» bigip_cm_device

`bigip_cm_device` provides details about a specific bigip

This resource is helpful when configuring the BIG-IP device in cluster or in HA mode.

» Example Usage

```
resource "bigip_cm_device" "my_new_device"

        {
            name = "bigip300.f5.com"
            configsync_ip = "2.2.2.2"
            mirror_ip = "10.10.10.10"
            mirror_secondary_ip = "11.11.11.11"
        }
```

» bigip_cm_devicegroup

`bigip_cm_devicegroup` A device group is a collection of BIG-IP devices that are configured to securely synchronize their BIG-IP configuration data, and fail over when needed.

» Example Usage

```
resource "bigip_cm_devicegroup" "my_new_devicegroup"

        {
            name = "sanjose_devicegroup"
            auto_sync = "enabled"
            full_load_on_sync = "true"
            type = "sync-only"
            device  { name = "bigip1.cisco.com"}
            device  { name = "bigip200.f5.com"}
        }
```

» Argument Reference

- `bigip_cm_devicegroup` - Is the resource used to configure new device group on the BIG-IP.

- `name` - Is the name of the device Group
- `auto_sync` - Specifies if the device-group will automatically sync configuration data to its members
- `type` - Specifies if the device-group will be used for failover or resource syncing
- `device` - Name of the device to be included in device group, this need to be configured before using devicegroup resource

» bigip_ltm_dns

`bigip_ltm_dns` Configures DNS server on F5 BIG-IP

» Example Usage

```
resource "bigip_ltm_dns" "dns1" {
   description = "/Common/DNS1"
   name_servers = ["1.1.1.1"]
   numberof_dots = 2
   search = ["f5.com"]
}
```

» Argument Reference

- `description`- Provide description for your DNS server
- `name_servers` - Name or IP address of the DNS server
- `number_of_dots` - Configures the number of dots needed in a name before an initial absolute query will be made.
- `search` - Specify what domains you want to search

» bigip_ltm_datagroup

`bigip_ltm_datagroup` Manages internal (in-line) datagroup configuration

Resource should be named with their "full path". The full path is the combination of the partition + name of the resource, for example /Common/mydatagroup.

» Example Usage

```
resource "bigip_ltm_datagroup" "datagroup" {
  name = "/Common/dgx2"
  type = "string"

  record {
    name = "abc.com"
    data = "pool1"
  }

  record {
    name = "test"
    value = "123"
  }
}
```

» Argument Reference

- `name` - (Required) Name of the datagroup
- `type` - (Required) datagroup type (applies to the `name` field of the record), supports: `string`, `ip` or `integer`
- `record` - (Optional) a set of `name` and `data` attributes, name must be of type specified by the `type` attributed (`string`, `ip` and `integer`), data is optional and can take any value, multiple `record` sets can be specified as needed.
  - `name` - (Required if `record` defined), sets the value of the record's `name` attribute, must be of type defined in `type` attribute
  - `data` - (Optional if `record` defined), sets the value of the record's `data` attribute, specifying a value here will create a record in the form of `name := data`

» bigip_ltm_irule

`bigip_ltm_irule` Creates iRule on BIG-IP F5 device

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
# Loading from a file is the preferred method
resource "bigip_ltm_irule" "rule" {
  name = "/Common/terraform_irule"
  irule = "${file("myirule.tcl")}"
}

resource "bigip_ltm_irule" "rule2" {
  name = "/Common/terraform_irule2"
  irule = <<EOF
when CLIENT_ACCEPTED {
     log local0. "test"
   }
EOF
}

myirule.tcl

when HTTP_REQUEST {

  if { [string tolower [HTTP::header value Upgrade]] equals "websocket" } {
    HTTP::disable
#    ASM::disable
    log local0. "[IP::client_addr] - Connection upgraded to websocket protocol. Disabling AS
  } else {
    HTTP::enable
#    ASM::enable
    log local0. "[IP::client_addr] - Regular HTTP request. ASM-checks and HTTP protocol enab
  }
}
```

» Argument Reference

- `name` - (Required) Name of the iRule
- `irule` - (Required) Body of the iRule

» bigip_ltm_monitor

`bigip_ltm_monitor` Configures a custom monitor for use by health checks.

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_monitor" "monitor" {
  name = "/Common/terraform_monitor"
  parent = "/Common/http"
  send = "GET /some/path\r\n"
  timeout = "999"
  interval = "999"
  destination = "1.2.3.4:1234"
}

resource "bigip_ltm_monitor" "test-ftp-monitor" {
    name = "/Common/ftp-test"
    parent = "/Common/ftp"
    interval          = 5
    time_until_up     = 0
    timeout           = 16
    destination       = "*:8008"
    filename = "somefile"
}
```

» Argument Reference

- `name` (Required) Name of the monitor
- `parent` - (Required) Existing LTM monitor to inherit from
- `interval` - (Optional) Check interval in seconds
- `timeout` - (Optional) Timeout in seconds
- `send` - (Optional) Request string to send
- `receive` - (Optional) Expected response string
- `receive_disable` - (Optional)
- `reverse` - (Optional)
- `transparent` - (Optional)
- `manual_resume` - (Optional)
- `ip_dscp` - (Optional)
- `time_until_up` - (Optional)

- `destination` - (Optional) Specify an alias address for monitoring
- `compatibility` - (Optional) Specifies, when enabled, that the SSL options setting (in OpenSSL) is set to ALL. Accepts 'enabled' or 'disabled' values, the default value is 'enabled'.
- `filename` - (Optional) Specifies the full path and file name of the file that the system attempts to download. The health check is successful if the system can download the file.
- `mode` - (Optional) Specifies the data transfer process (DTP) mode. The default value is passive. The options are passive (Specifies that the monitor sends a data transfer request to the FTP server. When the FTP server receives the request, the FTP server then initiates and establishes the data connection.) and active (Specifies that the monitor initiates and establishes the data connection with the FTP server.).

» bigip_ltm_node

`bigip_ltm_node` Manages a node configuration

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_node" "node" {
  name = "/Common/terraform_node1"
  address = "10.10.10.10"
  connection_limit = "0"
  dynamic_ratio = "1"
  monitor = "default"
  rate_limit = "disabled"
  fqdn = { address_family = "ipv4", interval = "3000" }
}
```

» Argument Reference

- `name` - (Required) Name of the node
- `address` - (Required) IP or hostname of the node
- `connection_limit` - (Optional) Specifies the maximum number of connections allowed for the node or node address.

- `dynamic_ratio` - (Optional) Specifies the fixed ratio value used for a node during ratio load balancing.
- `monitor` - (Optional) specifies the name of the monitor or monitor rule that you want to associate with the node.
- `rate_limit`- (Optional) Specifies the maximum number of connections per second allowed for a node or node address. The default value is 'disabled'.
- `state` - (Optional) Default is "user-up" you can set to "user-down" if you want to disable
- Below attributes needs to be configured under fqdn option.
- `interval` - (Optional) Specifies the amount of time before sending the next DNS query. Default is 3600. This needs to be specified inside the fqdn (fully qualified domain name).
- `address_family` - (Optional) Specifies the node's address family. The default is 'unspecified', or IP-agnostic. This needs to be specified inside the fqdn (fully qualified domain name).

» bigip_ltm_persistence_profile_cookie

Configures a cookie persistence profile

» Example

```
resource "bigip_ltm_persistence_profile_cookie" "test_ppcookie" {
    name = "/Common/terraform_cookie"
    defaults_from = "/Common/cookie"
    match_across_pools = "enabled"
    match_across_services = "enabled"
    match_across_virtuals = "enabled"
    timeout = 3600
    override_conn_limit = "enabled"
    always_send = "enabled"
    cookie_encryption = "required"
    cookie_encryption_passphrase = "iam"
    cookie_name = "ham"
    expiration = "1:0:0"
    hash_length = 0

    lifecycle {
        ignore_changes = [ "cookie_encryption_passphrase" ]
    }
```

```
}
```

» Reference

`name` - (Required) Name of the virtual address

`defaults_from` - (Required) Parent cookie persistence profile

`match_across_pools` (Optional) (enabled or disabled) match across pools with given persistence record

`match_across_services` (Optional) (enabled or disabled) match across services with given persistence record

`match_across_virtuals` (Optional) (enabled or disabled) match across virtual servers with given persistence record

`mirror` (Optional) (enabled or disabled) mirror persistence record

`timeout` (Optional) (enabled or disabled) Timeout for persistence of the session in seconds

`override_conn_limit` (Optional) (enabled or disabled) Enable or dissable pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.

`always_send` (Optional) (enabled or disabled) always send cookies

`cookie_encryption` (Optional) (required, preferred, or disabled) To required, preferred, or disabled policy for cookie encryption

`cookie_encryption_passphrase` (Optional) (required, preferred, or disabled) Passphrase for encrypted cookies. The field is encrypted on the server and will always return differently then set. If this is configured specify `ignore_changes` under the `lifecycle` block to ignore returned encrypted value.

`cookie_name` (Optional) Name of the cookie to track persistence

`expiration` (Optional) Expiration TTL for cookie specified in DAY:HOUR:MIN:SECONDS (Examples: 1:0:0:0 one day, 1:0:0 one hour, 30:0 thirty minutes)

`hash_length` (Optional) (Integer) Length of hash to apply to cookie

`hash_offset` (Optional) (Integer) Number of characters to skip in the cookie for the hash

`httponly` (Optional) (enabled or disabled) Sending only over http

» bigip_ltm_persistence_profile_dstaddr

Configures a cookie persistence profile

» Example

```
resource "bigip_ltm_persistence_profile_dstaddr" "dstaddr" {
    name = "/Common/terraform_ppdstaddr"
    defaults_from = "/Common/dest_addr"
    match_across_pools = "enabled"
    match_across_services = "enabled"
    match_across_virtuals = "enabled"
    mirror = "enabled"
    timeout = 3600
    override_conn_limit = "enabled"
    hash_algorithm = "carp"
    mask = "255.255.255.255"

}
```

» Reference

`name` - (Required) Name of the virtual address

`defaults_from` - (Optional) Specifies the existing profile from which the system imports settings for the new profile.

`match_across_pools` (Optional) (enabled or disabled) match across pools with given persistence record

`match_across_services` (Optional) (enabled or disabled) match across services with given persistence record

`match_across_virtuals` (Optional) (enabled or disabled) match across virtual servers with given persistence record

`mirror` (Optional) (enabled or disabled) mirror persistence record

`timeout` (Optional) (enabled or disabled) Timeout for persistence of the session in seconds

`override_conn_limit` (Optional) (enabled or disabled) Enable or dissable pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.

» bigip_ltm_persistence_profile_srcaddr

Configures a source address persistence profile

» Example

```
resource "bigip_ltm_persistence_profile_srcaddr" "srcaddr" {
    name = "/Common/terraform_srcaddr"
    defaults_from = "/Common/source_addr"
    match_across_pools = "enabled"
    match_across_services = "enabled"
    match_across_virtuals = "enabled"
    mirror = "enabled"
    timeout = 3600
    override_conn_limit = "enabled"
    hash_algorithm = "carp"
    map_proxies = "enabled"
    mask = "255.255.255.255"
}
```

» Reference

`name` - (Required) Name of the virtual address

`defaults_from` - (Required) Parent cookie persistence profile

`match_across_pools` (Optional) (enabled or disabled) match across pools with given persistence record

`match_across_services` (Optional) (enabled or disabled) match across services with given persistence record

`match_across_virtuals` (Optional) (enabled or disabled) match across virtual servers with given persistence record

`mirror` (Optional) (enabled or disabled) mirror persistence record

`timeout` (Optional) (enabled or disabled) Timeout for persistence of the session in seconds

`override_conn_limit` (Optional) (enabled or disabled) Enable or dissable pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.

`hash_algorithm` (Optional) Specify the hash algorithm

`mask` (Optional) Identify a range of source IP addresses to manage together as a single source address affinity persistent connection when connecting to the pool. Must be a valid IPv4 or IPv6 mask.

`map_proxies` (Optional) (enabled or disabled) Directs all to the same single pool member

» bigip_ltm_persistence_profile_ssl

Configures an SSL persistence profile

» Example

```
resource "bigip_ltm_persistence_profile_ssl" "ppssl" {
    name = "/Common/terraform_ssl"
    defaults_from = "/Common/ssl"
    match_across_pools = "enabled"
    match_across_services = "enabled"
    match_across_virtuals = "enabled"
    mirror = "enabled"
    timeout = 3600
    override_conn_limit = "enabled"
}
```

» Reference

`name` - (Required) Name of the virtual address

`defaults_from` - (Required) Parent cookie persistence profile

`match_across_pools` (Optional) (enabled or disabled) match across pools with given persistence record

`match_across_services` (Optional) (enabled or disabled) match across services with given persistence record

`match_across_virtuals` (Optional) (enabled or disabled) match across virtual servers with given persistence record

`mirror` (Optional) (enabled or disabled) mirror persistence record

`timeout` (Optional) (enabled or disabled) Timeout for persistence of the session in seconds

`override_conn_limit` (Optional) (enabled or disabled) Enable or dissable pool member connection limits are overridden for persisted clients. Per-virtual connection limits remain hard limits and are not overridden.

» bigip_ltm_policy

`bigip_ltm_policy` Configures Virtual Server

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_policy" "test-policy" {
name = "my_policy"
strategy = "first-match"
 requires = ["http"]
published_copy = "Drafts/my_policy"
 controls = ["forwarding"]
 rule  {
 name = "rule6"

  action = {
    tm_name = "20"
    forward = true
     pool = "/Common/mypool"
  }
 }
depends_on = ["bigip_ltm_pool.mypool"]
}
```

» Argument Reference

- `name`- (Required) Name of the Policy
- `strategy` - (Optional) Specifies the match strategy
- `requires` - (Optional) Specifies the protocol
- `published_copy` - (Optional) If you want to publish the policy else it will be deployed in Drafts mode.
- `controls` - (Optional) Specifies the controls

- `rule` - (Optional) Rules can be applied using the policy
- `tm_name` - (Required) If Rule is used then you need to provide the tm_name it can be any value
- `forward` - (Optional) This action will affect forwarding.
- `pool` - (Optional) This action will direct the stream to this pool.

» bigip_ltm_pool

`bigip_ltm_pool` Manages a pool configuration.

Resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_pool" "pool" {
  name = "/Common/terraform-pool"
  load_balancing_mode = "round-robin"
  monitors = ["${bigip_ltm_monitor.monitor.name}","${bigip_ltm_monitor.monitor2.name}"]
  allow_snat = "yes"
  allow_nat = "yes"
}
```

» Argument Reference

- `name` - (Required) Name of the pool
- `monitors` - (Optional) List of monitor names to associate with the pool
- `allow_nat` - (Optional)
- `allow_snat` - (Optional)
- `load_balancing_mode` - (Optional, Default = round-robin)

» bigip_ltm_pool_attachment

`bigip_ltm_pool_attachment` Manages nodes membership in pools

Resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_pool_attachment" "node-terraform_pool" {
  pool = "/Common/terraform-pool"
  node = "${bigip_ltm_node.node.name}:80"
}
```

» Argument Reference

- `pool` - (Required) Name of the pool in /Partition/Name format
- `node` - (Required) Node to add to the pool in /Partition/NodeName:Port format (e.g. /Common/Node01:80)

» bigip_ltm_profile_fasthttp

`bigip_ltm_profile_fasthttp` Configures a custom profile_fasthttp for use by health checks.

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_profile_fasthttp" "sjfasthttpprofile"

        {
            name = "sjfasthttpprofile"
            defaults_from = "/Common/fasthttp"
            idle_timeout = 300
            connpoolidle_timeoutoverride    = 0
            connpool_maxreuse = 2
            connpool_maxsize  = 2048
            connpool_minsize = 0
            connpool_replenish = "enabled"
            connpool_step = 4
            forcehttp_10response = "disabled"
            maxheader_size = 32768
      }
```

» Argument Reference

- `name` (Required) Name of the profile_fasthttp

- `defaults_from` - (Optional) Specifies the profile that you want to use as the parent profile. Your new profile inherits all settings and values from the parent profile specified.
- `connpoolidle_timeoutoverride` - (Optional) Specifies the number of seconds after which a server-side connection in a OneConnect pool is eligible for deletion, when the connection has no traffic.The value of this option overrides the idle-timeout value that you specify. The default value is 0 (zero) seconds, which disables the override setting.
- `connpool_maxreuse` - (Optional) Specifies the maximum number of times that the system can re-use a current connection. The default value is 0 (zero).
- `connpool_maxsize` - (Optional) Specifies the maximum number of connections to a load balancing pool. A setting of 0 specifies that a pool can accept an unlimited number of connections. The default value is 2048.
- `connpool_replenish` - (Optional) The default value is enabled. When this option is enabled, the system replenishes the number of connections to a load balancing pool to the number of connections that existed when the server closed the connection to the pool. When disabled, the system replenishes the connection that was closed by the server, only when there are fewer connections to the pool than the number of connections set in the connpool-min-size connections option. Also see the connpool-min-size option..
- `idle_timeout` - (Optional) Specifies an idle timeout in seconds. This setting specifies the number of seconds that a connection is idle before the connection is eligible for deletion.When you specify an idle timeout for the Fast L4 profile, the value must be greater than the bigdb database variable Pva.Scrub time in msec for it to work properly.The default value is 300 seconds.
- `connpool_minsize` - (Optional) Specifies the minimum number of connections to a load balancing pool. A setting of 0 specifies that there is no minimum. The default value is 10.
- `connpool_step` - (Optional) Specifies the increment in which the system makes additional connections available, when all available connections are in use. The default value is 4.
- `forcehttp_10response` - (Optional) Specifies whether to rewrite the HTTP version in the status line of the server to HTTP 1.0 to discourage the client from pipelining or chunking data. The default value is disabled.
- `maxheader_size` - (Optional) Specifies the maximum amount of HTTP header data that the system buffers before making a load balancing decision. The default setting is 32768.

» bigip_ltm_profile_fastl4

`bigip_ltm_profile_fastl4` Configures a custom profile_fastl4 for use by health checks.

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_profile_fastl4" "profile_fastl4" {
            name = "/Common/sjfastl4profile"
            partition = "Common"
            defaults_from = "/Common/fastL4"
            client_timeout = 40
            explicitflow_migration = "enabled"
            hardware_syncookie = "enabled"
            idle_timeout = "200"
            iptos_toclient = "pass-through"
            iptos_toserver = "pass-through"
            keepalive_interval = "disabled"  //This cannot take enabled
}
```

» Argument Reference

- `name` (Required) Name of the profile_fastl4
- `defaults_from` - (Optional) Specifies the profile that you want to use as the parent profile. Your new profile inherits all settings and values from the parent profile specified.
- `partition` - (Optional) Displays the administrative partition within which this profile resides
- `client_timeout` - (Optional) Specifies late binding client timeout in seconds. This setting specifies the number of seconds allowed for a client to transmit enough data to select a server when late binding is enabled. If it expires timeout-recovery mode will dictate what action to take.
- `explicitflow_migration` - (Optional) Enables or disables late binding explicit flow migration that allows iRules to control when flows move from software to hardware. Explicit flow migration is disabled by default hence BIG-IP automatically migrates flows from software to hardware.
- `hardware_syncookie` - (Optional) Enables or disables hardware SYN cookie support when PVA10 is present on the system. Note that when you

set the hardware syncookie option to enabled, you may also want to set the following bigdb database variables using the "/sys modify db" command, based on your requirements: pva.SynCookies.Full.ConnectionThreshold (default: 500000), pva.SynCookies.Assist.ConnectionThreshold (default: 500000) pva.SynCookies.ClientWindow (default: 0). The default value is disabled.

- `idle_timeout` - (Optional) Specifies an idle timeout in seconds. This setting specifies the number of seconds that a connection is idle before the connection is eligible for deletion.When you specify an idle timeout for the Fast L4 profile, the value must be greater than the bigdb database variable Pva.Scrub time in msec for it to work properly.The default value is 300 seconds.
- `iptos_toclient` - (Optional) Specifies an IP ToS number for the client side. This option specifies the Type of Service level that the traffic management system assigns to IP packets when sending them to clients. The default value is 65535 (pass-through), which indicates, do not modify.
- `iptos_toserver` - (Optional) Specifies an IP ToS number for the server side. This setting specifies the Type of Service level that the traffic management system assigns to IP packets when sending them to servers. The default value is 65535 (pass-through), which indicates, do not modify.
- `keepalive_interval` - (Optional) Specifies the keep alive probe interval, in seconds. The default value is disabled (0 seconds).

» bigip__ltm__profile__http2

`bigip_ltm_profile_http2` Configures a custom profile_http2 for use by health checks.

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_profile_http2" "nyhttp2"

        {
            name = "/Common/NewYork_http2"
            defaults_from = "/Common/http2"
            concurrent_streams_per_connection = 10
            connection_idle_timeout= 30
            activation_modes = ["alpn","npn"]
        }
```

» Argument Reference

- `name` (Required) Name of the profile_http2
- `defaults_from` - (Required) Specifies the profile that you want to use as the parent profile. Your new profile inherits all settings and values from the parent profile specified.
- `concurrent_streams_per_connection` - (Optional) Specifies how many concurrent requests are allowed to be outstanding on a single HTTP/2 connection.
- `connection_idle_timeout` - (Optional) Specifies the number of seconds that a connection is idle before the connection is eligible for deletion..
- `connpool_maxsize` - (Optional) Specifies the maximum number of connections to a load balancing pool. A setting of 0 specifies that a pool can accept an unlimited number of connections. The default value is 2048.
- `activation_modes` - (Optional) Specifies what will cause an incoming connection to be handled as a HTTP/2 connection. The default values npn and alpn specify that the TLS next-protocol-negotiation and application-layer-protocol-negotiation extensions will be used.

» bigip_ltm_profile_httpcompress

`bigip_ltm_profile_httpcompress` Virtual server HTTP compression profile configuration

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_profile_httpcompress" "sjhttpcompression"

       {
           name = "/Common/sjhttpcompression2"
           defaults_from = "/Common/httpcompression"
           uri_exclude   = ["www.abc.f5.com", "www.abc2.f5.com"]
           uri_include   = ["www.xyzbc.cisco.com"]
           content_type_include = ["nicecontent.com"]
           content_type_exclude = ["nicecontentexclude.com"]
       }
```

» Argument Reference

- `name` (Required) Name of the profile_httpcompress
- `defaults_from` - (Optional) Specifies the profile that you want to use as the parent profile. Your new profile inherits all settings and values from the parent profile specified.
- `uri_exclude` - (Optional) Disables compression on a specified list of HTTP Request-URI responses. Use a regular expression to specify a list of URIs you do not want to compress.
- `uri_include` - (Optional) Enables compression on a specified list of HTTP Request-URI responses. Use a regular expression to specify a list of URIs you want to compress.
- `content_type_include` - (Optional) Specifies a list of content types for compression of HTTP Content-Type responses. Use a string list to specify a list of content types you want to compress.
- `content_type_exclude` - (Optional) Excludes a specified list of content types from compression of HTTP Content-Type responses. Use a string list to specify a list of content types you want to compress.

» bigip_ltm_profile_oneconnect

`bigip_ltm_profile_oneconnect` Configures a custom profile_oneconnect for use by health checks.

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_profile_oneconnect" "oneconnect-sanjose"

        {
            name = "sanjose"
            partition = "Common"
            defaults_from = "/Common/oneconnect"
            idle_timeout_override = "disabled"
            max_age = 3600
            max_reuse = 1000
            max_size = 1000
            share_pools = "disabled"
            source_mask = "255.255.255.255"
```

```
}
```

» Argument Reference

- `name` (Required) Name of the profile_oneconnect
- `partition` - (Optional) Displays the administrative partition within which this profile resides
- `defaults_from` - (Optional) Specifies the profile that you want to use as the parent profile. Your new profile inherits all settings and values from the parent profile specified.
- `idle_timeout_override` - (Optional) Specifies the number of seconds that a connection is idle before the connection flow is eligible for deletion. Possible values are disabled, indefinite, or a numeric value that you specify. The default value is disabled.
- `share_pools` - (Optional) Specify if you want to share the pool, default value is "disabled"
- `max_age` - (Optional) Specifies the maximum age in number of seconds allowed for a connection in the connection reuse pool. For any connection with an age higher than this value, the system removes that connection from the reuse pool. The default value is 86400.
- `max_reuse` - (Optional) Specifies the maximum number of times that a server-side connection can be reused. The default value is 1000.
- `max_size` - (Optional) Specifies the maximum number of connections that the system holds in the connection reuse pool. If the pool is already full, then the server-side connection closes after the response is completed. The default value is 10000.
- `source_mask` - (Optional) Specifies a source IP mask. The default value is 0.0.0.0. The system applies the value of this option to the source address to determine its eligibility for reuse. A mask of 0.0.0.0 causes the system to share reused connections across all clients. A host mask (all 1's in binary), causes the system to share only those reused connections originating from the same client IP address.

» bigip_ltm_profile_tcp

`bigip_ltm_profile_tcp` Configures a custom profile_tcp for use by health checks.

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_profile_tcp" "sanjose-tcp-lan-profile"

        {
            name = "sanjose-tcp-lan-profile"
            idle_timeout = 200
            close_wait_timeout = 5
            finwait_2timeout = 5
            finwait_timeout = 300
            keepalive_interval = 1700
            deferred_accept = "enabled"
            fast_open = "enabled"
        }
```

» Argument Reference

- `name` (Required) Name of the profile_tcp
- `partition` - (Optional) Displays the administrative partition within which this profile resides
- `defaults_from` - (Optional) Specifies the profile that you want to use as the parent profile. Your new profile inherits all settings and values from the parent profile specified.
- `idle_timeout` - (Optional) Specifies the number of seconds that a connection is idle before the connection is eligible for deletion. The default value is 300 seconds.
- `close_wait_timeout` - (Optional) Specifies the number of seconds that a connection remains in a LAST-ACK state before quitting. A value of 0 represents a term of forever (or until the maxrtx of the FIN state). The default value is 5 seconds.
- `finwait_timeout` - (Optional) Specifies the number of seconds that a connection is in the FIN-WAIT-1 or closing state before quitting. The default value is 5 seconds. A value of 0 (zero) represents a term of forever (or until the maxrtx of the FIN state). You can also specify immediate or indefinite.
- `finwait_2timeout` - (Optional) Specifies the number of seconds that a connection is in the FIN-WAIT-2 state before quitting. The default value is 300 seconds. A value of 0 (zero) represents a term of forever (or until the maxrtx of the FIN state).
- `keepalive_interval` - (Optional) Specifies the keep alive probe interval, in seconds. The default value is 1800 seconds.

- `fast_open` - (Optional) When enabled, permits TCP Fast Open, allowing properly equipped TCP clients to send data with the SYN packet.
- `deferred_accept` - (Optional) Specifies, when enabled, that the system defers allocation of the connection chain context until the client response is received. This option is useful for dealing with 3-way handshake DOS attacks. The default value is disabled.

» bigip_ltm_snat

`bigip_ltm_snat` Manages a snat configuration

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_snat" "snat3" {
  // this is using snatpool translation is not required
  name = "snat3"
  origins = ["6.1.6.6"]
  mirror = "false"
  snatpool = "/Common/sanjaysnatpool"
  vlans = ["test-vlan"]
}
```

» Argument Reference

- `name` - (Required) Name of the snat
- `partition` - (Optional) Displays the administrative partition within which this profile resides
- `origins` - (Optional) IP or hostname of the snat
- `snatpool` - (Optional) Specifies the name of a SNAT pool. You can only use this option when automap and translation are not used.
- `mirror` - (Optional) Enables or disables mirroring of SNAT connections.
- `autolasthop` -(Optional) Specifies whether to automatically map last hop for pools or not. The default is to use next level's default.
- `sourceport` - (Optional) Specifies whether the system preserves the source port of the connection. The default is preserve. Use of the preserve-strict setting should be restricted to UDP only under very special circumstances

such as nPath or transparent (that is, no translation of any other L3/L4 field), where there is a 1:1 relationship between virtual IP addresses and node addresses, or when clustered multi-processing (CMP) is disabled. The change setting is useful for obfuscating internal network addresses.

- `translation` - (Optional) Specifies the name of a translated IP address. Note that translated addresses are outside the traffic management system. You can only use this option when automap and snatpool are not used.
- `vlansdisabled` - (Optional) Disables the SNAT on all VLANs.
- `vlans` - (Optional) Specifies the name of the VLAN to which you want to assign the SNAT. The default is vlans-enabled.

» bigip_ltm_snatpool

`bigip_ltm_snatpool` Collections of SNAT translation addresses

Resource should be named with their "full path". The full path is the combination of the partition + name of the resource, for example /Common/my-snatpool.

» Example Usage

```
 resource "bigip_ltm_snatpoolpool" "snatpool_sanjose" {
  name = "/Common/snatpool_sanjose"
  members = ["191.1.1.1","194.2.2.2"]
}
```

» Argument Reference

- `name` - (Required) Name of the snatpool
- `members` - (Required) Specifies a translation address to add to or delete from a SNAT pool (at least one address is required)

» bigip_ltm_virtual_address

`bigip_ltm_virtual_address` Configures Virtual Server

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_virtual_address" "vs_va" {
    name = "/Common/vs_va"
    advertize_route = true
}
```

» Argument Reference

- `name` - (Required) Name of the virtual address
- `description` - (Optional) Description of the virtual address
- `advertize_route` - (Optional) Enabled dynamic routing of the address
- `conn_limit` - (Optional, Default=0) Max number of connections for virtual address
- `enabled` - (Optional, Default=true) Enable or disable the virtual address
- `arp` - (Optional, Default=true) Enable or disable ARP for the virtual address
- `auto_delete` - (Optional, Default=true) Automatically delete the virtual address with the virtual server
- `icmp_echo` - (Optional, Default=true) Enable/Disable ICMP response to the virtual address
- `traffic_group` - (Optional, Default=/Common/traffic-group-1) Specify the partition and traffic group

» bigip_ltm_virtual_server

`bigip_ltm_virtual_server` Configures Virtual Server

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_ltm_virtual_server" "http" {
  name = "/Common/terraform_vs_http"
  destination = "10.12.12.12"
  port = 80
  pool = "/Common/the-default-pool"
}
```

```
# A Virtual server with SSL enabled
resource "bigip_ltm_virtual_server" "https" {
  name = "/Common/terraform_vs_https"
  destination = "${var.vip_ip}"
  port = 443
  pool = "${var.pool}"
  profiles = ["/Common/tcp","/Common/my-awesome-ssl-cert","/Common/http"]
  source_address_translation = "automap"
  translate_address = "enabled"
  translate_port = "enabled"
  vlans_disabled = true
}

# A Virtual server with separate client and server profiles
 resource "bigip_ltm_virtual_server" "https" {
  name = "/Common/terraform_vs_https"
  destination = "10.255.255.254"
  port = 443
  client_profiles = ["/Common/clientssl"]
  server_profiles = ["/Common/serverssl"]
  source_address_translation = "automap"
}
```

» Argument Reference

- `name`- (Required) Name of the virtual server
- `port` - (Required) Listen port for the virtual server
- `destination` - (Required) Destination IP
- `pool` - (Optional) Default pool name
- `mask` - (Optional) Mask can either be in CIDR notation or decimal, i.e.: 24 or 255.255.255.0. A CIDR mask of 0 is the same as 0.0.0.0
- `source_address_translation` - (Optional) Can be either omitted for none or the values automap or snat
- `translate_address` - Enables or disables address translation for the virtual server. Turn address translation off for a virtual server if you want to use the virtual server to load balance connections to any address. This option is useful when the system is load balancing devices that have the same IP address.

- `translate_port` - Enables or disables port translation. Turn port translation off for a virtual server if you want to use the virtual server to load balance connections to any service
- `ip_protocol`- (Optional) Specify the IP protocol to use with the the virtual server (all, tcp, or udp are valid)
- `profiles` - (Optional) List of profiles associated both client and server contexts on the virtual server. This includes protocol, ssl, http, etc.
- `client_profiles` - (Optional) List of client context profiles associated on the virtual server. Not mutually exclusive with profiles and server_profiles
- `server_profiles` - (Optional) List of server context profiles associated on the virtual server. Not mutually exclusive with profiles and client_profiles
- `source` - (Optional) Specifies an IP address or network from which the virtual server will accept traffic.
- `rules` - (Optional) The iRules you want run on this virtual server. iRules help automate the intercepting, processing, and routing of application traffic.
- `snatpool` - (Optional) Specifies the name of an existing SNAT pool that you want the virtual server to use to implement selective and intelligent SNATs. DEPRECATED - see Virtual Server Property Groups source-address-translation
- `vlans` - (Optional) The virtual server is enabled/disabled on this set of VLANs. See vlans-disabled and vlans-enabled.
- `vlans_enabled` - (Optional Bool) Enables the virtual server on the VLANs specified by the VLANs option.
- `vlans_disabled` - (Optional Bool) Disables the virtual server on the VLANs specified by the VLANs option.
- `persistence_profiles` - (Optional) List of persistence profiles associated with the Virtual Server.
- `fallback_persistence_profile` - (Optional) Specifies a fallback persistence profile for the Virtual Server to use when the default persistence profile is not available.

» bigip_net_route

`bigip_net_route` Manages a route configuration

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_net_route" "route2" {
  name = "external-route"
  network = "10.10.10.0/24"
  gw      = "1.1.1.2"
}
```

» Argument Reference

- `name` - (Required) Name of the route
- `network` - (Optional) The destination subnet and netmask for the route.
- `network` - (Optional) Specifies a gateway address for the route.

» bigip_net_selfip

`bigip_net_selfip` Manages a selfip configuration

Resource should be named with their "full path". The full path is the combination of the partition + name of the resource, for example /Common/my-selfip.

» Example Usage

```
resource "bigip_net_selfip" "selfip1" {
  name          = "/Common/internalselfIP"
  ip            = "11.1.1.1/24"
  vlan          = "/Common/internal"
  traffic_group = "traffic-group-1"

  depends_on = ["bigip_net_vlan.vlan1"]
}
```

» Argument Reference

- `name` - (Required) Name of the selfip
- `ip` - (Required) The Self IP's address and netmask.
- `vlan` - (Required) Specifies the VLAN for which you are setting a self IP address. This setting must be provided when a self IP is created.

- `traffic_group` - (Optional) Specifies the traffic group, defaults to `traffic-group-local-only` if not specified.

» bigip_net_vlan

`bigip_net_vlan` Manages a vlan configuration

For resources should be named with their "full path". The full path is the combination of the partition + name of the resource. For example /Common/my-pool.

» Example Usage

```
resource "bigip_net_vlan" "vlan1" {
    name = "/Common/Internal"
    tag = 101
    interfaces = {
                vlanport = 1.2,
                    tagged = false
                }
  }
```

» Argument Reference

- `name` - (Required) Name of the vlan
- `tag` - (Optional) Specifies a number that the system adds into the header of any frame passing through the VLAN.
- `interfaces` - (Optional) Specifies which interfaces you want this VLAN to use for traffic management.
- `vlanport` - Physical or virtual port used for traffic
- `tagged` - Specifies a list of tagged interfaces or trunks associated with this VLAN. Note that you can associate tagged interfaces or trunks with any number of VLANs.

» bigip_sys_iapp

`bigip_sys_iapp` resource helps you to deploy Application Services template that can be used to automate and orchestrate Layer 4-7 applications service deployments using F5 Network.

» Example Usage

```
 resource "bigip_sys_iapp" "simplehttp" {
  name = "simplehttp"
  jsonfile = "${file("simplehttp.json")}"
}
```

» Argument Reference

- `name` - Name of the iApp.
- `jsonfile` - Refer to the Json file which will be deployed on F5 BIG-IP.

» Example Usage of Json file

```
{
"fullPath":"/Common/simplehttp.app/simplehttp",
"generation":222,
"inheritedDevicegroup":"true",
"inheritedTrafficGroup":"true",
"kind":"tm:sys:application:service:servicestate",
"name":"simplehttp",
"partition":"Common",
"selfLink":"https://localhost/mgmt/tm/sys/application/service/~Common~simplehttp.app~simpleh
"strictUpdates":"enabled",
"subPath":"simplehttp.app",
"tables":[
   {
      "name":"basic__snatpool_members"
   },
   {
      "name":"net__snatpool_members"
   },
   {
      "name":"optimizations__hosts"
   },
   {
      "columnNames":[
         "name"
      ],
      "name":"pool__hosts",
      "rows":[
         {
            "row":[
```

```
                "f5.cisco.com"
            ]
         }
      ]
   },
   {
      "columnNames":[
         "addr",
         "port",
         "connection_limit"
      ],
      "name":"pool__members",
      "rows":[
         {
            "row":[
               "10.0.2.167",
               "80",
               "0"
            ]
         },
         {
            "row":[
               "10.0.2.168",
               "80",
               "0"
            ]
         }
      ]
   },
   {
      "name":"server_pools__servers"
   }
],
"template":"/Common/f5.http",
"templateModified":"no",
"templateReference":{
   "link":"https://localhost/mgmt/tm/sys/application/template/~Common~f5.http?ver=13.0.0"
},
"trafficGroup":"/Common/traffic-group-1",
"trafficGroupReference":{
   "link":"https://localhost/mgmt/tm/cm/traffic-group/~Common~traffic-group-1?ver=13.0.0"
},
"variables":[
   {
      "encrypted":"no",
      "name":"client__http_compression",
```

```
   "value":"/#create_new#"
},
{
   "encrypted":"no",
   "name":"monitor__monitor",
   "value":"/Common/http"
},
{
   "encrypted":"no",
   "name":"net__client_mode",
   "value":"wan"
},
{
   "encrypted":"no",
   "name":"net__server_mode",
   "value":"lan"
},
{
   "encrypted":"no",
   "name":"net__v13_tcp",
   "value":"warn"
},
{
   "encrypted":"no",
   "name":"pool__addr",
   "value":"10.0.1.100"
},
{
   "encrypted":"no",
   "name":"pool__pool_to_use",
   "value":"/#create_new#"
},
{
   "encrypted":"no",
   "name":"pool__port",
   "value":"80"
},
{
   "encrypted":"no",
   "name":"ssl__mode",
   "value":"no_ssl"
},
{
   "encrypted":"no",
   "name":"ssl_encryption_questions__advanced",
   "value":"no"
```

```
    },
    {
       "encrypted":"no",
       "name":"ssl_encryption_questions__help",
       "value":"hide"
    }
]
}
```


- `description` - User defined description.
- `deviceGroup` - The name of the device group that the application service is assigned to.
- `executeAction` - Run the specified template action associated with the application.
- `inheritedDevicegroup`- Read-only. Shows whether the application folder will automatically remain with the same device-group as its parent folder. Use 'device-group default' or 'device-group non-default' to set this.
- `inheritedTrafficGroup` - Read-only. Shows whether the application folder will automatically remain with the same traffic-group as its parent folder. Use 'traffic-group default' or 'traffic-group non-default' to set this.
- `partition` - Displays the administrative partition within which the application resides.
- `strictUpdates` - Specifies whether configuration objects contained in the application may be directly modified, outside the context of the system's application management interfaces.
- `template` - The template defines the configuration for the application. This may be changed after the application has been created to move the application to a new template.
- `templateModified` - Indicates that the application template used to deploy the application has been modified. The application should be updated to make use of the latest changes.
- `templatePrerequisiteErrors` - Indicates any missing prerequisites associated with the template that defines this application.
- `trafficGroup` - The name of the traffic group that the application service is assigned to.
- `lists` - string values
- `metadata` - User defined generic data for the application service. It is a name and value pair.
- `tables` - Values provided like pool name, nodes etc.
- `variables` - Name, values, encrypted or not

» bigip_sys_ntp

`bigip_sys_ntp` provides details about a specific bigip

This resource is helpful when configuring NTP server on the BIG-IP.

» Example Usage

```
provider "bigip" {
  address = "10.192.74.73"
  username = "admin"
  password = "admin"
}


resource "bigip_sys_ntp" "ntp1" {

description = "/Common/NTP1"
  servers = ["time.facebook.com"]
  timezone = "America/Los_Angeles"
}
```

» Argument Reference

- `bigip_sys_ntp` - Is the resource is used to configure ntp server on the BIG-IP.
- `/Common/NTP1` - Is the description of the NTP server in the main or common partition of BIG-IP.
- `time.facebook.com` - Is the NTP server configured on the BIG-IP.
- `servers` - (Optional) Adds NTP servers to or deletes NTP servers from the BIG-IP system.
- `timezone` - (Optional) Specifies the time zone that you want to use for the system time.

» bigip_sys_provision

`bigip_sys_provision` provides details bout how to enable ”ilx”, ”asm” ”apm” resource on BIG-IP

» Example Usage

```
provider "bigip" {
  address = "10.192.74.73"
  username = "admin"
  password = "admin"
}

resource "bigip_sys_provision" "provision-ilx" {
  name = "/Common/ilx"
  fullPath  = "ilx"
  cpuRatio = 0
  diskRatio = 0
  level = "nominal"
  memoryRatio = 0
}
```

» Argument Reference

- `bigip_sys_provision` - Is the resource which is used to provision big-ip modules like asm, afm, ilx etc
- `Common/ilx` - Common is the partition and ilx is the module being enabled it could be asm, afm apm etc.
- `cpuRatio` - how much cpu resources you need for this resource
- `diskRatio` - how much disk space you want to allocate for this resource.
- `memoryRatio` - how much memory you want to deidcate for this resource

» bigip_sys_snmp

`bigip_sys_snmp` provides details bout how to enable ”ilx”, ”asm” ”apm” resource on BIG-IP

» Example Usage

```
resource "bigip_sys_snmp" "snmp" {
  sys_contact = " NetOPsAdmin s.shitole@f5.com"
  sys_location = "SeattleHQ"
  allowedaddresses = ["202.10.10.2"]
}
```

» Argument Reference

- `sys_contact` - (Optional) Specifies the contact information for the system administrator.
- `sys_location` - Describes the system's physical location.
- `allowedaddresses` - Configures hosts or networks from which snmpd can accept traffic. Entries go directly into hosts.allow.

» bigip_sys_snmp_traps

`bigip_sys_snmp_traps` provides details bout how to enable snmp_traps resource on BIG-IP

» Example Usage

```
resource "bigip_sys_snmp_traps" "snmp_traps" {
name = "snmptraps"
community = "f5community"
host = "195.10.10.1"
description = "Setup snmp traps"
port = 111
}
```

» Argument Reference

- `name` - (Optional) Name of the snmp trap.
- `community` - (Optional) Specifies the community string used for this trap.
- `host` - The host the trap will be sent to.
- `description` - (Optional) The port that the trap will be sent to.
- `port` - (Optional) User defined description.