



IL PERCORSO VERSO LE APPLICAZIONI CLOUD-NATIVE

8 passi per guidarti nel tuo percorso

E-BOOK

SOMMARIO

1. VELOCITÀ: UN IMPERATIVO PER LE AZIENDE DIGITALI	3
2. CHE COS'È UN'APPLICAZIONE CLOUD-NATIVE?	3
3. APPLICAZIONI TRADIZIONALI E CLOUD-NATIVE	4
4. I QUATTRO FONDAMENTI DELLO SVILUPPO E DEL DEPLOYMENT DELLE APPLICAZIONI CLOUD-NATIVE	6
5. IL PERCORSO VERSO LE APPLICAZIONI CLOUD-NATIVE: 8 PASSI	7
PASSO 1: Sviluppare la cultura e le pratiche di DevOps	7
PASSO 2: Snellire le applicazioni esistenti utilizzando monoliti veloci	7
PASSO 3: Utilizzare servizi applicativi per accelerare lo sviluppo	8
PASSO 4: Scegliere lo strumento giusto per l'attività giusta	8
PASSO 5: Fornire un'infrastruttura self-service e on-demand	9
PASSO 6: Automatizzare l'IT per accelerare il rilascio delle applicazioni	9
PASSO 7: Adottare tecniche di continuous delivery e deployment avanzato	10
PASSO 8: Sviluppare un'architettura più modulare	11
6. BUSINESS CASE DI APPLICAZIONI CLOUD-NATIVE	12

"Le aziende digitalmente avanzate hanno 8 volte più probabilità di aumentare la loro presenza sul mercato ma restano comunque indietro rispetto ai nativi digitali".

[Indagine Bain: For Traditional Enterprises, the Path to Digital and the Role of Containers](#)

"Cloud-native" è un termine che descrive applicazioni, architetture, piattaforme/infrastrutture e processi che, insieme, rendono economico operare in modo da migliorare la nostra capacità di rispondere rapidamente al cambiamento e ridurre l'imprevedibilità."

CHRISTIAN POSTA
CHIEF ARCHITECT RED HAT
E AUTORE DI *MICROSERVICES FOR JAVA DEVELOPERS* (IN LINGUA INGLESE)

Fonte: [INFOQ "DEFINING CLOUD NATIVE: A PANEL DISCUSSION" 2017](#) (IN LINGUA INGLESE).

1. VELOCITÀ: UN IMPERATIVO PER LE AZIENDE DIGITALI

Il termine aziende digitali evoca tecnologie innovative: dispositivi mobili, sensori intelligenti, dispositivi indossabili, realtà virtuale, chatbot, blockchain, machine learning e molto altro. Richiama anche alla mente la rapida espansione delle nuove aziende native digitali che hanno rivoluzionato i modelli aziendali tradizionali, destabilizzando aziende e settori ormai consolidati. Per la maggior parte delle organizzazioni, le attività digitali implicano l'adattamento a una cultura di agilità organizzativa, in cui è possibile soddisfare la rapidità della domanda solo adottando modelli di sviluppo e di distribuzione più veloci e flessibili. In generale, tuttavia, le organizzazioni non possono permettersi il lusso di ricostruire completamente le proprie basi tecnologiche né di adottare immediatamente prassi e mentalità nuove. Scelgono quindi di trasformare in modo graduale ma radicale la cultura, i processi e le tecnologie, in modo da ottenere maggiore velocità e agilità.

Il software svolge un ruolo sempre più importante nel modo in cui gli utenti entrano in relazione con le aziende e nel modo in cui le aziende innovano per restare competitive. Di conseguenza, la velocità dello sviluppo e della distribuzione delle applicazioni rappresenta il nuovo imperativo per le aziende digitali.

L'approccio cloud-native consiste nel **modernizzare le applicazioni esistenti e nel creare applicazioni nuove** in base ai principi cloud, utilizzando servizi e adottando processi ottimizzati per l'agilità e l'automazione del cloud computing. Questo e-book descrive in modo dettagliato i passi che permettono una transizione efficace verso l'adozione di un approccio cloud-native alle applicazioni.

2. CHE COS'È UN'APPLICAZIONE CLOUD-NATIVE?

Un'applicazione cloud-native consente di sfruttare i modelli di cloud computing al fine di migliorare la velocità, la flessibilità e la qualità, riducendo al contempo i rischi di deployment. Nonostante il nome, l'approccio cloud-native non riguarda tanto **dove** vengono implementate le applicazioni, quanto **come** esse vengono costruite, implementate e gestite.

Gli approcci cloud-native sono simili alle architetture di microservizi. La creazione di applicazioni cloud-native può portare all'uso dei microservizi, tuttavia, per raggiungere un livello di maturità tale da poter gestire i microservizi in ambiente di produzione il processo è piuttosto lungo. I microservizi non sono indispensabili per poter accedere a tutti i vantaggi offerti dalle app cloud-native. Molte organizzazioni ottengono questi benefici specificamente creando migliori monoliti modulari basati sugli stessi principi.

L'evoluzione verso lo sviluppo e la distribuzione di applicazioni cloud-native è multidimensionale, poiché influisce su cultura, processi, architettura e tecnologia. Un processo di cambiamento di questa portata può naturalmente presentare criticità e per questo si presenta più come un viaggio che come una destinazione.

3. APPLICAZIONI TRADIZIONALI E CLOUD-NATIVE

Rispetto alle metodologie tradizionali, lo sviluppo di applicazioni cloud-native implica necessariamente dei cambiamenti.

TABELLA 1. SVILUPPO DI APPLICAZIONI TRADIZIONALI E CLOUD-NATIVE

	TRADIZIONALE	CLOUD-NATIVE
INCENTRATO SU	Longevità e stabilità	Time-to-market
METODOLOGIA DI SVILUPPO	Sviluppo semi-Agile a cascata	Sviluppo Agile, DevOps
TEAM	Team operativi, di sviluppo, QA e sicurezza isolati	Team DevOps collaborativi
CICLI DI DISTRIBUZIONE	Lunghi	Corti e continui
ARCHITETTURA APPLICATIVA	Con accoppiamento elevato Monolitica	Con basso accoppiamento Basata sui servizi Comunicazione basata su API (Application programming interface)
INFRASTRUTTURA	Incentrata su server Progettata per on-premise Dipendente dall'infrastruttura Scalabile verticalmente Predisposta per la capacità di picco	Incentrata su container Studiata per on-premise e cloud Portabile su tutta l'infrastruttura Scalabile orizzontalmente Capacità on-demand

3.1 SVILUPPO E DISTRIBUZIONE DELLE APPLICAZIONI TRADIZIONALI

Molte applicazioni fondamentali per le operazioni aziendali non sono state progettate per le esperienze digitali. Caratterizzate da tempi prolungati, sia di creazione che di durata, si basano su monoliti con accoppiamento elevato, realizzati a fronte di specifiche molto precise spesso stabilite molto prima della distribuzione.

Questi approcci allo sviluppo funzionavano generalmente secondo un meccanismo sequenziale e a cascata e richiedevano molto tempo. Il loro adattamento alle prassi semi-Agile è avvenuto solo recentemente. Le fasi di sviluppo, test, sicurezza e conformità, deployment e gestione delle applicazioni avvenivano in aree funzionali separate, ciascuna con team, ruoli e responsabilità diversi e flussi di comunicazione lineari tra gli interessati.

Si trattava di applicazioni di grandi dimensioni, multifunzionali e con accoppiamento elevato che, a prescindere dall'ambiente tecnologico, combinavano interfaccia utente, vari servizi applicativi, codice di accesso ai dati e altri componenti. Un'applicazione di e-commerce creata come un monolite ad accoppiamento elevato, ad esempio, includeva generalmente tutte le funzionalità dell'interfaccia utente per il Web, i cataloghi dei prodotti, il carrello della spesa, le raccomandazioni, le recensioni e i commenti sui prodotti, il sistema di pagamento e gli altri componenti necessari per effettuare gli acquisti sul sito web di e-commerce, tutto in un'unica applicazione.

L'infrastruttura della maggior parte delle applicazioni tradizionali era predisposta per la capacità di picco necessaria e resa scalabile aumentando la capacità hardware del server mediante scalabilità verticale.

Entro il 2020, oltre il 50 % delle applicazioni di modalità 1* migrate dai datacenter privati al cloud pubblico saranno riscritte in base ai principi architetturali cloud-native, rispetto a meno del 10 % del 2017.

Gartner: Why You Must Begin Delivering Cloud-Native Offerings Today, Not Tomorrow, January 2018 (in lingua inglese)

3.2 SVILUPPO E DISTRIBUZIONE DELLE APPLICAZIONI CLOUD-NATIVE

Lo sviluppo di applicazioni cloud-native è incentrato sulla rapidità del time-to-market, perciò richiede uno sviluppo più agile e basato su API e servizi, con approcci fondati sulla distribuzione continua. Alla base di queste caratteristiche si trova la collaborazione DevOps tra i team che si occupano dello sviluppo e della distribuzione, un'architettura più modulare e un'infrastruttura flessibile scalabile orizzontalmente on-demand, in grado di supportare più ambienti e di assicurare la portabilità delle applicazioni.

Con la flessibilità e l'agilità offerta dalle moderne tecnologie cloud, le organizzazioni desiderano spostare le applicazioni tradizionali agli ambienti cloud, per godere dei vantaggi dati dalla maggiore agilità e dalla capacità di elaborazione on-demand.

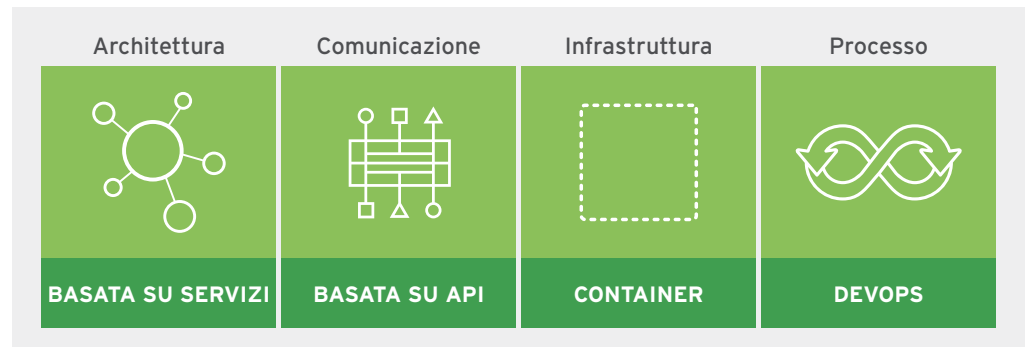
Molte delle funzionalità operative integrate nelle piattaforme tradizionali, tuttavia, sono obsolete e non necessarie in ambiente cloud o, semplicemente, sono già fornite e gestite dall'ambiente cloud stesso. Gli ambienti cloud semplificano la gestione del ciclo di vita dell'host, aiutano le aziende a sfruttare al meglio i principi infrastrutturali invariabili e a ottimizzare gli host in base alle esigenze delle singole istanze applicative.

Il percorso che porta alle applicazioni cloud-native è diverso a seconda dell'organizzazione. La semplice creazione di microservizi non assicura la qualità del servizio e la frequenza di distribuzione richiesta dalle attività digitali. Allo stesso modo, limitandosi ad adottare strumenti che supportano lo sviluppo Agile o l'automazione IT non si ottiene la maggiore velocità offerta dagli approcci cloud-native. Il successo è definito invece da una combinazione di prassi, tecnologie, processi e mentalità.

* *Gartner definisce bimodale la prassi di gestire due stili di lavoro separati ma coerenti: uno incentrato sulla prevedibilità, l'altro sull'esplorazione. La modalità 1 è ottimizzata per le aree più prevedibili e conosciute. Incentrata in particolare su ciò che è conosciuto, si occupa anche di rinnovare l'ambiente esistente adattandolo al mondo digitale. La modalità 2 si basa sull'esplorazione e sulla sperimentazione per risolvere nuovi problemi e ottimizzare le aree di incertezza.*

4. I QUATTRO FONDAMENTI DELLO SVILUPPO E DEL DEPLOYMENT DELLE APPLICAZIONI CLOUD-NATIVE

Lo sviluppo delle applicazioni cloud-native offre un approccio alla creazione e alla gestione di applicazioni che sfrutta appieno il modello del cloud computing, in base a quattro elementi fondamentali: architettura basata sui servizi, comunicazione basata su API, infrastruttura basata su container e processi DevOps.



ARCHITETTURA BASATA SU SERVIZI

L'architettura basata sui servizi, come i microservizi, comporta la creazione di servizi modulari e a basso accoppiamento. Gli altri approcci modulari all'architettura, ad esempio i miniservizi, che rispettano un design basato sui servizi e a basso accoppiamento permettono alle organizzazioni di accelerare la creazione delle applicazioni senza peraltro aumentare la complessità.



COMUNICAZIONE BASATA SU API

L'esposizione dei servizi avviene tramite API leggere e indipendenti dalla tecnologia impiegata che riducono la complessità e il carico derivanti da deployment, scalabilità e manutenzione. Con le API esposte, le aziende possono creare nuove funzionalità e opportunità, internamente ed esternamente.

Grazie al design basato su API, la comunicazione avviene solo tramite chiamate alle interfacce dei servizi sulla rete, evitando i rischi connessi a collegamenti diretti, modelli di memorie condivise o letture dirette dei datastore di altri team. Questo tipo di progetto permette di ampliare l'ambito delle applicazioni e dei servizi a dispositivi e moduli diversi.



INFRASTRUTTURA BASATA SU CONTAINER

Le applicazioni cloud-native si basano su container per un modello operativo comune tra i vari ambienti tecnologici e per una reale portabilità delle applicazioni tra ambienti e infrastrutture diverse, di tipo privato, pubblico e ibrido. La tecnologia dei container sfrutta le funzionalità di virtualizzazione del sistema operativo per ripartire le risorse elaborative disponibili tra più applicazioni, garantendo al contempo che le applicazioni siano protette e isolate le une dalle altre.

Le applicazioni cloud-native sono scalabili orizzontalmente e l'aggiunta di ulteriore capacità avviene semplicemente aggiungendo ulteriori istanze applicative, spesso mediante l'automazione nel contesto dell'infrastruttura dei container.

Il costo contenuto e l'elevata densità dei container permettono in molti casi di ospitarli nelle stesse macchine virtuali o server fisici, rendendoli la scelta ideale per la distribuzione delle applicazioni cloud-native.



PROCESSI DEVOPS

Lo sviluppo di applicazioni per approcci cloud-native si basa su metodologie Agile con distribuzione continua e principi DevOps incentrati sulla creazione e sul rilascio collaborativi delle applicazioni fra i team di sviluppo, assicurazione della qualità, sicurezza, processi IT e gli altri team che partecipano alla distribuzione.

5. IL PERCORSO VERSO LE APPLICAZIONI CLOUD-NATIVE: 8 PASSI

PASSO 1: TRASFORMARE LA CULTURA E LE PRATICHE DI DEVOPS

Il percorso verso le applicazioni cloud-native comporta per i team che si occupano di sviluppo e di processi IT una profonda trasformazione, per essere in grado di creare e implementare le app 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 e i processi che, in senso lato, permettono di creare una cultura DevOps. Per sfruttare la nuova tecnologia, le metodologie più rapide e la collaborazione più serrata, le organizzazioni devono adottare fino in fondo i principi e i valori culturali del DevOps, ponendoli alla base della loro struttura.

In un'era di rapida innovazione digitale, la scelta del DevOps può risultare impegnativa per alcune organizzazioni che devono affrontare la complessità di gestire più ambienti distribuiti, applicazioni esistenti altamente personalizzate e nuovi carichi di lavoro per le applicazioni. L'estensione delle pratiche DevOps a tutto il portafoglio delle applicazioni presenta un forte potenziale non ancora sfruttato.

L'adozione di una cultura DevOps si basa non solo su strumenti e tecnologie, ma anche sulla volontà e sulla fiducia delle persone, che devono essere disposte ad adottare un approccio più integrato e collaborativo allo sviluppo e alla distribuzione delle applicazioni. La cultura tipica dei progetti software open source può fornire una guida per la costruzione di una cultura DevOps.

I [Red Hat Open Innovation Labs](#) guidano le organizzazioni attraverso il processo DevOps, per incoraggiare la sperimentazione, il superamento rapido delle problematiche, la trasparenza nel prendere decisioni e la distribuzione di riconoscimenti e premi, con lo scopo di aumentare la fiducia e la cooperazione. In questo ambiente favorevole all'innovazione, i team utilizzano tecnologie open source innovative per realizzare velocemente prototipi, adottare strategie DevOps e snellire i flussi di lavoro.

Scopri come i Red Hat Open Innovation Labs possono aiutarti nel tuo viaggio verso il DevOps

[SCARICA L'E-BOOK](#) (in lingua inglese)

"Se non è possibile creare un monolite ben strutturato, come costruire un set ben strutturato di microservizi?"

SIMON BROWN
[CODINGTHEARCHITECTURE.COM/PRESENTATIONS/SA2015-MODULAR-MONOLITHS](https://codingthearchitecture.com/presentations/sa2015-modular-monoliths)

PASSO 2: SNELLIRE LE APPLICAZIONI ESISTENTI UTILIZZANDO MONOLITI VELOCI

All'inizio del loro percorso verso le applicazioni cloud-native, l'attenzione delle aziende non deve concentrarsi solo sul nuovo sviluppo. Per molte applicazioni esistenti, fondamentali per la continuità del servizio e per il fatturato, non è possibile pensare a una semplice sostituzione. Occorre invece integrarle con le nuove applicazioni cloud-native. Ma, come rendere più veloce un monolite esistente? La risposta consiste nell'adottare un approccio basato su monoliti veloci, spostando l'architettura monolitica esistente verso un'architettura più modulare e basata sui servizi e adottando una comunicazione basata su API.

Prima di cimentarsi con il compito oneroso di trasformare le applicazioni monolitiche in microservizi, è quindi necessario creare una solida base per l'architettura monolitica. Le applicazioni monolitiche sono sinonimo di scarsa agilità, una fama dovuta in gran parte al modo in cui sono state costruite. Un monolite veloce può invece offrire molti dei vantaggi di agilità tipici dei microservizi senza però la maggiore complessità e i costi aggiuntivi.

La valutazione di un approccio fondato su monoliti veloci garantisce che le applicazioni si basino 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'eventuale architettura di microservizi, se necessaria. Questo tipo di trasformazione in monoliti veloci pone le fondamenta di un'efficace architettura basata sui microservizi.

Anche se le applicazioni non sono state progettate seguendo un approccio che utilizza monoliti veloci, comunque, è possibile renderle più veloci spostando il monolite esistente su una piattaforma basata su container, accelerando in tal modo il deployment e ottenendo un maggior ritorno sugli investimenti. Le successive integrazioni o funzionalità del monolite potranno poi essere create mediante tecniche e approcci cloud-native.

Adottando un approccio strutturato in più fasi è possibile iniziare a suddividere ogni monolite in componenti di minori dimensioni, secondo le esigenze specifiche.

PASSO 3: USO DI SERVIZI APPLICATIVI PER ACCELERARE LO SVILUPPO

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 fornire tutti i vantaggi, tuttavia, i componenti riutilizzabili per le applicazioni cloud-native devono essere ottimizzati e integrati nella sottostante infrastruttura cloud-native.

Perché 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 quando è possibile utilizzare quelli esistenti, una volta ottimizzati e integrati nell'infrastruttura sottostante basata su container? Questi servizi applicativi, che si tratti di offerte Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) o Integration Platform-as-a-Service (iPaaS), sono strumenti efficaci e pronti all'uso da parte degli sviluppatori.

Nel caso delle applicazioni cloud-native, uno o più di questi tipi di servizi può essere necessario per permettere agli sviluppatori di accelerare i tempi di sviluppo e immettere più rapidamente le nuove applicazioni sul mercato. Se DevOps e container accelerano il deployment e la distribuzione delle applicazioni cloud-native, i servizi applicativi rendono più rapido il loro sviluppo.

Gli sviluppatori di applicazioni cloud-native, ad esempio, possono sfruttare i servizi applicativi creati specificamente non solo per offrire prestazioni ottimizzate in un'infrastruttura basata su container ma anche per trarre vantaggio da funzionalità della piattaforma, quali i flussi CI/CD (Continuous integration/Continuous delivery), il deployment in sequenza e il deployment blue-green, la scalabilità automatica, la tolleranza agli errori e così via.

PASSO 4: SCEGLIERE LO STRUMENTO ADATTO A OGNI ATTIVITÀ

La forte espansione degli studi nel settore del software, con l'Internet of Things, il machine learning, intelligenza artificiale, il data mining, il riconoscimento delle immagini, le auto a guida autonoma e molto altro ha portato a una forte diversificazione di framework, linguaggi e approcci allo sviluppo del software.

La creazione di applicazioni cloud-native sta subendo un processo di differenziazione e la scelta del linguaggio o del framework si adatta in misura crescente alle specifiche esigenze delle applicazioni aziendali. Il conseguente aumento della complessità richiede quindi 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, su design basato su dominio, progetto e sviluppo basati su test, MonolithFirst, monoliti veloci, mini o microservizi, la

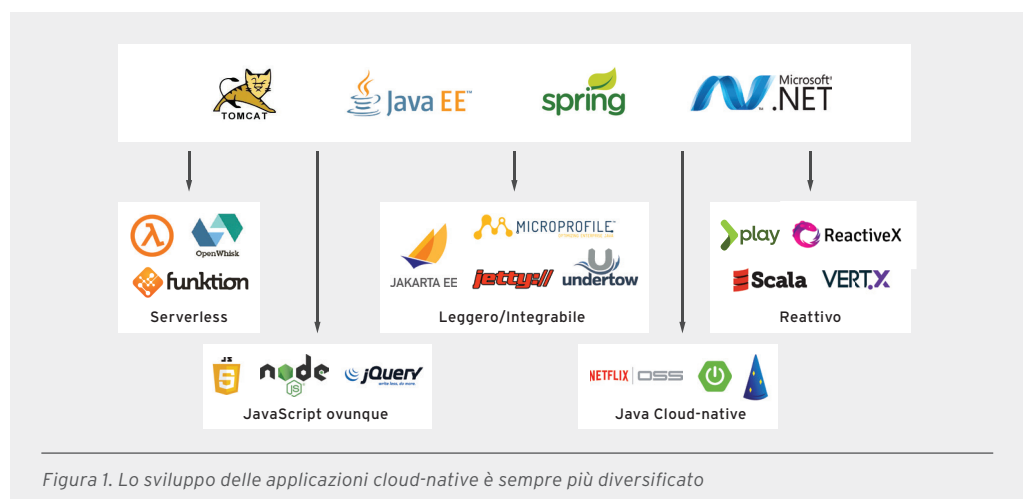


Figura 1. Lo sviluppo delle applicazioni cloud-native è sempre più diversificato

piattaforma cloud-native deve ottimizzare framework, linguaggi e architetture, in modo da soddisfare i requisiti di sviluppo. La sottostante piattaforma containerizzata, inoltre, deve supportare una serie di runtime e framework appositamente predisposti e continuamente aggiornati per adeguarsi alle innovazioni tecnologiche.

PASSO 5: FORNIRE UN'INFRASTRUTTURA SELF-SERVICE E ON-DEMAND

Le metodologie Agile permettono agli sviluppatori di creare e aggiornare rapidamente il software, ma quando è necessario accedere all'infrastruttura è difficile farlo con tempismo ed efficienza. La velocità di accesso al mercato quando si rilasciano le applicazioni in produzione ne risente. In un'era in cui le infrastrutture sono poco costose e il talento tecnico costoso, compilare un ticket e attendere settimane il rilascio di risorse da parte dell'IT non è più sostenibile.

Per evitare il ricorso allo shadow IT non autorizzato, il provisioning delle infrastrutture self-service e on-demand rappresenta 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 sottostante, fornendo affidabili funzionalità di gestione del ciclo di vita delle applicazioni nei vari ambienti infrastrutturali, come datacenter e cloud privati e pubblici. Le piattaforme containerizzate 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 concentrarsi sulla creazione di applicazioni senza gli ostacoli e i ritardi legati al provisioning dell'infrastruttura.

Un altro aspetto importante del modello self-service è la standardizzazione, poiché aiuta le organizzazioni ad automatizzare e distribuire le applicazioni in modo coerente, soddisfacendo gli obiettivi aziendali. La standardizzazione dei processi implica la mappatura dell'esatta sequenza di eventi e attività necessari per realizzare un task, ad esempio il rilascio di un'applicazione in un nuovo ambiente.

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, migrando agevolmente da uno all'altro, ottimizzando i relativi costi e sviluppando un'applicazione multicloud il cui codice non è legato ad alcuna specifica API di un provider cloud.

Scopri le diverse prassi e le tecniche che permettono di realizzare il passaggio al cloud-native.

[SCOPRI LA OPEN PRACTICE LIBRARY](#)

PASSO 6: Automatizzare l'IT per accelerare il rilascio delle applicazioni

L'automazione dell'IT o dell'infrastruttura è un passo essenziale per accelerare il rilascio delle applicazioni cloud-native e l'eliminazione delle attività manuali. L'automazione può essere integrata e applicarsi a qualsiasi task 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 time-to-market. 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 e accelera il time-to-value complessivo.

Linee guida per l'automazione IT

1. Adotta un approccio programmatico all'automazione dei processi IT da estendere all'intera organizzazione. Pratica il dialogo collaborativo a livello di tutta l'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. Fai una riflessione approfondita sull'automazione. Elimina tutte le fasi manuali non necessarie, anche se conservare controlli manuali potrebbe sembrare più sicuro.
4. Affronta l'automazione in modo graduale e sistematico, con passi piccoli e realizzabili. Ogni passo prende forma da quello precedente per creare una pratica di automazione generalizzata.
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. Implementa cataloghi self-service che offrono strumenti agli utenti e accelerano la distribuzione.
7. Adotta policy e processi di misurazione, monitoraggio e chargeback.

Nel tempo otterrai un'automazione integrata full-scale che darà alla tua azienda una maggiore efficienza, DevOps più rapidi e un'innovazione più veloce.

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

[SCARICA L'E-BOOK](#)

PASSO 7: Adottare tecniche di continuous delivery e deployment avanzato

Cicli di rilascio lunghi significano ritardi tra la scoperta e la risoluzione dei bug ma anche ostacoli nel rispondere tempestivamente ai clienti e ai cambiamenti della domanda del mercato. Per le applicazioni con traffico elevato, come quelle mobile, web or IoT, un errore non risolto può avere conseguenze su molti utenti, compromettendo l'esperienza dei clienti, creando problemi di sicurezza e riducendo la produttività o i ricavi. Anche per le altre applicazioni aziendali interne, le interruzioni dei servizi o i ritardi nella risoluzione degli errori software possono comportare costi elevati.

Le metodologie di sviluppo Agile hanno creato un modello che prevede rilasci rapidi e frequenti, mentre gli approcci DevOps e di distribuzione continua hanno ampliato questo modello, mettendo in stretto contatto i team che si occupano 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 circolo iterativo è basato sul feedback veloce ed è reso possibile da CI/CD, dall'estensione dell'automazione dell'infrastruttura a un sistema di distribuzione automatizzato end-to-end che copre tutti gli aspetti della distribuzione delle applicazioni, compresi test automatico, scansione delle vulnerabilità, sicurezza e conformità e verifiche normative. Con i flussi di distribuzione automatizzati l'obiettivo è fornire aggiornamenti senza influire sulla capacità operativa e riducendo i rischi.

Il primo passo per raggiungere la distribuzione continua (continuous delivery, CD) è l'integrazione continua (continuous integration, CI). I sistemi CI sono sistemi di compilazione che monitorano le modifiche nei repository di controllo del codice sorgente, eseguono i test applicabili e creano automaticamente la versione più recente dell'applicazione da ciascuna modifica al controllo del codice sorgente, come Jenkins.

Scopri come le moderne tecnologie di automazione come Red Hat Ansible® Automation supportano CI/CD

[SCARICA IL WHITE PAPER](#)

La distribuzione continua (Continuous delivery, CD) è un approccio all'ingegneria del software in cui i team producono continuamente software in cicli brevi, assicurando che il software possa essere rilasciato in sicurezza in qualsiasi momento. Grazie a rilasci affidabili e a basso rischio, la CD permette di adattare continuamente il software in modo da integrarvi i feedback degli utenti, le variazioni del mercato e i cambiamenti della strategia aziendale.

Definizione di Gartner

"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
[REDHAT.COM/IT/ENGAGE/
TEACHING-AN-ELEPHANT-TO-DANCE](https://redhat.com/it/engage/teaching-an-elephant-to-dance)

I modelli di deployment avanzato tendono 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 per i clienti.

Eliminando la scomodità che i nuovi deployment possono causare ai clienti, queste metodologie permettono di distribuire aggiornamenti e versioni ogniqualvolta ciò sia reso necessario dalle esigenze aziendali. Di seguito sono riportate alcune delle più comuni tecniche di deployment che consentono di eliminare le interruzioni, a seconda degli esempi 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 roll-out. Dopo l'implementazione, 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.

Come rendere agile un'architettura monolitica

[SCARICA L'E-BOOK](#)

PASSO 8: EVOLVERE VERSO 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. A differenza del tradizionale approccio monolitico, in cui ogni componente viene creato in un unico elemento, i microservizi sono costituiti da componenti separati che interagiscono per completare le stesse attività. Poiché particolarmente leggero, questo tipo di approccio allo sviluppo del software promuove la granularità e consente di condividere processi simili tra più app. Benché non sia necessaria una specifica infrastruttura sottostante, la base ideale di un'architettura a microservizi è costituita da una piattaforma containerizzata.

L'evoluzione verso un'architettura che utilizza i microservizi può risultare particolarmente vantaggiosa per team di grandi dimensioni o per i deployment di 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 implementare in modo indipendente ogni microservizio, eventualmente prevedendo vari team responsabili per i diversi cicli di vita.

L'implementazione di un'architettura a microservizi richiede tuttavia investimenti e competenze e può esercitare un forte impatto sull'organizzazione. Gli analisti e gli esperti raccomandano di adottare un approccio **MonolithFirst** ai microservizi, creando prima di tutto un'applicazione monolitica anche se con l'intenzione di implementare un'architettura a microservizi. In tal modo è possibile prima chiarire il dominio dell'applicazione, riconoscendo meglio i contesti delimitati che si prestano ad essere convertiti

in microservizi. Questo modo di procedere evita di creare debito tecnico, dovuto ad esempio a costi di riparazione durante la creazione degli insiemi di microservizi, prima di aver compreso a fondo il dominio e i contesti delimitati dell'applicazione.

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, senza la complessità di un'infrastruttura e di un progetto basati su microservizi. I miniservizi richiedono comunque un investimento in metodologie Agile, DevOps e CI/CD, cosicché la soluzione ideale è un moderno server delle applicazioni o un'offerta multi-framework, multi-architettura e multi-linguaggio combinata con un'infrastruttura containerizzata.

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.

6. BUSINESS CASE DI APPLICAZIONI CLOUD-NATIVE

Le aziende scelgono priorità diverse per le loro attività di trasformazione digitale. Alcune modernizzano l'architettura applicativa esistente e l'infrastruttura di pari passo con l'adozione dei moderni principi cloud-native, mentre altre decidono di innovare con nuovi modelli aziendali e nuove applicazioni. Qualunque sia l'intento e l'utilizzo aziendale, tutte condividono gli stessi obiettivi di maggiore velocità, flessibilità e preparazione per l'era digitale. Gli usi più comuni delle applicazioni cloud-native rispondono in generale alle seguenti quattro sfide aziendali:



"Il primo giorno della nostra acquisizione di una nuova banca, abbiamo immesso in produzione 10 cambiamenti con zero difetti."

JOHN RZESZOTARSKI
RESPONSABILE DEVOPS PRESSO
KEYBANK

Riduzione dei tempi del deployment: da 12 a 1 settimana

Entro il 2023, il 90 % delle applicazioni attuali sarà ancora in uso ma la maggior parte di esse sarà stata oggetto di investimenti insufficienti per la modernizzazione.

Gartner: Application Modernization should be business-centric, continuous, and multiplatform, January 2018.

"Grazie al supporto degli esperti per Red Hat JBoss Enterprise Application Platform, siamo in grado di superare le difficoltà legate alle attività quotidiane."

EX-DIRETTORE GENERALE
TECNOLOGIE AZIENDALI,
AUSTRALIAN SECURITIES
EXCHANGE (ASX)

riavvio delle applicazioni fino a 60 volte più veloce

Riduzione dei costi e del tempo per il supporto della piattaforma, che permette di dedicare più risorse allo sviluppo di servizi innovativi

SFIDA AZIENDALE N. 1: ACCELERARE LA DISTRIBUZIONE DELLE APPLICAZIONI

Obiettivo:

Accelerare la distribuzione ai clienti delle applicazioni nuove ed esistenti.

Approccio:

I container forniscono una piattaforma comune che permette di unificare i team che si occupano di sviluppo, operazioni, sicurezza, QA e altre attività, adottando il DevOps in modo indipendente dall'infrastruttura e dalla tecnologia delle applicazioni. Con un approccio DevOps, i team sfruttano l'automazione e le pratiche CI/CD per rilasciare il software in modo rapido e sicuro. Affrontando i problemi legati al deployment mediante l'automazione basata su container, è possibile accelerare il ciclo di distribuzione delle applicazioni, adattandolo ai tempi aziendali anziché adeguare le attività dell'organizzazione a ciò che l'IT è in grado di fornire.



Il focus del cliente:

KeyBank, una delle 15 banche più importanti degli Stati Uniti, ha avviato un'iniziativa di modernizzazione del canale digitale per rinnovare l'esperienza web e creare una nuova app web mobile. Red Hat OpenShift® ha fornito il supporto necessario alla migrazione dalle applicazioni monolitiche ai microservizi, permettendo a KeyBank di adottare un flusso automatizzato di distribuzione continua, portando i deployment da trimestrali a settimanali.

SFIDA AZIENDALE N. 2: MODERNIZZARE LE APPLICAZIONI ESISTENTI

Obiettivo:

Accelerare il cambiamento **modernizzando le applicazioni esistenti** per adattarsi ai mercati e ai clienti.

Approccio:

Molte importanti applicazioni aziendali sono applicazioni preesistenti, non progettate per l'era digitale. Non sempre, tuttavia, è possibile ed economico eliminarle e sostituirle completamente. Non tutte le applicazioni esistenti, inoltre, si prestano ad essere modernizzate.

Quando la migrazione delle applicazioni tradizionali al cloud è fattibile, questo approccio si basa sull'uso dei container, che permettono di eliminare le dipendenze dall'infrastruttura sottostante. Le applicazioni, di conseguenza, sono portabili da un'infrastruttura on-premise al cloud, dove, se necessario, possono essere rinnovate e la loro architettura modificata in modo da diventare cloud-native. L'approccio basato su una piattaforma costituita da container permette inoltre di sfruttare le funzionalità di automazione e le pratiche DevOps, semplificando la migrazione delle applicazioni esistenti.



Il focus del cliente:

Ogni giorno è il primo mercato finanziario ad aprire e per questo la Borsa australiana (ASX, Australian Securities Exchange) gioca un ruolo di primo piano nel settore internazionale dei servizi finanziari. Un'organizzazione a cui servono stabilità, sicurezza ed efficienza operativa non può permettersi una piattaforma server applicativa ormai inadatta a questi imperativi. ASX ha quindi avviato un'iniziativa per modernizzare la propria piattaforma digitale con le nuove tecnologie, scegliendo Red Hat JBoss® Enterprise Application Platform come base robusta per il proprio server delle applicazioni. Il deployment iniziale ha riguardato una delle applicazioni web business-to-business (B2B) critiche dell'organizzazione, ASX Online, che fornisce prezzi, annunci riguardanti le aziende e report critici al mercato, nel rispetto dei requisiti normativi.

"Red Hat OpenShift Container Platform è stato un'autentica rivelazione. È innovativo e permette di accelerare il deployment, oltre che di controllare facilmente i nostri container."

MICHAEL AALBERS
SENIOR TECHNICAL
APPLICATION COORDINATOR
AEROPORTO DI
AMSTERDAM-SCHIPHOL

sviluppo più veloce del
50% delle nuove API

SFIDA AZIENDALE N. 3: SVILUPPARE NUOVE APPLICAZIONI CLOUD-NATIVE

Obiettivo:

Accelerare lo **sviluppo di nuove applicazioni** per cogliere le nuove opportunità aziendali.

Approccio:

I cambiamenti del business e della domanda dei clienti costituiscono delle opportunità per le organizzazioni che riescono a trasformare rapidamente le idee in servizi e prodotti, valutare i risultati ottenuti alla luce della nuova situazione in cui si trovano e adattarvi. L'approccio cloud-native alla creazione di nuove applicazioni contribuisce ad accelerare il percorso che porta dall'idea iniziale ad applicazioni innovative, grazie al supporto fornito da un'architettura basata su servizi, integrazioni delle API, orchestrazione e servizi containerizzati e automazione, strumenti e prassi DevOps.



Focus del cliente:

L'aeroporto internazionale di Amsterdam-Schiphol è il terzo in Europa, con 64 milioni di passeggeri ogni anno, con l'obiettivo di diventare lo scalo digitale più avanzato al mondo. Per raggiungere questo obiettivo, era necessario rendere più rapido lo sviluppo delle applicazioni mediante l'adozione di una piattaforma indipendente dal cloud. Tra gli aspetti fondamentali su cui si basa strategia digitale di Schiphol, ci sono i servizi forniti tramite API, compresa l'API Flight, che offre ai passeggeri informazioni su gate, terminali e orari del check-in. Grazie a Red Hat OpenShift Container Platform, Schiphol sta creando una piattaforma self-service, multicloud per i propri team IT interni e per i partner aziendali, con una notevole riduzione nei tempi di sviluppo dei nuovi servizi.

"L'aspetto più interessante di questo lavoro è la riprogettazione del modo in cui funziona l'IT. Cambieremo totalmente il modo in cui lavora la nostra azienda [e] inizieremo anche a rinnovare il modo in cui funziona la banca nel suo complesso."

WAYNE MARCHANT
CHIEF INFORMATION OFFICER,
HERITAGE BANK

SFIDA AZIENDALE N. 4: GUIDARE L'INNOVAZIONE AZIENDALE

Obiettivo:

Accelerare l'innovazione in tutta l'organizzazione adeguandola alle richieste del business.

Approccio:

In un mondo in rapido movimento, chi non mantiene il passo è destinato a restare indietro. I team IT lavorano con tempi sempre più serrati per introdurre nuove funzionalità e servizi che soddisfino i clienti e permettano ai dipendenti di lavorare in modo più efficiente. Il successo dipende dall'innovazione costante e non si basa solo su nuovi strumenti e tecnologie. Richiede cultura, strumenti e processi innovativi, incentrati su prospettive rivoluzionarie, in grado di sostenere l'innovazione e la sperimentazione in tutta l'organizzazione.



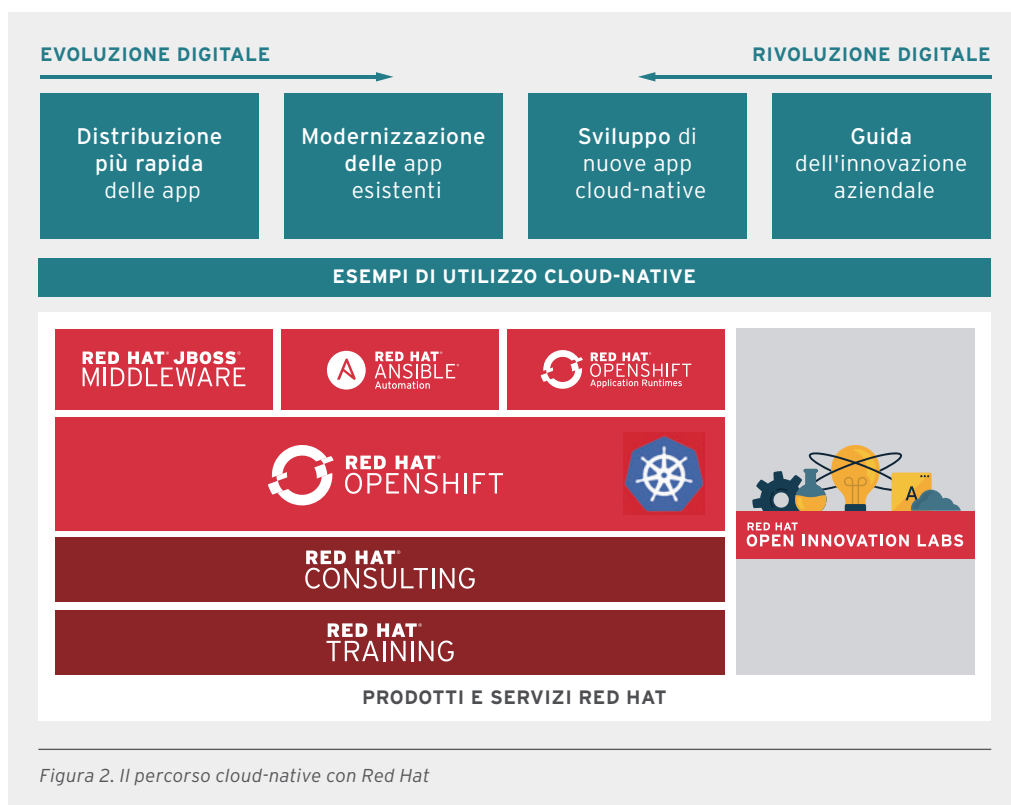
Focus del cliente:

Con i suoi 142 anni, Heritage Bank è una delle istituzioni finanziarie più antiche dell'Australia. Di fronte alla necessità di affrontare una concorrenza crescente e le nuove richieste del mercato, Heritage Bank ha dovuto trovare nuovi modi per distribuire il software in modo più rapido. Grazie alla partecipazione immersiva dei suoi team ai Red Hat Open Innovation Labs, Heritage Bank ha creato una soluzione bancaria innovativa e ha costituito un team ad alte prestazioni, che è in grado di dedicarsi allo sviluppo continuo e più rapido di software migliore, ora e in futuro.

[Video di Heritage Bank \(in lingua inglese\)](#)

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 del punto in cui ti trovi nel percorso digitale e cloud-native.



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

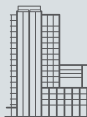
Red Hat ti affianca in questo percorso con [Red Hat OpenShift](#), una piattaforma di sviluppo containerizzata cloud-native. [Red Hat OpenShift Application Runtimes](#) offre framework e runtime open source per la creazione di applicazioni cloud-native, snellendo i tempi di sviluppo grazie a runtime containerizzati su OpenShift. Su OpenShift è possibile implementare numerose tecnologie Red Hat JBoss Middleware, tra cui le [tecnologie di automazione e gestione Ansible](#).

Per aiutarti ad affrontare la complessità della trasformazione digitale, [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.

HAI INIZIATO IL PERCORSO VERSO IL CLOUD-NATIVE?

Scopri come Red Hat può aiutarti nel tuo percorso verso le applicazioni cloud-native:

- Red Hat Consulting può affiancarti, fornendoti le best practice e indicazioni per la pianificazione mediante una [Discovery Session](#) con un consulente.
- Per informazioni, suggerimenti e molto altro, leggi il nostro [Services Speak Blog](#).
- Qual è il tuo livello di maturità DevOps? Sei pronto per il percorso cloud-native? La valutazione [Ready To Innovate](#) ti permetterà di saperlo.



INFORMAZIONI SU RED HAT

Red Hat è leader mondiale nella fornitura di soluzioni software open source. Con un approccio basato sul concetto di community, distribuisce tecnologie come Kubernetes, container, Linux e hybrid cloud caratterizzate da affidabilità e prestazioni elevate. Red Hat favorisce l'integrazione di applicazioni nuove ed esistenti, lo sviluppo di applicazioni cloud-native, la standardizzazione su uno tra i principali sistemi operativi enterprise, e consente di automatizzare e gestire ambienti complessi in modo sicuro. I pluripremiati servizi di consulenza, formazione e assistenza hanno reso Red Hat un partner affidabile per le aziende della classifica Fortune 500. Lavorando al fianco di provider di servizi cloud e applicazioni, system integrator, clienti e community open source, Red Hat prepara le organizzazioni ad affrontare un futuro digitale.



facebook.com/RedHatItaly
twitter.com/RedHatItaly
linkedin.com/company/red-hat

ITALIA
it.redhat.com
italy@redhat.com

**EUROPA, MEDIO ORIENTE,
E AFRICA (EMEA)**
00800 7334 2835
it.redhat.com
europe@redhat.com