

# Tips - Commandes

- [Tester les ports ouverts sur un serveur avec NMAP](#)
- [Gestion de la queue postfix](#)
- [Réaliser un Speedtest sur un serveur Linux avec iPerf](#)
- [Gestion d'un site Wordpress avec WP-CLI](#)
- [Gestion de plateformes avec Terraform](#)
- [Postfix - Gestion de la mail queue](#)
- [Configuration Ratios ruTorrent](#)
- [Trier un fichier CSV en retirant les doublons, basés sur la première colonne](#)
- [shell request failed on channel 0](#)
- [Configuration Cloudflare terraform](#)
- [Installer borgmatic et borgbackup sur Ubuntu](#)

# Tester les ports ouverts sur un serveur avec NMAP

Tester un port spécifique avec nmap

```
nmap -p 80 REMOTE_IP
```

Tester tous les ports ouverts avec nmap

```
nmap -p- REMOTE_IP
```

# Gestion de la queue postfix

## Lister les messages en queue

```
postqueue -p
```

## Supprimer un message en queue

```
postsuper -d DBB3F1A7
```

## Supprimer tous les messages en queue

```
postsuper -d ALL
```

## Mettre un message en attente

```
postsuper -h DBA3F1A7
```

## Remettre un message en mode normal

```
postsuper -H DBA3F1A7
```

## Afficher le contenu d'un message

```
postcat -q DBA3F1A9
```

## Forcer l'envoi des messages en queue

```
postqueue -f
```



# Réaliser un Speedtest sur un serveur Linux avec iPerf

## Speedtest avec speedtest-cli

La solution la plus simple pour réaliser un test de débit sur linux est d'utiliser **speedtest-cli** :

```
pip3 install --upgrade speedtest-cli
```

## Utilisation de l'outil

Lister les server les plus proches, pour réaliser le test de débit

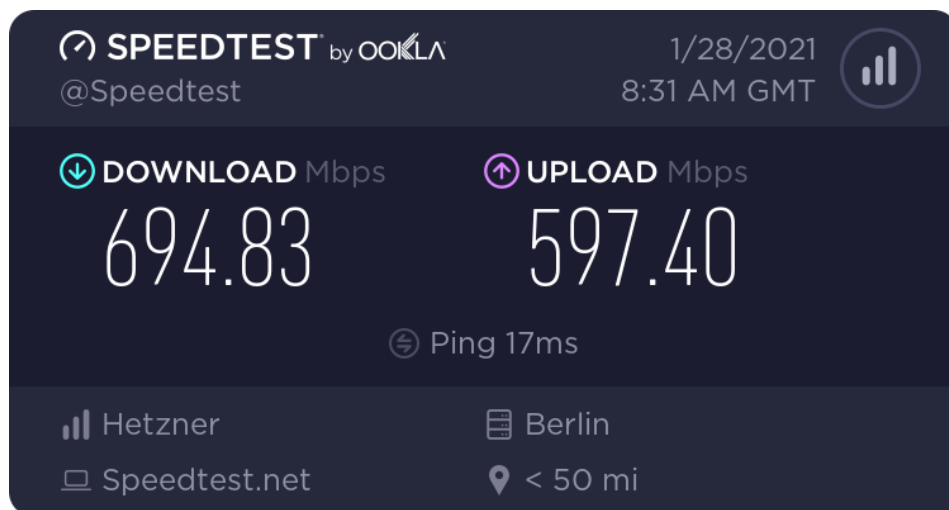
```
speedtest-cli --list
```

Réaliser un test basique et récupérer l'image du résultat

```
speedtest-cli --share
```

```
root@srvtest ~ $ speedtest-cli --
share
    master
Retrieving speedtest.net configuration...
Testing from Hetzner Online GmbH (138.201.68.55)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by Cronon GmbH (Berlin) [1.64 km]: 17.256 ms
Testing download speed.....
Download: 694.83 Mbit/s
Testing upload speed.....
Upload: 597.40 Mbit/s
```

Share results: <http://www.speedtest.net/result/10827725744.png>



## Réaliser un test de débit en spécifiant le serveur à utiliser

Identifiez le server ID sur lequel vous souhaitez pointer avec la commande **--list**, et lancez le speed-test avec l'option **--server** :

```
root@srvtest ~ $ speedtest-cli --server
18720
↵ master
Retrieving speedtest.net configuration...
Testing from Hetzner Online GmbH (138.201.68.55)...
Retrieving speedtest.net server list...
Retrieving information for the selected server...
Hosted by SATAN s.r.o. (Trutnov) [279.56 km]: 29.058 ms
Testing download speed.....
Download: 762.45 Mbit/s
Testing upload speed.....
Upload: 612.23 Mbit/s
```

1

# Speedtest avec iPerf3

## Installation d'iPerf3

iPerf3 est disponible dans les repos Debian/Ubuntu/CentOS mais peut également être installé via Python PIP. Nous allons récupérer le paquet dans les repos :

```
apt install iperf3
```

# Utilisation d'iPerf3 pour réaliser un test de débit entre deux hosts

L'outil utilise par défaut le port 5201 en TCP et UDP. Ouvrez donc ce port sur la machine cible :

```
ufw allow 5201/tcp
ufw allow 5201/udp
```

Lancez le mode server sur la machine cible :

```
iperf3 -s
```

Sur la machine source, lancez votre test de débit en spécifiant la machine cible :

```
iperf3 -c 192.168.1.10
```

Vous verrez ainsi passer les échanges sur les deux machines et une moyenne vous sera affichée :

```
Accepted connection from 192.168.1.11, port 38872
[ 5] local 192.168.1.10 port 5201 connected to 192.168.1.11 port 38874
[ ID] Interval      Transfer   Bitrate
[ 5]  0.00-1.00  sec  1.00 GBytes 8.62 Gbits/sec
[ 5]  1.00-2.00  sec   990 MBytes 8.31 Gbits/sec
[ 5]  2.00-3.00  sec  1012 MBytes 8.49 Gbits/sec
[ 5]  3.00-4.00  sec  1003 MBytes 8.41 Gbits/sec
[ 5]  4.00-5.00  sec  1006 MBytes 8.44 Gbits/sec
[ 5]  5.00-6.00  sec   1.07 GBytes 9.16 Gbits/sec
[ 5]  6.00-7.00  sec  1001 MBytes 8.39 Gbits/sec
[ 5]  7.00-8.00  sec   982 MBytes 8.24 Gbits/sec
[ 5]  8.00-9.00  sec   1.04 GBytes 8.92 Gbits/sec
[ 5]  9.00-10.00 sec   993 MBytes 8.33 Gbits/sec
[ 5] 10.00-10.00 sec   1.25 MBytes 7.64 Gbits/sec
-----
[ ID] Interval      Transfer   Bitrate
[ 5]  0.00-10.00 sec   9.93 GBytes 8.53 Gbits/sec          receiver
```

## Réalisation du test iPerf3 avec Docker

Sur la machine hôte, lancez le docker iPerfs comme ceci :

```
docker run -it --rm --network=host --name=iperf3-server -p 5201:5201 networkstatic/iperf3 -s
```

Ensuite, sur le client, lancez le test en pointant sur l'IP de la machine hôte :

```
docker run -it --rm networkstatic/iperf3 -c 192.168.1.10
```

# Gestion d'un site Wordpress avec WP-CLI

## Installation du binaire

```
curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
php wp-cli.phar --info
chmod +x wp-cli.phar
sudo mv wp-cli.phar /usr/local/bin/wp
which wp
```

## Mise à jour du binaire

```
wp cli update
```

## Gestion d'un site localement

Pour administrer un site localement, il est nécessaire d'être positionné dans le dossier du site, ou de le spécifier avec l'option **path** :

```
wp user list --path=/var/www/html/monsite.com --format=json
```

## Gestion d'un site à distance

Pour administrer un site à distance, il est nécessaire d'établir un lien SSH sur le serveur distant, ou d'établir une connexion identifiant / mot de passe :

# Gestion de plateformes avec Terraform

## Installation de Terraform

```
wget https://releases.hashicorp.com/terraform/0.14.5/terraform_0.14.5_linux_amd64.zip  
unzip terraform_0.14.5_linux_amd64.zip  
mv terraform ~/.local/bin/terraform
```

## Installation de tfswitcher pour gérer plusieurs versions

```
curl -L https://raw.githubusercontent.com/warrensbbox/terraform-switcher/release/install.sh | sudo bash
```

## Commandes de base Terraform

### Switcher de version Terraform

```
tfswitch 0.11.2
```

### Préparer l'environnement Terraform

```
terraform init
```

### Remise en forme des fichiers

```
terraform fmt
```

### Validation du code

```
terraform validate
```

## Préparation du plan

```
terraform plan -out plan.json
```

## Application du plan

```
terraform apply plan.json
```

# Postfix - Gestion de la mail queue

## Afficher la liste des mails en queue

```
mailq  
postqueue -p
```

## Afficher le contenu - headers d'un mail en queue

```
postcat -vq EC4D438AA57
```

## Vider la mail queue

```
postsuper -d ALL
```

Pour ne vider que les mails en statut **deferred** :

```
postsuper -d ALL deferred
```

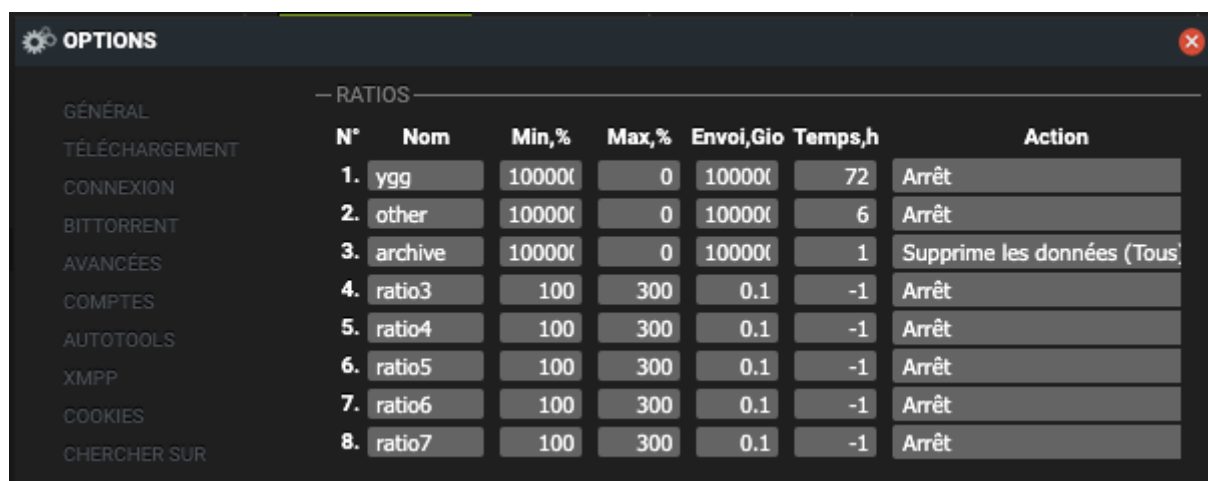
## Forcer l'envoi des mails en queue

```
postqueue -f  
sendmail -q
```

# Configuration Ratios ruTorrent

Configuration des règles de ratio ruTorrent pour l'automatisation des suppression et seed 72h.

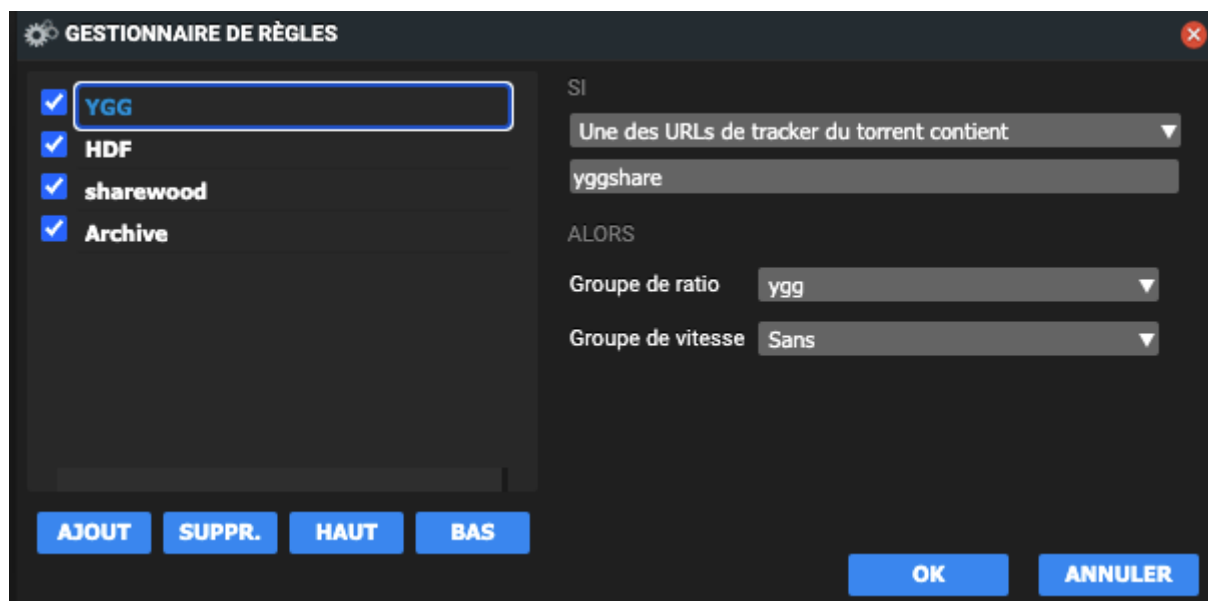
## Paramètre des ratios



The screenshot shows the 'OPTIONS' dialog box with the 'RATIOS' tab selected. On the left is a sidebar with menu items: GÉNÉRAL, TÉLÉCHARGEMENT, CONNEXION, BITTORRENT, AVANCÉES, COMPTES, AUTOTOOLS, XMPP, COOKIES, and CHERCHER SUR. The main area displays a table of ratio rules.

N°	Nom	Min, %	Max, %	Envoi, Gio	Temps, h	Action
1.	ygg	100000	0	100000	72	Arrêt
2.	other	100000	0	100000	6	Arrêt
3.	archive	100000	0	100000	1	Supprime les données (Tous
4.	ratio3	100	300	0.1	-1	Arrêt
5.	ratio4	100	300	0.1	-1	Arrêt
6.	ratio5	100	300	0.1	-1	Arrêt
7.	ratio6	100	300	0.1	-1	Arrêt
8.	ratio7	100	300	0.1	-1	Arrêt

## Règles de ratios



The screenshot shows the 'GESTIONNAIRE DE RÈGLES' dialog box. On the left, a list of rules with checkboxes: YGG, HDF, sharewood, and Archive. On the right, the configuration for the selected rule 'YGG' is shown. It includes a condition 'SI' (If) and an action 'ALORS' (Then).

**SI**

Une des URLs de tracker du torrent contient  
yggshare

**ALORS**

Groupe de ratio: ygg  
Groupe de vitesse: Sans

Buttons at the bottom: AJOUT, SUPPR., HAUT, BAS, OK, ANNULER.

GESTIONNAIRE DE RÈGLES

☒ YGG

☒ HDF

☒ sharewood

☒ Archive

AJOUT

SUPPR.

HAUT

BAS

SI

L'étiquette du torrent contient

▼

Archive

ALORS

Groupe de ratio

archive

▼

Groupe de vitesse

Sans

▼

OK

ANNULER

# Trier un fichier CSV en retirant les doublons, basés sur la première colonne

Commande :

```
sort -u -t',' -k1,1 monfichier.csv
```

Doc : <https://unix.stackexchange.com/questions/171091/remove-lines-based-on-duplicates-within-one-column-without-sort>

# shell request failed on channel 0

```
ssh user@host -- 'mount -o remount,rw /dev/pts'
```

# Configuration Cloudflare terraform

Ensemble de configurations Terraform pour la gestion de domaines, dns, cache et firewall rules sur Cloudflare

## Vars

```
variable "zone_name" {}

variable "zone_id" {}

variable "bing_verify" {
  default    = "unset"
  description = "TXT record DNS content for Bing Verify"
}

variable "brotli" {
  default    = "on"
  description = "Enable or not brotli compression"
}

variable "minify_css" {
  default    = "off"
  description = "Minify or not CSS for zone settings"
}

variable "minify_html" {
  default    = "off"
  description = "Minify or not HTML for zone settings"
}

variable "minify_js" {
  default    = "off"
  description = "Minify or not JS for zone settings"
}

variable "always_online" {
  default    = "on"
  description = "Enable or not Always Online"
}
```

```

variable "devmode" {
  default    = "off"
  description = "Enable or disable Dev Mode on cloudflare"
}
variable "additional_spf" {
  default    = ""
  description = "Additional spf configuration for TXT DNS record"
}
variable "reject_spf" {
  default    = "~"
  description = "SPF reject mode for TXT DNS Record"
}
variable "additional_dmarc" {
  default    = ""
  description = "Additional dmarc configuration for TXT DNS record"
}
variable "root_record" {
  default    = ""
  description = "DNS root record IP address"
}
variable "root_ipv4" {
  default    = ""
  description = ""
}
variable "alias_domain" {
  default    = ""
  description = "Secondary alias domain"
}
variable "main_domain" {
  default    = ""
  description = "Principal domain name"
}

```

## Zone settings

```

resource "cloudflare_zone_settings_override" "settings" {
  zone_id = var.zone_id

```

```

settings {
  always_online      = "on"
  always_use_https   = "off"
  automatic_https_rewrites = "off"
  brotli             = "on"
  cache_level        = "basic"
  development_mode    = var.devmode
  email_obfuscation   = "off"
  http3              = "on"
  browser_cache_ttl   = 0
  early_hints         = "off"
  ip_geolocation      = "on"
  ipv6               = "on"
  max_upload          = 100
  min_tls_version     = "1.2"
  pseudo_ipv4         = "off"
  rocket_loader       = "off"
  ssl                 = "strict"
  minify {
    css = var.minify_css
    js  = var.minify_js
    html = var.minify_html
  }
}

```

## Firewall rules

```

resource "cloudflare_ruleset" "bwa_custom_restrictions" {
  zone_id  = var.zone_id
  name     = "BLDWebAgency Firewall Rules"
  description = "BWA set of rules to protect websites against ddos"
  kind     = "zone"
  phase    = "http_request_firewall_custom"

  rules {
    action = "skip"
  }
}

```

```

action_parameters {
  phases = ["http_request_firewall_managed", "http_request_sbfm"]
  ruleset = "current"
}
description = "Allow Safe places"
enabled     = true
expression  = "(ip.src eq 82.66.241.38) or (cf.client.bot) or (http.request.uri.query contains
\"trustindex_reviews_hook_google\") or (http.request.uri.path contains \".ico\") or (http.user_agent contains
\"bitlybot\") or (http.user_agent contains \"updown.io daemon 2.8\") or (http.request.uri.path contains
\"favicon\") or (http.user_agent contains \"DuckDuckGo\") or (http.user_agent contains \"Pingdom\") or
(http.user_agent contains \"PetalBot\") or (http.user_agent contains \"CFNetwork\") or (http.user_agent contains
\"qwant.com\") or (http.user_agent contains \"bingbot\") or (http.user_agent contains \"updown.io daemon 2.6\")
or (http.user_agent contains \"Stripe/1.0\") or (ip.src eq 3.18.12.63) or (ip.src eq 3.130.192.231) or (ip.src eq
13.235.14.237) or (ip.src eq 13.235.122.149) or (ip.src eq 109.234.160.247) or (ip.src eq 18.211.135.69) or
(ip.src eq 35.154.171.200) or (ip.src eq 52.15.183.38) or (ip.src eq 54.88.130.119) or (ip.src eq 54.88.130.237)
or (ip.src eq 54.187.174.169) or (ip.src eq 54.187.205.235) or (ip.src eq 54.187.216.72) or (ip.src eq
163.172.33.112)\"
  logging {
    enabled = true
  }
}
rules {
  description = "Restrict referer for WP Paths"
  action      = "managed_challenge"
  expression  = "(http.request.uri eq \"/xmlrpc.php\") or (http.request.uri.path contains \"/wp-content/\" and not
http.referer contains \"${var.zone_name}\") or (http.request.uri.path contains \"/wp-includes/\" and not
http.referer contains \"${var.zone_name}\")\"
  enabled     = true
}
rules {
  description = "Challenge wp-admin out of France"
  action      = "managed_challenge"
  enabled     = true
  expression  = "(http.request.uri.path contains \"/wp-login.php\" and ip.geoip.country ne \"FR\") or
(http.request.uri.query contains \"action=lostpassword\" and http.referer ne \"${var.zone_name}\")\"
}
rules {
  description = "Restrict some WP Path and countries"
  action      = "managed_challenge"

```

```

    expression = "(ip.geoip.country in {\\"SG\\" \\"BR\\" \\"RU\\" \\"CN\\" \\"IQ\\" \\"AZ\\" \\"SG\\" \\"AF\\"}) or (http.request.uri
contains \"/wp-comments-post.php\\" and http.request.method eq \\"POST\\" and not http.referer contains
\\"${var.zone_name}\\")"
    enabled    = true
}
rules {
  description = "Block bad bots"
  action      = "managed_challenge"
  expression  = "(http.user_agent eq \\\"\\") or (http.user_agent contains \\\"muckrack\\\") or (http.user_agent
contains \\\"Sogou\\\") or (http.user_agent contains \\\"BUbiNG\\\") or (http.user_agent contains \\\"knowledge\\\") or
(http.user_agent contains \\\"CFNetwork\\\") or (http.user_agent contains \\\"Scrapy\\\") or (http.user_agent contains
\\\"SemrushBot\\\") or (http.user_agent contains \\\"AhrefsBot\\\") or (http.user_agent contains \\\"Baiduspider\\\") or
(http.user_agent contains \\\"python-requests\\\") or (http.user_agent contains \\\"crawl\\" and not cf.client.bot) or
(http.user_agent contains \\\"Crawl\\" and not cf.client.bot) or (http.user_agent contains \\\"bot\\" and not
http.user_agent contains \\\"bingbot\\" and not http.user_agent contains \\\"Google\\" and not http.user_agent
contains \\\"Twitter\\" and not cf.client.bot) or (http.user_agent contains \\\"Bot\\" and not http.user_agent contains
\\\"Google\\" and not cf.client.bot) or (http.user_agent contains \\\"Spider\\" and not cf.client.bot) or (http.user_agent
contains \\\"spider\\" and not cf.client.bot)"
  enabled    = true
}
}

```

## Wordpress cache rules

```

resource "cloudflare_ruleset" "custom_bwa_cache_ruleset" {
  zone_id = var.zone_id
  kind    = "zone"
  name    = "default"
  phase   = "http_request_cache_settings"
  rules {
    action = "set_cache_settings"
    action_parameters {
      browser_ttl {
        mode = "respect_origin"
      }
      cache = false
    }
  }
  description = "Skip admin pages"
}

```

```
enabled    = true
expression = "(http.request.uri.path contains \"wp-admin\") or (http.request.uri.path contains \"wp-login\") or
(http.request.uri.path contains \"bwa35-login\")"
}
rules {
  action    = "set_cache_settings"
  description = "Cache static assets"
  enabled    = true
  expression = "(http.request.uri.path contains \".webp\") or (http.request.uri.path contains \".avif\") or
(http.request.uri.path contains \".woff\") or (http.request.uri.path contains \".woff2\") or (http.request.uri.path
contains \".png\") or (http.request.uri.path contains \".svg\") or (http.request.uri.path contains \".jpeg\") or
(http.request.uri.path contains \".jpg\") or (http.request.uri.path contains \".js\") or (http.request.uri.path
contains \".css\")"
  action_parameters {
    browser_ttl {
      mode = "respect_origin"
    }
    cache = true
    cache_key {
      cache_deception_armor = false
      custom_key {
        query_string {
          exclude = ["*"]
        }
      }
    }
    ignore_query_strings_order = true
  }
  edge_ttl {
    default = 2678400
    mode    = "override_origin"
  }
  origin_error_page_passthru = true
  serve_stale {
    disable_stale_while Updating = true
  }
}
rules {
  action = "set_cache_settings"
```

```

action_parameters {
  browser_ttl {
    default = 14400
    mode    = "override_origin"
  }
  cache = true
  edge_ttl {
    default = 172800
    mode    = "override_origin"
  }
}
description = "Full cache on uploads"
enabled      = true
expression   = "(http.request.uri.path contains \"/wp-content/uploads/\")"
}

```

## Redirection vers le domaine principal

```

resource "cloudflare_ruleset" "redirect_to_main_domain" {
  zone_id    = var.zone_id
  name       = "redirects"
  description = "Redirect to main domain"
  kind       = "zone"
  phase      = "http_request_dynamic_redirect"

  rules {
    action = "redirect"
    action_parameters {
      from_value {
        status_code = 301
        target_url {
          value = "https://${var.main_domain}"
        }
        preserve_query_string = false
      }
    }
  }
  expression = "(http.host eq \"${var.alias_domain}\")"
}

```

```
description = "Redirecte to main domain"
enabled     = true
}
}
```

## Ruleset et Redirect list au niveau account

```
variable "account_id" {
  default = "XXXXXX"
}

resource "cloudflare_ruleset" "redirects_ruleset" {
  account_id = var.account_id
  name       = "Redirects Ruleset"
  description = "Ruleset for redirects list"
  kind       = "root"
  phase      = "http_request_redirect"

  rules {
    action = "redirect"
    action_parameters {
      from_list {
        name = "redirect_list"
        key  = "http.request.full_uri"
      }
    }
    expression = "http.request.full_uri in $redirect_list"
    description = "Apply redirects from redirect_list list"
    enabled     = true
  }
}

resource "cloudflare_list" "redirect_list" {
  account_id = var.account_id
  name       = "bwa_redirect_list"
  description = "Redirect list"
  kind       = "redirect"

  item {
```

```
value {
  redirect {
    source_url    = "review.mondomain.com"
    target_url    = "https://mondomain.com/review"
    status_code   = 301
    subpath_matching = "enabled"
  }
}
comment = "Review redirect"
}
item {
  value {
    redirect {
      source_url    = "feedback.mondomain.com"
      target_url    = "https://mondomain.com/feedback"
      status_code   = 301
      subpath_matching = "enabled"
    }
  }
}
}
```

# Installer borgmatic et borgbackup sur Ubuntu

## Préparer l'environnement

```
apt update && apt install python3 python3-dev libacl1-dev build-essential gcc libssl-dev python3-setuptools  
python3-openssl python3-venv python3-lfuse net-tools libfuse-dev fuse pkg-config python3-pkgconfig  
wget wget https://bootstrap.pypa.io/get-pip.py && python3 get-pip.py  
python3 -m pip install pipx  
python3 -m pipx ensurepath
```

## Installer borgbackup

```
pipx install borgbackup  
borg --version
```

## Installer borgmatic

```
pipx install borgmatic  
borgmatic --version  
generate-borgmatic-config  
vim /etc/borgmatic/config.yaml
```