



UNIVERSITAS
INDONESIA

CEP-CCIT
FAKULTAS TEKNIK

HOW TO SETUP BLOG

Implementasi Infrastruktur Blog Berbasis Container pada Home Server Menggunakan CasaOS dan Cloudflare Tunnel

Disusun Oleh :

Bergas Ahmad Ardiansyah (2320010123)

4CS1

Dosen Pengampu :

Tri Agus Riyadi, S. T., M. T.

CEP CCIT

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

2025

PROJECT ON
Cyber Security Ethics

Disusun Oleh :

1. Bergas Ahmad Ardiansyah

Batch Code : 4CS1

Implementasi Infrastruktur Blog Berbasis Container pada Home Server Menggunakan CasaOS dan Cloudflare Tunnel

Tanggal Mulai : 1 Maret 2025

Tanggal Selesai : 22 Maret 2025

Dosen Pengampu : Tri Agus Riyadi, S. T., M. T.

Disusun Oleh :

1. Bergas Ahmad Ardiansyah

Tanggal Pengumpulan : 23 Maret 2025

CERTIFICATE

Dengan ini disahkan bahwa laporan berjudul *"Implementasi Infrastruktur Blog Berbasis Container pada Home Server Menggunakan CasaOS dan Cloudflare Tunnel"* merupakan hasil karya asli yang dibuat oleh Bergas Ahmad Ardiansyah. Proyek ini disusun sebagai bagian dari pemenuhan persyaratan mata kuliah di CEP CCIT Fakultas Teknik Universitas Indonesia.

Coordinator :

Tri Agus Riyadi, S. T., M. T.

ACKNOWLEDGEMENT

Puji Syukur yang mendalam kehadiran Allah SWT, Tuhan yang maha esa atas segala limpahan rahmat dan karunia-Nya. Berkat rahmat dan kasih sayang-Nya, serta tidak lupa shalawat dan salam senantiasa kita curahkan kepada Nabi Muhammad SAW, penulis dapat menyelesaikan proyek ini yang berjudul *"Implementasi Infrastruktur Blog Berbasis Container pada Home Server Menggunakan CasaOS dan Cloudflare Tunnel"*. Tanpa pertolongan-Nya, proyek ini tidak akan terselesaikan tepat waktu sebagaimana yang telah direncanakan.

Penulis juga menyampaikan terima kasih yang sebesar-besarnya kepada Bapak **Tri Agus Riyadi, S.T., M.T.**, selaku dosen pembimbing yang telah memberikan arahan, saran, dan masukan yang sangat membantu dalam penyusunan proyek ini.

Meskipun dalam proses pengerjaan terdapat berbagai tantangan dan hambatan, penulis bersyukur proyek ini akhirnya dapat diselesaikan. Penulis menyadari bahwa tugas ini masih jauh dari kata sempurna. Oleh karena itu, penulis sangat mengharapkan saran dan kritik yang membangun dari para dosen serta rekan-rekan demi penyempurnaan proyek ini ke depannya.

SYSTEM ANALYSIS

System Overview

Sistem ini merupakan implementasi blog self-hosted menggunakan Docker Container yang dikelola melalui **CasaOS**, sebuah antarmuka web yang memudahkan manajemen layanan berbasis container. Blog ini ditujukan untuk keperluan personal/public dan diakses melalui domain yang terhubung menggunakan **Cloudflare Tunnel**. Layanan tidak menggunakan Apache/Nginx secara langsung, melainkan sepenuhnya berbasis container Docker.

Hardware & Platform

- Perangkat : STB (Set-Top Box) ZTE HG680P dengan arsitektur aarch64
- Sistem Operasi : Armbian (Debian/Ubuntu based)
- Manajemen WebUI : CasaOS
- Container Engine : Docker

Domain & Akses

- Domain : Dikelola melalui Cloudflare
- Akses Eksternal : Menggunakan Cloudflare Tunnel untuk port forwarding aman dari internet ke server lokal
- SSL : Disediakan oleh Cloudflare (tanpa pengaturan SSL lokal di server)

INSTALLING CASA OS

1. Perbarui sistem dan repository

Lakukan pembaruan pada sistem dan repository agar tidak terjadi error saat penginstalan CASA OS dengan menggunakan Perintah Berikut

```
sudo apt update && sudo apt upgrade -y
```

```
vboxuser@DOKUMENTASI:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for vboxuser:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
54 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  ubuntu-drivers-common
The following packages will be upgraded:
  apparmor apport apport-core-dump-handler cloud-init cryptsetup cryptsetup-bin cryptsetup-initramfs distro-info-data
  dmsetup dmsync dracut-install fwupd gir1.2-packagekit-glib-1.0 initramfs-tools initramfs-tools-bin
  initramfs-tools-core landscape-common libapparmor1 libcryptsetup12 libdevmapper-event1.02.1 libdevmapper1.02.1
  libfwupd2 libldap-common libldap2 liblvm2cmd2.03 libnetplan1 libnss-systemd libnvmelt64 libopeniscsiusr
  libpackagekit-glib2-18 libpam-systemd libplymouth5 libsystemd-shared libsystemd0 libudev1 linux-base lvm2
  netplan-generator netplan.io open-iscsi packagekit packagekit-tools pci.ids plymouth plymouth-theme-ubuntu-text
  pollinate python3-apport python3-netplan python3-problem-report python3-software-properties rsyslog snapd
  software-properties-common sosreport systemd systemd-dev systemd-resolved systemd-sysv systemd-timesyncd
  ubuntu-pro-client ubuntu-pro-client-l10n udev update-notifier-common
53 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
Need to get 52.0 MB of archives.
After this operation, 687 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu noble-updates/main amd64 libnss-systemd amd64 255.4-1ubuntu8.6 [159 kB]
```

2. Install Casa OS pada sistem

Install Casa OS sebagai Manajemen WebUI dengan perintah berikut

```
curl -fsSL https://get.casaos.io | sudo bash
```

```

  CasaOS
  -----

  -- Made by IceWhale with YOU --

[ OK ] Your hardware architecture is : x86_64
[ OK ] Your System is : Linux
[ OK ] Your Linux Distribution is : ubuntu
[ OK ] Memory capacity check passed.
[ OK ] Disk capacity check passed.
[ INFO ] Updating package manager...
[ OK ] Update package manager complete.
[ INFO ] Install the necessary dependencies: smartmontools
Selecting previously unselected package smartmontools.
(Reading database ... 86855 files and directories currently installed.)
Preparing to unpack .../smartmontools_7.4-2build1_amd64.deb ...
Unpacking smartmontools (7.4-2build1) ...
Setting up smartmontools (7.4-2build1) ...
/var/lib/smartmontools/drivedb/drivedb.h 7.3/5528 newly installed (NOT VERIFIED)
Created symlink /etc/systemd/system/multi-user.target.wants/smartd.service → /usr/lib/systemd/system/smartd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/smartmontools.service → /usr/lib/systemd/system/smartmontool
s.service.
smartd.service is a disabled or a static unit, not starting it.
Processing triggers for man-db (2.12.0-4build2) ...
```

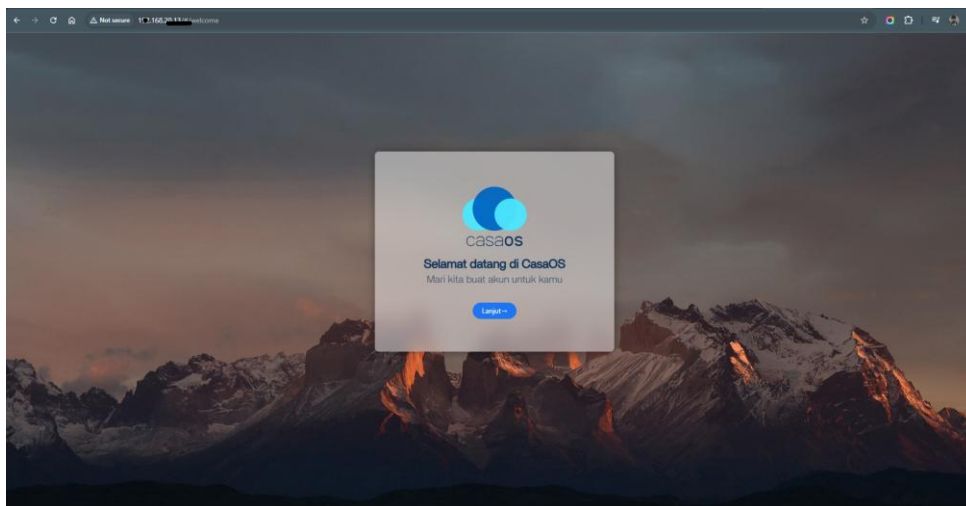
Instalasi Casa OS langsung terintegrasi dengan Docker, jadi tidak setelah langkah ini tidak perlu menginstall docker secara mandiri karena otomatis terinstall bersamaan dengan Casa OS

```
CasaOS v0.4.15 is running at:  
- http://192.168.20.13 (enp0s3)  
Open your browser and visit the above address.
```

Setelah Instalasi selesai, akan muncul informasi dimana Casa OS Dijalankan. Secara default, Casa OS akan Berjalan pada port 80

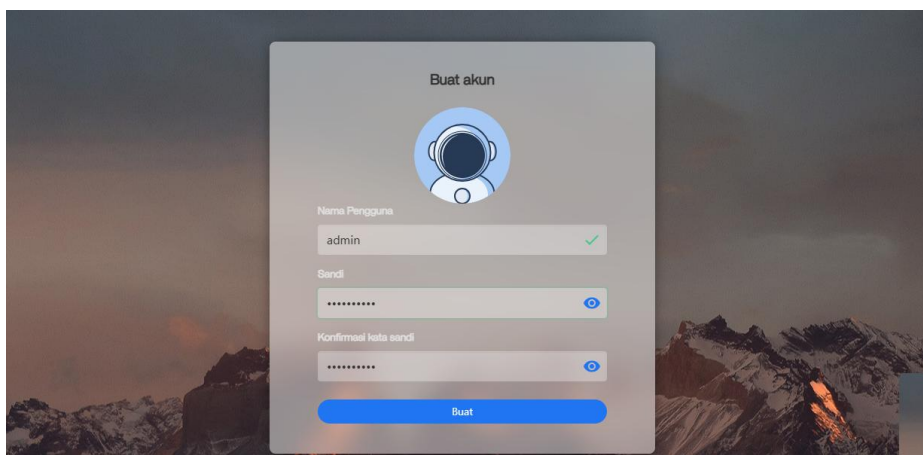
3. Buka Casa OS pada Browser

Untuk me buka WEB UI Casa OS pada browser, ketik ip address dan port host server pada browser pada device lain yang berada dalam satu jaringan.



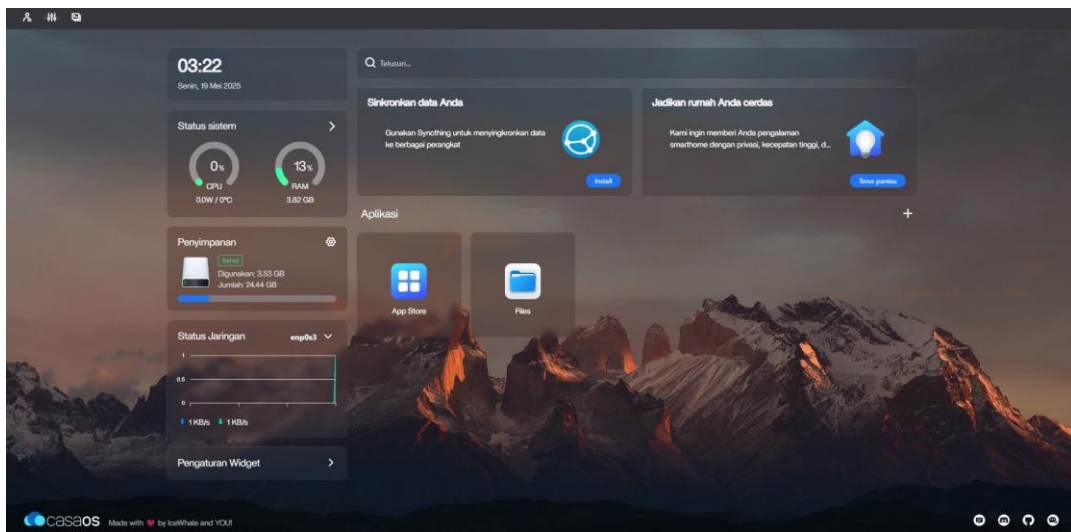
4. Buat akun admin

Klik lanjut, kemudian buat akun admin



5. Masuk Ke Dashboard

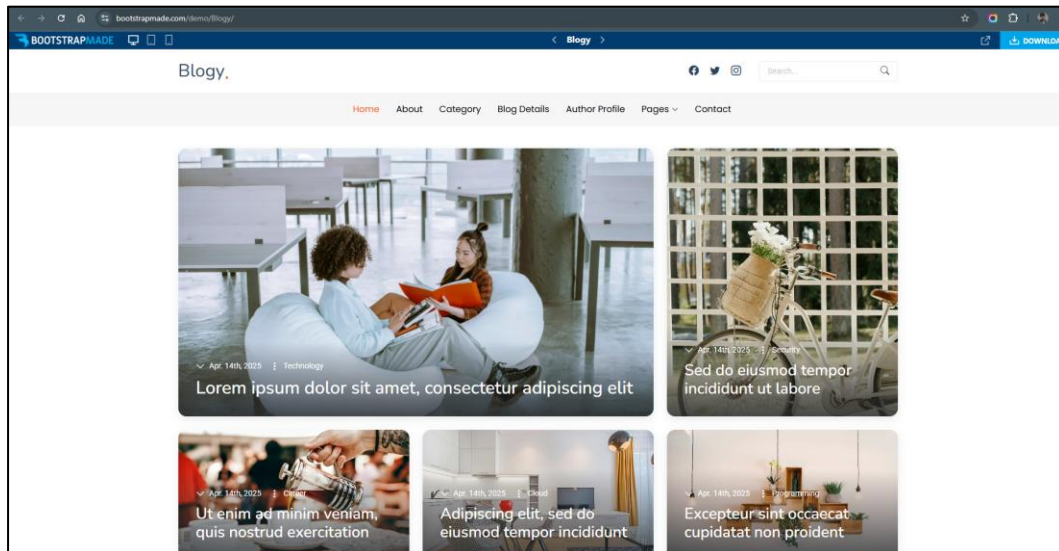
Setelah membuat akun admin, akan langsung terbuka halaman Dashboard dari Casa OS.



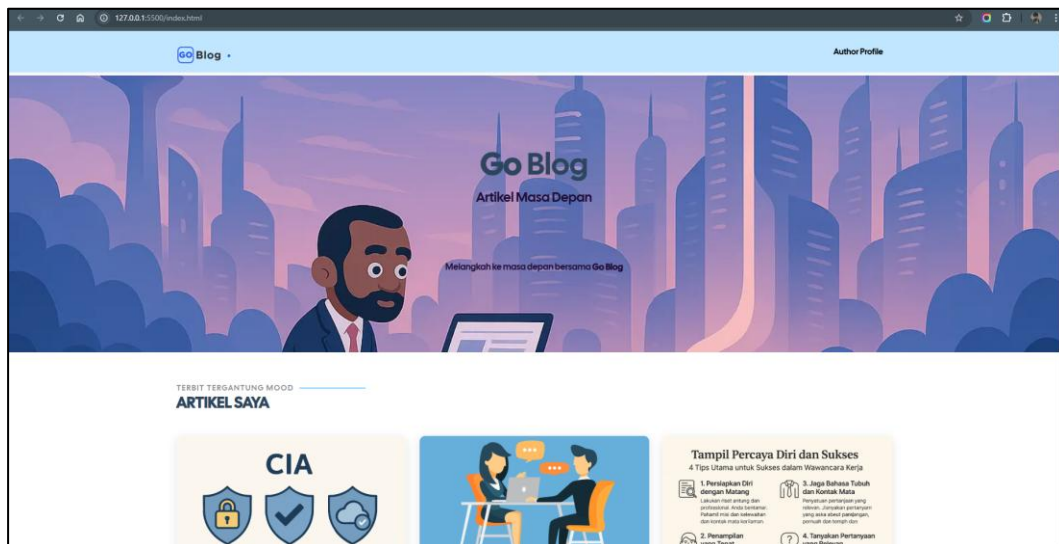
Deployment Website Via Docker Casa OS

1. Buat Website

Siapkan file website static (html,css,js) untuk blog, jika ingin mudah bisa menggunakan template blog dari bootstrap untuk mempercepat proses development



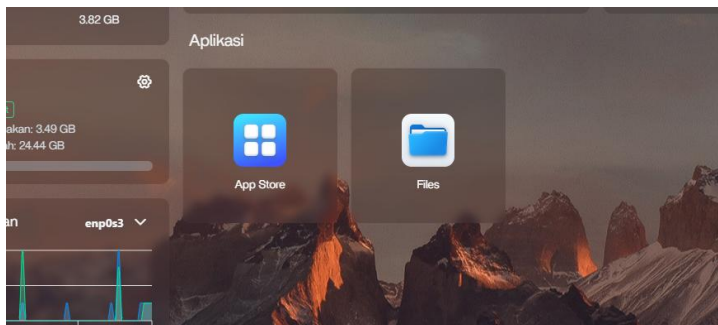
Contoh template website blog dari [bootstrapmade](https://bootstrapmade.com/)



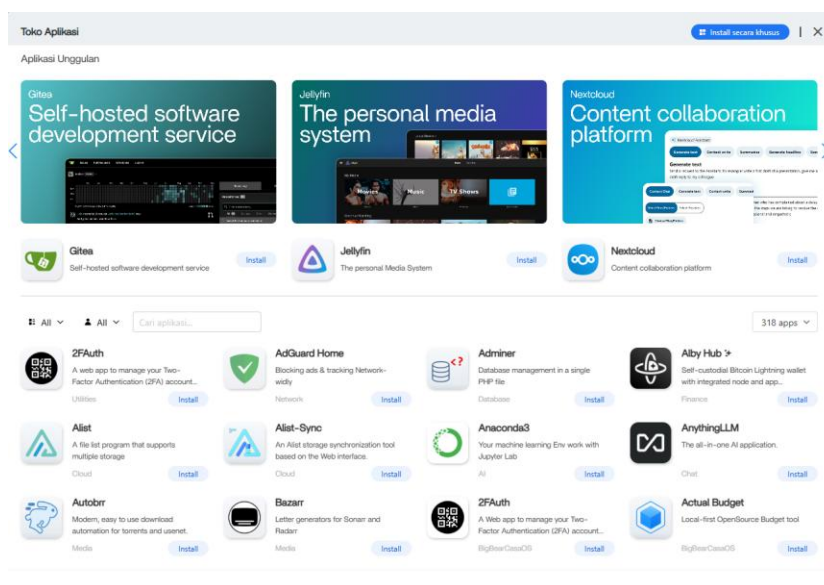
Kemudian edit dan buat artikel menggunakan VS Code agar menjadi lebih menarik

2. Install nginx dengan docker Pada Casa OS

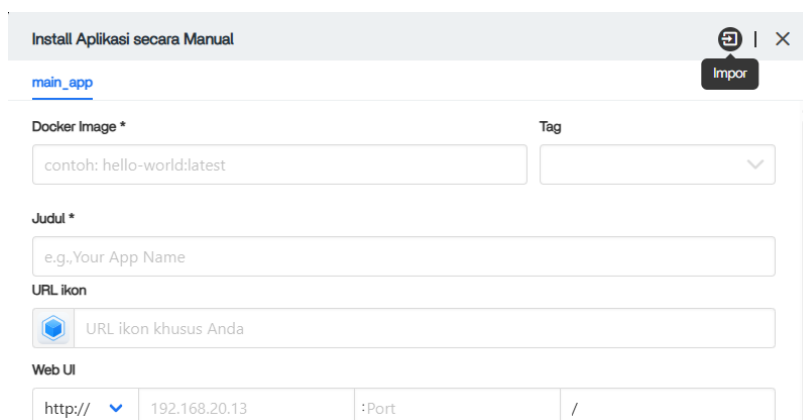
Buka kembali dashboard Casa OS, kemudian buka App Store



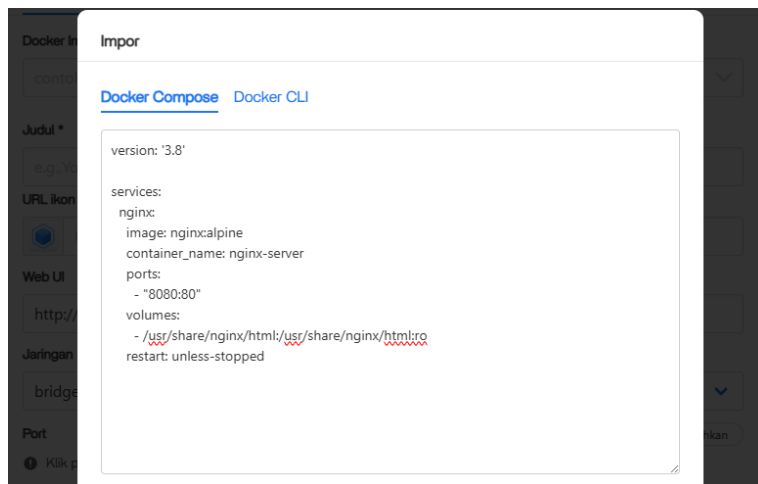
Klik install secara khusus pada pojok kanan atas



Klik Impor



Masukan Docker Compose untuk instalasi nginx

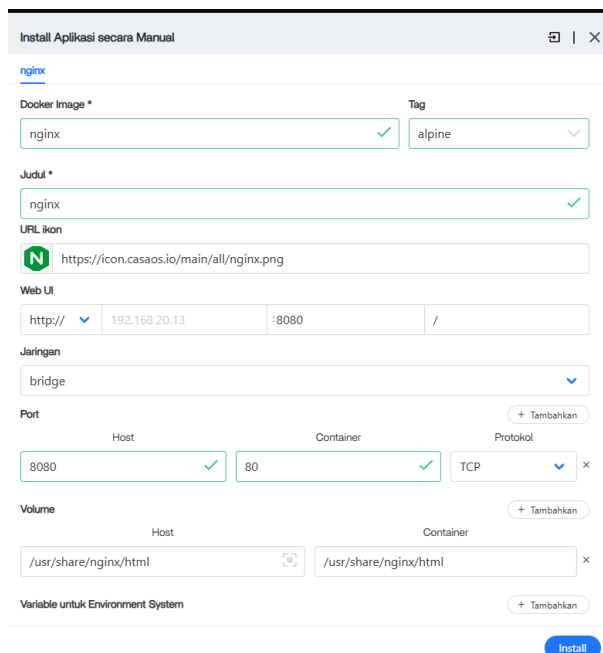


Docker compose dapat diunduh [disini](#)

Penjelasan:

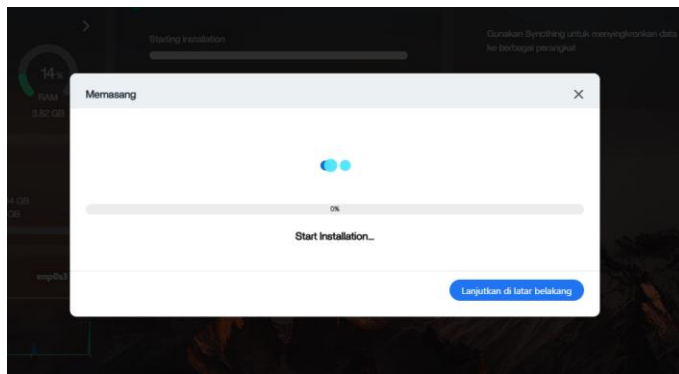
- nginx:alpine : image ringan Nginx
- ports : <port akses localhost>:<port internal docker container>
- volumes : tempat file html akan disimpan

Pastikan Konfigurasi pada Web UI sesuai dengan port pada host

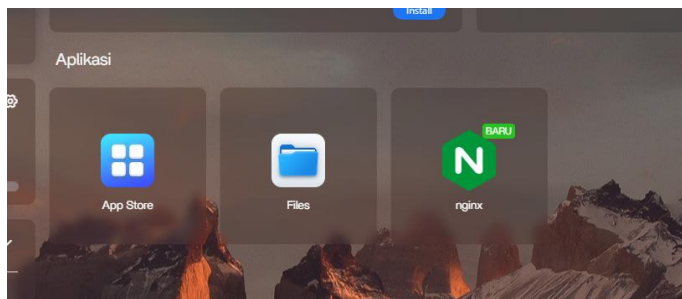


Klik Install

Tunggu Hingga Instalasi selesai



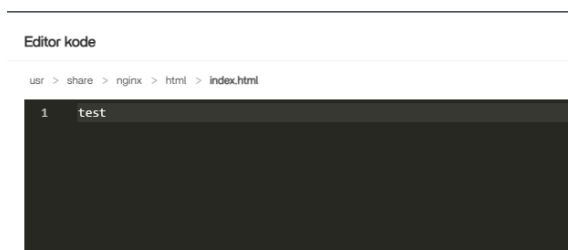
Jika Sudah selesai, akan muncul nginx pada Aplikasi di Dashboard Casa OS



Buka Folder html pada nginx sesuai dengan volumes pada docker compose (/usr/share/nginx/html) pada file manager,

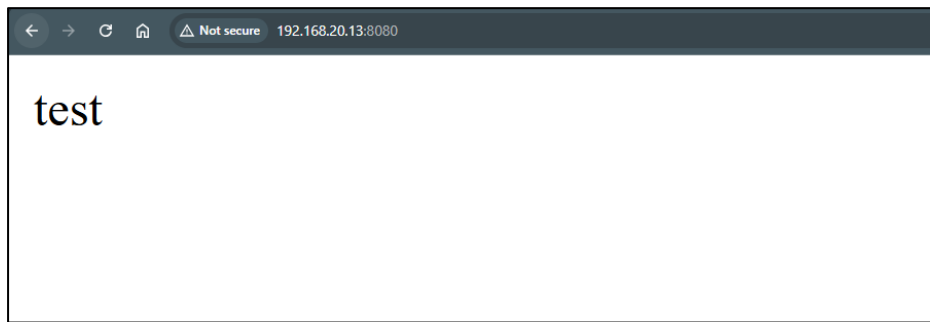


Klik kanan dan buat file baru berupa **index.html** untuk testing apakah nginx bekerja dengan baik



Isi file tersebut dengan **test** atau kata lainnya, kemudian simpan

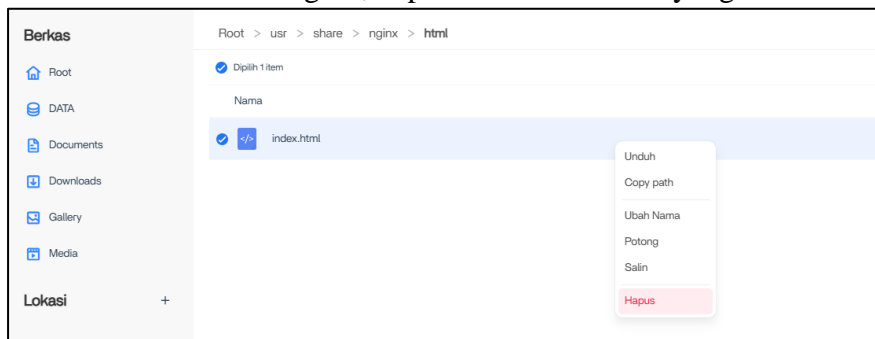
Buka ip dari server dan port yang dipakai oleh nginx



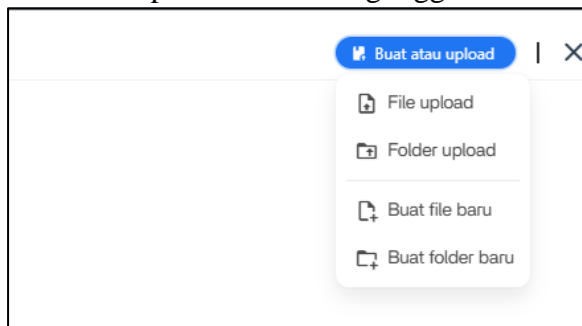
Jika teks yang dimasukkan kedalam file **index.html** muncul, berarti web server **nginx** telah berjalan dan siap digunakan

3. Upload File Website ke server

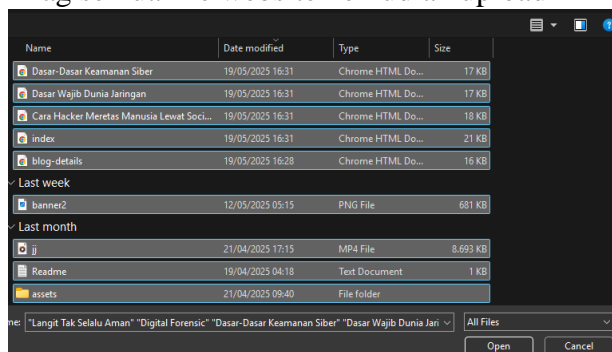
Pada direktori docker nginx, hapus File **index.html** yang dibuat untuk testing



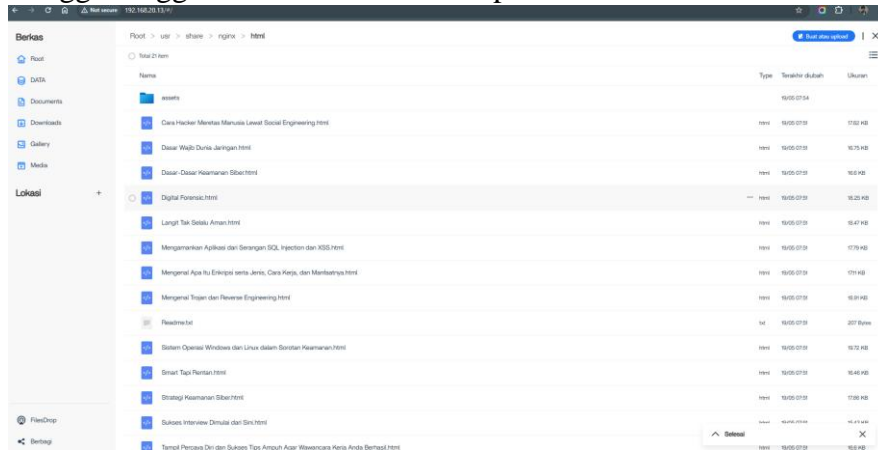
Klik File Upload untuk mengunggah file website



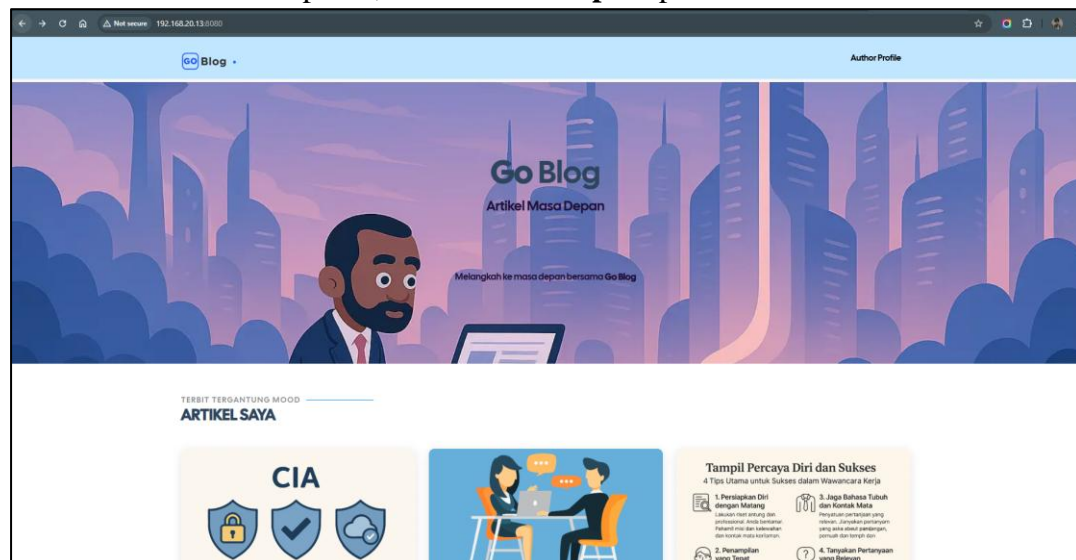
Drag semua file website kemudian upload



Tunggu hingga semua file selesai di upload



Setelah file selesai di upload, buka **localhost:port** pada browser



Jika muncul, maka blog telah berhasil di hosting pada server

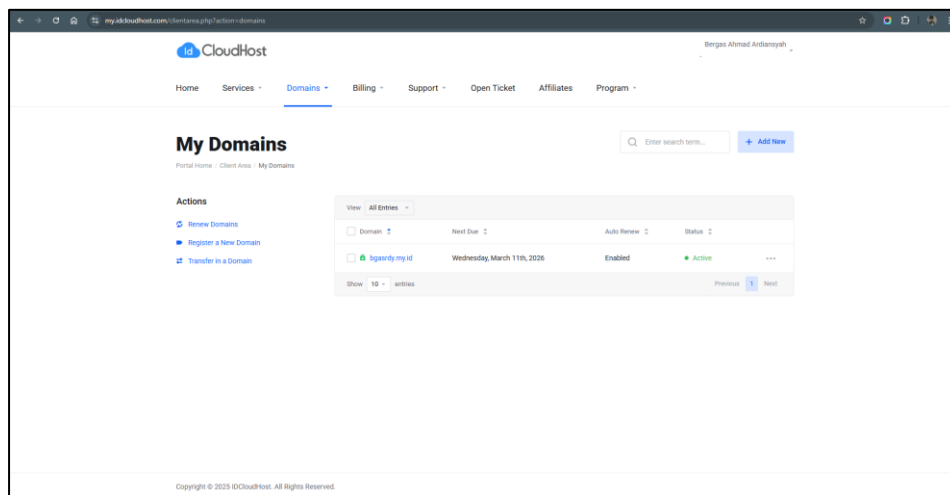
Catatan

- Port 8080 harus terbuka dan tidak diblokir oleh firewall
- Nginx berjalan di docker container dan tidak perlu meninstall nginx pada host secara langsung
- Jika ingin mengupdate file html, cukup unggah file kedalam folder nginx

Menghubungkan Server ke Domain dengan Cloudflare Tunnel

1. Beli Domain

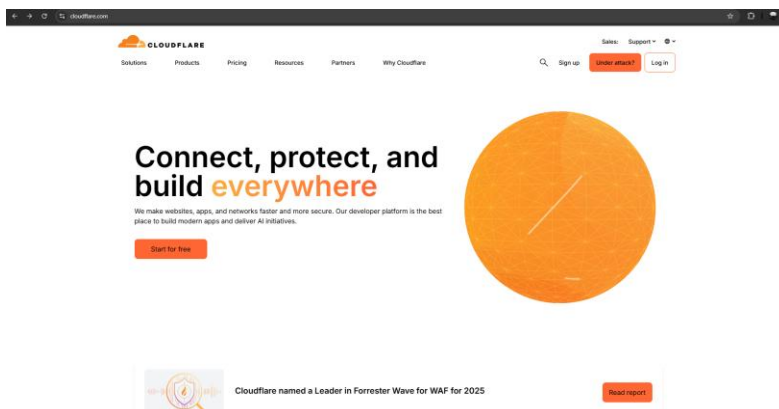
Beli domain atau bisa gunakan domain gratisan.



Pembelian Domain pada [idcloudhost](https://idcloudhost.com)

2. Menghubungkan domain ke cloudflare

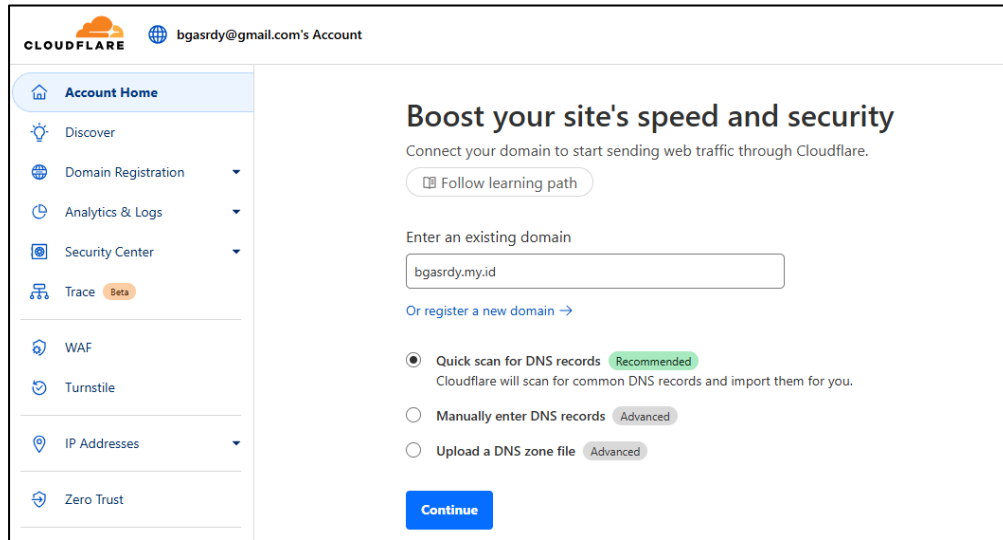
Buat akun/masuk ke cloudflare untuk mengakses dashboard [cloudflare](https://cloudflare.com)



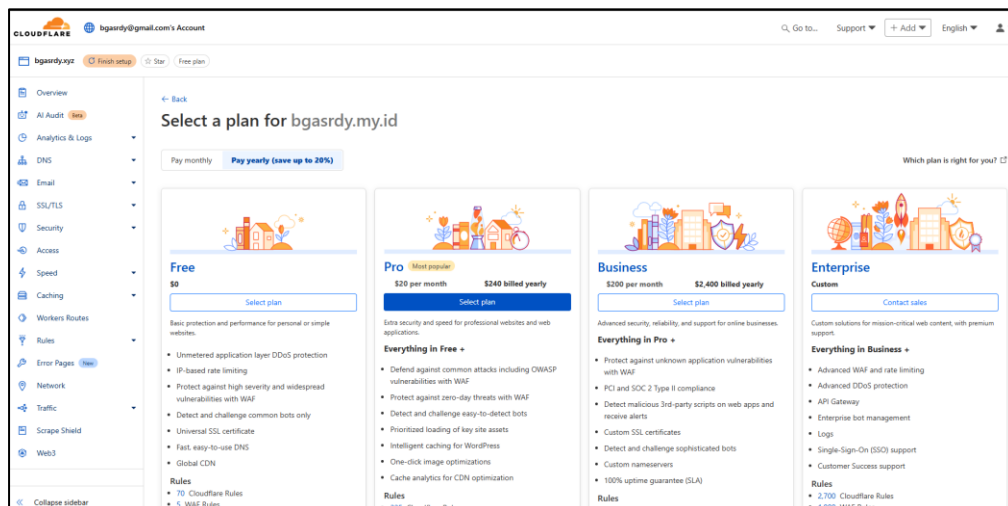
Jika suda masuk, maka otomatis akan dialihkan ke Dashboard Cloudflare

3. Masukkan domain ke Cloudflare

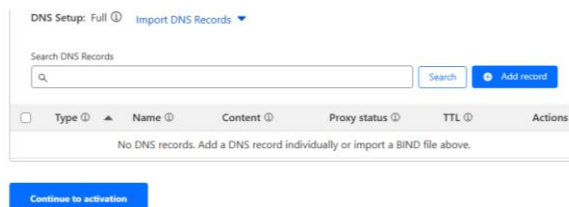
Masukkan domain yang sudah dibeli ke cloudflare, dan klik continue



Pilih plan sesuai kebutuhan



Kemudian klik Continue to Activate





Ubah Nameserver pada dashboard tempat anda beli domain, contohnya *idcloudhost.com*

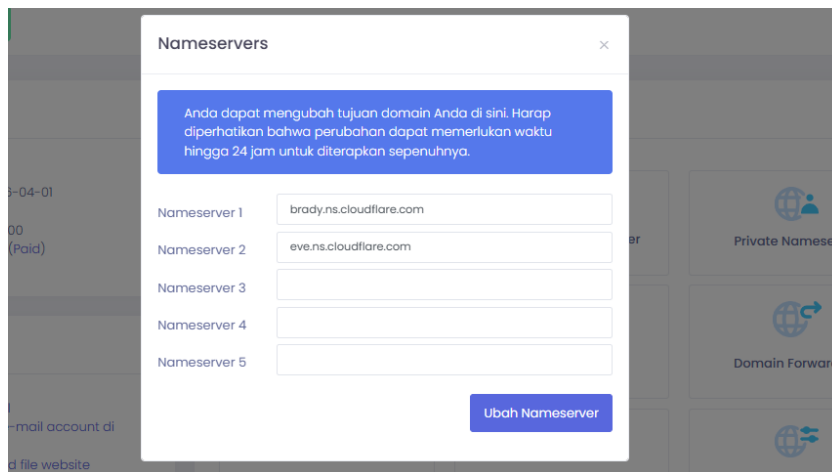
Sesuai dengan nameserver pada cloudflare

Last step: Update your nameservers to activate Cloudflare
This is the last step to allow Cloudflare to speed up and protect your web traffic.

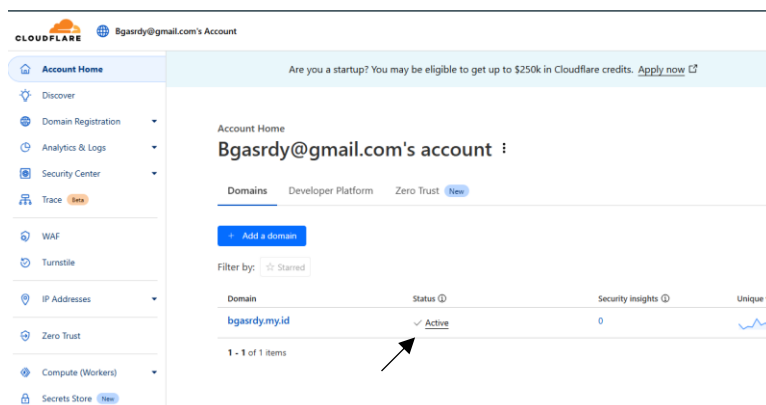
1. Log into your DNS provider (most likely your registrar)
→ [Find your registrar on ICANN Lookup](#)
If you purchased your domain through a reseller (e.g., Squarespace) or use a separate DNS provider, log into that account instead.
2. Make sure DNSSEC is off
Find and turn off the DNS security (DNSSEC) setting if it is on. You can re-enable it later through Cloudflare.
[Provider-specific instructions](#)
3. Replace your current nameservers with Cloudflare nameservers
This is unlikely to cause downtime, but you may skip this and check your [DNS records](#) first.
A. Find the nameservers section
B. Add both of your assigned Cloudflare nameservers:

 [Click to copy](#)

 [Click to copy](#)



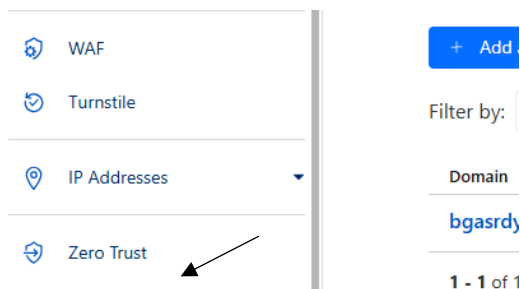
Jika sudah, tunggu hingga propagasi domain selesai



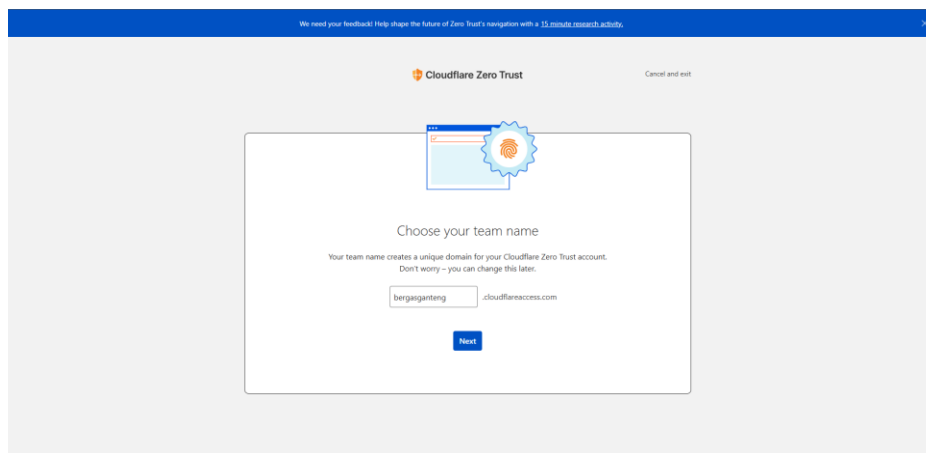
Jika sudah selesai, maka status domain akan active pada dashboard cloudflare

4. Konfigurasi Tunnel dengan Zero Trust

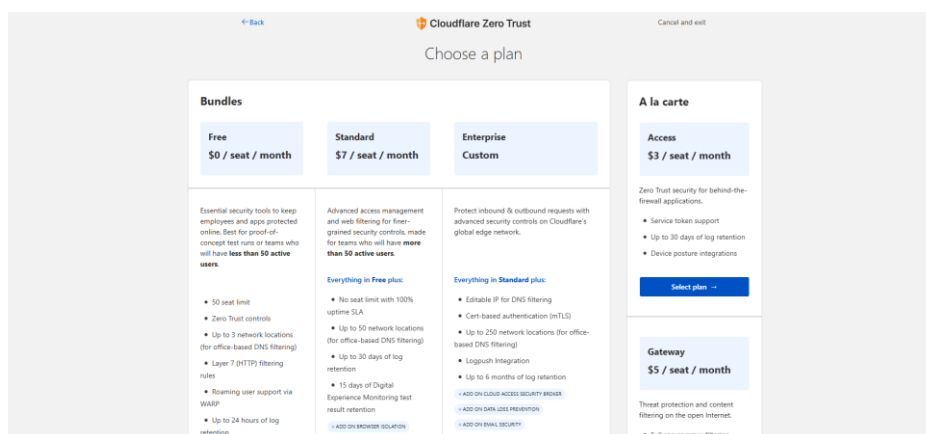
Buka menu Zero trust pada dashboard cloudflare



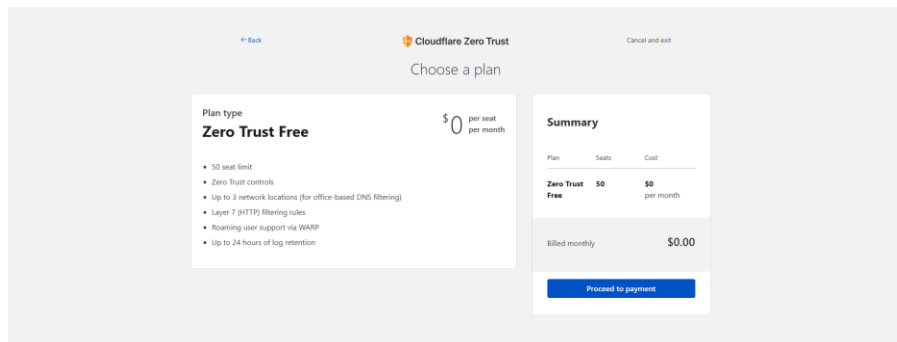
Masukan team name kemudian klik **Next**



Pilih plan sesuai kebutuhan dan keadaan isi dompet

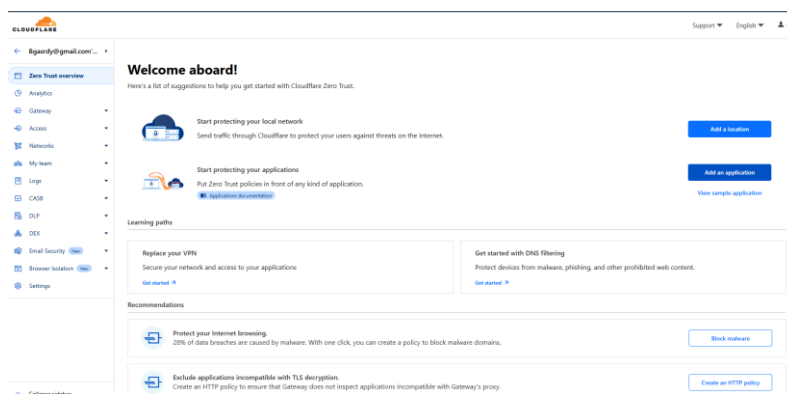


Jika keadaan dompet sama dengan **Author**, maka pilih free plan kemudian lanjut ke pembayaran*

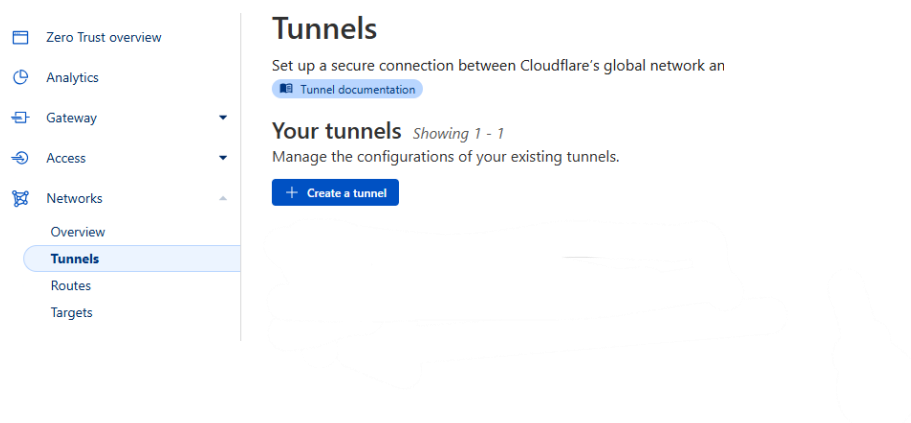


*Note: Siapkan Credit Card/Paypal untuk pembayaran meskipun ini free plan

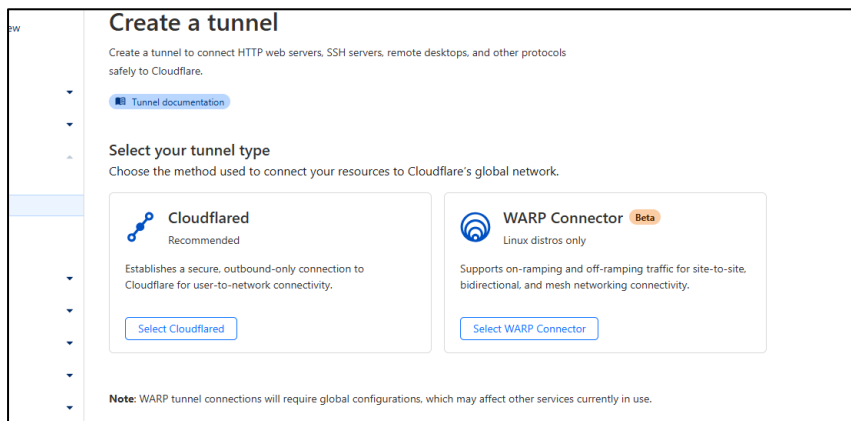
Setelah pembayaran selesai, dashboard dari zero trust akan terbuka



Pada dashboard zero trust, masuk ke Network>Tunnel>Create Tunnel



Pada bagian ini, pilih cloudflared




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

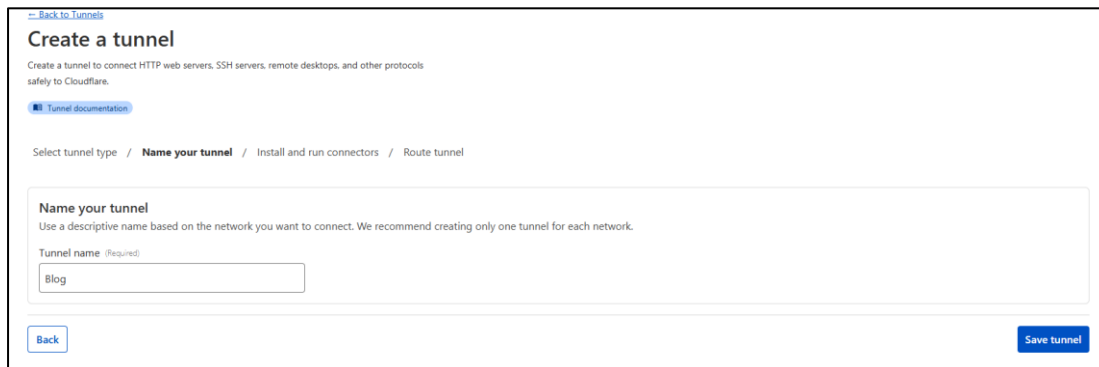
Choose the method used to connect your resources to Cloudflare's global network.

**Cloudflared**
Recommended
Establishes a secure, outbound-only connection to Cloudflare for user-to-network connectivity.
[Select Cloudflared](#)

**WARP Connector** Beta
Linux distros only
Supports on-ramping and off-ramping traffic for site-to-site, bidirectional, and mesh networking connectivity.
[Select WARP Connector](#)

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

Beri nama Tunnel Kemudian simpan Tunnel



[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 tunnel type / **Name your tunnel** / Install and run connectors / Route tunnel

Name your tunnel

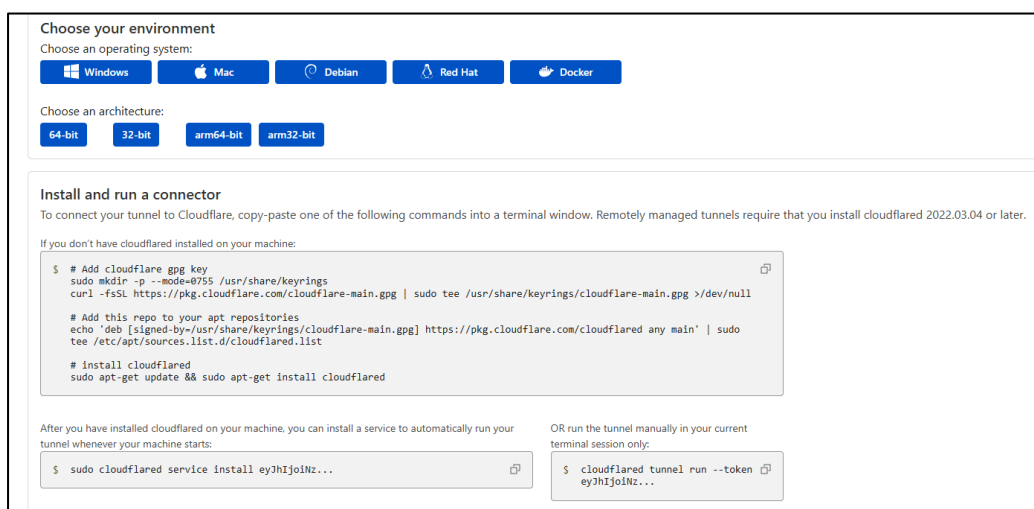
Use a descriptive name based on the network you want to connect. We recommend creating only one tunnel for each network.

Tunnel name (Required)

Blog

[Back](#) [Save tunnel](#)

Install connector pada server, sesuaikan environment dan arsitektur yang digunakan pada server agar tidak error



Choose your environment

Choose an operating system:

[Windows](#) [Mac](#) [Debian](#) [Red Hat](#) [Docker](#)

Choose an architecture:

[64-bit](#) [32-bit](#) [arm64-bit](#) [arm32-bit](#)

Install and run a connector

To connect your tunnel to Cloudflare, copy-paste one of the following commands into a terminal window. Remotely managed tunnels require that you install cloudflared 2022.03.04 or later.

If you don't have cloudflared installed on your machine:

```
$ # Add cloudflare gpg key
sudo mkdir -p --mode=0755 /usr/share/keyrings
curl -fsSL https://pkg.cloudflare.com/cloudflare-main.gpg | sudo tee /usr/share/keyrings/cloudflare-main.gpg >/dev/null

# Add this repo to your apt repositories
echo 'deb [signed-by=/usr/share/keyrings/cloudflare-main.gpg] https://pkg.cloudflare.com/cloudflare any main' | sudo tee /etc/apt/sources.list.d/cloudflared.list

# Install cloudflared
sudo apt-get update && sudo apt-get install cloudflared
```

After you have installed cloudflared on your machine, you can install a service to automatically run your tunnel whenever your machine starts:

```
$ sudo cloudflared service install ey3HjoiNz...
```

OR run the tunnel manually in your current terminal session only:

```
$ cloudflared tunnel run --token ey3HjoiNz...
```

Jalankan perintah pada server dan tunggu hingga selesai

```
vboxuser@DOKUMENTASI:~$ # Add cloudflare keykey
sudo mkdir -p --mode=0755 /usr/share/keyrings
curl -fsSL https://pkg.cloudflare.com/cloudflare-main.gpg | sudo tee /usr/share/keyrings/cloudflare-main.gpg >/dev/null

# Add this repo to your apt repositories
echo 'deb [signed-by=/usr/share/keyrings/cloudflare-main.gpg] https://pkg.cloudflare.com/cloudflared any main' | sudo tee
/etc/apt/sources.list.d/cloudflared.list

# install cloudflared
sudo apt-get update && sudo apt-get install cloudflared
[sudo] password for vboxuser:
deb [signed-by=/usr/share/keyrings/cloudflare-main.gpg] https://pkg.cloudflare.com/cloudflared any main
Hit:1 https://download.docker.com/linux/ubuntu noble InRelease
Get:2 https://pkg.cloudflare.com/cloudflare any InRelease [4,275 B]
Get:3 https://pkg.cloudflare.com/cloudflared any/main amd64 Packages [376 B]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:5 http://id.archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://id.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:8 http://id.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:12 http://id.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,090 kB]
Get:13 http://id.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [161 kB]
Get:14 http://id.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:15 http://id.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,062 kB]
Get:16 http://id.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [376 kB]
Get:17 http://id.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:18 http://id.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7,968 B]

vboxuser@DOKUMENTASI:~$ sudo cloudflared service install eyJhZjo1NzIjImJmNTBhZTZhYmUzLWQxZWVmfWwMzEzNGQ4MmYyZnUiLCJ0eio1NDOR
jNWuxYzYtOWNiOC09MjVjLTlIZDEtZjZmOWVlZjgzZDQxIiwicyI6IklUUUMLPV1JrLVRLVDE1ERTBzUZAwTVdFMUxXSTNOalF0TLRzWFlUWmpaVEpt0RJeCj
9
2025-05-19T08:51:20Z INF Using Systemd
2025-05-19T08:51:22Z INF Linux service for cloudflared installed successfully
```

Jika sudah selesai, maka status tunnel akan berubah menjadi healthy

Overview / Tunnels

Tunnels

Set up a secure connection between Cloudflare's global network and your infrastructure.

[Tunnel documentation](#)

Your tunnels Showing 1-2 of 2

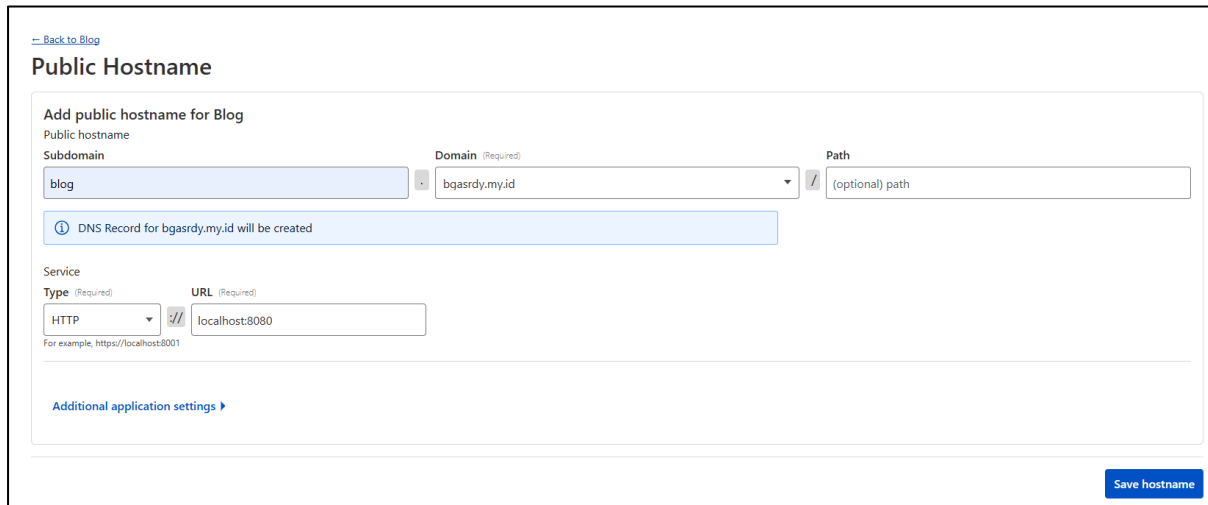
Manage the configurations of your existing tunnels.

[+ Create a tunnel](#)

Tunnel name ↑	Connector type	Connector ID	Tunnel ID	Routes	Status	Uptime
Blog	cloudflared	218816e7-71bf-4268-9a4f-e592175e20d1 🔗	7c19acab-140e-496f-9886-083bb74accd2	...	HEALTHY	...

Masuk ke konfigurasi, kemudian tambahkan Publik Hostname

Konfigurasi dan sesuaikan domain, subdomain, dan URL yang akan digunakan, kemudian *save hostname*



Penjelasan

- Subdomain : Bagian tambahan di depan domain utama yang digunakan sebagai pemisah atau pengelompokan layanan, seperti **blog.bgasrddy.my.id**
- Domain : Nama domain utama yang digunakan sebagai identitas web, misalnya **bgasrddy.my.id**
- Service Type : Jenis layanan yang dijalankan oleh server lokal, seperti HTTP dan HTTPS
- URL : Alamat IP lokal server yang menjalankan layanan beserta port yang digunakan, misalnya 192.168.20.13:8080.

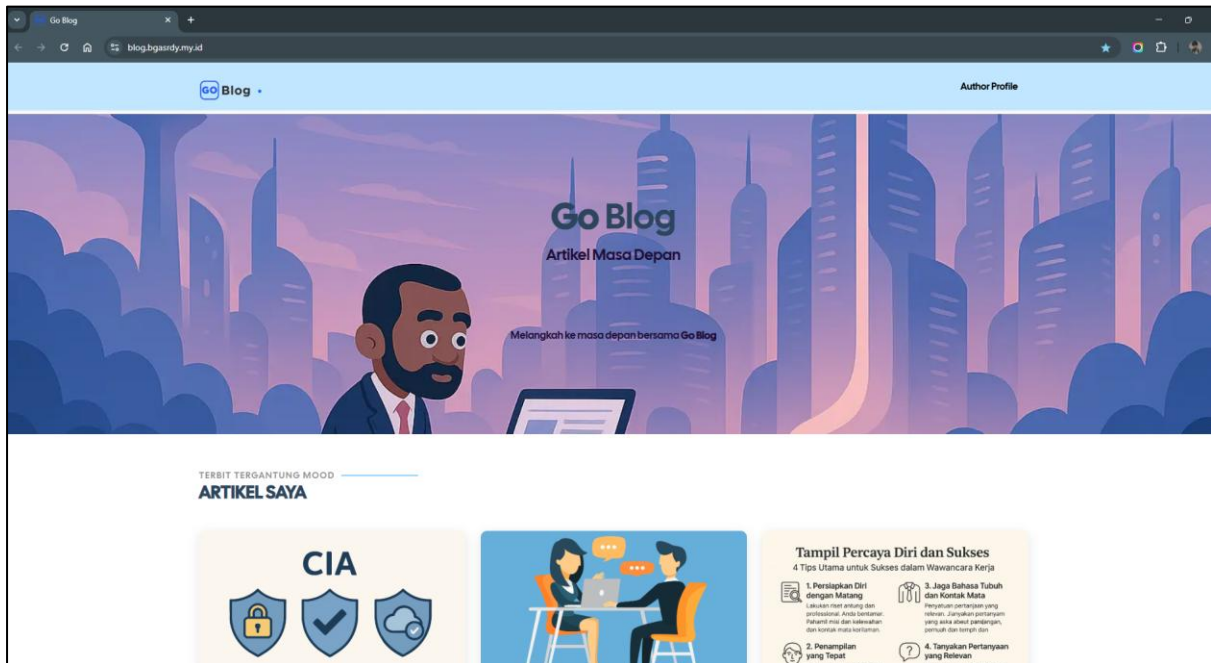


Setelah tersimpan, akses domain yang telah disambungkan dengan tunnel

blog.bgasrddy.my.id

BLOG PREVIEW

Halaman Utama



Artikel

