

Nama : Ali Akbar Qumainy

NIM : 23611121

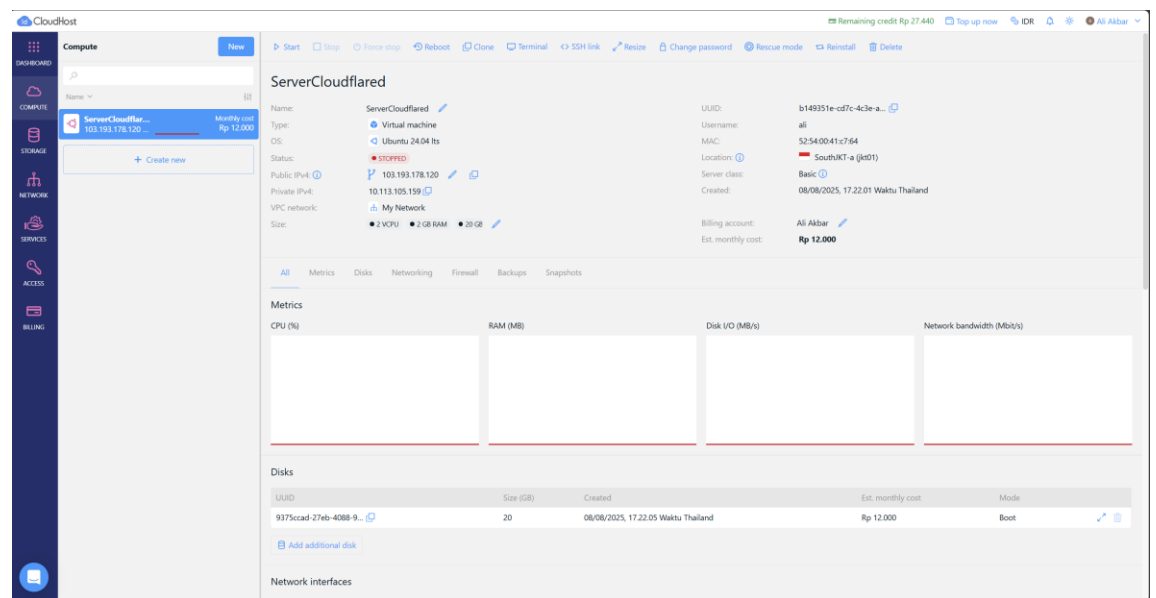
Deployment Aplikasi Web Python dengan VPS dan Cloudflare

Proses deployment ini menguraikan langkah-langkah untuk menjalankan aplikasi web berbasis Python secara online menggunakan Virtual Private Server (VPS) yang dihubungkan dengan Cloudflare Tunnel untuk keamanan dan aksesibilitas.

1. Konfigurasi Server Virtual (VPS)

Langkah awal adalah mempersiapkan server yang akan menjadi hosting aplikasi.

- Server dibuat menggunakan layanan VPS dari IDCloudhost. Spesifikasi server yang digunakan adalah 2 CPU, 2GB RAM, dan 20GB disk, dengan lokasi server di Jakarta Selatan.



- Server diaktifkan dengan menekan tombol "start" pada dashboard provider. Kemudian, koneksi ke server dilakukan dari perangkat lokal menggunakan protokol SSH (Secure Shell).

```
ali@ServerCloudflared: ~
Microsoft Windows [Version 10.0.22631.5699]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>ssh ali@103.193.178.120
ali@103.193.178.120's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-71-generic x86_64)

 * Panduan: https://idcloudhost.com/panduan
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Sat Aug 9 11:59:22 UTC 2025

System load: 0.47          Processes: 118
Usage of /: 16.6% of 18.3GB Users logged in: 0
Memory usage: 9%          IPv4 address for ens3: 10.113.105.159
Swap usage: 0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

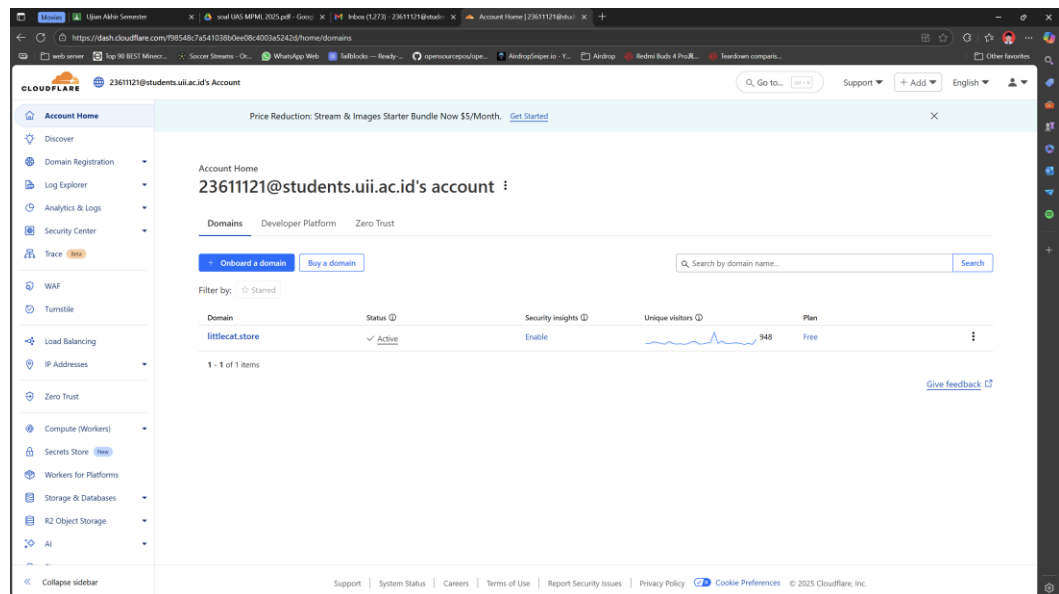
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Aug 9 04:46:13 2025 from 103.222.255.184
ali@ServerCloudflared:~$
```

2. Pengaruan Cloudflare Tunnel.

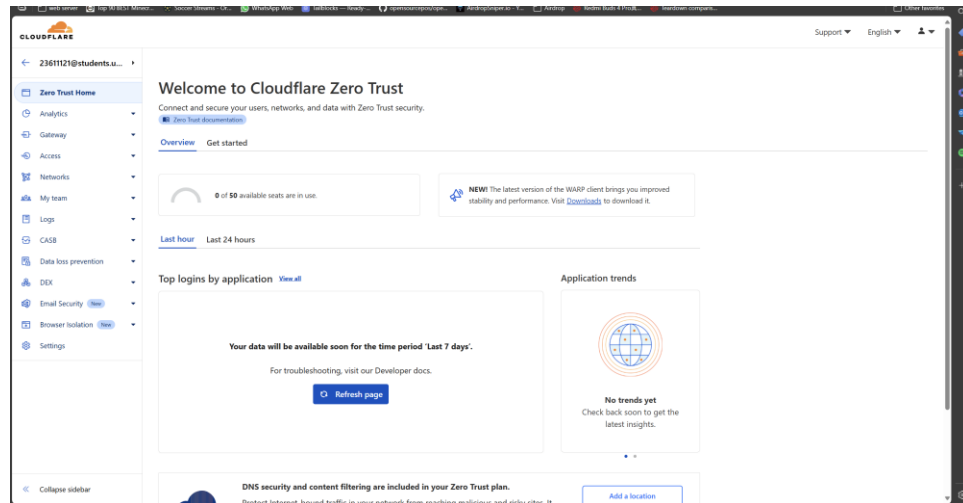
Cloudflare Tunnel digunakan untuk menghubungkan server VPS ke jaringan Cloudflare dengan aman tanpa harus membuka port publik di server.

- Akses Cloudflare



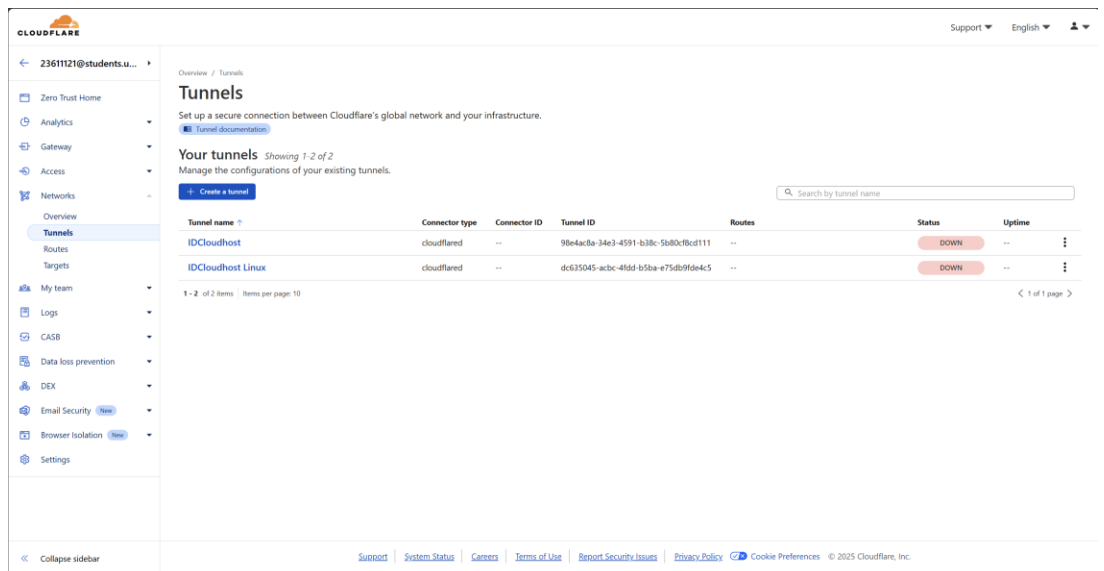
Disini saya tidak perlu menambahkan domain karena sudah ada dan jika belum ada bisa membeli domain baik lewat menu “Buy a domain” yang ada di Cloudflare atau provider lain.

- Mengakses menu “Zero Trust” yang akan digunakan untuk membuat tunnel



- Pembuatan Tunnel

1. Di dalam menu "Zero Trust", pilih "Networks" lalu klik "Tunnels".



2. Buat tunnel baru dengan memilih "Create a tunnel" dan pilih tipe tunnel "Cloudflared"

[← Back to Tunnels](#)


Create a tunnel

Create a tunnel to connect HTTP web servers, SSH servers, remote desktops, and other protocols safely to Cloudflare.

[Tunnel documentation](#)

Select your tunnel type

Choose the method used to connect your resources to Cloudflare's global network.




Cloudflared
Recommended

Establishes a secure, outbound-only connection to Cloudflare for user-to-network connectivity.

[Learn more](#)

Select Cloudflared



WARP Connector Beta
Linux distros only

Supports on-ramping and off-ramping traffic for site-to-site, bidirectional, and mesh networking connectivity.

[Learn more](#)

Select WARP Connector

Note: WARP tunnel connections will require global configurations, which may affect other services currently in use.

- Setelah tunnel dibuat, Cloudflare akan menyediakan perintah untuk menginstal konektor cloudflared di server. Salin dan jalankan perintah tersebut di terminal SSH VPS untuk menginstal layanan.

```
ali@ServerCloudflared:~$ sudo cloudflared service install eyJhIjoizjZk4NTQ4YzdhNTQxMDM4YjBkZTA4YzQwMDNhNTI0MmQlLCJ0Ijoiot
h1NGFjOGEtMzRLMy00NTkxLWlIZ0GMtNWl4MGNmOGNkMTExIiwicyI6IiIpUUTNOVFJqTm1VdE9EZGhNaTAwWpFeExUaGtaV0l0Tld0aVpEVmh0VFUwTVRRNS
J9
2025-08-09T12:01:56Z INF Using Systemd
2025-08-09T12:01:57Z INF Linux service for cloudflared installed successfully
```

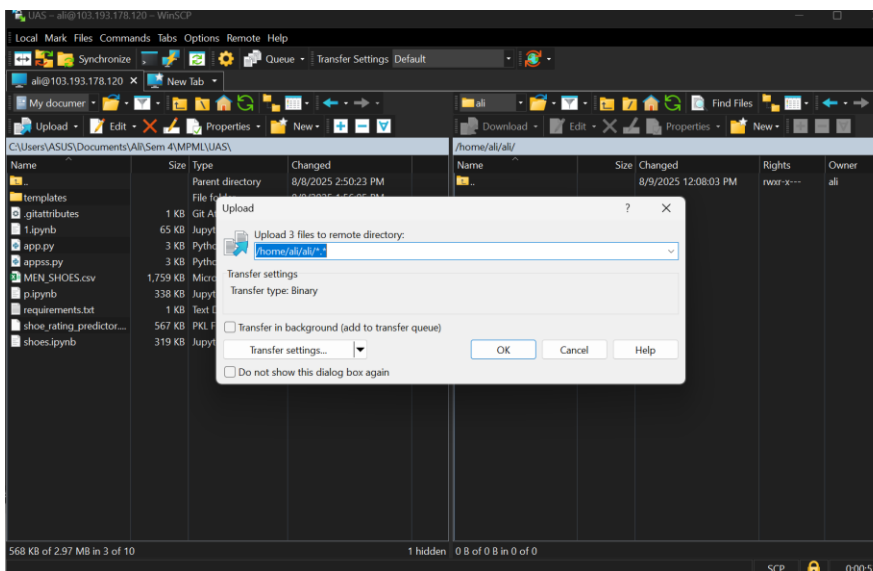
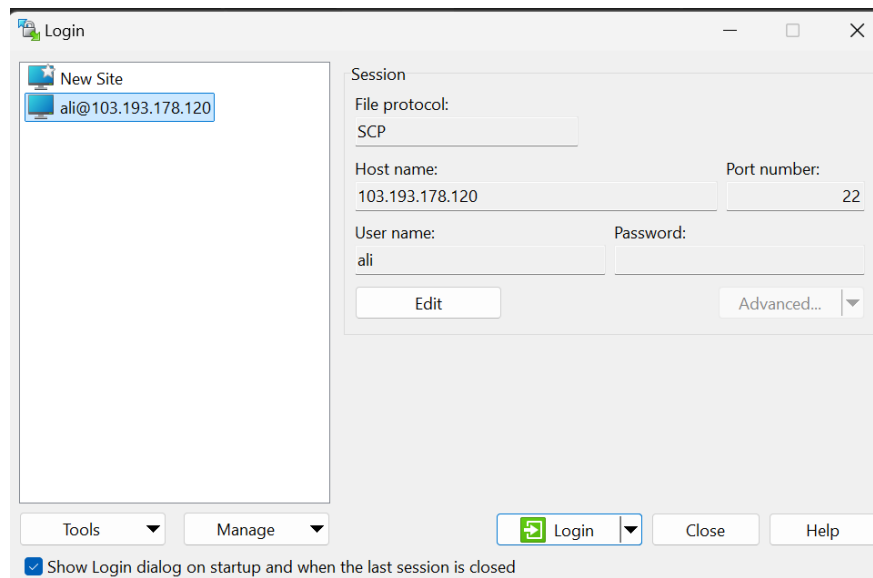
- Kembali ke dashboard Cloudflare. Jika koneksi berhasil, status tunnel akan menunjukkan "HEALTHY".

Tunnel name	Connector type	Connector ID	Tunnel ID	Routes	Status	Uptime
IDCloudhost	cloudflared	2e034a46-1185-4957-a772-9abc752c6a94	98e4ac8a-34e3-4591-b38c-5b80cf8cd111	--	HEALTHY	1 minutes

3. Deployment Aplikasi

Setelah server dan tunnel sudah siap, langkah selanjutnya adalah mengunggah aplikasi ke server dan menjalankannya.

- File-file yang diperlukan untuk aplikasi, seperti app.py (server), templates/index.html (tampilan website), dan file model machine learning, diunggah ke server. Proses ini dilakukan menggunakan protokol SCP (Secure Copy Protocol).



/home/ali/ali/				
Name	Size	Changed	Rights	Owner
templates		8/9/2025 12:08:03 PM	rw-r-x---	ali
venv		8/9/2025 12:09:03 PM	rw-r-xr-x	ali
app.py	3 KB	8/9/2025 12:13:03 PM	rw-rw-r-x	ali
requirements.txt	1 KB	8/9/2025 12:16:01 PM	rw-r--r--	ali
shoe_rating_predictor....	567 KB	8/8/2025 7:08:21 AM	rw-r--r--	ali

- Persiapan Environment Python

1. Buat environment virtual Python menggunakan perintah

```
ali@ServerCloudflared:~/ali$ python3 -m venv venv
```

2. Mengaktifkan environments

```
ali@ServerCloudflared:~/ali$ source venv/bin/activate
(venv) ali@ServerCloudflared:~/ali$
```

3. Menginstall packages yang diperlukan.

```
(venv) ali@ServerCloudflared:~/ali$ pip install -r requirements.txt
Collecting flask (from -r requirements.txt (line 1))
  Using cached flask-3.1.1-py3-none-any.whl.metadata (3.0 kB)
Collecting joblib (from -r requirements.txt (line 2))
  Using cached joblib-1.5.1-py3-none-any.whl.metadata (5.6 kB)
Collecting pandas (from -r requirements.txt (line 3))
  Using cached pandas-2.3.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (91 kB)
Collecting numpy (from -r requirements.txt (line 4))
  Using cached numpy-2.3.2-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (62 kB)
Collecting scikit-learn==1.6.1 (from -r requirements.txt (line 5))
  Using cached scikit-learn-1.6.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (18 kB)
Collecting scipy>=1.6.0 (from scikit-learn==1.6.1->-r requirements.txt (line 5))
  Using cached scipy-1.16.1-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (61 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn==1.6.1->-r requirements.txt (line 5))
  Using cached threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Collecting blinker>=1.9.0 (from flask->-r requirements.txt (line 1))
  Using cached blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from flask->-r requirements.txt (line 1))
  Using cached click-8.2.1-py3-none-any.whl.metadata (2.5 kB)
Collecting itsdangerous>=2.2.0 (from flask->-r requirements.txt (line 1))
  Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from flask->-r requirements.txt (line 1))
  Using cached jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from flask->-r requirements.txt (line 1))
  Using cached MarkupSafe-3.0.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.0 kB)
Collecting werkzeug>=3.1.0 (from flask->-r requirements.txt (line 1))
  Using cached werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting python-dateutil>=2.8.2 (from pandas->-r requirements.txt (line 3))
  Using cached python-dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas->-r requirements.txt (line 3))
  Using cached pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas->-r requirements.txt (line 3))
  Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas->-r requirements.txt (line 3))
  Using cached six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Using cached scikit-learn-1.6.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.1 MB)
Using cached flask-3.1.1-py3-none-any.whl (103 kB)
Using cached joblib-1.5.1-py3-none-any.whl (307 kB)
Using cached pandas-2.3.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.0 MB)
Using cached numpy-2.3.2-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl (16.6 MB)
Using cached blinker-1.9.0-py3-none-any.whl (8.5 kB)
Using cached click-8.2.1-py3-none-any.whl (102 kB)
Using cached itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Using cached jinja2-3.1.6-py3-none-any.whl (134 kB)
Using cached MarkupSafe-3.0.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (23 kB)
```

4. Menjalankan app.py.

```
(venv) ali@ServerCloudflared:~/ali$ python3 app.py
Model loaded successfully!
Could not load brand names from CSV: [Errno 2] No such file or directory: 'MEN_SHOES.csv'
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8000
* Running on http://10.113.105.159:8000
Press CTRL+C to quit
```

4. Konfigurasi Domain Publik

Langkah terakhir adalah membuat aplikasi dapat diakses melalui domain publik. Buka menu “Public Hostname” dan lakukan konfigurasi seperti berikut:

[← Back to IDCloudhost](#)

Public hostnames

Edit public hostname for IDCloudhost

Hostname

Subdomain

(optional) subdomain

Domain (Required)

littlecat.store

Path

(optional) path

Service

Type (Required)

HTTP

URL (Required)

localhost:8050

For example, <https://localhost:8001>

[Additional application settings >](#)

Save

5. Mencoba mengakses aplikasi web yang telah dibuat.

