

# Running, improving and maintaining a site in the real world

The logo for aspitalia.com, featuring a small cluster of blue dots to the left of the text "aspitalia.com" in a sans-serif font.The Microsoft Regional Director logo, featuring the Microsoft Dynamics logo (a blue circle with three white segments) to the left of the text "Microsoft Regional Director" in a sans-serif font.The icubed logo, featuring a stylized blue "i" with three horizontal bars to its left, followed by the word "cubed" in a sans-serif font.

**Daniele Bochicchio**

Lead Software Architect iCubed

Microsoft Regional Director, MVP ASP.NET

Email: [daniele@aspitalia.com](mailto:daniele@aspitalia.com)

Twitter: [@dbochicchio](https://twitter.com/dbochicchio)

# Chi sono

- Lead Software Architect @ iCubed srl
- Microsoft Regional Director for Italy
- Microsoft MVP per ASP.NET dal 2002
- Autore di 15 libri
- Speaker, trainer
- Content manager @ ASPItalia.com Network

# Agenda

- 1) Scenario: Scaling
- 2) Scenario: Adattarsi ai cambiamenti
- 3) Scenario: l'ambiente

# Prima di iniziare

- 1) Useremo Microsoft Azure
- 2) I concetti illustrati vanno bene anche per scenari on-premise\*

\* Ma dovrete fare parte del lavoro a mano

Scenario: Scaling

# Il nostro scenario

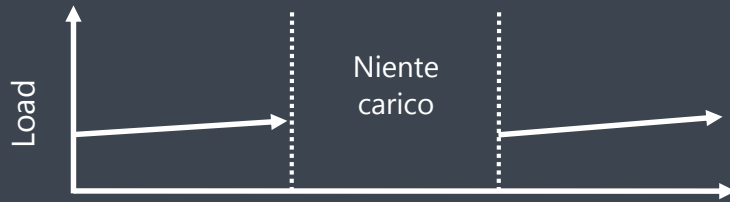
I tempi di caricamento del nostro sito variano di volta in volta.

Dobbiamo adattarci.

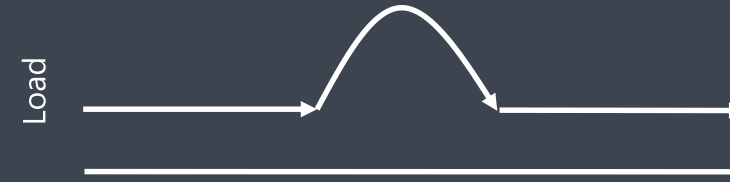
Dolcemente.

# Problematiche di loading

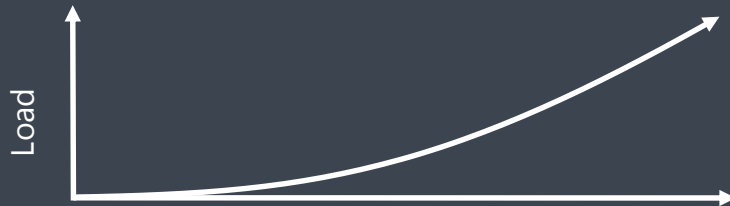
BATCH



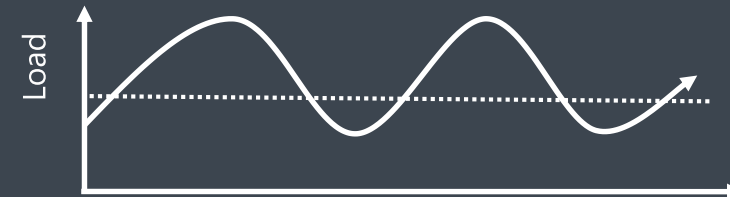
PREDICIBILE



INCREMENTALE



IMPREVISTO



# Step 1: conservare banda

## PROBLEMA:

Quando serviamo file statici sprechiamo banda.



# Step 1: conservare banda

## SOLUZIONE:

Servire i file statici dai blob di Azure.

Questo è il primo passo per servire i file da una content delivery network (CDN).

Usare CDN di terzi per i file delle library (jQuery, etc)

# Step 2: Caching

## PROBLEMA:

I nostri server hanno risorse finite.

È inutile ripetere lavoro che possiamo risparmiare.

# Step 2: Caching

## SOLUZIONE:

Utilizzare la cache di Azure per salvare il contenuto in cache.

Da poco c'è il supporto anche per Redis (preview).



# Step 3: scalare a mano

## PROBLEMA:

Ci servono più risorse lato server.

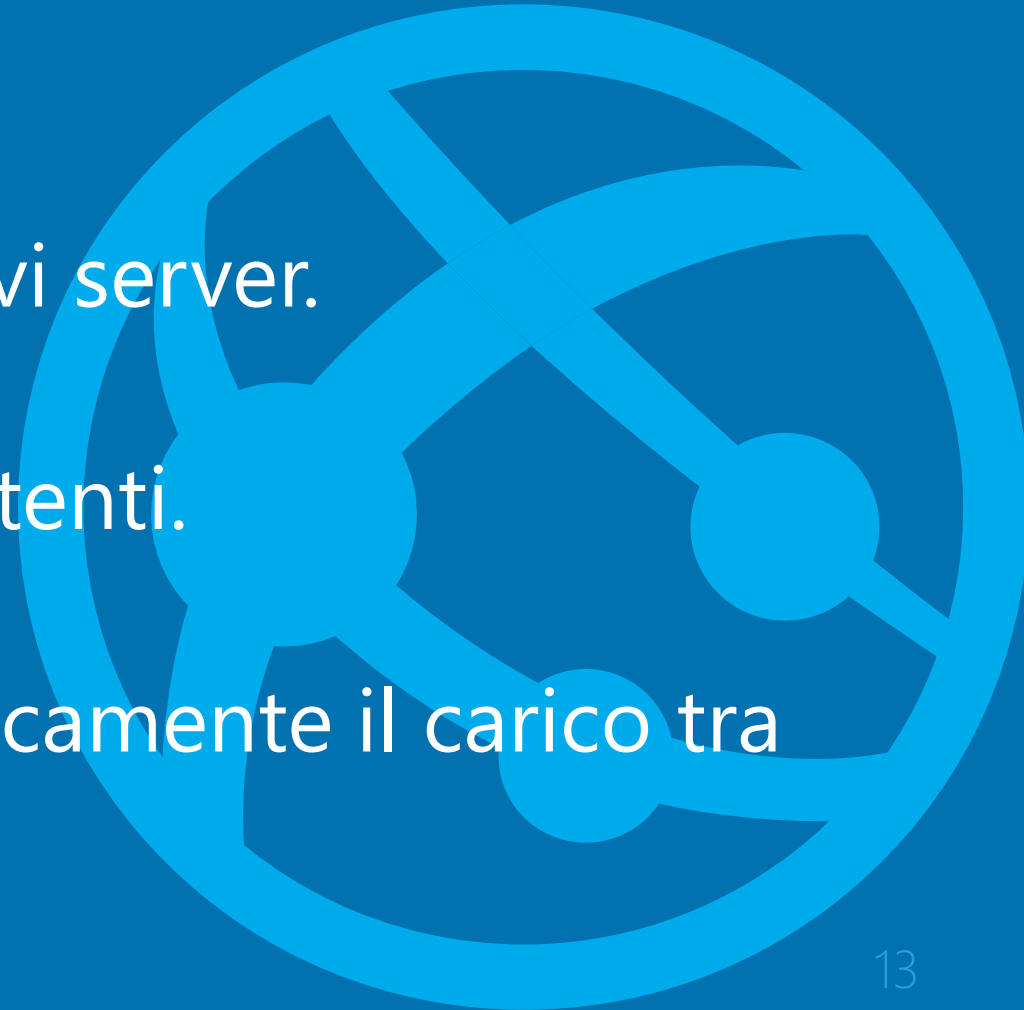
# Step 3: scalare a mano

## SOLUZIONE:

Horizontal scaling: aggiungere nuovi server.

Vertical scaling: usare server più potenti.

Traffic Manager: bilanciare geograficamente il carico tra più Web Site.



## capacity

INSTANCE SIZE

Small (1 core, 1.75 GB Memory )



EDIT SCALE SETTINGS FOR SCHEDULE

No scheduled times

set up schedule times

SCALE BY METRIC

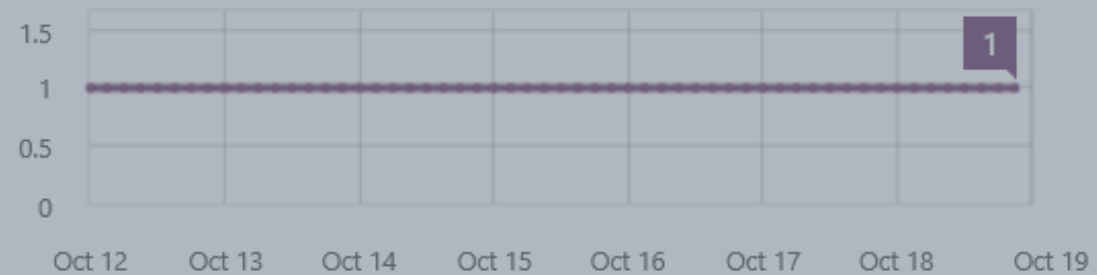
NONE

CPU



INSTANCES

2 MORE



INSTANCE COUNT



1

instances

# Step 4: auto-scale

## PROBLEMA:

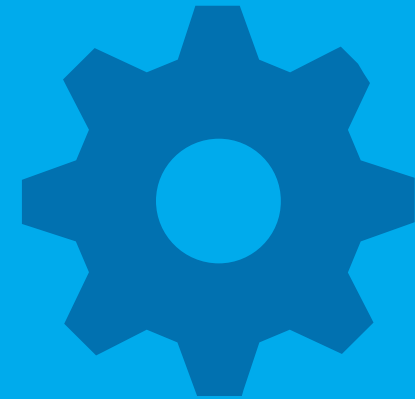
Fare lo scaling a mano implica l'eventualità di avere periodi di malfunzionamento.

È un'operazione delicata e che consuma tempo.

# Step 4: auto-scale

SOLUZIONE:

Usare l'auto scale di Azure.





capacity

INSTANCE SIZE

Small (1 core, 1.75 GB Memory )



EDIT SCALE SETTINGS FOR SCHEDULE

No scheduled times

set up schedule times

SCALE BY METRIC

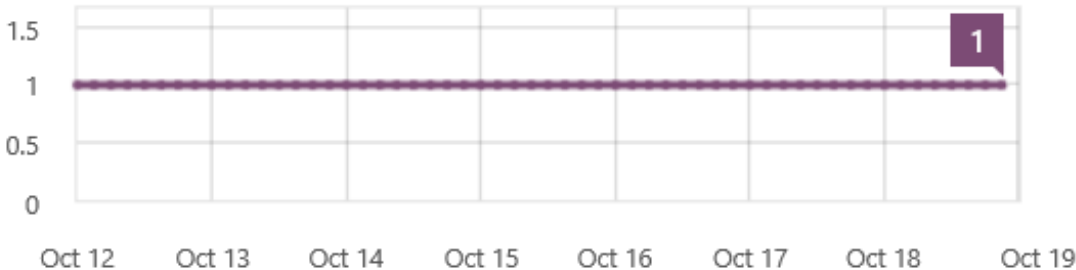
NONE

CPU



INSTANCES

2 MORE



INSTANCE COUNT



1

instances

# Demo

Scenario: Auto-scale di Azure

Scenario: ch-ch-ch-  
change

# Il nostro scenario

Un sito in produzione non è un sito finito.

Ci serve aggiungere nuove feature nel tempo.

Ma ogni modifica introduce rischi.



# Step 1: Gestire lo schema del database

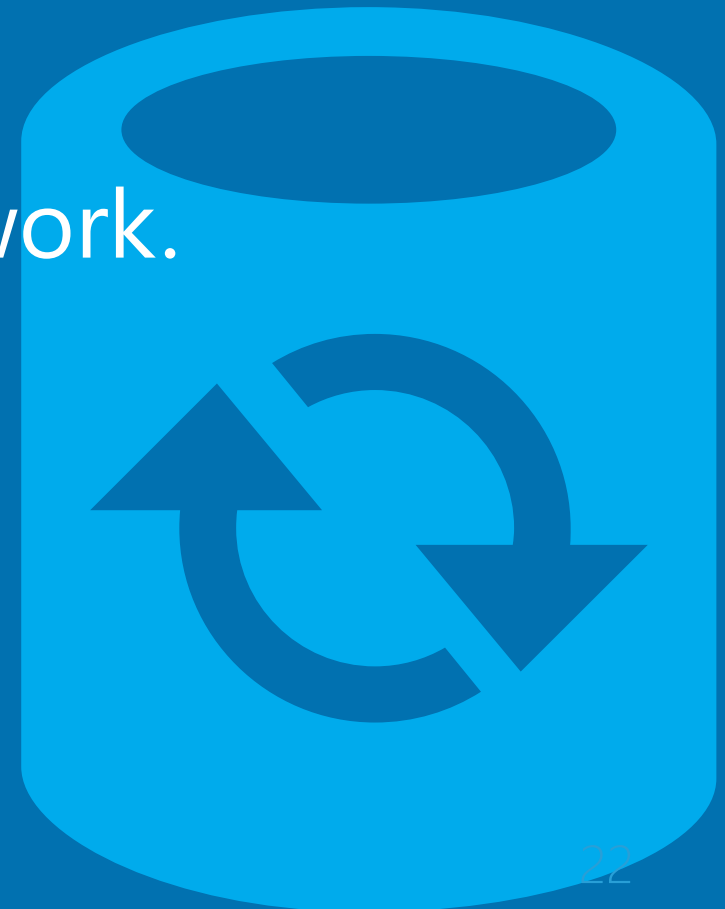
## PROBLEMA:

Il codice dell'applicazione deve essere sempre in sync con il database.

# Step 1: gestire lo schema del database

## SOLUZIONE:

Usare Data Migrations di Entity Framework.



# Demo

Scenario: Data migration di Entity Framework

# Step 2: Gestire i deployment

## PROBLEMA:

Gli errori succedono.  
Bisogna saperli gestire.



# Step 2: Gestire i deployment

SOLUZIONE:

Usare i deployment rollback.




## deployment history

VISUAL STUDIO ONLINE URL

<https://dbplayground.visualstudio.com>



 **ACTIVE DEPLOYMENT:** Tuesday, March 04, 2014 5:40 PM

Individual Continuous Integration - dbtfs\_CD\_20140304.2 - [View Log](#)

ID: 15

AUTHOR: Daniele Bochicchio

DEPLOYED BY: Daniele Bochicchio



Tuesday, March 04, 2014 5:36 PM

Individual Continuous Integration - dbtfs\_CD\_20140304.1 - [View Log](#)

ID: 14

AUTHOR: Daniele Bochicchio

DEPLOYED BY: Daniele Bochicchio



Wednesday, February 05, 2014 4:33 PM

Individual Continuous Integration - dbtfs\_CD\_20140205.2 - [View Log](#)

ID: 13

AUTHOR: Daniele Bochicchio

DEPLOYED BY: Daniele Bochicchio



Wednesday, February 05, 2014 4:30 PM

Individual Continuous Integration - dbtfs\_CD\_20140205.1 - [View Log](#)

ID: 12

AUTHOR: Daniele Bochicchio

DEPLOYED BY: Daniele Bochicchio

Last Refreshed: Monday, May 19, 2014 12:18 PM

# Demo

Scenario: Deployment rollback

Scenario:  
gestire i differenti ambienti

# Il nostro scenario

La gestione di un sito reale prevede la presenza di ambienti interni

Testing, QA, etc

La disponibilità di un sito a livello globale richiede deployment multipli.

# Step 1: ambienti Dev/Test

## PROBLEMA:

Impostare un ambiente con un server di sviluppo consuma tempo.

Ci serve un ambiente in cui far girare e testare il nostro codice che non sia né il sistema di produzione né quello di sviluppo.

# Step 1: ambienti Dev/Test

## SOLUZIONE:

Possiamo replicare un ambiente simile a quello di produzione con i Web Site o le VM di Azure.

Paghiamo solo quello che usiamo.

# 33%

**SCONTO**  
VIRTUAL MACHINE

# 25%

**SCONTO**  
RESERVED WEB SITES  
CLOUD SERVICES, HDINSIGHT

# NO

**CARTA DI CREDITO**

# €150

**CREDITO MENSILE**  
VISUAL STUDIO **ULTIMATE**  
WITH MSDN

# €100

**CREDITO MENSILE**  
VISUAL STUDIO **PREMIUM**  
WITH MSDN

# €50

**CREDITO MENSILE**  
VISUAL STUDIO **PROFESSIONAL**  
WITH MSDN



# Step 2: conquistare il mondo

## PROBLEMA:

È complesso gestire datacenter posti in parti diverse del mondo.

Più il server è vicino ai nostri clienti, migliori sono le performance.

# Step 2: conquistare il mondo

## SOLUZIONE:

Usare workflow automatizzati per gestire la consistenza durante il processo di deployment.

# Step 3: Continuous Delivery

## PROBLEMA:

Se bug e problemi nel nostro codice diventano difficili da gestire, ci mettiamo di più a risolverli.

L'operazione più rischiosa e soggetta ad errori è il deployment.

# Step 3: Continuous Delivery

## SOLUZIONE:

Usiamo la continuous integration per automatizzare build, unit & integration testing.

Possiamo definire dei workflow per assicurarci che ci sia consistenza durante il deployment.

Traffic Manager per avere presenza su data center differente.

# Demo

Scenario: gestire i differenti ambienti

# Grazie!

- Vi ringrazio per il vostro tempo!
- Restiamo in contatto
  - [daniele@aspitalia.com](mailto:daniele@aspitalia.com)
  - @dbochicchio



© 2012 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.