

# Blocky : une solution légère pour bloquer les publicités et les sites malveillants sur un réseau

Blocky se présente comme un ad-blocker et un proxy DNS pour le réseau local, qui va vous permettre de bloquer les publicités, les malwares et les sites web appartenant à certaines catégories (pornographie, réseaux sociaux, etc.). Léger, open source et écrit en Go, Blocky nécessite peu de ressources et sa configuration est flexible grâce à un fichier de configuration au format YAML.

Nous pouvons considérer Blocky comme un serveur DNS capable de contrôler et de filtrer les requêtes effectuées par les appareils connectés à un réseau. Il se comporte comme un DNS sinkhole, c'est-à-dire un DNS menteur, une technique bien connue. Elle consiste à rediriger certaines requêtes DNS vers une adresse contrôlée par l'administrateur réseau, au lieu de leur destination légitime, dans le but de bloquer certains domaines.

Par exemple, si une machine tente de se connecter à un site associé à une mauvaise réputation (malwares), le DNS sinkhole peut répondre avec une adresse IP comme 0.0.0.0 (ou l'adresse d'un serveur interne) pour que la connexion ne puisse pas aboutir. C'est une méthode efficace et reconnue pour protéger un réseau en bloquant certaines communications.

Voici quelques fonctionnalités clés de Blocky :

- Écrit en Go, Blocky est une application plus légère que d'autres alternatives (le binaire fait moins de 10 Mo)
- Bloquer (ou autoriser) les requêtes DNS à partir de listes externes ou directives statiques
- Gérer des règles par groupe de périphériques
- Blocage du domaine de la demande, du CNAME de la réponse (inspection approfondie du CNAME) et des adresses IP de la réponse
- Préchargement et mise en cache des requêtes souvent sollicitées par les clients
- Résolution DNS personnalisée pour certains noms de domaine
- Fonction de redirecteur conditionnel (utile pour votre nom de domaine local)
- Utilisation simultanée de plusieurs résolveurs externes
- Prise en charge de différents protocoles : DNS (non chiffré), DNS-over-HTTPS (DoH) et DNS-over-TLS (DoT)

## Environnement :

Machine virtuelle : Debian 12

Hyperviseur : UTM

## Prérequis, mettre l'utilisateur en mode root

CMD : su -

CMD : usermod -aG sudo raphael

CMD : visudo

```
raphael@debian:~$ su -
```

Mot de passe :

```
root@debian:~# usermod -aG sudo raphael
```

```
root@debian:~# visudo
```

Ajouté dans le fichier : raphael ALL=(ALL:ALL) ALL

```
# User privilege specification
```

```
root    ALL=(ALL:ALL) ALL
```

```
raphael  ALL=(ALL:ALL) ALL
```

## Installation de BLOCKY

CMD : sudo mkdir -p /opt/blocky

CMD : cd /opt/blocky

CMD : sudo wget

[https://github.com/0xERR0R/blocky/releases/download/v0.25/blocky\\_v0.25\\_Linux\\_arm64.tar.gz](https://github.com/0xERR0R/blocky/releases/download/v0.25/blocky_v0.25_Linux_arm64.tar.gz)

CMD : sudo tar -xvf blocky\_v0.25\_Linux\_arm64.tar.gz

```
raphael@debian:~$ sudo mkdir -p /opt/blocky
raphael@debian:~$ cd /opt/blocky/
raphael@debian:/opt/blocky$ sudo wget https://github.com/0xERR0R/blocky/releases/download/v0.25/blocky_v0.25_Linux_arm64.tar.gz
--2025-04-28 20:38:58-- https://github.com/0xERR0R/blocky/releases/download/v0.25/blocky_v0.25_Linux_arm64.tar.gz
Résolution de github.com (github.com)... 140.82.121.4
Connexion à github.com (github.com)[140.82.121.4]:443... connecté.
requête HTTP transmise, en attente de la réponse... 302 Found
Emplacement : https://objects.githubusercontent.com/github-production-release-asset-2e65be/219954477/8e05370f-9412-48cc-8bb6-9972201a57d2?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250428%2Fus-east-1%2F%3Faws4_request&X-Amz-Date=20250428T183858Z&X-Amz-Expires=300&X-Amz-Signature=1fd4325dcbdd046511c36bb8458017db0d393807677a35fdab257f0f642dbc7f&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dblocky_v0.25_Linux_arm64.tar.gz&response-content-type=application%2Foctet-stream [suivant]
--2025-04-28 20:38:58-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/219954477/8e05370f-9412-48cc-8bb6-9972201a57d2?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250428%2Fus-east-1%2F%3Faws4_request&X-Amz-Date=20250428T183858Z&X-Amz-Expires=300&X-Amz-Signature=1fd4325dcbdd046511c36bb8458017db0d393807677a35fdab257f0f642dbc7f&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dblocky_v0.25_Linux_arm64.tar.gz&response-content-type=application%2Foctet-stream
Résolution de objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Connexion à objects.githubusercontent.com (objects.githubusercontent.com)[185.199.109.133]:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 7799848 (7.4M) [application/octet-stream]
Sauvegarde en : « blocky_v0.25_Linux_arm64.tar.gz »

blocky_v0.25_Linux_arm64.tar.gz 100%[=====] 7,44M 5,32MB/s ds 1,4s

2025-04-28 20:39:00 (5,32 MB/s) - « blocky_v0.25_Linux_arm64.tar.gz » sauvegardé [7799848/7799848]

raphael@debian:/opt/blocky$ sudo tar -xvf blocky_v0.25_Linux_arm64.tar.gz
LICENSE
README.md
blocky
```

CMD : ls

CMD : sudo rm -rf blocky\_v0.25\_Linux\_arm64.tar.gz

CMD : ls

CMD : sudo setcap CAP\_NET\_BIND\_SERVICE=+eip /opt/blocky/blocky

CMD : sudo useradd -d /opt/blocky/blocky

```
raphael@debian:/opt/blocky$ ls
blocky  blocky_v0.25_Linux_arm64.tar.gz  LICENSE  README.md
raphael@debian:/opt/blocky$ sudo rm -rf blocky_v0.25_Linux_arm64.tar.gz
raphael@debian:/opt/blocky$ ls
blocky  LICENSE  README.md
raphael@debian:/opt/blocky$ sudo setcap CAP_NET_BIND_SERVICE=+eip /opt/blocky/blocky
raphael@debian:/opt/blocky$ sudo useradd -d /opt/blocky/blocky
```

CMD : sudo nano /etc/systemd/system/blocky.service

```
raphael@debian:/opt/blocky$ sudo nano /etc/systemd/system/blocky.service
```

Contenu :

[Unit]

Description=Blocky

After=syslog.target

After=network.target

[Service]

RestartSec=2s

Type=simple

User=blocky

Group=blocky

WorkingDirectory=/opt/blocky/

ExecStart=/opt/blocky/blocky -c config.yml

Restart=always

[Install]

WantedBy=multi-user.target

```

GNU nano 7.2 /etc/systemd/system/blocky.service *
[Unit]
Description=Blocky
After=syslog.target
After=network.target

[Service]
RestartSec=2s
Type=simple
User=blocky
Group=blocky
WorkingDirectory=/opt/blocky/
ExecStart=/opt/blocky/blocky -c config.yml
Restart=always

[Install]
WantedBy=multi-user.target

```

CMD : `sudo nano /opt/blocky/config.yml`

`raphael@debian: /opt/blocky$ sudo nano /opt/blocky/config.yml`

Contenu :

upstreams:

groups:

default:

- <https://cloudflare-dns.com/dns-query>
- <https://dns.google/dns-query>
- <https://dns.quad9.net/dns-query>

strategy: parallel\_best

timeout: 2s

connectIPVersion: v4

blocking:

denylists:

ads:

- <https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts>

porn:

- <https://raw.githubusercontent.com/StevenBlack/hosts/master/alternates/porn-only/hosts>

clientGroupsBlock:

default:

- ads
- porn

blockType: returnIp

blockIpAddress: 192.168.64.20

blockTTL: 1m

loading:

refreshPeriod: 6h

downloads:

timeout: 60s

attempts: 5

cooldown: 10s

concurrency: 16

strategy: fast

maxErrorsPerSource: 5

caching:

maxItemsCount: 1000  
prefetching: true  
prefetchExpires: 24h  
prefetchThreshold: 3  
prefetchMaxItemsCount: 1000

bootstrapDns:

- https://8.8.8.8/dns-query

queryLog:

type: csv-client

logRetentionDays: 31

# Log du service Blocky

# Journaliser les erreurs uniquement, par défaut "info"

log:

level: error

ports:

dns: 53

```
GNU nano 7.2 /opt/blocky/config.yml *
# Serveurs DNS en amont pour la résolution de noms
upstreams:
  groups:
    default:
      # DNS over HTTPS upstreams
      - https://cloudflare-dns.com/dns-query
      - https://dns.google/dns-query
      - https://dns.quad9.net/dns-query
      # Sélectionner les deux serveurs upstreams les plus rapides
      strategy: parallel_best
      timeout: 2s

# Connectivité IPv4 uniquement, sinon : v6, dual
connectIPv4Version: v4

# Déclarer les listes, ici des "denylists" = liste de blocage
# ads : liste de Steven Black pour les publicités
# porn : liste de Steven Black pour la pornographie
blocking:
  denylists:
    ads:
      - https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts
    porn:
      - https://raw.githubusercontent.com/StevenBlack/hosts/master/alternates/porn-only/hosts
  # Groupes de clients
  # Groupe par défaut, appliquer la liste "ads"
  # Groupe avec l'adresse IP 192.168.14.100, appliquer les listes "ads" et "porn"
  clientGroupsBlock:
    default:
      - ads
```

CMD : sudo chown -R blocky:blocky /opt/blocky

CMD : sudo systemctl enable blocky

CMD : sudo systemctl start blocky

CMD : sudo systemctl status blocky

```
raphael@debian:/opt/blocky$ sudo chown -R blocky:blocky /opt/blocky
raphael@debian:/opt/blocky$ sudo systemctl enable blocky
raphael@debian:/opt/blocky$ sudo systemctl start blocky
raphael@debian:/opt/blocky$ sudo systemctl status blocky
● blocky.service - Blocky
   Loaded: loaded (/etc/systemd/system/blocky.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-04-28 20:51:52 CEST; 1min 46s ago
     Main PID: 6911 (blocky)
       Tasks: 10 (limit: 4568)
      Memory: 53.2M
         CPU: 3.190s
    CGroup: /system.slice/blocky.service
            └─6911 /opt/blocky/blocky -c config.yml
```

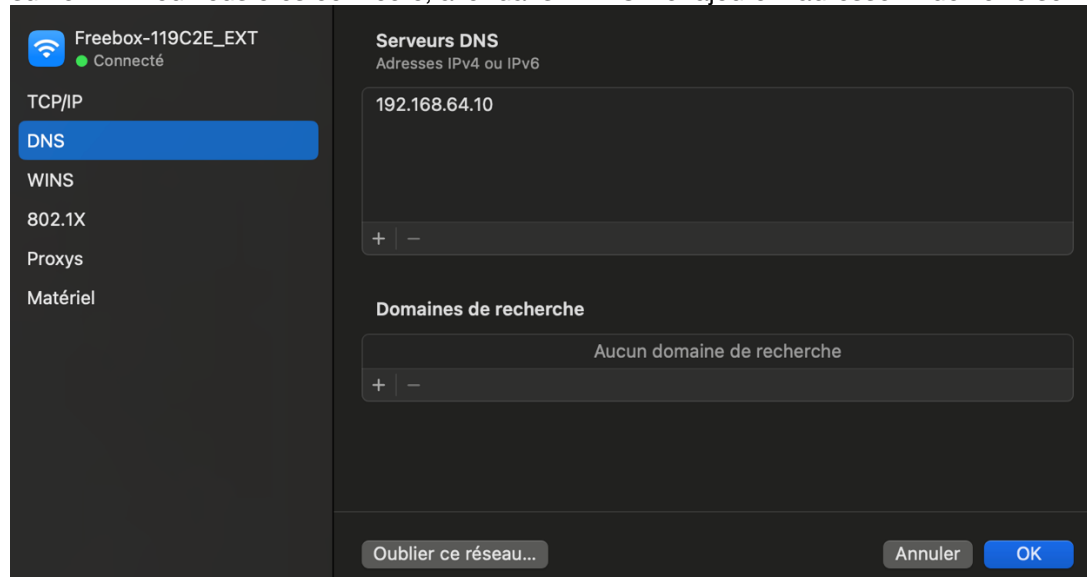
avril 28 20:51:52 debian systemd[1]: Started blocky.service - Blocky.

avril 28 20:51:52 debian blocky[6911]: - Grim reaper disabled, pid not 1

CMD : ip ad

```
raphael@debian:/opt/blocky$ ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 32:ef:53:a4:74:7e brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.10/24 brd 192.168.64.255 scope global dynamic noprefixroute enp0s1
        valid_lft 83751sec preferred_lft 83751sec
    inet6 fdd3:4017:fe26:9005:2137:d1e1:cd12:d5b1/64 scope global temporary dynamic
        valid_lft 602154sec preferred_lft 83217sec
    inet6 fdd3:4017:fe26:9005:30ef:53ff:fea4:747e/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 2591976sec preferred_lft 604776sec
    inet6 fe80::30ef:53ff:fea4:747e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Sur Mac, aller dans les réglages systèmes, puis dans « WI-FI », cliquer sur le bouton « détails » qui est sur le WI-FI ou vous êtes connecté, aller dans « DNS » et ajouter l'adresse IP de votre serveur :



On ne peut plus se rendre sur : <https://marketingplatform.google.com/about/enterprise/> sur des sites de publicités ou des sites pornographiques maintenant qu'on a mis en place ce serveur DNS avec Blocky :



## Configuration Prometheus & Grafana pour Blocky

CMD : sudo nano config.yml

```
raphael@debian:/opt/blocky$ sudo nano config.yml
```

Contenu :

ports:

dns: 53

http: 4000

prometheus:

enable: true

path: /metrics

```
GNU nano 7.2 config.yml
caching:
  maxItemsCount: 1000
  prefetching: true
  prefetchExpires: 24h
  prefetchThreshold: 3
  prefetchMaxItemsCount: 1000

# Résoudre les noms des serveurs upstreams
bootstrapDns:
  - https://8.8.8.8/dns-query

# Créer un fichier de log par client
# Conserver les données sur 31 jours
queryLog:
  type: csv-client
  logRetentionDays: 31

# Log du service Blocky
# Journaliser les erreurs uniquement, par défaut "info"
log:
  level: error

ports:
  dns: 53
  http: 4000

prometheus:
  enable: true
  path: /metrics
```

Aller sur le navigateur et marquer : localhost:4000/metrics

```
← → localhost:4000/metrics

# HELP blocky_blocking_enabled Blocking status
# TYPE blocky_blocking_enabled gauge
blocky_blocking_enabled 1
# HELP blocky_build_info Version number and build info
# TYPE blocky_build_info gauge
blocky_build_info{build_line="20250117-203209",version="v0.25"} 1
# HELP blocky_cache_entries Number of entries in cache
# TYPE blocky_cache_entries gauge
blocky_cache_entries 0
# HELP blocky_cache_hits_total Cache hit counter
# TYPE blocky_cache_hits_total counter
blocky_cache_hits_total 0
# HELP blocky_cache_misses_total Cache miss counter
# TYPE blocky_cache_misses_total counter
blocky_cache_misses_total 5
# HELP blocky_denylist_cache_entries Number of entries in the denylist cache
# TYPE blocky_denylist_cache_entries gauge
blocky_denylist_cache_entries{group="ads"} 162009
blocky_denylist_cache_entries{group="porn"} 75364
# HELP blocky_error_total Number of total errors
# TYPE blocky_error_total counter
blocky_error_total 0
# HELP blocky_failed_downloads_total Failed download counter
# TYPE blocky_failed_downloads_total counter
blocky_failed_downloads_total 0
# HELP blocky_last_list_group_refresh_timestamp_seconds Timestamp of last list refresh
# TYPE blocky_last_list_group_refresh_timestamp_seconds gauge
blocky_last_list_group_refresh_timestamp_seconds 1.760347804e9
# HELP blocky_prefetch_domain_name_cache_entries Number of entries in domain cache
# TYPE blocky_prefetch_domain_name_cache_entries gauge
blocky_prefetch_domain_name_cache_entries 0
# HELP blocky_prefetch_hits_total Prefetch hit counter
# TYPE blocky_prefetch_hits_total counter
blocky_prefetch_hits_total 4
# HELP blocky_prefetches_total Prefetch counter
# TYPE blocky_prefetches_total counter
blocky_prefetches_total 0
# HELP go_gc_duration_seconds A summary of the wall-time pause (stop-the-world) duration in garbage collection cycles.
# TYPE go_gc_duration_seconds summary
```

## Installation de Prometheus

CMD : sudo apt update

CMD : sudo apt install prometheus -y

raphael@debian:/opt/blocky\$ sudo apt update

Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease

Atteint :2 http://deb.debian.org/debian bookworm InRelease

Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease

Lecture des listes de paquets... Fait

Construction de l'arbre des dépendances... Fait

Lecture des informations d'état... Fait

28 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.

raphael@debian:/opt/blocky\$ sudo apt install prometheus -y

Lecture des listes de paquets... Fait

Construction de l'arbre des dépendances... Fait

Lecture des informations d'état... Fait

prometheus est déjà la version la plus récente (2.42.0+ds-5+b5).

0 mis à jour, 0 nouvellement installés, 0 à enlever et 28 non mis à jour.

```
raphael@debian:/opt/blocky$ sudo nano /etc/prometheus/prometheus.yml
```

```
- job_name: 'blocky'
  static_configs:
    - targets: ['localhost:4000']
```

```

GNU nano 7.2 /etc/prometheus/prometheus.yml
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  external_labels:
    monitor: 'example'

alerting:
  alertmanagers:
    - static_configs:
        - targets: ['localhost:9093']

rule_files:
scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    scrape_timeout: 5s
    static_configs:
      - targets: ['localhost:9090']
  - job_name: node
    static_configs:
      - targets: ['localhost:9100']
  - job_name: 'blocky'
    static_configs:
      - targets: ['localhost:4000']

```

CMD sudo systemctl status prometheus

```

raphael@debian:/opt/blocky$ sudo systemctl restart prometheus
raphael@debian:/opt/blocky$ sudo systemctl enable prometheus
Synchronizing state of prometheus.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable prometheus
raphael@debian:/opt/blocky$ sudo systemctl status prometheus
● prometheus.service - Monitoring system and time series database
   Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-05-04 20:55:54 CEST; 11s ago
     Docs: https://prometheus.io/docs/introduction/overview/
           man:prometheus(1)
   Main PID: 7131 (prometheus)
     Tasks: 9 (limit: 4568)
    Memory: 26.1M
       CPU: 130ms
    CGroup: /system.slice/prometheus.service
            └─7131 /usr/bin/prometheus

mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.717Z caller=head.go:685 level=info component=tsdb msg=WAL segment loaded"
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.718Z caller=head.go:685 level=info component=tsdb msg=WAL segment loaded"
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.718Z caller=head.go:685 level=info component=tsdb msg=WAL segment loaded"
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.718Z caller=head.go:722 level=info component=tsdb msg=WAL replay completed"
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.719Z caller=main.go:1069 level=info fs_type=EXT4_SUPER_MAGIC
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.719Z caller=main.go:1012 level=info msg="ISDB started"
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.719Z caller=main.go:1192 level=info msg="Loading configuration file" filename=
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.720Z caller=main.go:1229 level=info msg="Completed loading of configuration"
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.720Z caller=main.go:973 level=info msg="Server is ready to receive web requests"
mai 04 20:55:54 debian prometheus[7131]: ts=2025-05-04T18:55:54.720Z caller=manager.go:978 level=info component="rule manager" msg="Starting
lines 1-22/22 (END)

```

CMD : `sudo apt install -y apt-transport-https software-properties-common`

```

raphael@debian: /opt/blocky$ sudo apt install -y apt-transport-https software-properties-common
[sudo] Mot de passe de raphael :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
software-properties-common est déjà la version la plus récente (0.99.30-4.1~deb12u1).
software-properties-common passé en « installé manuellement ».
Les NOUVEAUX paquets suivants seront installés :
  apt-transport-https
0 mis à jour, 1 nouvellement installés, 0 à enlever et 28 non mis à jour.
Il est nécessaire de prendre 25,2 ko dans les archives.
Après cette opération, 35,8 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://debian.proxad.net/debian bookworm/main arm64 apt-transport-https all 2.6.1 [25,2 kB]
25,2 ko réceptionnés en 0s (372 ko/s)
Sélection du paquet apt-transport-https précédemment désélectionné.
(Lecture de la base de données... 156722 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../apt-transport-https-2.6.1_all.deb ...
Dépaquetage de apt-transport-https (2.6.1) ...
Paramétrage de apt-transport-https (2.6.1) ...

```



```

CMD : sudo mkdir -p /etc/apt/keyrings
CMD : sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo gpg --dearmor -o
/etc/apt/keyrings/grafana.gpg
CMD : echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://packages.grafana.com/oss/deb
stable main" | \ sudo tee /etc/apt/sources.list.d/grafana.list > /dev/null

```

```

raphael@debian:/opt/blocky$ sudo mkdir -p /etc/apt/keyrings
raphael@debian:/opt/blocky$ sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/grafana.gpg
raphael@debian:/opt/blocky$
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://packages.grafana.com/oss/deb stable main" | \
sudo tee /etc/apt/sources.list.d/grafana.list > /dev/null

```

```

CMD : sudo apt update
CMD : sudo apt install grafana -y
CMD : sudo systemctl enable --now grafana-server

```

```

raphael@debian:/opt/blocky$ sudo apt update
Atteint :1 http://debian.proxad.net/debian bookworm InRelease
Atteint :2 http://debian.proxad.net/debian bookworm-updates InRelease
Atteint :3 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :4 https://packages.grafana.com/oss/deb stable InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
28 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
raphael@debian:/opt/blocky$ sudo apt install grafana -y
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
grafana est déjà la version la plus récente (11.6.1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 28 non mis à jour.
raphael@debian:/opt/blocky$ sudo systemctl enable --now grafana-server
Synchronizing state of grafana-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /lib/systemd/system/grafana-server.service.

```

```

CMD : sudo systemctl status grafana-server

```

```

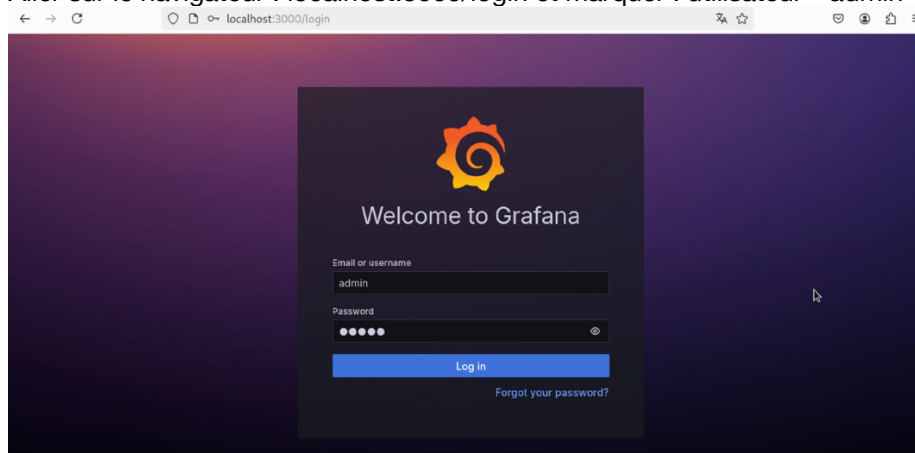
raphael@debian:/opt/blocky$ sudo systemctl status grafana-server
• grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-05-04 20:47:01 CEST; 9min ago
     Docs: http://docs.grafana.org
   Main PID: 6111 (grafana)
    Tasks: 19 (limit: 4568)
   Memory: 82.9M
      CPU: 5.900s
   CGroup: /system.slice/grafana-server.service
           └─6111 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --pac

mai 04 20:47:06 debian grafana[6111]: logger=installer.fs t=2025-05-04T20:47:06.820162519+02:00 level=info msg="Downloaded and extracted gra
mai 04 20:47:06 debian grafana[6111]: logger=plugins.registration t=2025-05-04T20:47:06.848345418+02:00 level=info msg="Plugin registered"
mai 04 20:47:06 debian grafana[6111]: logger=plugin.backgroundinstaller t=2025-05-04T20:47:06.848395862+02:00 level=info msg="Plugin succes
mai 04 20:47:06 debian grafana[6111]: logger=plugin.backgroundinstaller t=2025-05-04T20:47:06.848426311+02:00 level=info msg="Installing plu
mai 04 20:47:07 debian grafana[6111]: logger=plugin.installer t=2025-05-04T20:47:07.231133178+02:00 level=info msg="Installing plugin" plug
mai 04 20:47:07 debian grafana[6111]: logger=installer.fs t=2025-05-04T20:47:07.305441351+02:00 level=info msg="Downloaded and extracted gra
mai 04 20:47:07 debian grafana[6111]: logger=plugins.registration t=2025-05-04T20:47:07.354486168+02:00 level=info msg="Plugin registered"
mai 04 20:47:07 debian grafana[6111]: logger=plugin.backgroundinstaller t=2025-05-04T20:47:07.35454515+02:00 level=info msg="Plugin success
mai 04 20:47:27 debian grafana[6111]: logger=context.userId=0 orgId=0 uname= t=2025-05-04T20:47:27.633912986+02:00 level=info msg="Request t
mai 04 20:48:26 debian grafana[6111]: logger=infra.usagstats t=2025-05-04T20:48:26.072885365+02:00 level=info msg="Usage stats are ready t
lines 1-21/21 (END)

```

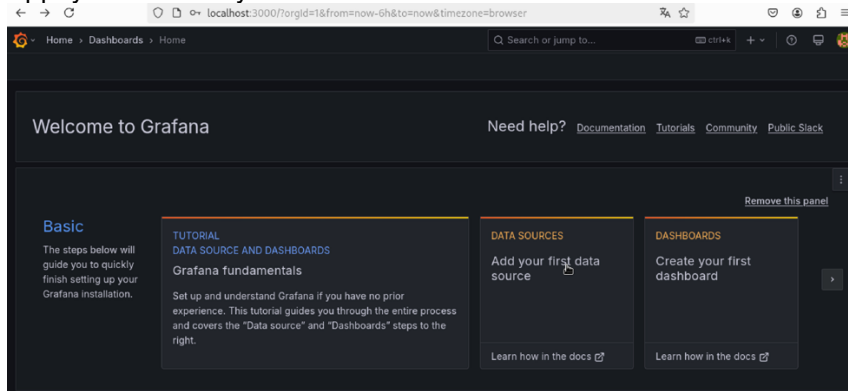
## Configuration Grafana

Aller sur le navigateur : localhost:3000/login et marquer : utilisateur « admin » et mot de passe « admin » :

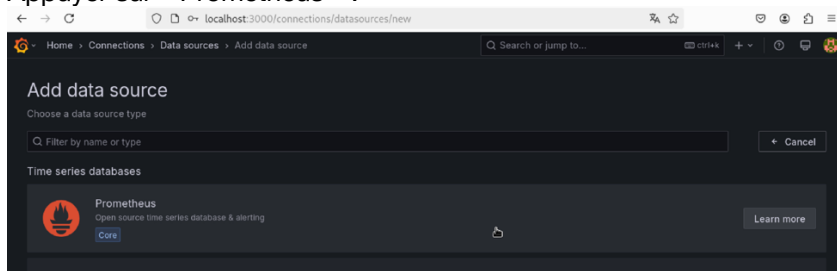




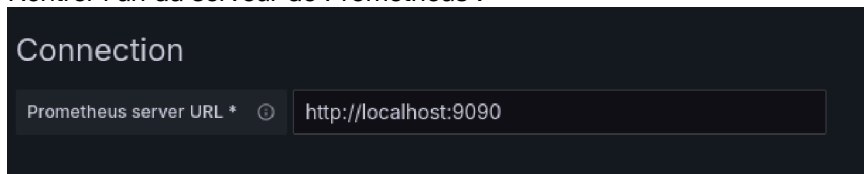
Appuyer sur « Add your first data » :



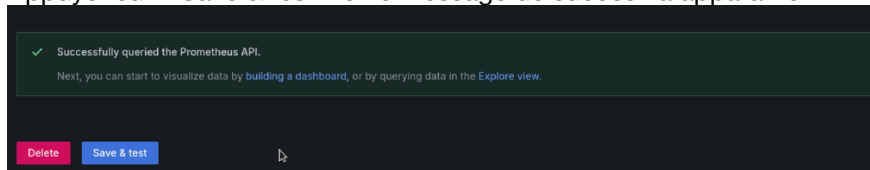
Appuyer sur « Prometheus » :



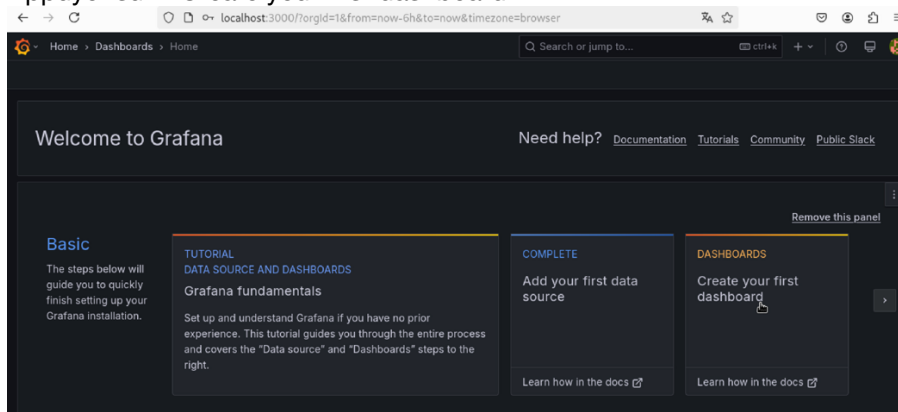
Rentrer l'URL du serveur de Prometheus :



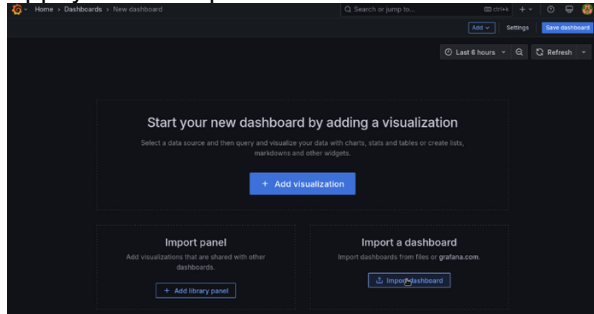
Appuyer sur « Save & test » et le message de succès va apparaître :



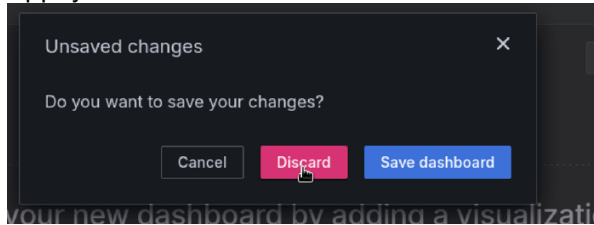
Appuyer sur « Create your first dashboard » :



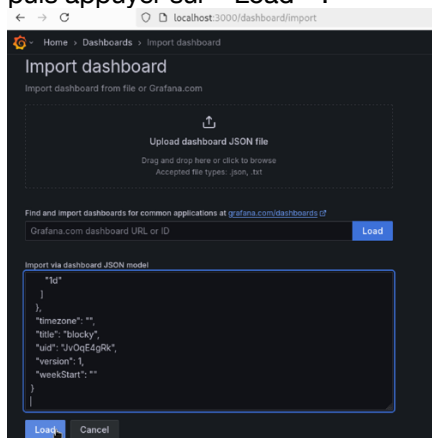
Appuyer sur « Import Dashboard » :



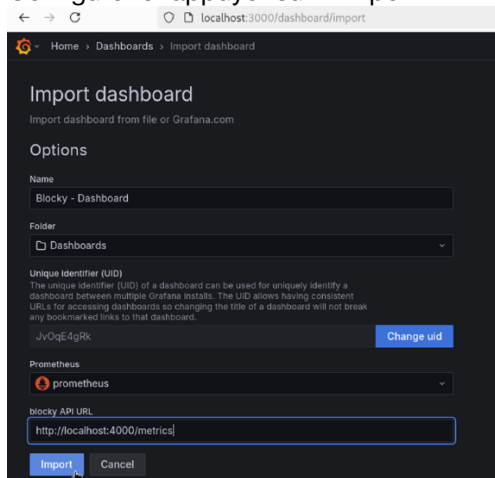
Appuyer sur « Discard » :



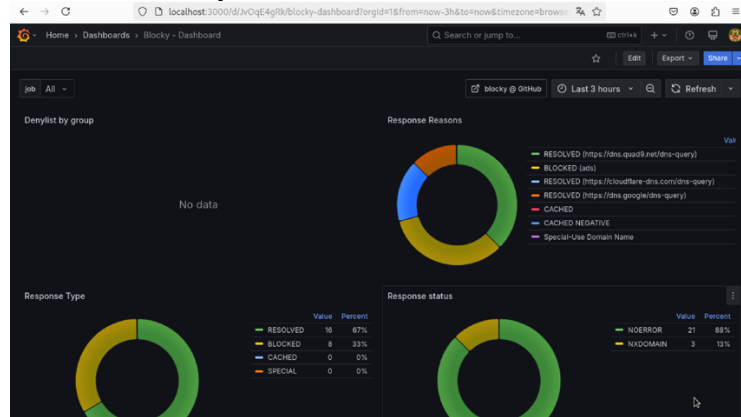
Copier le fichier JSON : <https://0xerr0r.github.io/blocky/latest/blocky-grafana.json> et coller le ci-dessous, puis appuyer sur « Load » :



Configurer et appuyer sur « Import »



## Dashboard Blocky :



## Essayer une autre solution pour gérer la redirection

### Redirection vers une page quand l'url est bloquée

CMD : `sudo apt install nginx`

`raphael@debian:~$ sudo apt install nginx`

Lecture des listes de paquets... Fait

Construction de l'arbre des dépendances... Fait

Lecture des informations d'état... Fait

nginx est déjà la version la plus récente (1.22.1-9+deb12u1).

0 mis à jour, 0 nouvellement installés, 0 à enlever et 1 non mis à jour.

CMD : `sudo nano /var/www/html/blocked.html`

`raphael@debian:~$ sudo nano /var/www/html/blocked.html`

### Contenu :

```
GNU nano 7.2 /var/www/html/blocked.html
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Site Bloqué</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      background-color: #f4f4f4;
      color: #333;
    }
    h1 {
      color: red;
    }
    p {
      font-size: 1.2em;
    }
  </style>
</head>
<body>
  <h1>Accès Bloqué</h1>
  <p>Ce site a été bloqué par Blocky.</p>
  <p>Nous vous recommandons de vérifier votre connexion ou de contacter l'administrateur réseau.</p>
</body>
</html>
```

CMD : `sudo nano /etc/nginx/sites-available/default`

`raphael@debian:~$ sudo nano /etc/nginx/sites-available/default`

### Contenu :

```
GNU nano 7.2 /etc/nginx/sites-available/default
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    index blocked.html;

    error_page 403 /blocked.html;

    location / {
        set $blocked 0;
        if ($remote_addr = "192.168.64.10") {
            set $blocked 1;
        }

        if ($blocked = 1) {
            return 403;
        }

        try_files $uri $uri/ =404;
    }
}
```

CMD : sudo systemctl restart nginx

**raphael@debian:~\$** sudo systemctl restart blocky

Aller sur 127.0.0.1 sur votre navigateur de votre VM pour voir la page :

