

Mettere la testa tra le nuvole

Tre servizi Cloud per sviluppare un'applicazione web, in grado di sostenere traffico elevato. Quasi a costo zero.

Una volta non era sufficiente avere una grande idea per creare un'applicazione web rivoluzionaria, bisognava anche trovare il modo di metterla in linea. Le offerte di hosting a basso costo di solito non consentivano la creazione di applicazioni dinamiche, mentre le soluzioni livello più di alto erano care e complicate da gestire e richiedevano decisione nell'investimento.

Adesso le cose sono cambiate: basta avere un po' di testa per "inventare una nuvola" e per offrire al mercato un'applicazione complessa e giocare il successo in serie A.

Le offerte di cloud computing, infatti, cambiano tutto aprendo interessanti possibilità a chiunque abbia una buona idea, la capacità di realizzarla e un budget minimo.

Esaminiamo le diverse possibilità che sono a disposizione di chi vuole creare a costo minimo un'applicazione web sofisticata. Ci proponiamo di trovare

una soluzione che consenta pagine attive, se possibile con un sistema di template o un framework Mvc (*model, view, controller*). Ci servirà anche un database, non dei più blasonati, non necessariamente Sql, ma in grado di contenere i parametri dinamici delle nostre pagine.

Sarà necessario un sistema di gestione delle identità e della profilazione degli utenti, capace di gestire l'autenticazione, ma anche di archiviare le preferenze e il necessario per consentire un'esperienza personalizzata e gestire l'archiviazione dei dati in modo corretto.

Ovviamente ci importa molto la praticabilità e la prevedibilità dei prezzi, così come la possibilità di sperimentare gratis o a basso costo, prima che il successo del nostro prodotto giustifichi qualche investimento.

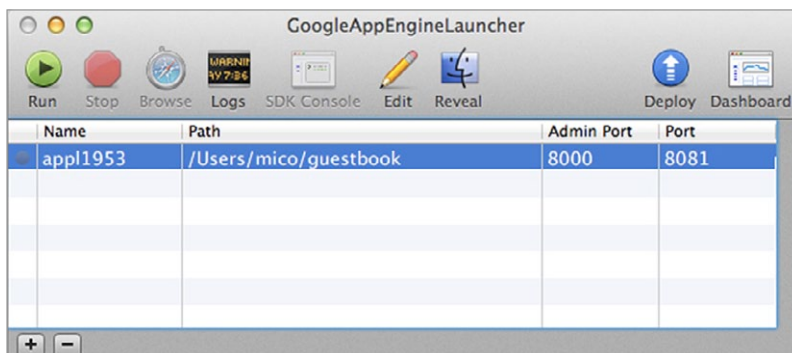
Cominciamo, quindi, a metterci in caccia esaminando le offerte cloud di Google, Amazon e Microsoft, per stare su tre pesi massimi della rete.

Per rimanere nello spirito di un piccolo gruppo di sviluppatori, scegliamo soluzioni *platform as a service*, cioè quelle che mettono a disposizione non un server virtuale da avviare e configurare, ma un ambiente di esecuzione completo, pronto per l'utilizzo. Vogliamo, cioè, una soluzione libera da problemi di installazione, configurazione e amministrazione.

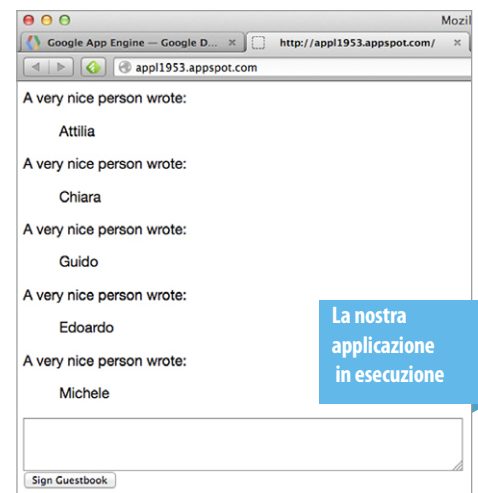
Tutti i servizi che esamineremo, inoltre, permettono a un'applicazione di scalare indefinitamente. Quindi, se andiamo a dormire con un'applicazione qualunque e ci svegliamo con il nuovo Facebook, sapremo di avere centinaia di server in bilanciamento di carico a disposizione senza dover cambiare null'altro che il nostro piano di pagamento.

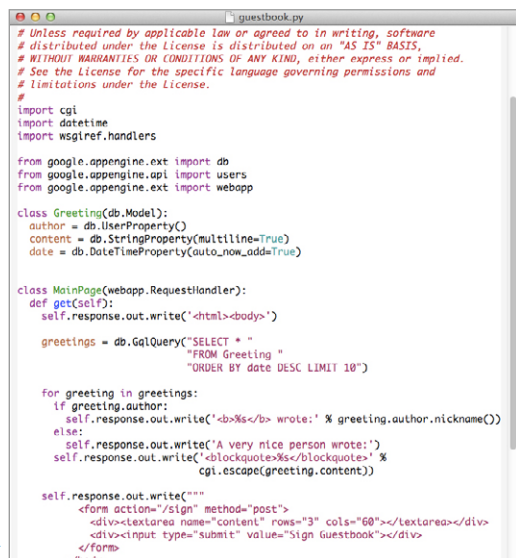
Google AppEngine

Si tratta di una piattaforma che ci è stata simpatica fin dall'inizio perché

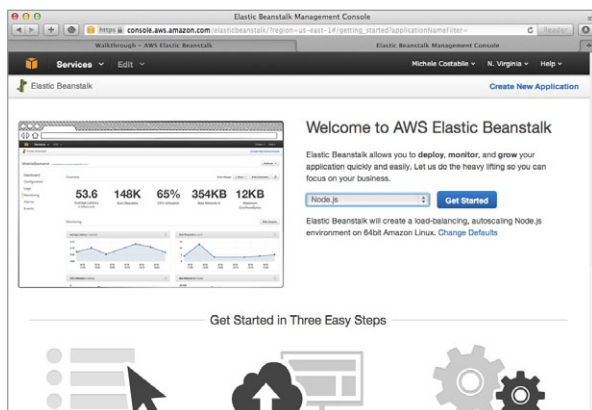


Lo strumento che permette l'esecuzione locale e il deploy delle applicazioni AppEngine.

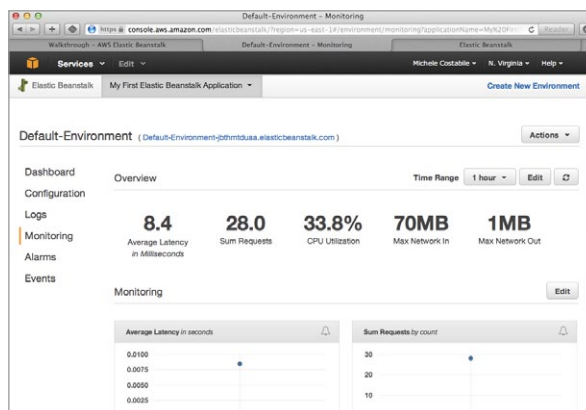




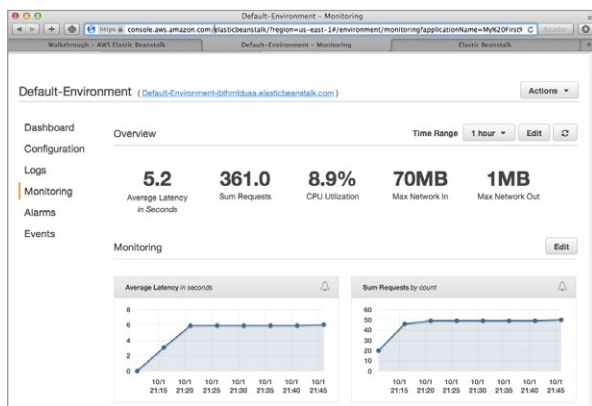
Il codice del nostro guestbook ▶



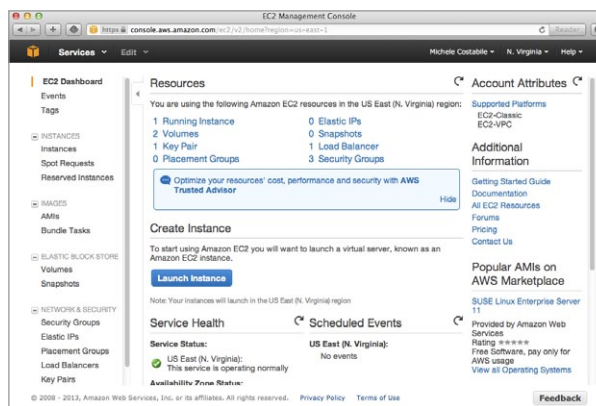
Con Amazon basta collegarsi, scegliere il tipo di server da attivare e crearlo.



Le statistiche di esecuzione di Elastic Beanstalk sono molto puntuali.



Non mancano i grafici del tempo di risposta e del numero di richieste servite.



Le risorse del cloud associate alla nostra applicazione.

Amazon AWS

Ci siamo occupati di recente della piattaforma cloud di Amazon, il primo dei grandi a imboccare la strada di questo genere di servizi. Se parliamo di soluzioni platform as a service nel cloud di Amazon, ci stiamo riferendo a *Aws Elastic Beanstalk* (aws.amazon.com/elasticbeanstalk). Elastic Beanstalk è un servizio che si appoggia sui servizi di virtualizzazione (EC2) e storage (S3) del cloud di Amazon, quindi dalla console dei servizi Aws si possono controllare le risorse allocate a un servizio.

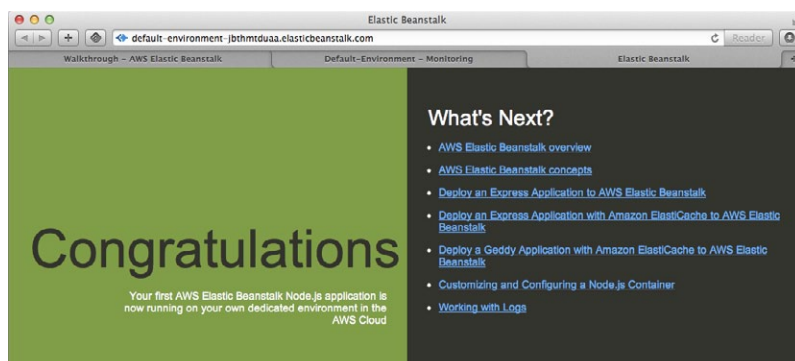
La storia di un'applicazione inizia con la creazione nella console di un ambiente operativo configurato secondo le richieste, per esempio un sistema Windows con IIS o un sistema Linux fornito di Python, dell'ambiente operativo Apache e di tutto il necessario per il tipo di applicazione progettato.

Elastic Beanstalk provvede a gestire la configurazione dei sistemi e l'ambiente di load balancing necessario per far scalare l'applicazione a richiesta.

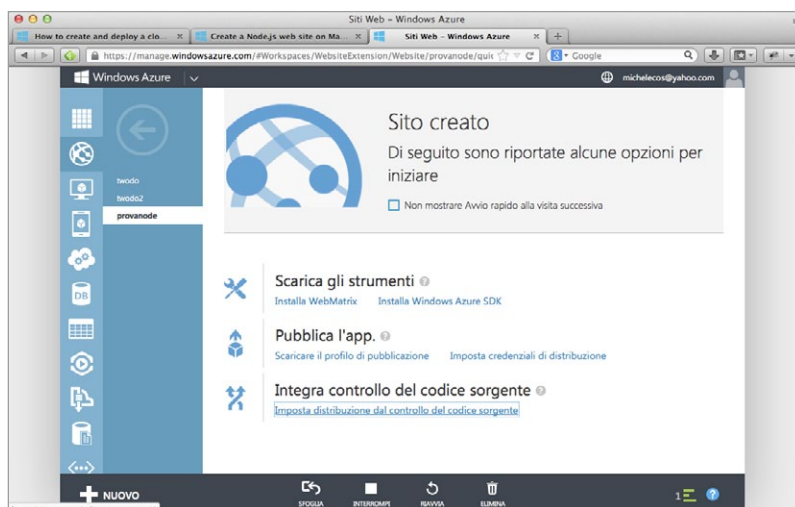
Per quanto riguarda linguaggi e ambienti

di sviluppo, l'offerta copre i tre ambiti principali: l'ambiente .net, i diversi ambienti open source e la macchina virtuale Java. Gli sviluppatori .net sono serviti da un'estensione per Visual Studio che permette di consegnare un'applicazione al cloud di Amazon e vederla in esecuzione nel giro di pochi minuti all'indirizzo *nomeapplicazione.elasticbeanstalk.com*. Amazon supporta anche la consegna di un pacchetto Zip in formato Web Deploy. Lo sviluppo open source è supportato, come c'è da aspettarsi, attraverso un

repository Git, quindi il ciclo di deploy di un'applicazione si interseca con la consegna nell'archivio dei sorgenti di una nuova release, come è conveniente che sia. Nel caso, infine, di Java, si usa la consueta tecnica dell'upload di un archivio in formato War, il modo abituale per chi lavora con Tomcat. Naturalmente è disponibile un plugin Eclipse per chi non sviluppa con Visual Studio. Per iniziare occorre un account Amazon con cui collegarsi alla console di Beanstalk (<https://console.aws.amazon.com>).



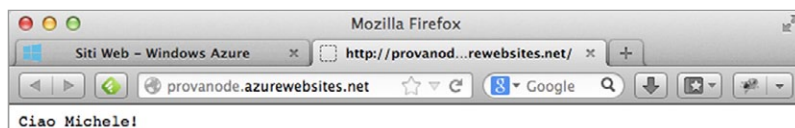
L'applicazione di test node.js in esecuzione.



Con Windows Azure si crea un sito web, si configura un deploy automatico, per esempio da una cartella Dropbox e si carica l'applicazione.



Un esempio minimo di server node.js.



L'applicazione node.js in esecuzione.

com/elasticbeanstalk). Creare un'applicazione è semplicissimo: si sceglie il tipo di applicazione fra quelli gestiti e si fa un clic sul pulsante di creazione.

L'allocazione di una macchina virtuale con il sistema operativo più adatto, una configurazione di sistema opportuna e la creazione di un'infrastruttura che comprende il bilanciamento del carico sono del tutto automatici.

Viene creata un'applicazione di default, che possiamo sostituire con il pacchetto aggiornato quando siamo pronti.

Abbiamo impiegato una decina di minuti per avere in funzione la prima applicazione basata su node.js.

Riassumendo, abbiamo trovato molto facile da usare la fabbrica di applicazioni di Amazon, la console web di gestione è molto chiara ed è migliorata di parecchio rispetto alle versioni precedenti.

La pagina dedicata alla *Account Activity*, raccoglie in maniera ordinata il conto delle risorse utilizzate e il conteggio del dovuto separato per risorsa.

Peccato che l'uso gratuito sia previsto solo per il primo anno e non per un livello di uso delle risorse. Dopo lo scadere del

primo anno di sviluppo bisognerà fare attenzione al numero di macchine che si lasciano accese per un'applicazione in sviluppo, più o meno come faremmo per delle macchine reali.

Windows Azure

Di questa soluzione abbiamo abbondantemente parlato nella rassegna degli ambienti cloud pubblicata sul numero 269 di *PC Professionale*. In sintesi Azure è un cloud facile da usare, con una interfaccia utente curata, che riesce a limare gli spigoli di una configurazione che è giocoforza complessa, per comunicare all'utente una sensazione di semplicità e linearità che lo rendono forse il cloud più simpatico. Come negli altri servizi esaminati partire è solo questione di aprire un account e creare un servizio. Azure non impone scelte tecnologiche tali da legare al cloud di Microsoft. La prima ovvia possibilità è sviluppare applicazioni Windows e integrare tutto l'ambiente con Visual Studio attraverso un insieme di strumenti che si possono scaricare da windowsazure.com. Azure

consente anche di creare applicazioni con Java, Php, Python e Node.js senza doversi occupare della creazione di una macchina virtuale Windows o Linux, come ci siamo proposti nelle specifiche di questa rassegna. Ci siamo divertiti a creare una minuscola applicazione Node.js con Emacs e distribuirla attraverso una cartella Dropbox, diciamo il minimo assoluto che possa emettere una pagina. Azure ci ha supportato con una certa docilità e ci ha consentito di pubblicare l'applicazione senza pagare nemmeno un euro. Il modello di pagamento di Azure, infatti, è il più amichevole per chi sta solo provando a mettere i piedi nell'acqua, ma non ha ancora deciso di tuffarsi.

Il cloud Microsoft offre la possibilità di usare Active Directory per la gestione degli utenti e realizzare l'autenticazione a due fattori, per capirci con login e password e una conferma su cellulare. L'importo previsto del prossimo pagamento è ben visibile nel pannello di controllo, per aiutare a tenere sotto controllo i costi per tempo. Come avviene con il cloud di Amazon, si possono combinare insieme diversi servizi, si può collegare un server di database e si possono mischiare semplici web service e macchine virtuali. Il database a disposizione è, ovviamente, Microsoft Sql Server. La documentazione, che è ricchissima e parzialmente in Italiano, semplifica l'approccio alla rete.

Conclusioni

Non ci sono ostacoli sulla strada dello sviluppatore individuale o della piccola startup che vogliano provare a realizzare un'idea tralasciando completamente i problemi di pianificazione della crescita man mano che il progetto decolla. Tutte le soluzioni analizzate permettono lo sviluppo su un qualsiasi portatile, la pubblicazione nel modo più semplice dell'applicazione e la possibilità di tenere acceso un server per il tempo necessario a fare i test con un impegno minimo di capitale.

Tutto questo, senza precludere la possibilità di passare automaticamente a uno scenario fatto di decine o centinaia di macchine con bilanciamento di carico automatico, mentre il numero di utenti cresce e il database comincia a essere misurato in giga. Diventa così più facile concentrarsi su quello che è il nucleo del business: pensare un grande servizio, realizzarlo rapidamente, farlo crescere viralmente. •