IBM Instana

Gestione delle prestazioni delle applicazioni per applicazioni di microservizi su Kubernetes

Come migliorare le prestazioni di app business-critical in un ambiente Kubernetes

Indice

01 →
Introduzione

02 →
Nozioni di base su Kubernetes, ovvero l'ABC di K8s

03 →
Sfide proposte dal monitoraggio delle applicazioni Kubernetes

04 →
Non farsi ingannare dalla complessità

05 →
Strumenti e strategie di monitoraggio Kubernetes

06 →
Conclusioni

07 →
IBM Instana è la scelta giusta per te?

01

Introduzione

Kubernetes merita l'attenzione ricevuta negli ultimi anni per un buon motivo. Con il passaggio delle organizzazioni ad ambienti di container gestiti con stack di applicazioni di microservizi, Kubernetes è diventato la soluzione di orchestrazione dei container di riferimento.

Nonostante i suoi punti di forza, Kubernetes da solo non è in grado di superare tutte le sfide di gestione che si affrontano, in particolare, per quanto riguarda le prestazioni. La gestione efficace delle applicazioni di microservizi orchestrate e il raggiungimento dell'eccellenza operativa richiedono una conoscenza completa di ciò che Kubernetes può e non può fare, oltre alle funzionalità che i team DevOps stanno cercando.

Questo ebook fornisce un esame approfondito di Kubernetes e tratta delle sue funzionalità di gestione e dei suoi limiti. La pubblicazione esamina i moderni processi DevOps, che includono strumenti di gestione delle prestazioni per la distribuzione continua di servizi di business per ottenere prestazioni operative eccezionali. Fornisce inoltre un'analisi dettagliata di come operare e gestire correttamente le prestazioni delle applicazioni di microservizi in esecuzione su Kubernetes.

Nozioni di base su Kubernetes, ovvero l'ABC di K8s

Kubernetes, a volte abbreviato in K8s, è uno strumento di orchestrazione dei container per l'implementazione di applicazioni di microservizi. Nasce come strumento di orchestrazione dell'infrastruttura creato da Google per aiutare a gestire l'implementazione dei container nel suo ambiente iperscalabile. Alla fine, Google ha rilasciato K8s come soluzione open-source attraverso la CNCF (Cloud Native Computing Foundation).

L'orchestrazione include queste funzioni base di Kubernetes:

- Distribuzione automatica dei container, in modo che gli amministratori non debbano più avviarli manualmente
- Gestione delle istanze o bilanciamento del numero di istanze di un determinato container contemporaneamente in esecuzione per soddisfare la richiesta dell'applicazione
- Gestione del DNS (Domain Name System) per quanto riguarda il bilanciamento del carico di microservizi e container e la creazione di cluster per facilitare la gestione della scalabilità dovuta all'incremento dei carichi di richieste
- Gestione della distribuzione dei container nei server host, suddividendo equamente il carico delle applicazioni nell'infrastruttura host per aumentare al massimo la disponibilità delle applicazioni

Si noti che non è presente un aspetto critico della gestione operativa: APM (Application Performance Management). L'intera disciplina della visibilità e gestione delle prestazioni dell'applicazione non fa parte della piattaforma Kubernetes.

Perché Kubernetes è importante
Ricorda, l'obiettivo di DevOps è la velocità. L'orchestrazione facilita modifiche veloci e semplici agli ambienti di produzione, in modo che le applicazioni di business possano evolvere rapidamente.

Il messaggio è chiaro: l'accelerazione dei cicli di distribuzione delle applicazioni può aggiungere enorme valore al tuo business.

L'automazione dell'orchestrazione dei container può essere un ottimo complemento ai metodi di sviluppo agile e all'architettura di microservizi. La moderna procedura CI/CD (continuous integration/continuous deployment, integrazione continua/distribuzione continua) automatizza le fasi di test e distribuzione dello sviluppo. Quindi, i container e Kubernetes rendono molto più semplice mettere in produzione il codice e gestire le risorse.

Distribuzioni Kubernetes
Molti provider cloud propongono le proprie versioni di Kubernetes, chiamate “distribuzioni”. Hanno funzionalità aziendali uniche, aggiunte alla versione open-source di Kubernetes, che possono fornire alcuni vantaggi specifici:

- Le organizzazioni sono preoccupate circa la preparazione aziendale a ricevere una versione completamente sottoposta a test e supportata di K8s.
- Sono incluse ulteriori funzionalità aziendali. Ad esempio, la distribuzione di Red Hat® OpenShift® K8s aggiunge funzioni di sicurezza e integra l'automazione.

Per la maggior parte dei casi d'uso aziendali, può essere più veloce e più semplice utilizzare la distribuzione Kubernetes di un provider cloud, piuttosto che configurare la versione open-source. È disponibile una vasta gamma di distribuzioni Kubernetes, che possono essere eseguite su infrastrutture locali o come servizio hosted nel cloud.

Nota: *sono disponibili molte distribuzioni, reperibili nella documentazione online di Kubernetes.*

03

Sfide proposte dal monitoraggio delle applicazioni Kubernetes

La gestione dei container non è la gestione delle prestazioni delle applicazioni
Ora che abbiamo parlato di cosa fa Kubernetes, spieghiamo cosa non fa. Ricorda, Kubernetes orchestra i container che fanno parte di un'applicazione. Tuttavia, non misura le prestazioni o la disponibilità di un'applicazione, con riferimento ad applicazioni ad elevata distribuzione. Allo stesso modo, non considera le prestazioni quando gestisce l'infrastruttura.

Kubernetes aggiunge essenzialmente un livello di astrazione tra l'esecuzione dell'applicazione (container) e la reale infrastruttura di calcolo. In modo autonomo, Kubernetes decide dove eseguire i container e può spostarli all'improvviso. La visibilità di come viene implementato esattamente lo stack tecnico e di come le richieste di servizio passano attraverso i microservizi non è facilmente disponibile con Kubernetes, né i dati sulle prestazioni—frequenza delle richieste, errori e durata o latenza—dei servizi rappresentano una parte nativa di Kubernetes.

Il monitoraggio di produzione delle prestazioni e dello stato di integrità di un'applicazione non fa parte della specifica di Kubernetes.

Esaminiamo altri aspetti di ambienti di applicazioni containerizzati e orchestrati, che possono complicare ulteriormente il monitoraggio.

Qualsiasi applicazione di microservizi, per definizione, può creare tre tipi di problemi:
- Un numero esponenzialmente sempre maggiore di singoli componenti
- Modifica costante nell'infrastruttura e nelle applicazioni (lo stack di applicazioni)
- Componenti dinamici dell'applicazione

Un ambiente Kubernetes tende ad includere molte più parti mobili di quante sarebbero incluse in uno stack di applicazioni tradizionale. E questo implica maggiore complessità.

Con l'aggiunta di container e, quindi, di orchestrazione con Kubernetes, è possibile che aumenti la difficoltà di ognuna di queste sfide di gestione. Ogni volta che si realizza una separazione tra implementazione fisica e funzionalità dell'applicazione, diventa spesso più difficile monitorare le prestazioni dell'applicazione e risolvere i problemi. Invece di server host connessi ad una rete fisica, Kubernetes utilizza un cluster di nodi e virtualizza la rete, che può essere distribuita attraverso una combinazione di infrastruttura on-premise e basata su cloud o anche attraverso più cloud.

Con così tanti componenti differenti di infrastruttura e middleware, insieme alla molteplicità di linguaggi delle applicazioni utilizzati per creare i microservizi, è difficile per gli strumenti di monitoraggio distinguere l'eterogeneità di esigenze e comportamenti di tutti questi componenti critici nello stack di applicazioni. Ad esempio, la raccolta e l'interpretazione dei dati di monitoraggio da una qualsiasi piattaforma è probabilmente diversa da tutte le altre piattaforme. Cosa fai quando ti trovi di fronte Python, Java, PHP, .NET, proxy di applicazioni, 4 diversi database e una moltitudine di middleware?

Separazione dei microservizi dall'infrastruttura fisica

Kubernetes assume il controllo dell'esecuzione dei container che costituiscono i microservizi dell'applicazione, automatizzando completamente la gestione del relativo ciclo di vita e astraendo l'hardware.

Kubernetes eseguirà i carichi di lavoro richiesti su qualsiasi host o nodo disponibile, utilizzando SDN (software-defined network) per garantire che tali carichi di lavoro siano raggiungibili e che il carico sia bilanciato. Anche le risorse di calcolo, memoria e CPU, sono astratte, con ogni carico di lavoro che ha un limite configurato per tali risorse. Poiché i container sono effimeri, qualsiasi archiviazione a lungo termine è fornita da richieste di volumi persistenti fornite da vari driver di archiviazione.

Il livello di astrazione già profondo può essere ulteriormente complicato da nodi Kubernetes in esecuzione su servizi esterni di cloud computing, quali Amazon Elastic Compute Cloud (Amazon EC2), Google Compute Engine (GCE) o Microsoft Azure.

L'alto livello di disconnessione tra il codice dell'applicazione e l'hardware su cui è in esecuzione tende a semplificare il monitoraggio dell'infrastruttura tradizionale. Questo processo, tuttavia, aumenta l'importanza della comprensione di come funzionano i microservizi e le applicazioni che li contengono e se rispettano gli SLA (service level agreements, accordi sul livello del servizio) desiderati. Una comprensione dello stato di integrità complessivo del backplane di Kubernetes è utile anche per garantire all'applicazione i massimi livelli di servizio.

Associazione di servizi: un nuovo livello di astrazione

Uno dei motivi principali per utilizzare un orchestrator, come ad esempio Kubernetes, è quello di automatizzare l'implementazione dei container e stabilire comunicazioni tra di loro. Tuttavia, gli orchestrator da soli non riescono necessariamente a garantire che i microservizi possano comunicare e integrarsi reciprocamente, in modo efficace. Per farlo, è utile monitorare direttamente i servizi e le relative interazioni.

Il monitoraggio diretto è impegnativo, perché Kubernetes non offre un modo per associare o visualizzare automaticamente le relazioni tra i microservizi.

Gli amministratori, normalmente, dovrebbero determinare quali microservizi siano effettivamente in esecuzione, dove si trovino all'interno del cluster, quali servizi dipendano da altri e come vengano trasmesse le richieste tra servizi, manualmente o da altre origini dati.

Quando si aggiorna il software o si eseguono test di regressione, maggiore sarà la rapidità con cui i proprietari dell'applicazione, DevOps e gli sviluppatori riusciranno a valutare l'impatto dell'aggiornamento, più tempestivamente potranno procedere alla fase successiva del progetto e sapere se le loro applicazioni stanno funzionando in modo appropriato o meno.

Ambiguità della causa principale
Gli strumenti APM esistono principalmente perché le applicazioni di business basate su middleware, prima utilizzando Java e .NET, quindi, utilizzando principi SOA (Service-Oriented Architecture) e persino microservizi e container, possono rendere difficile monitorare le prestazioni dell'applicazione, tenere traccia delle richieste dell'utente, identificare e risolvere i problemi, quando si verificano.

Quanto più è complesso l'ambiente delle applicazioni, tanto più è difficile per i team DevOps ottenere la visibilità delle prestazioni e le dipendenze dei componenti necessarie per gestire efficacemente le prestazioni delle applicazioni.

In un ambiente Kubernetes, determinare la causa principale di un problema in base ai sintomi a livello di superficie può risultare ancora più difficile. Questo perché le relazioni tra diversi componenti dell'ambiente tendono ad essere più difficili da associare e si modificano continuamente. Ad esempio, un problema in un'applicazione Kubernetes potrebbe essere causato da una problematica relativa all'infrastruttura fisica, ma potrebbe anche essere dovuto a un errore di configurazione oppure a un problema di codifica. O forse il problema sta nella rete virtuale che consente la comunicazione reciproca tra i microservizi.

Naturalmente, quando il problema risiede nel codice dell'applicazione, una profonda visibilità a livello di codice può aiutare ad eseguire il debug delle reali problematiche relative al codice. Questo tipo di visibilità più profonda potrebbe persino rilevare quando parametri o altri input errati causano problemi all'applicazione. In definitiva, potrebbe esserci una miriade di cause principali per la problematica, che vanno da problemi di configurazione in Kubernetes, ad una problematica con i flussi di dati tra container, ad un malfunzionamento dell'hardware fisico.

In parole semplici, far risalire i problemi in un ambiente Kubernetes alla loro causa principale in molti casi potrebbe non essere un processo attuabile, a meno che non si utilizzino strumenti in grado di analizzare automaticamente il complesso intreccio di dati e dipendenze che compongono il proprio cluster e la struttura della propria applicazione di microservizi.

04

Non farsi ingannare dalla complessità

Sebbene la gestione delle prestazioni e della disponibilità delle applicazioni Kubernetes possa essere impegnativa e scoraggiante, non è impossibile. Lo strumento APM appropriato, in grado di riconoscere le sfide peculiari del monitoraggio di un ambiente Kubernetes, può aiutare a gestire il proprio ambiente in modo da aumentare al massimo il tempo di attività e ottimizzare le prestazioni. Ottenere il meglio da entrambi i mondi ti aiuta a combinare i vantaggi di K8s con l'obiettivo dell'eccellenza DevOps.

Diamo un'occhiata ai tipi principali di visibilità per le applicazioni di supporto in esecuzione in un ambiente Kubernetes.

Identificazione e associazione del servizio applicazione
Come illustrato in precedenza, Kubernetes inserisce un nuovo livello di astrazione dell'applicazione che può rendere difficile valutare i singoli servizi o le interdipendenze tra tutti i servizi implementati. Quanto più in profondità gli strumenti APM riescono a vedere oltre Kubernetes e il sistema di container, tanto più è facile identificare i servizi applicativi e le loro dipendenze.

Un servizio Kubernetes è un'astrazione che definisce una serie logica di pod e una politica per consentirne l'accesso.

In un pod possono esistere molteplici servizi applicazione.

Relazioni tra microservizi
Se si conosce il modo in cui i propri servizi Kubernetes si associano ai servizi applicazione, i microservizi su cui si basano e la relativa infrastruttura fisica, allora sarà possibile determinare in che modo l'infrastruttura influisce sulla disponibilità e sulle prestazioni dei servizi. Con Kubernetes non è necessariamente facile ottenere tutte queste informazioni. Probabilmente, sarà necessario eseguire molteplici comandi kubectl, utilizzando lo strumento riga comandi per comunicare con il piano di controllo di un cluster Kubernetes, per creare manualmente un'associazione in un singolo momento. Questa può rivelarsi una vera distrazione quando si presenta una problematica di produzione.

Associazione e traccia di una richiesta dell'applicazione
I microservizi che compongono un'applicazione inviano e ricevono costantemente richieste reciproche. Può essere molto più facile monitorare e gestire le prestazioni di un'applicazione di microservizi se lo strumento APM riesce a rilevare tutti i servizi e le relative interdipendenze e a visualizzare le relazioni dinamiche—ovvero, associarli—in tempo reale.

La risoluzione dei problemi può essere semplificata se lo strumento è anche in grado di estrarre singole tracce da ogni specifica richiesta dell'applicazione, in modo particolare in tutti i microservizi distribuiti che contatta.

Errori di implementazione
Se Kubernetes non riesce ad implementare un pod come previsto, potrebbe essere utile sapere quando, perché e come è successo. Tuttavia, è probabilmente più importante capire se la funzionalità dell'applicazione è stata influenzata negativamente da questo errore di implementazione. La tua applicazione è più lenta e gestisce una quantità inferiore di carico di lavoro oppure genera errori perché un servizio critico non è disponibile?

Il flusso di eventi può mostrare in che punto si è verificato l'errore di implementazione, ma, poiché non è possibile esaminare le prestazioni della propria applicazione utilizzando i comandi kubectl, si dovrebbe trovare un modo alternativo per rispondere alla domanda posta precedentemente, molto probabilmente con uno strumento APM in grado di capire Kubernetes.

Regressioni delle prestazioni
Se l'applicazione sta rispondendo lentamente, probabilmente vorrai identificare rapidamente la problematica e tenere traccia della causa principale. Sfortunatamente, Kubernetes non include in modalità nativa il monitoraggio delle prestazioni, quindi, non esistono realmente comandi kubectl che si possono eseguire per comprendere le prestazioni del microservizio o dell'applicazione.

Quando si risolvono problemi di applicazioni di microservizi in esecuzione su Kubernetes, è utile che lo strumento APM possa correlare le metriche a tutti i livelli dello stack completo delle applicazioni: infrastruttura, codice dell'applicazione, informazioni sul sistema Kubernetes e dati di traccia tra i servizi.

Le metriche dell'infrastruttura, quali CPU (central processing unit), memoria, I/O (input e output) su disco, I/O di rete e così via, possono essere validi KPI a cui fare riferimento durante la risoluzione di problematiche relative alle prestazioni. Tuttavia, queste metriche tendono ad essere solo una parte delle informazioni necessarie per determinare pienamente la causa principale. Potrebbero esserci anche problemi con il codice dell'applicazione o problemi di configurazione di Kubernetes che causano il conflitto di risorse. Ad esempio, può essere comune allocare un numero eccessivo di risorse di CPU e memoria sui nodi Kubernetes attraverso una configurazione non corretta.

Opportunità di ottimizzazione delle prestazioni
In ambienti di sviluppo agile, gli sviluppatori spesso inviano nuovo codice in produzione quotidianamente. Come fanno a sapere che il loro codice sta fornendo un buon tempo di risposta e non sta consumando troppe risorse?

Più reattivo e flessibile è lo strumento APM rispetto ai cambiamenti, più è probabile che ti aiuti a riconoscere automaticamente e immediatamente quando è stato distribuito nuovo codice, oltre a qualsiasi modifica infrastrutturale o architettonica. È ancora meglio se riesce anche ad aiutare gli sviluppatori ad analizzare il loro codice con visibilità della riga di codice o creazione di profili di produzione.

Per questi casi d'uso, può essere utile disporre di una visibilità granulare delle richieste dell'utente, delle risorse host (nodi K8s) e dei modelli di carichi di lavoro. Sono utili anche potenti strumenti di analytics che possono includere tutti i dati che si stanno raccogliendo, specialmente quando si lavora con dati oltre l'ambito di ciò che Kubernetes acquisisce in modo nativo.

Si dovrebbe utilizzare Prometheus?
Prometheus è diventato uno dei metodi standard di monitoraggio delle prestazioni per Kubernetes, ma potrebbe anche presentare alcune lacune funzionali critiche. Iniziamo illustrando le funzioni efficaci di Prometheus.

Prometheus è uno strumento open-source di monitoraggio e generazione di avvisi per metriche di serie temporali. Viene tipicamente utilizzato per monitorare KPI, quali frequenze, contatori e misuratori dall'infrastruttura e dai servizi applicazione. È possibile utilizzare Prometheus per monitorare i tempi di risposta della richiesta. Ma questo metodo spesso richiede di modificare il codice sorgente per aggiungere le chiamate all'API (application programming interface) Prometheus. Prometheus può aiutarti a conoscere i tempi complessivi di risposta e le frequenze delle richieste, ma l'approccio può anche essere privo del livello di dettaglio necessario per risolvere i problemi o ottimizzare le prestazioni dell'applicazione.

L'implementazione di Kubernetes supporta Prometheus in modalità nativa. Una volta installato il pacchetto Prometheus Helm, saranno disponibili diversi dashboard preconfigurati allo scopo di effettuare controlli di base dello stato di integrità. Si troveranno anche alcuni avvisi predefiniti, configurati sul proprio cluster.

Metriche delle serie temporali

API flessibile

Elevata cardinalità

Monitoraggio e generazione di avvisi

In definitiva, Prometheus può essere un valido strumento di metriche autonomo. Ma questo non significa necessariamente essere in grado di affrontare le sfide di gestione delle prestazioni associate all'esecuzione di carichi di lavoro di microservizi business-critical su Kubernetes. Ecco alcuni inconvenienti che potrebbero rendere difficile per Prometheus la risoluzione di numerosi casi d'uso critici:

- Nessuna funzionalità di traccia distribuita
- Nessuna correlazione tra infrastruttura del servizio e host
- Nessuna correlazione tra risorse Kubernetes, tempi di risposta della richiesta e metriche dell'infrastruttura
- Nessuna interfaccia di analytics integrata, roll-up o aggregati
- Nessuna analisi automatica dell'origine del problema
- Nessuna generazione automatica di avvisi
- Costi amministrativi e di gestione

Strumenti e strategie di monitoraggio Kubernetes

Può essere più semplice ed efficace ottenere gli elementi di cui sopra in un'applicazione Kubernetes con uno strumento APM che includa funzioni non limitate ai tradizionali strumenti di monitoraggio.

Per Kubernetes, può essere molto più facile gestire e trattare problematiche se lo strumento non si limita a raccogliere semplicemente dati di monitoraggio e rilevare anomalie. Ecco alcune funzionalità chiave che si suggerisce di ricercare, che potrebbero aiutare uno strumento APM a garantire le prestazioni di applicazioni basate su Kubernetes.

Analisi dell'origine del problema all'interno dell'applicazione, dei container e dell'orchestrazione
Quando si verificano problemi, è utile che la propria soluzione di monitoraggio sia in grado di identificare automaticamente la causa principale delle problematiche relative alle prestazioni, indipendentemente dal punto dello stack di tecnologie in cui si verifica il problema.

Al di fuori dell'ambiente Kubernetes, la soluzione di monitoraggio non dovrebbe solo rilevare problemi all'interno di Kubernetes, ma anche tenere traccia dei problemi fino alla loro esatta causa per aiutare a risolverli rapidamente.

Data l'estrema complessità di un'applicazione basata su Kubernetes e la solita mancanza di visibilità di tale ambiente, l'identificazione delle cause principali delle problematiche relative a disponibilità o prestazioni può essere difficile da ottenere manualmente.

Associazione di servizi e infrastruttura integrata
Per aiutare a colmare le lacune nelle dipendenze tra servizi e componenti dell'applicazione, è utile che lo strumento di monitoraggio possa rilevare e associare i servizi.

Ancora meglio, dovrebbe anche comprendere le relazioni e le dipendenze tra tali servizi, aiutando a creare mappe più accurate. Questa accuratezza può aiutare gli utenti a identificare problemi tra due servizi eterogenei.

Creazione di una baseline dinamica
Con le architetture e le configurazioni dell'ambiente che cambiano costantemente, è più facile per il team tenersi aggiornato se lo strumento APM interpreta anche dati di monitoraggio estremamente dinamici, per distinguere reali anomalie dai normali cambiamenti.

Indicazioni per la correzione
Quando si verifica qualche problema nell'ambiente Kubernetes enormemente complesso, si deve risolvere velocemente il problema. Può essere difficile per gli amministratori farlo, senza aiuto o indicazioni provenienti da uno strumento APM, a causa delle grandi quantità di dati e delle variabili che cambiano rapidamente. È quasi impossibile per gli operatori districarsi tra i dati e formulare un piano di risposta agli incidenti autonomamente.

06

Conclusioni

Sebbene Kubernetes si stia rapidamente espandendo in tutto il mondo DevOps come piattaforma di infrastruttura standard, non sempre include la visibilità delle prestazioni dell'applicazione di cui i team hanno bisogno. Kubernetes introduce un nuovo livello di astrazione nel data center, creando spesso sfide relative all'osservabilità che rendono più difficile gestire la disponibilità dell'applicazione. Come si potrebbero offrire gli SLA di prestazioni richiesti dall'azienda in un ambiente Kubernetes?

Per gestire efficacemente applicazioni business-critical su Kubernetes, cerca uno strumento APM che includa queste funzionalità chiave:

- Visibilità dello stack completo—che include infrastruttura, codice, microservizi, tracce delle richieste, middleware, container e Kubernetes—per tutti i livelli di tecnologia
- Rilevamento continuo dello stack completo delle applicazioni per adeguarsi automaticamente ai cambiamenti dell'ambiente
- Associazione delle dipendenze e correlazione tra i livelli di tecnologia
- Determinazione automatica della causa principale e assistenza per i team DevOps nella risoluzione di problematiche relative all'applicazione

07

IBM Instana è la scelta giusta per te?

IBM Instana è una piattaforma di osservabilità aziendale che include funzionalità di monitoraggio automatico delle prestazioni dell'applicazione. È progettato per le aziende che utilizzano applicazioni cloud-native complesse e moderne, indipendentemente da dove risiedono: on-premises, su cloud pubblici e privati, su dispositivi mobili o in un ambiente IBM zSystems™.

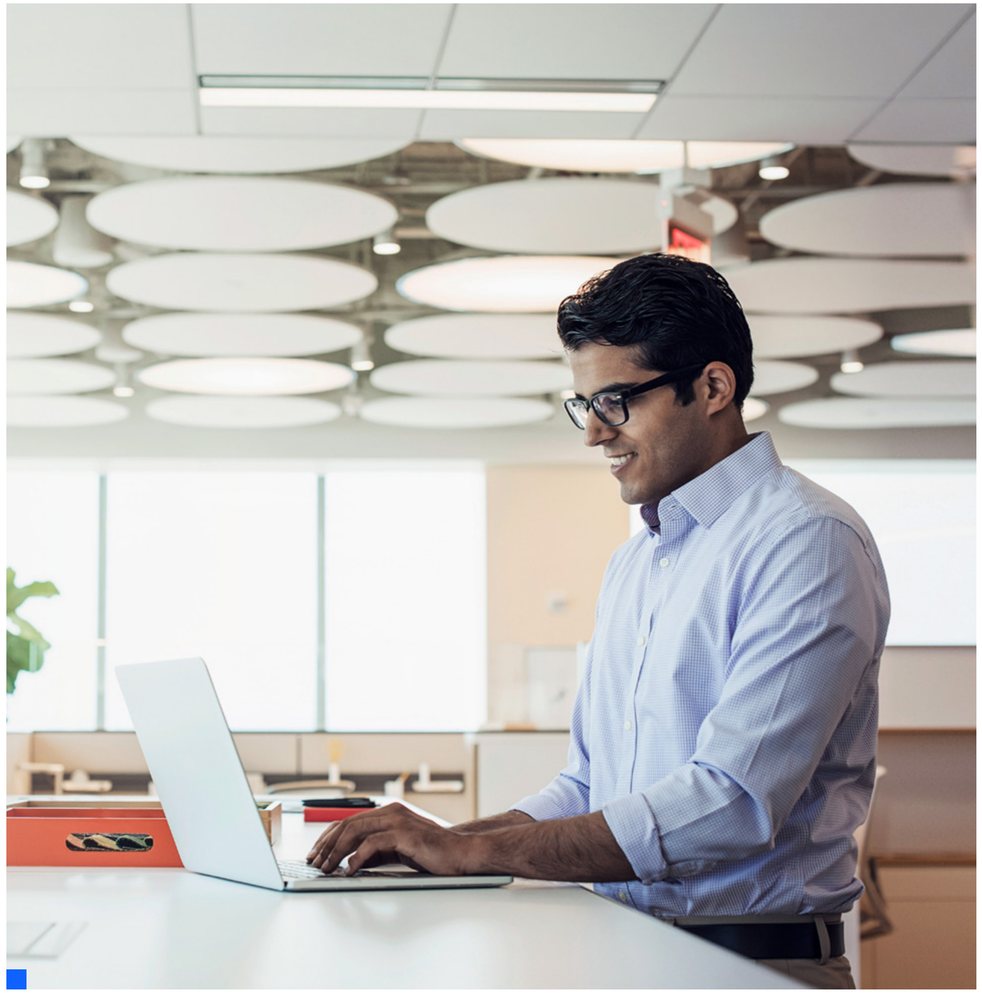
IBM Instana ti aiuta a controllare moderne applicazioni ibride con rilevamento basato sull'intelligenza artificiale di profonde dipendenze contestuali all'interno delle applicazioni ibride. IBM Instana fornisce inoltre visibilità delle pipeline di sviluppo, per facilitare l'abilitazione dell'automazione DevOps a ciclo chiuso.

Queste funzionalità forniscono feedback utilizzabili concretamente, necessari per i clienti quando ottimizzano le prestazioni dell'applicazione, abilitano l'innovazione e attenuano il rischio. Queste funzionalità aiutano DevOps ad aumentare l'efficienza e aggiungono valore alle pipeline di distribuzione del software, in modo da poter soddisfare i propri obiettivi a livello di business e di servizio.

Scopri tu stesso la potenza di IBM Instana. Iscriviti oggi per una prova gratuita per 14 giorni della versione completa del prodotto. Non è necessaria la carta di credito.

Versione di prova gratuita di IBM Instana →

Esplora IBM Instana →



© Copyright IBM Corporation 2023

IBM Italia S.p.A.
Circonvallazione Idroscalo
20090 Segrate (Milano)
Italia

Prodotto negli Stati Uniti d'America
Maggio 2023

IBM, il logo IBM, IBM Instana e zSystems sono marchi o marchi registrati di International Business Machines Corporation, negli Stati Uniti e/o in altri Paesi Altri nomi di prodotti e servizi potrebbero essere marchi di IBM o di altre società. Un elenco aggiornato dei marchi IBM è disponibile all'indirizzo ibm.com/trademark.

Red Hat e OpenShift sono marchi o marchi registrati di Red Hat, Inc. o di sue controllate negli Stati Uniti e in altri Paesi.

Microsoft è un marchio di Microsoft Corporation negli Stati Uniti e/o in altri Paesi.

Questo documento è aggiornato alla data di pubblicazione e può essere modificato da IBM in qualsiasi momento. Non tutte le offerte sono disponibili in ogni paese in cui IBM opera.

È responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi altro prodotto o programma con prodotti e programmi IBM. LE INFORMAZIONI CONTENUTE IN QUESTO DOCUMENTO SONO FORNITE "NELLO STATO IN CUI SI TROVANO" SENZA ALCUNA GARANZIA, ESPRESSA O IMPLICITA, SENZA GARANZIE DI COMMERCIABILITÀ O IDONEITÀ AD UNO SCOPO PARTICOLARE E SENZA ALCUNA GARANZIA O CONDIZIONE DI NON VIOLAZIONE. I prodotti IBM sono garantiti in accordo ai termini e alle condizioni dei contratti che ne regolano la fornitura.