

- Roberto Freato, Marco Parenzan -

Sviluppare per il **Cloud** con **Microsoft Azure**

Guida completa



[Installazione e configurazione >>](#)

[Le caratteristiche di Microsoft Azure >>](#)

[Creare servizi in modo facile e veloce >>](#)

[Implementare i servizi nel mondo reale >>](#)

***pro**
DigitalLifeStyle

*pro
DigitalLifeStyle

Sviluppare per il Cloud con
Microsoft
Azure
Guida completa

Roberto Freato, Marco Parenzan

EDIZIONI
LSWR

Titolo originale: Mastering Cloud Development using Microsoft Azure

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78217-333-5

www.packtpub.com

Copyright © 2016 Packt Publishing

Edizione italiana: Sviluppare per il cloud con Microsoft Azure | Guida completa

Editor in Chief: Marco Aleotti

Progetto grafico: Roberta Venturieri

Immagine di copertina: © Jane Kelly | Thinkstock

© 2017 Edizioni Lswr* - Tutti i diritti riservati

ISBN: 978-88-6895-502-1

I diritti di traduzione, di memorizzazione elettronica, di riproduzione e adattamento totale o parziale con qualsiasi mezzo (compresi i microfilm e le copie fotostatiche), sono riservati per tutti i Paesi. Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633.

Le fotocopie effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate a seguito di specifica autorizzazione rilasciata da CLEARedi, Centro Licenze e Autorizzazioni per le Riproduzioni Editoriali, Corso di Porta Romana 108, 20122 Milano, e-mail autorizzazioni@clearedi.org e sito web www.clearedi.org.

La presente pubblicazione contiene le opinioni dell'autore e ha lo scopo di fornire informazioni precise e accurate. L'elaborazione dei testi, anche se curata con scrupolosa attenzione, non può comportare specifiche responsabilità in capo all'autore e/o all'editore per eventuali errori o inesattezze.

L'Editore ha compiuto ogni sforzo per ottenere e citare le fonti esatte delle illustrazioni. Qualora in qualche caso non fosse riuscito a reperire gli aventi diritto è a disposizione per rimediare a eventuali involontarie omissioni o errori nei riferimenti citati.

Tutti i marchi registrati citati appartengono ai legittimi proprietari.

EDIZIONI
LSWR

Via G. Spadolini, 7

20141 Milano (MI)

Tel. 02 881841

www.edizionilswr.it

Printed in Italy

Finito di stampare nel mese di marzo 2017 presso "Rotomail Italia" S.p.A., Vignate (MI)

(*) Edizioni Lswr è un marchio di La Tribuna Srl. La Tribuna Srl fa parte di **LSWR GROUP**.

Sommario

1. COSTRUIRE I SERVIZI DI BASE.....	9
CloudMakers.XYZ.....	10
<i>Scenario 1: Avvio della compagnia</i>	11
Impostazione dell'ambiente di sviluppo.....	13
<i>Dov'è il mio datacenter?</i>	16
<i>Creazione della macchina di sviluppo</i>	17
<i>Generalizzazione della VM</i>	22
Implementazione del processo di sviluppo.....	25
<i>Creazione di un nuovo ambiente VSTS</i>	27
<i>Integrazione con Visual Studio</i>	30
<i>Integrazione di servizi di terze parti</i>	32
Automatizzare le operazioni ripetitive.....	34
<i>Azure Automation</i>	35
<i>Azure Scheduler</i>	42
2. INTEGRARE L'ESPERIENZA IBRIDA.....	45
CloudMakers.XYZ.....	45
Service Manager (Classic) Mode vs. Resource Manager Mode.....	47
Configurare una Virtual Network per i server di frontend.....	49
Configurare una Virtual Network in Service Mode per i server di backend.....	54
Connettere due Virtual Network con una connessione Site-To-Site.....	57
Configurare la connessione Point-To-Site per connettersi dall'ufficio.....	59
Creare una Virtual Machine in una Virtual Network ARM.....	62
3. COSTRUIRE UN FRONTEND WEB.....	67
Sono sempre dati.....	67
Polyglot Persistence.....	70
Scenario.....	71
Scrivere un catalogo in Azure DocumentDb.....	74
Costruire un frontend con ASP.NET MVC.....	87
Cercare nel catalogo con Azure Search.....	91
Memorizzare dati non strutturati in Azure Storage.....	98

Accelerare l'accesso ai dati con la cache di Azure Redis.....	105
Persistere il carrello in Azure Table Storage	109
Pubblicare l'applicazione in una Azure Web App.....	113
Configurare una web application con un nome di dominio custom.....	119
4. COSTRUIRE IL BACKEND	123
L'evoluzione dello scenario CloudMakers.XYZ	124
Gestione degli ordini	126
<i>Disaccoppiare le azioni utente dalle operazioni SQL</i>	127
<i>Produzione di dati denormalizzati per l'analisi</i>	134
<i>Estrazione, trasformazione e caricamento dei dati su Data Warehouse</i>	136
<i>Predizione di informazione di valore con Machine Learning</i>	145
Esternalizzare la gestione dell'identità	152
<i>Azure AD e autenticazione Claims-based</i>	154
<i>Azure Active Directory B2C</i>	156
<i>Funzionalità avanzate</i>	161
5. COSTRUIRE UN'ESPERIENZA MOBILE	165
Il caso di una "technical community" (comunità tecnica di utenti)	165
Scrivere un'applicazione mobile.....	167
Sviluppare un'applicazione mobile con i Visual Studio Tools per Apache Cordova.....	170
La necessità di una soluzione facile e veloce	178
Introduzione alle mobile app.....	179
Ottenere i dati dal servizio.....	186
Inviare dati al servizio.....	189
Azure e Node.js	193
6. COSTRUZIONE DELLO STRATO API	199
Evoluzione dello scenario CloudMakers.XYZ	200
Costruzione dell'API	201
<i>Usare ASP.NET Web API</i>	202
<i>ServiceStack</i>	206
<i>Documentazione dell'API con Swagger/OpenAPI</i>	207
Rilascio dell'API.....	209
<i>Funzionalità avanzate di App Service</i>	210
<i>API Gateway</i>	217
Usare l'Azure API Management	218
<i>Creazione di una API</i>	219
<i>Usare le policy</i>	222
<i>IAM (Identity and Access Management)</i>	226
<i>APIM Management avanzato</i>	227
Debug di workflow API.....	228
<i>Debug remoto con App Service</i>	229
<i>Usare il Service Bus Relay per debuggare i servizi REST</i>	230

7.	LAVORARE CON I MESSAGGI.....	233
	Gestire il futuro con i comandi.....	234
	Messaggi e code.....	235
	Lo scenario.....	237
	Mandare i messaggi in coda.....	237
	Processare i messaggi con i worker.....	242
	Gestire il passato con gli eventi.....	247
	Rispondere a ogni singolo evento.....	249
	Gestire gli eventi come uno stream.....	255
8.	RILASCIARE SOLUZIONI IN AZURE.....	263
	Azure Resource Manager.....	264
	Scrivere il template.....	266
	Programmare il template.....	274
	Organizzare attività di deployment con il pulsante "Deploy to Azure".....	278

Costruire i servizi di base

Il PaaS è una delle più grandi innovazioni che il Cloud ha reso disponibili negli ultimi anni ed è ormai sbarcato a pieno titolo nel panorama della produzione del software e dei servizi.

Con il **Platform-as-a-Service**, le aziende possono finalmente astrarre i loro servizi dall'ambiente di esecuzione, riducendo enormemente lo sforzo di governance connesso alla gestione di sistemi IT complessi per focalizzarsi completamente sugli aspetti primari dell'attività (*core business*).

Azure tuttavia non è solo PaaS, e in questo capitolo infatti affronteremo i più noti servizi di IaaS. Quando si parla di **Infrastructure-as-a-Service** si parla di solito di uno o più servizi che forniscono infrastruttura IT (macchine virtuali, connettività) senza richiedere tuttavia competenza e oneri di gestione degli strati più bassi dello stack che si dovrebbe gestire in ambito on-premise.

In questo capitolo presentiamo il primo di una serie di scenari che utilizzeremo nel libro per meglio esprimere l'utilizzo dei vari componenti di Azure. In particolare, vedremo come una azienda immaginaria voglia iniziare la propria attività, minimizzando l'investimento IT a il time-to-market. Vedremo come molti componenti/servizi di Azure (come le Macchine Virtuali, Visual Studio Team Services, Azure Resource Manager, Azure Storage) si possano combinare insieme per fornire una piattaforma di base per la divisione sviluppo della compagnia.

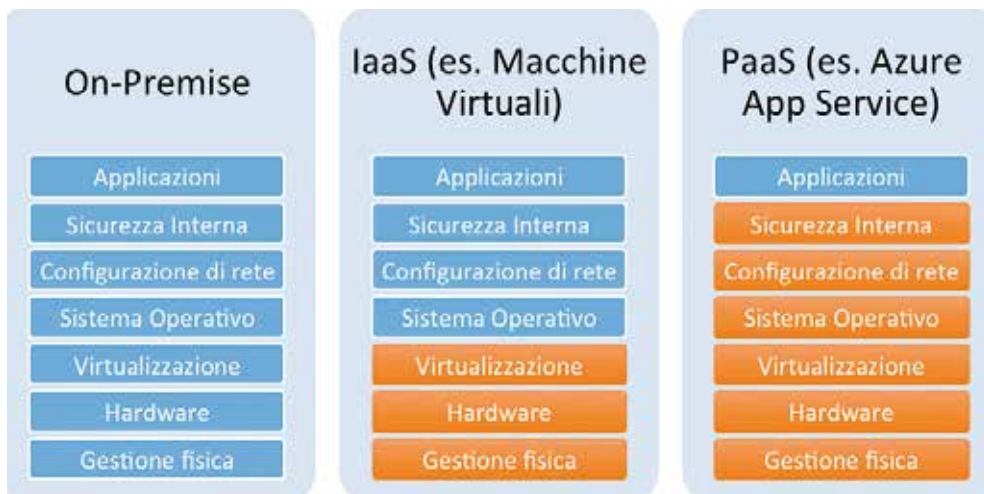


Figura 1.1 - Il grafico mostra quale sia il confine tra ciò che compete ai clienti e cosa al vendor di Cloud nei diversi scenari.

CloudMakers.XYZ

Abbiamo deciso di simulare dei casi reali creando una realtà fittizia che aiuti a contestualizzare meglio le necessità di Cloud che andremo ad affrontare. La compagnia si chiama CloudMakers.XYZ e nel resto del libro potremmo chiamarla anche CloudMakers o solo CM.

Il libro sviluppa quattro scenari che esplorano, passo dopo passo, le necessità di CM, focalizzandosi su come Microsoft Azure si possa usare per fornire vari servizi:

- **Scenario 1:** Avvio della compagnia.
- **Scenario 2:** Sviluppo del prodotto di e-Commerce.
- **Scenario 3:** Sviluppo del prodotto GeoProvider.
- **Scenario 4:** Connettere i servizi e automatizzare.

Questi quattro scenari sono stati concepiti per creare un percorso coerente per quei lettori che abbiano già confidenza con le tematiche da gestire durante lo sviluppo di software e servizi. Per questo è più utile imparare a conoscere i provider di Cloud Computing come Microsoft Azure, toccando con mano scenari di interconnessione, piuttosto che con esempi isolati e "Hello World" verticali sui singoli servizi.

Siamo stati ispirati dall'esperienza sul campo per fornire gli esempi che troverete negli scenari e la maggioranza dei consigli, delle procedure e gli esempi di codice arrivano proprio da ambienti di produzione.

NOTA

Per quanto spesso nei libri si faccia riferimento alle “best practice” o alle raccomandazioni di utilizzo, vorremmo che questo libro sia inteso più come un punto di vista. Solo il lettore conosce il proprio caso specifico e ciò che accade nella sua realtà: perciò vi raccomandiamo di considerare che le architetture indicate non sono universalmente o in assoluto quelle “giuste” e tutto dipende dai casi d’uso.

Procediamo ora con il primo scenario, dove CloudMakers avvia la propria divisione sviluppo.

Scenario 1: Avvio della compagnia

CloudMakers è una startup con sede a Milano in Italia. I fondatori sono:

- Han: il CEO (Amministratore Delegato), arriva da una forte esperienza nella gestione aziendale. Le sue migliori qualità sono la *leadership* e l’impegno: interpreta nel modo più opportuno gli input derivanti dal lavoro del team tecnico e dal CTO, trasformandoli in opportunità di mercato. Tuttavia, non ha alcuna conoscenza di come si sviluppi il software.
- Luke: il CTO (Responsabile della Tecnologia), arriva da una forte competenza tecnica. Si occupa di prototipare i nuovi prodotti, aiuta nella selezione dei candidati e, in generale, è responsabile della qualità del software prodotto da CM.
- Leila: il CFO (Responsabile Finanziario) e PM (Project Manager) principale, è responsabile di tutta la strategia finanziaria di CM ed è a capo di tutte le operazioni di produzione software.

In aggiunta a queste tre risorse chiave ci sono altre figure:

- Owen: il professionista IT senior, con una forte esperienza nella gestione delle infrastrutture IT. Prima di lavorare per CM, era responsabile dell’intera struttura IT fisica della sua vecchia azienda.
- Flint: uno sviluppatore full-stack, con competenze trasversali su tutti gli strati applicativi. Arriva da una precedente esperienza di startup fallita ed è stato assunto da Luke per sviluppare i prodotti cardine dell’azienda e imparare da lui tutto quanto necessario sul Cloud Computing e su Microsoft Azure.

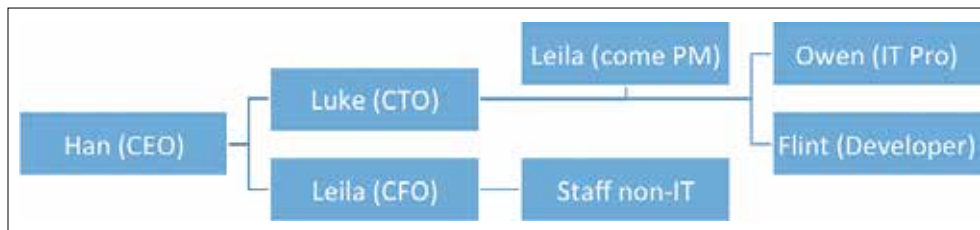


Figura 1.2 - Una sintesi delle risorse più rilevanti di CloudMakers.

CM è stata fondata con un budget molto limitato e la decisione dell'organo amministrativo è di minimizzare gli investimenti in infrastruttura IT per dare priorità alle iniziative di marketing e pubblicità fino alla *break even*. Sotto questa ipotesi, ogni investimento *up front* dovrebbe essere valutato e approvato, mentre il modello di *pay-as-you-go* è fortemente incentivato.

Luke ha pianificato di costruire la nuova componente principale di infrastruttura su Microsoft Azure, usando il Cloud per uscire sul mercato velocemente e produrre efficacemente i nuovi servizi, così come gli ambienti di sviluppo per l'intera azienda e per il team tecnico.

Ha anche pianificato di implementare una logica BYOD (Bring Your Own Device) per ridurre l'onere di acquisto, gestione e manutenzione dei dispositivi fisici di lavoro e accesso alla rete aziendale. Gli impiegati sono così invitati a usare i propri computer portatili e cellulari per il lavoro, sia a casa che in ufficio.

Luke infine ha anche pianificato di costruire un intero ambiente di sviluppo in Cloud, in modo che i dipendenti possano collegarsi a esso tramite **RDP (Remote Desktop Protocol)** per lavorare (guarda la sezione *Impostazione dell'ambiente di sviluppo*).

La maggior parte del codice sarà scritto usando Visual Studio e siccome Luke proviene da una lunga esperienza d'utilizzo di **TFS (Team Foundation Server)** per gestire gran parte dell'**ALM (Application Lifecycle Management)**, CM utilizzerà Visual Studio Team Services (ex Visual Studio Online) per archiviare il codice sorgente dell'azienda, così come per aiutare Leila a tenere traccia delle attività correnti del team tecnico (guarda la sezione *Implementazione del processo di sviluppo*).

Su richiesta di Owen, sovraccaricato dal crescere delle richieste di Flint di preparare ambienti di **Sviluppo/Test/Produzione** molto complessi, Han ha chiesto a Luke di trovare una soluzione appropriata che semplifichi e automatizzi il processo di creazione e manutenzione di questi ambienti, considerato che sono usati per fornire ai clienti le demo dei prodotti e che dovrebbero essere attivati il prima possibile con un impegno minimo da parte dell'amministratore (guarda *Automatizza le operazioni ripetitive*).

Nel Capitolo 2 vedremo come CM possa migliorare il processo appena descritto con particolare attenzione alla sicurezza, al disaster recovery e al controllo delle risorse.

Impostazione dell'ambiente di sviluppo

In questo libro abbiamo già dato per scontato che la scelta del *Cloud vendor* sia già stata presa. Tuttavia, in generale, quando si pianifica un primo contatto con il Cloud, ci sono alcune importanti domande che avrebbero bisogno di essere indirizzate per poter cominciare a lavorare:

- Cosa mi serve da un fornitore di Cloud?
- Quali fornitori hanno i prodotti/servizi che mi servono?
- Quanto costano, comparati tra loro?
- Ci sono vantaggi/svantaggi legati alla scelta o meno di un particolare vendor?
- Dov'è situato il mio business principale e, più importante, da dove saranno consumati principalmente i servizi che andrò a creare?

Queste sono solo alcune delle tante domande iniziali che dovremmo porci prima di fare una scelta e pianificare l'adozione. Più in generale, dovremmo fare una *vendor selection* così come in qualsiasi altro ambito.

IaaS	PaaS
Amazon Web Service	Salesforce
Microsoft Azure	Microsoft Azure
Rackspace	Google Cloud Platform
CenturyLink	Red Hat
Google Cloud Platform	Progress
VMWare	IBM Softlayer
IBM Softlayer	SAP

Figura 1.3 - Alcuni dei provider di Cloud Computing più importanti sul mercato, per i servizi IaaS e PaaS.

Nel contesto di CM, considerato il requisito (iniziale) della ricerca di un ambiente di sviluppo sul Cloud, probabilmente dei servizi IaaS sono sufficienti. Tuttavia, siccome molto spesso lo scenario evolve (e nel testo evolverà), è giusto pensare che per sfruttare al meglio i vantaggi del Cloud, forse le sole VMs (Virtual Machines - Macchine Virtuali) non siano sufficienti.

Queste altre domande potrebbero aiutare a identificare i servizi di cui l'azienda potrebbe aver bisogno.

- Abbiamo in programma di vendere? **SaaS (Software-as-a-Service)**?
 - Costruire servizi SaaS sfruttando servizi PaaS può essere molto efficace in termini di pro e contro. È provato che una minore attenzione sulla gestione

dell'infrastruttura riduce i costi e allo stesso tempo permette un maggior coinvolgimento su quello che per l'aziende è spesso definito *core business*.

- Il modello di costo del PaaS (tipicamente a consumo e *on-demand*) ben si sposa con i modelli di tariffazione di un servizio SaaS, aiutando l'azienda a bilanciare correttamente il *cash flow*, evitando la complessità economica, amministrativa e operativa di un approccio CAPEX.
- Abbiamo uno staff IT dedicato, capace di gestire/governare l'infrastruttura (ma soprattutto disponibile a farlo)?
 - Se la risposta fosse "no", si dovrebbe considerare **quasi esclusivamente** il PaaS, evitando il ricorso allo IaaS che, per quanto possa essere limitato, introduce un considerevole onere amministrativo.
- Abbiamo un orientamento predominante tra *make* o *buy*?
 - Nel caso l'azienda si trovi più a suo agio con scelte di *make* (produrre internamente piuttosto che ricorrere all'acquisto), il PaaS può essere evitato e lo IaaS permette un ampio controllo e, dal solo punto di vista dei "consumi", un ridotto risparmio.
 - Nel caso invece si trovi a suo agio con scelte di *buy*, il PaaS potrebbe essere ottimale. In questo caso "l'unità di acquisto" è uno o più servizi a valore aggiunto, che possono spaziare dall'aver una VM gestita (e quindi preoccuparsi solo della propria applicazione) a un sistema di ricerca chiavi in mano e così via. L'eventuale maggior costo (non sempre i PaaS costano di più e spesso hanno un costo analogo ai rispettivi IaaS, per fidelizzare il cliente al Cloud vendor) è giustificato proprio dal valore aggiunto.
 - Se, come è giusto sperare, venisse adottato un buon mix tra il *make* e il *buy*, allora entrambi i mondi avrebbero senso di coesistere, ed è esattamente ciò che accade nella maggioranza dei casi reali.

NOTA

Quest'ultimo è l'approccio preferibile, visto che non esiste un "ottimo" assoluto ma solo un "ottimale" relativo alla realtà di riferimento. Il PaaS è probabilmente un approccio e un trend in continua crescita visto il suo dialogo efficace con il business, ma molte aziende hanno bisogno di IaaS per gestire alcune aree specifiche di prodotto. È quindi giusto considerare entrambi due attori che sono destinati a restare sul mercato a lungo.

Questo probabilmente non è il posto più adatto per affrontare un paragone tra i vari Cloud vendors, innanzitutto perché è un'impresa tutt'altro che semplice e poi sarebbe fuori dal tema del libro, che invece può essere riassunto in questo principio:

Imparare a riconoscere, data una esigenza di prodotto/servizio, quali servizi cloud possono essere utili per accelerarne l'uscita e ridurre i costi complessivi di esercizio.

A ogni modo, chiunque conosca perlomeno gli attori più importanti del mercato del Cloud pubblico (in ordine alfabetico):

- Amazon Web Services.
- Google Cloud Platform.
- Microsoft Azure.

Nel momento di stesura di questo libro, ognuna delle tre piattaforme ha una offerta Cloud comparabile: la scelta dovrebbe quindi essere intrapresa non prima di una analisi.

Quindi, tornando alle domande, se avessimo bisogno di IaaS, ecco alcuni utili quesiti:

- Il vendor fornisce VM? Di che dimensioni?
- L'ambiente è personalizzabile (rete, dischi, bilanciatori)?
 - E a che grado/livello? Tale da colmare il nostro bisogno?
- Viene proposto/imposto l'utilizzo di qualche attrezzatura fisica nella scelta di uno o dell'altro vendor?
 - Che valore aggiunto introduce? Quali costi?

Al contrario, se avessimo bisogno di PaaS, ecco alcuni altri quesiti:

- Abbiamo già alcuni prodotti/servizi da pubblicare?
 - In quali linguaggi sono stati scritti?
 - Che dipendenze hanno (sistema operativo, runtime, tools)?
- Il PaaS in questione accelera effettivamente il nostro modo di concepire il processo di sviluppo?
 - Ne propone uno alternativo?
- C'è una buona integrazione tra i servizi dello stesso Cloud vendor?
 - È una pratica che viene enfatizzata? Per creare dipendenze (*lock-in*) o per dimostrare *openess* (magari verso soluzioni multi-cloud)?
- È chiaro il grado di *lock-in* del prodotto PaaS?
 - Quanto ci costerebbe uscirne, nel caso di una migrazione?
- C'è una curva di apprendimento per poter andare live?
 - Quanto ci costerebbe, nel caso di adozione?

Un maggior numero di domande corrisponde a una maggiore incertezza rispetto all'adozione, che reputiamo essere una posizione condivisibile in qualsiasi processo di adozione di un prodotto o, come nel Cloud Computing (e nel PaaS soprattutto), di un intero nuovo modo di pensare al business.

Infine, rimane da scegliere un aspetto che può sembrare secondario (ma non lo è): la posizione del datacenter in cui risiederanno i nostri servizi.

Dov'è il mio datacenter?

Nel Cloud pubblico non si presenta quasi mai il concetto di datacenter "privato", tantomeno di rack/server/zona "dedicata", in quanto le risorse sono, nella maggior parte dei casi, condivise tra i più tenant (clienti) sparsi sul territorio. Questo tema evidenzia come debbano essere trattati con attenzioni i temi della sicurezza, dell'isolamento e della disponibilità delle risorse. Temi che sono comunque principalmente di competenza del Cloud vendor.

Tuttavia, c'è un momento in cui è necessario (anche solo per i test) posizionare i nostri dati, il nostro codice o la nostra macchina virtuale: se il nostro business (inteso sia come *corporate* che come *customers*) è maggiormente concentrato negli Stati Uniti, sicuramente eviteremmo un datacenter in Asia, per via dell'alta latenza dovuta alla distanza fisica.

NOTA

La latenza indica il tempo che intercorre tra la partenza di un segnale da una sorgente e il suo arrivo a destinazione, spesso anche indicato come "tempo di risposta". In caso mandassimo un segnale satellitare, la latenza media sarebbe circa 600ms, che significherebbe più di un secondo per un *round-trip* completo (come per esempio una richiesta HTTP). La latenza, oltre che dalla qualità dei materiali costruttivi delle linee e dagli apparati di rete, dipende anche in modo proporzionale dalla distanza, motivo per cui quest'ultima è una fattore decisivo nella scelta di un datacenter.

Allo stesso modo, se decidessimo di posizionare i servizi in un datacenter vicino alle sedi della compagnia (per esempio UK) ma distanti dai fruitori del servizio (che potrebbero essere i clienti asiatici), faremmo nuovamente una scelta non ottimale. Il meglio potrebbe essere, previa definizione dei carichi, avere la possibilità di fare *deployment* su regioni diverse, in modo da creare repliche dei servizi il più vicino possibile agli utilizzatori finali.

Le **CDN (Content Delivery Networks)**, trattate in seguito nel libro, rappresentano una soluzione per "accorciare le distanze" tra il contenuto e l'utente. Mentre i datacenter (o più propriamente le *region*) sono appena manciate, i nodi di CDN sono centinaia e sparsi appositamente su tutto il pianeta, anche fuori dal perimetro dei datacenter del provider. Per fare un esempio, una Web Application che tratta foto per utenti sparsi in tutto il mondo, potrebbe avere la seguente topologia:

- Pubblicazione dell'applicazione Web su alcuni (pochi o appena uno solo) Datacenter.
- Replica dei contenuti (le foto) su tutti gli *edge-nodes* del pianeta tramite una CDN.

Il risultato è una migliorata velocità di navigazione dell'utente, siccome il contenuto è cachato/servito da un nodo molto più vicino a esso che l'origine sul DC.

Creazione della macchina di sviluppo

Un buon punto per iniziare a costruire l'ambiente di sviluppo dell'azienda è creare la prima VM per i dipendenti. Negli ultimi anni, sviluppare software su macchina virtuale piuttosto che su Workstation fisica è diventato abbastanza comune e, sebbene non la prassi, sicuramente un modo per ottenere vari vantaggi:

- Riduzione della necessità di mantenere aggiornato il parco hardware per stare al passo con le necessità di performance degli ambienti di sviluppo.
- Possibilità di usare hardware leggero (*thin client*, netbooks) come proxy verso la VM.
- Centralizzazione della governance in un singolo punto, ove è più semplice fornire supporto e svolgere attività di manutenzione.
- Fornitura di ambienti di sviluppo in tempi rapidissimi.

In Microsoft Azure, le VM sono fornite in due sistemi operativi: Windows e Linux. I requisiti per creare una VM sono:

- Uno Storage Account di backend, dove i dischi della VM sono salvati.
- Una **Virtual Network** che contenga la VM.
- Un IP pubblico, se la VM deve essere raggiunta dall'esterno.
- Un **Network Security Group (NSG)**, che definisce le regole di sicurezza/firewall tra le VM dentro le Virtual Network in Azure.

Siccome le Virtual Network, gli NSG e gli indirizzi IP sono discussi nei prossimi capitoli, vediamo cosa si intende per **Storage Account**. Questo account, così come gli altri requisiti in questione, può essere creato durante il processo stesso di provisioning della VM, nonostante sia meglio pianificarne la creazione in anticipo, secondo questa analisi:

- Quanti Storage Account dovremmo creare?
 - Uno per VM? Uno per divisione? Uno per tutti?
- Quali sono i vincoli di uno Storage Account?
 - Posso creare un numero indefinito di account?
 - Ogni account ha limiti (in termini di capacità)? Banda?
 - Ci sono tipi di Storage diversi che possiamo usare?

Queste domande dovrebbero aiutare a capire quanto sia importante fermarsi a pensare prima di creare compulsivamente nuove risorse sul Cloud. È una questione di igiene e di organizzazione e, sebbene sia vero che uno dei maggiori *mantra* del Cloud sia il termine “elastico” (in effetti possiamo creare e distruggere quasi ogni risorsa in pochi secondi), è raccomandato evitare troppi *wizard* e approcci *next>next*.

NOTA

In questo testo utilizzeremo l'inglese per tutte le parole chiave connesse ad Azure, così come per gli screenshot e le terminologie specifiche. Non solo la documentazione reperibile in rete è più che altro localizzata in inglese, ma crediamo che alcuni termini tecnici siano utilizzati, anche in Italia, più nella loro accezione originale che nella, spesso fuorivante, traduzione in italiano.

Creiamo quindi un nuovo Storage Account:

1. Vai su <https://portal.azure.com> (il nuovo portale) e trova la tua Directory/Subscription nell'angolo in alto a destra.
2. Cerca l'azione “**New**” e scegli “**Storage**” (oppure un altro gruppo semanticamente-equivalente).

NOTA

Diciamo “semanticamente-equivalente” siccome vorremmo astrarre dall'esatta sequenza di passi da compiere, in favore di un approccio più elastico, basato sui concetti. Pensiamo che un approccio basato sul ragionamento sia più efficace che una sequenza di istruzioni passo passo.

3. Cerca **Storage Account** (quello con l'icona verde) e segui il processo di creazione guidata, ricordando di specificare sempre “**Resource Manager**” come modello di *provisioning*, prima di iniziare il processo di creazione.

NOTA

Azure sta completando il passaggio dal “vecchio” modello (noto come **Classic**) al nuovo modello (noto come **Resource Manager**). Questo nuovo modello di provisioning/deployment delle risorse porta con sé notevoli miglioramenti e i nuovi servizi sono disponibili solamente in quest'ultima e più recente modalità.

Durante il processo di creazione, è richiesto qualche input:

- **Name:** è il nome dell'account da creare ed è anche il prefisso DNS sul quale risiederà il servizio (nella forma *.core.windows.net). Va scelto secondo le linee guida (<https://msdn.microsoft.com/it-it/library/azure/dd135715.aspx>) e non può essere cambiato dopo la creazione.
- **Account kind:** di default gli Storage Account includono 4 servizi (**Blob, Table, Queue, File**) ma molto spesso gli utenti ne utilizzano il solo servizio Blob. È quindi possibile indicare "General purpose" per avere un account completo oppure "Blob storage" per averne uno limitato alla sola parte Blob.
- **Performance:** può essere standard (a basse performance) o premium (ad alte performance, con dischi SSD di backend).
- **Replication:** definisce quante repliche del servizio (per ragioni di disaster recovery o alta disponibilità) debbano essere previste. Il livello più alto fornisce una replica geografica in una regione adiacente il cui accesso è disponibile in lettura (eventualmente per suddividere il carico). Il livello più basso prevede solamente delle repliche interne alla zona per il disaster recovery interno di Azure.
- **Storage service encryption:** permette di abilitare la crittografia dei contenuti prima che vengano scritti nei dispositivi del datacenter.
- **Resource group:** si può pensare ai gruppi di risorse come a dei gruppi logici dove posizionare le risorse. Ogni risorsa (creata secondo il modello Resource Manager) deve essere piazzata all'interno di un Resource Group. In base alle necessità di progetto, va scelto il corretto gruppo per suddividere in modo coerente le risorse (un utilizzo comune dei RG è quello di dividere le risorse in "centri di costi").

Nei passi seguenti, costruiremo un nuovo ambiente Windows Server pulito da customizzare successivamente con le applicazioni previste affinché rappresenti la base da cui creare le macchine di sviluppo:

1. Cerca nella casella di ricerca il nome dell'ambiente che desideri creare (per esempio "Windows Server" o, come per l'esempio sotto "Visual Studio Community 2015").
2. Seleziona l'immagine più appropriata (come nella **Figura 1.4**).
3. Di nuovo, assicurati di scegliere "**Resource Manager**" come modello di provisioning, prima di continuare con la creazione.




NAME	PUBLISHER
 Visual Studio Community 2015 Update 3 with Azure SDK 2.9 on Windows Server 2012 R2	Microsoft
 Visual Studio Community 2015 with Universal Windows Tools and Azure SDK 2.9 on Windows 10 Enterprise N (x64)	Microsoft
 Visual Studio Community 2015 Update 3 with Universal Windows Tools and Azure SDK 2.9 on Windows 10 Enterprise N (x64)	Microsoft

Figura 1.4 - Queste immagini forniscono una versione di Windows Server con pre-installato Visual Studio.

Seguendo i passi della procedura, ti verrà chiesto:

- **Basics:** informazioni riguardo all'amministratore, alla password e al tipo di storage del disco principale.
- **Size:** la dimensione della VM. Ci sono varie opzioni, visto che le possibilità ricoprono una vasta gamma di combinazione CPU/Memoria (per le istanze definite "general-purpose") e di hardware ottimizzato (per le istanze con GPU oppure con processori Xeon ottimizzati). Alcune istanze, per esempio le DSx (dove "x" è un numero progressivo), indicano l'utilizzo di dischi SSD per il sistema operativo e un elevato numero di IOPS.

Così come per lo Storage account (e in effetti per qualsiasi servizio di Azure) possiamo sceglierne il tipo, che costituisce il set di funzionalità. Nel caso delle VM, le differenze sono:

- **Core, Memoria e dischi dati:** metriche che definiscono principalmente delle performance.
- **Load balancing:** questo definisce la possibilità di introdurre la VM all'interno di un anello bilanciato (nel caso si voglia creare una sola VM, l'opzione è irrilevante).
- **Auto scaling:** la possibilità di creare dinamicamente nuove istanze della VM al verificarsi di alcuni eventi.
- **Funzionalità avanzate:** IOPS controllati, dischi locali SSD e supporto RDMA sono inclusi nelle versioni VM più potenti (e costose).
- **Settings:** come accennato prima, utilizziamo le impostazioni di default per Network, IP, NSG poiché trattati in seguito. Possiamo tuttavia specificare uno Storage account esistente per le immagini principali.