

# ICT INFORMATION AND COMMUNICATIONS TECHNOLOGY ACADEMY

---

MODULO: Progettazione  
UNITÀ: Progettazione.2

Ing. Roberto Delisio



Le basi del cloud  
computing

## Cos'è il cloud?

Il Cloud è un'astrazione di risorse di elaborazione, archiviazione e networking

## Cos'è il cloud computing?

Il cloud computing è un modello di erogazione di risorse informatiche (come server, spazio di archiviazione, database, rete, software, ecc.) tramite Internet, cioè "dal cloud", invece che in locale su un computer o un server fisico di tua proprietà.

In parole semplici:

Immagina di poter usare un computer potentissimo, ma che non è fisicamente sulla tua scrivania. È ospitato da qualcun altro (come Amazon, Microsoft, Google) e tu lo usi tramite Internet. Paghi solo per quanto lo usi, un po' come succede con la corrente elettrica.

## Data Center Astratto

Non ci interessa più su quale server fisico gira il nostro carico di lavoro. Inviamo semplicemente il nostro carico di lavoro al cloud e lasciamo che il provider si occupi di tutto il resto.



## I Tre Pilastri del Cloud Computing

Il cloud computing si basa su tre componenti fondamentali che replicano virtualmente tutto ciò che serve per far funzionare le applicazioni moderne. Questi pilastri rappresentano l'astrazione digitale di un data center tradizionale.

### Compute (Potenza di Calcolo Virtuale)

Il Compute rappresenta la capacità di elaborazione, ovvero CPU e memoria RAM virtualizzate. Nel cloud ottieni server virtuali on-demand senza dover acquistare hardware fisico. Puoi scalare la potenza di calcolo in base alle necessità: da un semplice sito web che richiede poca potenza, fino a complessi algoritmi di machine learning che necessitano di centinaia di core. In AWS, il servizio principale è EC2 (Elastic Compute Cloud). A cosa può servire nel concreto? A far girare il codice del tuo sito web o della tua applicazione.

### Storage (Spazio di Archiviazione)

Lo Storage è lo spazio dove conservi tutti i tuoi dati: file, database, backup, immagini, video. Nel cloud hai accesso a spazio di archiviazione praticamente illimitato, pagando solo per quello che usi. Esistono diversi tipi: storage veloce per database attivi, storage economico per archivi a lungo termine, storage distribuito per alta disponibilità. In AWS, il servizio principale è S3 (Simple Storage Service). A cosa può servire nel concreto? A salvare le foto caricate dagli utenti o i file dei tuoi prodotti.

### Network (Connettività e Rete)

Il Network gestisce tutta la comunicazione: come i tuoi server si collegano tra loro, come gli utenti accedono alle tue applicazioni, come i dati viaggiano in sicurezza. Include load balancer per distribuire il carico, firewall per la sicurezza, CDN per velocizzare i contenuti globalmente. Nel cloud puoi creare reti virtuali private, configurare routing personalizzato e gestire il traffico in modo intelligente. In AWS, il servizio principale è VPC (Virtual Private Cloud). A cosa può servire nel concreto? A proteggere il tuo sito da attacchi e a gestire chi può visitarlo.

## La Virtualizzazione: Il DNA del Cloud Computing

La **virtualizzazione** è letteralmente il motore che ha reso possibile l'esistenza del cloud moderno. Senza virtualizzazione, non avremmo AWS, Google Cloud o Azure come li conosciamo oggi.

### Cosa Significa "Virtualizzazione"?

La virtualizzazione è la tecnologia che permette di **dividere un singolo computer fisico in molteplici computer virtuali** indipendenti. È come avere un palazzo con tanti appartamenti: un solo edificio fisico, ma ogni inquilino ha la sua casa completamente separata e funzionale.

#### Prima della virtualizzazione:

- 1 server fisico = 1 sistema operativo = 1 applicazione
- Spreco enorme di risorse (server utilizzati al 10-15%)
- Costi altissimi per hardware dedicato

#### Con la virtualizzazione:

- 1 server fisico = 10-50+ macchine virtuali
- Utilizzo ottimale delle risorse (80-90%)
- Condivisione intelligente di CPU, RAM e storage
- VM possono spostarsi tra host automaticamente
- Non ci interessa su quale host fisico gira la VM



## Perché è il Fondamento del Cloud?

### 1. Astrazione dell'Hardware

Hardware Fisico → Hypervisor → Macchine Virtuali → Le Tue Applicazioni

Non ti interessa più su quale server fisico gira la tua applicazione. L'hypervisor (il "cervello" della virtualizzazione) gestisce tutto automaticamente. Il suo ruolo fondamentale è quello di creare e gestire le macchine virtuali (VM), consentendo a più sistemi operativi di funzionare su un unico server fisico.

In pratica, l'hypervisor si occupa di:

- **Astrazione dell'hardware:** "Nasconde" le reali risorse fisiche (CPU, memoria, storage) e le presenta alle macchine virtuali come se fossero risorse dedicate.
- **Allocazione delle risorse:** Distribuisce le risorse fisiche tra le varie macchine virtuali in base alle loro necessità, garantendo efficienza e flessibilità.
- **Isolamento:** Mantiene le macchine virtuali separate e indipendenti l'una dall'altra. In questo modo, un problema su una VM non influisce sulle altre.

# Cloud Computing

## Perché è il Fondamento del Cloud?

### 2. Elasticità e Scalabilità

- **Scaling Up:** Aggiungi CPU/RAM a una VM in tempo reale
- **Scaling Out:** Crei nuove VM in secondi, non settimane
- **Auto-scaling:** Le VM si creano/distruggono automaticamente in base al carico

Questi concetti trasformano il modo in cui le aziende gestiscono le risorse IT, passando da un modello statico e lento a uno dinamico e istantaneo.

#### A. Scaling Up (Scalabilità Verticale): Potenzia la tua macchina

Lo **Scaling Up** significa aumentare la potenza di una singola macchina virtuale (VM) aggiungendo più risorse.

- **Come funziona:** Grazie all'hypervisor, che gestisce il pool di risorse fisiche del server (CPU, RAM, storage), è possibile riallocare queste risorse dinamicamente. Se la tua applicazione su una VM ha bisogno di più potenza di calcolo, puoi entrare nel pannello di controllo del tuo cloud provider e, con pochi click, aumentare i core della CPU o i GB di RAM assegnati a quella VM.
- **Il vantaggio "in tempo reale":** La parte rivoluzionaria è che questa operazione richiede spesso solo un riavvio della VM (a volte neanche quello, a seconda della tecnologia) e avviene in pochi minuti. Non c'è bisogno di acquistare nuovo hardware, spegnere il server, aprire il case, installare fisicamente la nuova RAM o CPU e riavviare tutto. L'hypervisor si occupa di "scollegare" e "ricollegare" le risorse virtuali in modo trasparente.

**In sintesi:** Lo Scaling Up ti dà la flessibilità di "gonfiare" una macchina esistente quando prevedi un aumento del carico, come lanciare una campagna di marketing o elaborare un grande set di dati.

Perché è il Fondamento del Cloud?

## B. Scaling Out (Scalabilità Orizzontale): Aggiungi nuove macchine

Lo **Scaling Out** consiste nell'aumentare la capacità del tuo sistema aggiungendo altre macchine virtuali che lavorano in parallelo.

- **Come funziona:** Invece di potenziare una singola macchina, ne crei di nuove, identiche alla prima. L'hypervisor utilizza delle "immagini" o "template" (modelli pre-configurati della tua VM) per creare cloni perfetti in pochi secondi o minuti. Un sistema chiamato **load balancer** (bilanciatore di carico) si occupa poi di distribuire il traffico in entrata tra tutte le VM disponibili, evitando che una singola macchina venga sovraccaricata.
- **Il vantaggio "secondi, non settimane":** Questo è il vero punto di svolta rispetto all'IT tradizionale. Per aggiungere un nuovo server fisico, il processo richiedeva settimane o mesi: approvazione del budget, ordine dell'hardware, attesa della consegna, installazione fisica nel data center, cablaggio, installazione del sistema operativo e delle applicazioni. Con il cloud, l'hypervisor può replicare una VM quasi istantaneamente.

**In sintesi:** Lo Scaling Out è ideale per gestire aumenti di traffico imprevedibili e per costruire architetture ad alta affidabilità. Se una VM ha un problema, le altre continuano a funzionare e il servizio non si interrompe.

## C. Auto-scaling: Il pilota automatico per le tue risorse

L'**Auto-scaling** è la combinazione intelligente dei due concetti precedenti, gestita in modo completamente automatico.

- **Come funziona:** Imposti delle regole basate su metriche di performance. Ad esempio:
  - **Regola di Scaling Out:** "Se l'utilizzo medio della CPU di tutte le mie VM supera l'80% per più di 5 minuti, crea una nuova VM".
  - **Regola di Scaling In:** "Se l'utilizzo medio della CPU scende sotto il 20% per più di 10 minuti, distruggi una VM".
- **Il vantaggio:** L'auto-scaling offre il meglio di entrambi i mondi:
  - **Performance:** Durante un picco di traffico improvviso (ad esempio, il tuo sito viene menzionato in TV), il sistema crea automaticamente nuove VM per gestire il carico, garantendo che il servizio rimanga veloce e reattivo per tutti gli utenti.
  - **Ottimizzazione dei costi:** Non appena il picco di traffico finisce, il sistema distrugge le VM non necessarie. In questo modo, paghi solo per le risorse che usi effettivamente, evitando sprechi di denaro per macchine inutilizzate durante i periodi di calma.

**In sintesi:** L'auto-scaling è il cuore del modello "pay-as-you-go" del cloud. Ti assicura di avere sempre la potenza necessaria senza pagare un centesimo in più del dovuto, garantendo resilienza e efficienza economica.

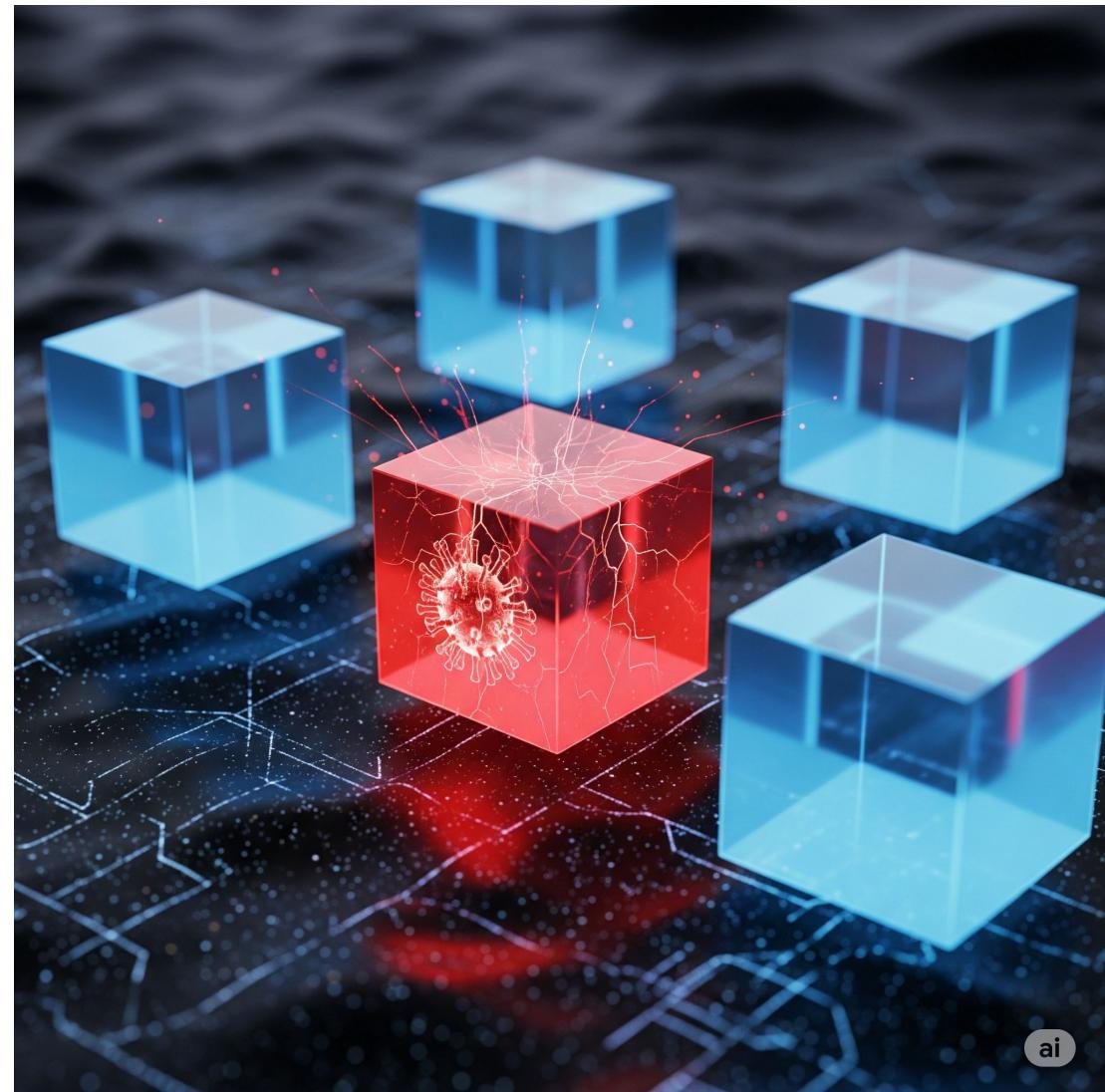
## Perché è il Fondamento del Cloud?

### 3. Isolamento e Sicurezza

Ogni VM è completamente isolata dalle altre. Se una VM si blocca o viene compromessa, le altre continuano a funzionare normalmente.

Pensa all'hypervisor come all'architetto di una nave a compartimenti stagni. Ogni macchina virtuale (VM) è una di queste stanze, sigillata e completamente separata dalle altre, anche se tutte fanno parte della stessa nave (l'hardware fisico).

- **Risorse Virtuali dedicate:** L'hypervisor crea per ogni VM un set completo di **risorse virtuali**: una CPU virtuale, una RAM virtuale, una scheda di rete virtuale e un disco virtuale. La VM crede di avere a disposizione hardware tutto suo e non ha alcuna consapevolezza delle altre VM che girano sullo stesso server.
- **Il Guardiano del Traffico:** L'hypervisor agisce come un controllore del traffico estremamente rigido. Ogni singola richiesta che una VM fa all'hardware fisico (ad esempio, "scrivi questo dato in memoria" o "invia questo pacchetto di rete") deve passare attraverso l'hypervisor. Questo controllore si assicura che la VM acceda **solo ed esclusivamente** allo spazio di memoria e alle risorse che le sono state assegnate, impedendole di "sbirciare" o interferire con le altre.



## Perché è il Fondamento del Cloud?

Questo meccanismo di separazione forzata porta a due vantaggi pratici enormi:

### A. Resistenza ai Guasti (Fault Tolerance)

Se una VM va in crash, il problema rimane confinato.

- **Scenario:** Immagina di avere tre VM sullo stesso server fisico.
  - VM 1: Esegue un'applicazione web.
  - VM 2: Gestisce un database critico.
  - VM 3: Fa girare un ambiente di test.
- **Cosa succede:** Un bug software nell'applicazione sulla VM 1 causa un "kernel panic" o esaurisce tutta la sua RAM virtuale, mandandola in crash.
- **Risultato:** Solo la VM 1 smette di funzionare. L'hypervisor si assicura che il suo "crollo" non consumi le risorse fisiche destinate alle altre. La VM 2 (con il database) e la VM 3 (con l'ambiente di test) **continuano a funzionare normalmente**, ignare del problema accaduto alla loro vicina. Il "raggio d'esplosione" del guasto è limitato alla singola stanza stagna.

### B. Sicurezza e Contenimento delle Minacce (Security)

L'isolamento è una barriera di sicurezza fondamentale, specialmente in un ambiente multi-tenant come il cloud, dove le tue VM potrebbero girare sullo stesso hardware di quelle di un'altra azienda.

- **Scenario:** Un hacker riesce a sfruttare una vulnerabilità nell'applicazione web sulla VM 1 e ottiene l'accesso al suo sistema operativo.
- **Risultato:** L'attaccante è, in pratica, **intrappolato all'interno della VM 1**. Grazie all'isolamento imposto dall'hypervisor, non può:
  - Accedere direttamente alla memoria della VM 2 per rubare i dati del database.
  - Intercettare il traffico di rete delle altre VM.
  - Accedere al file system del server fisico sottostante.

Questa "sandbox" (scatola di sabbia) virtuale rende l'infrastruttura cloud intrinsecamente più sicura. Anche se una singola applicazione viene compromessa, il danno è contenuto e non si propaga orizzontalmente al resto dell'infrastruttura, proteggendo gli altri carichi di lavoro.

## Perché è il Fondamento del Cloud?

### 4. Migrazione Live

Le VM possono spostarsi da un server fisico all'altro **senza interruzioni**. Se un server ha problemi hardware, le VM migrano automaticamente su server sani.

#### Come funziona la "Magia" della Migrazione Live

Spostare una macchina virtuale **accesa e funzionante** da un server fisico a un altro senza che gli utenti se ne accorgano sembra impossibile, ma l'hypervisor lo rende realtà attraverso un processo orchestrato in millisecondi:

1. **Copia della Memoria:** L'hypervisor sul server di partenza (Server A) inizia a copiare l'intero contenuto della RAM della VM verso il server di destinazione (Server B) attraverso la rete. Durante questa fase, la VM continua a funzionare normalmente sul Server A.
2. **Sincronizzazione delle Modifiche:** Mentre la copia è in corso, l'applicazione all'interno della VM continua a scrivere e modificare dati in RAM. L'hypervisor tiene traccia di queste piccole modifiche e le invia in un secondo momento al Server B.
3. **Il Momento del Passaggio:** Una volta che la RAM sul Server B è quasi identica a quella sul Server A, l'hypervisor mette in pausa la VM sul Server A per una frazione di secondo (pochi millisecondi), copia le ultimissime modifiche alla memoria, e attiva istantaneamente la VM sul Server B.
4. **Aggiornamento della Rete:** Immediatamente, l'hypervisor comunica alla rete che l'indirizzo della VM ora punta al Server B.

L'intero processo è così rapido che un utente che sta usando un'applicazione o guardando un video in streaming da quella VM non noterà alcuna interruzione, al massimo un impercettibile ritardo di un singolo pacchetto di rete.

## Perché è il Fondamento del Cloud?

### I Vantaggi Pratici

Questa tecnologia abilita due scenari fondamentali per l'affidabilità e la flessibilità del cloud.

#### Manutenzione a Zero Downtime

Questo è il caso d'uso più comune. Immagina che il Server A abbia bisogno di un aggiornamento hardware (più RAM, un update del firmware) o software (una patch di sicurezza per l'hypervisor).

- **Senza Migrazione Live:** Dovresti spegnere tutte le VM, spegnere il server, fare la manutenzione, riavviare tutto. Questo significa ore di downtime pianificato per le tue applicazioni.
- **Con la Migrazione Live:** Semplicemente, sposti a caldo tutte le VM dal Server A al Server B. Fai la tua manutenzione sul Server A in tutta tranquillità. Una volta finito, se vuoi, puoi spostare di nuovo le VM su A. **Risultato: zero interruzioni del servizio.**

#### Tolleranza ai Guasti Automatica

Questo è lo scenario che hai descritto e che garantisce l'alta affidabilità (**High Availability**). I sistemi cloud moderni monitorano costantemente la salute dei server fisici.

- **Scenario:** Il sistema rileva che un componente del Server A sta per guastarsi (ad esempio, una ventola si rompe e la temperatura della CPU sta salendo pericolosamente).
- **Azione Automatica:** Invece di aspettare che il server si spenga causando un'interruzione, il sistema di gestione del cloud (l'orchestratore) avvia **automaticamente la migrazione live** di tutte le VM critiche dal Server A a uno o più server sani del cluster.

Il problema hardware viene risolto **prima che causi un downtime**. Le tue applicazioni continuano a funzionare come se nulla fosse, mentre il team tecnico può intervenire per riparare il server guasto. Questo trasforma la gestione dell'infrastruttura da reattiva (riparo ciò che si rompe) a proattiva (prevengo le interruzioni).

## Data Center Astratto

Non ci interessa più su quale server fisico gira il nostro carico di lavoro. Inviamo semplicemente il nostro carico di lavoro al cloud e lasciamo che il provider si occupi di tutto il resto.

## I Benefici dell'Astrazione nel Cloud Computing

L'**Astrazione** è il superpotere del cloud che nasconde tutta la complessità dell'infrastruttura fisica dietro interfacce semplici e intuitive.

L'idea di base è semplice ma potentissima: **l'astrazione** prende un sistema incredibilmente complesso e lo presenta attraverso un'interfaccia semplice, nascondendo tutti i dettagli interni.

### Cos'è l'Astrazione?

È come guidare un'automobile:

- **NON hai bisogno di sapere** come funziona il motore a combustione, la trasmissione o l'elettronica.
- **HAI bisogno di sapere** come usare un volante, dei pedali e una leva del cambio.

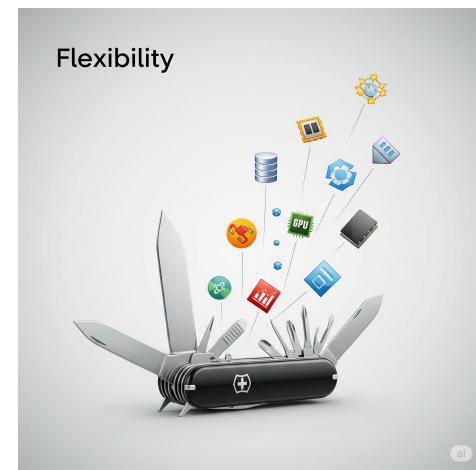
L'astrazione nasconde la complessità interna dietro un'interfaccia semplice.

Il Cloud fa la stessa cosa con l'infrastruttura IT.

**Due caratteristiche fondamentali sono:**

- **Il Pannello di Controllo Unificato**
- **Scalability (Scalabilità)**
- **Flexibility (Flessibilità)**

Ora vedremo in dettaglio i 3 superpoteri che l'astrazione ci regala, trasformando il modo in cui lavoriamo." Questo crea attesa e struttura la narrazione.



# Cloud Computing

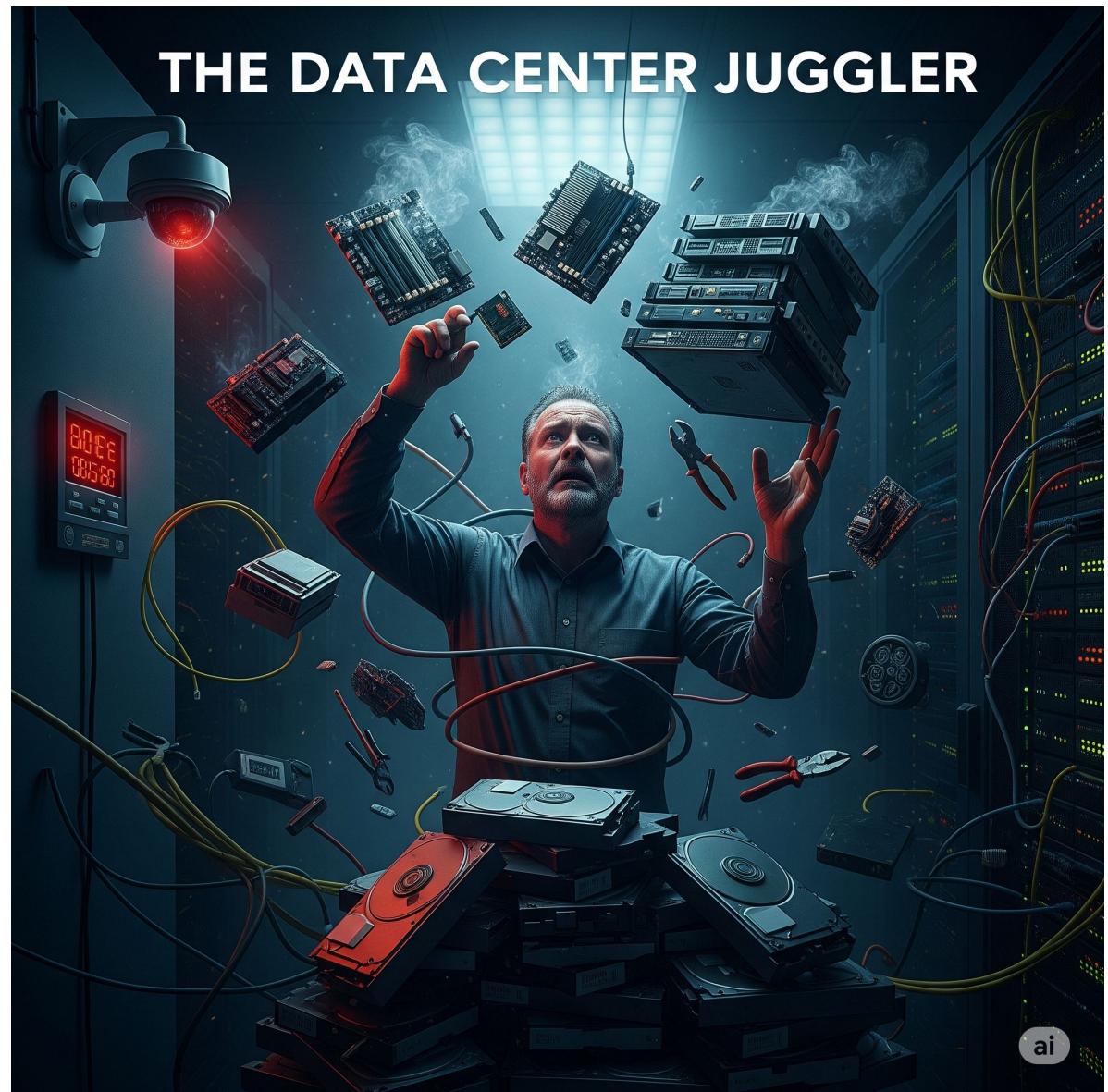
## 1. Simplified Management (Gestione Semplificata)

### Il Problema del "Prima": Il Fardello del Mondo Fisico

Gestire un data center fisico non è un singolo lavoro, è un insieme di almeno cinque professioni diverse, ognuna con le sue complessità e i suoi rischi.

- **Esperto Hardware:** Non si tratta solo di scegliere CPU e RAM. Devi studiare le matrici di compatibilità, assicurarti che la scheda madre supporti quella specifica RAM, che l'alimentatore sia sufficiente per tutti i componenti sotto carico massimo e che tutto entri fisicamente nel rack.
- **Esperto di Networking:** Devi progettare la topologia di rete, configurare VLAN per isolare il traffico, scrivere regole complesse su firewall fisici, gestire switch e router di diversi produttori e, fisicamente, passare e crimpate centinaia di cavi. Un cavo difettoso può richiedere ore di troubleshooting.
- **Esperto di Storage:** Devi decidere quale livello RAID usare (RAID 5? RAID 10?), ognuno con i suoi compromessi tra performance, ridondanza e costo. Devi gestire complesse e costose Storage Area Network (SAN), e pianificare backup su nastri o dischi esterni, un processo manuale e prone a errori.
- **Tecnico di Manutenzione:** Questo è il lavoro sporco. Devi pianificare downtime notturni o nel weekend, entrare fisicamente nel data center, spegnere un server, sostituire un disco o un banco di RAM, e poi incrociare le dita durante il riavvio, sperando che un aggiornamento firmware non "bricki" (rompa irrimediabilmente) la macchina.
- **Responsabile della Sicurezza Fisica:** Devi gestire l'accesso alla sala server con badge o dati biometrici, monitorare la temperatura e l'umidità per evitare surriscaldamenti, e assicurarti che i sistemi di alimentazione di emergenza (UPS, generatori) funzionino in caso di blackout.

Il risultato era un team IT che, per necessità, doveva essere un "tuttofare" dell'hardware, dedicando la maggior parte delle sue energie a "tenere le luci accese"



## La Soluzione dell'Astrazione: Il Pannello di Controllo Unificato

Il cloud astrae via **tutta** questa complessità.

Le decine di interfacce, strumenti e competenze diverse vengono sostituite da un unico punto di controllo:

**Interfaccia Web (o API) → Click (o comando) → Risorsa Attiva**

Ogni click che fai nella console di AWS, Azure o Google Cloud è in realtà una chiamata API che comanda un'orchestra di robot e software nel data center del provider.

Questi robot eseguono per te, in pochi secondi, tutto il lavoro che prima richiedeva settimane e un team di specialisti.



## Esempi Pratici: Il Giorno e la Notte

### Prima: Lanciare un nuovo progetto

1. **Mese 1:** Il team di sviluppo ha un'idea. Si fa una richiesta di budget per un nuovo server.
2. **Mese 2:** Il budget viene approvato. Si ordina l'hardware.
3. **Mese 3:** Il server arriva. Viene installato fisicamente nel rack, cablato, e gli viene installato il sistema operativo.
4. **Mese 4:** Il server viene configurato, messo in sicurezza e finalmente consegnato al team di sviluppo. *Tempo totale: 3-4 mesi. Costo: migliaia di euro, anche se il progetto fallisce.*

### Ora: Lanciare un nuovo progetto

1. **Giorno 1, Ora 1:** Il team di sviluppo ha un'idea.
2. **Giorno 1, Ora 1, Minuto 2:** Si accede alla console cloud, si sceglie la potenza della macchina, l'immagine del sistema operativo e si clicca "Launch". *Tempo totale: 2 minuti. Costo: pochi centesimi all'ora. Se il progetto fallisce, si spegne la macchina e si smette di pagare.*

### Prima: Guasto di un hard disk alle 3 del mattino

1. **3:05 AM:** Suona il telefono. Un alert automatico segnala il guasto.
2. **3:30 AM:** Arrivi al data center. Identifichi il server e il disco rotto dalla lucina rossa lampeggiante.
3. **4:00 AM:** Sostituisci fisicamente il disco. Avvii la procedura di "rebuild" del RAID.
4. **8:00 AM:** Dopo 4 ore di attesa snervante, il RAID è ricostruito. Il sistema è di nuovo al sicuro.

**Ora: Guasto di un hard disk** L'hypervisor e la piattaforma cloud rilevano un'anomalia nel disco fisico. Il sistema sposta automaticamente e senza interruzioni la tua VM su un altro server sano. Tu non ti accorgi di nulla. Forse ricevi un'email che ti informa che un potenziale problema è stato risolto proattivamente.

### Il Vantaggio Reale: Da Centro di Costo a Motore di Innovazione

Questo è il punto cruciale. Liberando il tuo team dalla schiavitù della gestione fisica, l'astrazione del cloud non gli dà solo più tempo, ma ne cambia la natura stessa.

- **Prima:** Il team IT spendeva il **60-80%** del suo tempo in manutenzione e gestione dell'infrastruttura (il "tenere le luci accese"). Solo il restante **20-40%** poteva essere dedicato a migliorare le applicazioni e supportare il business. L'IT era visto come un centro di costo.
- **Ora:** La gestione dell'infrastruttura si riduce a una frazione del tempo. Il team IT può dedicare quasi il **100%** delle sue energie a ciò che crea valore reale: sviluppare nuove funzionalità, analizzare dati per trovare nuove opportunità di business, e sperimentare nuove idee a un costo quasi nullo.

L'IT si trasforma da un reparto che "ripara cose" a un **motore strategico per l'innovazione e la crescita aziendale**.

## La Rivoluzione della Scalabilità Cloud

Da previsioni rischiose a un'elasticità istantanea

### 1. Il Problema del "Prima": La Gabbia dell'Infrastruttura Fisica

Prima del cloud, la crescita era tanto desiderata quanto temuta, perché ogni aumento di carico si scontrava con i limiti rigidi del mondo fisico. Scalare era un processo costoso, lento e pieno di rischi.

#### Previsioni Errate: La Scommessa Impossibile

La pianificazione della capacità era come cercare di predire il futuro in una sfera di cristallo. Dovevi rispondere a una domanda impossibile: "Di quanta potenza avremo bisogno tra 6 mesi?". Questo portava a due scenari, entrambi dannosi:

- **Comprare Troppo (Overprovisioning):** Per paura di non essere pronti, si acquistavano server molto più potenti del necessario. Questo significava un enorme spreco di capitale (CapEx) per hardware che rimaneva inutilizzato per il 90% del tempo, consumando energia e occupando spazio.
- **Comprare Troppo Poco (Underprovisioning):** Per risparmiare, si acquistava solo il minimo indispensabile. Al primo picco di traffico, i server si saturavano, i servizi rallentavano o andavano in downtime, causando perdite di fatturato, danni alla reputazione e clienti frustrati.

#### Tempi Lunghi: L'Attesa che Uccide il Business

Se le tue previsioni si rivelavano sbagliate e avevi bisogno di più potenza, il processo era un incubo logistico che poteva durare mesi:

1. **Approvazione Budget:** Giustificare la spesa ai vertici aziendali.
2. **Acquisto:** Negoziare con i fornitori e ordinare l'hardware.
3. **Consegna:** Attendere settimane per la spedizione.
4. **Installazione Fisica:** Trovare spazio nel rack, collegare alimentazione e rete.
5. **Configurazione:** Installare sistema operativo, software e integrare nella rete esistente. In un mercato che si muove velocemente, un ritardo di mesi significava perdere l'opportunità.

## La Rivoluzione della Scalabilità Cloud

### Investimenti Anticipati (CapEx vs. OpEx)

L'infrastruttura fisica richiedeva un enorme **investimento di capitale iniziale (CapEx)**. Dovevi pagare tutto subito, immobilizzando decine o centinaia di migliaia di euro per una capacità che, forse, avresti usato solo in futuro. Il cloud trasforma questo modello in una **spesa operativa (OpEx)**, un costo variabile che paghi mese per mese, come una bolletta, basato sull'uso effettivo.

### Limiti Fisici Invalicabili

A un certo punto, i soldi non bastavano più. Un data center ha limiti fisici concreti:

- **Spazio Rack**: I rack hanno un numero finito di slot.
- **Alimentazione**: L'impianto elettrico può fornire solo un tot di kilowatt.
- **Raffreddamento**: L'impianto di climatizzazione può dissipare solo una certa quantità di calore. Raggiunto il limite, l'unica soluzione era costruire un nuovo data center, un progetto da milioni di euro e anni di lavoro.

## 2. La Magia dell'Astrazione: L'Elasticità dell'Auto Scaling

L'astrazione del cloud sgancia le tue applicazioni da questi vincoli. Le risorse diventano un "pool" virtualmente infinito che può espandersi e contrarsi su richiesta, in modo automatico. È come avere un supermercato che apre e chiude le casse in tempo reale in base alla lunghezza delle code.

- **Carico Alto → Auto Scaling → Nuove Istanze (2 minuti)** Quando il sistema (tramite regole che imposta tu, es. "utilizzo CPU > 70%") rileva un aumento del carico, comanda all'hypervisor di creare nuove macchine virtuali da un template pre-configurato. In pochi minuti, queste nuove istanze sono attive e il bilanciatore di carico inizia a inviare loro traffico, distribuendo lo sforzo.
- **Carico Basso → Auto Scaling → IstanzeTerminate (1 minuto)** Quando il carico diminuisce (es. "utilizzo CPU < 30%"), il sistema fa l'opposto. Identifica le istanze superflue, le rimuove dal bilanciatore di carico e le "distrugge", smettendo istantaneamente di pagarle.

## La Rivoluzione della Scalabilità Cloud

### 3. Esempi Concreti: Dal Rischio all'Opportunità

#### E-commerce durante il Black Friday

- **Prima:** Per gestire il picco di un giorno, dovevi comprare e mantenere 10 server per 365 giorni l'anno. Un costo enorme per una capacità sfruttata solo per lo 0.2% del tempo.
- **Ora:** Parti con un solo server. L'auto-scaling ne aggiunge automaticamente altri 9 (o più) solo per le 24-48 ore del Black Friday. Finito il picco, torna a un solo server. Il costo è legato solo all'uso effettivo, trasformando una spesa folle in un costo marginale.

#### Startup in Crescita Esplosiva

- **Prima:** Un'idea geniale poteva essere bloccata dalla mancanza dei 50.000 € di capitale iniziale per l'hardware necessario a "prepararsi al successo".
- **Ora:** Lanci la tua applicazione con una spesa di 10 €/mese. Se gli utenti arrivano, il sistema scala automaticamente con loro. Il costo cresce in proporzione alle entrate, non prima. La barriera all'ingresso è stata demolita.

#### App che Diventa Virale

- **Prima:** Un successo improvviso e inaspettato era un "disastro felice". Il sito veniva travolto dal traffico e andava offline, trasformando un'enorme opportunità di marketing in un fallimento pubblico.
- **Ora:** L'auto-scaling è la tua rete di sicurezza. Che arrivino 100 o 100.000 utenti in un'ora, il sistema si adatta in tempo reale, garantendo che l'app rimanga stabile e reattiva. Il successo viene cavalcato, non subito.

### 4. I Tipi di Scalabilità a Disposizione

L'Auto Scaling è il cervello che utilizza due strumenti principali per adattare le risorse:

- **Vertical Scaling (Scale Up): Rendere una macchina più potente.** È come dare steroidi a un singolo server, aumentandone CPU, RAM o disco. Utile per carichi di lavoro difficili da distribuire (es. alcuni tipi di database), ma ha un limite fisico: non puoi ingrandire una singola VM all'infinito.
- **Horizontal Scaling (Scale Out): Aggiungere più macchine.** Invece di potenziare una macchina, ne aggiungi altre identiche che lavorano in parallelo. È il metodo più potente e flessibile, alla base della quasi totalità delle grandi architetture cloud.
- **Auto Scaling: L'orchestratore intelligente.** È il servizio che, basandosi sulle regole che hai definito, decide autonomamente se è il momento di fare Scale Up o Scale Out (o il contrario, Scale Down/In) per mantenere le performance ottimali al minor costo possibile.

## La Flessibilità del Cloud: Scegliere lo Strumento Giusto per Ogni Lavoro

Oltre la scalabilità: l'arte di adattarsi, sperimentare e innovare senza vincoli.

### 1. Il Problema del "Prima": Le Catene dell'Infrastruttura Fisica

Nel mondo dell'IT tradizionale, ogni decisione hardware era un impegno a lungo termine, una sorta di "matrimonio tecnologico" difficile da sciogliere. Questa rigidità soffocava l'agilità e rendeva ogni cambiamento un progetto complesso e costoso.

#### Vendor Lock-in Hardware: La Prigione Dorata

Una volta scelto un fornitore di hardware (come Dell, HP, IBM), si entrava nel suo ecosistema. I server, lo storage e i componenti di rete erano progettati per funzionare al meglio tra loro, ma spesso in modo incompatibile con la concorrenza. Questo creava una dipendenza forzata: anche se un altro fornitore offriva una soluzione migliore o più economica, il costo e la complessità della migrazione erano così alti da renderla impraticabile. Eri bloccato per anni, costretto a seguire le roadmap e i prezzi di un singolo vendor.

#### Configurazioni Fisse: La Camicia di Forza Tecnologica

Il server che compravi era una fotografia delle tue esigenze in quel preciso momento. Ma il business non è statico. Se dopo un anno la tua applicazione avesse avuto bisogno di più RAM e meno CPU, o di dischi più veloci, non c'era nulla da fare. La macchina rimaneva quella, con le sue specifiche fisse, per i suoi 3-5 anni di ciclo di vita. Era come comprare un'auto sportiva a due posti per poi scoprire di aver bisogno di un furgone: la decisione iniziale ti condizionava per tutto il viaggio.

#### Sperimentazione Costosa: L'Innovazione come Rischio Finanziario

Come si poteva testare una nuova tecnologia promettente? Se il team di sviluppo voleva provare un nuovo database NoSQL o una piattaforma di analisi dati, l'unica via era comprare nuovo hardware dedicato. Questo significava avviare un processo di acquisto da migliaia di euro solo per un "forse". La sperimentazione non era vista come un'opportunità, ma come un rischio finanziario. Di conseguenza, si innovava meno e ci si affidava solo a tecnologie collaudate, anche se obsolete.

## La Flessibilità del Cloud: Scegliere lo Strumento Giusto per Ogni Lavoro

### 2. Il Potere dell'Astrazione: Un Catalogo Infinito di "Lego" Tecnologici

L'astrazione del cloud demolisce queste barriere. Non compri più "server", ma accedi a un immenso catalogo di servizi specializzati, come se avessi a disposizione i mattoncini Lego di ogni forma e colore. La vera magia non è solo avere tante opzioni, ma la libertà di combinarle, scambiarle e riconfigurarle in qualsiasi momento.

#### Compute Options (Potenza di Calcolo): Lo strumento su misura

Puoi scegliere la potenza di calcolo come un abito sartoriale, perfettamente adatto al tuo carico di lavoro.

- **t2.micro (1 vCPU, 1GB RAM)** → Per lo sviluppo e i test iniziali, dove il costo è la priorità.
- **c5.xlarge (4 vCPU, 8GB RAM)** → Il cavallo di battaglia per applicazioni web standard, un equilibrio tra costo e performance.
- **r5.24xlarge (96 vCPU, 768GB RAM)** → Una bestia ottimizzata per la RAM, ideale per database in-memory come SAP HANA o cache di grandi dimensioni.
- **p3.16xlarge (64 vCPU + 8 GPU NVIDIA V100)** → Una centrale nucleare per il calcolo parallelo, indispensabile per il training di modelli complessi di Machine Learning.

#### Storage Options (Archiviazione): Il posto giusto per ogni dato

Non tutti i dati sono uguali. Il cloud ti permette di pagare solo per le performance di cui hai bisogno.

- **S3 Standard** → Per dati "caldi" ad accesso frequente, come le immagini e i file CSS del tuo sito web. Massima disponibilità e velocità.
- **S3 Infrequent Access (IA)** → Per dati "tiepidi" come backup mensili o file a cui accedi poche volte all'anno. Stessa velocità di S3 Standard, ma con un costo di archiviazione inferiore del 50%.
- **S3 Glacier** → Per dati "freddi" di archivio legale o storico. Il recupero richiede più tempo (da minuti a ore), ma il costo di archiviazione è fino al 90% più basso.
- **EBS SSD** → Dischi a stato solido ultra-veloci, collegati direttamente alle tue macchine virtuali. Indispensabili per i file di sistema e i database che richiedono tempi di risposta minimi.
- **EBS HDD** → Dischi magnetici tradizionali, più lenti ma molto più economici. Perfetti per grandi volumi di dati dove la velocità non è critica (es. log, elaborazioni batch).

#### Database Options (Basi di Dati): Il motore giusto per la tua applicazione

Scegliere il database giusto può decretare il successo o il fallimento di un progetto. Il cloud ti offre un'intera officina.

- **RDS MySQL/PostgreSQL** → Database relazionali tradizionali, perfetti per e-commerce e applicazioni gestionali, ma offerti come servizio gestito (niente più patch o backup manuali).
- **DynamoDB** → Un database NoSQL chiave-valore con performance garantite nell'ordine dei millisecondi, indipendentemente dalla scala. Ideale per app mobili, gaming e IoT.
- **Redshift** → Un data warehouse colonnare, ottimizzato per eseguire analisi complesse (query analitiche) su petabyte di dati in pochi secondi.
- **ElastiCache** → Un servizio di cache in-memory (Redis o Memcached) per ridurre la latenza delle tue applicazioni, servendo le richieste più comuni direttamente dalla memoria RAM.

## La Flessibilità del Cloud: Scegliere lo Strumento Giusto per Ogni Lavoro

### 3. Esempi di Flessibilità Reale: Adattarsi alla Velocità del Business

#### Progetto Machine Learning: Dal Prototipo alla Produzione

- **Giorno 1-29:** Un data scientist scrive il codice su un'istanza `t2.micro` a basso costo.
- **Giorno 30:** Per la fase di training, che richiede un'enorme potenza di calcolo per 12 ore, accende un'istanza `p3.16xlarge` con 8 GPU.
- **Giorno 31:** Finito il training, l'istanza GPU viene spenta. Il modello "addestrato" viene messo in produzione su un'istanza `c5.xlarge` economica, che gestisce le richieste degli utenti.
- **Risultato:** Ha pagato la potenza estrema della GPU solo per le 12 ore in cui era indispensabile, ottimizzando i costi in ogni fase del progetto.

#### Startup che Cambia Strategia (Pivot): Da E-commerce a IoT

- **Mesi 1-6:** Una startup lancia un e-commerce, basato su un'architettura classica con un database relazionale (RDS) per gestire ordini e utenti.
- **Mese 7:** Il mercato non risponde, ma si accorgono di un'opportunità nel settore dell'Internet of Things (IoT) per monitorare sensori. Questo richiede di gestire milioni di piccoli dati in arrivo ogni secondo.
- **Mese 8:** In pochi giorni, il team dismette il database RDS e attiva un database time-series ottimizzato per l'IoT. L'intera architettura viene riconfigurata senza comprare un solo pezzo di hardware nuovo. L'azienda si è adattata alla velocità del mercato, non a quella dei cicli di acquisto hardware.

#### Sperimentazione: Creare una Cultura dell'Innovazione

- **Lunedì, ore 10:00:** Un programmatore ha un'idea per una nuova funzionalità.
- **Lunedì, ore 10:05:** Crea un ambiente di test completo (server, database, rete) con un click.
- **Martedì, ore 16:00:** Il test dimostra che l'idea non funziona come previsto.
- **Martedì, ore 16:01:** L'intero ambiente viene eliminato con un altro click.
- **Costo Totale dell'Esperimento:** Meno di 5 €. Invece di un fallimento costoso, è diventato un apprendimento economico. Questa libertà di provare e sbagliare a basso costo è il vero motore dell'innovazione.

# Cloud Computing

## Il Risultato Combinato

Quando **Simplified Management**, **Scalability** e **Flexibility** lavorano insieme:

### Time-to-Market Rivoluzionario

- Prima: 6 mesi per setup infrastruttura + 6 mesi sviluppo = 12 mesi
- Ora: 1 giorno setup + 6 mesi sviluppo = 6 mesi

### Risk Reduction

- Prima: Investimento €100,000 upfront, sperando nel successo
- Ora: Investimento €100/mese, scaling in base al successo reale

### Innovation Acceleration

- Prima: 1 esperimento/anno (troppo costoso fallire)
- Ora: 10 esperimenti/mese (economico testare tutto)

### Focus on Core Business

#### Team Time Distribution:

Prima: 30% Business Logic + 70% Infrastructure Management

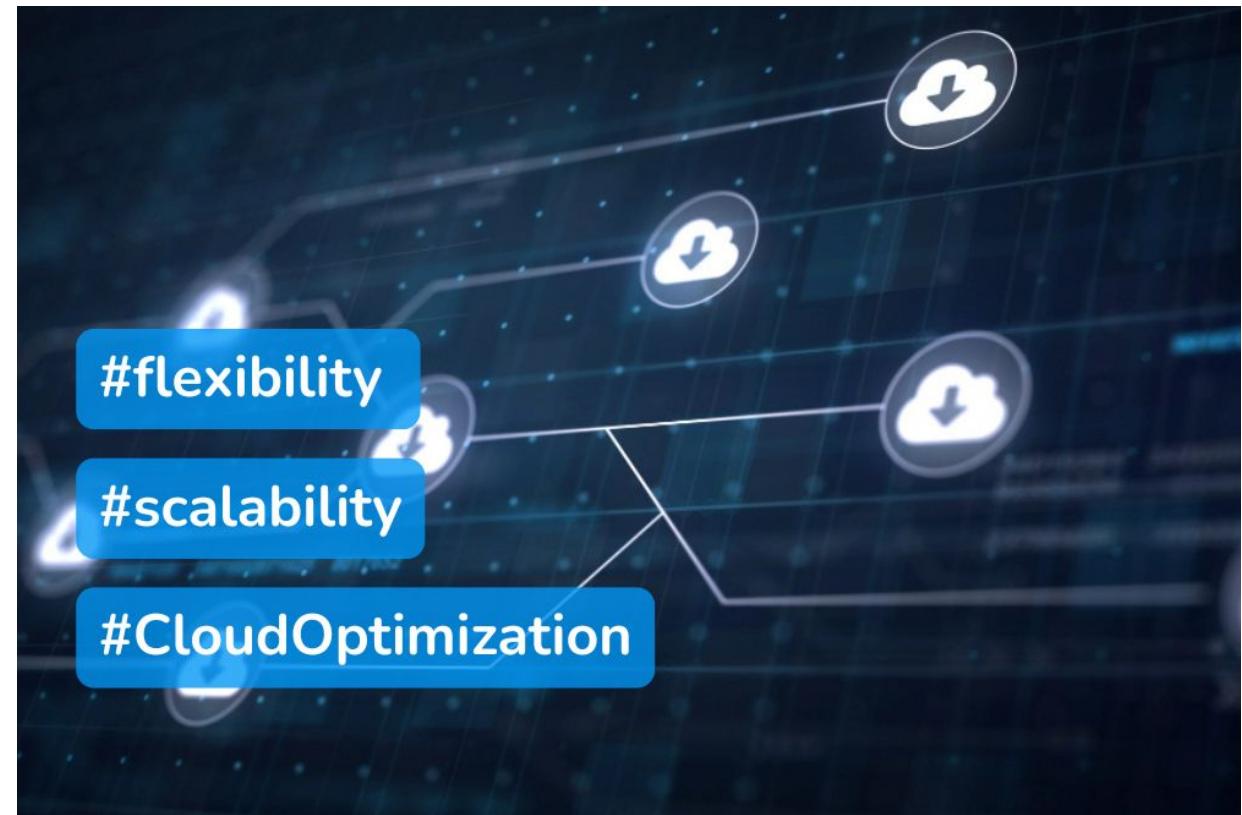
Ora: 90% Business Logic + 10% Cloud Configuration

### Conclusione: L'Astrazione è una Leva di Business

L'astrazione del cloud non è solo una comodità tecnica - è una **leva strategica di business** che:

- **Democratizza la tecnologia:** Startup possono accedere alla stessa infrastruttura di Google
- **Accelerà l'innovazione:** Fail fast, learn fast, scale fast
- **Riduce i rischi:** Pay-as-you-grow invece di bet-the-company investments
- **Libera talenti:** Sviluppatori che risolvono problemi business, non problemi di infrastruttura

L'astrazione trasforma l'IT da "centro di costo" a "motore di crescita"!



Immagina di dover organizzare una festa. Hai tre opzioni:

1. **Farla a casa tua** (devi comprare tutto, pulire, cucinare) = **Cloud Privato**
2. **Affittare una sala** (paghi solo per quella sera, qualcun altro pulisce) = **Cloud Pubblico**
3. **Usare entrambi** (antipasti a casa, cena in sala) = **Cloud Ibrido**
4. **Affittare più sale contemporaneamente** (una per il pranzo, una per la cena) = **Multi-Cloud**

Allo stesso modo, quando un'azienda ha bisogno di server, storage e risorse informatiche, può scegliere **dove e come** gestirle. Vediamo le quattro opzioni principali.

## Cloud Pubblico

### Cos'è in parole semplici?

Affitti computer potentissimi (server) da grandi aziende come Amazon (AWS), Microsoft (Azure) o Google (Google Cloud). Questi computer non sono fisicamente da te, ma nei loro enormi magazzini pieni di server chiamati **data center**.

**Data center** = Un gigantesco capannone pieno di migliaia di computer che lavorano 24 ore su 24.

### Come funziona?

1. Accedi a un sito web (la console di AWS)
2. Clicchi su "crea un server"
3. In 2 minuti hai un computer virtuale pronto
4. Paghi solo per il tempo che lo usi (come la corrente elettrica: usi più consumi, più paghi)
5. Quando non ti serve più, lo spegni e smetti di pagare

### Vantaggi spiegati semplicemente:

#### Non devi comprare nulla all'inizio

- Non serve spendere 50.000 euro per server fisici
- Inizi con 10 euro al mese

#### Velocità

- Prima: ordinavi un server, aspettavi 3 mesi
- Ora: clicchi un bottone, server pronto in 2 minuti

## Cresce con te

- Hai 10 utenti oggi? Usi un server piccolo
- Domani ne hai 10.000? Il sistema crea automaticamente 100 server
- Dopodomani tornano a essere 10? Si torna a 1 server
- Paghi sempre solo quello che usi

## Zero manutenzione

- Si rompe un disco? Amazon lo cambia (tu neanche te ne accorgi)
- Serve un aggiornamento? Lo fa Amazon
- Manca corrente? Amazon ha generatori di emergenza

# Modelli di Deployment del Cloud

## Svantaggi spiegati semplicemente:

### Non controlli l'hardware fisico

- Non puoi entrare nella stanza dei server
- Non sai su quale computer fisico gira il tuo sito

### Alcune leggi non lo permettono

- Alcune banche devono tenere i dati in Italia per legge
- Certi ospedali non possono mettere dati sanitari fuori dall'edificio

### La spesa non è sempre prevedibile

- Se dimentichi un server acceso, continui a pagare
- Se il tuo sito diventa virale, la bolletta può esplodere

### Esempio pratico:

#### Mario apre un negozio online

- Giorno 1: 10 visitatori → server piccolo → 5 euro al mese
- Black Friday: 10.000 visitatori → 50 server automatici per 2 giorni → 300 euro
- Gennaio: torna normale → 5 euro al mese

Se Mario avesse comprato server fisici per il Black Friday, avrebbe speso 50.000 euro e per 363 giorni all'anno sarebbero rimasti spesi soldi.

## Cloud Privato

### Cos'è in parole semplici?

Compri i tuoi server fisici e li metti in una stanza nel tuo ufficio (o in un capannone di tua proprietà). Sono tuoi, li tocchi, li controlli completamente.

Questo è il vecchio modo di lavorare, quello che tutte le aziende facevano prima del 2010.

### Come funziona?

1. Comperi server Dell o HP (computer molto potenti)
2. Li installi in una stanza con aria condizionata (si scaldano molto)
3. Assumi persone che li gestiscono
4. Se si rompe qualcosa, lo ripari tu
5. Se serve più potenza, compri nuovi server (e aspetti settimane)

## Vantaggi spiegati semplicemente:

### Controllo totale

- Puoi entrare nella stanza e vedere i server
- Decidi tu tutto: marca, modello, configurazione

### Nessuno può spiare i tuoi dati

- I server sono nel tuo ufficio, sotto chiave
- Nessun estraneo può accedervi fisicamente

### Nessuna bolletta a sorpresa

- Hai già pagato tutto all'inizio
- La corrente elettrica è l'unico costo variabile

### Funziona senza Internet

- Se salta la connessione Internet, i tuoi sistemi interni continuano a funzionare

## Cloud Privato

### Svantaggi spiegati semplicemente:

#### Devi comprare tutto subito

- Investimento iniziale: da 50.000 a 500.000 euro
- Se l'idea di business fallisce, hai buttato tutti i soldi

#### Serve molto tempo

- Vuoi un nuovo server? Ordina, aspetta 3 settimane, installalo, configuralo
- Sul cloud: 2 minuti

#### Devi assumere persone

- Servono tecnici informatici a tempo pieno
- Stipendi da 30.000 a 50.000 euro all'anno ciascuno

#### Se si rompe, problemi tuoi

- Disco rotto alle 3 di notte? Svegli il tecnico
- Finita la corrente? Se non hai un generatore, tutto si ferma

#### Spreco di risorse

- Comprì 10 server per gestire i picchi
- Per l'80% del tempo solo 2 sono usati, gli altri 8 sprecano corrente

## Cloud Privato

**Esempio pratico:**

**Una banca ha 200 dipendenti**

- Sala server nel seminterrato della sede
- 20 server fisici Dell in armadi metallici
- Sistema di raffreddamento dedicato
- 5 tecnici informatici che li gestiscono
- Generatore diesel per i blackout
- Investimento iniziale: 300.000 euro

**Perché lo fa?**

- La legge dice: "i dati bancari devono stare in Italia, sotto il tuo controllo"
- Non può usare AWS perché i server di Amazon potrebbero essere in America

## Cloud Ibrido

### Cos'è in parole semplici?

Usi entrambi i sistemi insieme: alcuni server sono tuoi (nel tuo ufficio), altri li affitti da Amazon (sul cloud pubblico). È come avere una cucina a casa ma ordinare anche pizze d'asporto quando serve.

### Come funziona?

1. Tieni i dati importanti/segreti sui tuoi server
2. Metti le cose meno critiche su AWS
3. I due sistemi comunicano tramite Internet in modo sicuro
4. Quando serve potenza extra, "salti" su AWS temporaneamente

### Vantaggi spiegati semplicemente:

#### Prendi il meglio di entrambi

- Dati sensibili: sui tuoi server (controllo totale)
- Sito web: su AWS (scalabilità automatica)

#### Rispetti le leggi e rimani agile

- Conti bancari: sui tuoi server (per legge)
- App mobile: su AWS (per innovare velocemente)

#### Gestisci i picchi senza sprechi

- Normale: usi i tuoi server
- Black Friday: affitti 50 server extra su AWS per 2 giorni
- Dopo: torni ai tuoi server

## Cloud Irido

**Svantaggi spiegati semplicemente:**

**Complessità doppia**

- Devi gestire sia i tuoi server che quelli su AWS
- Serve il doppio delle competenze

**Costi extra per connettere**

- Serve una connessione Internet velocissima e sicura
- Può costare migliaia di euro al mese

**Due posti da proteggere**

- Devi mettere sicurezza sui tuoi server
- E anche sui server AWS
- E sulla connessione tra i due

## Cloud Ibrido

### Esempio pratico:

Un ospedale con 500 medici

### Sul cloud privato (nel suo edificio):

- Cartelle cliniche dei pazienti (dati sensibili, per legge devono rimanere lì)
- Esami medici e diagnosi
- Sistema di prenotazioni

### Sul cloud pubblico (AWS):

- Sito web dell'ospedale
- App per prenotare visite
- Sistema di analisi dati (intelligenza artificiale per ricerca medica)

### Perché?

- I dati dei pazienti non possono uscire dall'ospedale per privacy
- Ma l'app mobile può stare su AWS per scalare meglio\

Quando un paziente prenota online, l'app su AWS comunica in modo sicuro con il sistema interno dell'ospedale.

# Modelli di Deployment del Cloud

## Multi-Cloud

### Cos'è in parole semplici?

Invece di usare solo Amazon (AWS), usi contemporaneamente Amazon, Microsoft (Azure) e Google (Google Cloud). È come fare la spesa in 3 supermercati diversi per prendere le offerte migliori da ciascuno.

### Come funziona?

1. Metti il sito web su Amazon perché costa meno
2. Usi l'intelligenza artificiale di Google perché è la migliore
3. Usi Azure di Microsoft perché si integra bene con Office 365
4. Tutto deve comunicare tra loro (complicato!)

### Vantaggi spiegati semplicemente:

#### Non dipendi da uno solo

- Se Amazon ha problemi, Google ti salva
- Se domani Amazon raddoppia i prezzi, hai alternative pronte

#### Prendi il meglio di ognuno

- Intelligenza artificiale? Google è il migliore
- Database? Amazon è il migliore
- Autenticazione? Microsoft è il migliore

#### Rispetti leggi diverse per paese

- In Europa usi AWS (deve rispettare GDPR, legge europea sulla privacy)
- In Cina usi Alibaba Cloud (obbligatorio per legge cinese)

## Multi-Cloud

### Svantaggi spiegati semplicemente:

#### Complessità massima

- Immagina dover imparare 3 lingue diverse contemporaneamente
- Ogni cloud funziona in modo diverso

#### Serve un team enorme

- Devi avere esperti di Amazon, Microsoft E Google
- Stipendi per almeno 10-15 persone

#### Trasferire dati costa

- Spostare file da Amazon a Google può costare migliaia di euro
- Come se ogni supermercato ti facesse pagare per portare la spesa da uno all'altro

#### Bollette separate

- 3 fatture diverse da controllare ogni mese
- Difficile capire dove spendi di più

## Multi-Cloud

**Esempio pratico:**

**Una grande compagnia internazionale che fa videogiochi**

**Usa Amazon AWS per:**

- Server di gioco in Europa e America
- Storage dei file del gioco (immagini, suoni)

**Usa Google Cloud per:**

- Intelligenza artificiale che controlla i bot nemici
- Analisi di come giocano gli utenti

**Usa Microsoft Azure per:**

- Accesso con account Microsoft (Xbox)
- Integrazione con console Xbox

**Perché lo fa?**

- Ognuno è specializzato in qualcosa
- Se un cloud ha problemi, il gioco continua a funzionare sugli altri
- Può negoziare sconti con tutti e tre

# Modelli di Deployment del Cloud

## Confronto Rapido: Quale Scegliere?

Criterio	Cloud Pubblico	Cloud Privato	Cloud Ibrido	Multi-Cloud
Costi iniziali	Bassissimi	Molto alti	Alti	Bassi
Scalabilità	Illimitata	Limitata	Alta	Illimitata
Controllo	Basso	Totale	Medio	Medio
Complessità	Bassa	Alta	Molto alta	Massima
Time-to-market	Minuti	Mesi	Settimane	Minuti
Conformità	Limitata	Totale	Alta	Media
Vendor lock-in	Alto	Nessuno	Medio	Basso
Team richiesto	Piccolo	Grande	Grande	Molto grande

# Considerazioni finali

## Cloud Pubblico (AWS, Azure, Google)

**Analogia:** Affittare un appartamento

- Non è tuo, ma lo usi quando serve
- Paghi un canone mensile
- Se si rompe qualcosa, chiama il padrone di casa

## Cloud Privato

**Analogia:** Comprare una casa

- È tua, fai quello che vuoi
- Paghi tutto all'inizio
- Se si rompe qualcosa, ripari tu

## Cloud Ibrido

**Analogia:** Avere casa tua + affittare un garage separato

- Casa per le cose importanti (famiglia)
- Garage affittato per cose extra (attrezzi)

## Multi-Cloud

**Analogia:** Avere 3 case in affitto in città diverse

- Una a Milano, una a Roma, una a Napoli
- Usi quella più comoda secondo dove sei
- Ma devi gestire 3 contratti diversi

## Quale Scegliere? Guida Semplice

- **Sei una startup o piccola azienda?** → Cloud Pubblico (AWS). Parti in piccolo, cresci velocemente.
- **Sei una banca o un ospedale?** → Cloud Privato o Ibrido. Hai dati sensibili che per legge devono rimanere sotto il tuo controllo.
- **Sei una multinazionale enorme?** → Multi-Cloud. Hai i soldi e le persone per gestire la complessità.
- **Non sai ancora?** → Inizia con Cloud Pubblico. È il modo più facile e economico per part

## La Piramide della Responsabilità

Immagina di voler mangiare una pizza. Hai tre opzioni:

1. **Compri gli ingredienti e la fai da zero** (farina, lievito, pomodoro, mozzarella) = **IaaS**
2. **Compri l'impasto pronto e aggiungi i condimenti** = **PaaS**
3. **Ordini la pizza già pronta a domicilio** = **SaaS**

Allo stesso modo, nel cloud puoi scegliere **quanto lavoro vuoi fare tu e quanto vuoi che faccia il provider**.

Più “in basso” sei nella piramide (IaaS), più controllo e libertà hai, ma anche più responsabilità nella gestione. Man mano che sali verso PaaS e SaaS, il provider si occupa di sempre più aspetti tecnici, permettendoti di concentrarti solo sul tuo obiettivo: sviluppare, gestire o semplicemente usare il servizio senza pensieri.

## (Infrastruttura come Servizio)

Ti danno i **mattoni base**: computer virtuali vuoti, spazio disco, rete. Tu devi installare tutto il resto: sistema operativo, database, programmi, configurazioni.

È come affittare un appartamento completamente vuoto: hai le mura e i pavimenti, ma devi portare mobili, elettrodomestici, decorazioni.

### Cosa fornisce il provider (AWS)?

- **Server virtuali** (chiamati "istanze" o "macchine virtuali")
- **Spazio disco virtuale** (storage)
- **Rete virtuale** (connessione tra i server)
- **Firewall virtuali** (per la sicurezza)

### Cosa devi fare TU?

- Installare il sistema operativo (Windows, Linux)
- Installare i programmi (database, web server)
- Configurare tutto
- Aggiornare regolarmente
- Gestire la sicurezza del software
- Fare backup

## Servizi AWS che sono IaaS:

### EC2 (Elastic Compute Cloud)

- Cosa significa: computer virtuali che puoi accendere e spegnere
- Elastic = si adatta, cresce o diminuisce
- Compute = calcolo, potenza del processore
- Cloud = nel cloud, non fisico

Esempio: "Crea un server virtuale con 4 processori e 16 GB di memoria RAM"

### VPC (Virtual Private Cloud)

- Virtual = virtuale, non fisico
- Private = privato, isolato dagli altri
- Cloud = nel cloud

Esempio: "Crea una rete privata dove i tuoi server comunicano in modo sicuro"

### EBS (Elastic Block Storage)

- Elastic = si adatta
- Block Storage = spazio disco organizzato in blocchi
- Come un hard disk che attacchi al server virtuale

## Vantaggi del IaaS:

### Controllo totale

- Installi quello che vuoi
- Configuri come preferisci
- Usi qualsiasi tecnologia

### Flessibilità massima

- Vuoi un database particolare? Lo installi
- Serve un software specifico? Nessun problema

### Economico per esperti

- Se sai fare tutto da solo, paghi solo l'infrastruttura base

## Svantaggi del IaaS:

### Serve competenza tecnica

- Devi sapere installare e configurare server
- Devi capire di sicurezza informatica
- Devi gestire aggiornamenti

### Tanto lavoro

- Ogni software va installato manualmente
- Devi monitorare che tutto funzioni
- Gli aggiornamenti sono responsabilità tua

### Tempo

- Configurare tutto può richiedere giorni o settimane

## (Piattaforma come Servizio)

### Cos'è in parole semplici?

Ti danno un **ambiente già pronto** dove puoi solo caricare il tuo codice. Il sistema operativo, il database, gli aggiornamenti li gestisce qualcun altro.

È come affittare un appartamento ammobiliato: i mobili ci sono già, tu porti solo le tue cose personali.

### Cosa fornisce il provider (AWS)?

- Server virtuali **già configurati**
- Sistema operativo **già installato**
- Database **già funzionante**
- Aggiornamenti **automatici**
- Backup **automatici**
- Scalabilità **automatica**

### Cosa devi fare TU?

- Caricare il tuo codice (la tua applicazione)
- Configurare poche impostazioni base
- Nient'altro, il resto è gestito

## Servizi AWS che sono PaaS:

### Elastic Beanstalk

- Elastic = si adatta automaticamente
- Beanstalk = germoglio di fagiolo (simbolo di crescita automatica)

Cosa fa: carichi il tuo codice, lui crea automaticamente server, database, rete, tutto.

Esempio: "Carico il mio sito WordPress e in 5 minuti è online"

### RDS (Relational Database Service)

- Relational Database = database tradizionale (MySQL, PostgreSQL)
- Service = servizio gestito

Cosa fa: ti dà un database già funzionante, fa backup automatici, aggiornamenti automatici.

Esempio: "Clicco 'crea database MySQL' e in 10 minuti è pronto, senza installare nulla"

### Lambda

- Nome ispirato alla programmazione funzionale
- Esegui solo codice, senza pensare ai server

Cosa fa: carichi una funzione (pezzo di codice), si attiva quando serve, paghi solo quando viene usata.

Esempio: "Ogni volta che qualcuno carica una foto, ridimensionala automaticamente"

## Vantaggi del PaaS:

### Velocità

- Da idea a prodotto funzionante in ore, non giorni
- Non devi configurare nulla

### Meno competenze richieste

- Non serve essere esperti di server
- Ti concentri solo sul tuo codice

### Manutenzione automatica

- Gli aggiornamenti li fa Amazon
- I backup li fa Amazon
- La sicurezza del sistema la gestisce Amazon

### Costa meno in tempo

- Nessun tecnico informatico da assumere per gestire i server

## Svantaggi del PaaS:

### Meno controllo

- Non puoi cambiare tutto, solo alcune configurazioni
- Sei vincolato a quello che offre il servizio

### Può costare di più

- Paghi per la comodità
- A volte costa il 20-30% in più di IaaS

### Dipendenza dal provider

- Se usi servizi specifici di AWS, migrare altrove è difficile

## (Software come Servizio)

### Cos'è in parole semplici?

Ti danno un **programma già pronto da usare** tramite browser. Non installi nulla, non configuri nulla, apri il sito e inizi a usarlo.

È come andare al ristorante: non cucini, non lavi i piatti, ti siedi e mangi.

### Cosa fornisce il provider?

- **Tutto:** software, server, database, sicurezza, backup
- Tu hai solo un **account** con username e password
- Accedi e usi

### Cosa devi fare TU?

- Creare un account
- Usare il programma
- Nient'altro

## Esempi comuni di SaaS (NON solo AWS):

### Gmail

- Email già pronta
- Non installi server, non configuri nulla
- Apri gmail.com e scrivi email

### Google Drive

- Spazio disco online
- Salvi file senza pensare a dove sono fisicamente

### Netflix

- Video streaming
- Non gestisci server video, guardi e basta

### Salesforce

- Software per gestire clienti (CRM - Customer Relationship Management)
- Le aziende lo usano per tenere traccia di vendite e clienti

### Dropbox

- Backup e sincronizzazione file
- Salvi un file, compare su tutti i tuoi dispositivi

## Servizi AWS che sono SaaS:

AWS non è famoso per SaaS (è più forte in IaaS e PaaS), ma ha alcuni esempi:

### Amazon Chime

- Videochiamate aziendali (come Zoom)

### Amazon WorkMail

- Email aziendale (come Gmail per aziende)

### Amazon WorkDocs

- Condivisione documenti (come Google Drive)

## Vantaggi del SaaS:

### Zero competenze tecniche

- Usi il programma come usi Facebook
- Nessuna configurazione

### Costi prevedibili

- Di solito paghi un abbonamento fisso mensile
- Esempio: 10 euro al mese per utente

### Sempre aggiornato

- Le nuove funzioni arrivano automaticamente
- Non devi scaricare update

### Accessibile ovunque

- Funziona su qualsiasi dispositivo con browser
- Computer, tablet, smartphone

### Nessuna installazione

- Non occupi spazio sul tuo PC
- Non rallenta il computer

## Svantaggi del SaaS:

### Zero controllo

- Il programma è quello che è
- Non puoi modificarlo o personalizzarlo molto

### Dipendenza totale

- Se il servizio chiude, perdi tutto
- Se cambia prezzo, devi accettare

### I dati sono altrove

- I tuoi file sono sui server della compagnia
- Devi fidarti della loro sicurezza

### Serve Internet sempre

- Senza connessione, non funziona
- (Alcuni hanno modalità offline limitata)

## Modello Tradizionale

**TU gestisci tutto:**

- ✓ Hardware fisico (server, dischi)
- ✓ Virtualizzazione
- ✓ Sistema operativo
- ✓ Software base (database, web server)
- ✓ Tua applicazione
- ✓ Dati

## IaaS (Infrastructure as a Service)

**Cloud gestisce:**

- ✓ Hardware fisico
- ✓ Virtualizzazione

**TU gestisci:**

- ✓ Sistema operativo
- ✓ Software base
- ✓ Tua applica

## PaaS (Platform as a Service)

**Cloud gestisce:**

- ✓ Hardware fisico
- ✓ Virtualizzazione
- ✓ Sistema operativo
- ✓ Software base

**TU gestisci:**

- ✓ Tua applicazione
- ✓ Dati

## SaaS (Software as a Service)

**Cloud gestisce tutto:**

- ✓ Hardware fisico
- ✓ Virtualizzazione
- ✓ Sistema operativo
- ✓ Software base
- ✓ Applicazione

**TU gestisci:**

- ✓ Solo i tuoi dati (cosa scrivi nell'applicazione)

# Quale Scegliere? Guida Pratica

## Scegli IaaS se:

- Sei uno sviluppatore esperto
- Vuoi controllo totale
- Hai esigenze molto specifiche
- Hai già software particolare da installare

**Esempio:** "Voglio installare un database particolare che nessuno offre come servizio gestito"

## Scegli PaaS se:

- Vuoi concentrarti sul codice, non sull'infrastruttura
- Vuoi velocità di sviluppo
- Non hai un team enorme di sistemisti
- Vuoi manutenzione automatica

**Esempio:** "Ho un'idea per un'app, voglio svilupparla velocemente senza pensare a server"

## Scegli SaaS se:

- Non sei uno sviluppatore
- Vuoi solo usare un programma
- Non vuoi gestire nulla di tecnico
- Hai bisogno di qualcosa di standard

**Esempio:** "Voglio una email professionale per la mia azienda, senza complicazioni"

# Quale Scegliere? Guida Pratica

## Startup che crea un'app mobile

### Usa IaaS per:

- Server API personalizzato con configurazione speciale

### Usa PaaS per:

- Database gestito (RDS)
- Funzioni serverless per notifiche push (Lambda)

### Usa SaaS per:

- Email azienda (Gmail/WorkMail)
- Chat interna team (Slack)
- Videoconferenze (Zoom)

## Agenzia di marketing (Non usa IaaS perché non ha competenze tecniche avanzate.)

### Usa PaaS per:

- Sito web clienti (Elastic Beanstalk)

### Usa SaaS per:

- Gestione clienti (Salesforce)
- Email marketing (Mailchimp)
- Analisi web (Google Analytics)
- Design (Canva, Figma)
- Archiviazione file (Dropbox)

# Riepilogo con Analogia del Trasporto

## IaaS = Noleggiare un'auto

- Ti danno l'auto (server)
- Tu la guidi dove vuoi
- Fai rifornimento tu
- Responsabilità della guida è tua

## PaaS = Usare un taxi

- Ti danno auto + autista (server + configurazione)
- Tu dici solo dove andare
- Guidano loro

## SaaS = Usare l'autobus

- Sali e scendi
- Percorso già deciso
- Paghi il biglietto e basta

## Esempio: Sito web piccolo

### Con IaaS (EC2):

- 20-50 euro/mese
- Ma serve: tuo tempo per configurare (valore 500-1000 euro), aggiornamenti continui

### Con PaaS (Elastic Beanstalk + RDS):

- 50-100 euro/mese
- Tutto gestito, nessun tempo sprecato

### Con SaaS (es. Shopify per e-commerce):

- 30-300 euro/mese a seconda del piano
- Zero competenze richieste, tutto incluso

## **FACCIAMO PRATICA**

### **Con Amazon AWS**

## 1. Creare un account AWS

Segui questi passaggi per registrarti su AWS:

1. Visita il sito di AWS: Vai alla pagina di registrazione di AWS: <https://aws.amazon.com/it/>
2. Fai clic su "Crea un account AWS": Nella parte superiore destra della pagina, troverai il pulsante "Crea un account AWS". Fai clic su di esso.
3. Inserisci un indirizzo email e crea una password: Inserisci l'indirizzo email che desideri associare all'account e crea una password sicura. Dovrai anche scegliere un nome utente dell'account AWS.
4. Inserisci le informazioni di contatto: AWS richiede informazioni di contatto personali o aziendali. Dovrai scegliere tra "Account professionale" o "Account personale", a seconda del tuo utilizzo.

# Cloud Computing

## 1. Creare un account AWS

5. Inserisci i dettagli di pagamento: Anche se AWS offre un'opzione gratuita, richiede di inserire una carta di credito o di debito valida per scopi di verifica. Non ti verrà addebitato alcun importo finché non superi i limiti del piano gratuito. Puoi anche impostare degli allarmi per essere avvisato in caso di consumo a pagamento.
6. Verifica la tua identità: AWS ti chiederà di verificare la tua identità, solitamente tramite una chiamata o un SMS. Dovrai inserire un codice di verifica che riceverai.
7. Scegli un piano di supporto: Dopo aver verificato l'identità, ti verrà chiesto di scegliere un piano di supporto. Puoi scegliere il piano "Basic Support", che è gratuito.
8. Accesso al portale AWS Management Console: Una volta completata la registrazione, potrai accedere alla AWS Management Console, il pannello di controllo da cui potrai gestire tutti i tuoi servizi AWS.

# Cloud Computing

## 2. Opzione gratuita (Free Tier)

L'opzione gratuita di AWS è chiamata AWS Free Tier, e offre tre tipi di vantaggi principali:

1. **Gratuità per 12 mesi:** Alcuni servizi sono gratuiti per 12 mesi a partire dalla data di registrazione. Ecco alcuni esempi:

- EC2 (Elastic Compute Cloud): 750 ore di utilizzo di una micro istanza t2.micro o t3.micro al mese.
- Amazon S3 (Simple Storage Service): 5 GB di spazio di archiviazione standard.
- RDS (Relational Database Service): 750 ore di utilizzo di un'istanza db.t2.micro o db.t3.micro.

2. **Sempre gratuiti:** Alcuni servizi sono gratuiti senza limiti temporali, ma con alcune restrizioni di utilizzo. Ecco alcuni esempi:

- AWS Lambda: 1 milione di richieste al mese gratuite.
- Amazon DynamoDB: 25 GB di spazio di archiviazione con throughput specifico gratuito.

## 2. Opzione gratuita (Free Tier)

Piano gratuito AWS					
Panoramica	Categorie del piano gratuito ▾	Come creare un account	Offerte in evidenza per le aziende ▾	Domande frequenti	Termini e condizioni
<b>▼ Tipo di piano</b> <ul style="list-style-type: none"> <li><input type="checkbox"/> In evidenza</li> <li><input type="checkbox"/> Gratis per 12 mesi</li> <li><input type="checkbox"/> Gratis senza limiti di tempo</li> <li><input type="checkbox"/> Prove gratuite</li> </ul> <b>▼ Categorie di prodotti</b> <ul style="list-style-type: none"> <li><input type="checkbox"/> Analisi dei dati</li> <li><input type="checkbox"/> Integrazione di applicazioni</li> <li><input type="checkbox"/> Produttività aziendale</li> <li><input type="checkbox"/> Calcolo</li> <li><input type="checkbox"/> Container</li> <li><input type="checkbox"/> Coinvolgimento dei clienti</li> <li><input type="checkbox"/> Database</li> <li><input type="checkbox"/> Strumenti per sviluppatori</li> <li><input type="checkbox"/> Elaborazione per utenti finali</li> <li><input type="checkbox"/> Front-end Web e di dispositivi mobili</li> <li><input type="checkbox"/> Game Tech</li> <li><input type="checkbox"/> Internet of Things</li> <li><input type="checkbox"/> Machine Learning</li> <li><input type="checkbox"/> Gestione e governance</li> <li><input type="checkbox"/> Servizi multimediali</li> <li><input type="checkbox"/> Migrazione e trasferimento</li> <li><input type="checkbox"/> Reti e distribuzione di contenuti</li> <li><input type="checkbox"/> Robotica</li> <li><input type="checkbox"/> Sicurezza, identità e conformità</li> <li><input type="checkbox"/> Serverless</li> <li><input type="checkbox"/> Storage</li> </ul>	<b>CALCOLO</b> Piano gratuito GRATIS PER 12 MESI <b>Amazon EC2</b> <b>750 ore</b> al mese Capacità di elaborazione ridimensionabile nel cloud. <small>750 ore al mese di istanze t2.micro o t3.micro</small>	<b>ARCHIVIAZIONE</b> Piano gratuito GRATIS PER 12 MESI <b>Amazon S3</b> <b>5 GB</b> di storage standard Infrastruttura di storage di oggetti sicura, durevole e scalabile. <small>5 GB di storage standard</small>	<b>DATABASE</b> Piano gratuito GRATIS PER 12 MESI <b>Amazon RDS</b> <b>750 ore</b> al mese di utilizzo del database (motori di database applicabili) Servizio di database relazionale gestito per MySQL, PostgreSQL, MariaDB o SQL Server.		
	<b>DATABASE</b> Piano gratuito GRATIS SENZA LIMITI DI TEMPO <b>Amazon DynamoDB</b> <b>25 GB</b> di storage Database NoSQL rapido e flessibile con scalabilità ottimale.	<b>MACHINE LEARNING</b> <span style="float: right;">NOVITÀ</span> Piano gratuito PROVA GRATUITA <b>Amazon SageMaker</b> <b>2 mesi</b> <small>prova gratuita</small> Machine learning per tutti gli sviluppatori e i data scientist <small>250 ore al mese di ml.t3.medium sui</small>		<b>CALCOLO</b> Piano gratuito GRATIS SENZA LIMITI DI TEMPO <b>AWS Lambda</b> <b>1 milione</b> <small>richieste gratuite al mese</small> Servizio di elaborazione che esegue codice in risposta a eventi e gestisce autonomamente le risorse di calcolo.	

## 3. Creazione di un budget

Una volta iscritti dalla barra di ricerca cerchiamo la voce budget, in questo modo andremo a configurare la piattaforma per avvisarci nel momento in cui andiamo a spendere oltre quanto abbiamo previsto

Gestione costi e fatturazione > Budget > Crea budget

### Configurazione budget

Utilizzo di un modello (semplificato)  
Utilizza le configurazioni consigliate. Puoi modificare alcune opzioni di configurazione dopo aver creato il budget.

Personalizzazione (avanzata)  
Personalizza un budget per impostare parametri specifici per il tuo caso d'uso. Puoi personalizzare la durata, il mese di inizio e account specifici.

#### Modelli - Nuovi

Scegli il modello che corrisponde meglio al tuo caso d'uso.

Budget a zero spesa  
Crea un budget che ti invia una notifica quando la spesa supera 0,01 USD.

Budget dei costi mensili  
Crea un budget mensile che invii una notifica se superi o stai per superare l'importo del budget.

Budget di copertura dei Savings Plans  
Crea un budget di copertura per i tuoi Savings Plans che ti invii una notifica quando sei al di sotto del target definito.

Budget di utilizzo giornaliero delle prenotazioni  
Crea un budget di utilizzo per le tue prenotazioni che ti invii una notifica quando sei al di sotto del target definito.

#### Budget dei costi mensili - Modello

**Nome budget**  
Specifica un nome descrittivo per il budget.

I nomi devono essere composti da un numero di caratteri compreso tra 1 e 100.

**Inserisci l'importo preventivo (\$)**  
Costo del mese scorso: 0,00 USD

Gestione costi e fatturazione

Scegli visualizzazione di fatturazione [Nuovo](#)

Scegli una visualizzazione...

Home

Guida introduttiva

Fatturazione e pagamenti

Fatture

Pagamenti

Crediti

Ordini di acquisto

Analisi dei costi e dell'utilizzo

Esploratore dei costi

Report salvati dall'Esploratore dei costi

Rilevamento di anomalie sui costi

Piano gratuito

Esportazioni di dati

# Cloud Computing

## Guida: Creare una Soglia di Consumo e Ricevere Alert per Email su AWS con CloudWatch

In questa guida vedremo come configurare un sistema di allerta basato su soglie di consumo dei servizi AWS utilizzando Amazon CloudWatch. CloudWatch consente di monitorare il consumo delle risorse AWS e impostare avvisi personalizzati tramite email per prevenire sorprese in fatturazione.

### Passaggi per Creare una Soglia di Consumo su AWS

#### 1. Accedi alla AWS Management Console

- Vai alla AWS Management Console e accedi con le tue credenziali.

#### 2. Accedi a Billing e Cost Management

- Una volta dentro la console, nella barra di ricerca in alto, digita "Billing" e seleziona

# Cloud Computing

Billing Dashboard.

## 3. Vai alla Sezione di Gestione dei Costi

- Nel menu a sinistra, seleziona Cost Management > Budgets (Budget).

## 4. Crea un Nuovo Budget

- Fai clic su Create budget.
- Scegli Cost budget per monitorare i costi totali e specifica un nome per il tuo budget.
- Imposta il limite di spesa che vuoi monitorare. Ad esempio, se non vuoi superare i 10\$ mensili, imposta questo come soglia.

## 5. Configura gli Avvisi (Alert)

- Scegli il tipo di alert che desideri ricevere:
  - Actual cost (Costo effettivo): Alert quando il costo effettivo supera una determinata soglia.
  - Forecasted cost (Costo previsto): Alert basato su una previsione di consumo, utile per prevenire superamenti futuri.
- Imposta la percentuale o l'importo del budget che, una volta raggiunto, attiverà l'invio dell'alert.

# Cloud Computing

## 6. Imposta l'Invio di Notifiche via Email

- Nella sezione **Notification settings**, inserisci l'indirizzo email dove desideri ricevere gli avvisi.
- Puoi inserire più indirizzi email separandoli con una virgola.

## 7. Rivedi e Conferma

- Rivedi tutte le impostazioni del budget e fai clic su **Create budget**.

### Configurare un Allarme con CloudWatch

Ora che hai impostato il budget, puoi anche configurare degli avvisi avanzati utilizzando CloudWatch per monitorare specifici servizi o metriche.

#### 1. Accedi a CloudWatch

- Nella AWS Management Console, cerca **CloudWatch** nella barra di ricerca e seleziona il servizio.

#### 2. Creare un Allarme

- Vai su **Alarms > Create Alarm**.

# Cloud Computing

## 3. Seleziona una Metrica

- Scegli una metrica da monitorare (es. utilizzo CPU, utilizzo di memoria, traffico di rete o metriche di utilizzo di servizi come EC2, S3 o Lambda).
- Se non sai quale metrica selezionare, AWS ha preset per il monitoraggio delle risorse più comuni.

## 4. Configura l'Allarme

- Imposta una soglia per la metrica selezionata, ad esempio se desideri ricevere un avviso quando l'utilizzo di CPU di un'istanza EC2 supera il 75%.
- Configura l'allarme su Threshold type come:
  - Static threshold per una soglia fissa.
  - Anomaly detection per rilevare anomalie rispetto a un comportamento atteso.

# Cloud Computing

## 5. Imposta l'Azione di Notifica

- Aggiungi un'azione di notifica: seleziona **Create a new topic** se non ne hai già uno configurato.
- Inserisci il nome del topic e l'indirizzo email dove vuoi ricevere le notifiche. AWS invierà una email di conferma.
- Conferma l'iscrizione cliccando sul link che ricevi via email.

## 6. Concludi la Creazione dell>Allarme

- Verifica tutte le impostazioni e fai clic su **Create Alarm**.

### Monitoraggio e Gestione degli Allarmi

- Una volta creato l'allarme, puoi visualizzare lo stato nella sezione **CloudWatch Alarms**.

- Gli stati degli allarmi possono essere:

- **OK:** La metrica monitorata è al di sotto della soglia.
- **ALARM:** La metrica ha superato la soglia impostata.
- **INSUFFICIENT\_DATA:** Non ci sono dati sufficienti per valutare l'allarme.

# Cloud Computing

## Identity and Access Management (IAM)

**Identity and Access Management (IAM)** è un servizio web di AWS che ci aiuta a controllare l'accesso alle risorse AWS in modo sicuro. Consente di creare e gestire utenti e gruppi, utilizzando autorizzazioni per consentire o negare l'accesso alle risorse AWS.

## Accesso al servizio IAM

Per iniziare, dobbiamo navigare e cliccare su **IAM**. Se non è visibile nel menu principale, è possibile cercarlo tra i servizi digitando "IAM" nella barra di ricerca. È consigliabile aggiungerlo ai preferiti, poiché si utilizza molto frequentemente.

## Protezione dell'utente root

La prima operazione fondamentale è proteggere l'utente root. Se siete collegati con l'utente root, una delle prime configurazioni di sicurezza è l'aggiunta della **Multi-Factor Authentication (MFA)**.

### Perché utilizzare MFA?

L'utente root controlla completamente il nostro account AWS, quindi è essenziale implementare un livello di sicurezza aggiuntivo attraverso l'MFA.

### Configurazione MFA

1. Dalla homepage di IAM, cliccare su **MFA**
2. Al primo accesso, apparirà il messaggio: "Non è stata assegnata la MFA. Come best practice di sicurezza, ti consigliamo di assegnare l'MFA"
3. Cliccare su **Assegna**

## Identity and Access Management (IAM)

**Identity and Access Management (IAM)** è un servizio web di AWS che ci aiuta a controllare l'accesso alle risorse AWS in modo sicuro. Consente di creare e gestire utenti e gruppi, utilizzando autorizzazioni per consentire o negare l'accesso alle risorse AWS.

## Accesso al servizio IAM

Per iniziare, dobbiamo navigare e cliccare su **IAM**. Se non è visibile nel menu principale, è possibile cercarlo tra i servizi digitando "IAM" nella barra di ricerca. È consigliabile aggiungerlo ai preferiti, poiché si utilizza molto frequentemente.

## Protezione dell'utente root

La prima operazione fondamentale è proteggere l'utente root. Se siete collegati con l'utente root, una delle prime configurazioni di sicurezza è l'aggiunta della **Multi-Factor Authentication (MFA)**.

### Perché utilizzare MFA?

L'utente root controlla completamente il nostro account AWS, quindi è essenziale implementare un livello di sicurezza aggiuntivo attraverso l'MFA.

### Configurazione MFA

1. Dalla homepage di IAM, cliccare su **MFA**
2. Al primo accesso, apparirà il messaggio: "Non è stata assegnata la MFA. Come best practice di sicurezza, ti consigliamo di assegnare l'MFA"
3. Cliccare su **Assegna**

**Attivata l'autenticazione a due fattori passiamo alla configurazione della nostra utenza**

## Identity and Access Management (IAM)

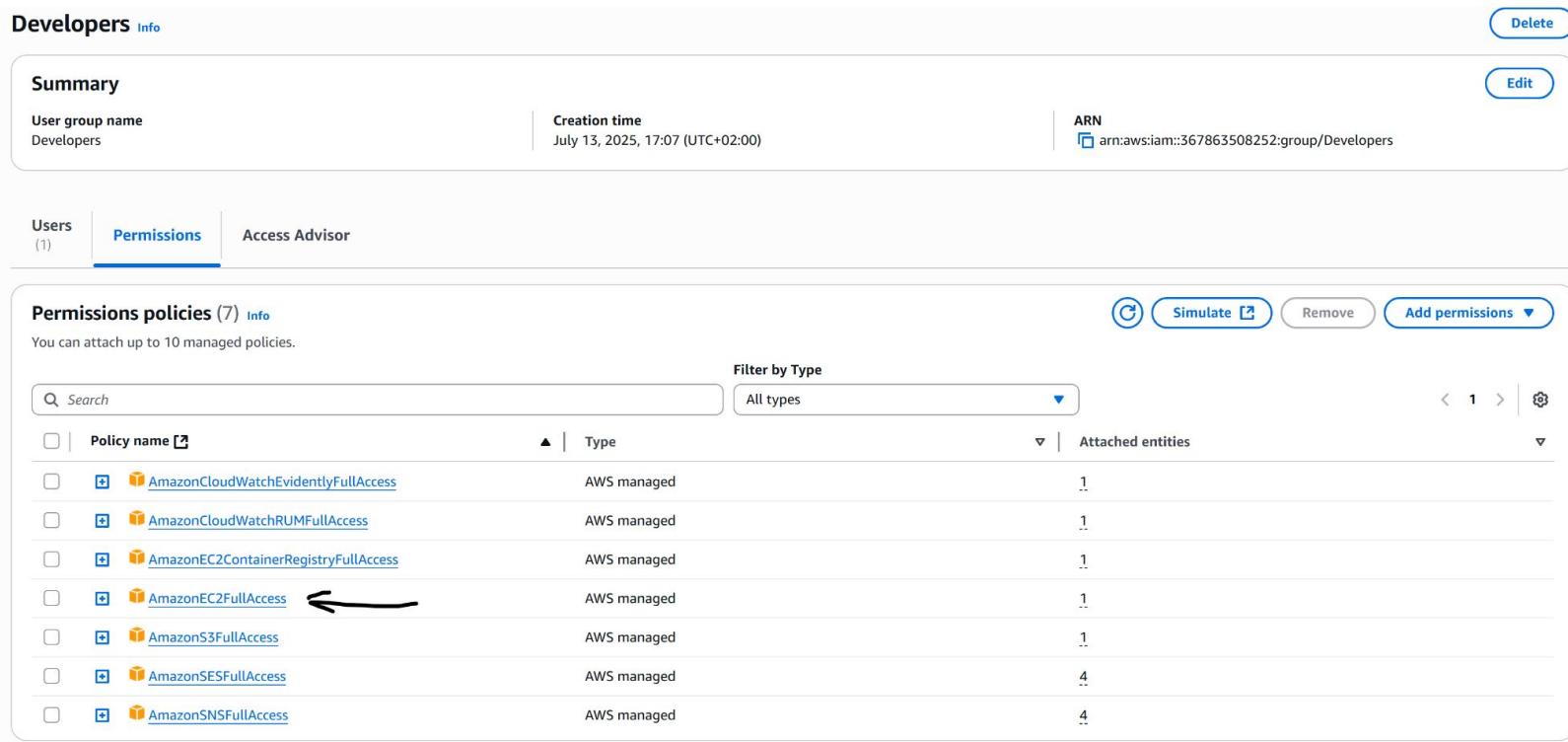
### Creazione del gruppo

Possiamo creare un gruppo a cui associamo i nostri utenti. In questo esempio lo chiamo Developers.

Abilitiamo il gruppo a determinati permessi.

Se per esempio fare un gruppo admins per tutti gli amministratori daremi full access come permessi.

Per i developers ho dato quelli dell'immagine



The screenshot shows the AWS IAM Groups page. At the top, there's a summary card for the 'Developers' group. Below it, there are tabs for 'Users' (1), 'Permissions', and 'Access Advisor'. The 'Permissions' tab is selected, showing a list of 7 managed policies attached to the group. A search bar and a 'Filter by Type' dropdown are at the top of the list. The policies listed are:

Policy name	Type	Attached entities
AmazonCloudWatchEvidentlyFullAccess	AWS managed	1
AmazonCloudWatchRUMFullAccess	AWS managed	1
AmazonEC2ContainerRegistryFullAccess	AWS managed	1
AmazonEC2FullAccess	AWS managed	1
AmazonS3FullAccess	AWS managed	1
AmazonSESFullAccess	AWS managed	4
AmazonSNSSFullAccess	AWS managed	4

An arrow points to the 'AmazonEC2FullAccess' policy in the list.

## Identity and Access Management (IAM)

### Creazione dell'utente

Creiamo ora un utente che assoceremo al gruppo precedentemente creato. In fase di creare utente avremo modo di scegliere se associare l'utente al gruppo, non è necessario. Se lo facciamo avrà le stesse policy del gruppo altrimenti possiamo associarle singolarmente

IAM > Users > Create user

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

Step 4: Retrieve password

### Specify user details

**User details**

User name: Roberto\_ITS

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ \_ - (hyphen)

Provide user access to the AWS Management Console - optional

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

**Are you providing console access to a person?**

User type:

Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

**Console password**

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

.....

Show password

Users must create a new password at next sign-in - Recommended

Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

**If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user.**

[Learn more](#)

Cancel Next

## Identity and Access Management (IAM)

### Set dei permessi nella creazione utente

- Step 1  
Specify user details
- Step 2  
Set permissions
- Step 3  
Review and create
- Step 4  
Retrieve password

#### Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

##### Permissions options

###### Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

###### Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

###### Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

#### User groups (1/1)

Search

Group name

Users

Attached policies

Created

Developers

0

[AmazonEC2ContainerRegistryFullAccess,...](#)

2025-07-13 (11 minutes ago)

[Create group](#)

1

##### ► Set permissions boundary - optional

[Cancel](#)

[Previous](#)

[Next](#)

## Identity and Access Management (IAM)

### Possibile aggiunta di Tag

Step 1  
Specify user details  
Step 2  
Set permissions  
Step 3  
**Review and create**  
Step 4  
Retrieve password

**Review and create**  
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details		Console password type	Require password reset
User name	Roberto_ITS	Custom password	Yes

**Permissions summary**

Name	Type	Used as
<a href="#">Developers</a>	Group	Permissions group
<a href="#">IAMUserChangePassword</a>	AWS managed	Permissions policy

**Tags - optional**  
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

Key	Value - optional	Remove
<input type="text" value="Enter key"/> <small>(*) You must specify a tag key</small>	<input type="text" value="Enter value"/>	<a href="#">Remove</a>

[Add new tag](#)  
You can add up to 49 more tags.

[Cancel](#) [Previous](#) [Create user](#)

# Cloud Computing

## Identity and Access Management (IAM)

### Utente creato

 User created successfully

[View user](#)



You can view and download the user's password and email instructions for signing in to the AWS Management Console.

- Step 1  
 Specify user details
- Step 2  
 Set permissions
- Step 3  
 Review and create
- Step 4  
 **Retrieve password**

### Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

#### Console sign-in details

[Email sign-in instructions](#)

##### Console sign-in URL

 <https://367863508252.signin.aws.amazon.com/console>

##### User name

 Roberto\_ITS

##### Console password

 \*\*\*\*\* [Show](#)

[Cancel](#)

[Download .csv file](#)

[Return to users list](#)

# Cloud Computing

## Creiamo un'istanza EC2

### Introduzione a EC2

**Amazon EC2 (Elastic Compute Cloud)** è un servizio web che fornisce capacità di elaborazione scalabile nel cloud di AWS. EC2 ti permette di creare e gestire **server virtuali**, chiamati **istanze**, con controllo completo sul sistema operativo e sulla configurazione.

### Cos'è un'istanza EC2?

Amazon Elastic Compute Cloud (Amazon EC2) è un servizio web che fornisce capacità di elaborazione scalabile nel cloud di Amazon Web Services (AWS). EC2 offre la possibilità di eseguire istanze virtuali, noti come server virtuali, in un ambiente cloud, consentendo agli utenti di avere un controllo completo su questi sistemi.

Questo servizio è stato progettato per semplificare la fornitura di scalabilità per applicazioni web e servizi associati, eliminando la necessità di investire in hardware fisico e semplificando la gestione delle risorse di elaborazione. Una delle principali attrazioni di EC2 è la sua flessibilità. Gli utenti possono scegliere tra diversi tipi di istanze, ognuna ottimizzata per specifici casi d'uso, come elaborazione ad alte prestazioni, storage ottimizzato, o applicazioni sensibili alla latenza.

Oltre alla flessibilità, EC2 offre una serie di servizi e funzioni che facilitano la gestione del ciclo di vita delle applicazioni, inclusi strumenti per la configurazione, il monitoraggio e la scalabilità. In breve, EC2 è un componente essenziale per qualsiasi azienda che voglia approfittare della potenza e della versatilità del cloud computing.

# Cloud Computing

Una delle principali attrazioni di EC2 è la sua flessibilità.

Gli utenti possono scegliere tra diversi tipi di istanze, ognuna ottimizzata per specifici casi d'uso, come elaborazione ad alte prestazioni, storage ottimizzato, o applicazioni sensibili alla latenza.

Oltre alla flessibilità, EC2 offre una serie di servizi e funzioni che facilitano la gestione del ciclo di vita delle applicazioni, inclusi strumenti per la configurazione, il monitoraggio e la scalabilità.

In breve, EC2 è un componente essenziale per qualsiasi azienda che voglia approfittare della potenza e della versatilità del cloud computing.

## Flessibilità e Personalizzazione

- Oltre 600 tipi di istanze diverse disponibili
- Possibilità di scegliere processori, memoria, storage e networking ottimali per il tuo caso d'uso
- Supporto per diversi sistemi operativi (Linux, Windows, macOS)

## Scalabilità

- Possibilità di scalare verticalmente (cambiare tipo di istanza)
- Possibilità di scalare orizzontalmente (aumentare il numero di istanze)
- Auto Scaling automatico in base al carico di lavoro

## Creiamo un istanza EC2

### Riduzione dei Costi

- Paghi solo per le risorse che utilizzi (On-Demand)
- Opzione di istanze riservate per risparmiare a lungo termine
- Istanze Spot per utilizzare capacità non usata a prezzi ridotti

### Sicurezza

- Security Groups per controllare il traffico in entrata e uscita
- Integrazione con altri servizi AWS (S3, RDS, VPC)
- Gruppi di sicurezza che agiscono come firewall virtuali

### Velocità e Provisioning Rapido

- Creazione di nuove istanze in pochi minuti
- Elimina la necessità di hardware fisico e manutenzione
- Deployment in diverse regioni globali per ottimizzare le prestazioni

### Monitoraggio

- Amazon CloudWatch per monitorare prestazioni e metriche
- Allarmi automatici per eventi specifici
- Dashboard per visualizzare lo stato delle risorse

# Tipi di istanze EC2

**Amazon EC2 offre diverse famiglie di istanze, ognuna ottimizzata per specifici casi d'uso:**

Famiglia	Uso Ideale	Esempio	Caratteristiche
<b>General Purpose</b>	Applicazioni web, server backend	t2.micro, t3.micro, m5.large	Bilanciamento tra CPU, memoria e networking
Compute Optimized	Calcolo ad alte prestazioni	c5.4xlarge	CPU potenti per applicazioni intensive
Memory Optimized	Database, caching	r5.2xlarge	Memoria RAM elevata
Storage Optimized	Elaborazione big data	i3en.large	Accesso veloce a grandi dataset
GPU Instances	Machine Learning, grafica	p3.8xlarge	GPU per calcolo parallelo

## Come Scegliere il Tipo di Istanza

- Analizza il carico di lavoro:** Identifica se la tua applicazione richiede più CPU, memoria o storage
- Bilancia costo e performance:** Considera il budget disponibile
- Considera la scalabilità futura:** Prevedi la crescita dell'applicazione

# Creazione di un'istanza EC2

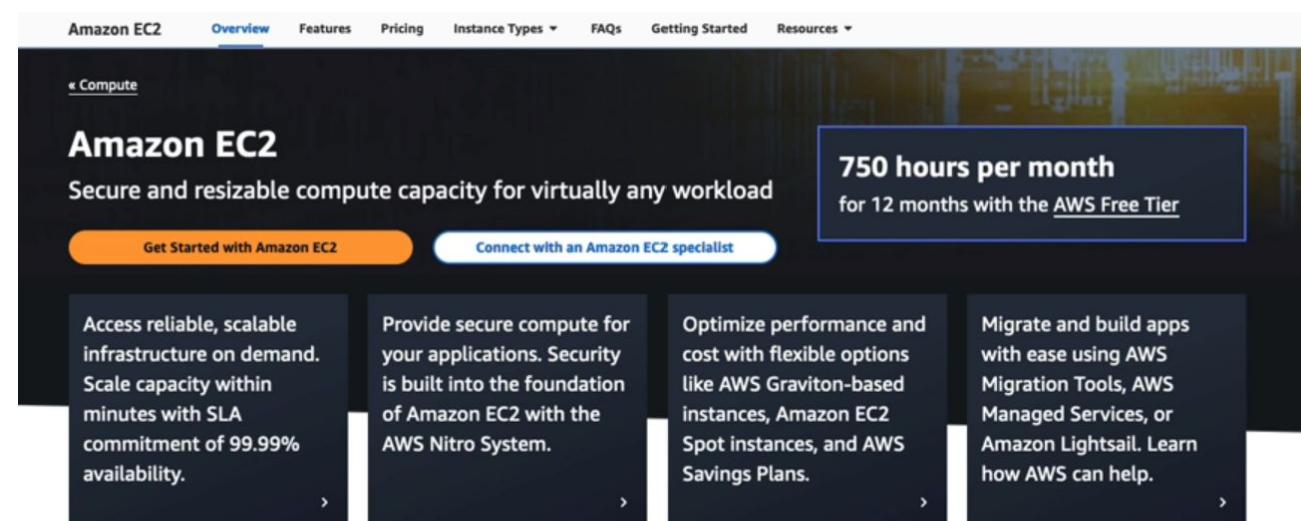
## Passo 1: Accesso al Servizio EC2

1. Accedi alla **AWS Management Console**
2. Nella barra di ricerca, digita **EC2**
3. Seleziona il servizio EC2 dai risultati
4. **Consiglio:** Aggiungi EC2 ai preferiti (cliccando sulla stellina) per accesso rapido

## Passo 2: Panoramica della Console EC2

Nella dashboard EC2 troverai:

- **Istanze in esecuzione:** Server attualmente attivi
- **IP elastici:** Indirizzi IP pubblici assegnati
- **Volumi:** Dischi collegati alle istanze
- **Gruppi di Auto Scaling:** Configurazioni per scalabilità automatica
- **Regione corrente:** Es. Europe (Frankfurt) - EU-Central-1



Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instances and choice of the latest processor, storage, networking, operating system, and purchase model to help you best match the needs of your workload. We are the first major cloud provider that supports Intel, AMD, and Arm processors, the only cloud with on-demand EC2 Mac instances, and the only cloud with 400 Gbps Ethernet networking. We offer the best price performance for machine learning training, as well as the lowest cost per inference instances in the cloud. More SAP, high performance computing (HPC), ML, and Windows workloads run on AWS than any other cloud.

# Creazione di un'istanza EC2

## Passo 3: Selezione della Regione

### Regioni disponibili in Europa:

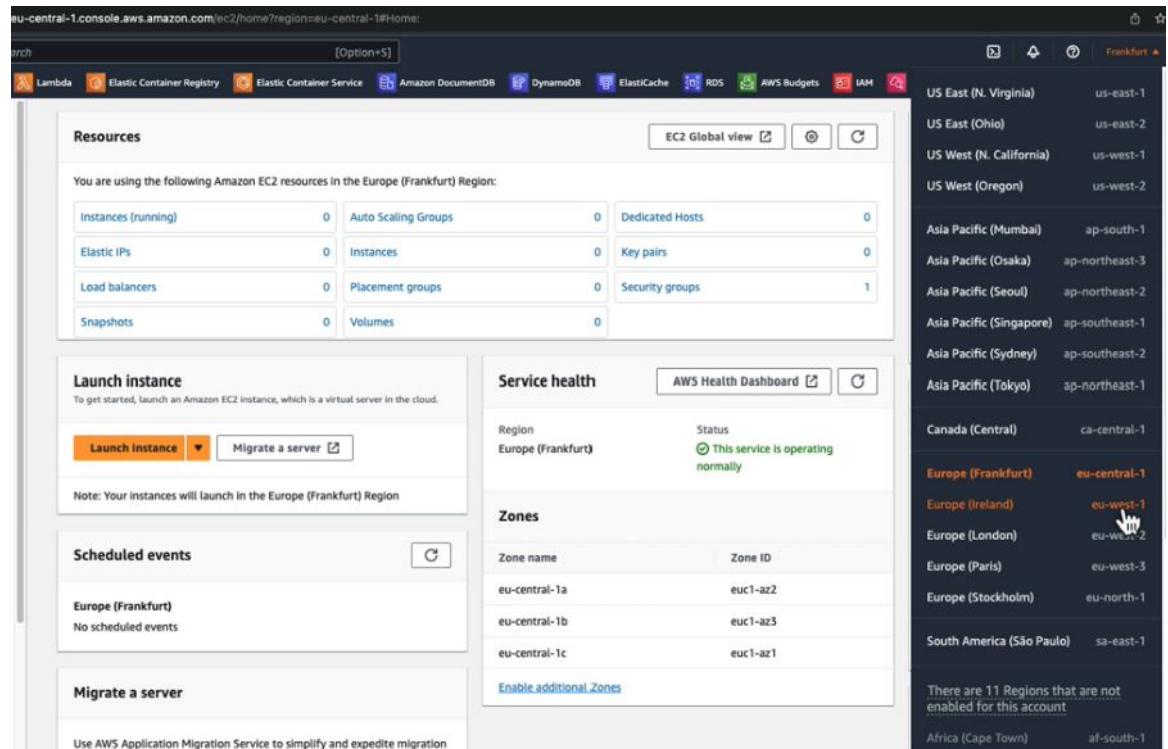
- Frankfurt (EU-Central-1)
- Irlanda (EU-West-1)
- Londra (EU-West-2)
- Parigi (EU-West-3)
- Stoccolma (EU-North-1)
- Milano, Madrid, Zurigo (disponibilità dipende dall'account)

**Importante:** Scegli sempre la regione più vicina ai tuoi utenti per ridurre la latenza.

## Passo 4: Avvio dell'Istanza

1. Clicca su **Launch Instance** (Avvia istanza)
2. Puoi anche usare il menu laterale: **Instances → Instances → Launch Instance**

**Opzione Template:** Puoi creare istanze da template preconfigurati con tutte le caratteristiche già impostate.



The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a search bar and a navigation bar with links like Lambda, Elastic Container Registry, Elastic Container Service, Amazon DocumentDB, DynamoDB, ElastiCache, RDS, AWS Budgets, IAM, and CloudWatch Metrics. Below the navigation bar, there's a section titled "Metrics" which lists various metrics grouped by category: CloudWatch Metrics, Lambda, and CloudWatch Metrics. Under each category, there are several metrics with their names and current values. For example, under "CloudWatch Metrics", there are metrics like "CloudWatch Metrics Metrics", "CloudWatch Metrics Metrics", "CloudWatch Metrics Metrics", and "CloudWatch Metrics Metrics". The overall interface is clean and modern, typical of the AWS web console.



The screenshot shows the AWS EC2 Instances page. At the top, there's a navigation bar with links for EC2, Elastic Beanstalk, Lambda, Elastic Container Registry, Elastic Container Service, Amazon DocumentDB, DynamoDB, ElastiCache, RDS, AWS Budgets, IAM, CloudWatch Metrics, and Billing. Below the navigation bar, there's a search bar and a "Find instance by attribute or tag (case-sensitive)" input field. The main area is titled "Instances" and shows a table with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv6 DNS. A message at the bottom says "No instances. You do not have any instances in this region." There are also buttons for "Launch Instances", "Launch instance from template", and "Migrate a server". The left sidebar has a "New EC2 Experience" button and links for EC2 Dashboard, EC2 Global View, Events, Instances, Instances, and Instance Types.

# Creazione di un'istanza EC2

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags Info

Name

apache-webserver

Add additional tags

### ▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below



Q Search our full catalog including 1000s of application and OS images

### Quick Start



Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

### Amazon Machine Image (AMI)

#### Amazon Linux 2023 AMI

ami-0766f68f0b06ab145 (64-bit (x86)) / ami-0d84a3c966e10e500 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

### Description

Q apache

Q

#### t2.nano

Family: t2 1 vCPU 0.5 GiB Memory Current generation: true  
On-Demand SUSE base pricing: 0.0067 USD per Hour  
On-Demand Linux base pricing: 0.0067 USD per Hour  
On-Demand Windows base pricing: 0.009 USD per Hour

Free tier eligible

#### t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.018 USD per Hour  
On-Demand SUSE base pricing: 0.0134 USD per Hour  
On-Demand Linux base pricing: 0.0134 USD per Hour  
On-Demand RHEL base pricing: 0.0734 USD per Hour

#### t2.small

Family: t2 1 vCPU 2 GiB Memory Current generation: true  
On-Demand SUSE base pricing: 0.0568 USD per Hour  
On-Demand Linux base pricing: 0.0268 USD per Hour  
On-Demand Windows base pricing: 0.036 USD per Hour  
On-Demand RHEL base pricing: 0.0868 USD per Hour

#### t2.medium

Family: t2 2 vCPU 4 GiB Memory Current generation: true  
On-Demand RHEL base pricing: 0.1136 USD per Hour  
On-Demand Linux base pricing: 0.0536 USD per Hour  
On-Demand SUSE base pricing: 0.1536 USD per Hour  
On-Demand Windows base pricing: 0.0716 USD per Hour

#### t2.large

Family: t2 2 vCPU 8 GiB Memory Current generation: true  
On-Demand RHEL base pricing: 0.1673 USD per Hour

Free tier eligible

#### t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.018 USD per Hour  
On-Demand SUSE base pricing: 0.0134 USD per Hour  
On-Demand Linux base pricing: 0.0134 USD per Hour  
On-Demand RHEL base pricing: 0.0734 USD per Hour

Additional costs apply for AMIs with pre-installed software

X



Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

device type ENA Enabled

Yes

All generations

Compare instance types

# Creazione di un'istanza EC2

## Passo 5: Configurazione di Base

### Nome dell'Istanza

Assegna un nome descrittivo, ad esempio:

- Apache-Web-Server
- Database-Production
- App-Backend

### Selezione dell'Immagine (AMI - Amazon Machine Image)

Un'**AMI** è un template che contiene:

- Sistema operativo
- Server applicativi preinstallati
- Software e configurazioni

### Opzioni disponibili:

#### Immagini AWS Marketplace

- Oltre 1000 immagini di sistemi operativi
- Template preconfigurati con software già installato (Apache, PHP, MySQL, ecc.)
- Filtri per tipo di sistema operativo e piattaforma

#### Immagini Popolari:

- **Amazon Linux 2023**: Ottimizzata per AWS, gratuita nel Free Tier
- **Ubuntu Server**: Popolare distribuzione Linux
- **Windows Server**: Per applicazioni Windows
- **Red Hat Enterprise Linux**: Per ambienti enterprise
- **macOS**: Per sviluppo Apple (limitato)

**Free Tier Eligible:** Cerca le immagini contrassegnate come idonee per il livello gratuito

# Creazione di un'istanza EC2

## Tipo di Istanza

Seleziona le risorse hardware:

### Per il Free Tier:

- **t2.micro o t3.micro:** 1 vCPU, 1 GB RAM
- Idonee per 750 ore/mese gratuite nel primo anno

### Caratteristiche da considerare:

- **vCPU:** Numero di processori virtuali
- **Memoria:** RAM disponibile
- **Network Performance:** Velocità di rete
- **Storage:** Tipo e dimensione del disco

**Importante:** Puoi cambiare il tipo di istanza in seguito per scalare verticalmente.

## Passo 6: Configurazione delle Chiavi di Accesso (Key Pair)

Le chiavi SSH sono necessarie per accedere all'istanza in modo sicuro.

### Creare una Nuova Chiave:

1. Clicca su **Create new key pair**
2. Scegli il tipo: **RSA** (raccomandato)
3. Formato file:
  - **.pem:** Per Linux, macOS e Windows 10+
  - **.ppk:** Per Windows con PuTTY
4. Assegna un nome: es. **Apache-KeyPair**
5. Clicca **Create key pair**
6. **IMPORTANTE:** Scarica e conserva il file **.pem** in modo sicuro - AWS non può rigenerarlo

### Utilizzare una Chiave Esistente:

- Seleziona una chiave dal menu a discesa se ne hai già una

**▼ Key pair (login) Info**

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Select



**Create key pair**

Key pair name  
Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type  
 RSA RSA encrypted private and public key pair  
 ED25519 ED25519 encrypted private and public key pair

Private key file format  
 .pem For use with OpenSSH  
 .ppk For use with PuTTY

**⚠️** When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel
Create key pair

# Creazione di un'istanza EC2

## Passo 7: Configurazione di Rete (Network Settings)

### Virtual Private Cloud (VPC)

AWS crea automaticamente una **VPC** (rete privata virtuale) per isolare le tue risorse.

#### Elementi di rete:

- VPC ID:** Identificativo della rete
- Subnet:** Sotto-rete all'interno della VPC
- IP pubblico:** Per accesso da Internet
- IP privato:** Per comunicazione interna

### Security Groups (Gruppi di Sicurezza)

I Security Groups funzionano come **firewall virtuali** che controllano il traffico in entrata e uscita.

#### Regole di Traffico In Entrata (Inbound Rules):

- SSH (porta 22)**
  - Per accesso remoto al server
  - Consiglio di sicurezza:** Limita l'accesso al tuo IP ([My IP](#))
  - Evita `0.0.0.0/0` (accesso da qualsiasi IP) se non necessario
- HTTP (porta 80)**
  - Per traffico web non crittografato
  - Necessario per server web pubblici
- HTTPS (porta 443)**
  - Per traffico web crittografato (SSL/TLS)
  - Richiede certificati SSL configurati

**Key pair (login)**

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Create new key pair

**Network settings**

Network [Info](#)  
vpc-0fb96135ec25009e4

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  
Enable

Firewall (security groups) [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group     Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from [Helps you connect to your instance](#)  
Anywhere `0.0.0.0/0`

Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

**Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

# Creazione di un'istanza EC2

upload a file with your user data or enter it in the field.

**Network settings** [Info](#)

[Edit](#)

Network [Info](#)  
vpc-0fb96135ec25009e4

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  
Enable

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group     Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from Anywhere  
Helps you connect to your instance  
0.0.0.0/0

Allow HTTPS traffic from the Internet  
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the Internet  
To set up an endpoint, for example when creating a web server

**⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.**

#!/bin/bash  
yum update -y  
yum install -y httpd mariadb-server php php-mysqlnd  
systemctl start httpd  
systemctl enable httpd  
systemctl start mariadb  
systemctl enable mariadb

tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Launch instance](#)  [Review commands](#)

# Creazione di un'istanza EC2

## Passo 8: Configurazione dello Storage

### Volume Root (Principale)

#### Configurazione predefinita:

- **Dimensione:** 8 GB (Free Tier include fino a 30 GB)
- **Tipo:** gp3 (General Purpose SSD)
- **Velocità:** Performance bilanciate
- **Costo:** Incluso nel Free Tier (30 GB)

#### Tipi di Volume EBS:

- **gp3/gp2 (General Purpose SSD):** Uso generale, bilanciato
- **io2/io1 (Provisioned IOPS SSD):** Performance garantite, più costoso
- **st1 (Throughput Optimized HDD):** Per dati sequenziali
- **sc1 (Cold HDD):** Per dati ad accesso infrequente

#### Aggiungere Volumi Aggiuntivi

Puoi separare sistema operativo e dati:

- **Volume 1:** 8 GB per OS e applicazioni
- **Volume 2:** Dimensione personalizzata per dati

#### Per aggiungere un volume:

1. Clicca **Add New Volume**
2. Seleziona tipo e dimensione
3. Configura opzioni di crittografia se necessario

# Creazione di un'istanza EC2

Screenshot of the AWS EC2 Instances page showing a single running instance (i-051b06eefc8fa9344) of type t2.micro.

The instance summary details include:

- Instance ID: i-051b06eefc8fa9344
- Public IPv4 address: 18.193.102.84 | open address
- Private IPv4 address: 172.31.16.10
- Public IPv4 DNS: ec2-18-193-102-16. eu-central-1.compute.amazonaws.com | open
- Private IP DNS name (IPv4 only): ip-172-31-16-10.eu-central-1.compute.internal
- Instance state: Running
- Instance type: t2.micro
- VPC ID: vpc-0fb96135ec25009e4
- Subnet ID: subnet-050cb2fa71a6b129a
- Elastic IP addresses: -
- AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
- Auto Scaling Group name: -

The screenshot also shows a large callout arrow pointing from the Public IPv4 address section towards the Private IPv4 address section.

# Creazione di un'istanza EC2

**Connect to instance** Info

Connect to your instance i-051b06eefc8fa9344 using any of these options

**EC2 Instance Connect**  Session Manager  SSH client  EC2 serial console

**Instance ID**

**Connection Type**

**Connect using EC2 Instance Connect**  
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

**Connect using EC2 Instance Connect Endpoint**  
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

**Public IP address**

**User name**  
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

**Note:** In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.



AWS Services Search [Option+S]

EC2 Elastic Beanstalk Lambda Elastic Container Registry Elastic Container Service Amazon DocumentDB Dyn

Amazon Linux 2023  
https://aws.amazon.com/linux/amazon-linux-2023

```
[ec2-user@ip-172-31-16-10 ~]$ pw
```