

Fondamenti di Cloud Computing e Setup PaaS

Introduzione al Cloud Computing e Configurazione Ambiente PythonAnywhere

Autore: Elena Pagnacco

Sintesi

Il presente documento riporta l'analisi teorica dei principali concetti legati al **Cloud Computing** e documenta l'attività pratica di configurazione di un ambiente di sviluppo in cloud. Nella prima fase è stata effettuata una ricerca comparativa sui principali **Cloud Service Provider (CSP)** del mercato e sono stati definiti i modelli di servizio standard (**IaaS, PaaS, SaaS**). Successivamente, è stata eseguita la registrazione e l'attivazione di un account presso **PythonAnywhere**, identificato come caso d'uso pratico di una soluzione PaaS (Platform as a Service), documentando i passaggi necessari per l'accesso alla dashboard di sviluppo.

Obiettivi

L'attività si prefigge i seguenti obiettivi didattici:

- Comprendere il panorama attuale dei fornitori di servizi Cloud.
 - Distinguere le differenze, i vantaggi e i casi d'uso dei modelli IaaS, PaaS e SaaS.
 - Acquisire familiarità con la procedura di registrazione e primo accesso a un servizio Cloud (PythonAnywhere).
-

Analisi Teorica: I Main Player del Cloud

In questa sezione vengono analizzati i tre principali fornitori di servizi cloud a livello globale.

Amazon Web Services (AWS)

AWS è la piattaforma cloud più completa e ampiamente adottata al mondo. Offre oltre 200 servizi completi dai data center a livello globale.

- **Caratteristiche Principali:** Dominanza di mercato, vastissima gamma di servizi (dal calcolo puro come EC2 ai servizi di AI/ML), elevata affidabilità e scalabilità. È spesso la scelta predefinita per le grandi imprese che necessitano di infrastrutture complesse.

Microsoft Azure

Azure è la piattaforma di cloud computing di Microsoft. È in forte crescita e compete direttamente con AWS.

- **Caratteristiche Principali:** Integrazione nativa con l'ecosistema Microsoft (Windows Server, Active Directory, Office 365), forte presenza nel cloud ibrido (connessione tra data center on-premise e cloud) e strumenti avanzati per sviluppatori .NET.

Google Cloud Platform (GCP)

Offerta da Google, questa piattaforma sfrutta la stessa infrastruttura che Google utilizza per i suoi prodotti end-user (come Search e YouTube).

- **Caratteristiche Principali:** Eccellenza nell'analisi dei dati (Big Data), Machine Learning e gestione dei container (Kubernetes è stato creato da Google). È spesso scelta per progetti ad alto contenuto innovativo e open source.
-

Analisi dei Modelli di Servizio Cloud

Il cloud computing viene erogato secondo tre modelli fondamentali, che differiscono per il livello di controllo e gestione lasciato all'utente rispetto al provider.

IaaS (Infrastructure as a Service)

In questo modello, il provider fornisce l'infrastruttura di base: server virtuali, storage, reti e firewall. L'utente gestisce il sistema operativo e le applicazioni.

- **Esempio:** Amazon EC2 (Elastic Compute Cloud), Microsoft Azure Virtual Machines.
- **Vantaggi:** Massimo controllo sulla configurazione del software e del sistema operativo; ideale per migrazioni "lift-and-shift" di applicazioni esistenti.

PaaS (Platform as a Service)

Il provider fornisce una piattaforma che include hardware, sistema operativo, database e ambienti di esecuzione (runtime). L'utente si concentra solo sullo sviluppo e sul deploy del codice.

- **Esempio:** Google App Engine, Heroku, **PythonAnywhere**.
- **Vantaggi:** Riduzione drastica dei tempi di gestione sistemistica; gli sviluppatori possono rilasciare codice senza preoccuparsi degli aggiornamenti del server o della sicurezza del sistema operativo.

SaaS (Software as a Service)

Il provider gestisce tutto, dall'infrastruttura all'applicazione finale. L'utente accede al software via internet, solitamente tramite browser.

- **Esempio:** Google Workspace (Gmail, Drive), Microsoft 365, Salesforce, Dropbox.
 - **Vantaggi:** Accessibilità immediata ovunque, nessun bisogno di installazione o manutenzione locale, aggiornamenti automatici gestiti dal fornitore.
-

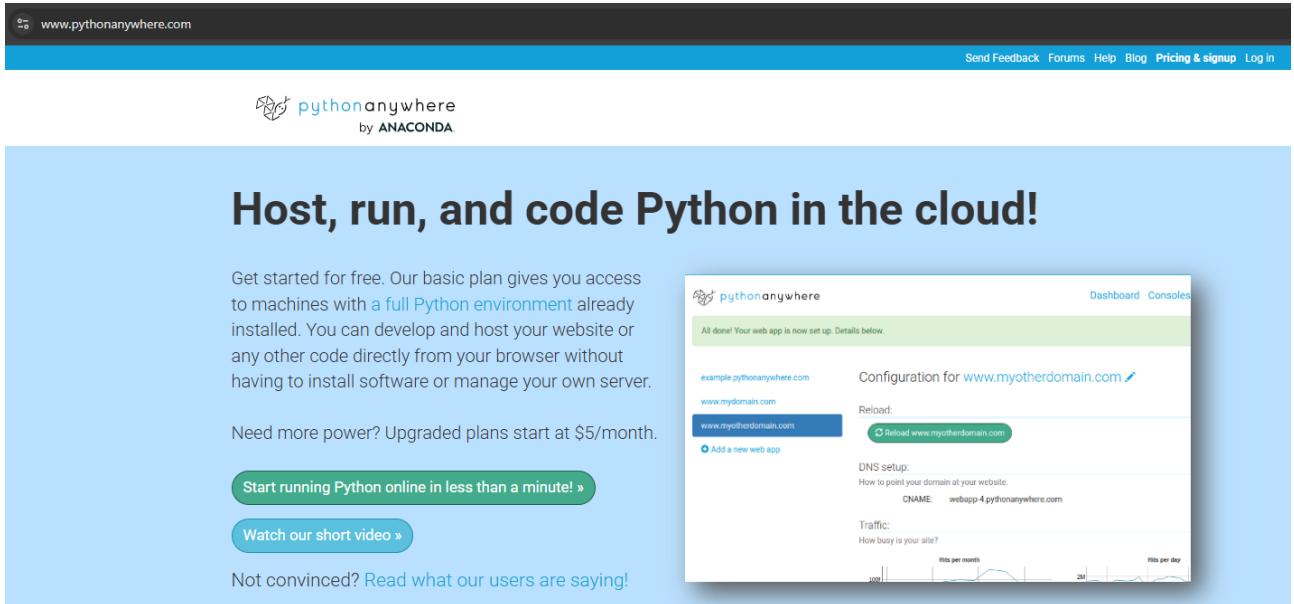
Attività Pratica: Registrazione su PythonAnywhere

Per applicare i concetti di **PaaS (Platform as a Service)**, è stato scelto il servizio PythonAnywhere. Questo servizio permette di scrivere ed eseguire codice Python direttamente dal browser, senza dover configurare server web o sistemi operativi Linux.

Procedura di Configurazione

Passo 1: Accesso al portale

Mi sono collegata al sito ufficiale www.pythonlywhere.com. La homepage presenta chiaramente l'offerta del servizio focalizzata sulla semplicità di utilizzo per sviluppatori Python.

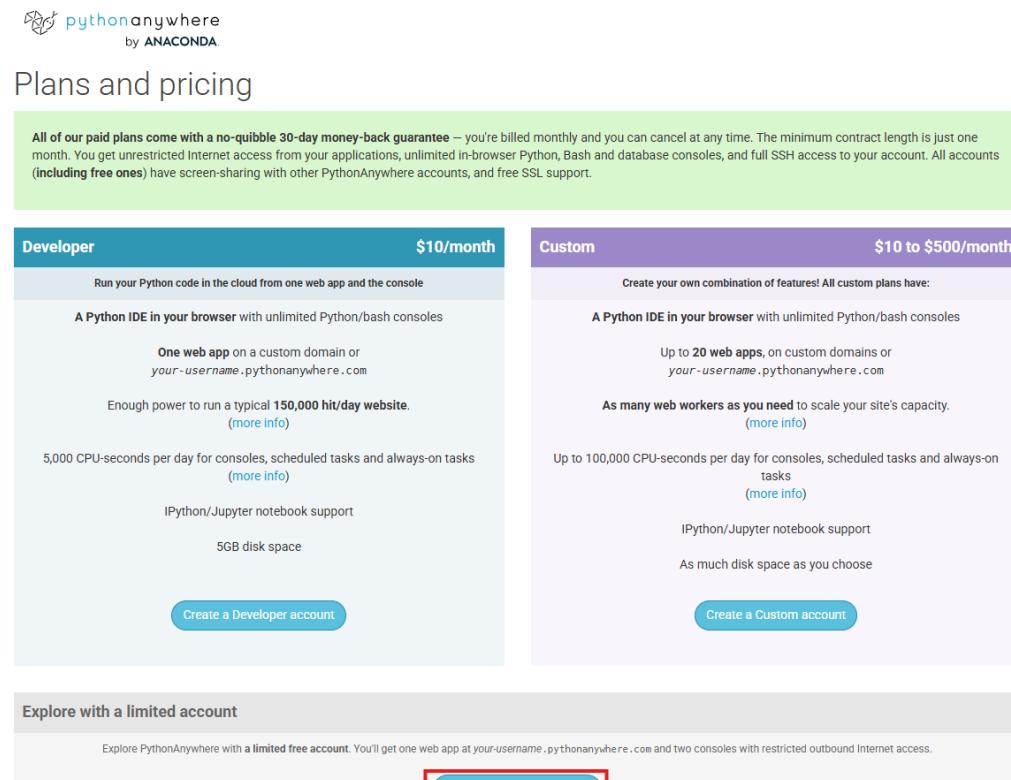


The screenshot shows the PythonAnywhere homepage. At the top, there's a navigation bar with links for 'Send Feedback', 'Forums', 'Help', 'Blog', 'Pricing & signup', and 'Log in'. Below the navigation is the PythonAnywhere logo and a tagline 'by ANACONDA.'. The main heading 'Host, run, and code Python in the cloud!' is prominently displayed. To the left, there's a section about getting started for free, mentioning a basic plan with a full Python environment. Below that, a message says 'Need more power? Upgraded plans start at \$5/month.' There are two buttons: 'Start running Python online in less than a minute!' and 'Watch our short video ›'. At the bottom, a link reads 'Not convinced? Read what our users are saying!'. On the right, there's a configuration panel for a custom domain 'www.myotherdomain.com'. It shows the CNAME entry as 'webapp-4.pythonlywhere.com'. Below this is a traffic graph showing hits per month and hits per day.

Figura 1 Homepage del sito PythonAnywhere.

Passo 2: Selezione del piano

Ho cliccato su "Pricing & Signup" e successivamente sul pulsante "Create a Beginner account". Questo piano è gratuito ("Free") e permette di avere una console limitata e un'applicazione web, ideale per scopi didattici.



The screenshot shows the 'Plans and pricing' section. It highlights the 'Developer' plan, which costs \$10/month. The plan includes a Python IDE in the browser, unlimited Python/bash consoles, one web app on a custom domain, and 5,000 CPU-seconds per day. A 'Create a Developer account' button is shown. Next to it is the 'Custom' plan, which starts at \$10 to \$500/month and allows creating your own combination of features. It includes up to 20 web apps, as many web workers as needed, and as much disk space as you choose. A 'Create a Custom account' button is shown. At the bottom, there's a section for exploring with a limited account, which offers a free account for one web app and two restricted consoles. A 'Create a Beginner account' button is highlighted with a red border.

Figura 2 Selezione del piano gratuito Beginner.

Passo 3: Creazione dell'account

Nel modulo di registrazione ho inserito i seguenti dati:

- **Username:** (Scelto un identificativo univoco)
- **Email:** (Indirizzo email valido)
- **Password:** (Password complessa secondo le best practice di sicurezza) Dopo aver accettato i termini di servizio, ho proceduto con la registrazione.

The screenshot shows the PythonAnywhere registration page. At the top, there's a logo for PythonAnywhere by ANACONDA. Below it, a message says "Welcome to www.pythonanywhere.com! It looks like you're in Europe. You're quite welcome to sign up here on our US-hosted system, but if you want complete peace of mind about the location of any data you store on our servers – especially if you're planning to store personal data and need GDPR compliance – you can try our EU-hosted site instead. The servers are all in Germany, and so is all of your data. Also, for paid accounts, prices are in euros :-)"

Create your account

Username:

Email: @gmail.com

Password:

Password (again):

I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old.

Register

We promise not to spam or pass your details on to anyone else.

Figura 3 Form di registrazione utente.

Passo 4: Accesso alla Dashboard

Una volta completata la registrazione, e verificata la mail, il sistema ha effettuato il reindirizzamento automatico alla **Dashboard** principale. Da qui è possibile visualizzare le console attive (Bash, Python), i file e le applicazioni web.

The screenshot shows the PythonAnywhere dashboard. At the top left is the PythonAnywhere logo with the text "pythonanywhere" and "by ANACONDA". At the top right are links for "Dashboard", "Consoles", "Files", "Web", "Tasks", and "Databases". A green banner at the top states "Your email address has been confirmed. You can manage your email preferences from [this page](#)". Below the banner, the word "Welcome" is followed by a redacted name. On the left, there's a "Recent Consoles" section with a "+ 5 -" button, stating "You have no recent consoles." and buttons for "New console:" (with options "\$ Bash", ">>> Python", and "More..."), "Recent Files" with a "+ 5 -" button, stating "You have no recently edited files.", and "Recent Notebooks" with a "+ 5 -" button, stating "Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!" and a "Browse files" button. On the right, there's an "All Web apps" section with a "+ 5 -" button, stating "You don't have any web apps." and a "Open Web tab" button. A blue "Upgrade Account" button is located in the top right corner of the dashboard area.

Dashboard

Welcome, [REDACTED]

CPU Usage: 0% used – 0.00s of 100s. Resets in 23 hours, 59 minutes [More Info](#)

File storage: 0% full – 60.0 KB of your 512.0 MB quota [More Info](#)

[Upgrade Account](#)

Recent Consoles [+ 5 -](#)
You have no recent consoles.
New console: [\\$ Bash](#) [>>> Python](#) [More...](#)

Recent Files [+ 5 -](#)
You have no recently edited files.
[+ Open another file](#) [Browse files](#)

Recent Notebooks [+ 5 -](#)
Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!
[Browse files](#)

All Web apps
You don't have any web apps.
[Open Web tab](#)

Figura 4 Dashboard di benvenuto di PythonAnywhere.

Passo 5: Test della Console

Per verificare il funzionamento, ho aperto una console "Bash" cliccando sul link apposito nella dashboard. L'ambiente ha fornito un terminale Linux funzionante direttamente nel browser.

The screenshot shows a web browser window with the URL "www.pythonanywhere.com/user/consoles/45114813". The title bar of the browser window says "Bash console 45114813". The terminal window displays the following text:
14:40 ~ \$ python --version
Python 3.13.1
14:40 ~ \$ █

Figura 5 Terminale Bash attivo nel browser.

Analisi Tecnica Avanzata: Limitazioni e Sicurezza

Dopo aver configurato l'ambiente di base, è stata effettuata un'analisi più approfondita delle caratteristiche tecniche del piano sottoscritto e delle opzioni di sicurezza disponibili sulla piattaforma.

Limitazioni del Piano "Free"

L'analisi del piano gratuito "Beginner" ha evidenziato alcune restrizioni tecniche significative, tipiche degli ambienti PaaS dimostrativi:

- **Connattività in uscita limitata:** L'ambiente non permette connessioni arbitrarie verso l'esterno. È possibile connettersi solo a una "whitelist" di siti pubblici approvati. Questo impedisce, ad esempio, di usare l'ambiente per creare bot che scansionano liberamente il web.
- **Risorse di Calcolo:** Esiste una quota limitata di tempo CPU (CPU seconds) giornaliero.

- **Persistenza dell'Applicazione:** Le web app ospitate con il piano gratuito vengono messe in stato di "sleep" se non ricevono traffico per un certo periodo o se non vengono "rinnovate" manualmente dall'utente tramite la dashboard, rendendo questo piano inadatto per servizi di produzione critici.

Implementazione della Sicurezza (2FA)

In ottica di *Hardening* dell'account (messa in sicurezza), è stata verificata la possibilità di attivare l'autenticazione a due fattori (2FA). PythonAnywhere supporta il protocollo standard **TOTP** (Time-based One-Time Password).

Procedura di attivazione:

1. Dalla Dashboard, ho navigato nella sezione "**Account**" in alto a destra.
2. Nella scheda "**Security**", è disponibile l'opzione per abilitare la "Two-factor authentication".
3. L'attivazione richiede la scansione di un codice QR tramite un'app di autenticazione (come Google Authenticator o Microsoft Authenticator) e l'inserimento del codice di verifica generato.

Questa configurazione è fortemente raccomandata poiché mitiga il rischio di accesso non autorizzato anche in caso di compromissione della password (attacchi di Brute Force o Credential Stuffing).

The screenshot shows the PythonAnywhere account settings interface. At the top, there's a logo for 'pythonanywhere by ANACONDA'. Below it, a navigation bar has tabs for 'Upgrade/Downgrade Account', 'Security' (which is highlighted in white), 'Email', 'Education', 'API Token', and 'System Image'. The main content area is titled 'Change your password:' and contains three input fields: 'Your old password', 'Your new password', and 'Your new password again'. A blue button labeled 'Change password' is below these fields. Further down, under the heading 'Set up Two-Factor Authentication:', there's a paragraph explaining the process and a green button with the text 'Click here to set up two-factor authentication' which is enclosed in a red rectangular box.

Figura 6 Pannello di configurazione della sicurezza e attivazione 2FA.

Conclusioni

L'attività svolta ha permesso di consolidare le conoscenze teoriche sui modelli di servizio cloud. La registrazione su PythonAnywhere ha dimostrato concretamente il concetto di **PaaS**:

1. Non è stato necessario installare Python o configurare un server web sulla macchina locale (Windows).
2. Non è stato necessario configurare il sistema operativo sottostante (Linux) o la rete.
3. L'ambiente era pronto all'uso in pochi secondi, confermando che il vantaggio principale del PaaS è permettere all'utente di concentrarsi esclusivamente sul codice e sull'applicazione.

Questo tipo di soluzione è raccomandata per lo sviluppo rapido, la didattica e il prototyping, mentre soluzioni IaaS potrebbero essere necessarie per architetture che richiedono configurazioni di rete personalizzate o software non standard.