# Cloud Green Computing

Stefano Piccoli

11 aprile 2022

# Indice

# Parte I

# Cloud Computing

# Capitolo 1

# IaaS (Infrastructure as a Service)

## 1.1 Virtualizzazione

La **virtualizzazione** rende possibile al sistema operativo di un server di eseguire su uno **strato virtuale** (**Hypervisor**).
Questo permette di eseguire molteplici **macchine virtuali**, ognuna con il proprio sistema operativo, sullo stesso server fisico.

## 1.2 Hypervisor

L'**hypervisor** crea lo strato di **virtualizzazione** che rende la virtualizzazione server possibile e contiene la **Virtual Machine Manager (VMM)**.

**Tipologie**

- **Type 1**: caricata direttamente sull'hardware, può eseguire più virtual server, usato per data center o server
    - Hyper-v
    - ESX/ESXi
    - XenServer
- **Type 2**: caricata in un sistema operativo eseguito sull'hardware, greater overhead, usato per desktop e laptop
    - Workstation
    - Virtual Server

- Fusion

## 1.3 Amazon Elastic Compute Cloud 2 (EC2)

- Mette a disposizione server virtuali (**istanze**) in modo semplice, veloce ed economico
- Scelta tipo istanza e template da utilizzare (Windows/Linux) e numero istanze con AWS management console (o librerie SDK)
- **Opzioni di pagamento**: on demand, istanze riservate, istanze spot
- **Sicurezza** (Virtual Private Cloud - VPC
- **Storage persistente**: Amazon Elastic Block Store (EBS)
- **Autoscaling**

## 1.4 Amazon Simple Storage Service (S3)

- Fornisce uno **storage sicuro e facile** da usare
- Diverse **classi di memorizzazione** (standard / standard infrequent access / glacier)
- **Controllo** configurabile di **accesso ai dati**

## 1.5 Amazon Elastic Block Store (EBS)

- Blocco persistente di archiviazione di volumi di memoria usato con le istanze di Amazon EC2
- Ogni volume di Amazon EBS viene automaticamente replicato senza la sua Aviabilty Zone in modo da offrire alta disponibilità e durata.

## 1.6 Dropbox exodus

- I primi 8 anni della sua vita archiviava miliardi di file su Amazon S3
- Tra il 2014 e 2016 ha costruito la propria rete di server ideata dai propri ingegneri per spostare i dati

### 1.6.1 L'esodo

- Hardware propietario che archivierà petabyte di dati
- Nuovo codice ("Magic Pocket")
- Installare 50 rack di hardware al giorno
- Completare lo spostamento prima della scadenza del contratto con Amazon per evitare un rinnovo

### 1.6.2 Conclusioni

Dropbox è riuscita a completare lo spostamento con successo entro i tempi previsiti.

# Capitolo 2

# Container

I **containers** sono un meccanismo di virtualizzazione differente dalle Virtual Machines poichè permetto di avere più istanze **isolate** e **volatili** che scompaiono quando interrotte.
I containers sono **leggeri**, **veloci**, più **semplici da buildare** ma **meno sicuri** delle Virtual Machines.

## 2.1 Docker

**Docker** è un'azienda che ha realizzato una piattaforma che permette di **eseguire una applicazione in ambiente "isolato"**.
Docker sfrutta la **virtualizzazione basata sui container** per eseguire in maniera isolata diverse **GUEST INSTANCES** sullo stesso sistema operativo.

### 2.1.1 Caratteristiche

- **Portabilità**: il software può essere impacchettato in **images**, file read only che può essere mandato in esecuzione da docker e creare quindi il container
- Possono avere più istanze separate degli spazi utente (**containers**)
- **Interfaccia** utente **semplificata**
- **Svantaggio**: sono meno isolati delle macchine virtuali, **condividono le risorse di sistema**

### 2.1.2 Componenti

- **Docker Engine**: permette di creare e mandare in esecuziuone container
- **Docker Hub**: repository enorme che contiene molte immagini di container
- **Docker Swarm Mode**: permette di eseguire un container su più docker host e divide gli swarm node in manager e worker, permettendo una **gestione dichiarativa** della nostra **applicazione**
- **Images**: **template di sola lettura** usati per creare container, registrate in registry
    - **Stratificazione**: ogni strato può essere a sua volta una immagine
- **Registry**: **strutture di repository** che contengono insiemi di immagini per diverse versioni del sw

### 2.1.3 Comandi

- **PULL**: tiro un'immagine dal registry alla macchina
- **RUN**: viene creato il container dell'immagine
- **COMMIT**: salvare una nuova immagine
- **PUSH**: caricare una immagine nel registry
- **BUILD**: si crea un dockerfile che permette di creare un'immagine automaticamente

### 2.1.4 Swarm mode

- I nodi possono agire da **managers**, delegando tasks, o **workers**, eseguendo task assegnati.
- È possibile definire lo **stato dei vari servizi** nello stack dell'applicazione, incluso il numero di **task da eseguire in ogni servizio**
- **Swarm manager**:
    - assegna ad ogni servizio nello swarm un unico DNS name
    - bilancia il carico dei container in esecuzione
    - monitora lo stato del cluster e lo allinea con quello desiderato

# Capitolo 3

# PaaS (Platform as a Service)

Servizio che fornisce hardware e software per lo sviluppo di applicazioni. L'utente deve fornire solo l'applicazione e i dati

**Vantaggi**

- Facilità di gestione e modifica dell'applicazione
- Facilità nell'adottare nuove tecnologie

**Rischi**

- Disponibilità del servizio: l'interruzione del servizio da parte del fornitore comporta un immediato disservizio
- Vendor lock-in: difficoltà di cambiare servizio da parte del cliente

## 3.1 Heroku

**Heroku** è una piattaforma cloud basata su **container** con servizi integrati e un potente ecosistema che permette il deployment e running di applicazioni.

### 3.1.1 Dynos

I **dynos** sono container Linux virtualizzati, Heroku trasforma l'applicazione utente in diversi **dynos**.

**Vantaggi**

- Scalabilità
- Evitare di gestire l'infrastruttura

**Premium:**

- **Scaling**
- **Autoscaling**: permette di inserire politiche per quando usare lo scaling

### 3.1.2 Buildtime

Per sviluppare una applicazione Heroku richiede:

- **Codice sorgente**
- **Lista di dipendenze**
- **Procfile**: file di testo che indica quale comando usare per far eseguire l'applicazione

Slug: Un insieme di codice sorgente, dipendenze, supporto per output, etc...
Stack: Sistema operativo Ubuntu

### 3.1.3 Runtime

Nel **runtime** si prende lo slug e lo stack e vengono creati i dynos, che rappresentano le istanze utente, il dyno manager fa partire i container con il comando specificato dall'utente.

### 3.1.4 Esempio

1. Applicazione riceve richiesta

### 3.1.5 Add-ons

Gli **add-ons** sono funzionalità fornite da Heroku che possono essere aggiunte facilmente all'applicazione.

## 3.2 Altri PaaS

- Microsoft Azure
- OpenShift

# Capitolo 4

# Modelli di Business

Un **business model** descrive il razionale di come una azienda **crea**, **consegna** e **acquisisce valore**.

- **Customer Segments**: il gruppo di persone o organizzazioni a cui il servizio mira di raggiungere
- **Valuer Propositions**: cosa rende speciale il servizio
- **Channels**: le modalità in cui la compagnia raggiunge il cliente
- **Customer Relationships**: tipo di relazione che la compagnia stabilisce col cliente
- **Revenue Streams**: il flusso di entrate che la compagnia genera da ogni segmento di clientela
- **Key Resources**: le risorse più importanti richieste per il modello di business
- **Key Activities**: le attività più importanti che la compagnia deve svolgere
- **Key Partners**: la rete di fornitori e partners per il business
- **Cost Structure**: i costi che si incontrano per operare nel modello di business

## 4.1 Business innovation

- **Resource-driven**: ha origine da **infrastrutture o partner gia esistenti** usate per espandere o trasformare il business model

- **Offer-driven**: crea nuova value proposition che influenza altri ambiti del business model
- **Custmer-driven**: basato sulle necessità del cliente, accesso facilitato o aumento di convenienza
- **Finance-driven**: guidata dal **revenue stream**, meccanismo di prezzi o riduzione dei cost structure

# Parte II

# Green Computing

# Capitolo 5

# Introduzione

**ICT è una minaccia per la sostenibilità ambientale?**

- La produzione di hardware produce inquinamento
- Gas serra
- e-waste

**Green Computing** Pratica ambientale per calcolare la sostenibilità della computazione

- Minimizzare consumo energetico
- Progettare soluzioni efficienti energeticamente
- Riduzione e-waste

## 5.1 Sprechi

- 1,800 kg di materiale grezzo per produrre un personal computer
- ICT contribuisce al 9% di consumo elettrico in Europa
- ICT contribuisce al 4% di emissioni di carbonio in Europa
- È previsto ulteriore aumento nei prossimi anni (esponenzialmente nel networking 20,9%)
- 50 milioni di tonnellate all'anno di e-waste
- Il traffico di e-waste ha un traffico monetario illegale maggiore del traffico di droga

## 5.2 Datacenters

- 40% di consumo energetico per il raffreddamento
- PUE (Power Usage Effectiveness) $\frac{\text{Total Facility Power}}{\text{IT Equipment Power}}$ ma **non** misura la quantità di energia rinnovabile usata

## 5.3 Obsolescenza programmata

Caratteristica di un prodotto volontariamente ideato per avere un "più breve" ciclo di vita. La vita del dispositivo deve durare abbastanza da soddisfare il cliente e fargli desiderare un nuovo dispositivo.

### 5.3.1 Cartello Phoebus

I più grandi produttori di lampadine si riunirono per accordarsi di produrre lampadine che durassero 1000 ore invece di 2500 ore, anche se erano tutti in grado di produrre lampadine più longeve.

### 5.3.2 Obsolescenza percepita

Il cliente è convinto di aver bisogno di un prodotto aggiornato, anche se effettivamente quello attuale è perfettamente funzionante.