

# 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 = "XXXXX"
}

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"
    }
  }
}
}
```

---

Revision #6

Created 27 September 2023 09:10:40 by Martin Bouillaud

Updated 29 September 2023 09:01:51 by Martin Bouillaud