» akamai_contract

Use `akamai_contract` data source to retrieve a group id.

» Example Usage

Basic usage:

```
data "akamai_contract" "example" {
     group = "group name"
}

resource "akamai_property" "example" {
    contract = "${data.akamai_contract.example.id}"
    ...
}
```

» Argument Reference

The following arguments are supported:

- `group` — (Optional) The group within which the contract can be found.

» Attributes Reference

The following are the return attributes:

- `id` — The contract ID.

» akamai_group

Use `akamai_group` data source to retrieve a group id.

» Example Usage

Basic usage:

```
data "akamai_group" "example" {
    name = "group name"
}
```

```
resource "akamai_property" "example" {
    group    = "${data.akamai_group.example.id}"
    ...
}
```

» Argument Reference

The following arguments are supported:

- `name` — (Required) The group name.
- `contract` — (Optional) The contract ID

» Attributes Reference

The following are the return attributes:

- `id` — The group ID.

» akamai_authorities_set

Use `akamai_authorities_set` datasource to retrieve a contracts authorities set for use when creating new zones.

» Example Usage

Basic usage:

```
data "akamai_authorities_set" "example" {
     contract = "ctr_#####"
}
```

» Argument Reference

The following arguments are supported:

- `contract` — (Required) The contract ID.

» Attributes Reference

The following are the return attributes:

- `authorities` — A list of authorities

» akamai_dns_record

The `akamai_dns_record` provides the resource for configuring a dns record to integrate easily with your existing DNS infrastructure to provide a secure, high performance, highly available and scalable solution for DNS hosting.

» Example Usage

Basic usage:

```
# A record
resource "akamai_dns_record" "origin" {
    zone = "origin.org"
    name = "origin.example.org"
    recordtype =  "A"
    active = true
    ttl =  30
    target = ["192.0.2.42"]
}

# CNAME record
resource "akamai_dns_record" "www" {
    zone = "example.com"
    name = "www.example.com"
    recordtype =  "CNAME"
    active = true
    ttl =  600
    target = "origin.example.org.edgesuite.net"
}
```

» Argument Reference

The following arguments are supported:

- `name` — (Required) The name of the record. The name is an owner name, that is, the name of the node to which this resource record pertains.
- `zone` — (Required) Domain zone, encapsulating any nested subdomains.
- `recordType` — (Required) The DNS record type.
- `active` — (Required,Boolean) Whether the record is active.
- `ttl` — (Required,Boolean) The TTL is a 32-bit signed integer that specifies the time interval that the resource record may be cached before

the source of the information should be consulted again. Zero values are interpreted to mean that the RR can only be used for the transaction in progress, and should not be cached. Zero values can also be used for extremely volatile data.

- `target` — (Required) A domain name that specifies the canonical or primary name for the owner. The owner name is an alias.

» akamai_dns_zone

The `akamai_dns_zone` provides the resource for configuring a dns zone to integrate easily with your existing DNS infrastructure to provide a secure, high performance, highly available and scalable solution for DNS hosting.

» Example Usage

Basic usage:

```
resource "akamai_dns_zone" "demozone" {
    contract = "ctr_XXX"
    group = 100

    zone = "example.com"
    type =  "primary"
    masters = [
      "1.2.3.4",
      "1.2.3.5"
    ]

    comment =  "some comment"
    signandserve = true
}
```

» Argument Reference

The following arguments are supported:

- `contract` — (Required) The contract ID.
- `group` — (Required) The currently selected group ID.

- `zone` — (Required) Domain zone, encapsulating any nested subdomains.

- `type` — (Required) Whether the zone is primary or secondary.
- `masters` — (Required) The names or addresses of the customer's name-servers from which the zone data should be retrieved.
- `comment` — (Required) A descriptive comment.
- `sign_and_serve` — (Required) Whether DNSSEC Sign&Serve is enabled.

» akamai_cp_code

Use `akamai_cp_code` data source to retrieve a group id.

» Example Usage

Basic usage:

```
data "akamai_cp_code" "example" {
     name = "cpcode name"
     group = "grp_#####"
     contract = "ctr_#####"
}

resource "akamai_property" "example" {
    contract = "${data.akamai_cpcode.example.id}"
    ...
}
```

» Argument Reference

The following arguments are supported:

- `name` — (Required) The CP code name.
- `group` — (Required) The group ID
- `contract` — (Required) The contract ID

» Attributes Reference

The following are the return attributes:

- `id` — The CP code ID.

» akamai_property_rules

The `akamai_property_rules` data source allows you to configure a nested block of property rules, criteria, and behaviors. A property's main functionality is encapsulated in its set of rules and rules are composed of the matches and the behavior that applies under those matches.

» Example Usage

Basic usage:

```
data "akamai_property_rules" "example" {
  rules { # Default rule

    behavior { # Downstream Cache behavior
      name = "downstreamCache"
      option { # behavior option
        key = "behavior"
        value = "TUNNEL_ORIGIN"
      }
    }

    rule { # "Performance" child rule
      name = "Performance"

      rule { # "JPEG Images" child rule
        name = "JPEG Images"

        behavior { # Adaptive Image Compression behavior
          name = "adaptiveImageCompression"

          # Options
          option {
            key = "tier1MobileCompressionMethod"
            value = "COMPRESS"
          }
          option {
            key = "tier1MobileCompressionValue"
            value = "80"
          }
          option {
            key = "tier2MobileCompressionMethod"
            value = "COMPRESS"
          }
        }
```

```
      }
    }
  }
}

resource "akamai_property" "example" {
  rules = "${data.akamai_property_rules.example.json}"

  // ...
}
```

» Argument Reference

The following arguments are supported:

The `rule` block supports:

- `is_secure` — (Optional) Whether the property is a secure (Enhanced TLS) property or not (top-level only).
- `criteria` — (Optional) One or more criteria to match requests on.
- `behavior` — (Optional) One or more behaviors to apply to requests that match.
- `rule` — (Optional) Child rules (may be nested five levels deep).

The `criteria` block supports:

- `name` — (Required) The name of the criteria.
- `option` — (Optional) One or more options for the criteria.

The `behavior` block supports:

- `name` — (Required) The name of the behavior.
- `option` — (Optional) One or more options for the behavior.

The `option` block supports:

- `key` — (Required) The option name.
- `value` — (Optional) A single value for the option.
- `values` — (Optional) An array of values for the option.

One of `value` or `values` is required.

» Attributes Reference

The following are the return attributes:

- `json` — The resulting JSON rule tree

» akamai_edge_hostname

The `akamai_edge_hostname` provides the resource for configuring a secure edge hostname that determines how requests for your site, app, or content are mapped to Akamai edge servers.

An edge hostname is the CNAME target you use when directing your end user traffic to Akamai. In a typical DNS CNAME, your www.customer.com hostname corresponds to an edge hostname of www.customer.com.edgesuite.net.

» Example Usage

Basic usage:

```
resource "akamai_edge_hostname" "terraform-demo" {
    product  = "prd_####"
    contract = "ctr_####"
    group    = "grp_####"
    edge_hostname = "www.example.org.edgesuite.net"
}
```

» Argument Reference

The following arguments are supported:

- `contract` — (Required) The contract ID.
- `group` — (Required) The group ID.
- `product` — (Required) The product ID.
- `edge_hostname` — (Required) One or more edge hostnames (must be <= to the number of public hostnames).
- `ipv4` — (Optional) Whether the property supports IPv4 to origin. (Default: `true`).
- `ipv6` — (Optional) Whether the property supports IPv6 to origin. (Default: `false`).
- `certificate` — (Optional) The certificate enrollment ID.

» Attributes Reference

The following attributes are returned:

- `ip_behavior` — Whether the hostname uses `IPV4`, `IPV6` or `IPV6_COMPLIANCE`.

» akamai_property

The `akamai_property` resource represents an Akamai property configuration, allowing you to create, update, and activate properties on the Akamai platform.

» Example Usage

Basic usage:

```
resource "akamai_property" "example" {
    name    = "terraform-demo"
    contact = ["user@example.org"]

    product  = "prd_SPM"
    contract = "ctr_####"
    group    = "grp_####"
    cp_code  = "cpc_#####"

    hostnames = {
      "example.org" = "example.org.edgesuite.net"
      "www.example.org" = "example.org.edgesuite.net"
      "sub.example.org" = "sub.example.org.edgesuite.net"
    }

    rule_format = "v2018-02-27"
    rules       = "${data.local_file.terraform-demo.content}"
    variables   = "${akamai_property_variables.origin.json}"
}
```

» Argument Reference

The following arguments are supported:

» Property Basics

- `account` — (Required) The account ID.
- `contract` — (Optional) The contract ID.
- `group` — (Optional) The group ID.
- `product` — (Optional) The product ID. (Default: `prd_SPM` for Ion)
- `name` — (Required) The property name.
- `contact` — (Required) One or more email addresses to inform about activation changes.

- `hostnames` — (Required) A map of public hostnames to edge hostnames (e.g. `{"example.org" = "example.org.edgesuite.net"}`)
- `is_secure` — (Optional) Whether the property is a secure (Enhanced TLS) property or not.

» Property Rules

- `rules` — (Required) A JSON encoded string of property rules (see: `akamai_property_rules`)
- `rule_format` — (Optional) The rule format to use (more).

In addition the specifying the rule tree in it's entirety, you can also set the default CP Code and Origin explicitly. *This will override your JSON configuration.*

- `cp_code` — (Required) The CP Code id or name to use (or create).
- `origin` — (Optional) The property origin (an origin must be specified to activate a property, but may be defined in your rules block).
  - `hostname` — (Required) The origin hostname.
  - `port` — (Optional) The origin port to connect to (default: 80).
  - `forward_hostname` — (Optional) The value for the Hostname header sent to origin. (default: `ORIGIN_HOSTNAME`).
  - `cache_key_hostname` — (Optional) The hostname uses for the cache key. (default: `ORIGIN_HOSTNAME`).
  - `compress` — (Optional, boolean) Whether origin supports gzip compression (default: `false`).
  - `enable_true_client_ip` — (Optional, boolean) Whether the X-True-Client-IP header should be sent to origin (default: `false`).

You can also define property manager variables. *This will override your JSON configuration.*

- `variables` — (Optional) A JSON encoded string of property manager variable definitions (see: `akamai_property_variables`)

» Attribute Reference

The following attributes are returned:

- `account` — the Account ID under which the property is created.
- `version` — the current version of the property config.
- `production_version` — the current version of the property active on the production network.
- `staging_version` — the current version of the property active on the staging network.
- `edge_hostnames` — the final public hostname to edge hostname map

» akamai_cp_code

The `akamai_cp_code` resource allows you to create or re-use CP Codes.

If the CP Code already exists it will be used instead of creating a new one.

» Example Usage

Basic usage:

```
resource "akamai_cp_code" "cp_code" {
  name = "My CP Code"
  contract = "${akamai_contract.contract.id}"
  group = "${akamai_group.group.id}"
  product = "prd_xxx"
}
```

» Argument Reference

The following arguments are supported:

- `name` — (Required) The CP Code name
- `contract` — (Required) The Contract ID
- `group` — (Required) The Group ID
- `product` — (Required) The Product ID

» akamai_property_activation

The `akamai_property_activation` provides the resource for activating a property in the appropriate environment. Once you are satisfied with any version of a property, an activation deploys it, either to the Akamai staging or production network. You activate a specific version, but the same version can be activated separately more than once.

» Example Usage

Basic usage:

```
resource "akamai_property_activation" "example" {
     property = "${akamai_property.example.id}"
     network  = "STAGING"
     activate = "${var.akamai_property_activate}"
     contact  = ["user@example.org"]
```

```
}
```

» Argument Reference

The following arguments are supported:

- `property` — (Required) The property ID.
- `version` — (Optional) The version to activate. When unset it will activate the latest version of the property.
- `network` — (Optional) Akamai network to activate on. Allowed values `staging` or `production` (Default: `staging`).
- `activate` — (Optional, boolean) Whether to activate the property on the network. (Default: `true`).
- `contact` — (Required) One or more email addresses to inform about activation changes.

» Attribute Reference

The follwing attributes are returned:

- `status` — the current activation status

» akamai_property_variables

The `akamai_property_variables` allows you to implement dynamic functionality. You can perform conditional logic based on the variable's value, and catch any unforeseen errors that execute on the edge at runtime.

Typical uses for variables include:

- Simplify configurations by reducing the number of rules and behaviors.
- Improve self serviceability by replacing or extending advanced metadata.
- Automate redirects, forward path rewrites, HTTP header and cookie manipulation.
- Move origin functionality to the edge.

» Example Usage

Basic usage:

```
resource "akamai_property_variables" "origin" {
  variables {
     variable {
        name        = "PMUSER_ORIGIN"
```

```
        value       = "origin.example.org"
        description = "Origin Hostname"
        hidden      = true
        sensitive   = true
    }
  }
}
```

» Argument Reference

The following arguments are supported:

The `variables` block may contain many `variable` blocks which support the following arguments:

- `name` — (Required) The name of the variable.
- `value` — (Required) The default value to assign to the variable
- `description` — (Optional) A human-readable description
- `hidden` — (Optional) Whether to hide the variable when debugging requests
- `sensitive` — (Optional) Whether to obscure the value when debugging requests

» akamai_gtm_default_datacenter

Use `akamai_gtm_default_datacenter` data source to retrieve default datacenter id and nickname.

» Example Usage

Basic usage:

```
data "akamai_gtm_default_datacenter" "example_ddc" {
     name = "example_domain.akadns.net"
     datacenter = 5400
}

resource "akamai_gtm_cidrmap" "example_cidrmap" {
    domain = "example_domain.akadns.net"
    default_datacenter {
        datacenter_id = data.akamai_gtm_default_datacenter.example.datacenter_id
        nickname = data.akamai_gtm_default_datacenter.example.nickname
    ...
}
```

» Argument Reference

The following arguments are supported:

- `domain` — (Required)
- `domain` — (Optional. Default 5400)
- `datacenter_id` — (Computed - do not configure) default datacenter Id
- `nickname` — (Computed - do not configure) default datacenter nickname

» Attributes Reference

The following are the return attributes:

- `id` — The data resource id. Format: :default_datacenter:

» akamai_gtm_domain

`akamai_gtm_domain` provides the resource for creating, configuring and importing a gtm domain to integrate easily with your existing GTM infrastructure to provide a secure, high performance, highly available and scalable solution for Global Traffic Management. Note: Import requires an ID of the format: `existing_domain_name`

» Example Usage

Basic usage:

```
resource "akamai_gtm_domain" "demodomain" {
    contract = "XXX"
    group = 100
    name = "demo.akadns.net"
    type =  "basic"
    comment =  "some comment"
}
```

» Argument Reference

The following arguments are supported:

Required

- `contract` — The contract ID (if creating domain)
- `group` — The currently selected group ID (if creating domain)

- `name` — Domain name
- `type` — Domain type

Optional

- `wait_on_complete` — (Boolean, Default: true) Wait for transaction to complete
- `comment` — A descriptive comment
- `email_notification_list` — (List)
- `default_timeout_penalty` — (Default: 25)
- `load_imbalance_percentage`
- `default_ssl_client_private_key`
- `default_error_penalty` — (Default: 75)
- `cname_coalescing_enabled` — (Boolean)
- `load_feedback` — (Boolean)
- `default_ssl_client_certificate`
- `end_user_mapping_enabled` — (Boolean)

Computed

The following arguments will be found in terraform.tfstate and can be referenced throughout the configuration. The values can NOT be changed.

- `default_unreachable_threshold`
- `min_pingable_region_fraction`
- `servermonitor_liveness_count`
- `round_robin_prefix`
- `servermonitor_load_count`
- `ping_interval`
- `max_ttl`
- `default_health_max`
- `map_update_interval`
- `max_properties`
- `max_resources`
- `default_error_penalty`
- `max_test_timeout`
- `default_health_multiplier`
- `servermonitor_pool`
- `min_ttl`
- `default_max_unreachable_penalty`
- `default_health_threshold`
- `min_test_interval`
- `ping_packet_size`

» Backing Schema Reference

The GTM Domain backing schema and element descriptions can be found at Akamai Developer Website

» akamai_gtm_property

`akamai_gtm_property` provides the resource for creating, configuring and importing a gtm property to integrate easily with your existing GTM infrastructure to provide a secure, high performance, highly available and scalable solution for Global Traffic Management. Note: Import requires an ID of the format: `existing_domain_name:existing_property_name`

» Example Usage

Basic usage:

```
resource "akamai_gtm_property" "demo_property" {
    domain = "demo_domain.akadns.net"
    name = "demo_property"
    type =  "weighted-round-robin"
    score_aggregation_type = "median"
    handout_limit = 5
    handout_mode = "normal"
    traffic_target {
        datacenter_id = 3131
    }
}
```

» Argument Reference

The following arguments are supported:

Required

- `domain` — Domain name
- `name` — Property name
- `type` — Property type
- `score_aggregation_type`
- `handout_limit`
- `handout_mode`

- `traffic_target` — (multiple allowed)
  - `datacenter_id`
  - `enabled` — (Boolean)
  - `weight`
  - `servers` — (List)
  - `name` — Traffic target name
  - `handout_cname`

Optional

- `liveness_test` — (multiple allowed)
  - `name` — Liveness test name
  - `test_interval`
  - `test_object_protocol`
  - `test_timeout`
  - `answers_required` — (Boolean)
  - `disabled` — (Boolean)
  - `disable_nonstandard_port_warning` — (Boolean)
  - `error_penalty`
  - `http_header` — (multiple allowed) `name value`
  - `http_error3xx` — (Boolean)
  - `http_error4xx` — (Boolean)
  - `http_error5xx` — (Boolean)
  - `peer_certificate_verification` — (Boolean)
  - `recursion_requested` — (Boolean)
  - `request_string`
  - `resource_type`
  - `response_string`
  - `ssl_client_certificate`
  - `ssl_client_private_key`
  - `test_object`
  - `test_object_password`
  - `test_object_port`
  - `test_object_username`
  - `timeout_penalty`
- `wait_on_complete` — (Boolean, Default: true) Wait for transaction to complete
- `failover_delay`
- `failback_delay`
- `ipv6` — (Boolean)
- `stickiness_bonus_percentage`
- `stickiness_bonus_constant`
- `health_threshold`
- `use_computed_targets` — (Boolean)
- `backup_ip`
- `balance_by_download_score` — (Boolean)
- `static_ttl`

- unreachable_threshold
- health_multiplier
- dynamic_ttl
- max_unreachable_penalty
- map_name
- load_imbalance_percentage
- health_max
- cname
- comments
- ghost_demand_reporting
- min_live_fraction
- static_rr_set — (multiple allowed)
  - type
  - ttl
  - rdata — (List)

Computed

The following arguments will be found in terraform.tfstate and can be referenced throughout the configuration. The values can NOT be changed.

- weighted_hash_bits_for_ipv4
- weighted_hash_bits_for_ipv6

» Backing Schema Reference

The GTM Property backing schema and element descriptions can be found at Akamai Developer Website

» akamai_gtm_datacenter

akamai_gtm_datacenter provides the resource for creating, configuring and importing a gtm datacenter to integrate easily with your existing GTM infrastructure to provide a secure, high performance, highly available and scalable solution for Global Traffic Management. Note: Import requires an ID of the format: existing_domain_name:existing_datacenter_id

» Example Usage

Basic usage:

```
resource "akamai_gtm_datacenter" "demo_datacenter" {
    domain = "demo_domain.akadns.net"
    nickname = "demo_datacenter"
}
```

» Argument Reference

The following arguments are supported:

Required

- `domain` — Domain name

Optional

- `wait_on_complete` — (Boolean, Default: true) Wait for transaction to complete
- `nickname` — datacenter nickname
- `default_load_object`
  - `load_object`
  - `load_object_port`
  - `load_servers` — (List)
- `city`
- `clone_of`
- `cloud_server_targeting` — (Boolean)
- `cloud_server_host_header_override` — (Boolean)
- `continent`
- `country`
- `latitude`
- `longitude`
- `state_or_province`

Computed

The following arguments will be found in terraform.tfstate and can be referenced throughout the configuration. The values can NOT be changed.

- `datacenter_id`
- `ping_interval`
- `ping_packet_size`
- `score_penalty`
- `servermonitor_liveness_count`
- `servermonitor_load_count`
- `servermonitor_pool`
- `virtual` — (Boolean)

» Backing Schema Reference

The GTM Datacenter backing schema and element descriptions can be found at Akamai Developer Website

» akamai_gtm_resource

`akamai_gtm_resource` provides the resource for creating, configuring and importing a gtm resource to integrate easily with your existing GTM infrastructure to provide a secure, high performance, highly available and scalable solution for Global Traffic Management. Note: Import requires an ID of the format: `existing_domain_name:existing_resource_name`

» Example Usage

Basic usage:

```
resource "akamai_gtm_resource" "demo_resource" {
    domain = "demo_domain.akadns.net"
    name = "demo_resource"
    aggregation_type = "latest"
    type = "XML load object via HTTP"
}
```

» Argument Reference

The following arguments are supported:

Required

- `domain` — Domain name
- `name` — Resource name
- `aggregation_type`
- `type` — Resource type

Optional

- `wait_on_complete` — (Boolean, Default: true) Wait for transaction to complete
- `resource_instance` — (multiple allowed)
  - `datacenter_id`
  - `load_object`
  - `load_object_port`
  - `load_servers` — (List)
  - `use_default_load_object` — (Boolean)
- `host_header`
- `least_squares_decay`
- `upper_bound`
- `description`
- `leader_string`
- `constrained_property`

- load_imbalance_percent
- max_u_multiplicative_increment
- decay_rate

» Backing Schema Reference

The GTM Resource backing schema and element descriptions can be found at Akamai Developer Website

» akamai_gtm_cidrmap

akamai_gtm_cidrmap provides the resource for creating, configuring and importing a gtm Cidr Map to integrate easily with your existing GTM infrastructure to provide a secure, high performance, highly available and scalable solution for Global Traffic Management. Note: Import requires an ID of the format: existing_domain_name:existing_map_name

» Example Usage

Basic usage:

```
resource "akamai_gtm_cidrmap" "demo_cidrmap" {
    domain = "demo_domain.akadns.net"
    name = "demo_cidr"
    default_datacenter {
        datacenter_id = 5400
        nickname = "All Other CIDR Blocks"
    }
}
```

» Argument Reference

The following arguments are supported:

Required

- domain — Domain name
- name — Resource name
- default_datacenter
  - datacenter_id
  - nickname

Optional

- `wait_on_complete` — (Boolean, Default: true) Wait for transaction to complete
- `assignment` — (multiple allowed)
  - `datacenter_id`
  - `nickname`
  - `blocks` — (List)

» Backing Schema Reference

The GTM Cidr Map backing schema and element descriptions can be found at Akamai Developer Website

» akamai_gtm_asmap

`akamai_gtm_asmap` provides the resource for creating, configuring and importing a gtm AS Map to integrate easily with your existing GTM infrastructure to provide a secure, high performance, highly available and scalable solution for Global Traffic Management. Note: Import requires an ID of the format: `existing_domain_name:existing_map_name`

» Example Usage

Basic usage:

```
resource "akamai_gtm_asmap" "demo_asmap" {
    domain = "demo_domain.akadns.net"
    name = "demo_as"
    default_datacenter {
        datacenter_id = 5400
        nickname = "All Other AS numbers"
    }
}
```

» Argument Reference

The following arguments are supported:

Required

- `domain` — Domain name
- `name` — Resource name
- `default_datacenter`
  - `datacenter_id`

    - `nickname`

Optional

- `wait_on_complete` — (Boolean, Default: true) Wait for transaction to complete
- `assignment` — (multiple allowed)
    - `datacenter_id`
    - `nickname`
    - `as_numbers` — (List)

» Backing Schema Reference

The GTM AS Map backing schema and element descriptions can be found at Akamai Developer Website

» akamai_gtm_geomap

`akamai_gtm_geomap` provides the resource for creating, configuring and importing a gtm Geographic map to integrate easily with your existing GTM infrastructure to provide a secure, high performance, highly available and scalable solution for Global Traffic Management. Note: Import requires an ID of the format: `existing_domain_name:existing_map_name`

» Example Usage

Basic usage:

```
resource "akamai_gtm_geomap" "demo_geomap" {
    domain = "demo_domain.akadns.net"
    name = "demo_geo"
    default_datacenter {
        datacenter_id = 5400
        nickname = "All Others"
    }
}
```

» Argument Reference

The following arguments are supported:

Required

- `domain` — Domain name

- `name` — Resource name
- `default_datacenter`
  - `datacenter_id`
  - `nickname`

Optional

- `wait_on_complete` — (Boolean, Default: true) Wait for transaction to complete
- `assignment` — (multiple allowed)
  - `datacenter_id`
  - `nickname`
  - `countries` — (List)

» Backing Schema Reference

The GTM Geographic Map backing schema and element descriptions can be found at Akamai Developer Website