

Sviluppo remoto su AWS: da Cloud9 a VS Code

20 Novembre 2024 - 8 min. read

Cloud9

Visual Studio Code

Negli ultimi anni, lo sviluppo software ha visto una trasformazione significativa, con un numero crescente di sviluppatori che abbracciano approcci di lavoro distribuiti e collaborativi. Con il diffondersi delle metodologie DevOps, l'adozione del paradigma cloud-native e l'uso crescente di strumenti di automazione, la flessibilità e la capacità di lavorare su diverse piattaforme sono diventate essenziali. Tuttavia, questa flessibilità introduce anche sfide, in particolare quando si tratta di mantenere la coerenza degli ambienti di sviluppo, che spesso varia notevolmente a seconda del dispositivo o del sistema operativo utilizzato, nonché dei progetti su cui si sta lavorando.

AWS ha annunciato di non avere piani per sviluppare nuove funzionalità per Cloud9, e ormai da diversi mesi il servizio non è disponibile per i nuovi clienti. Questo ha portato molti sviluppatori a cercare alternative che offrano maggiore personalizzazione, flessibilità e una migliore esperienza utente. Visual Studio Code (VS Code), combinato con un'istanza EC2 o ECS su AWS, rappresenta una valida alternativa. Questa configurazione permette agli sviluppatori di accedere a un ambiente di sviluppo remoto, potente e unificato, mantenendo le stesse configurazioni e strumenti cui sono abituati, indipendentemente dal dispositivo client utilizzato.

L'utilizzo di VS Code per lo sviluppo remoto garantisce un flusso di lavoro portabile, l'accesso a risorse di calcolo on demand, e ad una connettività ultra veloce per il download di librerie, dipendenze ed artefatti.

Questo approccio riduce i problemi di compatibilità, sfrutta appieno le capacità delle istanze EC2 e offre la possibilità di integrare strumenti e configurazioni personalizzate.

In questo articolo, vedremo come configurare Visual Studio Code per lavorare su un'istanza EC2, esplorando passo dopo passo le impostazioni necessarie e i vantaggi che ne derivano.

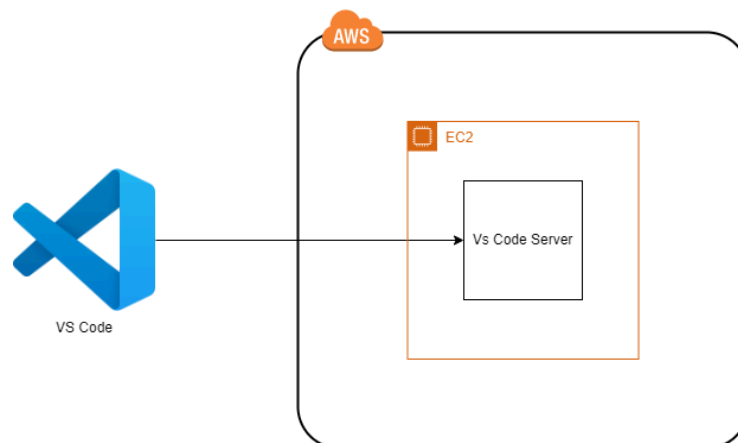
La soluzione

Per ottenere un'esperienza simile a quella di Cloud9 occorre che il codice risieda fisicamente sull'istanza remota, così come le dipendenze, eventuali librerie e configurazioni specifiche. Il client locale deve essere quindi completamente integrato con l'host, mostrandone il terminale ed usando l'ambiente remoto per la ricerca delle librerie, per l'auto-completamento e in generale per tutte le funzioni che richiedono accesso al sorgente.

Inoltre, deve essere possibile eseguire i test sull'istanza remota, o avviare un server di sviluppo per testare il software su cui si sta lavorando.

Con Visual Studio Code, è possibile ottenere tutto quello che ci occorre mediante una singola estensione, che permette di collegare il client ad un **Visual Studio Code Server**.

Quello che andremo a realizzare quindi, è molto semplicemente la configurazione di un'istanza EC2 per accogliere il server di sviluppo, vedremo anche come configurare il client per connettersi all'istanza ed utilizzarla per lo sviluppo remoto.



La comunicazione tra client e server è basata su SSH, ed è possibile configurare una singola istanza per più sviluppatori sfruttando il meccanismo di autenticazione e segregazione dello spazio utente già presente in Linux e supportato da SSH.

Per ogni sviluppatore è possibile creare un account utente sull'istanza e per ogni utente verrà eseguita un'istanza del server VS Code.

Vantaggi dello sviluppo su istanza remota

Sviluppare su un'istanza remota offre una serie di vantaggi significativi che possono migliorare notevolmente il flusso di lavoro di un team e semplificare la gestione degli

ambientati di sviluppo.

Uno dei problemi più comuni, per esempio, è l'incoerenza tra le configurazioni delle macchine degli sviluppatori. Differenze anche lievi possono causare errori difficili da diagnosticare, oppure ostacolare e rallentare il lavoro collaborativo.

Con un'istanza remota, tutte queste configurazioni sono centralizzate. Ogni sviluppatore lavora su un ambiente identico, potenzialmente sulla stessa istanza, laddove non vi siano controindicazioni specifiche, riducendo al minimo i problemi legati alla compatibilità e garantendo che le dipendenze siano sempre aggiornate e coerenti.

è possibile configurare l'host per offrire gli interpreti e le librerie più comunemente utilizzate, centralizzandone il versionamento e la distribuzione.

Laddove fosse necessario, è anche possibile installare sistemi di gestione delle versioni degli interpreti, come pyenv o nvm, e renderli disponibili a tutti gli sviluppatori.

Un ulteriore aspetto da non trascurare sono le performance. Molti framework moderni sono esosi di risorse durante le fasi di compilazione o di build, senza contare il crescente impiego di algoritmi di machine learning che richiedono ingenti quantità di tempo CPU, GPU dedicate e grandi quantità di RAM.

Svincolando l'IDE dall'ambiente di esecuzione utilizzando un'istanza remota è possibile assegnare agli sviluppatori risorse hardware notevolmente più performanti rispetto ai dispositivi locali, aumentandole o riducendole al bisogno per ottimizzare i costi e velocizzare il lavoro.

Inoltre, se un dispositivo locale si guasta o necessita di manutenzione, non ci sono interruzioni significative: l'ambiente remoto rimane intatto e pronto all'uso.

Infine, centralizzando lo sviluppo su un server remoto, è più facile implementare e monitorare politiche di sicurezza e controllare l'attività degli utenti.

Configurazione del client: Estensione "Remote - SSH"

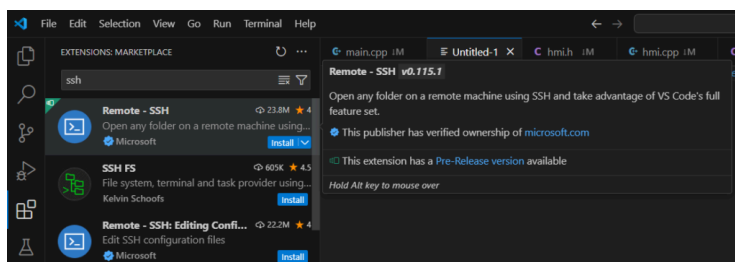
Il primo passo è installare e configurare l'estensione "Remote - SSH" in Visual Studio Code. Questa estensione è il cuore della connessione remota, poiché consente di lavorare su un'istanza remota come se fosse locale, mantenendo al contempo la potenza e la flessibilità del proprio ambiente di sviluppo.

Si tratta di un componente aggiuntivo per VS Code che permette di aprire una cartella su un server remoto direttamente dall'editor, utilizzando un tunnel SSH. Con questa

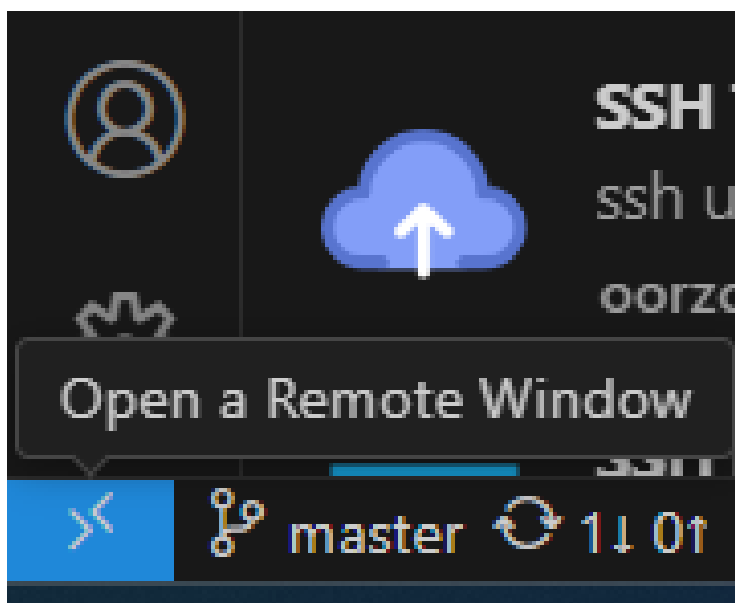
estensione, i file vengono visualizzati, modificati e gestiti all'interno di VS Code, ma rimangono fisicamente sul server remoto.

All'interno del tunnel SSH vengono convogliati anche i comandi scritti sul terminale integrato, che è in realtà un terminale remoto, così come tutte le operazioni che VS Code esegue sul FS per fornire il supporto al linguaggio ed i framework desiderati.

Installare l'estensione è facilissimo: nella barra laterale di sinistra di VS Code, clicca sull'icona delle estensioni e cerca "Remote - SSH". Il primo risultato dovrebbe essere l'estensione ufficiale di Microsoft.



Procediamo con l'installazione dell'estensione, una volta completato il processo sarà disponibile una nuova opzione nel pannello di comando che permette di collegarsi ad un host remoto.



Configurazione dell'istanza

Non sono necessarie particolari configurazioni sull'istanza remota. VS Code provvederà all'installazione del server automaticamente una volta instaurato il tunnel SSH.

La componente server infatti viene avviata in spazio utente, e non necessita di privilegi elevati.

Sarà poi direttamente il client a mantenere aggiornata la componente server side, proponendo gli aggiornamenti all'utilizzatore proprio come avviene per la parte client.

Ciò che dobbiamo fare quindi, è solo avviare un'istanza EC2, avendo cura di scegliere le risorse più adatte in base al caso d'uso.

Se stai facendo una prova o se il caso d'uso non è avido di risorse, la famiglia "t" permette di ottenere il miglior prezzo, e di solito si adatta bene al pattern di utilizzo della CPU di uno sviluppatore che non compila in modo intensivo.

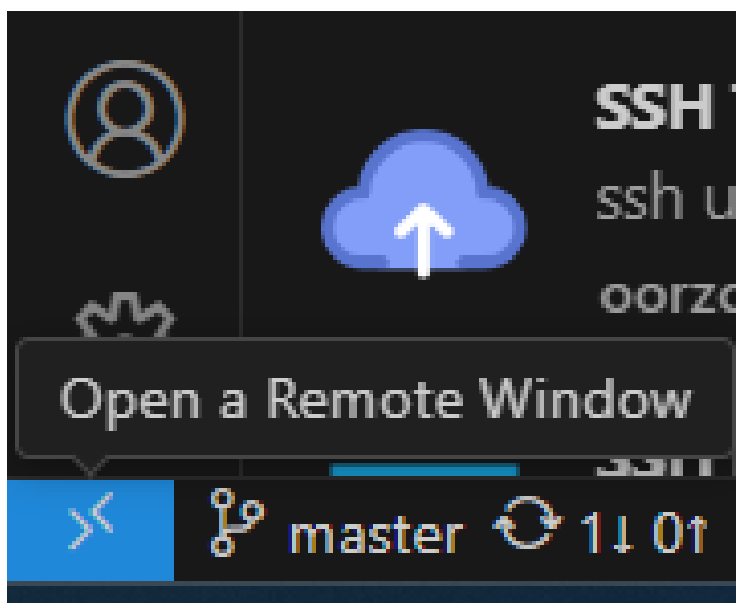
Se invece l'istanza remota serve per accelerare lo sviluppo ed il testing di algoritmi di ML esistono istanze particolarmente performanti e che hanno a disposizione GPU potenti. La famiglia "Inf2" è la miglior scelta se il carico di lavoro è veramente importante e rappresenta il massimo delle performance attualmente disponibile per i progetti AI compute intensive.

Nell'avviare l'istanza assicurati di collegarla ad un security group che permetta le connessioni sulla porta 22 dal tuo IP o dall'IP range aziendale, e seleziona o genera una chiave SSH per l'accesso.

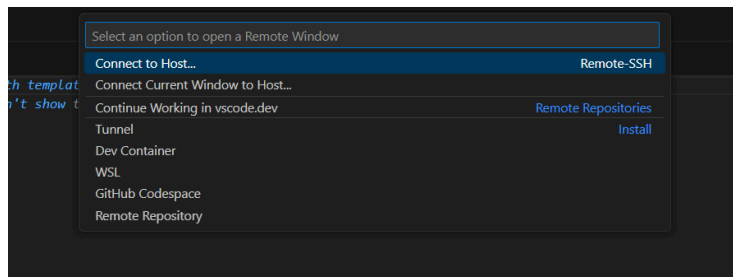
Come ulteriore configurazione, è possibile collegarsi e creare un utente ad hoc, così da evitare di utilizzare l'utente di default. Consigliamo di utilizzare comunque un'autenticazione basata su chiave SSH e di evitare la sola autenticazione con password.

Collegamento all'istanza

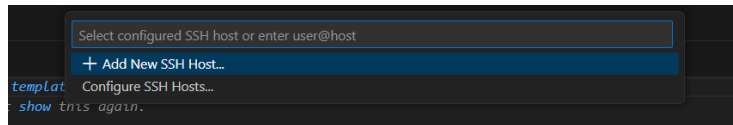
Una volta che l'istanza è attiva e raggiungibile, per collegarsi basta cliccare sull'icona dell'estensione nel pannello di comando in basso



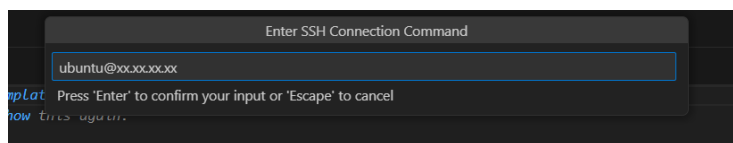
e successivamente selezionare "Connect to host"



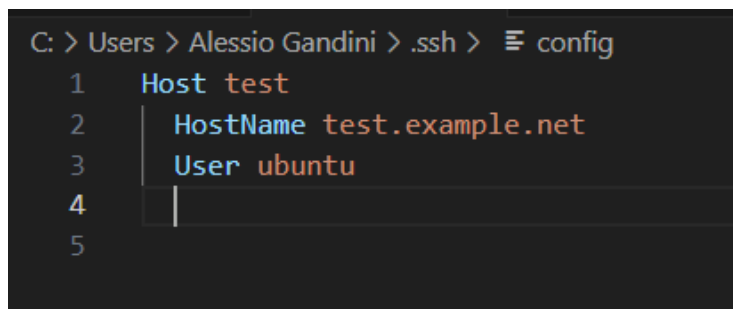
Si può quindi procedere ad aggiungere un nuovo host, specificando nome utente e ip o nome di dominio associato alla nostra istanza.



Il formato da utilizzare è utente@ip o fqdn



Successivamente, ti verrà chiesto dove salvare il file di configurazione SSH. Questo file conterrà le informazioni di connessione degli host, così da non doverle riscrivere ogni volta.



Una volta selezionato il file di configurazione è possibile visualizzarlo e modificarlo per configurare opzioni aggiuntive.

Durante l'accesso, VS Code cercherà di intuire il meccanismo di autenticazione, se avete utilizzato una chiave SSH dovrete indicare a VS Code la posizione, e questa verrà salvata nel file di configurazione per le successive connessioni.

Se utilizzi una passphrase per proteggere la tua chiave privata SSH, dovrai inserirla al primo accesso.

Dopo una connessione riuscita, VS Code ti chiederà quale directory aprire sul server remoto. Una volta selezionata la directory, VS Code configurerà automaticamente l'ambiente di sviluppo remoto, installando le estensioni necessarie direttamente dal tunnel SSH attivo.

Conclusioni

Abbiamo visto come sviluppare su un'istanza remota in modo facile ed economico. Come abbiamo visto, i principali benefici di questa soluzione di sviluppo sono il poter centralizzare la configurazione degli ambienti e poter accedere on demand a risorse hardware potenti.

Se sei interessato a ottimizzare il tuo ambiente di sviluppo e scoprire nuove strategie per lavorare in modo più efficiente, continua a seguire il nostro blog. Troverai altri approfondimenti, guide e suggerimenti per sfruttare al meglio gli strumenti e le tecnologie più recenti e per trarre il massimo beneficio dal paradigma Cloud-Native!

About Proud2beCloud

Proud2beCloud è il blog di **beSharp**, APN Premier Consulting Partner italiano esperto nella progettazione, implementazione e gestione di infrastrutture Cloud complesse e servizi AWS avanzati. Prima di essere scrittori, siamo Solutions Architect che, dal 2007, lavorano quotidianamente con i servizi AWS. Siamo innovatori alla costante ricerca della soluzione più all'avanguardia per noi e per i nostri clienti. Su Proud2beCloud condividiamo regolarmente i nostri migliori spunti con chi come noi, per lavoro o per passione, lavora con il Cloud di AWS. Partecipa alla discussione!



Alessio Gandini

Cloud-native Development Line Manager @ beSharp, DevOps Engineer e AWS expert. Computer geek da quando avevo 6 anni, appassionato di informatica ed elettronica a tutto tondo. Ultimamente sto esplorando l'esperienza utente vocale e il mondo dell'IoT. Appassionato di cinema e grande consumatore di serie TV, videogiocatore della domenica.
