

# Entwicklung und Deployment von Containern mit Lösungen von Red Hat und AWS



## Beschleunigte Container-Praktiken mit Red Hat und AWS

Die Konvergenz von Cloud-Technologieentwicklungen, Microservices und Marktstörungen haben zu einem Handlungsdruck bei der digitalen Transformation geführt. Unternehmen möchten ihre Prozesse beschleunigen, agiler und effizienter werden. Anwendungen sind der Schlüssel zum Erfolg dieser digitalen Initiativen, aber der Zeitfaktor ist von entscheidender Bedeutung. Heutzutage ist es gang und gäbe, dass Anwendungen innerhalb weniger Wochen bereitgestellt werden müssen.

Mit Anwendungs-Containern können Sie die Anwendungsbereitstellung beschleunigen. Das Container Image ist der Schlüssel, eine unveränderliche, statische Datei mit ausführbarem Code, sodass sie als isolierter Prozess in IT-Infrastruktur ausgeführt werden kann. Das Image besteht aus System-Libraries, Systemtools und anderen Plattformeinstellungen, die ein Softwareprogramm zur Ausführung auf einer Containerisierungsplattform benötigt. Das Image teilt den Betriebs-Kernel des jeweiligen Host-Rechners.

Container sorgen für Konsistenz in diesem Prozess, weil die Entwicklungs- und Deployment-Umgebungen identisch sind. Daraus resultiert ein viel schnellerer Releasezyklus, weil die Validierung in den einzelnen Phasen anhand einer standardmäßigen, unveränderlichen Darstellung der Anwendung erfolgt.

Linux®-Container sind für die meisten Digitalisierungsanforderungen ideal, weil sie von den Vorteilen der Prozessisolation in Linux sowie der Namespaces profitieren, um isolierte Prozesse zu erstellen. Red Hat® Enterprise Linux ist ein Standard für das Ausführen von Linux-Containern in Unternehmensumgebungen und bietet zahlreiche Optionen für das Erstellen von Containern. Mit diesen Optionen können Entwicklungsteams Container ganz einfach hochfahren. Allerdings müssen sie auch das Deployment und die Orchestrierung managen.

In diesem Dokument wird erläutert, wie Container-Tools von Red Hat und Entwicklungstools von Amazon Web Services (AWS) Entwicklungsteams beim Containerisieren und Bereitstellen von Anwendungen unterstützen.

## Die Vorteile von Linux-Anwendungs-Containern

Mit Linux-Anwendungs-Containern kann die IT Geschäftsanforderungen schneller gerecht werden. Entwicklungsteams segmentieren Anwendungen in modulare Microservices, die dann als Linux-Container bereitgestellt werden. Nach der Containerisierung haben diese Anwendungskomponenten kleine Footprints und können schneller bereitgestellt werden.

Linux-Container-Anwendungen können kinderleicht zwischen Architekturen portiert werden, weil sie von der Hardware und dem Betriebssystem abstrahiert werden. Mit Containern wird das Bereitstellen und Ausführen von Anwendungen in verschiedenen Umgebungen vereinfacht – vom Laptop eines Entwickelnden bis hin zu Produktions-Clustern. Außerdem werden Anwendungs-Upgrades und -Rollbacks von den Servern, auf denen sie ausgeführt werden, entkoppelt. Container bieten einen einfachen Betrieb, Modularität und Flexibilität, sodass der Aufwand für IT-Administrationsteams reduziert wird. Die Vorteile von Modularität und Flexibilität erhöhen sich, wenn Orchestrierungssysteme, wie Kubernetes oder Red Hat OpenShift, für das Container-Management verwendet werden.

Linux-Container können die Infrastrukturkosten senken. Aufgrund ihrer geringen Größe können Container problemlos in Bare Metal-Systeme sowie Public, Private, Hybrid und Multi Cloud-Umgebungen verschoben werden, in denen das gleiche Betriebssystem ausgeführt wird. Mit Containern können Sie die Entwicklung neuer Anwendungen beschleunigen, vorhandene Anwendungen optimieren und Anwendungen miteinander verbinden, vorausgesetzt, sie sind mit dem zugrunde liegenden Betriebssystem kompatibel.

Im Vergleich zu virtuellen Maschinen (VMs) eignen sich Container besser für:

- ▶ Entwickeln von cloudnativen Anwendungen
- ▶ Paketieren von Microservices
- ▶ Einführen von DevOps- oder CI/CD-Praktiken (Continuous Integration/Continuous Delivery)
- ▶ Verschieben von skalierbaren IT-Projekten zwischen unterschiedlichen IT-Umgebungen mit dem gleichen Betriebssystem

Im Gegensatz dazu können VMs mehr Vorgänge ausführen als ein einzelner Container. Aus diesem Grund werden sie traditionell für monolithische Workloads verwendet. Durch diese erweiterte Funktionalität können VMs jedoch aufgrund ihrer Abhängigkeit von Betriebssystem, Anwendung und Libraries schlechter portiert werden.

Im Vergleich zu Containern eignen sich VMs besser für:

- ▶ Housing traditioneller, Legacy- und monolithischer Workloads
- ▶ Isolieren riskanter Entwicklungszyklen
- ▶ Provisionieren von Infrastrukturressourcen, wie Netzwerke, Server und Daten
- ▶ Ausführen eines Betriebssystems unter einem anderen Betriebssystem (z. B. Unix unter Linux)

Wenn Sie die Anzahl der in einer Umgebung ausgeführten VMs und Betriebssysteme senken, werden dadurch die Lizenzierungsanforderungen erheblich reduziert. Durch Linux-Container können diese Kosten weiter gesenkt werden. Mit Tools und Plattformen von Red Hat und AWS können Sie cloudnative, containerisierte Anwendungen entwickeln sowie DevOps- und CI/CD-Praktiken einführen. So können Sie Ihre Agilität steigern und die Markteinführungszeiten verkürzen.

## **Entwicklung von Anwendungs-Containern unter Red Hat Enterprise Linux**

Mit den Lösungen von Red Hat können Sie Anwendungs-Container unter Red Hat Enterprise Linux entwickeln. Red Hat Enterprise Linux 8 unterstützt und enthält schlanke, auf offenen Standards basierende Container-Tools. Diese Tools sind auch mit den SDKs (Software Development Kits) und dem CDK (Cloud Development Kit) von AWS kompatibel, sodass Sie Unternehmensanwendungen entwickeln und testen können.

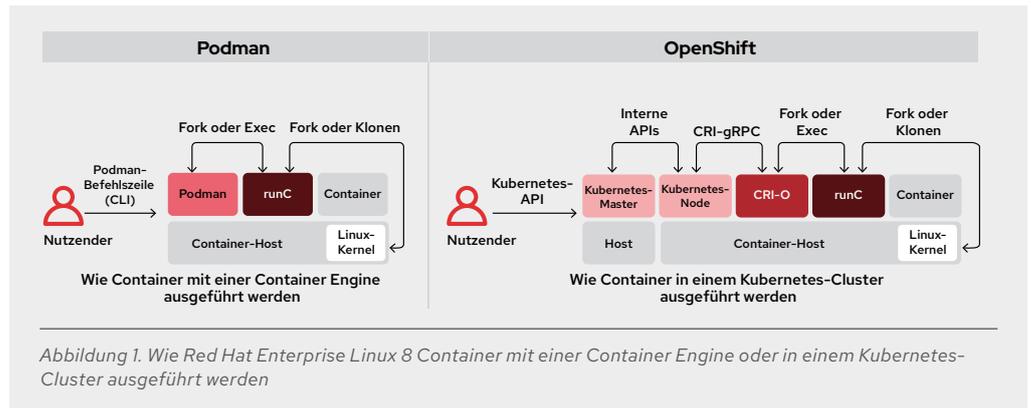
Nach der Entwicklung von Containern müssen sie bereitgestellt werden. Wenn Sie Kubernetes für die Orchestrierung verwenden, treten möglicherweise Probleme im Zusammenhang mit Komplexität, Skalierbarkeit, Service Discovery usw. auf, die Ihren Fortschritt behindern können. Red Hat OpenShift kann die potenziellen Herausforderungen bezüglich Kubernetes sowie Container-Deployment und -Management abstrahieren, damit Sie sich auf das Erstellen von Code konzentrieren können. Red Hat Enterprise Linux enthält flexible Tools, mit denen die Container-Entwicklung vereinfacht wird.

## **Red Hat Enterprise Linux 8 und Container-Entwicklung**

Red Hat Enterprise Linux vereinfacht die Container-Entwicklung mit weniger Repositories und mehr Entwicklungstools. Neben diesen Tools bietet Red Hat auch Basis-Images, die als Grundlage für eigene Images verwendet werden können. Einige dieser Basis-Images sind für Use Cases konzipiert, die von der Entwicklung von Unternehmensanwendungen mit Node.js, PHP, Java™ und Python bis hin zu Infrastrukturfunktionen wie Protokollierung, Datenerfassung und Authentifizierung reichen.

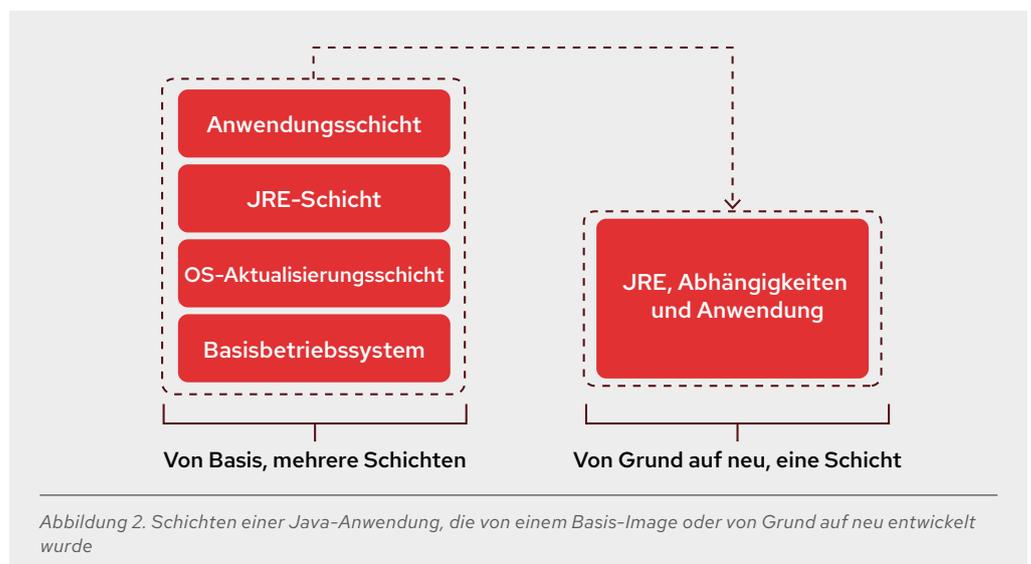
Red Hat Enterprise Linux 8 führt neue Funktionen für Befehlszeilentools wie Buildah, Podman, runC und Skopeo, das Universal Base Image (UBI), das Repository in Red Hat Quay und das ergänzende Repository ein, mit denen sich die Komplexität bei der Container-Entwicklung reduzieren lässt.

Die folgende Abbildung zeigt die Architektur eines Red Hat Enterprise Linux 8 Containers:



### Buildah

Mit **Buildah** können Sie einen Container von Grund auf neu erstellen, ohne einen Daemon oder Docker zu installieren. Ein guter Use Case für eine Entwicklung von Grund auf ist die Erwägung von Entwicklungs-Images gegenüber Staging- oder Produktions-Images einer Java-Anwendung. Bei der Entwicklung sind für das Container Image einer Java-Anwendung möglicherweise der Java-Compiler und Maven sowie weitere Tools erforderlich. In der Produktion benötigen Sie unter Umständen nur die Java Runtime und Ihre Pakete.



Da Buildah ohne Daemon auskommt, kann das Tool einfacher in einem Container ausgeführt werden, ohne dass eine spezielle Infrastruktur auf dem Host eingerichtet werden muss.

Buildah erfüllt nur die allgemeinen Anforderungen, die zum Erstellen oder Ändern OCI-kompatibler (Open Container Initiative) Linux-Container Images erforderlich sind. Daher kann es einfacher in vorhandene Build-Pipelines integriert werden. Buildah kann beispielsweise Container assemblieren, die keine Paketmanager (DNF/YUM) enthalten, wenn sie für das finale Image nicht erforderlich sind. Mit Buildah können Sie diese Container ganz einfach und mit konsistenter Sicherheit erstellen. Allerdings kann das Tool auch den Aufwand (und damit die Image-Größe) verringern und die Anpassung entsprechend Ihren Anforderungen in cloudnativen Anwendungen erweitern.

Ein weiterer wichtiger Use Case für Buildah ist das Schreiben von Bash-Skripten für das Erstellen Ihrer Images, ähnlich wie das Schreiben von Dockerfiles.

### **Podman**

Mit [Podman](#) können Sie Container ohne separaten Daemon verwalten. Das Tool ist auch mit der Docker-Befehlszeile (CLI) kompatibel. Über den Container-Runtime-Prozess runC ermöglicht Podman eine direkte Interaktion mit der Image Registry, mit dem Container und dem Image Storage sowie mit dem Linux-Kernel.

[Sie können Podman ohne Root-Berechtigungen](#) und ohne einen Daemon mit Root-Funktionen ausführen. Daher benötigt das Tool einen separaten Ort für das Schreiben von Images. Podman verwendet ein Repository im Home-Verzeichnis des Nutzens, um zu verhindern, dass `/var/lib/containers` global beschreibbar wird oder dass andere Praktiken angewendet werden, die möglicherweise zu Sicherheitsproblemen führen. Außerdem wird so dafür gesorgt, dass die Nutzenden jeweils über eigene Sets von Containern und Images verfügen und Podman gleichzeitig auf demselben Host nutzen können, ohne sich gegenseitig zu beeinträchtigen. Wenn die Arbeit abgeschlossen ist, können Änderungen in eine gemeinsame Registry übertragen und Images miteinander geteilt werden.

### **runC**

Buildah und Podman verwenden [runC, die OCI-Runtime](#), um Container zu starten. Sie können ein Image erstellen und ausführen oder aber Docker-formatierte Images mit runC ausführen. Dieses auf der Programmiersprache Go basierende Tool liest eine Runtime-Spezifikation, konfiguriert den Linux-Kernel und erstellt und startet die Container-Prozesse. Obwohl der Low-Level-Code von runC mit Docker geteilt werden kann, ist runC von keiner der Komponenten der Docker-Plattform abhängig. Das Tool unterstützt Linux-Namespaces, Live-Migration und verfügt über portierbare Performanceprofile. Außerdem werden Linux-Sicherheitsfunktionen wie beispielsweise SELinux, cgroups (Control Groups), seccomp (Secure Computing Mode) vollständig unterstützt.

### **Skopeo**

[Skopeo](#) prüft Images und transportiert sie an einen beliebigen Ort, an dem OCI-Images gespeichert werden können. Vor dem Release von Skopeo mussten Sie das gesamte Image abrufen, um es prüfen zu können, auch wenn Sie nur bestimmte Metadaten prüfen wollten. [In Skopeo werden sämtliche Image-Eigenschaften angezeigt](#), einschließlich der Schichten. So müssen Sie das Image nicht auf den Host herunterladen. Wenn sich bei der Prüfung ergibt, dass ein Container Image von einem Speicherort zu einem anderen kopiert werden muss, können Sie hierzu Skopeo verwenden.

Außerdem können Sie mit Skopeo ein Image aus einem Repository löschen und ein externes Image Repository mit einer internen Registry synchronisieren. So erreichen Sie mehr Sicherheit für Deployments mit einem getrennten Netzwerk (auch als Air Gap bezeichnet). Falls für das Repository erforderlich, kann Skopeo die entsprechenden Zugangsdaten und Zertifikate zur Authentifizierung übergeben.

## Red Hat Universal Base Image

Traditionell müssen Linux-Container für die jeweilige Zielplattform erstellt werden. Mit dem [Red Hat Universal Base Image \(UBI\)](#) wird dieses Problem behoben. Ein UBI ist ein unternehmensgerechtes Basis-Container-Image, mit dem Entwicklungsteams ihre Anwendungen erstellen und bereitstellen können. Für Red Hat Enterprise Linux 8 sind sämtliche Red Hat Basis-Images als UBIs verfügbar. Mit dem UBI können Sie Container Images basierend auf Red Hat Enterprise Linux ohne eine Red Hat Subskription erstellen und neu verteilen. Dabei benötigen auch die Nutzenden keine Subskription.

Das [UBI](#) basiert auf langjährig erprobten Red Hat Enterprise Linux Basis-Images und kann als Grundlage für in Containern entwickelte cloudnative und Webanwendungen verwendet werden. Dabei entfällt der zusätzliche Aufwand für das Erstellen CentOS-basierter Container Images für Community-Projekte oder für Nutzende, die Self-Support bevorzugen.

Sie können eine [containerisierte Anwendung mit einem UBI](#) erstellen, auf einem beliebigen Registry-Server bereitstellen, in die Produktion übertragen und mit anderen teilen. Da es nach Belieben weiter verteilt werden kann, können Sie es sogar auf Plattformen anderer Anbieter bereitstellen. Diese Funktion ist hilfreich, wenn Sie Ubuntu oder eine andere Linux-Plattform verwenden. So können Sie sich auch auf einer anderen Plattform mit Red Hat Enterprise Linux vertraut machen.

## Repository in Red Hat Quay

Sie können ein Image Repository auch in [Red Hat Quay](#) erstellen. Hierbei handelt es sich um eine private Container Registry, in der Container Images gespeichert, entwickelt und bereitgestellt werden. Das Tool analysiert Ihre Images auf Sicherheitschwachstellen und identifiziert potenzielle Probleme, um Sicherheitsrisiken zu mindern. Sie können ein Repository in Red Hat Quay auf 2 Arten erstellen: per Push (in Docker oder Podman) und über die Red Hat Quay-Benutzeroberfläche (UI). Darüber hinaus können Sie in Red Hat Quay auf öffentliche Repositories zugreifen.

## Ergänzendes Repository

Das [ergänzende Repository](#) enthält Pakete mit proprietären Lizenzen, die nicht in den Open Source-Repositories von Red Hat Enterprise Linux zu finden sind. Obwohl diese Pakete von Red Hat Enterprise Linux 8 nicht unterstützt werden, bieten sie flexiblen Zugriff auf andere Software als Bestandteil containerisierter Anwendungen.

## Red Hat Enterprise Linux auf Amazon Elastic Compute Cloud (EC2)

Red Hat Enterprise Linux auf Amazon Elastic Compute Cloud (Amazon EC2) ist die Kombination aus Amazon EC2 Compute und Red Hat Enterprise Linux. Red Hat übernimmt die Wartung der Red Hat Enterprise Linux Basis-Images für Amazon EC2. Updates werden empfangen, sobald sie bei Red Hat verfügbar sind. Die Zuverlässigkeit Ihrer Computing-Umgebung und die Sicherheit bleiben erhalten, und für Red Hat Enterprise Linux zertifizierte Anwendungen werden weiterhin unterstützt. Red Hat Enterprise Linux auf Amazon EC2 bietet eine virtuelle Entwicklungsumgebung und eine betriebssichere Plattform für die Bereitstellung zahlreicher Anwendungen, darunter auch containerisierte Anwendungen. Dank der verteilten Architektur können Sie Anwendungen flexibler und agiler bereitstellen.

Mit Red Hat Enterprise Linux auf EC2 haben Sie Zugriff auf die AWS-SDKs für gängige Sprachen, darunter Go (eine grundlegende Container-Sprache) sowie C++, Java und Rust, die allesamt zum Containerisieren von Anwendungen verwendet werden können. Außerdem können Sie das AWS-SDK mit Container-Bausteinen kombinieren, die in Red Hat Enterprise Linux enthalten sind. Darüber hinaus können Sie die Vorteile des Red Hat Universal Base Image (UBI) und von Anwendungs-Streams nutzen, um standortunabhängig containerisierte Anwendungen zu erstellen, zu teilen und zusammen daran zu arbeiten.

Wie Sie sehen, ist die Container-Entwicklung in Red Hat Enterprise Linux 8 eine sinnvolle Option für das Entwickeln, Erstellen und Verwalten von individuellen Containern und Container Images. Allerdings wird die Entwicklung und Bereitstellung von Containern mit [Red Hat OpenShift®](#) noch skalierbarer und effizienter.

## **Red Hat OpenShift on AWS: Container-Deployment und -Entwicklung kombinieren**

Wenn Sie cloudnative Anwendungen entwickeln, Microservices paketieren, DevOps- oder CI/CD-Praktiken verwenden oder skalierbare IT-Projekte innerhalb eines IT-Footprint mit verschiedenen Umgebungen mit denselben Betriebssystemen verschieben, sind Container ideal. Da sie nicht an bestimmte Betriebssysteme, Anwendungen und Libraries gebunden sind, sind sie portierbar und kompakt. Allerdings bringen sie auch Herausforderungen mit sich. Wenn Sie sich für die Verwendung von Containern entscheiden, müssen Sie sich damit beschäftigen, wie Sie sie skalieren, bereitstellen und ersetzen, wie Sie persistenten Storage verwalten und mit Service Discovery umgehen.

Wenn Sie nur eine Container Runtime verwenden, kann auch die Komplexität ein Hindernis darstellen. Bei einer Erweiterung der Operationen müssen Container hinzugefügt werden, was schnell zu einer Kette aus separaten Prozessen führt. Wenn dabei auf einem Server eine zu große Anzahl von Containern erstellt wird, kann die Performance negativ beeinträchtigt werden. Sie können auch schnell den Überblick über den Storage verlieren, der von Containern (Volumes) entkoppelt wird. Aus diesem Grund ist eine Orchestrierungsschicht für das Container-Deployment wichtig. Kubernetes ist dafür der geeignete Orchestrator. Allerdings kann der Versuch, eigenes natives Kubernetes bereitzustellen, komplex und problematisch sein. Sie haben die Möglichkeit, Red Hat Quay mit Kubernetes zu verwenden, um Ihre Images und die erforderlichen Konfigurationsdetails zu verwalten, um eine komplette Anwendung zum Laufen zu bringen. Allerdings beseitigt Red Hat OpenShift fast die gesamte Komplexität von Kubernetes, sodass Sie Anwendungen unter Kubernetes sehr schnell zum Laufen bringen können.

### **In Red Hat OpenShift verfügbare Entwicklungstools:**

- ▶ [CodeReady Workspaces](#) ermöglicht Remote-Entwicklungsteams das Provisionieren und Teilen von Umgebungen mit nur einem Klick für einen schnellen Einstieg und Interaktionen mit niedriger Latenz.
- ▶ Mit dem [Quarkus-Framework](#) können Sie Kubernetes-native Java-Anwendungen entwickeln.
- ▶ Preview-Support für [Buildpacks](#) und [Kaniko](#) sowie S2I- und Dockerfile-Builds über Buildah sind verfügbar.
- ▶ [Helm](#) vereinfacht das Arbeiten mit Diagrammen und Releases.

### **Container-Deployment auf Red Hat OpenShift**

[Red Hat OpenShift](#) wurde entwickelt, um Deployment und Management der Container-Plattform zu vereinfachen. Dank der Integration von Plattformüberwachung und automatisierten Wartungsabläufen und Upgrades ist es eine administratorfreundliche Kubernetes-basierte Container-Plattform für Unternehmen.

Red Hat OpenShift basiert auf dem „Write once, run anywhere“-Prinzip, das heißt, Sie können Workloads dort ausführen, wo es am sinnvollsten ist: On-Premise oder in Public, Private oder Hybrid Cloud-Umgebungen. Red Hat OpenShift ist auf AWS als ein weiterer Service integriert und paketierte, der in der AWS-Konsole aufgelistet ist, damit Sie OpenShift-Cluster mit weniger Zeit- und Arbeitsaufwand erstellen und konsistent verwalten können. Außerdem erhalten Sie eine On-Demand-Abrechnung, eine einzige Rechnung und die Option, AWS für Support zu kontaktieren. Das [Deployment eines Red Hat OpenShift Clusters](#) auf AWS kann mit Standardeinstellungen oder benutzerdefinierten AWS-Einstellungen erfolgen. Darüber hinaus können Sie einen Cluster auf AWS-Infrastruktur bereitstellen, die Sie selbst provisioniert haben. Sie können die bereitgestellten AWS CloudFormation-Vorlagen an Ihre Anforderungen anpassen.

### **Container-Entwicklung auf Red Hat OpenShift**

Red Hat OpenShift enthält das S2I-Framework (Source-to-Image), mit dem Sie von Anwendungscode zu Containern wechseln können. So wird die Komplexität bei der Container-Entwicklung erheblich reduziert. Sie können einfach Images schreiben, bei denen der Quellcode der Anwendung als Eingabe verwendet wird, und ein neues Image als Ausgabe erzeugen, das die assemblierte Anwendung ausführt. Die Entwicklung in OpenShift kann auf 2 Arten erfolgen: über Dockerfiles und über S2I.

Von Nutzenden bereitgestellte Dockerfiles rufen Buildah auf, das im OpenShift Builder enthalten ist, um das Image zu erstellen. Das resultierende Image wird auch von Buildah getaggt und übertragen. Alternativ können Sie Dockerfiles mit S2I generieren. Dabei wird derselbe Ablauf wie bei der Buildah-basierten Dockerfile-Erstellung verwendet. Das resultierende Builder Image enthält die speziellen Informationen, die zum Erzeugen dieses ausführbaren Images (auch als Build-Artefakt bezeichnet) erforderlich sind.

Wie S2I zeigt, ist Red Hat OpenShift nicht nur eine Plattform für die Entwicklung, sondern auch für das Deployment von Containern. Entwicklungsteams können auf einer Kubernetes-Plattform entwickeln, statt entsprechend den Besonderheiten vorhandener AWS-Infrastrukturen Code zu schreiben. Darüber hinaus können Sie in Red Hat OpenShift viele verschiedene Ansätze für die Anwendungsentwicklung mit Containern verwenden, sodass Sie den richtigen Ansatz für verschiedene Situationen nutzen können. Dabei werden die Anforderungen von Entwicklerinnen und Entwicklern erfüllt, die nicht mit Kubernetes vertraut sind und nur Code schreiben möchten, sowie die Anforderungen von Kubernetes-Profis, die maximale Flexibilität wünschen.

Red Hat OpenShift unterstützt die Sprachen, Datenbanken und Tools, die Entwicklungsteams bereits nutzen. Außerdem wird der Zugriff auf Anwendungsservices vereinfacht, wie diejenigen, die von AWS angeboten werden.

#### Weitere Entwicklungstools in Red Hat OpenShift:

- ▶ **Odo** bietet Entwicklungsteams die Möglichkeit, Code mit der CLI zu iterieren. Es unterstützt Kubernetes und Red Hat OpenShift, ein offenes Modell für Tools über eine Standarddefinition sowie schnelle, iterative Java-Entwicklung mit Quarkus.
- ▶ Die Serverless-Unterstützung der **Knative**-Komponenten „Serving“ und „Eventing“ ermöglicht Entwicklungsteams das Erstellen von Serverless- und eventgesteuerten Anwendungen, darunter **Strimzi** (für die Ausführung von Apache Kafka auf OpenShift) und **Service Mesh**.
- ▶ **Tekton** in Red Hat OpenShift Pipelines, OpenShift-Plugins für GitHub Actions, Jenkins und die GitLab Runner-Unterstützung bieten kontinuierliche Integrationsfunktionen.

### AWS-Entwicklungstools für Container auf Red Hat OpenShift

Die Container-Entwicklung in Red Hat OpenShift wird durch den Einsatz von AWS-Tools sogar noch einfacher. Diese Tools können auch auf anderen Plattformen als Red Hat OpenShift verwendet werden, einschließlich Red Hat Enterprise Linux.

**AWS CodeArtifact** ist ein vollständig gemanagter Repository-Service für Artefakte, mit dem Organisationen unterschiedlicher Größe Softwarepakete, die beim Softwareentwicklungsprozess verwendet werden, sicher speichern, veröffentlichen und gemeinsam nutzen können.

**AWS Cloud9** ist eine integrierte Entwicklungsumgebung (Integrated Development Environment, IDE) zum Schreiben, Ausführen und Debuggen von Code in einer browserbasierten Shell, die das Installieren zusätzlicher Software, das Ausführen eines Git Pushes und das Eingeben von Befehlen ermöglicht.

Das **AWS Cloud Development Kit** verwendet vertraute und leistungsstarke Programmiersprachen zur Modellierung von Anwendungen. Dabei wird der Einstieg in AWS beschleunigt, weil keine neuen Kompetenzen erlernt werden müssen.

**Amazon CodeGuru** bietet Visualisierungen und Empfehlungen zur Behebung von Performance-Problemen sowie zu den voraussichtlichen Kosten für die Ausführung von ineffizientem Code, um Entwicklungsteams bei der Priorisierung der Problembehebung zu unterstützen. Es lässt sich in Ihren bestehenden Softwareentwicklungs-Workflow in Red Hat OpenShift integrieren und erteilt intelligente Vorschläge zur Verbesserung der Codequalität und zur Erkennung der kostspieligsten Codezeilen einer Anwendung.

### Schneller Einstieg in Entwicklung, Bereitstellung und Verwaltung von Containern

Die Entwicklung von Anwendungs-Containern muss keine Herausforderung darstellen. Die Container-Tools von Red Hat und die Entwicklungstools von AWS können Sie dabei unterstützen. Die Bereitstellung und Orchestrierung von Containern – die letzten Schritte der Container-Implementierung – sind oft der anspruchsvollste Teil des Prozesses. Red Hat OpenShift beseitigt den Großteil der Komplexität von Kubernetes, sodass Sie Container in großem Umfang bereitstellen und verwalten können, unabhängig davon, wie sie entwickelt wurden.

Der Vorteil von Red Hat OpenShift besteht darin, dass es nicht nur eine Deployment-Lösung ist. Entwicklungsteams können aus zahlreichen Optionen für die Entwicklung ihrer Container auf der Plattform und das anschließende Deployment wählen. Red Hat OpenShift ist als selbst gemanagte und als vollständig gemanagte Version auf AWS verfügbar. Mit Red Hat OpenShift können Sie Ihre eigenen OpenShift-Implementierungen auf AWS bereitstellen und verwalten. Red Hat OpenShift Service on AWS und Red Hat OpenShift Dedicated können als vollständig gemanagte Red Hat Cloud Services genutzt werden, damit Sie sich auf die Entwicklung und Markteinführung von Anwendungen konzentrieren können.

Unabhängig davon, wie Sie Ihre Container entwickeln oder bereitstellen, mit den Lösungen von Red Hat und AWS lässt sich der Prozess optimieren und beschleunigen. So können Sie die Anwendungen, die Ihr Unternehmen braucht, in kurzer Zeit entwickeln und bereitstellen.

Erfahren Sie mehr darüber, wie die [Partnerschaft zwischen Red Hat und AWS](#) Sie beim Aufbau einer [modernen Infrastruktur mit Red Hat Enterprise Linux](#) unterstützt, oder über folgende Themen:

- ▶ [Dokumentation zu Red Hat Enterprise Linux](#)
- ▶ [Podman](#)
- ▶ [Entwicklungstools auf AWS](#)



## Über Red Hat

Red Hat, weltweit führender Anbieter von Open Source-Softwarelösungen für Unternehmen, folgt einem communitybasierten Ansatz, um zuverlässige und leistungsstarke Linux-, Hybrid Cloud-, Container- und Kubernetes-Technologien bereitzustellen. Red Hat unterstützt Kunden bei der Entwicklung cloudnativer Applikationen, der Integration neuer und bestehender IT-Anwendungen sowie der Automatisierung, Sicherung und Verwaltung komplexer Umgebungen. [Als bewährter Partner der Fortune 500](#)-Unternehmen stellt Red Hat [vielfach ausgezeichnete](#) Support-, Trainings- und Consulting-Services bereit, die jeder Branche die Vorteile der Innovation mit Open Source erschließen können. Als Mittelpunkt eines globalen Netzwerks aus Unternehmen, Partnern und Communities unterstützt Red Hat Unternehmen bei der Steigerung ihres Wachstums und auf ihrem Weg in die digitale Zukunft.

**f** [facebook.com/redhatinc](https://facebook.com/redhatinc)  
**t** [@RedHatDACH](https://twitter.com/RedHatDACH)  
**in** [linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)

[de.redhat.com](https://de.redhat.com)  
 #F30136\_1021

### EUROPA, NAHOST UND AFRIKA (EMEA)

00800 7334 2835

[de.redhat.com](https://de.redhat.com)

[europe@redhat.com](mailto:europe@redhat.com)

### TÜRKEI

00800 448820640

### ISRAEL

1 809 449548

### VAE

8000-4449549

Copyright © 2021 Red Hat, Inc. Red Hat, das Red Hat Logo und OpenShift sind Marken oder eingetragene Marken von Red Hat, Inc. oder dessen Tochterunternehmen in den USA und anderen Ländern. Linux® ist eine in den USA und anderen Ländern eingetragene Marke von Linus Torvalds. Alle anderen in diesem Dokument genannten Marken sind Eigentum der jeweiligen Inhaber. Java und alle Java-basierten Marken sowie Logos sind in den USA und anderen Ländern Marken oder eingetragene Marken von Oracle America, Inc. Bei Platzmangel verwenden Sie stattdessen „Java ist eine Marke von Oracle America, Inc.“