



WHITE PAPER

LA GUIDA PER
CIO ALLE
APPLICAZIONI
CLOUD NATIVE

Introduzione

Le tecniche di sviluppo Cloud Native permettono di ottimizzare il funzionamento e la gestione delle applicazioni in ambienti ibridi e Multi Cloud, migliorando le performance, la portabilità, la scalabilità e la velocità di delivery del software.

In questa breve guida rivolta a CIO, CTO e leader IT valuteremo come procedere alla modernizzazione del parco applicativo aziendale grazie al Cloud Native.



Indice

1. IL MODELLO CLOUD NATIVE E GLI IMPATTI SUL BUSINESS	3
La tecnologia	4
La filosofia e il metodo	5
2. MODERNIZZARE LE APPLICAZIONI: TRE APPROCCI	6
Rehosting: la via più rapida	8
Refactoring: la via di mezzo	9
Rearchitecting: la sterzata	10
3. IL RUOLO DI KUBERNETES NELLE APPLICAZIONI CLOUD NATIVE	11
Kubernetes: case study italiani e internazionali	13
Valutare la convenienza del modello Cloud Native	14
4. CONCLUSIONE	15



An abstract graphic on the left side of the page, consisting of multiple layers of wavy, horizontal lines in various shades of blue, creating a sense of depth and movement. The lines are more densely packed and vibrant in the foreground, fading into a darker blue background towards the top and right.

Il modello Cloud Native e gli impatti sul business

La tecnologia

Gli ambienti IT sono sempre più orientati al **modello ibrido** e le applicazioni monolitiche, progettate per girare in architetture client-server, spesso risultano inefficienti. Le **applicazioni Cloud Native**, ottimizzate per funzionare in ecosistemi eterogenei e distribuiti, permettono invece di raggiungere alti livelli di qualità e performance, oltre a garantire flessibilità, portabilità e rilasci più veloci.

Elemento caratteristico delle applicazioni Cloud Native è l'**architettura a microservizi**: ogni software è progettato come un insieme di unità funzionali indipendenti, che comunicano tramite API e possono essere modificate singolarmente senza interferire con il comportamento complessivo dell'applicazione. Insomma, la stessa rivoluzione portata sulle infrastrutture con l'avvento del Cloud

viene trasposta in ambito applicativo. I microservizi possono girare all'interno di **container** (ambienti di runtime completi) ed essere così spostati da un sistema all'altro, indipendentemente dall'hardware sottostante.

Il ricorso a soluzioni **PaaS (Platform-as-a-Service)**, pronte all'uso per semplificare lo sviluppo e la gestione applicativa, è un'altra caratteristica tipica del modello Cloud Native.



Secondo Red Hat, il **56%** dei leader IT intende aumentare l'uso dei container nel 2020.

fonte: *Red Hat*





La filosofia e il metodo

Tuttavia, la questione non è meramente tecnologica. L'approccio Cloud Native è un **fattore abilitante per la trasformazione digitale del business** e non riguarda soltanto il cambiamento dei paradigmi IT. Coinvolge piuttosto l'intera organizzazione aziendale, portando innovazione in processi e opportunità competitive.

Adottare la filosofia Cloud Native per la progettazione applicativa significa innanzitutto effettuare un **cambio di passo culturale e organizzativo**, sia in ambito Information Technology sia in ambito Business Operations.

Ciò implica la capacità di pensare all'evoluzione dei sistemi informativi **mettendo al centro il software e non le infrastrutture**. Sono infatti le applicazioni a fornire i servizi che abilitano i processi di business, gestiscono i dati e creano valore per l'azienda. Bisogna quindi immaginare il futuro di ogni applicazione: quali utilizzi avrà, quanto dovrà scalare, su quali sistemi è meglio che giri, ecc. Su questa base diventa possibile decidere quale approccio

seguire: non tutti gli applicativi infatti necessitano di modernizzazione e in alcuni casi risulta più conveniente mantenere l'architettura tradizionale.

Il modello Cloud Native non va concepito come un punto di arrivo statico, ma piuttosto come **un insieme di best practice in continuo divenire e diverse a seconda dei casi**, che permettono alle applicazioni di ottimizzare funzionalità e prestazioni in ambienti ibridi e Multi Cloud.

Queste pratiche consentono, inoltre, di adottare framework agili per le pipeline di sviluppo e rilascio, facilitando l'adozione di **metodologie DevOps** che abilitano una più rapida e proficua collaborazione tra Developer e Operation. Il risultato è un'**accelerazione dei processi di gestione delle applicazioni** durante l'intero ciclo di vita, che permette di rilasciare più frequentemente e velocemente, soddisfacendo meglio le aspettative degli utenti.

Modernizzare le applicazioni: tre approcci



Al fine di concretizzare il modello Cloud Native e coglierne le opportunità di business, esistono tipicamente **tre approcci che permettono di migrare e modernizzare le applicazioni esistenti** per la fruizione in Cloud:

- il **rehosting** (conosciuto anche come **Lift&Shift**) consiste nella semplice trasposizione dell'applicazione nel Cloud, senza apportare modifiche e previa analisi;
- il **refactoring** prevede interventi al codice sorgente per eliminare le inefficienze dell'applicazione e ottimizzarne l'utilizzo in Cloud;
- il **rearchitecting** introduce all'interno del software alcune proprietà tipiche del modello Cloud Native, attingendo a tecniche di sviluppo moderne e modificando profondamente il codice, ad esempio tagliando il monolite in microservizi verticali.

Andando in crescendo, verso approcci più drastici all'adozione dei modelli Cloud Native, si trovano anche le attività di:

- **rebuilding**, quando l'applicazione da modernizzare viene riscritta totalmente ex-novo secondo le pratiche Cloud Native, preservandone tuttavia le funzionalità;
- **replacing**, se l'applicativo o buona parte di esso viene sostituito direttamente con una soluzione SaaS equivalente per funzioni.

Come si muovono le aziende?

Strategie di gestione delle applicazioni legacy nei prossimi 24 mesi.

Disattivazione	16%
Nessuna azione	31%
Modernizzazione	17%
Rearchitecting in Cloud	16%
Rearchitecting Cloud Native	14%

fonte: *Red Hat - The State of Enterprise Open Source*,



Rehosting: la via più rapida

Il **rehosting** viene considerato la tecnica di modernizzazione **più semplice, economica e meno rischiosa**.

Consiste nel passare le applicazioni dal data center aziendale on-premise alle infrastrutture as-a-service erogate da un Cloud provider esterno.

Sebbene richieda investimenti minimi, è anche la pratica che offre i minori benefici dalla migrazione sul Cloud. Può essere utilizzato in caso di applicazioni legacy che richiederebbero modifiche troppo onerose oppure per **accelerare le tempistiche di migrazione**, qualora necessario. Si pensi ad esempio a un'azienda che abbia necessità di passare rapidamente al Cloud, posticipando eventuali future modifiche. In questo caso il rehosting **potrebbe rappresentare il primo passo di una strategia più a lungo termine**.

Tra i vantaggi, il rehosting offre all'applicazione un guadagno in termini di **sicurezza, resilienza e scalabilità**, proprio perché va a beneficiare delle infrastrutture in Cloud, senza richiedere grandi investimenti di tempo e denaro. Permette anche di ridurre i tempi di elaborazione aumentando le prestazioni grazie ad una più flessibile computing capacity offerta dai Cloud provider.

Il principale punto a sfavore riguarda invece la **mancanza di ritorni sul business**: si ottiene al massimo un risparmio sui costi IT, tra l'altro non sempre dell'ordine auspicato. Inoltre, non si acquisisce **nessun miglioramento della user experience**.

Tra i casi di eccellenza che hanno praticato il rehosting, è da annoverare **Enel, che ha attuato la migrazione Lift&Shift già nel 2015**, quando ha deciso di passare tutte le applicazioni delle IT Operations sulla nuvola.

Pro

- Rapido e poco dispendioso
- Continuità con la soluzione esistente
- Nessuna o poche modifiche alla codebase
- Più sicurezza, resilienza e scalabilità rispetto alla soluzione esistente

Contro

- ROI basso o inesistente
- Nessun miglioramento della user experience
- Non sfrutta al massimo le potenzialità Cloud Native

Refactoring: la via di mezzo

Il **refactoring** può essere considerato il primo autentico step verso l'adozione del modello Cloud Native. Infatti, mentre si correggono le inefficienze (apportando piccole modifiche al codice senza alterare l'architettura e le funzionalità dell'applicazione), c'è **un avvicinamento alle caratteristiche Cloud Native**.

Un approccio diffuso concerne la **containerizzazione**, che migliora le applicazioni sotto il profilo della portabilità in ambienti eterogenei. Le applicazioni infatti vengono incluse e fatte girare all'interno di container e funzionano indipendentemente dall'hardware sottostante. La containerizzazione permette quindi di accelerare **il passaggio a IaaS e l'eventuale dismissione dei data center on-premise**, fungendo eventualmente da passaggio intermedio

prima di optare per un rearchitecting o rebuilding completo. Grazie ai container e ai **software di orchestrazione come Kubernetes**, si ottiene anche una maggiore automazione dei processi di gestione applicativa.

Il refactoring in generale permette un **guadagno di performance**, a fronte però di costi non sempre contenuti.

Pro

- **Migliori performance**
- **Coglie i vantaggi Cloud Native (come flessibilità e automazione)**
- **Abbatte i costi delle infrastrutture on-premise**

Contro

- **Basso impatto sulla user experience**
- **Non permette di generare il massimo valore di business**
- **Costi più elevati del rehosting**



Rearchitecting: la sterzata

Il gradino più alto nei percorsi di modernizzazione applicativa spetta al **rearchitcting**, che sfrutta diversi metodi per **acquisire appieno le proprietà Cloud Native**: ricorso a servizi PaaS, adozione di tecniche di sviluppo Cloud Native come i microservizi e i container, modifiche profonde al codice.

Scomporre le applicazioni monolitiche in microservizi indipendenti restituisce maggiore velocità alla gestione del software. Si riesce a **lavorare contemporaneamente su più feature con team dedicati** e a procedere per piccoli rilasci frequenti e incrementali, stimolando ogni volta i feedback degli utenti. Si possono adottare quindi **pratiche Agili e DevOps** che accelerano la delivery delle applicazioni, rispondono alle esigenze del business e riducono il rischio di non raggiungere gli obiettivi.

Il rearchitcting presuppone massicci interventi di modifica al codice e un **cambio di mentalità** in ottica di collaborazione e revisione dei processi, non sempre scontato. Inoltre, le tecniche di progettazione Cloud Native richiedono **competenze specifiche**, spesso assenti in azienda.

Netflix offre un esempio eccellente di rearchitcting. Dopo avere effettuato la completa migrazione dei sistemi informativi dal data center proprietario al Cloud di Amazon Web Services, l'azienda ha deciso di adottare l'architettura a microservizi e le tecniche di containerizzazione per le proprie applicazioni, con l'obiettivo di accelerare i processi di sviluppo, rilascio e distribuzione.

Pro

- Introduce tecnologie e framework Cloud Native (come ad esempio GKE)
- Maggiore portabilità
- Alte performance ed efficienza

Contro

- Necessita di un completo cambio di prospettiva (DevOps e Agile)
- Richiede skill non sempre presenti in azienda



Il ruolo di Kubernetes nelle applicazioni Cloud Native

Come ribadito, la containerizzazione gioca un ruolo primario nel passaggio al modello Cloud Native, aumentando la portabilità del software all'interno di ecosistemi hybrid eterogenei e **scongiurando il rischio di lock-in**.

Tuttavia, il ricorso ai container (che girano su server diversi e possono raggiungere l'ordine delle migliaia) **aggiunge una complessità significativa alla gestione del portafoglio di applicazioni**, soprattutto se l'azienda ha optato per un'adozione estesa dei microservizi. Più "pezzi" si hanno da gestire, maggiore sarà il grado di difficoltà nel mantenere governance, sicurezza e velocità, soprattutto in ambienti distribuiti.

Grazie alle **piattaforme di orchestrazione Open Source come lo standard de facto Kubernetes**, le operazioni di amministrazione e integrazione dei



container possono essere standardizzate e automatizzate, attivando cicli di test e rilascio continui nel rispetto delle pratiche DevOps.

Diventa inoltre possibile la gestione di un ampio parco di applicazioni attraverso un ecosistema altamente distribuito, con **allocazione delle risorse dinamica** ed efficiente che migliora i livelli di disponibilità del software.

Introducendo l'automazione, Kubernetes **libera risorse umane** che, dispensate dalle operazioni di gestione, possono concentrarsi sulle attività core. Diminuiscono quindi i costi e gli errori di amministrazione IT, a vantaggio dell'innovazione. Insomma, **Kubernetes aiuta a gestire ed in qualche modo a ridurre la complessità** insita nell'implementazione dei container e nella gestione del ciclo di vita delle applicazioni, ma è tuttavia una **macchina estremamente articolata** e richiede **competenze specifiche**.

La curva di apprendimento per l'utilizzo dei container e degli orchestratori è ripida e Kubernetes potrebbe non offrire i ritorni auspicati qualora non ci si avvalga di un supporto qualificato: i progetti di successo infatti presuppongono una solida revisione dei processi di sviluppo di applicazioni, attività di **change management** e conoscenze tecniche specializzate spesso non disponibili in azienda.



Il **78%** degli intervistati per il CNCF survey 2019 ha dichiarato di usare Kubernetes, rispetto al **58%** del 2018.

fonte: *CNCF survey 2019*



Kubernetes: case study italiani e internazionali

Kubernetes si è rivelata la scelta vincente per moltissime organizzazioni in tutto il mondo.

Utilizzano l'orchestratore Open Source, infatti, diverse multinazionali dell'era digitale come [Airbnb](#), che ha containerizzato oltre 250 applicazioni critiche accelerandone la distribuzione, e [Shopify](#), che ha riscritto l'intero parco delle applicazioni in ottica container per la migrazione sul Cloud di Google.

Esistono anche diversi casi di aziende italiane. Ad esempio, per supportare l'espansione sul mercato cinese, la [multinazionale chimico-farmaceutica Zambon e Caleffi Hydronic Solutions](#) hanno costruito ecosistemi Multi Cloud appoggiandosi alle piattaforme di Google e Alibaba. Grazie alla containerizzazione e a Kubernetes, le applicazioni possono essere spostate tra i diversi ambienti, in modo trasparente e con alti livelli di efficienza.

Valutare la convenienza del modello Cloud Native

Quando bisogna decidere per la modernizzazione applicativa e l'adozione dell'approccio Cloud Native, le considerazioni da mettere sul piatto sono diverse e occorre tenere sempre presente che **non esiste una soluzione univoca** a coprire tutte le casistiche.

A volte non conviene investire in costose operazioni di rearchiteting, soprattutto se le applicazioni coinvolte non sono critiche, non richiedono particolari livelli di performance o scalabilità, oppure potrebbero essere dismesse in periodi relativamente brevi. È preferibile invece optare per **soluzioni Lift&Shift**, che possono essere viste anche come uno step iniziale non definitivo per accelerare la migrazione al Cloud e la trasformazione dell'ecosistema IT aziendale, prima di procedere a modifiche più pesanti sul codice e sull'architettura software.

Il refactoring può essere un buon compromesso per risolvere rapidamente le inefficienze o gli errori bloccanti che le applicazioni possono presentare quando

vengono migrate al Cloud: lo sforzo economico e l'impiego di risorse è minimo, ma i vantaggi del modello Cloud Native vengono acquisiti solo parzialmente.

Da non scartare neppure l'alternativa del rebuilding (molto costosa e time-consuming, ma preferibile se gli interventi sull'applicazione risultassero eccessivi e di dubbio successo). Infine, per applicazioni commodity o comunque senza la necessità di personalizzazioni, **il ricorso a soluzioni SaaS già pronte è un espediente più che valido**.

Da non sottovalutare anche il **tema delle competenze**: il modello Cloud Native richiede conoscenze specifiche e sarebbe inutile tentarne l'adozione se in azienda mancassero expertise e know-how. Meglio allora optare per una semplice migrazione Lift&Shift oppure per il **supporto di un partner** referenziato e competente.



Conclusione

Bisogna ricordare che l'approccio Cloud Native presuppone anche un **profondo cambiamento culturale e organizzativo**, per massimizzarne i ritorni attraverso una sapiente adozione delle pratiche Agili e DevOps. Anche in questo caso, la consulenza di specialisti del settore può risultare l'opzione vincente.

Insomma, affinché la modernizzazione applicativa verso il modello Cloud Native non si riduca a un mero esercizio tecnico, ma piuttosto diventi un fattore concreto di trasformazione del business, occorre una **disamina attenta del contesto IT, dei processi aziendali e delle risorse in campo**.



