SINTHERA.COM
WeAreProject

Microservizi e container

Guida alle tecnologie che rendono agile il tuo business

ABSTRACT

Compiendo la trasformazione digitale, imprese e organizzazioni d'ogni genere hanno oggi l'opportunità d'innovare i propri modelli di business, e generare più valore, sviluppando nuovi canali e servizi che potenziano le interazioni con utenti interni, partner, clienti. La condizione è però ammodernare l'architettura applicativa, e renderla più agile, migrando verso un paradigma cloud-native, in cui microservizi e container sono pilastri chiave: questa guida spiega cosa sono, e come possono essere utili per attuare una strategia di modernizzazione applicativa.

INDICE

INTRODUZIONE

Innovare di continuo prodotti e servizi, ed elevarne la qualità, è sempre stata la strategia al cuore di ogni attività imprenditoriale determinata a differenziarsi e distinguersi sul mercato rispetto alla concorrenza: oggi, però, la **trasformazione digitale accelera fortemente il ritmo dell'innovazione**, grazie alla disponibilità di tecnologie e paradigmi di gestione dell'IT, come cloud, microservizi, software container, API (application programming interface). Metodi e strumenti che possono permettere a un'impresa di sviluppare modelli di business originali, basati su applicazioni e servizi digitali ancora una volta capaci di ridefinire livelli e standard di competitività aziendale, e di aprire la strada a nuove opportunità e spazi di mercato.

Riuscire a compiere questa trasformazione comporta però, da parte delle diverse organizzazioni, la necessità non solo di sviluppare nuove competenze su tali metodi e strumenti, ma anche l'**abilità di armonizzarli e integrarli con l'infrastruttura e l'architettura IT preesistente**, su cui le imprese hanno investito per anni. Un'infrastruttura che, nella maggior parte dei casi, gestisce ancora molti processi core dell'organizzazione, e va salvaguardata, ma al contempo potenziata per rispondere a nuovi requisiti di agilità, disponibilità, affidabilità, prestazioni, versatilità, senza i quali le moderne applicazioni e servizi digitali non potrebbero adeguatamente funzionare, per fornire a utenti finali e consumatori esperienze di utilizzo di qualità sempre maggiore.

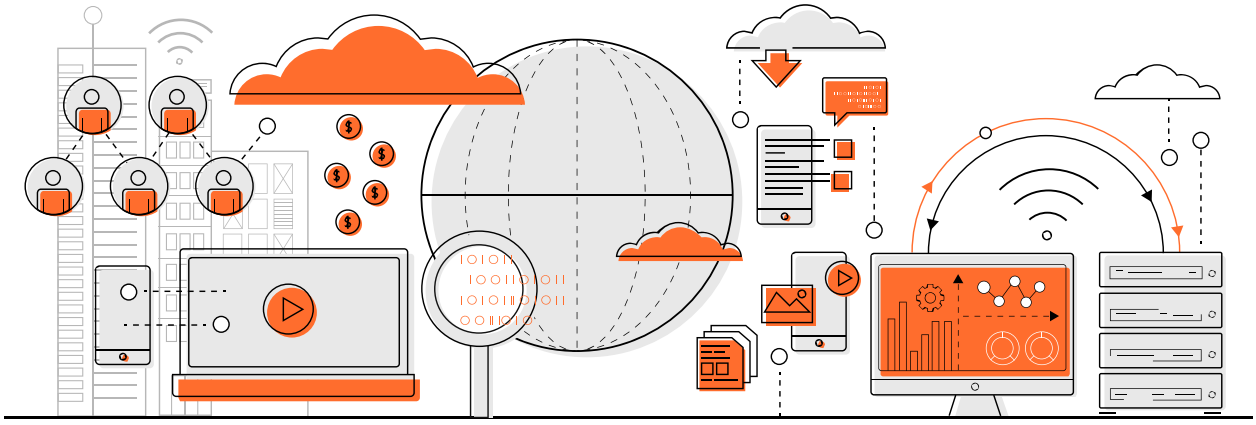
DEVSECOPS PER LA SICUREZZA “BY DESIGN” NEI PROCESSI

Poiché oggi la **security e la privacy dei dati** sono aspetti vitali nel disegno delle nuove applicazioni e nel delivery agli utenti finali, è importante considerare questi aspetti all’interno della catena DevOps. Con la metodologia **DevSecOps** (DevOps + Security) gli aspetti critici del controllo sulle vulnerabilità del codice e delle modalità d’erogazione dei servizi vengono considerati fin dalle fasi iniziali dello sviluppo, aiutando l’integrazione anche delle competenze. **DevSecOps** fa in modo che la sicurezza dei dati non faccia da collo di bottiglia nel ciclo di sviluppo e di rilascio DevOps, permettendo a tutti i membri dei team di condividere l’impegno nella tutela dei dati e trarre grande vantaggio dall’automazione. L’automazione permette di segnalare al programmatore parti di codice scritte al di fuori delle best practices di security e migliorare il proprio lavoro, integra i controlli nei test, consente alle segnalazioni di anomalie di risalire la catena di produzione, essere tradotte velocemente in azioni di rimedio sui sistemi o sviluppo di patch.

INFRASTRUTTURA IT TRADIZIONALE: GLI OSTACOLI CHE FRENANO UN BUSINESS MODERNO

A livello globale, l'esigenza di ammodernare l'infrastruttura IT, renderla più semplice da gestire, agile nel provisioning delle risorse, accomuna tutte le organizzazioni. In un'analisi del 2018, la società di ricerche di mercato IDC già rilevava il riconoscimento, da parte dei dirigenti d'impresa, che la trasformazione digitale sia da considerare un investimento a lungo termine: e l'aggiornamento di maggio 2020 della "Worldwide Digital Transformation Spending Guide" conferma che la spesa per la trasformazione digitale di attività commerciali, prodotti e organizzazioni continuerà a passo sostenuto, nonostante le sfide presentate dalla pandemia da COVID-19. La spesa globale in tecnologie e servizi per la digital transformation è prevista crescere del 10,4% nel 2020, raggiungendo 1.300 miliardi di dollari: e tale percentuale, pur essendo più bassa in confronto al 17,9% di crescita registrata nel 2019, commenta IDC, resta una delle poche luci in un anno caratterizzato da drastiche riduzioni della spesa tecnologica complessiva.

Come piattaforma abilitante per la trasformazione digitale, anche il cloud continua a espandersi: i servizi di infrastruttura cloud, nel primo trimestre 2020, secondo i dati della società di analisi **Canalys,** sono cresciuti del 34%, toccando 31 miliardi di dollari: una crescita guidata da un incremento nella domanda di strumenti di collaborazione online, servizi di e-commerce, servizi cloud per i consumatori, in risposta alle situazioni di lockdown. Prima dell'attuale crisi sanitaria, in Italia, nel 2019, secondo l'Osservatorio Cloud Transformation della School of Management del **Politecnico di Milano**, il mercato cloud ha toccato 2,77 miliardi di euro, crescendo del 18% rispetto al 2018. L'82% delle organizzazioni, secondo lo studio, è ormai consapevole che **il cloud consente di ottenere una maggiore agilità dell'IT aziendale**.



SILOS INFORMATIVI E RIGIDITÀ DI GESTIONE DELL'IT

Inoltre, tali applicazioni legacy sono in genere caratterizzate da un'**architettura monolitica** e strettamente legate all'hardware on-premise su cui vengono installate. Questa infrastruttura IT, oltre ad avere richiesto elevati investimenti iniziali in hardware e software, oggi si rivela **troppo costosa e complessa da aggiornare e manutenere**, soprattutto in rapporto all'attuale velocità di cambiamento delle esigenze imprenditoriali.

In aggiunta, **negli ambienti IT tradizionali, il ciclo di sviluppo software segue un ordine sequenziale** (waterfall) e un approccio che non razionalizza e velocizza la progettazione del codice, e non facilita l'individuazione di bug e difetti del software subito nelle prime fasi del processo di progettazione.

APPLICAZIONI MONOLITICHE, ALCUNI LIMITI CRITICI

Le applicazioni legacy con architettura monolitica, in cui database, logica di business, interfaccia utente ed altri componenti software sono strettamente integrati e interconnessi tra loro all'interno di un solo, compatto blocco di codice, apportare modifiche o sviluppare nuovo codice risulta complesso: e ciò perché una modifica, anche di un singolo componente, può avere implicazioni sugli altri.

Soprattutto quando la base di codice dell'applicazione aumenta di volume e raggiunge grandi dimensioni, per il team di sviluppo diventa più complicato comprendere completamente tutti i meccanismi e interrelazioni del sistema. In un'applicazione con architettura monolitica, anche quando l'aggiornamento o lo sviluppo di nuove funzionalità riguarda un solo componente, occorre rieseguire il deployment e il test dell'intera applicazione, prima di poterla rilasciare di nuovo nell'ambiente di produzione.

Per tutte queste ragioni, le grosse applicazioni caratterizzate da architettura monolitica non favoriscono l'adozione nel reparto IT delle moderne metodologie di sviluppo **DevOps** e delle pratiche CI/CD (continuous integration/continuous delivery/continuous deployment), che hanno l'obiettivo di accelerare la quantità e la frequenza dei rilasci software, per rendere più rapidi e agili gli aggiornamenti delle applicazioni e l'aggiunta delle nuove funzionalità e servizi.

La natura monolitica del software si riflette in maniera negativa anche sulla sua **scalabilità**, perché la stretta interdipendenza dei componenti software rende arduo scalare un singolo componente in maniera indipendente. Anche sul piano dell'affidabilità di funzionamento, in un'applicazione monolitica il verificarsi di un guasto in un solo modulo o componente può determinare il malfunzionamento o il blocco dell'intera applicazione.

PERCHÉ MIGRARE VERSO UN'ARCHITETTURA BASATA SU MICROSERVIZI E CONTAINER

Tutte le organizzazioni oggi hanno l'esigenza di **modernizzare l'IT e creare nuovi canali digitali per interagire** con i propri dipendenti, partner, clienti, i**n modo sempre più completo, continuo, coinvolgente**: le applicazioni e servizi digitali di ultima generazione definiscono tuttavia standard e **requisiti di operatività**, in termini ad esempio di disponibilità del dato, prestazioni di funzionamento, scalabilità, affidabilità, che le applicazioni legacy con architettura monolitica non sono progettate per supportare. Si pensi, ad esempio, alla necessità di sviluppare un'applicazione per il canale mobile, in cui gli utenti, tramite una app sullo smartphone, devono avere la possibilità di visitare un sito di commercio elettronico 24 ore su 24 per sette giorni alla settimana; sfogliare cataloghi prodotti, acquistare offerte di prodotti o servizi correlati, eseguire transazioni in tempo reale, e ritrovare i dati del proprio account sempre sincronizzati in modo coerente con gli altri canali d'accesso (sito web, punti vendita offline, canali social). Lo stesso ragionamento, sui requisiti necessari, lo si potrebbe ripetere per una moderna mobile app nel mondo banking, o in molti altri settori.

"CLOUD-NATIVE", UN MODELLO DI SVILUPPO PIÙ AGILE

Nello scenario descritto, è evidente che adottare le classiche strategie d'integrazione, connettendo di volta in volta i **sistemi di back-end legacy** con i **nuovi canali digitali** può comportare un notevole effort in termini di attività di sviluppo software. Senza poi considerare che tali sistemi IT aziendali, in caso di picchi di traffico, potrebbero collassare sotto il carico di un numero di richieste API che non sono stati progettati per supportare. Inoltre, più cresce il numero d'integrazioni e connessioni ai sistemi di back-end dell'organizzazione, più aumentano i rischi di violazione della sicurezza.

Una strategia di migrazione verso un'**architettura cloud-native** può rispondere a queste problematiche: "Le tecnologie cloud-native - spiega la **Cloud Native Computing Foundation** (CNCF) - permettono alle organizzazioni di **costruire e gestire applicazioni scalabili in ambienti moderni e dinamici, come i cloud pubblici, privati, ibridi**". Tra le tecnologie cloud-native, CNCF include i **microservizi, i container, le API.** L'utilizzo dei container in ambienti di produzione è cresciuto in maniera significativa, secondo la survey condotta da **CNCF** in Europa, Nord America, Asia, a settembre e ottobre 2019, sondando le opinioni di funzioni aziendali di punta, tra cui software architect (41%), DevOps manager (39%) e sviluppatori back-end (24%), ed elaborata sulla base di 1.337 risposte: **l'84% dei rispondenti sta usando i container in produzione**, con un incremento del 15% rispetto al 2018 (73%). Inoltre, il 78% dei rispondenti adotta in produzione Kubernetes, un software per l'orchestrazione di container: una percentuale che cresce parecchio rispetto al 58% dell'anno precedente.

MICROSERVIZI: COSA SONO, E PERCHÉ SAPERLI USARE

Lo stile architetturale basato su **microservizi**, come spiega anche l'esperto di sviluppo software Martin Fowler, "è un approccio allo sviluppo di una singola applicazione come una suite di piccoli servizi, ciascuno in esecuzione nel proprio processo e in grado di comunicare attraverso meccanismi leggeri". Questi ultimi sono spesso il **protocollo HTTP e le API**.

Essendo basata su un insieme di microservizi funzionanti in maniera indipendente gli uni dagli altri, un'**architettura a microservizi** si differenzia radicalmente dall'architettura monolitica, rispetto alla quale presenta diversi vantaggi, di cui possono beneficiare sia il reparto IT, sia i responsabili del business:

- con un'architettura a microservizi **il ciclo di sviluppo software può accelerare e divenire più agile**, perché diventa possibile progettare un'applicazione, anche complessa, come un insieme di microservizi. Ciascun microservizio è dedicato a una singola funzione, è sviluppabile in modo indipendente, ed aggiornabile con maggior facilità, senza implicare modifiche sugli altri. Potendo lavorare in parallelo sulla progettazione di differenti microservizi della stessa applicazione, i team di sviluppo riescono a migliorare la produttività, riducendo il time-to-market per il rilascio di nuovi prodotti e applicazioni digitali.

- Dal punto di vista dell'adozione delle moderne pratiche DevOps, i microservizi favoriscono, da parte degli sviluppatori, l'utilizzo dei metodi e strumenti CI/CD, che generano automazione, instaurando una pipeline di integrazione, rilascio e deployment continuo del codice: quest'ultima, pur facilitando gli aggiornamenti e l'aggiunta continua di nuove funzionalità secondo i requisiti dettati dal mercato e dagli utenti, salvaguarda al contempo la qualità del software, **anticipando l'identificazione e correzione di bug ed errori** già nelle fasi iniziali del ciclo di sviluppo.

- Usando un'**architettura a microservizi**, il team di sviluppo non è obbligato a selezionare una determinata tecnologia, ma può scegliere di volta in volta quella giudicata più consona allo sviluppo delle specifiche funzionalità. Inoltre, i microservizi si possono immaginare come mattoncini che, una volta sviluppati, possono essere anche riutilizzabili in applicazioni differenti.

- In termini di **scalabilità**, ciascun microservizio può essere scalato in maniera indipendente dagli altri, utilizzando risorse disponibili su ambienti IT differenti (sistemi on-premise, cloud).

- Anche a livello di **affidabilità** di funzionamento dell'applicazione, se un microservizio accusa un'avaria, un rallentamento, o si blocca, ciò non pregiudica in alcun modo l'operatività del resto dell'applicazione.

SOFTWARE CONTAINER E MICROSERVIZI, UN BINOMIO PER ALLINEARE IT E BUSINESS

I **container** rappresentano una tecnologia software molto utile nello sviluppo di microservizi: **un singolo microservizio può infatti funzionare all'interno di un software container, in grado di comunicare con gli altri tramite API**. In sostanza, un software container, si può immaginare come un contenitore logico che, applicando una funzionalità di virtualizzazione del sistema operativo, è in grado di eseguire un processo, o un insieme di processi, in uno spazio isolato, rispetto al resto del sistema operativo host. Un software container può anche impacchettare un'applicazione assieme a tutti i componenti (runtime, codice binario, librerie, dipendenze, eseguibili, file di configurazione) necessari al suo funzionamento. Questa capacità di impacchettamento e isolamento di processi e componenti rende simili i container alle macchine virtuali (VM), dalle quali però si differenziano per una importante caratteristica: invece di virtualizzare l'hardware della macchina fisica, **i container virtualizzano il sistema operativo, condividendone il kernel**, e non hanno quindi necessità d'incapsulare un'immagine completa dello stesso, come invece fanno le VM. Da questa differenza chiave, derivano alcuni benefici fondamentali:

- Un software container ha un'**immagine molto più piccola**, dell'ordine dei megabyte (MB) rispetto a quella di una macchina virtuale, grande anche diversi gigabyte (GB). Essendo più 'leggero', un software container **si avvia più rapidamente, e permette anche di ottimizzare l'utilizzo delle risorse hardware** (storage, memoria).

- Con i container le **attività di aggiornamento e manutenzione vanno eseguite su un solo sistema operativo**. Inoltre, nelle pratiche CI/CD, è possibile creare e rilasciare le immagini container con alta frequenza in modo semplice.

- Condividendo il kernel del sistema operativo, **i container hanno una portabilità, in ambienti di sviluppo e produzione, maggiore rispetto alle VM**, e possono funzionare su macchine fisiche, virtuali, cloud privati, pubblici o ibridi.

Nel settore, lo standard industriale che si è imposto de facto sul mercato per la costruzione e condivisione di applicazioni containerizzate in qualunque ambiente IT (desktop, server, cloud) è la tecnologia **Docker.** Il Docker Engine è il motore runtime di Docker, **in grado di funzionare su svariate distribuzioni Linux** (CentOS, Debian, Fedora, Oracle Linux, RHEL, SUSE, Ubuntu) **e su sistemi operativi Windows Server**.

I container permettono di costruire, distribuire ed eseguire applicazioni e servizi IT in produzione in manie rapida e agile: tuttavia, quando i container che costituiscono l'applicazione sono molti, e vanno tutti gestiti, non si può procedere manualmente ed è **fondamentale introdurre automazione**, ad esempio **per riavviare i container interrotti e garantire la continuità del servizio**. Per rispondere a questa necessità è stato sviluppato un **software open source** come **Kubernete**s, una **piattaforma per l'orchestrazione di container ch automatizza il deployment e la gestione di applicazioni in container**, controllando e amministrando requisiti funzionamento chiave, tra cui scalabilità, failover, resilienza dell'applicazione stessa.

MIGRARE VERSO IL MODELLO CLOUD-NATIVE CON MICROSERVIZI E CONTAINER

Le strategie di **migrazione applicativa verso il cloud dipendono dalla tipologia di software applicativo e workload**, e non richiedono necessariamente l'utilizzo di microservizi o container: l'approccio **rehosting** ("lift-and-shift"), ad esempio, prevede lo spostamento dell'applicazione selezionata, dal data center on-premise, sul cloud, così com'è, senza apportare alcuna modifica al codice. Per farlo si può installarla in una VM, da trasferire poi nella nuvola. La soluzione lift-and-shift ha il vantaggio di non richiedere un effort di sviluppo, ma anche il limite di non beneficiare appieno delle funzionalità cloud-native, ad esempio in termini di scalabilità. All'estremo opposto, scegliendo il **rebuilding**, si può riscrivere l'applicazione monolitica da zero, scomponendola in microservizi, ma tale strada, oltre a costare molto lavoro di sviluppo, espone a elevati rischi di errori di progettazione e malfunzionamenti.

Un altro approccio, meno estremo, il **refactoring**, prevede la riprogettazione solo di parte del codice dell'applicazione, con l'obiettivo di adattarla a sfruttare il paradigma e le funzionalità cloud-native. Anche il refactoring è comunque un processo complesso e lungo da compiere. Una via di mezzo tra il rehosting e il refactoring è il **replatforming**, che preserva le funzionalità esistenti, richiedendo solo modifiche minime dell'applicazione, per metterla in condizione di beneficiare dei vantaggi del cloud: tipicamente, ad esempio, è possibile modificare la modalità d'interazione dell'applicazione con il database on-premise, connettendola con un'infrastruttura database sul cloud, in grado di fornire prestazioni e scalabilità più elevate.

Al di là della strategia scelta in ogni caso particolare, un principio generale comunque valido è **convertire con gradualità i moduli dell'applicazione monolitica in microservizi, avendo sempre chiare le esigenze aziendali**, e ciò che è realmente importante **rendere più agile e scalabile, in funzione degli specifici obiettivi di business**.

CONCLUSIONE

Come illustrato in questa guida, **microservizi e container sono strumenti utili per modernizzare e rendere più agile l'infrastruttura IT tradizionale**, e consentono di farla migrare, anche con gradualità, verso un paradigma di gestione cloud-native, che non pregiudica la salvaguardia degli investimenti tecnologici preesistenti.

PER RICEVERE ULTERIORI INFORMAZIONI

CONTATTACI

MYSINTHERA.COM
WeAreProject