

- ✚ Filippo Bosi – [fbosi@imolinfo.it](mailto:fbosi@imolinfo.it)
- ✚ Mirco Casoni – [mcasoni@imolinfo.it](mailto:mcasoni@imolinfo.it)
- ✚ Maria Seralessandri – [mseralessandri@imolinfo.it](mailto:mseralessandri@imolinfo.it)

Università degli Studi di Bologna - 9 Maggio 2014

## CLOUD COMPUTING E ARCHITETTURE SOFTWARE: QUALI IMPATTI SULLO SVILUPPO DELLE APPLICAZIONI?

# Agenda

- › Cloud Overview
- › Approcci del Cloud Computing
- › Conseguenze dell'utilizzo di Piattaforme Cloud sullo sviluppo di applicazioni
- › Demo – Sessione pratica
- › Conclusioni

## Concetti

- › disponibilità di risorse IT “on demand”
- › provisioning delle risorse
- › affidabilità delle risorse (QoS – “on demand”)
- › virtualizzazione di componenti IT
- › architetture orientate all’alta scalabilità

## Parole Chiave

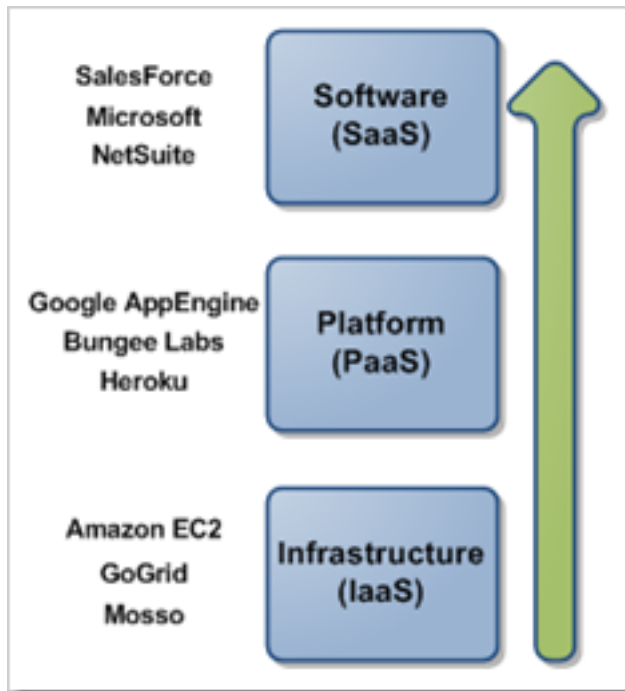
- › On demand
- › Provisioning
- › Reliability
- › Virtualization
- › Scalability

# Cos' è il Cloud Computing?

- › Dal punto di vista dello sviluppo software, il Cloud Computing è costituito da un'infrastruttura di sviluppo e di esecuzione che offre
  - › un'interfaccia utente che nasconde le complessità di gestione a basso livello dell'infrastruttura
  - › riduzione dei costi di amministrazione e dei relativi skill necessari
  - › un insieme di servizi orientati all'amministrazione dell'infrastruttura stessa
  - › alta scalabilità, preconfigurata

# Cloud Computing Continuum

- › Il Cloud può essere visto come un continuum di servizi su differenti livelli, dall'infrastruttura al software



- › Infrastructure: fornitura di hardware su cui generalmente si ha controllo sul Sistema Operativo
- › Platform: ambienti di più alto livello per i quali gli sviluppatori realizzano applicazioni custom
- › Software: software specializzato reso disponibile attraverso internet

# Cloud Computing - Modelli

- › Public cloud
  - › Infrastruttura condivisa tra più clienti (multi-tenant)
  - › Riduzione dei costi tramite economie di scala
  
- › Private cloud
  - › Infrastruttura dedicata a un unico tenant
    - › On-premise private cloud
      - › Hosting su datacenter interno all'azienda
    - › Off-premise private cloud
      - Hosting su datacenter di terze parti
  
- Hybrid cloud
  - Combina i vantaggi di entrambi i modelli

# Un esempio di IaaS – Amazon EC2 (I)

- › Specifiche tecniche
  - › AMI (Amazon Machine Image)
    - › Immagini di macchine virtuali preconfigurate e ready-to-go
  - › Amazon S3 (Simple Storage Service)
    - › Infrastruttura di storage proprietaria di Amazon che funge da repository per lo storage delle AMIs
  - › Amazon EC2 (Elastic Cloud Computing)
    - › Applicazione Web che permette l'amministrazione delle macchine virtuali offerte all'utente finale
- › Schema "Pay per Use" basato su
  - › SLA
  - › Traffico dati
  - › Consumo di CPU
  - › Spazio di Storage utilizzato

# IaaS – Amazon EC2 (II)

## › Passi necessari per l'utilizzo

- › Un account Amazon EC2
- › Connessione internet e un browser
- › Startup dell'AMI
- › Amministrazione da interfaccia utente Amazon

**Sign Up For Amazon EC2**

| AMI ID       | Instance Type | Status  | Key Pair Name | Monitoring | Virtualization | Placement Group |
|--------------|---------------|---------|---------------|------------|----------------|-----------------|
| ami-8679e7f0 | m1.large      | running | flippo        | basic      | paravirtual    |                 |

**1 EC2 Instance selected**

**EC2 Instance: i-8679e7f0**

| Description  | Monitoring  | Tags                        |
|--|---|-----------------------------|
| AMI: <a href="#">RHEL-5.6-OpenShift-Flex-x86_64-13010 [ami-5786b123]</a> | Zone: eu-west-1c  | Type: m1.large              |
| Security Groups: <a href="#">jclouds-funboClouds373545414-eu-west-1</a>  | Status: running   | Owner: 497036449291         |
| VPC ID: -  | Subnet ID: -  | Virtualization: paravirtual |
| Source/Dest. Check: -  | Placement Group: -  | Reservation: r-5ade792c     |
| RAM Disk ID: -   | Key Pair Name: -  | Platform: -                 |
| Key Pair Name: -   | Monitoring: basic   | Kernel ID: aki-428ba136     |
| Elastic IP: -  | Root Device Type: ebs   | AMI Launch Index: 0         |
| Root Device Type: ebs  | Lifecycle: normal   | Root Device: sda            |
| Block Devices: sda   | Public DNS: ec2-46-137-12-232.eu-west-1.compute.amazonaws.com | Tenancy: default            |
| Private DNS: ip-10-58-143-88.eu-west-1.compute.internal                  |   |                             |

© 2008 - 2011, Amazon Web Services LLC or its affiliates. All right reserved. | [Feedback](#) | [Support](#) | [Privacy Policy](#) | [Terms of Use](#) | An [amazon.com](#) company



Terminal



- › Caratteristiche di gestione
  - › Capacità di auto-configurazione dell'infrastruttura hardware in termini di
    - › CPU
    - › RAM
    - › Storage
    - › Network firewall
  - › Shell remota SSH
  - › Disponibilità di template di immagini di macchina (Amazon AMI) con un approccio di condivisione di tipo “Community”
  - › Disponibilità – nell'ambiente – di una serie di macchine preconfigurate da Amazon stessa
  - › Monitoraggio delle risorse in real-time

# PaaS – Google App Engine

- › <https://cloud.google.com/appengine/docs>
- › Sviluppo e esecuzione di applicazioni web sullo stesso sistema che utilizza Google
- › Sviluppo e messa in produzione rapida
- › Amministrazione semplificata
- › Non ci si deve preoccupare dell'hardware, dell'installazione di patch nel sistema operativo o nel middleware, né dei backup
- › Python, Java, Go (<http://golang.org>)
- › Data Store di tipo NoSQL ("BigTable")
- › Scalabilità automatica delle risorse
- › Orientato all'esecuzione di algoritmi specializzati, quali ad esempio il Map-Reduce

- › La piattaforma CloudBees offre la possibilità di effettuare build, test ed esecuzione di applicativi Java EE
- › Offre
  - › Una piattaforma di sviluppo
    - › Maven repository
    - › sistema di versionamento dei sorgenti: SVN, Git
    - › sistema di Continuous Integration preconfigurato
  - › Una piattaforma di esecuzione di applicazioni J2EE e Spring
    - › load balancing
    - › scalabilità
    - › alta affidabilità

# PaaS – RedHat OpenShift (I)

- › La piattaforma OpenShift offre supporto di sviluppo ed esecuzione multi-linguaggio e multi-framework, ed è un PaaS basato principalmente sull'offerta di piattaforme software di RedHat.
  - › Costruito su RHEL (RedHat Enterprise Linux)
  - › Selezione di linguaggi, framework e middleware
    - › Approccio “cartdrige”, che permette allo sviluppatore di selezionare i componenti che interessano
  - › Java, Ruby, PHP, Python, Perl, node.js
  - › MongoDB, JBoss, Jenkins...
  - › Ruby on Rails, Drupal, Wordpress, MySQL
- › E' un esempio di come nell'arco di 6 mesi un'offerta PaaS possa cambiare radicalmente

# PaaS – RedHat OpenShift (II)

## › Architettura

### › Nodi

#### › Gear

- › CPU, RAM, storage e banda di un nodo allocata ad un'applicazione
- › Isolamento e sicurezza tramite Cgroup e SELinux
- › Proxy port per la comunicazione con altre gear

#### › Cartridge built-in

- › Componenti combinabili per formare un'applicazione (database, stack applicativi, tool )

### › Broker

- › Gestisce i nodi, coordina il provisioning, gestisce le applicazioni
- › RHC e la console web console interagiscono con il broker API REST
- › Gli sviluppatori effettuano il push nel codice direttamente nelle gear nei nodi

### › Message bus

- › Mette in comunicazione il broker con i nodi

# Sviluppo di applicazioni su IaaS

- + Il programmatore ha a disposizione un'infrastruttura virtualizzata pronta all'uso
- + Possibilità di ottenere configurazioni avanzate da interfaccia utente
- - Il programmatore deve avere gli skill necessari per gestire i componenti infrastrutturali (ad es. aggiornamenti del sistema operativo, applicazione di patch di sicurezza, ecc.) , del middleware (ad es. il web server, gli application server, ecc.) e in generale dell'ambiente di esecuzione dei propri programmi

# Sviluppo di applicazioni su PaaS

- › + Il programmatore può concentrarsi solo sullo sviluppo
  - › L'infrastruttura è nascosta allo sviluppatore
    - › ...eccetto la possibilità di monitorare le performance e l'addebito dei costi (basato solitamente sull'utilizzo di risorse infrastrutturali)
  - › Lo stack software viene aggiornato automaticamente dal fornitore della piattaforma
- › + Piattaforme software general-purpose complete (J2EE, Microsoft .Net, LAMP Stack...)
- › - Vi sono maggiori limitazioni imposte dalla definizione stessa della piattaforma da parte del provider (sia inizialmente, sia in corso d'opera, ad es. evoluzioni della piattaforma)

# ...verso il “personal” PaaS?

- › L'utilità di avere a disposizione ambienti cloud privati per lo sviluppatore è talmente alta che ultimamente hanno iniziato ad essere disponibili soluzioni che si possono definire “personal” PaaS
- › Da 1 anno e mezzo è disponibile un tool open source che ha rivoluzionato l'approccio alla costruzione degli ambienti PaaS
  - › GIT+Linux Containers+Unionfs+Command Line di Gestione=Docker
  - › <http://www.docker.com/whatisdocker/>
  - › Efficienza nell'esecuzione di ambienti virtualizzati
  - › Versionamento degli ambienti
  - › Estrema facilità d'uso
  - › Marketplace open di ambienti virtualizzati
  - › Docker Hub <http://hub.docker.com>



- › L'uscita di Docker ha avuto impatti notevoli sia sull'implementazione di alcuni PaaS
  - › RedHat ha dichiarato che la prossima versione di OpenShift sarà basata su Docker
  - › Sono uscite distribuzioni Linux ottimizzate per eseguire Docker (<http://www.coreos.com>)
  - › Stanno uscendo implementazioni di Personal PaaS ad una velocità incredibile
    - › <http://www.panamax.io>
    - › Dokku – “Docker powered mini-Heroku in around 100 lines of Bash”
    - › ...

## DEMO

- › Creazione ed utilizzo di un'infrastruttura IaaS su Amazon EC2
- › **Creazione ed esecuzione di una web application su OpenShift (PaaS)**
- › Creazione ed esecuzione di una web application su CloudBees

...collaborativamente ☺

## › Installazione Ruby e Git (Ubuntu)

```
sudo apt-get install ruby-full rubygems git-core
```

## › Verifica installazione

```
ruby -e 'puts "Welcome to Ruby" '  
git -version
```

## › Installazione RHC

```
sudo gem install rhc
```

## › Setup ambiente

```
rhc setup
```

## › Installazione Ruby, Git, TortoiseGit, Putty

```
rubyinstaller  
msysGit-netinstall  
TortoiseGit  
putty
```

## › Verifica installazione

```
ruby -e 'puts "Welcome to Ruby" '  
git -version
```

## › Installazione RHC

```
gem install rhc
```

## › Setup ambiente

```
rhc
```

## › Per usare TortoiseGit e Putty occorre importare le chiavi SSH

# Conclusioni

- › Evoluzione dell'offerta di servizi Cloud
  - › IaaS
    - › Da un ambiente standard di sviluppo e deploy di applicazioni, il programmatore trae beneficio di virtualizzazione avanzata/meccanismi di templating/interfaccia di management
  - › PaaS
    - › Il programmatore trae beneficio da una piattaforma *completamente virtualizzata*
  - › Gli IaaS sono per i system manager, ciò che i PaaS sono per i programmatori/architetti
- › I PaaS sono la “vera” rivoluzione per gli sviluppatori (gli IaaS sono un passo evolutivo della virtualizzazione)
  - › Un programmatore non dovrà più occuparsi dei dettagli infrastrutturali per il funzionamento della piattaforma (e, di conseguenza, della propria applicazione)
  - › Un programmatore si dovrà preoccupare solo di un'attività: lo sviluppo applicativo

DOMANDE?



# Riferimenti

- › Cloud Platform Research Report 2014  
[http://cdn.dzone.com/sites/all/files/whitepapers/DZR\\_CloudPlatformReport\\_Download.pdf](http://cdn.dzone.com/sites/all/files/whitepapers/DZR_CloudPlatformReport_Download.pdf)
  
- › Ebook (gratuiti!) <http://it-ebooks.info/search>
  - › Understanding PaaS - Michael P. McGrath
  - › Cloud Architecture Patterns - Bill Wilder
  - › Cloud Application Architectures - George Reese
  
- › Google App Engine <http://code.google.com/appengine>
- › CloudBees <http://www.cloudbees.com>
- › OpenShift <http://openshift.redhat.com>
  - › <https://www.openshift.com/developers/rhc-client-tools-install>
  - › <https://www.openshift.com/blogs/access-your-application-gear-using-ssh-with-putty>
  - › <http://stackoverflow.com/questions/18684847/cannot-git-clone-an-openshift-application>
  
- › Docker <http://www.docker.io>
  - › <http://hub.docker.com>
  - › <https://github.com/veggie Monk/awesome-docker>



- More than **20 years of experience** on Enterprise IT
- Consulting and Skill Transfer on **Architecture, Integration** and **Process**
- *OMG* Influence Member, *JSR 312* Expert Group, *CSI*, *WWISA*, *OpenESB* Key Partner, *NetBeans* Strategic Partner
- [http://www.imolinfo.it/index\\_en.php](http://www.imolinfo.it/index_en.php)



- **Semantic Web company, now!**
- Applied research, training, projects
- <http://www.sensiblelogic.it>



- The online italian community dedicated to **Java**
- **10 years of** articles, publications, books, events, training
- From programmers to architect
- More than **1.000.000** read pages per month
- <http://www.mokabyte.it>