

Il percorso verso le applicazioni cloud native

L'adozione del cloud native in 8 passaggi



Contenuti

Introduzione	3
Passaggio 1: Trasforma la cultura e le prassi a supporto del cloud	3
Passaggio 2: Introduci i microservizi per ottenere applicazioni più veloci	3
Passaggio 3: Adotta servizi applicativi per uno sviluppo più rapido	4
Passo 4: Scegli lo strumento adatto a ogni attività	4
Passo 5: Predisponi un'infrastruttura self service e on demand	5
Passaggio 6: Automatizza l'IT per una distribuzione più rapida delle applicazioni	6
Passaggio 7: Adotta tecniche di continuous delivery e deployment avanzato	6
Passaggio 8: Sviluppa un'architettura più modulare	8
Il ruolo di Red Hat	8

Introduzione

L'interazione della maggior parte delle organizzazioni con i propri clienti, partner e dipendenti avviene oggi tramite le applicazioni. Questo repentino avvento di nuove capacità native digitali ha rivoluzionato i modelli di business convenzionali e imposto ad aziende e settori già consolidati di adeguare e innovare le proprie modalità operative.

Per gran parte delle organizzazioni, creare un'esperienza digitale innovativa significa adattarsi a una cultura organizzativa agile, che consenta di soddisfare la rapidità della domanda adottando modelli di sviluppo e di distribuzione più veloci e flessibili. Tuttavia, sono poche le organizzazioni che riescono a ricostruire da zero le proprie basi tecnologiche o ad adottare all'istante prassi e approcci nuovi. La maggior parte sceglie quindi di trasformare in modo graduale ma radicale la cultura, i processi e le tecnologie, in modo da ottenere maggiore velocità e agilità. In questo ebook vengono delineati gli otto passaggi verso l'adozione di un approccio cloud native per le applicazioni.

"Sostanzialmente, lo sviluppo cloud native riguarda i team, le persone e la collaborazione tanto quanto la tecnologia. La collaborazione è strategica per creare app in modo iterativo e flessibile, ma tutti, inclusi stakeholder e autori, devono contribuire alla creazione, alla codifica, al test e alla distribuzione del prodotto".

Passaggio 1: Trasforma la cultura e le prassi a supporto del cloud

Il percorso di adozione delle applicazioni cloud native comporta, per i team che si occupano di sviluppo, linee di business e di processi IT, una profonda trasformazione, che consenta loro di realizzare e distribuire le applicazioni in modo più rapido ed efficiente. Indipendentemente dalle dimensioni o dal settore in cui opera, ogni azienda deve prendere in esame le attività, le tecnologie, i team e i processi che richiedono collaborazione e coordinamento per la riuscita del deployment delle applicazioni cloud. Gli approcci tradizionali all'uso delle risorse di cloud pubblico o ibrido permettono ai team di prendere decisioni rapide e in autonomia, ma continuano a generare dati isolati e processi ambientali che ostacolano le possibilità di innovazione.

In un'epoca di rapida innovazione, la scelta di sviluppare una strategia cloud che uniformi le applicazioni può risultare impegnativa per le organizzazioni che devono affrontare la complessità di gestire più ambienti distribuiti, applicazioni esistenti altamente personalizzate e nuovi carichi di lavoro per le applicazioni. Spesso, se non hanno esteso la strategia cloud a tutta l'azienda, le organizzazioni non riescono a sfruttare appieno tutte le potenzialità delle loro applicazioni.

L'adozione di una cultura cloud collaborativa si basa non solo sull'uso di strumenti e tecnologie innovativi, ma anche sulla volontà e sulla capacità delle persone di adottare un approccio più integrato e collaborativo allo sviluppo e alla distribuzione delle applicazioni. La cultura su cui si basano i progetti software open source offre una guida alla creazione di una strategia cloud connessa e coesa per le applicazioni.

Passaggio 2: Introduci i microservizi per ottenere applicazioni più veloci

All'avvio del processo di adozione dell'approccio cloud native, l'attenzione delle aziende non deve concentrarsi solo sullo sviluppo di nuove applicazioni, poiché molte di quelle esistenti sono fondamentali per la continuità del servizio e per il fatturato. Aniché pensare a una semplice sostituzione delle applicazioni in uso, occorre integrarle e renderle compatibili con le nuove applicazioni cloud native. La soluzione ideale consiste nel trasformare gradualmente l'architettura monolitica esistente in un'architettura più modulare e basata sui microservizi, introducendo una comunicazione basata sulle interfacce di programmazione delle applicazioni (API).

Prima di iniziare a gestire le complessità legate alla transizione di applicazioni monolitiche in microservizi, è quindi necessario creare una base robusta per l'architettura di microservizi.

Report di ricerca Red Hat: panoramica sullo sviluppo cloud native, giugno 2021

L'adozione di un approccio ai microservizi non significa cambiare tutto e subito. È possibile iniziare a suddividere ogni monolite in componenti di minori dimensioni, secondo le esigenze specifiche, adottando un approccio strutturato in grado di garantire che le applicazioni siano basate su solidi principi di progettazione, entro domini dai confini ben definiti. Questo approccio consente di realizzare una transizione più graduale e meno rischiosa a un'architettura di microservizi, e getta le basi per un'efficace architettura basata sui microservizi.

Le applicazioni fortemente dipendenti dalle piattaforme già in uso possono diventare più veloci spostando il monolite esistente su una piattaforma basata su container. In tal modo si velocizzano il deployment e la distribuzione, garantendo un maggior ritorno sugli investimenti (ROI). Le successive integrazioni o funzionalità del monolite potranno poi essere create mediante tecniche e approcci cloud native.

Passaggio 3: Adotta servizi applicativi per uno sviluppo più rapido

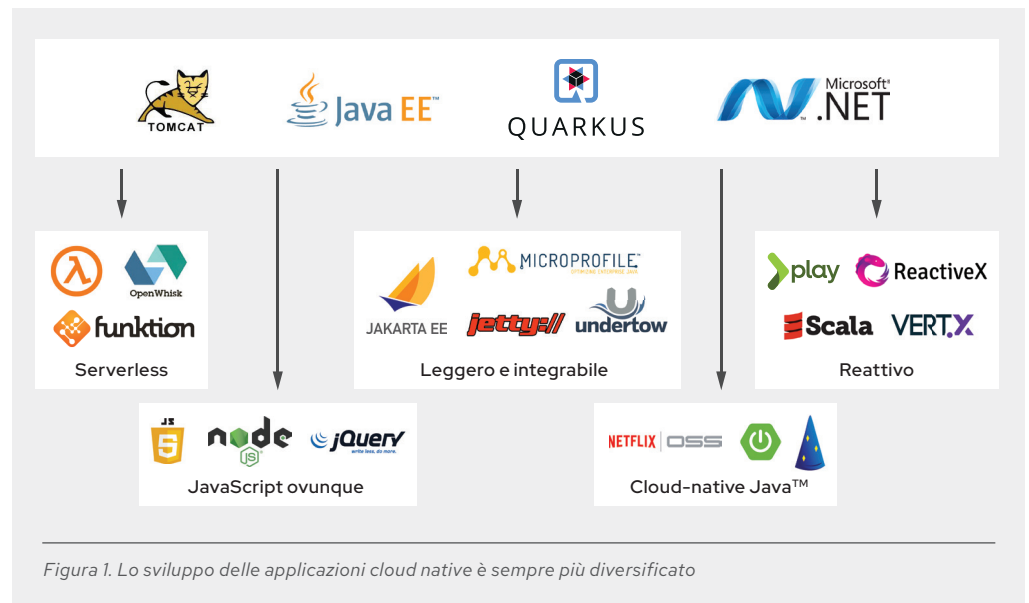
La riutilizzabilità è sempre stata una caratteristica fondamentale per rendere più rapido lo sviluppo del software e le applicazioni cloud native non fanno eccezione. Per garantire i vantaggi in termini di velocità e scalabilità che caratterizzano il cloud, tuttavia, i componenti riutilizzabili per le applicazioni cloud native devono essere ottimizzati e integrati nell'infrastruttura cloud native alla base.

Anziché creare da zero servizi di cache, motori di gestione delle regole o del flusso di lavoro, connettori di integrazione, funzionalità di gestione mobile e delle API, servizi di virtualizzazione dei dati, broker di messaggistica o un framework serverless, è possibile utilizzare quelli esistenti, dopo averli ottimizzati e integrati nell'infrastruttura basata su container. Questi servizi applicativi, che si tratti di soluzioni Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) o Integration Platform-as-a-Service (iPaaS), sono strumenti efficaci e pronti per essere utilizzati da subito dagli sviluppatori.

Mentre i processi DevOps e la containerizzazione accelerano l'erogazione e il deployment delle applicazioni, nel caso delle applicazioni cloud native, uno o più di questi tipi di servizi può essere necessario per ottenere cicli di sviluppo più rapidi e accorciare il time to market delle nuove applicazioni. Ad esempio, gli sviluppatori possono avvalersi di servizi applicativi creati specificamente per offrire prestazioni ottimizzate in un'infrastruttura basata su container, ma anche per trarre vantaggio da funzionalità della piattaforma, quali pipeline di integrazione e distribuzione continue (CI/CD), deployment in sequenza e deployment blue green, scalabilità automatica, tolleranza agli errori e così via.

Passaggio 4: Scegli lo strumento adatto a ogni attività

Le applicazioni cloud native consentono di adattare i linguaggi e i framework di sviluppo scelti alle specifiche esigenze dell'applicazione aziendale. La gestione del conseguente aumento della complessità e della varietà delle applicazioni richiede piattaforme applicative basate su container, che supportino la giusta combinazione di framework, linguaggi e architetture necessarie per lo sviluppo cloud native.



Lo sviluppo cloud native implica inoltre la scelta dello strumento più indicato per ogni attività. Che l'implementazione delle applicazioni cloud native si basi su approccio 12-factor, progettazione basata su dominio, progetto e sviluppo basati su test, MonolithFirst, monoliti veloci, mini o microservizi, la piattaforma cloud native deve ottimizzare framework, linguaggi e architetture, in modo da soddisfare i requisiti di sviluppo adeguati. La piattaforma alla base incentrata sui container deve inoltre supportare una serie di runtime e framework appositamente predisposti e continuamente aggiornati per adeguarsi alle innovazioni tecnologiche.

Passaggio 5: Predisponi un'infrastruttura self service e on demand

Le metodologie agili permettono agli sviluppatori di creare e aggiornare rapidamente il software, ma quando è necessario accedere all'infrastruttura è spesso difficile farlo con tempismo ed efficienza. La velocità di accesso al mercato quando si rilasciano le applicazioni in produzione ne risente. In un'epoca in cui le infrastrutture sono poco costose e il talento tecnico oneroso, compilare un ticket e attendere settimane prima che le risorse siano rilasciate dall'IT non è più sostenibile.

Per evitare il ricorso allo shadow IT non autorizzato, il provisioning delle infrastrutture self service e on demand offre l'alternativa più efficace poiché consente agli sviluppatori di accedere all'infrastruttura che desiderano, quando ne hanno bisogno. Questo modello, tuttavia, è efficace solo se i team dei processi IT mantengono il controllo e la visibilità su questo ambiente dinamico e complesso.

I container e le relative tecnologie di orchestrazione permettono di astrarre e semplificare l'accesso all'infrastruttura alla base, fornendo affidabili funzionalità di gestione del ciclo di vita delle applicazioni nei vari ambienti infrastrutturali, come datacenter e cloud privati e pubblici. Le piattaforme basate sui container offrono ulteriori capacità self service, di automazione e di gestione del ciclo di vita delle applicazioni. Questo modello permette agli sviluppatori e ai team operativi di attivare rapidamente ambienti coerenti, grazie ai quali gli sviluppatori possono dedicarsi alla creazione di applicazioni senza dover gestire le difficoltà legate al provisioning dell'infrastruttura.

I container supportano inoltre la portabilità delle applicazioni, compresa la creazione di applicazioni cloud native che possono essere implementate ed eseguite su qualsiasi provider cloud. Grazie alla portabilità è possibile scegliere qualsiasi provider cloud in qualsiasi momento, passando agevolmente da uno all'altro, ottimizzando i relativi costi e sviluppando un'applicazione multicloud il cui codice non è vincolato ad alcuna specifica API di un provider cloud.

Passaggio 6: Automatizza l'IT per una distribuzione più rapida delle applicazioni

L'automazione dell'IT o dell'infrastruttura è un passaggio essenziale per ottenere una distribuzione più rapida delle applicazioni cloud native poiché elimina le attività manuali. L'automazione può essere integrata e applicarsi a qualsiasi attività o componente, dal provisioning di rete e dell'infrastruttura al deployment delle applicazioni e alla gestione delle configurazioni.

Gli strumenti per l'automazione e la gestione dell'IT creano regole, framework e processi ripetibili che possono sostituire o ridurre interazioni in cui la rilevante componente umana può ritardare il rilascio. Possono inoltre estendersi a tecnologie specifiche come i [container](#), a metodologie come [DevOps](#) o ad aree più vaste come [cloud computing](#), sicurezza, test, monitoraggio e avvisi. L'automazione, quindi, è una scelta strategica che consente l'ottimizzazione dell'IT e la trasformazione digitale accelerando il time to value dell'azienda.

Scopri l'importante ruolo dell'automazione IT in "L'azienda automatizzata"

[Scarica l'ebook](#)

La distribuzione continua è una procedura per lo sviluppo software che si avvale dell'automazione per accelerare il rilascio di nuovo codice. Definisce un processo con il quale le modifiche apportate da uno sviluppatore a un'applicazione possono essere inviate, tramite l'[automazione](#), a un repository di codice o a un [registro di container](#).

Guida all'automazione dell'IT

1. Adotta un approccio programmatico all'automazione dei processi IT da estendere all'intera organizzazione. Promuovi l'adozione di metodi basati su confronto e collaborazione nell'intera organizzazione per progettare i requisiti dei servizi.
2. Prendi in considerazione le sandbox come base per apprendere il linguaggio e i processi dell'automazione.
3. Analizza in maniera approfondita ogni aspetto dell'automazione. Elimina tutte le fasi manuali non necessarie, anche se mantenere i controlli manuali sembra più sicuro.
4. Adotta l'automazione in modo sistematico, con passaggi graduali e realizzabili. Crea una pratica di automazione generalizzata basata su fasi collaudate e condivise.
5. Inizia con l'automazione di un'attività o un servizio, che si tratti di elaborazione, rete, storage o provisioning. Condividi il risultato e sviluppa una strategia sistematica.
6. Utilizza cataloghi self service che offrono strumenti agli utenti e accelerano la distribuzione.
7. Introduci policy e processi di misurazione, monitoraggio e chargeback.

Nel tempo otterrai un'automazione integrata completa che garantirà alla tua azienda una maggiore efficienza, processi DevOps più rapidi e un'innovazione più veloce.

Passaggio 7: Adotta tecniche di distribuzione continua e di deployment avanzato

Cicli di rilascio lunghi significano ritardi tra l'individuazione e la risoluzione dei bug ma anche difficoltà nell'adattarsi tempestivamente all'evoluzione della domanda del mercato e delle esigenze dei clienti. Per le applicazioni con traffico elevato, come quelle mobile, web, IoT o di edge computing, un errore non risolto può avere conseguenze su molti utenti, compromettendo l'esperienza dei clienti, creando problemi di sicurezza e riducendo la produttività o il fatturato. Anche per le altre applicazioni aziendali interne, le interruzioni dei servizi o i ritardi nella risoluzione degli errori software possono comportare costi elevati.

"Le tecniche di deployment avanzate conferiscono struttura e trasparenza all'innovazione. Le metodologie di deployment mature generano un ambiente che consente sperimentazione, feedback e analisi reali. Una sperimentazione migliore porta a un'innovazione migliore."

Burr Sutter
Director of Developer Experience,
Red Hat

Le metodologie di sviluppo agili hanno creato un modello che prevede rilasci rapidi e frequenti, mentre gli approcci DevOps e di distribuzione continua hanno ampliato questo modello, consolidando la collaborazione tra i team che si occupano di sviluppo, operazioni, assicurazione della qualità e sicurezza, al fine di migliorare i processi di distribuzione del software. Le modifiche apportate al codice, di conseguenza, vengono trasferite alla produzione in modo rapido e affidabile, per fornire velocemente un feedback agli sviluppatori. Questo ciclo iterativo basato su feedback veloci è reso possibile dalla pipeline CI/CD, che amplia le funzionalità di automazione dell'infrastruttura a un sistema di distribuzione automatizzato end to end in grado di coprire tutti gli aspetti della distribuzione delle applicazioni, compresi test automatizzati, scansione delle vulnerabilità, sicurezza e conformità e verifiche normative. Con i flussi di distribuzione automatizzati l'obiettivo è fornire aggiornamenti senza incidere sulla capacità operativa e riducendo i rischi.

Il primo passo per raggiungere la **distribuzione continua** (CD) è l'attivazione dell'integrazione continua (CI), un sistema di compilazione che monitora le modifiche nei repository di controllo della sorgente, esegue i test applicabili e crea automaticamente la versione più recente dell'applicazione.

I modelli di deployment avanzato puntano a ridurre il rischio dei rilasci software e a creare un ambiente che consenta la sperimentazione con risultati controllati senza conseguenze negative indesiderate per i clienti, un obiettivo essenziale per favorire l'innovazione nell'organizzazione.

Le tecniche di deployment avanzato trasformano la distribuzione da un'attività che si svolge al di fuori dell'orario di lavoro, durante i week-end, con finestre di servizio e periodi di indisponibilità, in un'operazione normale dell'orario lavoro, senza tempi di fermo della produzione, effettuata mentre l'applicazione è disponibile ai clienti.

Eliminando le difficoltà che i nuovi deployment possono causare ai clienti, queste metodologie permettono di distribuire aggiornamenti e versioni ogni volta che le esigenze aziendali lo richiedono. Di seguito sono riportate alcune delle più comuni tecniche di deployment che consentono di eliminare le interruzioni, a seconda degli scenari di utilizzo delle applicazioni:

Con il deployment in sequenza, anziché aggiornare contemporaneamente tutte le istanze di un'applicazione, ogni istanza viene aggiornata individualmente escludendola dal bilanciamento del carico, in modo che non riceva traffico. Dopo l'aggiornamento viene nuovamente inclusa nel bilanciamento del carico e il processo continua finché tutte le istanze non vengono aggiornate.

Il deployment blue green consiste nell'esecuzione di due ambienti identici, di cui uno solo è attivo. Le modifiche vengono effettuate nell'ambiente inattivo e, dopo la verifica in produzione, il traffico reale viene convogliato nell'ambiente aggiornato. Il rollback alla versione precedente è semplice, poiché è sufficiente indirizzare nuovamente il traffico all'ambiente precedente, a condizione che si sia svolta la transizione dei dati.

Simile al deployment blue green, il canary deployment utilizza anch'esso due ambienti identici ma in questo caso è diverso il controllo esercitato sul rollout. Dopo il deployment, il nuovo rilascio viene testato su un piccolo gruppo di clienti in produzione. Se tutto funziona correttamente, il traffico viene gradualmente spostato sulla nuova versione, mentre i risultati sono monitorati e verificati finché tutti gli utenti non utilizzano la nuova versione.

Passaggio 8: Sviluppa un'architettura più modulare

In un approccio architetturale alla scrittura del software basato sui microservizi, le applicazioni vengono scomposte nei loro elementi più piccoli e indipendenti tra loro. Rispetto all'approccio monolitico tradizionale, in cui tutti i componenti vanno a costituire un'unica entità inscindibile, i microservizi sono invece unità indipendenti che operano in sinergia per completare le stesse attività. Poiché particolarmente leggero, questo tipo di approccio allo sviluppo del software offre un livello maggiore di granularità e consente di condividere processi simili tra più app. Benché non sia necessaria una specifica infrastruttura, la base ideale di un'architettura di microservizi è costituita da una piattaforma incentrata sui container.

L'evoluzione verso un'architettura basata sui microservizi può risultare particolarmente vantaggiosa per team di grandi dimensioni o per i deployment in produzione effettuati più volte al giorno. Dal punto di vista architetturale, i microservizi implicano che ciascun servizio venga scomposto in una propria unità di deployment. In tal modo, è possibile gestire e distribuire in maniera indipendente ogni microservizio, eventualmente prevedendo vari team responsabili per i diversi cicli di vita.

Un'altra alternativa ai microservizi sono i miniservizi, cioè raccolte di servizi suddivise in base al dominio ed eseguite solitamente su un server applicativo. Questa soluzione migliora l'agilità e la scalabilità delle applicazioni, rimuovendo la complessità di un'infrastruttura e di un progetto basati su microservizi. I miniservizi richiedono comunque un investimento in metodologie agili, DevOps e CI/CD, cosicché la soluzione ideale è un moderno server delle applicazioni o un'offerta con più framework, architetture e linguaggi combinata con un'infrastruttura basata sui container.

Il segreto del successo con le applicazioni cloud native è una piattaforma che supporti diversi framework, linguaggi e approcci allo sviluppo di applicazioni cloud native, cioè microservizi, miniservizi o MonolithFirst.

Il ruolo di Red Hat

Red Hat dispone delle tecnologie e dei servizi necessari per fornirti tutto il supporto di cui hai bisogno, a seconda delle tue priorità e della fase del processo di adozione del cloud native in cui ti trovi.

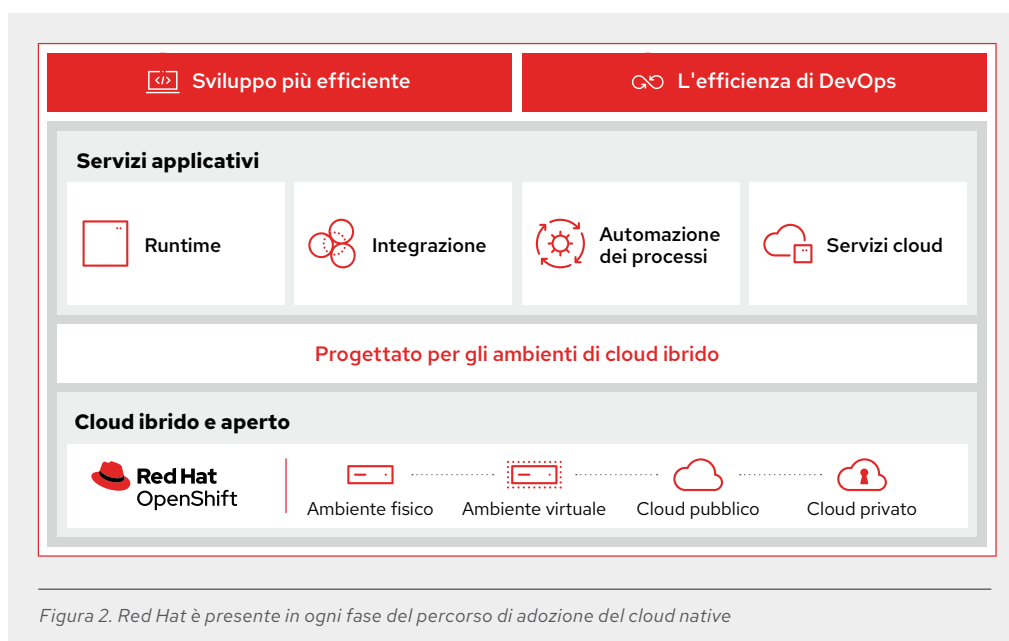


Figura 2. Red Hat è presente in ogni fase del percorso di adozione del cloud native

Alcune aziende decidono di concentrarsi su un solo scenario di utilizzo cloud native, mentre altre scelgono di affrontare contemporaneamente più situazioni. Che l'approccio sia evolutivo o rivoluzionario, il percorso è individuale e non necessariamente lineare. Qualunque sia la strategia scelta, per portare rapidamente le applicazioni sul mercato servono la tecnologia più appropriata, prassi DevOps e il giusto approccio.

Red Hat ti affianca in questo percorso con [Red Hat® OpenShift®](#), una piattaforma di sviluppo e orchestrazione containerizzata cloud native. [Red Hat Application Services](#) è una gamma completa di servizi per lo sviluppo applicativo in esecuzione su Red Hat OpenShift. Include framework, strumenti e soluzioni per lo sviluppo, il deployment e la scalabilità delle applicazioni cloud native. Per le organizzazioni che puntano a ridurre i tempi di accesso al mercato, i servizi applicati di [Red Hat OpenShift](#) (parte di [Red Hat Cloud Services](#)) sono servizi cloud in hosting e gestiti per Red Hat OpenShift che garantiscono un'esperienza di sviluppo ottimizzata per la creazione, il deployment e la scalabilità di nuove applicazioni cloud native.

Per aiutarti ad affrontare la complessità dello sviluppo cloud native, [Red Hat Consulting](#) fornisce consulenza strategica ed esperienza tecnica approfondita. Da [Red Hat Open Innovation Labs](#) alle discovery session e ai piani di implementazione del progetto, i nostri consulenti ti affiancheranno in ogni fase del tuo percorso cloud native.



Informazioni su Red Hat

Red Hat è leader mondiale nella fornitura di soluzioni software enterprise open source. Con un approccio basato sul concetto di 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.

f facebook.com/RedHatItaly
t twitter.com/RedHatItaly
in linkedin.com/company/red-hat

Italia
it.redhat.com
italy@redhat.com

**Europa, Medio Oriente,
e Africa (EMEA)**
00800 7334 2835
it.redhat.com
europa@redhat.com