

# Dispensa didattica: Modelli di deployment e di fruizione dei servizi cloud

## Argomenti principali:

1. Introduzione al Cloud Computing
  2. Modelli di fruizione (IaaS, PaaS, SaaS)
  3. Modelli di deployment (Public, Private, Hybrid, Multicloud)
  4. Containerizzazione e Serverless
  5. Casi di studio e attività pratica
- 

## 1. Introduzione al Cloud Computing

### 1.1 Definizione e concetti fondamentali

- Il **Cloud Computing** è un modello di erogazione di risorse informatiche (server, storage, servizi applicativi) su richiesta, tramite Internet.
- Le risorse possono essere fornite e rilasciate rapidamente con uno sforzo minimo di gestione o interazione con il provider.
- Si fonda su **Elasticità**, **Scalabilità** e **Pay-as-you-go** (pagamento basato esclusivamente sul consumo).

#### Approfondimento:

- Cloud Computing (Wikipedia, EN) ([https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing)).

### 1.2 Virtualizzazione

- **Virtualizzazione**: tecnologia che separa le risorse hardware dal sistema operativo attraverso un livello software (hypervisor), consentendo di eseguire più macchine virtuali (VM) su un unico server fisico.
- Riduce i costi hardware e semplifica la gestione delle risorse.

#### Approfondimento:

- Virtualization (Wikipedia, EN) (<https://en.wikipedia.org/wiki/Virtualization>).

### 1.3 Motivazioni e vantaggi del cloud

- **Costi ridotti** e passaggio da CAPEX (spese di capitale) a OPEX (spese operative).

- **Scalabilità** automatica e velocità di provisioning di nuove risorse.
  - **Flessibilità** tecnologica: accesso a servizi, librerie e framework avanzati.
  - **Gestione semplificata**: molte attività (manutenzione, aggiornamenti) sono gestite dal provider.
- 

## 2. Modelli di fruizione dei servizi Cloud

I modelli di fruizione definiscono il **livello di astrazione** di cui dispone l'utente rispetto all'infrastruttura sottostante.

### 2.1 Infrastructure as a Service (IaaS)

- **Descrizione**: fornitura di infrastruttura virtualizzata (server, storage, rete) su cui l'utente installa i propri sistemi operativi e applicazioni.
- **Caratteristiche**:
  - Massimo controllo su macchine virtuali, configurazioni di rete e gestione della sicurezza.
  - L'utente è responsabile di mantenimento, patching e configurazioni dei server.
- **Vantaggi**:
  - Grande flessibilità e personalizzazione.
  - Riduzione dei costi di acquisto e gestione dell'hardware fisico.
- **Svantaggi**:
  - Richiede competenze di gestione sistemistica e di networking.
  - Responsabilità elevata in termini di sicurezza (patch, firewall, antivirus).

#### Esempi di provider IaaS:

- Amazon EC2 (<https://aws.amazon.com/ec2/>).
- Microsoft Azure Virtual Machines (<https://azure.microsoft.com/en-us/products/virtual-machines/>).
- Google Compute Engine (<https://cloud.google.com/compute>).

#### Approfondimento:

- Infrastructure as a Service (Wikipedia, EN) ([https://en.wikipedia.org/wiki/Infrastructure\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Infrastructure_as_a_service))
- 

### 2.2 Platform as a Service (PaaS)

- **Descrizione**: fornitura di un'intera piattaforma gestita (runtime, middleware, database) per sviluppare e distribuire applicazioni senza doversi occupare della gestione dell'infrastruttura.
- **Caratteristiche**:
  - Il provider gestisce il sistema operativo, gli aggiornamenti e la sicurezza di base.
  - L'utente si concentra sullo sviluppo dell'applicazione.
- **Vantaggi**:
  - Velocità di sviluppo e semplificazione delle operazioni di deploy.
  - Spesso integrato con servizi aggiuntivi (monitoraggio, bilanciamento del carico).
- **Svantaggi**:
  - Limitata possibilità di configurazione dell'ambiente sottostante.

- Rischio di vendor lock-in a causa di servizi proprietari.

#### Esempi di provider PaaS:

- [Azure App Service](https://azure.microsoft.com/en-us/products/app-service/) (<https://azure.microsoft.com/en-us/products/app-service/>)
- [Heroku](https://www.heroku.com/) (<https://www.heroku.com/>)
- [Google App Engine](https://cloud.google.com/appengine) (<https://cloud.google.com/appengine>)

#### Approfondimento:

- [Platform as a Service \(Wikipedia, EN\)](https://en.wikipedia.org/wiki/Platform_as_a_service) ([https://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Platform_as_a_service))
- 

## 2.3 Software as a Service (SaaS)

- **Descrizione:** fornitura di applicazioni complete, fruibili tramite browser o app senza necessità di installazione locale.
- **Caratteristiche:**
  - Aggiornamenti, manutenzione e patch sono responsabilità del fornitore.
  - Molte soluzioni SaaS seguono modelli di abbonamento mensile/annuale.
- **Vantaggi:**
  - Immediata disponibilità di servizi e rapidità nell'attivazione.
  - Bassa complessità nella gestione interna.
- **Svantaggi:**
  - Minor grado di personalizzazione.
  - Possibile dipendenza dalle politiche del provider.

#### Esempi di servizi SaaS:

- [Microsoft 365](https://www.office.com/) (<https://www.office.com/>)
- [Google Workspace](https://workspace.google.com/) (<https://workspace.google.com/>)
- [Salesforce](https://www.salesforce.com/) (<https://www.salesforce.com/>)

#### Approfondimento:

- [Software as a Service \(Wikipedia, EN\)](https://en.wikipedia.org/wiki/Software_as_a_service) ([https://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Software_as_a_service))
- 

## 2.4 Confronto tra IaaS, PaaS e SaaS

Caratteristica	IaaS	PaaS	SaaS
Gestione dell'infrastruttura	Utente	Provider	Provider
Gestione del software di base	Utente	Provider	Provider
Gestione dell'applicazione	Utente	Utente	Provider
Complessità tecnica	Alta	Media	Bassa
Livello di controllo	Alto	Medio	Basso

---

# 3. Modelli di Deployment del Cloud

I modelli di deployment stabiliscono **dove** e **come** le risorse cloud vengono collocate e gestite.

## 3.1 Public Cloud

- **Definizione:** infrastruttura condivisa e di proprietà di un provider, accessibile su Internet a più clienti.
- **Vantaggi:**
  - Scalabilità rapida, costi variabili e ridotti.
  - Aggiornamenti tecnologici gestiti dal provider.
- **Svantaggi:**
  - Minor controllo su sicurezza, localizzazione e conformità dei dati.
  - Possibili vincoli di latenza e bandwidth.

### Esempi:

- [Amazon Web Services \(https://aws.amazon.com/\)](https://aws.amazon.com/)
  - [Microsoft Azure \(https://azure.microsoft.com/\)](https://azure.microsoft.com/)
  - [Google Cloud Platform \(https://cloud.google.com/\)](https://cloud.google.com/)
- 

## 3.2 Private Cloud

- **Definizione:** infrastruttura cloud dedicata ad una singola organizzazione, installata on-premise o presso un fornitore esterno ma in ambiente esclusivo.
- **Vantaggi:**
  - Massimo controllo e sicurezza su hardware e dati.
  - Maggiore conformità a normative severe (es. settori bancario o sanitario).
- **Svantaggi:**
  - Costi più elevati di acquisto e manutenzione.
  - Necessità di personale interno altamente specializzato.

### Tecnologie di Private Cloud:

- [OpenStack \(https://www.openstack.org/\)](https://www.openstack.org/)
  - [VMware \(https://www.vmware.com/\)](https://www.vmware.com/)
  - [Azure Stack \(https://azure.microsoft.com/en-us/products/azure-stack/\)](https://azure.microsoft.com/en-us/products/azure-stack/)
- 

## 3.3 Hybrid Cloud

- **Definizione:** combinazione di servizi cloud pubblici e privati che cooperano per offrire un'unica infrastruttura ibrida.
- **Vantaggi:**
  - Possibilità di tenere dati sensibili in un ambiente privato e sfruttare la scalabilità del pubblico.
  - Flessibilità e costi ottimizzati a seconda delle esigenze.

- **Svantaggi:**
  - Architetture più complesse.
  - Integrazione e sicurezza richiedono competenze elevate.

**Approfondimento:**

- [Hybrid Cloud \(Wikipedia, EN\)](https://en.wikipedia.org/wiki/Hybrid_cloud) ([https://en.wikipedia.org/wiki/Hybrid\\_cloud](https://en.wikipedia.org/wiki/Hybrid_cloud))
- 

## 3.4 Multicloud

- **Definizione:** utilizzo di più cloud pubblici di fornitori diversi (ad esempio AWS, Azure, GCP), gestiti contemporaneamente.
- **Vantaggi:**
  - Riduzione del rischio di vendor lock-in.
  - Maggiore resilienza: se un provider presenta disservizi, si può sfruttare un altro.
- **Svantaggi:**
  - Aumento della complessità gestionale (strumenti di monitoring, networking, sicurezza).
- **Esempi reali:**
  - Netflix, che sfrutta principalmente AWS ma utilizza anche Google Cloud per specifici servizi.

**Approfondimento:**

- [Multicloud \(Wikipedia, EN\)](https://en.wikipedia.org/wiki/Multicloud) (<https://en.wikipedia.org/wiki/Multicloud>)
  - [Vendor lock-in \(Wikipedia, EN\)](https://en.wikipedia.org/wiki/Vendor_lock-in) ([https://en.wikipedia.org/wiki/Vendor\\_lock-in](https://en.wikipedia.org/wiki/Vendor_lock-in))
- 

# 4. Tecnologie avanzate di fruizione del Cloud

## 4.1 Containerizzazione

- **Docker:**
  - Sistema di containerizzazione che consente di impacchettare applicazioni e relative dipendenze in un'unica immagine leggera.
  - Facilita la portabilità tra diversi ambienti (sviluppo, test, produzione).
  - [Docker \(Wikipedia, EN\)](https://en.wikipedia.org/wiki/Docker_(software)) ([https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))).
- **Kubernetes:**
  - Piattaforma di orchestrazione dei container.
  - Automatizza il deployment, lo scaling e la gestione di applicazioni containerizzate.
  - [Kubernetes \(Wikipedia, EN\)](https://en.wikipedia.org/wiki/Kubernetes) (<https://en.wikipedia.org/wiki/Kubernetes>)

## 4.2 Serverless Computing

- **Definizione:** modello di esecuzione del codice in cui il provider gestisce automaticamente l'infrastruttura, consentendo all'utente di concentrarsi unicamente sulle funzioni da eseguire.
  - **Vantaggi:**
    - Costi correlati esclusivamente all'esecuzione del codice (nessun server fisso sempre attivo).
    - Scalabilità automatica, gestione semplificata.
  - **Esempi di servizi serverless:**
    - [AWS Lambda \(https://aws.amazon.com/lambda/\)](https://aws.amazon.com/lambda/).
    - [Azure Functions \(https://azure.microsoft.com/en-us/products/functions/\)](https://azure.microsoft.com/en-us/products/functions/).
    - [Google Cloud Functions \(https://cloud.google.com/functions\)](https://cloud.google.com/functions).
- 

## 5. Caso di studio: Migrazione aziendale al Cloud

### 5.1 Contesto

- Una PMI con un data center on-premise, applicazioni legacy e budget limitato vuole migliorare scalabilità e ridurre i costi operativi.

### 5.2 Strategie di migrazione

1. **Re-host (Lift & Shift):** migrare i server in un ambiente IaaS (es. AWS EC2).
2. **Refactor:** modificare le applicazioni per sfruttare PaaS e container.
3. **Re-platform:** riscrivere parzialmente l'applicazione per modelli serverless.

### 5.3 Scelta del modello di deployment

- **Pubblico** se i requisiti di sicurezza e controllo sono meno stringenti.
- **Privato** se si gestiscono dati estremamente sensibili e normative severe.
- **Ibrido** per dividere carichi di lavoro tra risorse on-premise e cloud pubblico.
- **Multicloud** se si vuole evitare la dipendenza da un singolo provider e massimizzare la resilienza.

#### Best practices:

- Analisi dei costi TCO (Total Cost of Ownership).
  - Pianificazione e test di compatibilità delle applicazioni.
  - Formazione del personale e gestione del cambiamento.
- 

## 6. Attività pratica e conclusioni

### 6.2 Conclusioni

- Il Cloud Computing offre molteplici vantaggi (costo, scalabilità, flessibilità), ma occorre valutare attentamente requisiti e vincoli di sicurezza, budget e competenze.
  - Conoscere i **diversi modelli di fruizione** (IaaS, PaaS, SaaS) e di **deployment** (Public, Private, Hybrid, Multicloud) è cruciale per operare scelte informate.
  - Le **tecnologie container** e i modelli **serverless** rappresentano strumenti potenti per implementare applicazioni moderne, resilienti e scalabili.
- 

# Risorse di approfondimento (in Inglese)

- **Cloud Computing**
    - [Cloud Computing \(Wikipedia\)](https://en.wikipedia.org/wiki/Cloud_computing) ([https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing))
    - [AWS Documentation](https://docs.aws.amazon.com/) (<https://docs.aws.amazon.com/>)
    - [Azure Documentation](https://learn.microsoft.com/en-us/azure/) (<https://learn.microsoft.com/en-us/azure/>)
    - [Google Cloud Documentation](https://cloud.google.com/docs) (<https://cloud.google.com/docs>)
  - **IaaS, PaaS, SaaS**
    - [Infrastructure as a Service \(Wikipedia\)](https://en.wikipedia.org/wiki/Infrastructure_as_a_service) ([https://en.wikipedia.org/wiki/Infrastructure\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Infrastructure_as_a_service))
    - [Platform as a Service \(Wikipedia\)](https://en.wikipedia.org/wiki/Platform_as_a_service) ([https://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Platform_as_a_service))
    - [Software as a Service \(Wikipedia\)](https://en.wikipedia.org/wiki/Software_as_a_service) ([https://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Software_as_a_service))
  - **Deployment Models**
    - [Hybrid Cloud \(Wikipedia\)](https://en.wikipedia.org/wiki/Hybrid_cloud) ([https://en.wikipedia.org/wiki/Hybrid\\_cloud](https://en.wikipedia.org/wiki/Hybrid_cloud))
    - [Multicloud \(Wikipedia\)](https://en.wikipedia.org/wiki/Multicloud) (<https://en.wikipedia.org/wiki/Multicloud>)
  - **Container e Serverless**
    - [Docker \(Wikipedia\)](https://en.wikipedia.org/wiki/Docker_(software)) ([https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)))
    - [Kubernetes \(Wikipedia\)](https://en.wikipedia.org/wiki/Kubernetes) (<https://en.wikipedia.org/wiki/Kubernetes>)
    - [Serverless Computing \(Wikipedia\)](https://en.wikipedia.org/wiki/Serverless_computing) ([https://en.wikipedia.org/wiki/Serverless\\_computing](https://en.wikipedia.org/wiki/Serverless_computing))
-