 OS를 사용할 것을 권장합니다. Red Hat OpenShift의 운영 체제 기반인 Red Hat Enterprise Linux CoreOS는 호스트 환경을 최소화하고 컨테이너에 맞게 튜닝함으로써 공격 표면을 줄입니다. Red Hat Enterprise Linux CoreOS에는 Red Hat OpenShift를 실행하는 데 필요한 패키지만 포함되어 있고 사용자 공간은 읽기 전용입니다. 이 플랫폼은 테스트 및 버전 설정을 거쳐 Red Hat OpenShift와 함께 제공되며 클러스터에 의해 관리됩니다. Red Hat Enterprise Linux CoreOS 설치 및 업데이트는 자동화되어 있으며 항상 클러스터와 호환됩니다. 또한 사용자가 선택한 인프라를 지원하므로 Red Hat Enterprise Linux 에코시스템의 대부분을 상속합니다.

Red Hat OpenShift 플랫폼에서 실행 중인 모든 Linux 컨테이너는 Red Hat OpenShift 노드에 구축된 강력한 Red Hat Enterprise Linux 보안 기능으로 보호됩니다. Linux 네임스페이스, SELinux, Cgroups, Linux 기능, 보안 컴퓨팅 모드(seccomp)는 Red Hat Enterprise Linux에서 실행되는 컨테이너를 보호하는 데 사용됩니다.

- Linux 네임스페이스는 컨테이너 격리의 기반을 제공합니다. 네임스페이스는 자체 시스템 자원을 소유한 것처럼 독립된 인스턴스로서 프로세스들에게 보여집니다. 네임스페이스는 추상화를 통해 사용자가 컨테이너 내부에서 자신의 운영 체제를 기반으로 실행하고 있는 것처럼 느끼게 만듭니다.
- SELinux는 보안 계층을 추가로 제공하여 컨테이너 간의 격리 또는 호스트로부터의 컨테이너 격리를 유지할 수 있도록 합니다. 관리자는 SELinux를 통해 모든 사용자, 애플리케이션, 프로세스, 파일에 대해 MAC(필수 액세스 제어)를 시행할 수 있습니다. SELinux는 사용자가 실수로 또는 의도적으로 네임스페이스 추상화 영역을 벗어나게 되는 경우 사용자를 저지하는 장벽과 같은 역할을 수행합니다. SELinux는 컨테이너 런타임 취약점을 완화하며, SELinux를 잘 구성하면 컨테이너 프로세스가 자체적인 제한에서 벗어나는 것을 방지할 수 있습니다.

- Cgroups(제어 그룹)는 프로세스 모음의 리소스 사용량(예: CPU, 메모리, 디스크 I/O, 네트워크)을 제한하고 처리하며 격리합니다. Cgroups를 사용하면 동일한 호스트에 있는 다른 컨테이너가 사용자의 컨테이너 리소스를 방해하지 않게 할 수 있습니다. 자주 이용되는 공격 벡터인 의사(pseudo) 장치를 제어하는 데에도 Cgroups를 사용할 수 있습니다.
- Linux 기능으로 컨테이너 내에서 권한을 잠글 수 있습니다. 이 기능은 독립적으로 활성화되거나 비활성화할 수 있는 권한의 단위입니다. 이를 사용하여 원시 인터넷 프로토콜(IP) 패킷을 전송하거나 1024 미만의 포트에 바인딩하는 등의 작업을 할 수 있습니다. 컨테이너를 실행할 때 컨테이너화된 애플리케이션 실행에 큰 영향이 없어 여러 기능을 중지할 수 있습니다.
- 마지막으로 보안 컴퓨팅 모드(seccomp) 프로필을 사용하여 컨테이너에서 사용 가능한 시스템 콜에 제한을 둘 수 있습니다.

10. 애플리케이션 및 네트워크 격리

쿠버네티스를 엔터프라이즈 규모로 사용하려면 멀티테넌트 보안이 필수입니다. 멀티테넌시를 통해 여러 팀이 동일한 클러스터를 사용할 수 있게 지원하는 동시에 서로의 환경에 무단 액세스하지 못하게 할 수 있습니다. Red Hat OpenShift는 커널 네임스페이스, SELinux, RBAC, 쿠버네티스(프로젝트) 네임스페이스, 네트워크 정책의 조합을 통해 멀티테넌시를 지원합니다.

- **Red Hat OpenShift 프로젝트**는 SELinux 주석이 있는 쿠버네티스 네임스페이스입니다. 프로젝트를 통해 여러 팀, 그룹, 부서에 걸쳐 애플리케이션을 격리합니다. 로컬 역할 및 바인딩은 개별 프로젝트에 액세스할 수 있는 권한을 누구에게 부여할지 제어하는 데 사용됩니다.
- **보안 컨텍스트 제한 조건**은 권한 축소 기능을 기본적으로 제공합니다. 이 중요한 기능은 여전히 모범 사례에 해당합니다. Red Hat OpenShift 보안 컨텍스트 제한 조건(SCC)은 권한 있는 컨테이너가 OpenShift 작업자 노드에서 실행되지 않도록 기본적으로 보장합니다. 호스트 네트워크 및 호스트 프로세스 ID에 대한 액세스는 기본적으로 거부됩니다.

현대적인 마이크로서비스 기반 애플리케이션을 컨테이너에 배포하는 경우 다수의 노드에 여러 컨테이너를 분산하여 배포하게 됩니다. 이 마이크로서비스는 상호 검색 및 통신이 가능해야 합니다. 네트워크 방어를 감안하면 단일 클러스터를 취해 트래픽을 세그먼트화함으로써 해당 클러스터 내의 여러 사용자, 팀, 애플리케이션, 환경을 격리할 수 있는 컨테이너 플랫폼이 필요합니다. 클러스터에 대한 외부 액세스와 클러스터 서비스에서 외부 구성 요소에 대한 액세스를 관리할 수 있는 툴도 필요합니다. 네트워크 격리를 완벽히 수행하려면 다음과 같은 주요 속성이 필요합니다.

- **인그레스 트래픽 제어.** Red Hat OpenShift에는 이름 확인 서비스를 포드에 제공하는 CoreDNS가 포함되어 있습니다. Red Hat OpenShift 라우터(HAProxy)는 클러스터에서 실행되는 서비스에 대한 외부 액세스를 제공하는 인그레스 라우팅을 지원합니다. 둘 다 재암호화 및 패스스루 정책을 지원합니다. 즉 "재암호화"는 HTTP 트래픽을 포워딩할 때 이 HTTP 트래픽을 해독했다가 다시 암호화하는 반면, "패스스루"는 TLS를 종료하지 않고 트래픽을 패스스루합니다.
- **네트워크 네임스페이스.** 네트워크의 1차 방어선은 네트워크 네임스페이스에서 시작됩니다. 각 컨테이너 컬렉션('포드'라고 함)이 고유한 IP 및 포트 범위를 가져와 바인딩을 수행하며, 이를 통해 노드의 포드 네트워크가 서로 격리됩니다. 포드 IP 주소는 Red Hat OpenShift 노드가 연결되어 있는 물리 네트워크와는 별개로 독립되어 있습니다.

- **네트워크 정책:** Red Hat OpenShift SDN은 네트워크 정책을 사용해 포드 간 통신을 정교하게 제어합니다. 한 포드에서 다른 포드로의 네트워크 트래픽은 특정 포트에서, 특정 방향으로 제어할 수 있습니다. 네트워크 정책이 멀티테넌트 모드로 구성되어 있으면 각 프로젝트는 자체 가상 네트워크 ID를 가져와 노드에서 프로젝트 네트워크를 서로 격리합니다. 멀티테넌트 모드(기본 설정)에서 프로젝트 내 포드는 서로 통신할 수 있지만 서로 다른 네임스페이스의 포드는 다른 프로젝트의 포드나 서비스와 패킷을 주고받을 수 없습니다.

- **이그레스 트래픽 제어.** Red Hat OpenShift는 라우터 또는 방화벽 메서드를 사용해 클러스터에서 실행되는 서비스에서 발신되는 이그레스 트래픽을 제어할 수 있는 기능을 제공합니다. 예를 들어 IP 화이트리스팅을 사용해 외부 데이터베이스에 대한 액세스를 제공할 수 있습니다.

11. 애플리케이션 액세스 보안

애플리케이션 보안에는 애플리케이션 사용자와 API 인증 및 권한 부여를 관리하는 것이 포함됩니다.

- **사용자 액세스 제어**

웹 SSO(Single Sign-On) 기능은 현대적인 애플리케이션에서 핵심이 되는 부분입니다. 개발자는 컨테이너 플랫폼과 함께 제공되는 다수의 컨테이너화 서비스를 애플리케이션을 빌드할 때 사용할 수 있습니다. Red Hat Single Sign-On은 완전히 지원되는 탁월한 SAML(Security Assertion Markup Language) 2.0 또는 OpenID Connect 기반 인증, 웹 Single Sign-On, 업스트림 Keycloak 프로젝트 기반 페더레이션 서비스입니다. Red Hat Single Sign-On은 Red Hat JBoss Fuse 및 Red Hat JBoss Enterprise Application Platform을 위한 클라이언트 어댑터를 제공합니다. Red Hat Single Sign-On은 Node.js 애플리케이션을 위한 인증 및 웹 Single Sign-On을 지원하며 Microsoft Active Directory, Red Hat Enterprise Linux Identity Management와 같은 LDAP 기반 디렉터리 서비스와 통합할 수 있습니다. Facebook, Google, Twitter와 같은 소셜 로그인 제공업체와도 통합됩니다.

- **API 액세스 제어**

API는 마이크로서비스로 구성된 애플리케이션의 핵심 요소입니다. 이러한 애플리케이션에는 여러 개의 독립적 API 서비스가 있어 거버넌스를 위해서는 추가적인 툴이 필요한 서비스 엔드포인트가 급증하게 됩니다. API 관리 툴을 사용하는 것이 권장됩니다. Red Hat 3scale API Management는 API 인증 및 보안을 위한 다양한 표준 옵션을 제공합니다. 이 옵션은 단독으로 또는 조합하여 자격 증명을 발행하고 액세스를 제어하는 데 사용할 수 있습니다.

Red Hat 3scale API Management에서 제공하는 액세스 제어 기능은 기본적인 보안 및 인증에 국한되지 않습니다. 애플리케이션 및 계정 계획을 사용하면 특정 엔드포인트, 메서드, 서비스에 대한 액세스를 제한하고 사용자 그룹에 액세스 정책을 적용할 수 있습니다. 애플리케이션 계획을 통해서는 API 사용에 관한 비율(Rate) 제한을 설정하고 개발자 그룹을 위해 트래픽 흐름을 제어할 수 있습니다. 수신 API 호출에 기간별 제한을 설정하여 인프라를 보호하고 트래픽 흐름을 원활하게 유지할 수 있습니다. 또한 애플리케이션이 속도 제한에 도달하거나 이를 초과하는 경우 과잉 경고를 자동으로 트리거하고 제한 초과 애플리케이션에 대한 동작을 정의할 수 있습니다.

- **애플리케이션 트래픽 보안**

클러스터 인그레스 및 이그레스 옵션을 이용한 애플리케이션 트래픽 보안은 이 문서의 섹션 10에서 다룹니다. 마이크로서비스 기반 애플리케이션의 경우 클러스터의 서비스 간 보안 트래픽도 마찬가지로 중요합니다. 서비스 메쉬를 사용해 이 관리 계층을 제공할 수 있습니다. "서비스 메쉬"는 분산된 마이크로서비스 아키텍처의 애플리케이션을 구성하는 마이크로서비스 네트워크와 이러한 마이크로서비스 간 상호 작용을 지칭하는 용어입니다.

오픈소스 Istio 프로젝트를 바탕으로 Red Hat OpenShift Service Mesh는 서비스 코드 변경 없이 서비스 간 커뮤니케이션을 관리하기 위해 기존의 분산된 애플리케이션에 투명한 계층을 추가합니다. Red Hat OpenShift Service Mesh는 멀티테넌트 오퍼레이터를 사용해 컨트롤 플레인 라이프사이클을 관리하므로 OpenShift Service Mesh를 프로젝트별로 사용할 수 있습니다. 뿐만 아니라 OpenShift Service Mesh는 클러스터 범위의 RBAC 리소스가 필요 없습니다.

Red Hat OpenShift Service Mesh는 검색, 부하 분산, 보안 키, 서비스 간 인증 및 암호화, 장애 복구, 지표, 모니터링 등의 기능을 제공합니다.

3scale Istio Adapter는 Red Hat OpenShift Service Mesh 내에서 실행되는 서비스에 레이블을 지정할 수 있는 어댑터입니다(선택 사항).

12. 관측성

Red Hat OpenShift 클러스터 모니터링 및 감사는 클러스터와 클러스터 사용자를 부적절한 사용으로부터 보호하는 중요한 기능입니다. Red Hat OpenShift에는 기본 제공되는 모니터링 및 감사뿐 아니라 선택 사항으로 제공되는 로깅 스택도 포함되어 있습니다.

OpenShift Container Platform 서비스는 Prometheus와 그 에코시스템으로 구성된 기본 제공 모니터링 솔루션에 연결됩니다. 경고 대시보드가 제공됩니다. 클러스터 관리자는 사용자 정의 프로젝트에 대한 모니터링을 선택적으로 활성화할 수 있습니다. Red Hat OpenShift에 배포되는 애플리케이션을 구성하여 클러스터 모니터링 구성 요소를 활용할 수 있습니다.

이벤트 감사는 보안 모범 사례이며 일반적으로 규제 프레임워크를 준수하는 데 필요합니다. 본질적으로 Red Hat OpenShift 감사는 클라우드 네이티브 접근 방식을 사용해 중앙화 및 복구를 모두 제공하도록 설계되었습니다. Red Hat OpenShift에서 호스트 감사 및 이벤트 감사는 모든 노드에서 기본적으로 활성화되어 있습니다. Red Hat OpenShift는 데이터 감사에 대한 관리 및 액세스를 구성할 수 있는 탁월한 유연성을 제공합니다. 사용할 감사 로그 정책 프로필을 선택하여 API 서버 감사 로그에 로깅되는 정보의 양을 제어할 수 있습니다.

모니터링, 감사, 로그 데이터는 RBAC로 보호됩니다. 프로젝트 데이터는 프로젝트 관리자에게 제공되고, 클러스터 데이터는 클러스터 관리자에게 제공됩니다.

모범 사례로 클러스터가 모든 감사 및 로그 이벤트를 무결성 관리, 유지, 분석을 위한 보안 정보 및 이벤트 관리(SIEM) 시스템으로 전달하도록 구성하십시오. 클러스터 관리자는 클러스터 로깅을 배포하여 애플리케이션 컨테이너 로그 및 인프라 로그뿐 아니라 호스트 및 API 감사 로그와 같은 Red Hat OpenShift 클러스터에서 모든 로그를 집계할 수 있습니다. 클러스터 로깅은 클러스터 노드 전체에서 이러한 로그를 집계하여 기본 로그 스토어에 저장합니다. 로그를 선택한 SIEM으로 전달할 수 있는 옵션이 여러 가지 제공됩니다.

강력한 에코시스템을 이용한 보안 확장

컨테이너 및 쿠버네티스 보안을 더 강화하거나 기존 정책을 충족하기 위해 타사 보안 툴과의 통합을 선택할 수 있습니다. Red Hat은 다음과 같은 솔루션을 제공하는 인증된 파트너로 구성된 광범위한 에코시스템을 보유하고 있습니다.

- 권한 있는 액세스 관리
- 외부 인증 기관
- 외부 암호 저장소(vault) 및 핵심 관리 솔루션
- 컨테이너 콘텐츠 스캐너 및 취약점 관리 툴
- 컨테이너 런타임 분석 툴
- SIEM

결론

컨테이너 기반 애플리케이션 및 마이크로서비스 배포는 단지 보안과 관련된 것은 아닙니다. 컨테이너 플랫폼은 개발자와 운영팀이 유용하게 사용할 수 있어야 합니다. 각 팀에 필요한 기능을 모두 제공하면서 개발자와 운영자 모두를 지원하며 운영 효율성과 인프라 활용률도 높일 수 있는 보안 중심의 엔터프라이즈급 컨테이너 기반 애플리케이션 플랫폼이 필요합니다.

Red Hat OpenShift는 다음과 같은 기본 보안 기능을 제공하는 이식 가능한 표준 Linux 컨테이너의 중심에 구축됩니다.

- 안전한 DevOps 적용 사례를 위한 통합 빌드 및 CI/CD 툴
- 플랫폼 구성, 컴플라이언스, 라이프사이클 관리가 내장된 엔터프라이즈 수준의 강화된 쿠버네티스
- 엔터프라이즈 인증 시스템과의 통합 기능을 갖춘 강력한 RBAC
- 클러스터 인그레스 및 이그레스를 관리할 수 있는 옵션
- 네트워크 마이크로 세그멘테이션을 지원하는 통합 SDN 및 서비스 메쉬
- 원격 스토리지 볼륨 보안에 대한 지원
- 강력한 격리로 규모에 맞게 컨테이너를 실행하는 데 최적화된 Red Hat Enterprise Linux CoreOS
- 런타임 보안 자동화를 위한 배포 정책
- 통합된 모니터링, 감사, 로깅

Red Hat OpenShift는 지원되는 프로그래밍 언어, 프레임워크, 서비스의 방대한 컬렉션도 제공합니다(그림 4). Red Hat Advanced Cluster Management for Kubernetes는 긴밀하게 통합된 멀티클러스터 관리를 제공합니다.

Red Hat OpenShift는 OpenStack, VMware, 베어 메탈, AWS, Google Cloud Platform(GCP), Azure, IBM Cloud와 Red Hat Enterprise Linux를 지원하는 모든 플랫폼에서 실행할 수 있습니다. Red Hat은 AWS 및 GCP에서 Red Hat OpenShift Dedicated도 퍼블릭 클라우드 서비스로 제공합니다. Azure Red Hat OpenShift는 Red Hat과 Microsoft의 협력을 통해 제공됩니다. AWS 기반 Red Hat OpenShift Service는 Red Hat과 Amazon의 협력을 통해 제공됩니다.

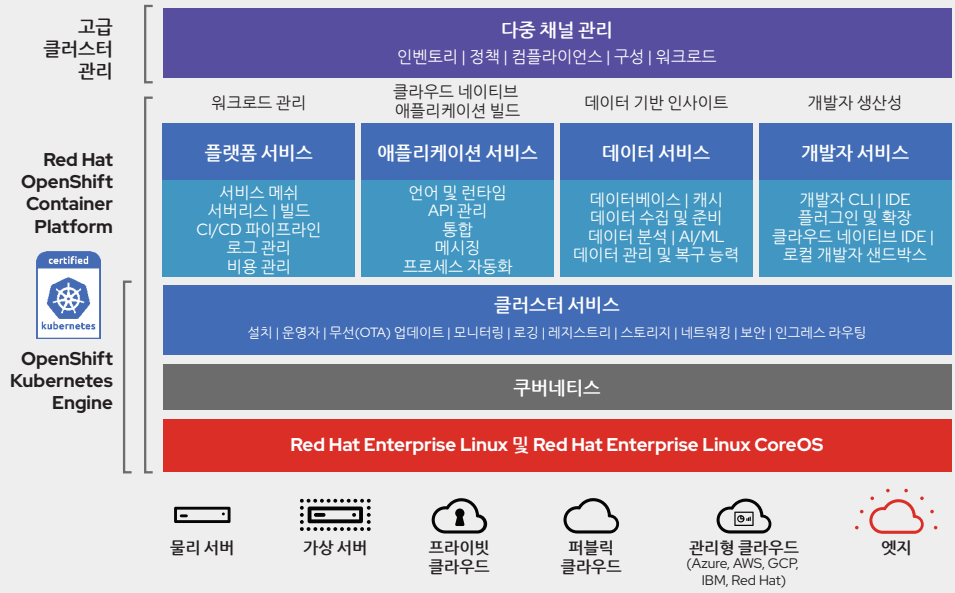

그림 4. Red Hat OpenShift는 물리적인 가상, 프라이빗, 퍼블릭 클라우드 전반에 걸쳐 기능이 완비된 기술 스택을 제공합니다.

Red Hat은 20년 넘게 신뢰할 수 있는 오픈소스 솔루션을 기업에 제공하는 선두 업체였습니다. 이제 Red Hat OpenShift Container Platform, Red Hat Advanced Cluster Management for Kubernetes, 컨테이너를 지원하는 Red Hat 제품 포트폴리오와 같은 솔루션을 통해 이와 동일한 수준의 신뢰도와 보안을 컨테이너에서 실현하고 있습니다.

한국레드햇 홈페이지 https://www.redhat.com/ko

RED HAT 정보

Red Hat은 세계적인 엔터프라이즈 오픈소스 솔루션 공급업체로서 커뮤니티 기반 접근 방식을 통해 신뢰도 높은 고성능 Linux, 하이브리드 클라우드, 컨테이너, 쿠버네티스 기술을 제공합니다. 또한 고객으로 하여금 신규 및 기존 IT 애플리케이션을 통합하고, 클라우드 네이티브 애플리케이션을 개발하며, 업계를 선도하는 Red Hat의 운영 체제를 기반으로 표준화하는 동시에 복잡한 환경의 자동화, 보안 및 관리를 실현할 수 있도록 지원합니다. Red Hat은 전세계 고객에게 높은 수준의 지원과 교육 및 컨설팅 서비스를 제공하여 권위있는 어워드를 다수 수상한 바 있으며, Fortune 선정 500대 기업의 신뢰를 받는 어드바이저로 인정받고 있습니다. 또한 기업, 파트너, 오픈소스 커뮤니티의 전략적인 파트너로서 고객들이 디지털 미래에 대비할 수 있도록 지원하고 있습니다.

www.facebook.com/redhatkorea
구매문의 080 708 0880
buy-kr@redhat.com

www.redhat.com/ko
#F26463_1220