

# Sviluppo e deployment di container con le soluzioni Red Hat e AWS



## **Red Hat e AWS accelerano la strategia container**

La concomitanza di sviluppi nella tecnologia cloud, microservizi e interruzioni del mercato ha suscitato un senso di urgenza nell'ambito della trasformazione digitale. Le aziende vogliono essere più agili e scattanti, e le applicazioni, così come il fattore tempo, giocano un ruolo fondamentale per il successo delle iniziative digitali, considerando, ad esempio, che le applicazioni vengono sviluppate in appena poche settimane.

I container accelerano il deployment delle applicazioni grazie alle immagini: file statici e non modificabili che includono codice eseguibile in modo che possano funzionare autonomamente in un'altra infrastruttura. Ogni immagine è costituita da librerie, strumenti di sistema e altre impostazioni, necessarie per l'esecuzione del programma nella piattaforma in cui verrà ospitato. L'immagine condivide lo stesso kernel operativo del computer host.

Dato che gli ambienti di sviluppo e deployment sono identici, i container garantiscono inoltre la massima coerenza, circostanza che concorre ad accelerare il ciclo di release, poiché ogni passaggio viene convalidato a fronte di una rappresentazione standard e immutabile dell'applicazione.

I container Linux® sono ideali per la gran parte delle necessità correlate alla trasformazione digitale, perché approfittano dei vantaggi tipici di Linux e dello spazio dei nomi per creare processi isolati. Red Hat® Enterprise Linux è uno standard per l'esecuzione in ambienti aziendali di container Linux e offre numerose opzioni per crearli, gestirli, distribuirli e orchestrarli con semplicità.

Questo documento illustra agli sviluppatori come utilizzare gli strumenti per container Red Hat e Amazon Web Services (AWS) per containerizzare e distribuire le applicazioni.

## **Vantaggi della containerizzazione delle applicazioni Linux**

I container di applicazioni Linux consentono di raggiungere più rapidamente gli obiettivi di business IT. Gli sviluppatori suddividono le applicazioni in microservizi modulari, che vengono poi distribuiti come container Linux. Dopo la containerizzazione, i diversi componenti delle applicazioni risultano meno ingombranti e possono essere distribuiti più in fretta.

I container Linux sono architetture estremamente portatili perché sono indipendenti dall'hardware e dal sistema operativo, ecco perché semplificano il processo di deployment ed esecuzione delle applicazioni nei diversi ambienti e possono essere trasferiti direttamente dal portatile dello sviluppatore fino ai cluster di produzione. Inoltre disaccoppiano rollback e upgrade delle applicazioni dal server su cui sono in esecuzione. In definitiva, i container rendono più agile la gestione per gli amministratori IT, che possono approfittare di semplicità operativa, nonché di modularità e flessibilità, vantaggi che aumentano con l'utilizzo di strumenti di gestione dei container come Kubernetes o Red Hat OpenShift.

I container Linux riducono i costi dell'infrastruttura, perché di natura sono leggeri e di poco ingombro, possono quindi essere facilmente spostati tra sistemi bare metal e ambienti di cloud pubblico, privato, ibrido e multicloud con lo stesso sistema operativo. I container accelerano inoltre lo sviluppo di nuove app, l'ottimizzazione di quelle esistenti e la loro connessione, a patto che siano compatibili con il sistema operativo sottostante.

Rispetto alle VM, i container sono indicati per:

- ▶ Creare app cloud native.
- ▶ Raggruppare microservizi.
- ▶ Implementare processi e pipeline di integrazione e distribuzione continua (CI/CD) e DevOps.
- ▶ Spostare progetti IT scalabili in un contesto diverso che condivide lo stesso sistema operativo.

Le VM, invece, sono in grado di eseguire più operazioni di un singolo container, per questo vengono tradizionalmente usate per carichi di lavoro monolitici. Tuttavia, questa funzionalità estesa e la dipendenza da sistema operativo, applicazione e librerie ne limita fortemente la portabilità.

Rispetto ai container, le VM sono indicate per:

- ▶ Ospitare carichi di lavoro tradizionali, preesistenti e monolitici.
- ▶ Isolare cicli di sviluppo ad alto rischio.
- ▶ Eseguire il provisioning delle risorse dell'infrastruttura quali reti, server e dati.
- ▶ Eseguire un sistema operativo diverso in un altro sistema operativo, ad esempio Unix su Linux.

Meno VM e sistemi operativi si eseguono in un ambiente, meno licenze serviranno. Ecco un altro vantaggio economico dei container Linux. Strumenti e piattaforme Red Hat e AWS consentono di realizzare applicazioni containerizzate cloud native e di adottare pratiche DevOps e CI/CD per aumentare l'agilità e ridurre i tempi di rilascio.

### **Sviluppo dei container di applicazioni su Red Hat Enterprise Linux**

Grazie alle soluzioni Red Hat è possibile realizzare container di applicazioni su Red Hat Enterprise Linux. Red Hat Enterprise Linux 8 include e supporta una serie di strumenti leggeri e basati su standard open, che interagiscono con Software Development Kit (SDK) e Cloud Development Kit (CDK) AWS, per realizzare e testare applicazioni aziendali.

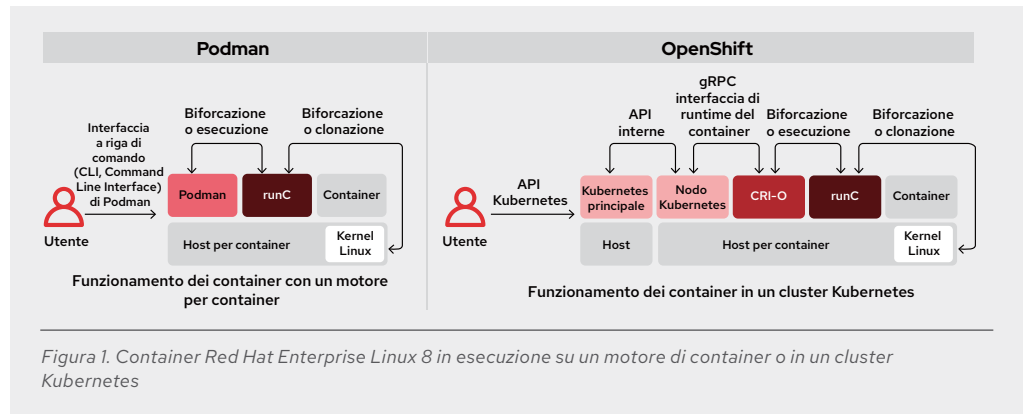
Dopo aver creato i container, è necessario distribuirli. L'impiego di Kubernetes per l'orchestrazione può causare problemi relativi a complessità, scalabilità, individuazione dei servizi e di altro tipo che possono impedire il corretto svolgimento del processo. Red Hat OpenShift invece è progettato appositamente per individuare e risolvere i potenziali problemi relativi a Kubernetes, nonché a deployment e gestione dei container, per lasciare più spazio alla creazione del codice. In particolare, per Red Hat Enterprise Linux, sono disponibili pochi, semplici strumenti per accelerare lo sviluppo dei container.

### **Sviluppo dei container e Red Hat Enterprise Linux 8**

Red Hat Enterprise Linux semplifica lo sviluppo dei container riducendo il numero di repository e fornendo più strumenti agli sviluppatori. Oltre a questi strumenti, Red Hat fornisce immagini di base sulle quali realizzare le tue immagini personali. Alcune di queste immagini di base sono ideate per coprire casi di utilizzo come build di applicazioni aziendali che utilizzano Node.js, PHP, Java™ e Python fino a funzioni per l'infrastruttura come registrazione, raccolta dati e autenticazione.

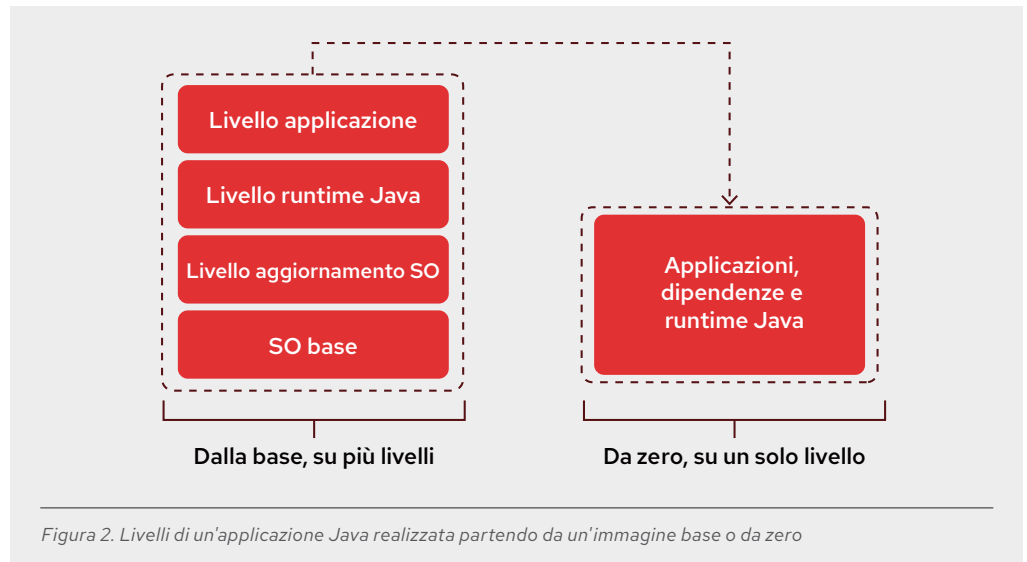
Red Hat Enterprise Linux 8 introduce nuove funzionalità per strumenti da riga di comando come Buildah, Podman, runC e Skopeo, le Red Hat Universal Base Image (UBI), il repository in Red Hat Quay e il repository Supplemental, tutti componenti che riducono le complessità legate allo sviluppo dei container.

In questo schema viene illustrata l'architettura di un container Red Hat Enterprise Linux 8:



## Buildah

**Buildah** consente di compilare container da zero senza dover installare daemon o Docker. Un container può essere realizzato da zero per lo sviluppo delle immagini, oppure per lo staging o la produzione di immagini di un'applicazione Java. Durante lo sviluppo, per l'immagine container di un'applicazione Java potrebbe essere necessario anche un compiler Java, Maven o altri strumenti, ma in produzione dovrebbero bastare il runtime Java e i pacchetti.



Data la natura daemonless di Buildah, eseguirlo in un container senza impostare un'infrastruttura speciale sull'host è più semplice.

Buildah fornisce solo i requisiti di base necessari per creare o modificare immagini container di Linux che siano compatibili con gli standard Open Container Initiative (OCI) per semplificare l'integrazione nella pipeline delle build. Ad esempio, è in grado di assemblare container che non includano gestori di pacchetti (DNF/YUM), se non necessari per l'immagine finale. I container vengono realizzati con semplicità e applicando la protezione in modo coerente, si riduce l'overhead (e di conseguenza la dimensione dell'immagine), mentre si possono personalizzare anche le applicazioni cloud native.

Buildah viene utilizzato principalmente per scrivere script Bash per la creazione di immagini, come se fosse un Dockerfile.

### **Podman**

[Podman](#) consente di gestire i container senza un daemon separato ed è compatibile con l'interfaccia a riga di comando (CLI) Docker. Offre inoltre l'interazione diretta con il registro delle immagini, nonché con lo storage di container e immagini e con il kernel Linux, grazie al processo di runtime per container runC.

[Podman si può eseguire senza privilegi root](#) e senza un daemon che fornisca queste capacità, per cui serve uno spazio separato in cui scriva le immagini. Per evitare di rendere modificabili `/var/lib/containers` o applicare altre procedure che possono esporre a rischi di sicurezza, Podman utilizza un repository nella directory home dell'utente. Ciò garantisce inoltre che ogni utente disponga di un set di container e immagini distinto e che possa utilizzare simultaneamente Podman sullo stesso host, senza interferenze con gli altri. Al termine del loro lavoro, gli utenti possono inviare le modifiche apportate a un registro comune per condividere le immagini con altri.

### **runC**

Per l'avvio dei container, Podman e Buildah utilizzano [runC](#), il runtime OCI predefinito. Puoi creare ed eseguire un'immagine oppure eseguire immagini formattate in docker con runC. Questo strumento in linguaggio Go legge una specifica di runtime, configura il kernel Linux e quindi crea e avvia processi di container. Sebbene il suo codice di base possa essere condiviso con Docker, runC non dipende da alcun componente di questa piattaforma. Supporta spazi dei nomi Linux, migrazione in tempo reale e dispone di profili per prestazioni portatili. Fornisce inoltre supporto completo per le funzionalità di sicurezza di Linux come SELinux, gruppi di controllo (cgroups), profili di elaborazione sicura (seccomp) e altri.

### **Skopeo**

[Skopeo](#) controlla le immagini e le sposta ovunque possano essere archiviate in conformità agli standard OCI. Prima del suo rilascio, per ispezionare un'immagine era necessario estrarla per intero anche se occorreva ispezionare solo alcuni metadati. [Skopeo mostra le proprietà dell'immagine](#), inclusi i livelli, evitando così di eseguire il pull dell'immagine nell'host. Se l'ispezione segnala la necessità di copiare un'immagine container da un tipo di posizione a un altro, è possibile utilizzare Skopeo.

Skopeo consente anche di eliminare un'immagine da un repository e di sincronizzare il repository di un'immagine esterna con un registro interno, per i deployment di reti disconnesse. Quando richiesto dal repository, Skopeo può comunicare le credenziali e i certificati appropriati per l'autenticazione.

### **Red Hat Universal Base Image**

In genere i container Linux dovevano essere realizzati in base alla piattaforma di destinazione che li avrebbe ospitati, ma con [Red Hat Universal Base Image \(UBI\)](#) non è più necessario. Un'UBI è un'immagine container di base di livello enterprise a partire dalla quale gli sviluppatori possono realizzare e distribuire le proprie applicazioni. Per Red Hat Enterprise Linux 8, tutte le immagini di base Red Hat sono UBI. Partendo da questi modelli è possibile realizzare e redistribuire le immagini dei container basate su Red Hat Enterprise Linux senza alcuna sottoscrizione Red Hat, neanche per gli utenti.

Partendo da immagini di base Red Hat Enterprise Linux testate a lungo, le [immagini UBI sono progettate](#) per fungere da fondamenta in scenari di utilizzo per applicazioni web e cloud native sviluppati in container e per eliminare la necessità di creare immagini container basate su CentOS per progetti della community o per quelli in cui si preferisce un supporto autonomo.

Puoi creare un'[applicazione containerizzata in una UBI](#), inviarla alla tua selezione di server di registro, passarla in produzione e condividerla con altri con semplicità. Trattandosi di una soluzione distribuibile liberamente, può essere anche introdotta su piattaforme di altri vendor, capacità che si rende utile se si sta utilizzando Ubuntu o un'altra piattaforma Linux, e che consente di familiarizzare con Red Hat Enterprise Linux, anche se si usa una piattaforma diversa.

### **Repository su Red Hat Quay**

Puoi creare repository di immagini anche su [Red Hat Quay](#), un registro di container privato che offre archiviazione, creazione e deployment di immagini di container. Analizza le immagini alla ricerca di potenziali problemi, che riduce prima che possano trasformarsi in rischi per la sicurezza. Per creare un repository su Red Hat Quay eseguendone il push (da Docker o Podman) e utilizzando l'interfaccia utente di Red Hat Quay stesso. Inoltre, su Red Hat Quay è possibile accedere a repository pubblici.

### **Ulteriori repository**

Sono disponibili [ulteriori repository](#) compresi i pacchetti con licenza proprietaria non inclusi nei repository Red Hat Enterprise Linux open source. Sebbene non siano supportati da Red Hat Enterprise Linux 8, questi pacchetti offrono la flessibilità di accedere ad altro software come parte di applicazioni containerizzate.

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

Red Hat Enterprise Linux on Amazon Elastic Compute Cloud (Amazon EC2) unisce le capacità di elaborazione di Amazon EC2 e Red Hat Enterprise Linux. Red Hat mantiene le immagini Red Hat Enterprise Linux di base per Amazon EC2 e rende immediatamente disponibili gli aggiornamenti. L'ambiente di elaborazione rimane affidabile e protetto e le app certificate Red Hat Enterprise Linux sono comunque supportate. Amazon EC2 esegue Red Hat Enterprise Linux e offre un ambiente di sviluppo virtuale e una piattaforma affidabile compatibile con una vasta gamma di applicazioni, comprese quelle containerizzate. L'architettura distribuita consente di eseguire il deployment di applicazioni in modo più agile e flessibile.

Red Hat Enterprise Linux on EC2 consente di accedere agli SDK AWS per utilizzare i linguaggi comuni, come Go, considerato il principale mezzo di comunicazione per i container, C++, Java e Rust, tutti impiegabili per le applicazioni containerizzate. Puoi anche aggiungere l'SDK agli elementi fondanti dei container inclusi in Red Hat Enterprise Linux. Puoi inoltre approfittare di UBI e flussi di applicazioni per realizzare, condividere e collaborare sulle applicazioni containerizzate dove preferisci.

Red Hat Enterprise Linux 8 è dunque un'opzione valida per lo sviluppo, la realizzazione e la gestione di container singoli, come di immagini. E l'utilizzo di [Red Hat OpenShift®](#) rende ancora più scalabili ed efficienti le operazioni di sviluppo e deployment dei container.

### **Red Hat OpenShift in AWS: deployment e sviluppo di container**

Devi realizzare app cloud native, raggruppare microservizi, adottare pratiche DevOps o CI/CD, oppure spostare progetti IT scalabili in ambienti diversi che condividono il sistema operativo? Allora i container sono la soluzione ideale. Non dipendendo dal sistema operativo, dalle applicazioni, né dalle librerie e sono portatili e leggeri. Ma presentano anche alcune difficoltà: se decidi di utilizzarli devi comprendere come renderli scalabili, distribuirli e sostituirli e come gestire storage persistente e rilevamento dei servizi.

E se usi solo un runtime per container, sarà tutto molto complesso. Qualsiasi aggiunta operativa richiede ulteriori container, legati a processi separati, che inficiano le prestazioni del server. Inoltre, anche tenere traccia dello storage disaccoppiato dai container (volumi) potrebbe presentare delle difficoltà importanti. Ecco perché, quando si distribuiscono container, è fondamentale disporre di un livello di orchestrazione. Kubernetes è la scelta più naturale, anche se la distribuzione di una soluzione nativa potrebbe non essere altrettanto semplice. Se usi Red Hat Quay con Kubernetes per gestire immagini e dettagli della configurazione ti serve un'applicazione completa e funzionante, mentre Red Hat OpenShift semplifica talmente il processo, che puoi praticamente iniziare subito a utilizzare Kubernetes.

### **Deployment di container in Red Hat OpenShift**

[Red Hat OpenShift](#) è progettato appositamente per semplificare deployment e gestione della piattaforma per container. Dotato di funzioni predisposte per il monitoraggio e l'automazione degli aggiornamenti e delle operazioni di manutenzione integrate, semplifica al massimo le attività dell'amministratore della piattaforma container enterprise basata su Kubernetes.

Red Hat OpenShift predispone una strategia che si adatta ad ogni ambiente, per consentirti di eseguire le applicazioni dove ti servono di più: on-premise o su cloud pubblici, privati o ibridi. In AWS, Red Hat OpenShift viene integrato come servizio ulteriore, aggiunto alla console AWS in modo che sia possibile creare e gestire in modo coerente cluster OpenShift risparmiando tempo e impegno. Puoi anche scegliere una fatturazione on demand, un'unica fattura e l'opzione di ricevere supporto da AWS. Puoi eseguire il [deployment di un cluster Red Hat OpenShift](#) su AWS mantenendo le impostazioni predefinite o personalizzandole, oppure su un'infrastruttura AWS di cui hai eseguito il provisioning e modificare i modelli CloudFormation AWS forniti per soddisfare le tue necessità specifiche.

### **Sviluppo di container in Red Hat OpenShift**

Red Hat OpenShift include il framework source-to-image (S2I), che consente di trasferire il codice da un'applicazione al container, riducendo così la complessità. Devi solo scrivere immagini che prendano il codice sorgente dell'applicazione come input e produrre una nuova immagine che esegua l'applicazione risultante come output. Puoi eseguire queste operazioni con Dockerfiles ed S2I.

#### **Strumenti per sviluppatori disponibili in Red Hat OpenShift:**

- ▶ [CodeReady Workspaces](#) consente ai team di sviluppatori remoti di eseguire il provisioning e condividere i propri ambienti in un clic, accelerando gli avvii e garantendo una bassa latenza nelle interazioni.
- ▶ Il [framework Quarkus](#) permette di creare applicazioni Java Kubernetes native.
- ▶ Tramite Buildah sono disponibili anteprime per [Buildpacks](#) e [Kaniko](#) su build S2I e Dockerfile.
- ▶ [Helm](#) semplifica le interazioni con grafici e release.

I Dockerfile forniti dall'utente richiamano Buildah, parte del builder OpenShift, che contrassegna e invia l'immagine risultante. Oppure, puoi usare la stessa procedura, ma utilizzare S2I per generare il Dockerfile iniziale: l'immagine builder risultante contiene le informazioni necessarie per riprodurre l'immagine eseguibile (ossia un componente).

Come dimostra S2I, Red Hat OpenShift è una piattaforma che supporta sia la realizzazione che il deployment dei container. Gli sviluppatori possono realizzare una piattaforma Kubernetes invece di scrivere codice compatibile con le specifiche delle infrastrutture AWS esistenti. Inoltre, Red Hat OpenShift consente di realizzare le applicazioni in molti modi diversi, offrendo una scelta adeguata a ogni situazione. Si adatta quindi alle necessità degli sviluppatori che non conoscono Kubernetes e vogliono dedicarsi esclusivamente al codice, così come a quelle degli esperti che hanno bisogno della massima flessibilità.

Red Hat OpenShift supporta linguaggi, database e strumenti già noti agli sviluppatori e semplifica l'accesso ai servizi necessari per lo sviluppo di applicazioni, come quelli offerti da AWS.

### **Strumenti per sviluppatori di container in Red Hat OpenShift**

Gli strumenti AWS semplificano ulteriormente il processo di sviluppo dei container in Red Hat OpenShift e possono essere usati anche su altre piattaforme, come Red Hat Enterprise Linux.

**AWS CodeArtifact** è un servizio per componenti completamente gestito che semplifica archiviazione, pubblicazione e condivisione di pacchetti software utilizzati nei processi di sviluppo software di organizzazioni di ogni dimensione.

**AWS Cloud9** è un ambiente di sviluppo integrato (IDE) utilizzato per scrivere, eseguire ed effettuare il debug di codice in una shell tramite browser, che consente anche di installare software aggiuntivo, eseguire git push o immettere comandi.

**AWS Cloud Development Kit** crea applicazioni tramite i linguaggi di programmazione, molto diffusi e utilizzati, accelerando così il processo di onboarding in AWS, visto che gran parte delle nozioni sono già note.

**Amazon CodeGuru** offre immagini e consigli su come risolvere i problemi relativi alle prestazioni e stima i costi dell'utilizzo di codice inefficiente, per consentire agli sviluppatori di intervenire secondo la giusta priorità. Si integra nel workflow di sviluppo software in Red Hat OpenShift esistente, suggerendo modifiche per migliorare la qualità del codice e identificando le linee di codice più costose dell'applicazione.

### **Inizia subito a creare, distribuire e gestire container**

Realizzare container di applicazioni non deve essere per forza difficile: affidati agli strumenti specifici di Red Hat e a quelli per sviluppatori di AWS. Distribuzione e orchestrazione non sono solo gli ultimi passaggi della distribuzione dei container, sono anche i più complicati. Red Hat OpenShift elimina gran parte delle complessità legate a Kubernetes, per consentirti di distribuire e gestire i container su larga scala, indipendentemente dalla modalità con cui sono stati creati.

Red Hat OpenShift non è una semplice soluzione per il deployment: offre infatti agli sviluppatori una vasta scelta per realizzare e distribuire i container. Red Hat OpenShift è disponibile su AWS sia come servizio autogestito che completamente gestito e consente di distribuire e gestire le implementazioni OpenShift in AWS. Red Hat OpenShift Service on AWS e Red Hat OpenShift Dedicated si uniscono alla gamma completa Red Hat Cloud Services, che ti consente di concentrarti sulla creazione e l'introduzione delle tue applicazioni sul mercato.

#### **Altri strumenti per sviluppatori disponibili in Red Hat OpenShift:**

- ▶ **Odo** consente agli sviluppatori, tramite la sua CLI, di eseguire operazioni sul codice. Supporta Kubernetes e Red Hat OpenShift, è un modello open basato su definizioni standard e offre uno sviluppo Java rapido e iterativo tramite Quarkus.
- ▶ **Knative** consente di compilare applicazioni serverless e basate sugli eventi come **Strimzi** (per eseguire Apache Kafka in OpenShift) e **Service Mesh**.
- ▶ **Tekton** in Red Hat OpenShift Pipelines, plug-in OpenShift per azioni GitHub, Jenkins e runner per GitLab offrono capacità di integrazione continua.



Indipendentemente da come sviluppi o distribuisca i container, Red Hat e AWS hanno soluzioni per accelerarne e semplificarne la realizzazione, così puoi creare e distribuire le applicazioni più rapidamente.

Scopri come realizzare un'infrastruttura moderna con [Red Hat Enterprise Linux](#) grazie alla [partnership fra Red Hat e AWS](#) o approfondisci l'argomento:

- ▶ [Documentazione su Red Hat Enterprise Linux](#)
- ▶ [Podman](#)
- ▶ [Strumenti per sviluppatori in AWS](#)



### Informazioni su Red Hat

Red Hat è leader mondiale nella fornitura di soluzioni software open source. Con un approccio che si avvale della collaborazione delle community, distribuisce tecnologie come Kubernetes, container, Linux e cloud ibrido caratterizzate da affidabilità e prestazioni elevate. Red Hat consente di sviluppare applicazioni cloud native, integrare applicazioni IT nuove ed esistenti, e automatizzare e gestire ambienti complessi. [Considerata un partner affidabile dalle aziende della classifica Fortune 500](#), Red Hat fornisce [pluripremiati](#) servizi di consulenza, formazione e assistenza, che portano i vantaggi dell'innovazione open source in qualsiasi settore. Red Hat è l'elemento catalizzatore in una rete globale di aziende, partner e community, e permette alle organizzazioni di crescere, evolversi e prepararsi a un futuro digitale.

---

#### Italia

it.redhat.com  
italy@redhat.com

#### Europa, Medio Oriente, e Africa (EMEA)

00800 7334 2835  
it.redhat.com  
europe@redhat.com

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

it.redhat.com  
#F30136\_1021

Copyright © 2021 Red Hat, Inc. Red Hat, il logo Red Hat e OpenShift sono marchi commerciali registrati di proprietà di Red Hat, Inc. o delle società da essa controllate con sede negli Stati Uniti e in altri Paesi. Linux® è un marchio registrato di proprietà di Linus Torvalds depositato negli Stati Uniti e in altri Paesi. Tutti gli altri marchi sono di proprietà delle aziende qui menzionate. Java e tutti i marchi e loghi relativi sono marchi commerciali o marchi registrati di Oracle America, Inc. negli Stati Uniti e in altri Paesi." In assenza di spazio, sostituire con "Java è un marchio registrato di Oracle America, Inc.