Ein mehrschichtiger Sicherheitsansatz für Container und Kubernetes

Container in allen Phasen sichern – beim Erstellen, Bereitstellen und Ausführen

Inhaltsverzeichnis

facebook.com/redhatinc
@RedHatDACH
linkedin.com/company/red-hat

Einleitung

Container haben großen Anklang gefunden, da sie Anwendungen und ihre Abhängigkeiten in einem einzigen Image verpacken können, das dann von der Entwicklung zum Test und weiter in die Produktion gebracht werden kann. Mit Containern kann Konsistenz in Umgebungen und mehreren Bereitstellungszielen wie physischen Servern, virtuellen Maschinen (VMs) und Private wie Public Clouds auf einfache Weise sichergestellt werden. Außerdem können Teams mit Containern die Anwendungen einfacher entwickeln und managen, die für geschäftliche Agilität sorgen.

- **Anwendungen:** Mit Containern können Entwickler Anwendungen und ihre Abhängigkeiten leichter als Einheit erstellen und durch den Entwicklungsprozess schieben. Container können innerhalb von Sekunden bereitgestellt werden. In einer containerisierten Umgebung ist die Software-Erstellung die Phase des Lifecycles, in der der Anwendungscode in die erforderlichen Laufzeitbibliotheken integriert wird.
- **Infrastruktur:** Container sind Sandbox-Anwendungsprozesse auf einem gemeinsam genutzten Linux®-BS-Kernel. Sie sind kompakter, kleiner und weniger komplex als virtuelle Maschinen und zwischen verschiedenen Umgebungen portierbar – von lokalen bis zu Public Cloud-Plattformen.

Kubernetes ist die beliebteste Container-Orchestrierungsplattform für Unternehmen. Da viele Organisationen grundlegende Services jetzt auf Containern ausführen, ist es wichtiger denn je, die Sicherheit von Containern zu gewährleisten. In diesem Dokument werden die wichtigsten Sicherheitselemente für containerisierte Anwendungen beschrieben.

Umfassende Sicherheit für Container und Kubernetes: Schichten und Lifecycle

Die Sicherung von Containern läuft ähnlich ab wie bei laufenden Linux-Prozessen. Sie sollten die Sicherheit aller Schichten des Lösungs-Stacks miteinbeziehen, bevor Sie Ihren Container bereitstellen und ausführen. Genauso sollten Sie an die Sicherheit aller Phasen des Anwendungs- und Container-Lifecycles denken. Und ganz wichtig: Sicherheit muss ein kontinuierlicher Prozess sein, der in den gesamten IT-Lifecycle integriert ist und auch die Reaktion auf neue Bedrohungen und Lösungen mit einschließt. Abbildung 1 zeigt einen umfassenden Sicherheitsansatz für Container.

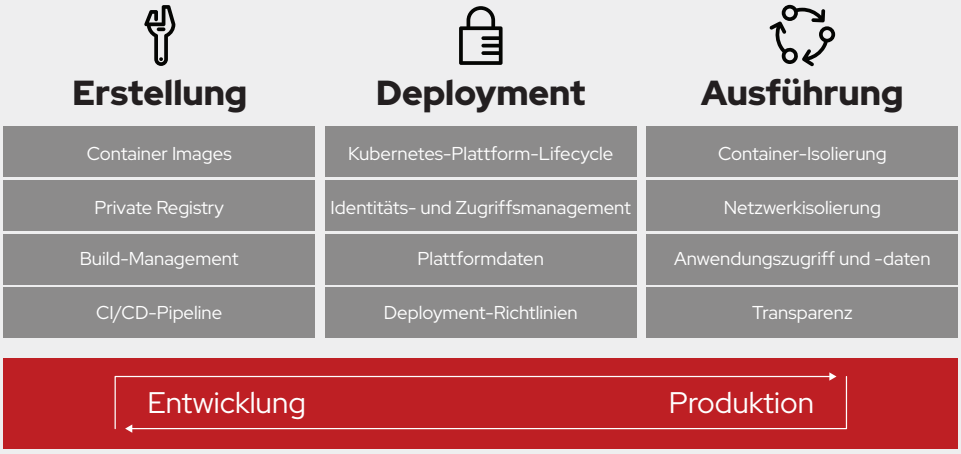

Abbildung 1. Umfassende Container-Sicherheit beinhaltet die Sicherung von Build, Deployment und Runtime sowie bei Bedarf die Erweiterung der Sicherheit durch spezielle Tools.

Mit Containern können Entwickler Anwendungen und ihre Abhängigkeiten leichter als Einheit erstellen und durch den Entwicklungsprozess schieben. Container machen es auch einfacher, Server optimal zu nutzen, da sie Deployments von mandantenfähigen Anwendungen auf einem gemeinsam verwendeten Host ermöglichen. Sie können mehrere Anwendungen einfach auf einem einzigen Host bereitstellen und einzelne Container nach Bedarf hoch- und herunterfahren. Im Gegensatz zur traditionellen Virtualisierung brauchen Sie keinen Hypervisor, um Guest-Betriebssysteme auf allen VMs zu managen. Container virtualisieren Ihre Anwendungsprozesse, nicht Ihre Hardware.

Natürlich werden Anwendungen selten in einem einzigen Container bereitgestellt. Sogar einfache Anwendungen haben üblicherweise ein Frontend, ein Backend und eine Datenbank. Außerdem bedeutet das Deployment moderner, auf Microservices basierender Anwendungen in Containern, dass mehrere Container bereitgestellt werden müssen – manchmal auf dem gleichen Host, manchmal auf mehreren Hosts oder Knoten, wie in Abbildung 2 dargestellt.

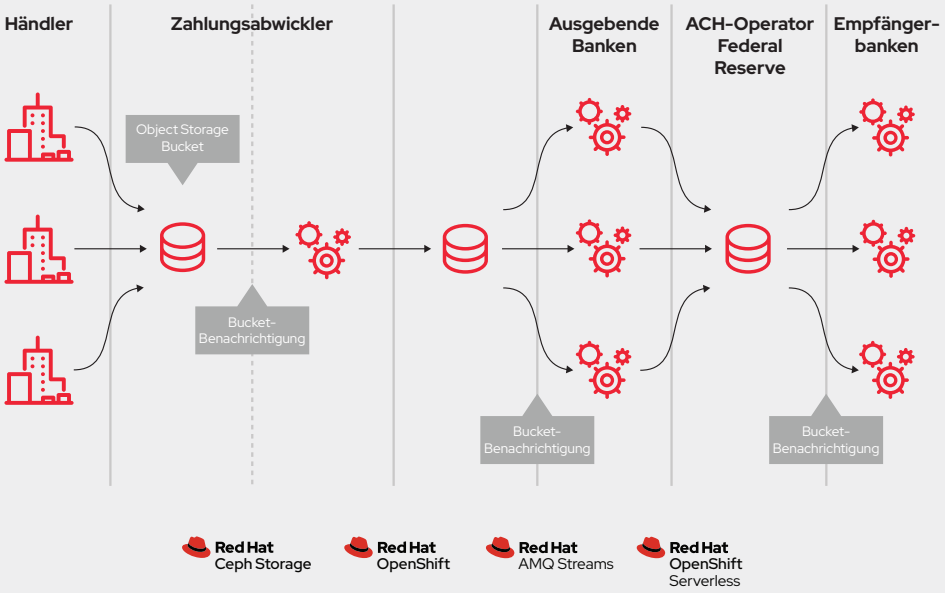

Abbildung 2. Container-Umgebungen können komplex und auf mehrere Knoten verteilt sein.

Wenn Sie Container Deployments in großem Umfang managen, sollten Sie Folgendes berücksichtigen:

- Welche Container sollen auf welchen Hosts bereitgestellt werden?
- Welcher Host hat eine größere Kapazität?
- Welche Container müssen aufeinander zugreifen und wie können sie sich finden?
- Wie kontrollieren Sie den Zugang zu und das Management von gemeinsamen Ressourcen wie Netzwerk und Storage?
- Wie überwachen Sie den Container-Zustand?
- Wie lässt sich die Anwendungskapazität bedarfsabhängig skalieren?
- Wie können Sie Self-Service-Funktionen für Entwickler ermöglichen und gleichzeitig Sicherheitsanforderungen gerecht werden?

Sie können Ihre eigene Umgebung für das Management von Containern entwickeln. Es braucht aber Zeit, individuelle Komponenten zu integrieren und zu verwalten. Oder Sie stellen eine Container-Plattform bereit, in die Management- und Sicherheitsfunktionen bereits integriert sind. Mit diesem Ansatz kann Ihr Team seine Energie auf die Entwicklung von Anwendungen konzentrieren, die zum Geschäftswert beitragen statt die Infrastruktur neu zu erfinden.

Die Red Hat® OpenShift® Container Platform bietet Ihnen eine unternehmensgerechte Kubernetes-Plattform für die Hybrid Cloud, mit der Sie containerisierte Anwendungen konsistent entwickeln und skalieren können. Für eine unternehmensweite Nutzung von Kubernetes benötigen Sie zusätzliche Sicherheitsfunktionen: Funktionen, mit denen Sie Sicherheit in Ihre Anwendungen integrieren und Richtlinien zur Verwaltung der Sicherheit von Container Deployments automatisieren können, sowie Funktionen zum Schutz der Container Runtime.

Sicherheit in Anwendungen integrieren

Für cloudnative Deployments ist es wichtig, dass Sie Sicherheit von Anfang an in Ihre Anwendungen integrieren. Zur Sicherung Ihrer containerisierten Anwendungen müssen Sie Folgendes sicherstellen:

1. Verwenden Sie vertrauenswürdige Container-Inhalte.
2. Verwenden Sie eine Container Registry für Unternehmen.
3. Kontrollieren und automatisieren Sie die Erstellung von Containern.
4. Integrieren Sie Sicherheit in die Anwendungs-Pipeline.

1. Vertrauenswürdige Container-Inhalte verwenden

Beim Sicherheits-Management kommt es auf den Inhalt Ihrer Containern an. Seit einiger Zeit werden Anwendungen und Infrastrukturen aus allgemein verfügbaren Komponenten erstellt. Viele sind als Open Source-Pakete erhältlich, etwa das Linux-Betriebssystem, Apache Web Server, die Red Hat JBoss® Enterprise Application Platform, PostgreSQL und Node.js. Da es auch containerisierte Versionen dieser Pakete gibt, brauchen Sie nicht Ihre eigenen zu erstellen. Aber wie bei jedem anderen Code, den Sie von einer externen Quelle herunterladen, müssen Sie wissen, woher die Pakete ursprünglich stammen, wer sie entwickelt hat und ob sie bösartigen Code enthalten. Beantworten Sie folgende Fragen:

- Können die Container-Inhalte meine Infrastruktur gefährden?
- Gibt es bekannte Schwachstellen in der Anwendungsschicht?
- Sind die Betriebssystems- und Runtime-Schichten im Container auf dem neuesten Stand?
- Wie oft wird der Container aktualisiert, und woher weiß ich, wann er aktualisiert wird?

Red Hat stellt seit Jahren vertrauenswürdige Linux-Inhalte in Paketen bereit – über Red Hat Enterprise Linux und unser Portfolio. Jetzt stellt Red Hat dieselben vertrauenswürdigen Inhalte in Linux-Container-Paketen bereit. Mit der Einführung von Red Hat Universal Base Images genießen Sie eine höhere Zuverlässigkeit, Sicherheit und Performance bei Red Hat Container Images, auf denen OCI-konforme (Open Container Initiative) Linux-Container ausgeführt werden. Dadurch können Sie containerisierte Anwendungen auf Red Hat Universal Base Image erstellen, in einer beliebigen Container Registry bereitstellen und anschließend teilen.

Über den Red Hat Ecosystem Catalog bietet Red Hat außerdem eine Vielzahl an zertifizierten Images und Operatoren für verschiedene Sprach-Runtimes, Middleware, Datenbanken und mehr. Von Red Hat zertifizierte Container und Operatoren lassen sich überall ausführen, wo Red Hat Enterprise Linux ausgeführt wird – von Bare Metal über VMs bis hin zur Cloud – und werden von Red Hat und unseren Partnern unterstützt.

Red Hat überwacht den Zustand der bereitgestellten Images kontinuierlich. Der Container Health Index zeigt die „Einstufung" jedes Container Images und beschreibt, wie Container Images kuratiert, genutzt und getestet werden sollen, um den Anforderungen von Produktionssystemen gerecht zu werden. Die Einstufung von Containern basiert zum Teil auf dem Alter und den Auswirkungen von nicht angewendete Sicherheits-Errata auf die Komponenten eines Containers. Dadurch ergibt sich eine aggregierte Bewertung der Container-Sicherheit, die von Sicherheitsexperten wie Laien gleichermaßen verstanden werden kann.

Beim Release von Sicherheits-Updates – wie zum Beispiel Fixes für runc CVE-2019-5736, MDS CVE-2019-11091 oder VHOST-NET CVE-2019-14835 – erstellen wir bei Red Hat auch unsere Container Images neu und stellen sie in der öffentlichen Registry zur Verfügung. Red Hat macht Sie mit Sicherheitshinweisen auf neu entdeckte Probleme in zertifizierten Container Images aufmerksam und leitet Sie zum aktualisierten Image weiter, damit Sie Ihrerseits alle Anwendungen aktualisieren können, die das Image verwenden.

Es kann vorkommen, dass Red Hat Inhalte, die Sie brauchen, nicht zur Verfügung stellt. Wir empfehlen Ihnen, die Container mit Tools zu scannen, die kontinuierlich aktualisierte Sicherheitslücken-Datenbanken verwenden. So stellen Sie sicher, dass Sie immer die aktuellsten Informationen zu bekannten Sicherheitslücken haben, wenn Sie Container Images von anderen Quellen verwenden. Da die Liste bekannter Sicherheitslücken sich ständig erweitert, müssen Sie die Inhalte Ihrer Container Images beim Herunterladen prüfen und den Sicherheitsstatus im Laufe der Zeit für alle von Ihnen zugelassenen und bereitgestellten Images verfolgen – genau das tut auch Red Hat für Red Hat Container Images.

2. Container Registry für Unternehmen für mehr Sicherheit beim Zugriff auf Container Images verwenden

Natürlich erstellen Ihre Teams Container, in denen Inhalte als zusätzliche Schicht auf die heruntergeladenen öffentlichen Container Images aufgesetzt werden. Sie müssen verwalten, wie heruntergeladene Container Images und intern entwickelte Images zugänglich sind und bereitgestellt werden – so, wie Sie es auch für andere Arten von Binärdateien tun. Es gibt einige private Registries, die den Storage von Container Images unterstützen. Wählen Sie am besten eine private Registry, mit der Sie die Richtlinien zur Verwendung der Container Images automatisieren können, die in der Registry gespeichert sind.

Red Hat OpenShift beinhaltet eine private Registry mit grundlegenden Funktionen zur Verwaltung Ihrer Container Images. Die Registry von Red Hat OpenShift bietet Role-based Access Control (RBAC), wodurch Sie managen können, wer zum Push und Pull bestimmter Container Images berechtigt ist. Red Hat OpenShift unterstützt außerdem die Integration mit anderen privaten Registries, die Sie möglicherweise bereits verwenden, wie Artifactory von JFrog oder Sonatype Nexus.

Red Hat Quay ist als eigenständige Unternehmens-Registry verfügbar. Red Hat Quay bietet mehrere zusätzliche Funktionen für Unternehmen, darunter geografische Replikation und Trigger für Build-Images.

Das Open Source-Projekt Clair unterstützt den Sicherheits-Scanner von Red Hat Quay, um Schwachstellen in den Images in Red Hat Quay zu erkennen. Der Red Hat OpenShift Container Security Operator lässt sich mit Red Hat Quay integrieren und bietet so für alle Images, die Sie in der OpenShift-Konsole bereitgestellt haben, einen Überblick über alle bekannten Schwachstellen im gesamten Cluster.

3. Erstellung von Container Images kontrollieren und automatisieren

Diesen Build-Prozess zu managen, ist für die Sicherung des Software-Stacks entscheidend. Wenn Sie dem Motto „Einmal erstellen, überall bereitstellen" folgen, stellen Sie sicher, dass im Build-Prozess genau das Produkt entsteht, das in der Produktion bereitgestellt wird. Genauso wichtig ist es, die Unveränderlichkeit Ihrer Container aufrechtzuerhalten. Anders ausgedrückt, wenden Sie keine Patches auf aktive Container an – es ist besser, sie neu zu erstellen und bereitzustellen.

Um die Sicherheit Ihrer benutzerdefinierten Images zu verbessern, stellt Red Hat OpenShift eine Reihe von Funktionen bereit, mit denen Sie Builds auf Basis von externen Events automatisieren können.

- Red Hat Quay Trigger bieten einen Mechanismus, mit dem ein Repository Build einer Dockerfile von externen Events erzeugt werden, beispielsweise GitHub-Pushes, BitBucket -Pushes, GitLab-Pushes oder Webhooks.
- Source-to-Image (S2I) ist ein Open Source Framework für die Kombination von Quellcode mit Basis-Images (Abbildung 3). S2I erleichtert Ihren Entwicklungs- und Operations-Teams die Zusammenarbeit an einer reproduzierbaren Build-Umgebung. Wenn ein Entwickler in S2I mit Git Code schreibt, kann Red Hat OpenShift folgende Aktionen ausführen:
  - Automatische Zusammensetzung eines neuen Images aus verfügbaren Artefakten (einschließlich des S2I-Basis-Images) und des neu geschriebenen Codes (über Webhooks auf dem Code-Repository oder einen anderen automatisierten CI-Prozess)
  - Automatisches Deployment des neu erstellten Images für Tests
  - Hochstufen des getesteten Images in den Produktionsstatus und automatisches Deployment des neuen Images durch den CI/CD-Prozess (Continuous Integration/Continuous Deployment)

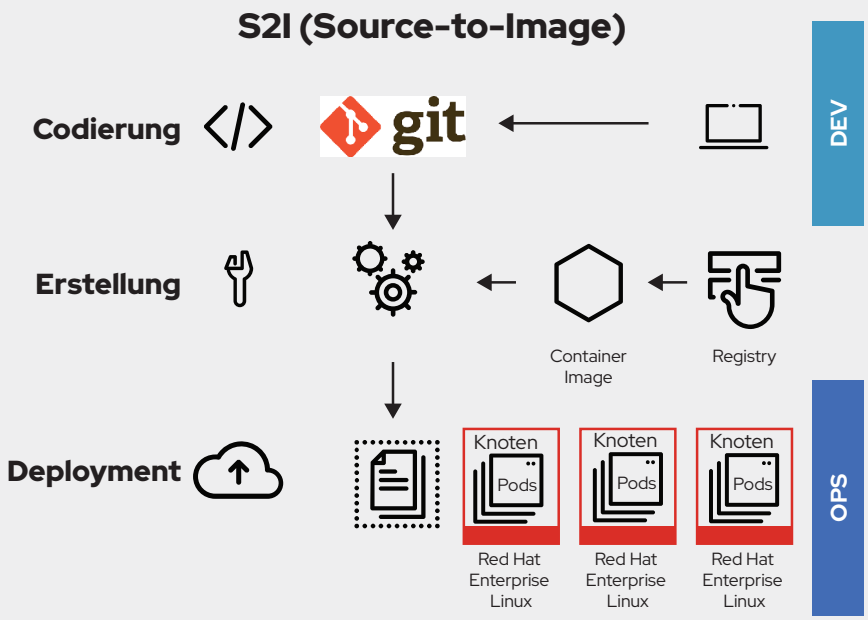

Abbildung 3. S2I ist ein Toolkit und Workflow zur Erstellung reproduzierbarer, sofort einsatzbereiter Container Images aus Quellcode.

- Mit den Image Streams von Red Hat OpenShift können Sie die Veränderungen an externen Images beobachten, die in Ihrem Cluster bereitgestellt werden. Image Streams funktionieren mit allen in Red Hat OpenShift verfügbaren nativen Ressourcen wie Builds, Deployments, Jobs, Replikations-Controller oder Replikat-Sets. Durch das Beobachten von Image Streams können Builds und Deployments Benachrichtigungen erhalten, wenn neue Images hinzugefügt oder geändert werden, und als Reaktion darauf automatisch ein Build bzw. Deployment ausführen.

  Stellen Sie sich beispielsweise ein Anwendungs-Build mit einem dreischichtigen Container Image vor: Basis-, Middleware- und Anwendungsschicht. Im Basis-Image wird ein Problem festgestellt, daher wird das Image von Red Hat neu erstellt und per Push in den Red Hat's Ecosystem Catalog übertragen. Wenn Image Streams aktiviert sind, erkennt Red Hat OpenShift, dass das Image geändert wurde. Für Builds, die von diesem Image abhängen und Trigger festgelegt haben, erstellt Red Hat OpenShift das Anwendungs-Image automatisch neu und integriert dabei das korrigierte Basis-Image.

  Sobald der Build abgeschlossen ist, wird das aktualisierte benutzerdefinierte Image in die interne Registry von Red Hat OpenShift übertragen. Red Hat OpenShift erkennt Änderungen an Images in der internen Registry sofort und stellt das aktualisierte Image automatisch für Anwendungen bereit, für die Trigger festgelegt wurden. So wird sichergestellt, dass der in der Produktion ausgeführte Code immer mit dem zuletzt aktualisierten Image übereinstimmt. Alle diese Funktionen arbeiten zusammen, um Sicherheit in Ihre CI/CD-Prozesse und -Pipeline zu integrieren.

4. Sicherheit in die Anwendungs-Pipeline integrieren

Red Hat OpenShift beinhaltet integrierte Instanzen von Jenkins für CI und Tekton, eine Kubernetes-CI/CD-Pipeline der nächsten Generation, die für Container verwendet werden kann (auch Serverless). Außerdem bietet Red Hat OpenShift RESTful APIs, mit denen Sie Ihre eigenen Builds oder CI/CD-Tools integrieren können, einschließlich einer privaten Image Registry.

Best Practices für die Anwendungssicherheit empfehlen, automatische Sicherheitstests in Ihre Pipeline zu integrieren, einschließlich Registry, IDE (Integrated Development Environment) und CI/CD-Tools.

Registry: Container Images können und sollten in Ihrer privaten Container Registry gescannt werden. Sie können Red Hat Quay mit dem Sicherheits-Scanner Clair verwenden, um Entwickler zu benachrichtigen, wenn Schwachstellen entdeckt wurden. Der Red Hat OpenShift Container Security Operator lässt sich mit Red Hat Quay integrieren und bietet so für alle Images, die Sie in der OpenShift-Konsole bereitgestellt haben, einen Überblick über alle bekannten Schwachstellen im gesamten Cluster. Alternativ finden Sie im Red Hat Ecosystem Catalog verschiedene zertifizierte Container-Sicherheitslösungen von Drittanbietern.

IDE: Die IDE-Plugins (Integrated Development Environment) in Red Hat Dependency Analytics warnen Sie bei der Übertragung von Code in die IDE über Sicherheitslücken und geben Ihnen Empfehlungen, wie Sie diese für Projektabhängigkeiten beheben können.

CI/CD: Zur Überprüfung in Echtzeit auf bekannte Sicherheitslücken können Sie Scanner mit CI integrieren. Diese katalogisieren die Open Source-Pakete in Ihrem Container, benachrichtigen Sie über alle bekannten Sicherheitslücken und informieren Sie, wenn neue Schwachstellen in zuvor gescannten Paketen entdeckt werden.

Ihr CI-Prozess sollte außerdem Richtlinien beinhalten, die Builds identifizieren, bei denen Sicherheits-Scanner Probleme festgestellt haben. So kann Ihr Team diese Probleme mit geeigneten Maßnahmen umgehend beheben.

Abschließend empfehlen wir Ihnen, benutzerdefinierte Container zu signieren und so sicherzustellen, dass sie zwischen Build und Deployment nicht manipuliert werden.

Konfiguration, Sicherheit und Compliance von Deployments managen

Eine effektive Sicherung Ihrer Deployments umfasst sowohl die Sicherheit der Kubernetes-Plattform als auch die Automatisierung von Deployment-Richtlinien. Red Hat OpenShift umfasst folgende Funktionen out of the box:

1. Plattformkonfiguration und Lifecycle-Management
2. Identitäts- und Zugriffsmanagement
3. Sicherung von Plattformdaten und verknüpftem Storage
4. Deployment-Richtlinien

5. Plattformkonfiguration und Lifecycle-Management

Im Kubernetes-Sicherheitsaudit der Cloud Native Computing Foundation (CNCF), veröffentlicht im Sommer 2019, wurde festgestellt, dass die größte Sicherheitsbedrohung für Kubernetes die Komplexität ist, die das Konfigurieren und Härten der Kubernetes-Komponenten mit sich bringt. Red Hat OpenShift wird dieser Herausforderung durch Kubernetes Operators gerecht.

Ein Operator ist eine Methode zur Paketierung, Bereitstellung und Verwaltung einer Kubernetes-nativen Anwendung. Er dient als benutzerdefinierter Controller und kann die Kubernetes-API (Application Programming Interface) durch die anwendungsspezifische Logik erweitern, die für das Verwalten der Anwendung erforderlich ist. Jede Komponente der Red Hat OpenShift Plattform wird durch einen Operator geschützt, der die automatische Konfiguration, Überwachung und Verwaltung für OpenShift leistet. Einzelne Operatoren konfigurieren Komponenten wie den API-Server und das SDN (Software-Defined Network) direkt, während der Cluster Version Operator mehrere Operatoren in der gesamten Plattform managt. Mithilfe von Operatoren können Sie das Cluster-Management einschließlich Updates automatisieren – vom Kernel bis hin zu Services auf höheren Ebenen im Stack.

Zu den Hauptvorteilen einer Container-Plattform gehört, dass sie Self-Service-Funktionen für Entwickler ermöglichen. Dadurch können Ihre Entwicklungs-Teams Anwendungen, die auf genehmigten Schichten erstellt wurden, einfacher und schneller bereitstellen. Ein Self-Service-Portal gibt Ihren Teams ausreichende Steuerungsmöglichkeiten für die Zusammenarbeit, ohne die Sicherheit zu beeinträchtigen. Der Operator Lifecycle Manager (OLM) bietet den Nutzern von Red Hat OpenShift Clustern das Framework, mit denen sie Operatoren suchen und verwenden können, um die für die Aktivierung ihrer Anwendungen erforderlichen Services bereitzustellen. OLM ermöglicht Nutzern das Installieren, Upgraden und Zuweisen von Role-based Access Control für verfügbare Operatoren.

Zur Unterstützung der Compliance bietet Red Hat OpenShift den Compliance Operator, über den Sie die Compliance der Plattform mit technischen Kontrollen automatisieren können, die für entsprechende Frameworks erforderlich sind. Mit dem Compliance Operator können Red Hat OpenShift Administratoren den gewünschten Compliance-Zustand eines Clusters beschreiben und bietet ihnen einen Überblick über Lücken sowie Möglichkeiten, diese zu beheben. Dabei wird die Compliance aller Plattformschichten bewertet, einschließlich der Knoten, auf denen der Cluster ausgeführt wird. Mit dem File Integrity Operator können Sie ebenfalls regelmäßige Integritätsprüfungen auf den Cluster-Knoten ausführen.

6. Identitäts- und Zugriffsmanagement

Angesichts der Fülle an Kubernetes-Funktionen für Entwickler und Administratoren sind ein strenges Identitätsmanagement und RBAC zentrale Elemente der Container-Plattform. Die Kubernetes-APIs sind für die Automatisierung des Container-Managements in großem Umfang entscheidend. APIs werden beispielsweise genutzt, um Anfragen zu initiieren und zu validieren, darunter auch Konfigurations- und Deployment-Anfragen für Pods und Services.

Die Authentifizierung und Autorisierung von APIs ist ein wichtiger Aspekt für die Sicherung Ihrer Container-Plattform. Der API-Server ist ein wesentlicher Zugriffspunkt und sollte daher den strengsten Sicherheitskontrollen unterliegen. Die Control Plane von Red Hat OpenShift bietet integrierte Authentifizierung durch den Cluster Authentication Operator. Entwickler, Administratoren und Service-Accounts erhalten OAuth-Zugriffs-Tokens, mit denen sie sich gegenüber der API identifizieren können. Als Administrator können Sie die Identitätsanbieter Ihrer Wahl für das Cluster konfigurieren, damit Nutzer sich authentifizieren können, bevor sie Tokens erhalten. Es werden neun Identitätsanbieter unterstützt, darunter auch LDAP-Verzeichnisse (Lightweight Directory Access Protocol).

Präzise RBAC ist standardmäßig in Red Hat OpenShift aktiviert. RBAC-Objekte ermitteln, ob ein Nutzer berechtigt ist, eine bestimmte Aktion in einem Cluster auszuführen. Cluster-Administratoren können mithilfe von Cluster-Rollen und -Bindungen die Zugriffsstufen für OpenShift-Cluster sowie Projekte innerhalb der Cluster steuern.

7. Sicherung von Plattformdaten

Red Hat OpenShift härtet Kubernetes standardmäßig, um Daten bei der Übertragung zu sichern. Außerdem bietet es Möglichkeiten zur Sicherung von Daten im Ruhezustand.

So schützt Red Hat OpenShift Plattformdaten bei der Übertragung:

- HTTPS-Verschlüsselung von Daten bei der Übertragung für alle miteinander kommunizierenden Komponenten der Container-Plattform.
- Übermittlung jeglicher Kommunikation mit der Control Plane über TLS (Transport Layer Security).
- API-Server-Zugriffe nur über X.509-Zertifikate oder Tokens.
- Projekt-Quotas zur Begrenzung möglicher Schäden durch verdächtige Tokens.
- Konfiguration von etcd durch die eigene Zertifizierungsstelle und Zertifikate. (In Kubernetes speichert etcd den persistenten Masterzustand, während andere Komponenten auf Änderungen in etcd warten, um sich dann selbst in den definierten Zustand zu bringen.)
- Automatische Rotation von Plattformzertifikaten.

So schützt Red Hat OpenShift Plattformdaten im Ruhezustand:

- Optionale Verschlüsselung der Red Hat Enterprise Linux CoreOS-Disks und des etcd-Datastores für zusätzliche Sicherheit.
- Orientierung von Red Hat OpenShift an FIPS (Federal Information Processing Standards). FIPS 140-2 ist ein Sicherheitsstandard der US-Regierung für die Genehmigung kryptographischer Module. Wenn Red Hat Enterprise Linux CoreOS im FIPS-Modus gestartet wird, rufen die Plattformkomponenten von Red Hat OpenShift die kryptographischen Module von Red Hat Enterprise Linux auf.

Container sind sowohl für zustandslose als auch zustandsbehaftete Anwendungen nützlich. Red Hat OpenShift unterstützt flüchtigen und persistenten Storage. Der Schutz von verknüpftem Storage ist ein wichtiger Bestandteil bei der Sicherung von zustandsbehafteten Services. Red Hat OpenShift unterstützt mehrere Storage-Typen, darunter Network File System (NFS), Elastic Block Stores (EBS) von Amazon Web Services (AWS), Google Compute Engine (GCE) Persistent Disks, Azure Disk, iSCSI und Cinder.

Außerdem handelt es sich bei Red Hat OpenShift Container Storage um persistenten softwaredefinierten Storage, der für die Red Hat OpenShift Container Platform optimiert ist und sich in sie integrieren lässt. OpenShift Container Storage bietet hochgradig skalierbaren, persistenten Storage für cloudnative Anwendungen, für die Funktionen wie Verschlüsselung, Replikation und Verfügbarkeit in der gesamten Hybrid Multi-Cloud erforderlich sind.

- **Persistente Volumes (PVs)** lassen sich auf alle vom jeweiligen Ressourcenanbieter unterstützten Arten auf einen Host mounten. Die Funktionalitäten unterscheiden sich je nach Anbieter, und die Zugriffsmodi der PVs sind dabei auf die spezifischen Modi festgelegt, die das jeweilige Volume unterstützt. So kann NFS beispielsweise mehrere Lese- und Schreib-Clients unterstützen, aber ein spezifisches NFS-PV kann vielleicht nur im schreibgeschützten Modus auf den Server übertragen werden. Jedes PV erhält seine eigenen Zugriffsmodi, mit denen die Funktionalitäten dieses spezifischen PV beschrieben werden. Dazu gehören etwa ReadWriteOnce, ReadOnlyMany und ReadWriteMany.
- Bei **Shared Storage** (z. B. NFS, Ceph, Gluster) besteht die Kunst darin, die Gruppen-ID (GID) des Shared Storage-PV als Annotation auf der PV-Ressource zu registrieren. Wenn das PV vom Pod beansprucht wird, wird die annotierte GID zu den zusätzlichen Gruppen des Pods hinzugefügt und der Pod erhält Zugriff auf die Inhalte des Shared Storage.
- Bei **Block Storage** (z. B. EBS, GCE Persistent Disks, iSCSI) können Container-Plattformen auf SELinux-Funktionen zurückgreifen, um die Root des gemounteten Volumes für nicht privilegierte Pods zu sichern. Der Container, mit dem das Volume verbunden ist, wird dadurch zum alleinigen Eigentümer des Volumes und es ist auch nur für ihn sichtbar.

Sie sollten natürlich die Sicherheitsfunktionen nutzen, die in der von Ihnen gewählten Storage-Lösung verfügbar sind.

8. Automatisierung von richtlinienbasierten Deployments

Zu einer zuverlässigen Sicherheit gehören auch automatisierte Richtlinien, mit denen Sie das Container- und Cluster-Deployment aus sicherheitstechnischer Sicht managen können.

- Richtlinienbasiertes Container Deployment

Red Hat OpenShift Cluster können so konfiguriert werden, dass sie das Herunterladen von Images aus bestimmten Image Registries zulassen oder ablehnen. Zu den Best Practices für Produktions-Cluster gehört, nur das Deployment von Images aus Ihrer privaten Registry zuzulassen.

Red Hat OpenShift bietet Ihnen mit Security Context Constraints (SCCs) ein Plugin für die Zugangskontrolle, das eine Reihe von Bedingungen definiert, die ein Pod für die Zulassung in einem System ausführen muss. Mit **SCCs** können Sie Berechtigungen standardmäßig löschen – eine wichtige Option und weiterhin die bestbewährte Methode. So stellen die SCCs von Red Hat OpenShift auch sicher, dass standardmäßig kein privilegierter Container auf OpenShift-Worker-Knoten ausgeführt wird. Der Zugriff auf die Netzwerk- und Prozess-IDs des Hosts wird ebenfalls standardmäßig verweigert.

Nutzer mit den erforderlichen Berechtigungen können die Standardrichtlinien der SCCs bei Bedarf anpassen, wenn sie weniger strenge Einstellungen bevorzugen.

Red Hat Advanced Cluster Management for Kubernetes bietet Ihnen ein **erweitertes Application Lifecycle Management**, das mit offenen Standards Anwendungen bereitstellt, deren Platzierungsrichtlinien in vorhanden CI/CD-Pipelines und Governance-Kontrollen integriert sind.

- Richtlinienbasiertes Multi-Cluster-Management

Das Deployment mehrerer Cluster kann nützlich sein, um für die Hochverfügbarkeit von Anwendungen in mehreren Verfügbarkeitszonen zu sorgen oder um Funktionen für das zentrale Management von Deployments oder Migrationen über verschiedene Cloud-Anbieter hinweg zu ermöglichen, wie etwa Amazon Web Services (AWS), Google Cloud und Microsoft Azure. Beim Verwalten von mehreren Clustern müssen Ihre Orchestrierungs-Tools die erforderliche Sicherheit in den verschiedenen bereitgestellten Instanzen bieten. Wie immer sind Konfiguration, Authentifizierung und Autorisierung dabei wesentliche Bestandteile – ebenso wie die Möglichkeit, Anwendungsrichtlinien in mehreren Clustern zu managen und Daten sicher zu Ihren Anwendungen zu übermitteln, unabhängig davon, wo diese ausgeführt werden. Red Hat Advanced Cluster Management for Kubernetes bietet:

- **Multi-Cluster-Lifecycle-Management**, um Kubernetes-Cluster zuverlässig, konsistent und in großem Umfang zu erstellen, zu aktualisieren und zu löschen.
- **Richtliniengesteuertes GRC (Governance, Risk and Compliance)**, das Sicherheitskontrollen mithilfe von Richtlinien automatisch konfiguriert und ihre Konsistenz bewahrt, um Branchenstandards gerecht zu werden. Sie können auch eine Compliance-Richtlinie festlegen, die in einem oder mehreren gemanagten Clustern angewendet werden soll.

Aktive Anwendungen schützen

Die Anwendungssicherheit zu erhalten ist auch über die Infrastruktur hinaus entscheidend. Die Sicherung Ihrer containerisierten Anwendungen erfordert Folgendes:

1. Container-Isolierung
2. Anwendungs- und Netzwerkisolierung
3. Gesicherter Anwendungszugriff
4. Transparenz

9. Container-Isolierung

Um die Technologien für Container-Paketierung und -Orchestrierung optimal zu nutzen, braucht Ihr Operations-Team die richtige Umgebung für die Ausführung von Containern. Operations-Teams brauchen ein Betriebssystem, das Container in den Grenzbereichen sichert. Dabei wird der Host-Kernel vor Container Escapes und die Container voreinander geschützt.

Container sind Linux-Prozesse mit Isolierung und Ressourcenbeschränkung, mithilfe derer Sie Sandbox-Anwendungen auf einem gemeinsamen Host-Kernel ausführen können. Ihr Sicherheitsansatz für Container sollte derselbe Ansatz sein, nach dem Sie alle aktiven Prozesse auf Linux sichern.

Die NIST-Sonderveröffentlichung 800-190 empfiehlt für zusätzliche Sicherheit die Verwendung eines für Container optimierten Betriebssystems. Red Hat Enterprise Linux CoreOS, das zugrunde liegende Betriebssystem von Red Hat OpenShift, reduziert die Angriffsfläche, indem es die Host-Umgebung minimiert und für Container optimiert. Red Hat Enterprise Linux CoreOS enthält nur die Pakete, die für die Ausführung von Red Hat OpenShift nötig sind. Außerdem ist der Userspace schreibgeschützt. Die Plattform wird zusammen mit Red Hat OpenShift getestet, versioniert und bereitgestellt und durch einen Cluster gemanagt. Die Installation und Updates von Red Hat Enterprise Linux CoreOS sind automatisiert und stets mit dem Cluster kompatibel. Das Betriebssystem unterstützt die Infrastruktur Ihrer Wahl und bietet fast alle Möglichkeiten des Red Hat Enterprise Linux-Partnersystems.

Jeder Linux-Container, der auf einer Red Hat OpenShift Plattform ausgeführt wird, ist durch die leistungsstarken Sicherheitsfunktionen von Red Hat Enterprise Linux geschützt, die in die Red Hat OpenShift Knoten integriert sind. Die Sicherung der auf Red Hat Enterprise Linux ausgeführten Container erfolgt durch Linux-Namespaces, SELinux, cGroups, Capabilities und seccomp (Secure Computing Mode).

- Linux-Namespaces schaffen die Grundlagen für die Container-Isolierung. Für die Prozesse innerhalb eines Namespaces sieht es so aus, als hätten sie ihre eigene Instanz von globalen Ressourcen. Namespaces sorgen für eine Abstraktion, die den Eindruck erweckt, dass die Ausführung aus einem Container heraus auf Ihrem eigenen Betriebssystem erfolgt.
- SELinux bietet eine zusätzliche Sicherheitsschicht, um Container voneinander und vom Host zu isolieren. Administratoren können MACs (Mandatory Access Controls) für alle Nutzer, Anwendungen, Prozesse und Dateien erzwingen. SELinux ist wie eine Mauer, die Sie aufhält, wenn Sie (versehentlich oder absichtlich) aus der Namespace-Abstraktion ausbrechen. Es behebt Schwachstellen von Container Runtimes und kann durch entsprechende Konfigurationen verhindern, dass Container-Prozesse aus der Struktur ausbrechen.

- cGoups (Kontrollgruppen) begrenzen, verantworten und isolieren die Ressourcennutzung (z. B. CPU, Arbeitsspeicher, Disk-I/O, Netzwerk) einer Reihe von Prozessen. Mit cGroups können Sie verhindern, dass Ihre Container-Ressourcen von einem anderen Container auf demselben Host angezapft werden. Außerdem können Sie mit ihnen Pseudogeräte kontrollieren – eine gängige Angriffsmethode.
- Linux Capabilities können dafür genutzt werden, Privilegien in Container einzuschließen. Bei Capabilities handelt es sich um separate Berechtigungseinheiten, die getrennt voneinander aktiviert oder deaktiviert werden können. Sie können damit beispielsweise unverarbeitete IP-Pakete (Internet Protocol) übertragen oder diese an Ports bis 1024 binden. Bei der Ausführung von Containern lassen sich mehrere Capabilities ohne Auswirkungen auf die meisten containerisierten Anwendungen löschen.
- Und schließlich kann ein seccomp-Profil (Secure Computing Mode) einem Container zugeordnet werden, um verfügbare Systemaufrufe zu beschränken.

10. Anwendungs- und Netzwerkisolierung

Mandantenfähige Sicherheit ist für die unternehmensweite Nutzung von Kubernetes von wesentlicher Bedeutung. Mit Mandantenfähigkeit können verschiedene Teams denselben Cluster verwenden und dabei unberechtigte Zugriffe auf die jeweils anderen Umgebungen verhindern. Red Hat OpenShift unterstützt Mandantenfähigkeit durch eine Kombination aus Kernel-Namespaces, SELinux, RBAC, Kubernetes-(Projekt-)Namespaces und Netzwerkrichtlinien.

- **Red Hat OpenShift Projekte** sind Kubernetes-Namespaces mit SELinux-Annotationen. Projekte isolieren Anwendungen über Teams, Gruppen und Abteilungen hinweg. Mithilfe von lokalen Rollen und Bindungen kann gesteuert werden, wer Zugriff auf einzelne Projekte hat.
- **SCCs** ermöglichen Ihnen das standardmäßige Löschen von Berechtigungen – eine wichtige Option und weiterhin die bestbewährte Methode. So stellen die SCCs von Red Hat OpenShift auch sicher, dass standardmäßig kein privilegierter Container auf OpenShift-Worker-Knoten ausgeführt wird. Der Zugriff auf die Netzwerk- und Prozess-IDs des Hosts wird ebenfalls standardmäßig verweigert.

Das Deployment moderner, auf Microservices basierender Anwendungen in Containern bedeutet oft, dass mehrere Container bereitgestellt werden müssen, die auf verschiedene Knoten verteilt sind. Diese Microservices müssen sich erkennen und miteinander kommunizieren. Zur Abwehr von Netzwerkangriffen brauchen Sie eine Container-Plattform, mit der Sie den Datenverkehr für einen einzelnen Cluster segmentieren können, um so verschiedene Nutzer, Teams, Anwendungen und Umgebungen innerhalb des Clusters zu isolieren. Zusätzlich benötigen Sie Tools, mit denen Sie den externen Zugriff auf den Cluster und den Zugriff von Cluster-Services auf externe Komponenten managen. Zur Netzwerkisolierung sind die folgenden wichtigen Eigenschaften erforderlich:

- **Kontrolle des Ingress-Datenverkehrs:** Red Hat OpenShift bietet mit CoreDNS einen Service zur Namensauflösung für Pods. Der Red Hat OpenShift Router (HAProxy) unterstützt Ingress und Routen für den externen Zugriff auf Services, die auf dem Cluster ausgeführt werden. Beide unterstützen Wiederverschlüsselungs- und Durchgangsrichtlinien: Bei der Wiederverschlüsselung wird der HTTP-Datenverkehr bei der Weiterleitung entschlüsselt und verschlüsselt, während die Durchgangsrichtlinie den Verkehr zulässt, ohne TLS zu beenden.
- **Netzwerk-Namespaces:** Die erste Verteidigungslinie beim Netzwerkschutz bilden Netzwerk-Namespaces. Jede Ansammlung von Containern (auch als „Pod" bezeichnet) erhält eine eigene IP und einen eigenen Bereich zur Port-Bindung, wodurch die Pod-Netzwerke auf dem Knoten voneinander isoliert werden. Die Pod-IP-Adressen sind vom physischen Netzwerk unabhängig, mit dem die Red Hat OpenShift Knoten verbunden sind.

- **Netzwerkrichtlinien:** Die Red Hat OpenShift SDN verwendet Netzwerkrichtlinien für eine präzise Steuerung der Kommunikation zwischen Pods. Die Steuerung des Netzwerkverkehrs ist von jedem Pod aus zu jedem anderen Pod sowie auf bestimmten Ports und in bestimmte Richtungen möglich. Bei der Konfiguration von Netzwerkrichtlinien im mandantenfähigen Modus erhält jedes Projekt eine eigene virtuelle Netzwerk-ID, wodurch die Projektnetzwerke auf dem Knoten voneinander isoliert werden. Im mandantenfähigen Modus können Pods (standardmäßig) innerhalb eines Projekts miteinander kommunizieren. Pods von anderen Namespaces können aber von Pods oder Services eines anderen Projekts keine Pakete senden oder empfangen.
- **Kontrolle des Egress-Datenverkehrs:** Red Hat OpenShift ermöglicht Ihnen auch die Kontrolle des Egress-Datenverkehrs von Services, die auf dem Cluster ausgeführt werden. Dazu können Sie entweder Router- oder Firewall-Methoden verwenden. So können Sie beispielsweise mithilfe von IP-Whitelists Zugriff auf externe Datenbanken gewähren.

11. Gesicherter Anwendungszugriff

Zur Sicherung Ihrer Anwendungen gehört auch, die Authentifizierung und Autorisierung von Anwendungsnutzern und APIs zu managen.

- **Kontrolle des Nutzerzugriffs**

Web-SSO-Funktionen (Single Sign-On) sind ein zentraler Bestandteil moderner Anwendungen. Container-Plattformen bieten oft eine Reihe von containerisierten Services für Entwickler, die diese beim Erstellen von Anwendungen nutzen können. Red Hat Single Sign-On (RH SSO) ist ein vollständig unterstützter, sofort einsatzbereiter Web-SSO- und Föderations-Service, der auf dem Upstream-Projekt Keycloak basiert und Authentifizierung über SAML 2.0 (Security Assertion Markup Language) oder OpenID Connect ermöglicht. RH SSO bietet Client-Adapter für Red Hat Fuse und die Red Hat JBoss Enterprise Application Platform. Es ermöglicht Authentifizierung und Single Sign-On für Node.js-Anwendungen und kann in LDAP-basierte Directory Services wie Microsoft Active Directory und Red Hat Enterprise Linux Identity Management integriert werden. Red Hat Single Sign-On lässt sich auch mit Social Login-Anbietern wie Facebook, Google und Twitter integrieren.

- **Kontrolle des API-Zugriffs**

APIs sind für Anwendungen, die aus Microservices bestehen, entscheidend. Diese Anwendungen verfügen über unabhängige API-Services und erhöhen dadurch die Anzahl der Service-Endpunkte enorm, die zusätzliche Governance-Tools benötigen. Wir empfehlen Ihnen die Verwendung eines API-Management-Tools. Red Hat 3scale API Management stellt Ihnen verschiedene Standardoptionen für die Authentifizierung und Sicherung von APIs zur Verfügung, die einzeln oder in Kombination verwendet werden können, um Zugangsdaten auszugeben und den Zugriff zu kontrollieren.

Die in Red Hat 3scale API Management verfügbaren Funktionen für die Zugriffskontrolle gehen über grundlegende Sicherheits- und Authentifizierungsoptionen hinaus. Mithilfe von Anwendungs- und Accountplänen können Sie den Zugriff auf bestimmte Endpunkte, Methoden und Services einschränken und Zugriffsrichtlinien für Nutzergruppen anwenden. Anwendungspläne lassen Sie Rate Limits für die Nutzung von APIs festlegen und den Datenverkehr für Entwicklergruppen steuern. Dabei können Sie für eingehende API-Aufrufe Limits für bestimmte Zeiträume festlegen, um Ihre Infrastruktur zu schützen und für einen reibungslosen Datenverkehr zu sorgen. Außerdem können Sie für Anwendungen bei Erreichen oder Überschreiten der Rate Limits automatisch Warnungen auslösen und das Verhalten der Anwendungen bei Überschreitungen festlegen.

Sicherung des Anwendungsdatenverkehrs

Die Sicherung des Anwendungsdatenverkehrs mit Cluster-Ingress- und -Egress-Optionen wird in Abschnitt 10 dieses Dokuments behandelt. Für Microservice-Anwendungen ist die Sicherung des Datenverkehrs zwischen Services auf dem Cluster gleichermaßen wichtig. Für die Bereitstellung dieser Managementschicht kann ein Service Mesh verwendet werden. Der Begriff „Service Mesh" bezeichnet ein Netzwerk aus Microservices, die Anwendungen in einer verteilten Microservice-Architektur ausmachen, und die Interaktionen zwischen diesen Microservices.

Red Hat OpenShift Service Mesh basiert auf dem Open Source-Projekt Istio und erweitert vorhandene verteilte Anwendungen zur Verwaltung der Service-to-Service-Kommunikation um eine transparente Schicht, ohne dass Änderungen am Service-Code erforderlich sind. Der Lifecycle der Control Plane wird über einen mandantenfähigen Operator gemanagt, wodurch OpenShift Service Mesh auf Projektbasis genutzt werden kann. Darüber hinaus erfordert OpenShift Service Mesh keine RBAC-Ressourcen für Cluster.

Red Hat OpenShift Service Mesh bietet Discovery, Load Balancing und – für die Sicherheit entscheidend – Service-to-Service-Authentifizierung und -Verschlüsselung, Wiederherstellung bei Fehlern sowie Metriken und Monitoring.

3scale Istio Adapter ist ein optionaler Adapter, mit dem Sie Services markieren können, die in Red Hat OpenShift Service Mesh ausgeführt werden.

12. Transparenz

Die Möglichkeit, Red Hat OpenShift Cluster zu überwachen und zu prüfen, ist ein wichtiger Teil der Sicherung von Clustern und ihren Nutzern vor unangemessener Nutzung. Red Hat OpenShift umfasst integriertes Monitoring und Auditing sowie einen optionalen Protokollierungs-Stack.

OpenShift Container Platform-Services verbinden sich mit der integrierten Monitoring-Lösung, das auf Prometheus und seinem IT-Ökosystem basiert. Auch ein Dashboard für Warnungen ist verfügbar. Optional können Cluster-Administratoren das Monitoring für benutzerdefinierte Projekte aktivieren. Anwendungen, die auf Red Hat OpenShift bereitgestellt werden, lassen sich auch für die Nutzung der Cluster-Monitoring-Komponenten konfigurieren.

Das Auditing von Events ist eine bewährte Sicherheitspraktik und üblicherweise erforderlich, um regulatorische Frameworks einzuhalten. Red Hat OpenShift Auditing wurde im Grunde mithilfe eines cloudnativen Ansatzes entwickelt, um sowohl Zentralisierung als auch Resilienz zu bieten. Host-Auditing und Event-Auditing sind dabei standardmäßig auf allen Knoten aktiviert. Mit Red Hat OpenShift lassen sich die Verwaltung von Auditing-Daten sowie der Zugriff auf diese Daten äußerst flexibel konfigurieren. Sie können die Datenmenge steuern, die in den Auditprotokollen des API-Servers erfasst werden, indem Sie festlegen, welches Richtlinienprofil für das Auditprotokoll verwendet werden soll.

Monitoring-, Auditing- und Protokolldaten sind RBAC-geschützt. Projektdaten stehen Projektadministratoren zur Verfügung, während Cluster-Daten für Cluster-Administratoren verfügbar sind.

Als Best Practice wird empfohlen, ein Cluster so zu konfigurieren, dass alle Audit- und Protokollierungs-Events zur Integritätsverwaltung, Speicherung und Analyse an ein SIEM-System (Security Information and Event Management) weitergeleitet werden. Cluster-Administratoren können die Cluster-Protokollierung bereitstellen, um alle Protokolle zu aggregieren: Protokolle des Red Hat OpenShift Clusters (z. B. Host- und API-Auditprotokolle), Protokolle von Anwendungs-Containern sowie Infrastrukturprotokolle. Die Cluster-Protokollierung aggregiert diese Protokolle aus allen Cluster-Knoten und speichert sie in einem Standard-Protokollspeicher. Sie haben dann mehrere Optionen, um die Protokolle an das SIEM-System Ihrer Wahl weiterzuleiten.

Erweiterte Sicherheit durch ein robustes Partnernetzwerk

Sie können Sicherheits-Tools von Drittanbietern integrieren, um die Sicherheit für Ihre Container und Kubernetes zu erweitern oder bestehende Richtlinien zu erfüllen. Red Hat verfügt über ein umfassendes Netzwerk aus zertifizierten Partnern, die u. a. folgende Lösungen anbieten:

- Privileged Access Management (PAM)
- Externe Zertifizierungsstellen
- Externe Vaults und Schlüssel-Management-Lösungen
- Scanner für Container-Inhalte und Managementtools für Sicherheitsrisiken
- Analyse-Tools für Container Runtimes
- SIEM

Fazit

Beim Deployment von containerbasierten Anwendungen und Microservices geht es nicht nur um die Sicherheit. Ihre Container-Plattform muss für eine Umgebung sorgen, die Ihre Entwickler und Ihr Operations-Team unterstützt. Sie brauchen eine containerbasierte Anwendungsplattform, die sowohl sicherheitsorientiert als auch unternehmensgerecht ist. Sie sollte Entwickler und das Operations-Team unterstützen, ohne die erforderlichen Funktionen der jeweiligen Teams zu beeinträchtigen, und gleichzeitig die betriebliche Effizienz und die Infrastrukturnutzung optimieren.

Red Hat OpenShift baut auf einem Kern aus standardmäßigen und portierbaren Linux-Containern auf, die integrierte Sicherheitsfunktionen bereitstellen, darunter:

- Integrierte Build- und CI/CD-Tools für sichere DevOps-Praktiken
- Gehärtetes, unternehmensgerechtes Kubernetes mit integrierter Plattformkonfiguration, Compliance und Lifecycle-Management
- Strenges RBAC mit Integration in Authentifizierungssysteme von Unternehmen
- Möglichkeiten zur Verwaltung von Cluster-Ingress- und -Egress-Datenverkehr
- Integriertes SDN und Service Mesh mit Support für Netzwerk-Mikrosegmentierung
- Support für die Sicherung von Remote Storage Volumes
- Red Hat Enterprise Linux CoreOS, optimiert für die Ausführung von Containern in großem Umfang mit starker Isolierung
- Deployment-Richtlinien zur Automatisierung der Runtime-Sicherheit
- Integrierte Überwachung, Audits und Protokollierung

Red Hat OpenShift bietet außerdem die größte Sammlung unterstützter Programmiersprachen, Frameworks und Services (Abbildung 4). Red Hat Advanced Cluster Management for Kubernetes ermöglicht ein nahtlos integriertes Multi-Cluster-Management.

Red Hat OpenShift ist verfügbar für OpenStack, VMware, Bare Metal, AWS, Google Cloud Platform (GCP), Azure, IBM Cloud und Plattformen, die Red Hat Enterprise Linux unterstützen. Red Hat stellt Red Hat OpenShift Dedicated außerdem auf AWS und GCP als Public Cloud-Service zur Verfügung. Das Produkt wird gemeinsam von Microsoft und Red Hat angeboten. Red Hat OpenShift Service auf AWS wird gemeinsam von Red Hat und Amazon angeboten.

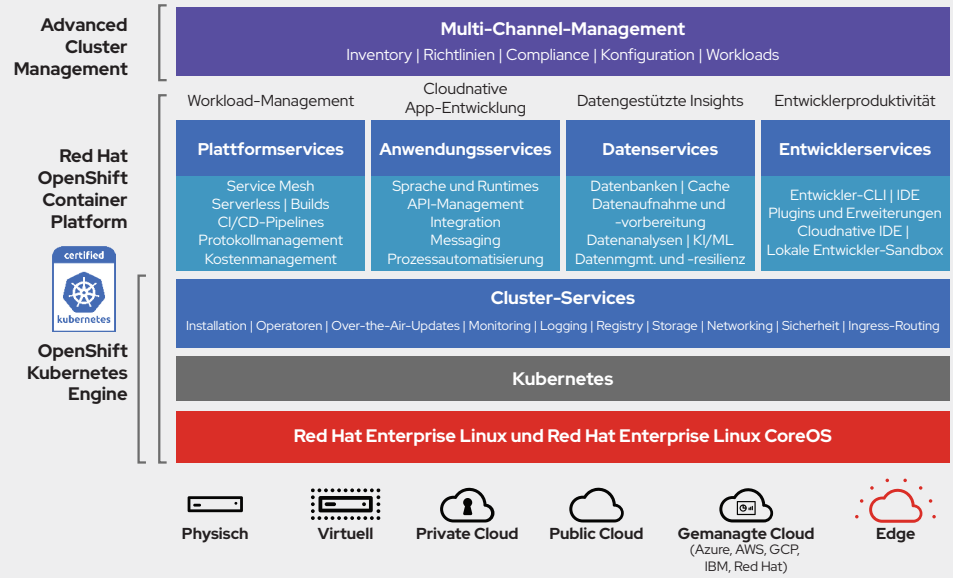

Abbildung 4. Red Hat OpenShift bietet einen Technologie-Stack mit umfangreichen Funktionen für physische und virtuelle sowie Private und Public Clouds.

Red Hat ist seit über zwei Jahrzehnten ein führender Anbieter vertrauenswürdiger Open Source-Lösungen für Unternehmenskunden und stellt diese Zuverlässigkeit und Sicherheit auch für Container bereit – durch Lösungen wie Red Hat OpenShift Container Platform, Red Hat Advanced Cluster Management for Kubernetes und unser containerfähiges Red Hat Produktportfolio.

ÜBER RED HAT

Red Hat, weltweit führender Anbieter von Open-Source-Software-Lösungen für Unternehmen, folgt einem community-basierten Ansatz, um zuverlässige und leistungsstarke Linux-, Hybrid Cloud-, Container- und Kubernetes-Technologien bereitzustellen. Red Hat unterstützt Kunden bei der Integration neuer und bestehender IT-Anwendungen, der Entwicklung cloudnativer Applikationen, der Standardisierung auf unserem branchenführenden Betriebssystem sowie der Automatisierung, Sicherung und Verwaltung komplexer Umgebungen. Dank der vielfach ausgezeichneten Support-, Trainings- und Consulting-Services ist Red Hat ein bewährter Partner der Fortune 500-Unternehmen. Als strategischer Partner von Cloud-Providern, Systemintegratoren, Applikationsanbietern, Kunden und Open Source Communities unterstützt Red Hat Unternehmen auf ihrem Weg in die digitale Zukunft.

facebook.com/redhatinc
@RedHatDACH
linkedin.com/company/red-hat

EUROPA, NAHOST, UND AFRIKA (EMEA)
00800 7334 2835
de.redhat.com
europe@redhat.com

TÜRKEI
00800 448820640

ISRAEL
1 809 449548

VAE
8000-4449549

de.redhat.com
#F26463_1220