» cloudflare_ip_ranges

Use this data source to get the IP ranges of Cloudflare edge nodes.

» Example Usage

```
data "cloudflare_ip_ranges" "cloudflare" {}

resource "google_compute_firewall" "allow_cloudflare_ingress" {
  name    = "from-cloudflare"
  network = "default"

  source_ranges = data.cloudflare_ip_ranges.cloudflare.ipv4_cidr_blocks

  allow {
    ports    = "443"
    protocol = "tcp"
  }
}
```

» Attributes Reference

- `cidr_blocks` - The lexically ordered list of all CIDR blocks.
- `ipv4_cidr_blocks` - The lexically ordered list of only the IPv4 CIDR blocks.
- `ipv6_cidr_blocks` - The lexically ordered list of only the IPv6 CIDR blocks.

» cloudflare_waf_groups

Use this data source to look up WAF Rule Groups.

» Example Usage

The example below matches all WAF Rule Groups that contain the word `example` and are currently `on`. The matched WAF Rule Groups are then returned as output.

```
data "cloudflare_waf_groups" "test" {
  filter {
    name   = ".*example.*"
    mode = "on"
  }
}

output "waf_groups" {
  value = data.cloudflare_waf_groups.test.groups
}
```

» Argument Reference

- `zone_id` - (Required) The ID of the DNS zone in which to search for the WAF Rule Groups.
- `package_id` - (Optional) The ID of the WAF Rule Package in which to search for the WAF Rule Groups.
- `filter` - (Optional) One or more values used to look up WAF Rule Groups. If more than one value is given all values must match in order to be included, see below for full list.

filter

- `name` - (Optional) A regular expression matching the name of the WAF Rule Groups to lookup.
- `mode` - (Optional) Mode of the WAF Rule Groups to lookup. Valid values: on and off.

» Attributes Reference

- `groups` - A map of WAF Rule Groups details. Full list below:

groups

- `id` - The WAF Rule Group ID
- `name` - The WAF Rule Group name
- `description` - The WAF Rule Group description
- `mode` - The WAF Rule Group mode
- `rules_count` - The number of rules in the WAF Rule Group
- `modified_rules_count` - The number of modified rules in the WAF Rule Group
- `package_id` - The ID of the WAF Rule Package that contains the WAF Rule Group

» cloudflare_waf_packages

Use this data source to look up WAF Rule Packages.

» Example Usage

The example below matches all `high` sensitivity WAF Rule Packages, with a `challenge` action mode and an `anomaly` detection mode, that contain the word `example`. The matched WAF Rule Packages are then returned as output.

```
data "cloudflare_waf_packages" "test" {
  filter {
    name   = ".*example.*"
    detection_mode = "anomaly"
    sensitivity = "high"
    action_mode = "challenge"
  }
}

output "waf_packages" {
  value = data.cloudflare_waf_packages.test.packages
}
```

» Argument Reference

- `zone_id` - (Required) The ID of the DNS zone in which to search for the WAF Rule Packages.
- `filter` - (Optional) One or more values used to look up WAF Rule Packages. If more than one value is given all values must match in order to be included, see below for full list.

filter

- `name` - (Optional) A regular expression matching the name of the WAF Rule Packages to lookup.
- `detection_mode` - (Optional) Detection mode of the WAF Rule Packages to lookup.
- `sensitivity` - (Optional) Sensitivity of the WAF Rule Packages to lookup. Valid values: high, medium, low and off.
- `action_mode` - (Optional) Action mode of the WAF Rule Packages to lookup. Valid values: simulate, block and challenge.

» Attributes Reference

- `packages` - A map of WAF Rule Packages details. Full list below:

packages

- `id` - The WAF Rule Package ID
- `name` - The WAF Rule Package name

- `description` - The WAF Rule Package description
- `detection_mode` - The WAF Rule Package detection mode
- `sensitivity` - The WAF Rule Package sensitivity
- `action_mode` - The WAF Rule Package action mode

» cloudflare_waf_rules

Use this data source to look up WAF Rules.

» Example Usage

The example below matches all WAF Rules that are in the group of ID `de677e5818985db1285d0e80225f06e5`, contain `example` in their description, and are currently `on`. The matched WAF Rules are then returned as output.

```
data "cloudflare_waf_rules" "test" {
  zone_id    = "ae36f999674d196762efcc5abb06b345"
  package_id = "a25a9a7e9c00afc1fb2e0245519d725b"

  filter {
    description = ".*example.*"
    mode        = "on"
    group_id    = "de677e5818985db1285d0e80225f06e5"
  }
}

output "waf_rules" {
  value = data.cloudflare_waf_rules.test.rules
}
```

» Argument Reference

- `zone_id` - (Required) The ID of the DNS zone in which to search for the WAF Rules.
- `package_id` - (Optional) The ID of the WAF Rule Package in which to search for the WAF Rules.
- `filter` - (Optional) One or more values used to look up WAF Rules. If more than one value is given all values must match in order to be included, see below for full list.

filter

- `description` - (Optional) A regular expression matching the description of the WAF Rules to lookup.

- `mode` - (Optional) Mode of the WAF Rules to lookup. Valid values: `"on"` and `"off"`.
- `group_id` - (Optional) The ID of the WAF Rule Group in which the WAF Rules to lookup have to be.

» Attributes Reference

- `rules` - A map of WAF Rules details. Full list below:

rules

- `id` - The WAF Rule ID
- `description` - The WAF Rule description
- `priority` - The WAF Rule priority
- `mode` - The WAF Rule mode
- `group_id` - The ID of the WAF Rule Group that contains the WAF Rule
- `group_name` - The Name of the WAF Rule Group that contains the WAF Rule
- `package_id` - The ID of the WAF Rule Package that contains the WAF Rule
- `allowed_modes` - The list of allowed `mode` values for the WAF Rule

» cloudflare_zones

Use this data source to look up Zone records.

» Example Usage

The example below matches all `active` zones that begin with `example.` and are not paused. The matched zones are then locked down using the `cloudflare_zone_lockdown` resource.

```
data "cloudflare_zones" "test" {
  filter {
    name   = "example.*"
    status = "active"
    paused = false
  }
}

resource "cloudflare_zone_lockdown" "endpoint_lockdown" {
  zone        = "${lookup(data.cloudflare_zones.test.zones[0], "name")}"
  paused      = "false"
  description = "Restrict access to these endpoints to requests from a known IP address"
  urls = [
    "api.mysite.com/some/endpoint*",
  ]
  configurations {
    target = "ip"
    value = "198.51.100.4"
  }
}
```

» Argument Reference

- `filter` - (Required) One or more values used to look up zone records. If more than one value is given all values must match in order to be included, see below for full list.

filter

- `name` - (Optional) A regular expression matching the zone to lookup.
- `status` - (Optional) Status of the zone to lookup. Valid values: active, pending, initializing, moved, deleted, deactivated and read only.
- `paused` - (Optional) Paused status of the zone to lookup. Valid values are `true` or `false`.

» Attributes Reference

- `zones` - A map of zone details. Full list below:

zones

- `id` - The zone ID
- `name` - Zone name

» cloudflare_access_application

Provides a Cloudflare Access Application resource. Access Applications are used to restrict access to a whole application using an authorisation gateway managed by Cloudflare.

» Example Usage

```
resource "cloudflare_access_application" "staging_app" {
  zone_id          = "1d5fdc9e88c8a8c4518b068cd94331fe"
  name             = "staging application"
  domain           = "staging.example.com"
  session_duration = "24h"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone to which the access rule should be added.
- `name` - (Required) Friendly name of the Access Application.
- `domain` - (Required) The complete URL of the asset you wish to put Cloudflare Access in front of. Can include subdomains or paths. Or both.
- `session_duration` - (Optional) How often a user will be forced to re-authorise. Must be one of `30m`, `6h`, `12h`, `24h`, `168h`, `730h`.

» Attributes Reference

The following additional attributes are exported:

- `id` - ID of the application
- `aud` - Application Audience (AUD) Tag of the application
- `domain` - Domain of the application
- `session_duration` - Length of session for the application before prompting for a sign in

» Import

Access Applications can be imported using a composite ID formed of zone ID and application ID.

```
$ terraform import cloudflare_access_application.staging cb029e245cfdd66dc8d2e570d5dd3322/d4
```

» cloudflare_access_identity_provider

Provides a Cloudflare Access Identity Provider resource. Identity Providers are used as an authentication or authorisation source within Access.

» Example Usage

```
# one time pin
resource "cloudflare_access_identity_provider" "pin_login" {
  account_id = "1d5fdc9e88c8a8c4518b068cd94331fe"
  name       = "PIN login"
  type       = "onetimepin"
}

# oauth
resource "cloudflare_access_identity_provider" "github_oauth" {
  account_id = "1d5fdc9e88c8a8c4518b068cd94331fe"
  name       = "GitHub OAuth"
  type       = "github"
  config {
    client_id     = "example"
    client_secret = "secret_key"
  }
}

# saml
resource "cloudflare_access_identity_provider" "jumpcloud_saml" {
  account_id = "1d5fdc9e88c8a8c4518b068cd94331fe"
  name       = "JumpCloud SAML"
  type       = "saml"
  config {
    issuer_url = "jumpcloud"
    sso_target_url = "https://sso.myexample.jumpcloud.com/saml2/cloudflareaccess"
    attributes = [ "email", "username" ]
    sign_request = false
    idp_public_cert = "MIIDpDCCAoygAwIBAgIGAV2ka+55MA0GCSqGSIb3DQEBCwUAMIGSMQswCQ...GF/Q2/MH
  }
}
```

Please refer to the developers.cloudflare.com Access documentation for full reference on what is available and how to configure your provider.

» Argument Reference

The following arguments are supported:

- `account` - (Required) The account ID the provider should be associated with.
- `name` - (Required) Friendly name of the Access Identity Provider configuration.
- `type` - (Required) The provider type to use. Must be one of: `"centrify"`, `"facebook"`, `"google-apps"`, `"oidc"`, `"github"`, `"google"`, `"saml"`, `"linkedin"`, `"azureAD"`, `"okta"`, `"onetimepin"`, `"onelogin"`, `"yandex"`.
- `config` - (Optional) Provider configuration from the developer documentation.

» Attributes Reference

The following additional attributes are exported:

- `id` - ID of the Access Identity Provider
- `name` - Friendly name of the Access Identity Provider configuration.
- `type` - The provider type to use.
- `config` - Access Identity Provider configuration.

» Import

Access Identity Providers can be imported using a composite ID formed of account ID and Access Identity Provider ID.

```
$ terraform import cloudflare_access_identity_provider.my_idp cb029e245cfdd66dc8d2e570d5dd33
```

» cloudflare_access_policy

Provides a Cloudflare Access Policy resource. Access Policies are used in conjunction with Access Applications to restrict access to a particular resource.

» Example Usage

```
# Allowing access to `test@example.com` email address only
resource "cloudflare_access_policy" "test_policy" {
  application_id = "cb029e245cfdd66dc8d2e570d5dd3322"
  zone_id        = "d41d8cd98f00b204e9800998ecf8427e"
  name           = "staging policy"
  precedence     = "1"
  decision       = "allow"

  include {
    email = ["test@example.com"]
  }
}

# Allowing `test@example.com` to access but only when coming from a
# specific IP.
resource "cloudflare_access_policy" "test_policy" {
  application_id = "cb029e245cfdd66dc8d2e570d5dd3322"
  zone_id        = "d41d8cd98f00b204e9800998ecf8427e"
  name           = "staging policy"
  precedence     = "1"
  decision       = "allow"

  include {
    email = ["test@example.com"]
  }

  require = {
    ip = [var.office_ip]
  }
}
```

» Argument Reference

The following arguments are supported:

- `application_id` - (Required) The ID of the application the policy is associated with.
- `zone_id` - (Required) The DNS zone to which the access rule should be added.
- `decision` - (Required) Defines the action Access will take if the policy matches the user. Allowed values: `allow`, `deny`, `non_identity`, `bypass`
- `name` - (Required) Friendly name of the Access Application.
- `precedence` - (Optional) The unique precedence for policies on a single application. Integer.
- `require` - (Optional) A series of access conditions, see Access Groups.
- `exclude` - (Optional) A series of access conditions, see Access Groups.
- `include` - (Required) A series of access conditions, see Access Groups.

» Import

Access Policies can be imported using a composite ID formed of zone ID, application ID and policy ID.

```
$ terraform import cloudflare_access_policy.staging cb029e245cfdd66dc8d2e570d5dd3322/d41d8cc
```

where

- `cb029e245cfdd66dc8d2e570d5dd3322` - Zone ID
- `d41d8cd98f00b204e9800998ecf8427e` - Access Application ID
- `67ea780ce4982c1cfbe6b7293afc765d` - Access Policy ID

» cloudflare_access_group

Provides a Cloudflare Access Group resource. Access Groups are used in conjunction with Access Policies to restrict access to a particular resource based on group membership.

» Example Usage

```
# Allowing access to `test@example.com` email address only
resource "cloudflare_access_group" "test_group" {
  account_id     = "975ecf5a45e3bcb680dba0722a420ad9"
  name           = "staging group"

  include {
    email = ["test@example.com"]
  }
}

# Allowing `test@example.com` to access but only when coming from a
# specific IP.
resource "cloudflare_access_group" "test_group" {
  account_id     = "975ecf5a45e3bcb680dba0722a420ad9"
  name           = "staging group"

  include {
    email = ["test@example.com"]
  }

  require = {
    ip = [var.office_ip]
  }
}
```

» Argument Reference

The following arguments are supported:

- `account_id` - (Required) The ID of the account the group is associated with.
- `name` - (Required) Friendly name of the Access Group.
- `require` - (Optional) A series of access conditions, see below for full list.
- `exclude` - (Optional) A series of access conditions, see below for full list.
- `include` - (Required) A series of access conditions, see below for full list.

» Conditions

require, exclude and include arguments share the available conditions which can be applied. The conditions are:

- ip - (Optional) A list of IP addresses or ranges. Example: ip = ["1.2.3.4", "10.0.0.0/2"]
- email - (Optional) A list of email addresses. Example: email = ["test@example.com"]
- email_domain - (Optional) A list of email domains. Example: email_domain = ["example.com"]
- service_token - (Optional) A list of service token ids. Example: service_token = cloudflare_access_service_token.demo.id
- any_valid_service_token - (Optional) Boolean indicating if allow all tokens to be granted. Example: any_valid_service_token = true
- group - (Optional) A list of access group ids. Example: group = [cloudflare_access_group.demo.id]
- everyone - (Optional) Boolean indicating permitting access for all requests. Example: everyone = true
- certificate - (Optional) Whether to use mTLS certificate authentication.
- common_name - (Optional) Use a certificate common name to authenticate with.
- gsuite - (Optional) Use GSuite as the authentication mechanism. Example:

```
# ... other configuration
include {
  gsuite {
    email = "admins@example.com"
    identity_provider_id = "ca298b82-93b5-41bf-bc2d-10493f09b761"
  }
}
```

- github - (Optional) Use a GitHub team as the include condition. Example:

```
# ... other configuration
include {
  github {
    name = "my-github-team-name"
    identity_provider_id = "ca298b82-93b5-41bf-bc2d-10493f09b761"
  }
}
```

- azure - (Optional) Use Azure AD as the include condition. Example:

```
# ... other configuration
```

```
include {
  azure {
    id = "86773093-5feb-48dd-814b-7ccd3676ff50e"
    identity_provider_id = "ca298b82-93b5-41bf-bc2d-10493f09b761"
  }
}
```

- `okta` - (Optional) Use Okta as the `include` condition. Example:

```
# ... other configuration
include {
  okta {
    name = "admins"
    identity_provider_id = "ca298b82-93b5-41bf-bc2d-10493f09b761"
  }
}
```

- `saml` - (Optional) Use an external SAML setup as the `include` condition. Example:

```
# ... other configuration
include {
  saml {
    attribute_name = "group"
    attribute_value = "admins"
    identity_provider_id = "ca298b82-93b5-41bf-bc2d-10493f09b761"
  }
}
```

» Import

Access Groups can be imported using a composite ID formed of account ID and group ID.

```
$ terraform import cloudflare_access_group.staging 975ecf5a45e3bcb680dba0722a420ad9/67ea780c
```

where

- `975ecf5a45e3bcb680dba0722a420ad9` - Account ID
- `67ea780ce4982c1cfbe6b7293afc765d` - Access Group ID

» cloudflare_access_rule

Provides a Cloudflare IP Firewall Access Rule resource. Access control can be applied on basis of IP addresses, IP ranges, AS numbers or countries.

» Example Usage

```
# Challenge requests coming from known Tor exit nodes.
resource "cloudflare_access_rule" "tor_exit_nodes" {
  notes = "Requests coming from known Tor exit nodes"
  mode = "challenge"
  configuration = {
    target = "country"
    value = "T1"
  }
}

# Whitelist (sic!) requests coming from Antarctica, but only for single zone.
resource "cloudflare_access_rule" "antarctica" {
  notes = "Requests coming from Antarctica"
  mode = "whitelist"
  configuration = {
    target = "country"
    value = "AQ"
  }
  zone_id = "cb029e245cfdd66dc8d2e570d5dd3322"
}

# Whitelist office's network IP ranges on all account zones (or other lists of resources).
# Resulting Terraform state will be a list of resources.
provider "cloudflare" {
  # ... other provider configuration
  account_id = "d41d8cd98f00b204e9800998ecf8427e"
}
variable "my_office" {
  type = "list"
  default = ["192.0.2.0/24", "198.51.100.0/24", "2001:db8::/56"]
}
resource "cloudflare_access_rule" "office_network" {
  count = length(var.my_office)
  notes = "Requests coming from office network"
  mode = "whitelist"
  configuration = {
    target = "ip_range"
    value = element(var.my_office, count.index)
  }
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Optional) The DNS zone to which the access rule should be added.
- `mode` - (Required) The action to apply to a matched request. Allowed values: ”block”, ”challenge”, ”whitelist”, ”js_challenge”
- `notes` - (Optional) A personal note about the rule. Typically used as a reminder or explanation for the rule.
- `configuration` - (Required) Rule configuration to apply to a matched request. It’s a complex value. See description below.

Note: If both `zone` and `zone_id` are empty, then access rule will be set to the account level and apply to all their zones.

The **configuration** block supports:

- `target` - (Required) The request property to target. Allowed values: ”ip”, ”ip6”, ”ip_range”, ”asn”, ”country”
- `value` - (Required) The value to target. Depends on target’s type.

» Attributes Reference

The following attributes are exported:

- `id` - The access rule ID.
- `zone_id` - The DNS zone ID.

» Import

Records can be imported using a composite ID formed of access rule type, access rule type identifier and identifer value, e.g.

```
$ terraform import cloudflare_access_rule.default zone/cb029e245cfdd66dc8d2e570d5dd3322/d41c
```

where:

- `zone` - access rule type (`account`, `zone` or `user`)
- `cb029e245cfdd66dc8d2e570d5dd3322` - access rule type ID (i.e the zone ID or account ID you wish to target)
- `d41d8cd98f00b204e9800998ecf8427e` - access rule ID as returned by respective API endpoint for the type you are attempting to import.

» cloudflare_access_service_token

Access Service Tokens are used for service-to-service communication when an application is behind Cloudflare Access.

» Example Usage

```
resource "cloudflare_access_service_token" "my_app" {
  account_id = "d41d8cd98f00b204e9800998ecf8427e"
  name       = "CI/CD app"
}
```

» Argument Reference

The following arguments are supported:

- `account_id` - (Required) The ID of the account where the Access Service is being created.
- `name` - (Required) Friendly name of the token's intent.

» Attributes Reference

The following attributes are exported:

- `client_id` - UUID client ID associated with the Service Token.
- `client_secret` - A secret for interacting with Access protocols.

» Import

Important: If you are importing an Access Service Token you will not have the `client_secret` available in the state for use. The `client_secret` is only available once, at creation. In most cases, it is better to just create a new resource should you need to reference it in other resources.

Access Service Tokens can be imported using a composite ID formed of account ID and Service Token ID.

```
$ terraform import cloudflare_access_service_token.my_app cb029e245cfdd66dc8d2e570d5dd3322/d
```

where

- `cb029e245cfdd66dc8d2e570d5dd3322` - Account ID
- `d41d8cd98f00b204e9800998ecf8427e` - Access Service Token ID

» cloudflare_account_member

Provides a resource which manages Cloudflare account members.

» Example Usage

```
resource "cloudflare_account_member" "example_user" {
  email_address = "user@example.com"
  role_ids = [
    "68b329da9893e34099c7d8ad5cb9c940",
    "d784fa8b6d98d27699781bd9a7cf19f0"
  ]
}
```

» Argument Reference

The following arguments are supported:

- `email_address` - (Required) The email address of the user who you wish to manage. Note: Following creation, this field becomes read only via the API and cannot be updated.
- `role_ids` - (Required) Array of account role IDs that you want to assign to a member.

» Import

Account members can be imported using a composite ID formed of account ID and account member ID, e.g.

```
$ terraform import cloudflare_account_member.example_user d41d8cd98f00b204e9800998ecf8427e/b
```

where:

- `d41d8cd98f00b204e9800998ecf8427e` - account ID as returned by the API
- `b58c6f14d292556214bd64909bcdb118` - account member ID as returned by the API

» cloudflare_argo

Cloudflare Argo controls the routing to your origin and tiered caching options to speed up your website browsing experience.

» Example Usage

```
resource "cloudflare_argo" "example" {
  zone_id        = "d41d8cd98f00b204e9800998ecf8427e"
  tiered_caching = "on"
  smart_routing  = "on"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID that you wish to manage Argo on.
- `tiered_caching` - (Optional) Whether tiered caching is enabled. Valid values: `on` or `off`. Defaults to `off`.
- `smart_routing` - (Optional) Whether smart routing is enabled. Valid values: `on` or `off`. Defaults to `off`.

» Import

Argo settings can be imported the zone ID.

```
$ terraform import cloudflare_argo.example d41d8cd98f00b204e9800998ecf8427e
```

where `d41d8cd98f00b204e9800998ecf8427e` is the zone ID.

» cloudflare_custom_pages

Provides a resource which manages Cloudflare custom error pages.

» Example Usage

```
resource "cloudflare_custom_pages" "basic_challenge" {
  zone_id = "d41d8cd98f00b204e9800998ecf8427e"
  type    = "basic_challenge"
  url     = "https://example.com/challenge.html"
  state   = "customized"
}
```

» Argument Reference

The following arguments are supported:

- zone_id - (Optional) The zone ID where the custom pages should be updated. Either zone_id or account_id must be provided.
- account_id - (Optional) The account ID where the custom pages should be updated. Either account_id or zone_id must be provided. If account_id is present, it will override the zone setting.
- type - (Required) The type of custom page you wish to update. Must be one of basic_challenge, waf_challenge, waf_block, ratelimit_block, country_challenge, ip_block, under_attack, 500_errors, 1000_errors, always_online.
- url - (Required) URL of where the custom page source is located.
- state - (Required) Managed state of the custom page. Must be one of default, customised. If the value is default it will be removed from the Terraform state management.

» Import

Custom pages can be imported using a composite ID formed of:

- customPageLevel - Either account or zone.
- identifier - The ID of the account or zone you intend to manage.
- pageType - The value from the type argument.

Example for a zone:

```
$ terraform import cloudflare_custom_pages.basic_challenge zone/d41d8cd98f00b204e9800998ecf8
```

Example for an account:

```
$ terraform import cloudflare_custom_pages.basic_challenge account/e268443e43d93dab7ebef303b
```

» cloudflare_custom_ssl

Provides a Cloudflare custom ssl resource.

» Example Usage

```
# Add a custom ssl certificate to the domain
resource "cloudflare_custom_ssl" "foossl" {
  zone_id = "${var.cloudflare_zone_id}"
  custom_ssl_options = {
    "certificate" = "-----INSERT CERTIFICATE-----"
    "private_key" = "-----INSERT PRIVATE KEY-----"
    "bundle_method" = "ubiquitous",
    "geo_restrictions" = "us",
    "type" = "legacy_custom"
  }
}

variable "cloudflare_zone_id" {
  type = "string"
  default = "1d5fdc9e88c8a8c4518b068cd94331fe"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone id to the custom ssl cert should be added.
- `custom_ssl_options` - (Required) The certificate, private key and associated optional parameters, such as bundle_method, geo_restrictions, and type.

custom_ssl_options block supports:

- `certificate` - (Required) Certificate certificate and the intermediate(s)
- `private_key` - (Required) Certificate's private key
- `bundle_method` - (Optional) Method of building intermediate certificate chain. A ubiquitous bundle has the highest probability of being verified everywhere, even by clients using outdated or unusual trust stores. An optimal bundle uses the shortest chain and newest intermediates. And the force bundle verifies the chain, but does not otherwise modify it. Valid values are `ubiquitous` (default), `optimal`, `force`.
- `geo_restrictions` - (Optional) Specifies the region where your private key can be held locally. Valid values are `us`, `eu`, `highest_security`.
- `type` - (Optional) Whether to enable support for legacy clients which do not include SNI in the TLS handshake. Valid values are `legacy_custom` (default), `sni_custom`.

» Import

Custom SSL Certs can be imported using a composite ID formed of the zone ID and certificate ID, separated by a "/" e.g.

```
$ terraform import cloudflare_custom_ssl.default 1d5fdc9e88c8a8c4518b068cd94331fe/0123f0ab-9
```

» cloudflare_filter

Filter expressions that can be referenced across multiple features, e.g. Firewall Rule. The expression format is similar to Wireshark Display Filter.

» Example Usage

```
resource "cloudflare_filter" "wordpress" {
  zone_id = "d41d8cd98f00b204e9800998ecf8427e"
  description = "Wordpress break-in attempts that are outside of the office"
  expression = "(http.request.uri.path ~ \".*wp-login.php\" or http.request.uri.path ~ \".*x
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone to which the Filter should be added.
- `paused` - (Optional) Whether this filter is currently paused. Boolean value.
- `expression` - (Required) The filter expression to be used.
- `description` - (Optional) A note that you can use to describe the purpose of the filter.
- `ref` - (Optional) Short reference tag to quickly select related rules.

» Attributes Reference

The following attributes are exported:

- `id` - Filter identifier.

» Import

Filter can be imported using a composite ID formed of zone ID and filter ID, e.g.

```
$ terraform import cloudflare_filter.default d41d8cd98f00b204e9800998ecf8427e/9e107d9d372bb6
```

where:

- `d41d8cd98f00b204e9800998ecf8427e` - zone ID
- `9e107d9d372bb6826bd81d3542a419d6` - filter ID as returned by API

» cloudflare_firewall_rule

Define Firewall rules using filter expressions for more control over how traffic is matched to the rule. A filter expression permits selecting traffic by multiple criteria allowing greater freedom in rule creation.

Filter expressions needs to be created first before using Firewall Rule. See Filter.

» Example Usage

```
resource "cloudflare_filter" "wordpress" {
  zone_id = "d41d8cd98f00b204e9800998ecf8427e"
  description = "Wordpress break-in attempts that are outside of the office"
  expression = "(http.request.uri.path ~ \".*wp-login.php\" or http.request.uri.path ~ \".*x
}

resource "cloudflare_firewall_rule" "wordpress" {
  zone_id = "d41d8cd98f00b204e9800998ecf8427e"
  description = "Block wordpress break-in attempts"
  filter_id = cloudflare_filter.wordpress.id
  action = "block"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone to which the Filter should be added.
- `action` - (Required) The action to apply to a matched request. Allowed values: ”block”, ”challenge”, ”allow”, ”js_challenge”, ”bypass”. Enterprise plan also allows ”log”.
- `priority` - (Optional) The priority of the rule to allow control of processing order. A lower number indicates high priority. If not provided, any rules with a priority will be sequenced before those without.
- `paused` - (Optional) Whether this filter based firewall rule is currently paused. Boolean value.
- `description` - (Optional) A description of the rule to help identify it.
- `products` - (Optional) List of products to bypass for a request when the bypass action is used. Allowed values: ”zoneLockdown”, ”uaBlock”, ”bic”, ”hot”, ”securityLevel”, ”rateLimit”, ”waf”.

» Attributes Reference

The following attributes are exported:

- `id` - Firewall Rule identifier.

» Import

Firewall Rule can be imported using a composite ID formed of zone ID and rule ID, e.g.

```
$ terraform import cloudflare_firewall_rule.default d41d8cd98f00b204e9800998ecf8427e/9e107d9
```

where:

- `d41d8cd98f00b204e9800998ecf8427e` - zone ID
- `9e107d9d372bb6826bd81d3542a419d6` - rule ID as returned by API

» cloudflare_load_balancer

Provides a Cloudflare Load Balancer resource. This sits in front of a number of defined pools of origins and provides various options for geographically-aware load balancing. Note that the load balancing feature must be enabled in your Cloudflare account before you can use this resource.

» Example Usage

```
# Define a load balancer which always points to a pool we define below
# In normal usage, would have different pools set for different pops (cloudflare points-of-p
# Within each pop or region we can define multiple pools in failover order
resource "cloudflare_load_balancer" "bar" {
  zone_id = "d41d8cd98f00b204e9800998ecf8427e"
  name = "example-load-balancer"
  fallback_pool_id = cloudflare_load_balancer_pool.foo.id
  default_pool_ids = [cloudflare_load_balancer_pool.foo.id]
  description = "example load balancer using geo-balancing"
  proxied = true
  steering_policy = "geo"
  pop_pools {
    pop = "LAX"
    pool_ids = [cloudflare_load_balancer_pool.foo.id]
  }
  region_pools {
    region = "WNAM"
    pool_ids = [cloudflare_load_balancer_pool.foo.id]
  }
}
```

```
resource "cloudflare_load_balancer_pool" "foo" {
  name = "example-lb-pool"
  origins {
    name = "example-1"
    address = "192.0.2.1"
    enabled = false
  }
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The zone ID to add the load balancer to.
- `name` - (Required) The DNS name (FQDN, including the zone) to associate with the load balancer.
- `fallback_pool_id` - (Required) The pool ID to use when all other pools are detected as unhealthy.
- `default_pool_ids` - (Required) A list of pool IDs ordered by their failover priority. Used whenever region/pop pools are not defined.
- `description` - (Optional) Free text description.
- `ttl` - (Optional) Time to live (TTL) of this load balancer's DNS `name`. Conflicts with `proxied` - this cannot be set for proxied load balancers. Default is `30`.
- `steering_policy` - (Optional) Determine which method the load balancer uses to determine the fastest route to your origin. Valid values are: `"off"`, `"geo"`, `"dynamic_latency"`, `"random"` or `""`. Default is `""`.
- `proxied` - (Optional) Whether the hostname gets Cloudflare's origin protection. Defaults to `false`.
- `enabled` - (Optional) Enable or disable the load balancer. Defaults to `true` (enabled).
- `region_pools` - (Optional) A set containing mappings of region/country codes to a list of pool IDs (ordered by their failover priority) for the given region. Fields documented below.
- `pop_pools` - (Optional) A set containing mappings of Cloudflare Point-of-Presence (PoP) identifiers to a list of pool IDs (ordered by their failover priority) for the PoP (datacenter). This feature is only available to enterprise customers. Fields documented below.
- `session_affinity` - (Optional) Associates all requests coming from an end-user with a single origin. Cloudflare will set a cookie on the initial response to the client, such that consequent requests with the cookie in the request will go to the same origin, so long as it is available. Valid values are: `""`, `"none"`, `"cookie"`, and `"ip_cookie"`. Default is `""`.

region_pools requires the following:

- `region` - (Required) A region code which must be in the list defined here. Multiple entries should not be specified with the same region.
- `pool_ids` - (Required) A list of pool IDs in failover priority to use in the given region.

pop_pools requires the following:

- `pop` - (Required) A 3-letter code for the Point-of-Presence. Allowed values can be found in the list of datacenters on the status page. Multiple entries should not be specified with the same PoP.
- `pool_ids` - (Required) A list of pool IDs in failover priority to use for traffic reaching the given PoP.

» Attributes Reference

The following attributes are exported:

- `id` - Unique identifier in the API for the load balancer.
- `created_on` - The RFC3339 timestamp of when the load balancer was created.
- `modified_on` - The RFC3339 timestamp of when the load balancer was last modified.

» cloudflare_load_balancer_monitor

If you're using Cloudflare's Load Balancing to load-balance across multiple origin servers or data centers, you configure one of these Monitors to actively check the availability of those servers over HTTP(S) or TCP.

» Example Usage

» HTTP Monitor

```
resource "cloudflare_load_balancer_monitor" "http_monitor" {
  type = "http"
  expected_body = "alive"
  expected_codes = "2xx"
  method = "GET"
  timeout = 7
  path = "/health"
  interval = 60
  retries = 5
  description = "example http load balancer"
  header {
    header = "Host"
    values = ["example.com"]
  }
  allow_insecure = false
  follow_redirects = true
}
```

» TCP Monitor

```
resource "cloudflare_load_balancer_monitor" "tcp_monitor" {
  type = "tcp"
  method = "connection_established"
  timeout = 7
  interval = 60
  retries = 5
  description = "example tcp load balancer"
}
```

» Argument Reference

The following arguments are supported:

- `expected_body` - (Optional) A case-insensitive sub-string to look for in the response body. If this string is not found, the origin will be marked as unhealthy. Only valid if `type` is ”http” or ”https”. Default: ””.
- `expected_codes` - (Optional) The expected HTTP response code or code range of the health check. Eg `2xx`. Only valid and required if `type` is ”http” or ”https”.
- `method` - (Optional) The method to use for the health check. Valid values are any valid HTTP verb if `type` is ”http” or ”https”, or `connection_established` if `type` is ”tcp”. Default: ”GET” if `type` is ”http” or ”https”, or ”connection_established” if `type` is ”tcp” .
- `timeout` - (Optional) The timeout (in seconds) before marking the health check as failed. Default: 5.
- `path` - (Optional) The endpoint path to health check against. Default: ”/”. Only valid if `type` is ”http” or ”https”.
- `interval` - (Optional) The interval between each health check. Shorter intervals may improve failover time, but will increase load on the origins as we check from multiple locations. Default: 60.
- `retries` - (Optional) The number of retries to attempt in case of a timeout before marking the origin as unhealthy. Retries are attempted immediately. Default: 2.
- `header` - (Optional) The HTTP request headers to send in the health check. It is recommended you set a Host header by default. The User-

Agent header cannot be overridden. Fields documented below. Only valid if `type` is ”http” or ”https”.
- `type` - (Optional) The protocol to use for the healthcheck. Currently supported protocols are ’HTTP’, ’HTTPS’ and ’TCP’. Default: ”http”.
- `description` - (Optional) Free text description.
- `allow_insecure` - (Optional) Do not validate the certificate when monitor use HTTPS. Only valid if `type` is ”http” or ”https”.
- `follow_redirects` - (Optional) Follow redirects if returned by the origin. Only valid if `type` is ”http” or ”https”.

header requires the following:

- `header` - (Required) The header name.
- `values` - (Required) A list of string values for the header.

» Attributes Reference

The following attributes are exported:

- `id` - Load balancer monitor ID.
- `created_on` - The RFC3339 timestamp of when the load balancer monitor was created.
- `modified_on` - The RFC3339 timestamp of when the load balancer monitor was last modified.

» cloudflare_load_balancer_pool

Provides a Cloudflare Load Balancer pool resource. This provides a pool of origins that can be used by a Cloudflare Load Balancer. Note that the load balancing feature must be enabled in your Cloudflare account before you can use this resource.

» Example Usage

```
resource "cloudflare_load_balancer_pool" "foo" {
  name = "example-pool"
  origins {
    name = "example-1"
    address = "192.0.2.1"
    enabled = false
  }
  origins {
    name = "example-2"
    address = "192.0.2.2"
  }
  description = "example load balancer pool"
  enabled = false
  minimum_origins = 1
  notification_email = "someone@example.com"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A short name (tag) for the pool. Only alphanumeric characters, hyphens, and underscores are allowed.
- `origins` - (Required) The list of origins within this pool. Traffic directed at this pool is balanced across all currently healthy origins, provided the pool itself is healthy. It's a complex value. See description below.
- `check_regions` - (Optional) A list of regions (specified by region code) from which to run health checks. Empty means every Cloudflare data center (the default), but requires an Enterprise plan. Region codes can be found here.
- `description` - (Optional) Free text description.
- `enabled` - (Optional) Whether to enable (the default) this pool. Disabled pools will not receive traffic and are excluded from health checks. Disabling a pool will cause any load balancers using it to failover to the next pool (if any).
- `minimum_origins` - (Optional) The minimum number of origins that must be healthy for this pool to serve traffic. If the number of healthy origins falls below this number, the pool will be marked unhealthy and we will failover to the next available pool. Default: 1.
- `monitor` - (Optional) The ID of the Monitor to use for health checking origins within this pool.
- `notification_email` - (Optional) The email address to send health status notifications to. This can be an individual mailbox or a mailing list.

The **origins** block supports:

- `name` - (Required) A human-identifiable name for the origin.
- `address` - (Required) The IP address (IPv4 or IPv6) of the origin, or the publicly addressable hostname. Hostnames entered here should resolve directly to the origin, and not be a hostname proxied by Cloudflare.
- `weight` - (Optional) The weight (0.01 - 1.00) of this origin, relative to other origins in the pool. Equal values mean equal weighting. A weight of 0 means traffic will not be sent to this origin, but health is still checked. Default: 1.
- `enabled` - (Optional) Whether to enable (the default) this origin within the Pool. Disabled origins will not receive traffic and are excluded from

health checks. The origin will only be disabled for the current pool.

» Attributes Reference

The following attributes are exported:

- `id` - ID for this load balancer pool.
- `created_on` - The RFC3339 timestamp of when the load balancer was created.
- `modified_on` - The RFC3339 timestamp of when the load balancer was last modified.

» cloudflare_logpush_job

Provides a resource which manages Cloudflare logpush jobs.

» Example Usage

```
resource "cloudflare_logpush_job" "example_job" {
  enabled = true
  zone_id = "d41d8cd98f00b204e9800998ecf8427e"
  name = "My-logpush-job"
  logpull_options = "fields=RayID,ClientIP,EdgeStartTimestamp&timestamps=rfc3339"
  destination_conf = "s3://my-bucket-path?region=us-west-2"
  ownership_challenge = "00000000000000000"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the logpush job to create. Must match the regular expression `^[a-zA-Z0-9\-\.]*$`.
- `zone_id` - (Required) The zone ID where the logpush job should be created.
- `destination_conf` - (Required) Uniquely identifies a resource (such as an s3 bucket) where data will be pushed. Additional configuration parameters supported by the destination may be included. See Logpush destination documentation.
- `ownership_challenge` - (Required) Ownership challenge token to prove destination ownership. See Developer documentation.

- `logpull_options` - (Optional) Configuration string for the Logshare API. It specifies things like requested fields and timestamp formats. See Logpull options documentation.
- `enable` - (Optional) Whether to enable to job to create or not.

» cloudflare_origin_ca_certificate

Provides a Cloudflare Origin CA certificate used to protect traffic to your origin without involving a third party Certificate Authority.

This resource requires you use your Origin CA Key as the `api_user_service_key`.

» Example Usage

```
# Create a CSR and generate a CA certificate
resource "tls_private_key" "example" {
  algorithm = "RSA"
}

resource "tls_cert_request" "example" {
  key_algorithm   = tls_private_key.example.algorithm
  private_key_pem = tls_private_key.example.private_key_pem

  subject {
    common_name  = ""
    organization = "Terraform Test"
  }
}

resource "cloudflare_origin_ca_certificate" "example" {
  csr                = tls_cert_request.example.cert_request_pem
  hostnames          = [ "example.com" ]
  request_type       = "origin-rsa"
  requested_validity = 7
}
```

» Argument Reference

- `csr` - (Required) The Certificate Signing Request. Must be newline-encoded.
- `hostnames` - (Required) An array of hostnames or wildcard names bound to the certificate.

- `request_type` - (Required) The signature type desired on the certificate.
- `requested_validity` - (Required) The number of days for which the certificate should be valid.

» Attributes Reference

The following attributes are exported:

- `id` - The x509 serial number of the Origin CA certificate.
- `certificate` - The Origin CA certificate
- `expires_on` - The datetime when the certificate will expire.

» Import

Origin CA certificate resource can be imported using an ID, e.g.

```
$ terraform import cloudflare_origin_ca_certificate.example 2762665387716118026071536872881
```

» cloudflare_page_rule

Provides a Cloudflare page rule resource.

» Example Usage

```
# Add a page rule to the domain
resource "cloudflare_page_rule" "foobar" {
  zone_id = var.cloudflare_zone_id
  target = "sub.${var.cloudflare_zone}/page"
  priority = 1

  actions {
    ssl = "flexible"
    email_obfuscation = "on"
    minify {
      html = "off"
      css  = "on"
      js   = "on"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to which the page rule should be added.
- `target` - (Required) The URL pattern to target with the page rule.
- `actions` - (Required) The actions taken by the page rule, options given below.
- `priority` - (Optional) The priority of the page rule among others for this target, the higher the number the higher the priority as per API documentation.
- `status` - (Optional) Whether the page rule is active or disabled.

Action blocks support the following:

- `always_online` - (Optional) Whether this action is `"on"` or `"off"`.
- `always_use_https` - (Optional) Boolean of whether this action is enabled. Default: false.
- `automatic_https_rewrites` - (Optional) Whether this action is `"on"` or `"off"`.
- `browser_cache_ttl` - (Optional) The Time To Live for the browser cache. `0` means 'Respect Existing Headers'
- `browser_check` - (Optional) Whether this action is `"on"` or `"off"`.
- `bypass_cache_on_cookie` - (Optional) String value of cookie name to conditionally bypass cache the page.
- `cache_by_device_type` - (Optional) Whether this action is `"on"` or `"off"`.
- `cache_deception_armor` - (Optional) Whether this action is `"on"` or `"off"`.
- `cache_level` - (Optional) Whether to set the cache level to `"bypass"`, `"basic"`, `"simplified"`, `"aggressive"`, or `"cache_everything"`.
- `cache_on_cookie` - (Optional) String value of cookie name to conditionally cache the page.
- `disable_apps` - (Optional) Boolean of whether this action is enabled. Default: false.
- `disable_performance` - (Optional) Boolean of whether this action is enabled. Default: false.
- `disable_railgun` - (Optional) Boolean of whether this action is enabled. Default: false.
- `disable_security` - (Optional) Boolean of whether this action is enabled. Default: false.
- `edge_cache_ttl` - (Optional) The Time To Live for the edge cache.
- `email_obfuscation` - (Optional) Whether this action is `"on"` or `"off"`.
- `explicit_cache_control` - (Optional) Whether origin Cache-Control action is `"on"` or `"off"`.
- `forwarding_url` - (Optional) The URL to forward to, and with what

status. See below.
- host_header_override - (Optional) Value of the Host header to send.
- ip_geolocation - (Optional) Whether this action is "on" or "off".
- minify - (Optional) The configuration for HTML, CSS and JS minification. See below for full list of options.
- mirage - (Optional) Whether this action is "on" or "off".
- opportunistic_encryption - (Optional) Whether this action is "on" or "off".
- origin_error_page_pass_thru - (Optional) Whether this action is "on" or "off".
- polish - (Optional) Whether this action is "off", "lossless" or "lossy".
- resolve_override - (Optional) Overridden origin server name.
- respect_strong_etag - (Optional) Whether this action is "on" or "off".
- response_buffering - (Optional) Whether this action is "on" or "off".
- rocket_loader - (Optional) Whether to set the rocket loader to "on", "off".
- security_level - (Optional) Whether to set the security level to "off", "essentially_off", "low", "medium", "high", or "under_attack".
- server_side_exclude - (Optional) Whether this action is "on" or "off".
- smart_errors - (Optional) Whether this action is "on" or "off".
- sort_query_string_for_cache - (Optional) Whether this action is "on" or "off".
- ssl - (Optional) Whether to set the SSL mode to "off", "flexible", "full", "strict", or "origin_pull".
- true_client_ip_header - (Optional) Whether this action is "on" or "off".
- waf - (Optional) Whether this action is "on" or "off".

Forwarding URL actions support the following:

- url - (Required) The URL to which the page rule should forward.
- status_code - (Required) The status code to use for the redirection.

Minify actions support the following:

- html - (Required) Whether HTML should be minified. Valid values are "on" or "off".
- css - (Required) Whether CSS should be minified. Valid values are "on" or "off".
- js - (Required) Whether Javascript should be minified. Valid values are "on" or "off".

» Attributes Reference

The following attributes are exported:

- `id` - The page rule ID.
- `target` - The URL pattern targeted by the page rule.
- `actions` - The actions applied by the page rule.
- `priority` - The priority of the page rule.
- `status` - Whether the page rule is active or disabled.

» Import

Page rules can be imported using a composite ID formed of zone ID and page rule ID, e.g.

```
$ terraform import cloudflare_page_rule.default d41d8cd98f00b204e9800998ecf8427e/ch8374ftwdg
```

» cloudflare_rate_limit

Provides a Cloudflare rate limit resource for a given zone. This can be used to limit the traffic you receive zone-wide, or matching more specific types of requests/responses.

» Example Usage

```
resource "cloudflare_rate_limit" "example" {
  zone_id = var.cloudflare_zone_id
  threshold = 2000
  period = 2
  match {
    request {
      url_pattern = "${var.cloudflare_zone}/*"
      schemes = ["HTTP", "HTTPS"]
      methods = ["GET", "POST", "PUT", "DELETE", "PATCH", "HEAD"]
    }
    response {
      statuses = [200, 201, 202, 301, 429]
      origin_traffic = false
    }
  }
  action {
    mode = "simulate"
    timeout = 43200
    response {
      content_type = "text/plain"
      body = "custom response body"
    }
  }
  correlate {
    by = "nat"
  }
  disabled = false
  description = "example rate limit for a zone"
  bypass_url_patterns = ["${var.cloudflare_zone}/bypass1","${var.cloudflare_zone}/bypass2"]
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to apply rate limiting to.
- `threshold` - (Required) The threshold that triggers the rate limit mitigations, combine with period. i.e. threshold per period (min: 2, max: 1,000,000).
- `period` - (Required) The time in seconds to count matching traffic. If the count exceeds threshold within this period the action will be performed (min: 1, max: 86,400).
- `action` - (Required) The action to be performed when the threshold of matched traffic within the period defined is exceeded.
- `match` - (Optional) Determines which traffic the rate limit counts towards the threshold. By default matches all traffic in the zone. See definition below.
- `disabled` - (Optional) Whether this ratelimit is currently disabled. Default: `false`.
- `description` - (Optional) A note that you can use to describe the reason for a rate limit. This value is sanitized and all tags are removed.
- `bypass_url_patterns` - (Optional) URLs matching the patterns specified here will be excluded from rate limiting.
- `correlate` - (Optional) Determines how rate limiting is applied. By default if not specified, rate limiting applies to the clients IP address.

The **match** block supports:

- `request` - (Optional) Matches HTTP requests (from the client to Cloudflare). See definition below.
- `response` (Optional) Matches HTTP responses before they are returned to the client from Cloudflare. If this is defined, then the entire counting of traffic occurs at this stage. This field is not required.

The **match.request** block supports:

- `methods` - (Optional) HTTP Methods, can be a subset ['POST','PUT'] or all ['_ALL_']. Default: ['_ALL_'].

- `schemes` - (Optional) HTTP Schemes, can be one ['HTTPS'], both ['HTTP','HTTPS'] or all ['_ALL_']. Default: ['_ALL_'].
- `url_pattern` - (Optional) The URL pattern to match comprised of the host and path, i.e. example.org/path. Wildcard are expanded to match applicable traffic, query strings are not matched. Use * for all traffic to your zone. Default: '*'.

The **match.response** block supports:

- `statuses` - (Optional) HTTP Status codes, can be one [403], many [401,403] or indicate all by not providing this value.
- `origin_traffic` - (Optional) Only count traffic that has come from your origin servers. If true, cached items that Cloudflare serve will not count towards rate limiting. Default: `true`.

The **action** block supports:

- `mode` - (Required) The type of action to perform. Allowable values are 'simulate', 'ban', 'challenge' and 'js_challenge'.
- `timeout` - (Optional) The time in seconds as an integer to perform the mitigation action. This field is required if the `mode` is either `simulate` or `ban`. Must be the same or greater than the period (min: 1, max: 86400).
- `response` - (Optional) Custom content-type and body to return, this overrides the custom error for the zone. This field is not required. Omission will result in default HTML error page. Definition below.

The **action.response** block supports:

- `content_type` - (Required) The content-type of the body, must be one of: 'text/plain', 'text/xml', 'application/json'.
- `body` - (Required) The body to return, the content here should conform to the content_type.

The **correlate** block supports:

- `by` - (Optional) If set to 'nat', NAT support will be enabled for rate limiting.

» Attributes Reference

The following attributes are exported:

- `id` - The Rate limit ID.

» Import

Rate limits can be imported using a composite ID formed of zone name and rate limit ID, e.g.

```
$ terraform import cloudflare_rate_limit.default d41d8cd98f00b204e9800998ecf8427e/ch8374ftwc
```

» cloudflare_record

Provides a Cloudflare record resource.

» Example Usage

```
# Add a record to the domain
resource "cloudflare_record" "foobar" {
  zone_id = var.cloudflare_zone_id
  name    = "terraform"
  value   = "192.168.0.11"
  type    = "A"
  ttl     = 3600
}

# Add a record requiring a data map
resource "cloudflare_record" "_sip_tls" {
  zone_id = var.cloudflare_zone_id
  name    = "_sip._tls"
  type    = "SRV"

  data = {
    service  = "_sip"
    proto    = "_tls"
    name     = "terraform-srv"
    priority = 0
    weight   = 0
    port     = 443
    target   = "example.com"
  }
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to add the record to
- `name` - (Required) The name of the record
- `type` - (Required) The type of the record
- `value` - (Optional) The (string) value of the record. Either this or `data` must be specified

- `data` - (Optional) Map of attributes that constitute the record value. Primarily used for LOC and SRV record types. Either this or `value` must be specified
- `ttl` - (Optional) The TTL of the record (automatic: '1')
- `priority` - (Optional) The priority of the record
- `proxied` - (Optional) Whether the record gets Cloudflare's origin protection; defaults to `false`.

» Attributes Reference

The following attributes are exported:

- `id` - The record ID
- `hostname` - The FQDN of the record
- `proxiable` - Shows whether this record can be proxied, must be true if setting `proxied=true`
- `created_on` - The RFC3339 timestamp of when the record was created
- `modified_on` - The RFC3339 timestamp of when the record was last modified
- `metadata` - A key-value map of string metadata Cloudflare associates with the record

» Import

Records can be imported using a composite ID formed of zone name and record ID, e.g.

```
$ terraform import cloudflare_record.default ae36f999674d196762efcc5abb06b345/d41d8cd98f00b2
```

where:

- `ae36f999674d196762efcc5abb06b345` - the zone ID
- `d41d8cd98f00b204e9800998ecf8427e` - record ID as returned by API

» cloudflare_spectrum_application

Provides a Cloudflare Spectrum Application. You can extend the power of Cloudflare's DDoS, TLS, and IP Firewall to your other TCP-based services.

» Example Usage

```
# Define a spectrum application proxies ssh traffic
resource "cloudflare_spectrum_application" "ssh_proxy" {
  zone_id      = var.cloudflare_zone_id
  protocol     = "tcp/22"
  traffic_type = "direct"
  dns {
    type = "CNAME"
    name = "ssh.example.com"
  }

  origin_direct = [
    "tcp://109.151.40.129:22"
  ]
}
```

» Argument Reference

- `zone_id` - (Required) The DNS zone ID to add the application to
- `protocol` - (Required) The port configuration at Cloudflare's edge. e.g. `tcp/22`.
- `dns` - (Required) The name and type of DNS record for the Spectrum application. Fields documented below.
- `origin_direct` - (Optional) A list of destination addresses to the origin. e.g. `tcp://192.0.2.1:22`.
- `origin_dns` - (Optional) A destination DNS addresses to the origin. Fields documented below.
- `origin_port` - (Optional) If using `origin_dns` this is a required attribute. Origin port to proxy traffice to e.g. `22`.
- `tls` - (Optional) TLS configuration option for Cloudflare to connect to your origin. Valid values are: `off`, `flexible`, `full` and `strict`. Defaults to `off`.
- `ip_firewall` - (Optional) Enables the IP Firewall for this application. Defaults to `true`.
- `proxy_protocol` - (Optional) Enables a proxy protocol to the origin. Valid values are: `off`, `v1`, `v2`, and `simple`. Defaults to `off`.
- `traffic_type` - (Optional) Sets application type. Valid values are: `direct`, `http`, `https`. Defaults to `direct`.
- `argo_smart_routing` - (Optional). Enables Argo Smart Routing. Defaults to `false`.
- `edge_ip_connectivity` - (Optional). Choose which types of IP addresses will be provisioned for this subdomain. Valid values are: `all`, `ipv4`, `ipv6`. Defaults to `all`.
- `edge_ips` - (Optional). A list of edge IPs (IPv4 and/or IPv6) to configure Spectrum application to. Requires Bring Your Own IP provisioned.

dns

- `type` - (Required) The type of DNS record associated with the application.

  Valid values: `CNAME`.
- `name` - (Required) The name of the DNS record associated with the application.i.e. `ssh.example.com`.

origin_dns

- `name` - (Required) Fully qualified domain name of the origin e.g. origin-ssh.example.com.

» Attributes Reference

The following attributes are exported:

- `id` - Unique identifier in the API for the spectrum application.

» Import

Spectrum resource can be imported using a zone ID and Application ID, e.g.

```
$ terraform import cloudflare_spectrum_application.example d41d8cd98f00b204e9800998ecf8427e/
```

where:

- `d41d8cd98f00b204e9800998ecf8427e` - zone ID, as returned from API
- `9a7806061c88ada191ed06f989cc3dac` - Application ID

» cloudflare_waf_group

Provides a Cloudflare WAF rule group resource for a particular zone. This can be used to configure firewall behaviour for pre-defined firewall groups.

» Example Usage

```
resource "cloudflare_waf_group" "honey_pot" {
  group_id = "de677e5818985db1285d0e80225f06e5"
  zone_id = "ae36f999674d196762efcc5abb06b345"
  mode = "on"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to apply to.

- `group_id` - (Required) The WAF Rule Group ID.
- `package_id` - (Optional) The ID of the WAF Rule Package that contains the group.
- `mode` - (Optional) The mode of the group, can be one of [”on”, ”off”].

» Attributes Reference

The following attributes are exported:

- `id` - The WAF Rule Group ID, the same as `group_id`.
- `package_id` - The ID of the WAF Rule Package that contains the group.

» Import

WAF Rule Groups can be imported using a composite ID formed of zone ID and the WAF Rule Group ID, e.g.

```
$ terraform import cloudflare_waf_group.honey_pot ae36f999674d196762efcc5abb06b345/de677e581
```

» cloudflare_waf_package

Provides a Cloudflare WAF rule package resource for a particular zone. This can be used to configure firewall behaviour for pre-defined firewall packages.

» Example Usage

```
resource "cloudflare_waf_package" "owasp" {
  package_id = "a25a9a7e9c00afc1fb2e0245519d725b"
  zone_id = "ae36f999674d196762efcc5abb06b345"
  sensitivity = "medium"
  action_mode = "simulate"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to apply to.
- `package_id` - (Required) The WAF Package ID.
- `sensitivity` - (Optional) The sensitivity of the package, can be one of [”high”, ”medium”, ”low”, ”off”].

- `action_mode` - (Optional) The action mode of the package, can be one of [”block”, ”challenge”, ”simulate”].

» Attributes Reference

The following attributes are exported:

- `id` - The WAF Package ID, the same as package_id.

» Import

Packages can be imported using a composite ID formed of zone ID and the WAF Package ID, e.g.

```
$ terraform import cloudflare_waf_package.owasp ae36f999674d196762efcc5abb06b345/a25a9a7e9c0
```

» cloudflare_waf_rule

Provides a Cloudflare WAF rule resource for a particular zone. This can be used to configure firewall behaviour for pre-defined firewall rules.

» Example Usage

```
resource "cloudflare_waf_rule" "100000" {
  rule_id = "100000"
  zone_id = "ae36f999674d196762efcc5abb06b345"
  mode = "simulate"
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to apply to.
- `rule_id` - (Required) The WAF Rule ID.
- `package_id` - (Optional) The ID of the WAF Rule Package that contains the rule.
- `mode` - (Required) The mode of the rule, can be one of [”block”, ”challenge”, ”default”, ”disable”, ”simulate”].

» Attributes Reference

The following attributes are exported:

- `id` - The WAF Rule ID, the same as rule_id.
- `package_id` - The ID of the WAF Rule Package that contains the rule.
- `group_id` - The ID of the WAF Rule Group that contains the rule.

» Import

Rules can be imported using a composite ID formed of zone ID and the WAF Rule ID, e.g.

```
$ terraform import cloudflare_waf_rule.100000 ae36f999674d196762efcc5abb06b345/100000
```

» cloudflare_worker_route

Provides a Cloudflare worker route resource. A route will also require a `cloudflare_worker_script`.

» Example Usage

```
# Runs the specified worker script for all URLs that match `example.com/*`
resource "cloudflare_worker_route" "my_route" {
  zone_id = "d41d8cd98f00b204e9800998ecf8427e"
  pattern = "example.com/*"
  script_name = cloudflare_worker_script.my_script.name
}

resource "cloudflare_worker_script" "my_script" {
  # see "cloudflare_worker_script" documentation ...
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The zone ID to add the route to.
- `pattern` - (Required) The route pattern
- `script_name` Which worker script to run for requests that match the route pattern. If `script_name` is empty, workers will be skipped for matching requests.

» Import

Records can be imported using a composite ID formed of zone ID and route ID, e.g.

```
$ terraform import cloudflare_worker_route.default d41d8cd98f00b204e9800998ecf8427e/9a780606
```

where:

- `d41d8cd98f00b204e9800998ecf8427e` - zone ID
- `9a7806061c88ada191ed06f989cc3dac` - route ID as returned by API

» cloudflare_worker_script

Provides a Cloudflare worker script resource. In order for a script to be active, you'll also need to setup a `cloudflare_worker_route`.

» Example Usage

```
resource "cloudflare_workers_kv_namespace" "my_namespace" {
  title = "example"
}

# Sets the script with the name "script_1"
resource "cloudflare_worker_script" "my_script" {
  name = "script_1"
  content = file("script.js")

  kv_namespace_binding {
    name = "my_binding"
    namespace_id = cloudflare_workers_kv_namespace.my_namespace.id
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name for the script.
- `content` - (Required) The script content.

kv_namespace_binding (optional) block supports:

- `name` - (Required) The name for the binding.
- `namespace_id` - (Required) ID of KV namespace.

» Import

To import a script, use a script name, e.g. `script_name`

```
$ terraform import cloudflare_worker_script.default script_name
```

where:

- `script_name` - the script name

» cloudflare_workers_kv

Provides a Workers KV Pair. *NOTE:* This resource uses the Cloudflare account APIs. This requires setting the `CLOUDFLARE_ACCOUNT_ID` environment variable or `account_id` provider argument.

» Example Usage

```
resource "cloudflare_workers_kv_namespace" "example_ns" {
  title = "test-namespace"
}

resource "cloudflare_workers_kv" "example" {
  namespace_id = cloudflare_workers_kv_namespace.example_ns.id
  key = "test-key"
  value = "test value"
}
```

» Argument Reference

The following arguments are supported:

- `namespace_id` - (Required) The ID of the Workers KV namespace in which you want to create the KV pair
- `key` - (Required) The key name
- `value` - (Required) The string value to be stored in the key

» Import

Workers KV Namespace settings can be imported using it's ID. **Note:** Because the same key can exist in multiple namespaces, the ID is a composite key of the format _.

```
$ terraform import cloudflare_workers_kv_namespace.example beaeb6716c9443eaa4deef11763ccca6_
```

where: - `beaeb6716c9443eaa4deef11763ccca6` is the ID of the namespace and `test-key` is the key

» cloudflare_workers_kv_namespace

Provides a Workers KV Namespace

» Example Usage

```
resource "cloudflare_workers_kv_namespace" "example" {
  title = "test-namespace"
}
```

» Argument Reference

The following arguments are supported:

- `title` - (Required) The name of the namespace you wish to create.

» Import

Workers KV Namespace settings can be imported using it's ID

```
$ terraform import cloudflare_workers_kv_namespace.example beaeb6716c9443eaa4deef11763ccca6
```

where: - `beaeb6716c9443eaa4deef11763ccca6` is the ID of the namespace

» cloudflare_zone

Provides a Cloudflare Zone resource. Zone is the basic resource for working with Cloudflare and is roughly equivalent to a domain name that the user purchases.

» Example Usage

```
resource "cloudflare_zone" "example" {
    zone = "example.com"
}
```

» Argument Reference

The following arguments are supported:

- `zone` - (Required) The DNS zone name which will be added.
- `paused` - (Optional) Boolean of whether this zone is paused (traffic bypasses Cloudflare). Default: false.
- `jump_start` - (Optional) Boolean of whether to scan for DNS records on creation. Ignored after zone is created. Default: false.
- `plan` - (Optional) The name of the commercial plan to apply to the zone, can be updated once the one is created; one of `free`, `pro`, `business`, `enterprise`.
- `type` - A full zone implies that DNS is hosted with Cloudflare. A partial zone is typically a partner-hosted zone or a CNAME setup. Valid values: `full`, `partial`. Default is `full`.

» Attributes Reference

The following attributes are exported:

- `id` - The zone ID.
- `plan` - The name of the commercial plan to apply to the zone.
- `vanity_name_servers` - List of Vanity Nameservers (if set).
- `meta.wildcard_proxiable` - Indicates whether wildcard DNS records can receive Cloudflare security and performance features.
- `meta.phishing_detected` - Indicates if URLs on the zone have been identified as hosting phishing content.
- `status` - Status of the zone. Valid values: `active`, `pending`, `initializing`, `moved`, `deleted`, `deactivated`.
- `name_servers` - Cloudflare-assigned name servers. This is only populated for zones that use Cloudflare DNS.
- `verification_key` - Contains the TXT record value to validate domain ownership. This is only populated for zones of type `partial`.

» Import

Zone resource can be imported using a zone ID, e.g.

```
$ terraform import cloudflare_zone.example d41d8cd98f00b204e9800998ecf8427e
```

where:

- `d41d8cd98f00b204e9800998ecf8427e` - zone ID, as returned from API

» cloudflare_zone_lockdown

Provides a Cloudflare Zone Lockdown resource. Zone Lockdown allows you to define one or more URLs (with wildcard matching on the domain or path) that will only permit access if the request originates from an IP address that matches a safelist of one or more IP addresses and/or IP ranges.

» Example Usage

```
# Restrict access to these endpoints to requests from a known IP address.
resource "cloudflare_zone_lockdown" "endpoint_lockdown" {
  zone_id     = "d41d8cd98f00b204e9800998ecf8427e"
  paused      = "false"
  description = "Restrict access to these endpoints to requests from a known IP address"
  urls = [
    "api.mysite.com/some/endpoint*",
  ]
  configurations {
    target = "ip"
    value = "198.51.100.4"
  }
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to which the access rule should be added.
- `description` - (Optional) A description about the lockdown entry. Typically used as a reminder or explanation for the lockdown.
- `urls` - (Required) A list of simple wildcard patterns to match requests against. The order of the urls is unimportant.
- `configurations` - (Required) A list of IP addresses or IP ranges to match the request against specified in target, value pairs. It's a complex value. See description below. The order of the configuration entries is unimportant.
- `paused` - (Optional) Boolean of whether this zone lockdown is currently paused. Default: false.

Note: Either `zone` or `zone_id` is required and `zone` will be resolved to `zone_id` upon creation.

The list item in **configurations** block supports:

- `target` - (Required) The request property to target. Allowed values: ”ip”, ”ip_range”
- `value` - (Required) The value to target. Depends on target’s type. IP addresses should just be standard IPv4/IPv6 notation i.e. `198.51.100.4` or `2001:db8::/32` and IP ranges in CIDR format i.e. `198.51.0.0/16`.

» Attributes Reference

The following attributes are exported:

- `id` - The access rule ID.

» Import

Records can be imported using a composite ID formed of zone name and record ID, e.g.

```
$ terraform import cloudflare_zone_lockdown  d41d8cd98f00b204e9800998ecf8427e/37cb64fe4a90ad
```

where:

- `d41d8cd98f00b204e9800998ecf8427e` - zone ID
- `37cb64fe4a90adb5ca3afc04f2c82a2f` - zone lockdown ID as returned by API

» cloudflare_zone_settings_override

Provides a resource which customizes Cloudflare zone settings. Note that after destroying this resource Zone Settings will be reset to their initial values.

» Example Usage

```
resource "cloudflare_zone_settings_override" "test" {
    zone_id = var.cloudflare_zone_id
    settings {
        brotli = "on"
        challenge_ttl = 2700
        security_level = "high"
        opportunistic_encryption = "on"
        automatic_https_rewrites = "on"
        mirage = "on"
        waf = "on"
        minify {
            css = "on"
            js = "off"
            html = "off"
        }
        security_header {
            enabled = true
        }
    }
}
```

» Argument Reference

The following arguments are supported:

- `zone_id` - (Required) The DNS zone ID to which apply settings.
- `settings` - (Optional) Settings overrides that will be applied to the zone. If a setting is not specified the existing setting will be used. For a full list of available settings see below.

The **settings** block supports settings that may be applied to the zone. These may be on/off values, unitary fields, string values, integers or nested objects.

» On/Off Values

These can be specified as ”on” or ”off” string. Similar to boolean values, but here the empty string also means to use the existing value. Attributes available:

- `always_online` (default: `on`)
- `always_use_https` (default: `off`)
- `automatic_https_rewrites` (default value depends on the zone’s plan level)
- `brotli` (default: `off`)
- `browser_check` (default: `on`)
- `development_mode` (default: `off`)
- `email_obfuscation` (default: `on`)
- `hotlink_protection` (default: `off`)
- `http2` (default: `off`)
- `http3` (default: `off`)
- `image_resizing` (default: `off`)
- `ip_geolocation` (default: `on`)
- `ipv6` (default: `off`)
- `mirage` (default: `off`)
- `opportunistic_encryption` (default value depends on the zone’s plan level)
- `opportunistic_onion` (default: `off`)

- `origin_error_page_pass_thru` (default: `off`)
- `prefetch_preload` (default: `off`)
- `privacy_pass` (default: `on`)
- `response_buffering` (default: `off`)
- `rocket_loader` (default: `off`)
- `server_side_exclude` (default: `on`)
- `sort_query_string_for_cache` (default: `off`)
- `tls_client_auth` (default: `on`)
- `true_client_ip_header` (default: `off`)
- `waf` (default: `off`)
- `webp` (default: `off`). Note that the value specified will be ignored unless `polish` is turned on (i.e. is ”lossless” or ”lossy”)
- `websockets` (default: `off`)
- `zero_rtt` (default: `off`)

» String Values

- `cache_level`. Allowed values: ”aggressive” (default) - delivers a different resource each time the query string changes, ”basic” - delivers resources from cache when there is no query string, ”simplified” - delivers the same resource to everyone independent of the query string.
- `cname_flattening`. Allowed values: ”flatten_at_root” (default), ”flatten_all”, ”flatten_none”.
- `h2_prioritization`. Allowed values: ”on”, ”off” (default), ”custom”.
- `min_tls_version`. Allowed values: ”1.0” (default), ”1.1”, ”1.2”, ”1.3”.
- `polish`. Allowed values: ”off” (default), ”lossless”, ”lossy”.
- `pseudo_ipv4`. Allowed values: ”off” (default), ”add_header”, ”overwrite_header”.
- `security_level`. Allowed values: ”off” (Enterprise only), ”essentially_off”, ”low”, ”medium” (default), ”high”, ”under_attack”.
- `ssl`. Allowed values: ”off” (default), ”flexible”, ”full”, ”strict”, ”origin_pull”.
- `tls_1_3`. Allowed values: ”off” (default), ”on”, ”zrt”.

» Integer Values

- `browser_cache_ttl` (default: `14400`)
- `challenge_ttl` (default: `1800`)
- `edge_cache_ttl` (default: `7200`)
- `max_upload` (default: `100`)

» Nested Objects

- `minify`

- `mobile_redirect`
- `security_header`

The **minify** attribute supports the following fields:

- `css` (Required) "on"/"off"
- `html` (Required) "on"/"off"
- `js` (Required)"on"/"off"

The **mobile_redirect** attribute supports the following fields:

- `mobile_subdomain` (Required) String value
- `status` (Required) "on"/"off"
- `strip_uri` (Required) true/false

The **security_header** attribute supports the following fields:

- `enabled` (Optional) true/false
- `preload` (Optional) true/false
- `max_age` (Optional) Integer
- `include_subdomains` (Optional) true/false
- `nosniff` (Optional) true/false

» Attributes Reference

The following attributes are exported:

- `id` - The zone ID.
- `initial_settings` - Settings present in the zone at the time the resource is created. This will be used to restore the original settings when this resource is destroyed. Shares the same schema as the `settings` attribute (Above).
- `intial_settings_read_at` - Time when this resource was created and the `initial_settings` were set.
- `readonly_settings` - Which of the current `settings` are not able to be set by the user. Which settings these are is determined by plan level and user permissions.
- `zone_status`. A full zone implies that DNS is hosted with Cloudflare. A partial zone is typically a partner-hosted zone or a CNAME setup.
- `zone_type`. Status of the zone. Valid values: active, pending, initializing, moved, deleted, deactivated.