

I container: tutto quello che ogni sviluppatore di applicazioni aziendali deve sapere

Sebbene gli standard del cloud computing la considerino sostanzialmente obsoleta, la tecnologia dei container è recentemente emersa come motore fondamentale della trasformazione digitale per molte organizzazioni in tutto il mondo. Per capire questo cambiamento, esaminiamo cosa sono effettivamente i container e perché vengono promossi dagli sviluppatori di applicazioni aziendali.

Che cosa sono i container?

Un container è un'unità standardizzata che riunisce tutti i componenti e le dipendenze di un'applicazione, inclusi il codice, il runtime, i file binari e le librerie. Ne risulta un pacchetto logico e chiaro, facile da gestire e spostare da un ambiente di elaborazione a un altro. Dato che tutto ciò che serve all'esecuzione è incluso, lo sviluppatore ha un'applicazione con aspetto e comportamento coerenti, che si trovi su un laptop, su un server di staging, in un centro dati privato o sulla cloud pubblica.

I container sono spesso visti come possibile alternativa alle macchine virtuali, anche loro in grado di isolare le risorse per le applicazioni. Mentre le macchine virtuali virtualizzano l'hardware sottostante, i container eseguono la virtualizzazione a livello di singolo sistema operativo, laddove container multipli funzionano direttamente su un kernel di sistema operativo condiviso. Ciò si traduce in una maggiore velocità di avvio e minore utilizzo di memoria da parte dei container rispetto alle macchine virtuali, che invece devono ogni volta riavviare un sistema operativo completo.

In che modo i container stanno modificando lo sviluppo delle applicazioni aziendali?

Per gli sviluppatori di applicazioni aziendali, i container rappresentano una rivoluzione in termini di produttività, flessibilità e valore di business. Con i container gli sviluppatori possono creare ambienti prevedibili, isolati da altre applicazioni, ed elaborare ipotesi che rimarranno valide in produzione. Un container può incapsulare le dipendenze software necessarie all'applicazione, quali ad esempio specifiche versioni di runtime del linguaggio di programmazione e altre librerie software, in modo che le applicazioni rimangano coerenti indipendentemente dalla destinazione di implementazione. Di conseguenza, sviluppatori e squadre operative IT impiegano meno tempo nel debug e nella diagnosi delle differenze nei diversi ambienti, avendo così a disposizione più tempo per realizzare nuove funzionalità basate sul feedback degli utenti.

Questi sono solo i primi di molti vantaggi a venire, motivo per cui gli sviluppatori si sono incaricati di diffondere l'adozione su larga scala della tecnologia dei container. Sembra che il loro obiettivo sia stato raggiunto: secondo un [sondaggio di NGINX](#), nel 2015 due terzi delle organizzazioni hanno sperimentato l'utilizzo dei container, in ambito di sviluppo e di produzione.

Dove possono essere eseguite le applicazioni basate su container?

Una delle caratteristiche più interessanti delle applicazioni basate sui container è la capacità di funzionare potenzialmente ovunque: su sistemi operativi Linux, Windows e Mac; su macchine virtuali o bare metal; sui computer degli sviluppatori oppure in data center on-premise e, naturalmente, sulla cloud pubblica. Per poter accedere alla gamma completa di possibilità, però, ed evitare di rimanere vincolati al vendor, gli sviluppatori necessitano di una soluzione di orchestrazione che sia altrettanto flessibile quanto i container stessi.

Questa è l'idea di fondo di [Kubernetes](#), una tecnologia open source sviluppata da Google per automatizzare l'implementazione, la scalabilità e la gestione dei container. Basato sulla stessa tecnologia che Google usa per lanciare miliardi di container ogni settimana, Kubernetes permette agli sviluppatori di creare, testare ed eseguire davvero ovunque applicazioni in container. Dovendo dedicare meno tempo alle attività di routine, gli sviluppatori possono scegliere la praticità e l'efficienza di un ambiente di orchestrazione completamente gestito, come quello di [Google Container Engine](#). In alternativa, possono mantenere un livello di controllo superiore utilizzando Kubernetes per implementare le loro applicazioni autonomamente, nell'infrastruttura on-premise o su qualunque provider di cloud pubblica.

Da chi vengono utilizzati i container?

Molte delle ultime e più valide applicazioni sfruttano l'agilità di sviluppo dei container. Nella più grande implementazione di Google Container Engine vista finora, [Niantic ha lanciato Pokémon GO](#) per un pubblico di giocatori impazienti sparsi in tutto il mondo. Utilizzando i container, Niantic è stata in grado di concentrarsi sulla distribuzione continua di miglioramenti ai giocatori, anche più volte al giorno. Sia l'orchestrazione sia la scalabilità del cluster di container di Pokémon GO a livello planetario sono state gestite completamente da Kubernetes e Google Container Engine. Allo stesso modo, [CCP Games](#), una società islandese che sviluppa giochi, ha utilizzato Kubernetes per ripulire 20 anni di codice legacy e automatizzare i flussi di lavoro offrendo così una maggiore efficienza a tutto il team.

I piccoli team innovativi all'interno delle grandi aziende possono imparare da società agili come Niantic e CCP Games. I container possono aiutarli a focalizzarsi completamente sulle loro applicazioni e i loro utenti, lasciando ad altri i dettagli della scalabilità e del mantenimento dell'infrastruttura.

“Container Engine offre un'infrastruttura flessibile e a elevate prestazioni che consente di scalare i componenti in modo indipendente per ottenere la massima efficienza.”

—George Yianni, inventore di Hue e direttore dell'illuminazione connessa per la domotica presso [Philips Lighting](#)

Da dove posso iniziare?

Per testare [Kubernetes](#) o [Google Container Engine](#), [iscriviti gratuitamente alla prova su Google Cloud Platform](#) e [procedi nel seguente modo](#). In alternativa, [divertiti con la magia di Minecraft](#).