Brightbox Provider

The Brightbox provider is used to interact with the resources supported by Brightbox Cloud. The provider needs to be configured with the proper credentials before it can be used.

Use the navigation to the left to read about the available resources.

Example Usage

```
# Configure the Brightbox Provider
provider "brightbox" {
  version = "~> 1.0"
  username = "${var.user_email_address}"
  password = "${var.user_secret_password}"
  account = "${var.account_to_work_on}"
}

# Create a web server
resource "brightbox_server" "web" {
  # ...
}
```

Authentication

The Brightbox provider offers a flexible means of providing credentials for authentication. The following methods are supported, in this order, and explained below:

- Username credentials
- Static credentials
- Username Environment variables
- Static Environment variables

Username credentials

Username credentials can be provided by adding a `username` and `password` in-line in the Brightbox provider block:

Usage:

```
provider "brightbox" {
  version = "~> 1.0"
  username = "someone@example.com"
  password = "secretpassword"
}
```

This will operate on the default account for the user. If you are the collaborator on more than one account, you can select a different account by adding an `account` argument.

```
provider "brightbox" {
  version = "~> 1.0"
  username = "someone@example.com"
  password = "secretpassword"
  account  = "acc-diffr"
}
```

Static credentials

Static credentials can be provided by adding an `apiclient` and `apisecret` in-line in the Brightbox provider block:

Usage:

```
provider "brightbox" {
  version = "~> 1.0"
  apiclient = "cli-testy"
  apisecret = "secretcode"
}
```

API clients will only work on the account they are generated for.

Username Environment variables

You can provide your username and password via the `BRIGHTBOX_USER_NAME` and `BRIGHTBOX_PASSWORD` environment variables. If required you can provide a non-default account with the `BRIGHTBOX_ACCOUNT` variable.

```
provider "brightbox" {
  version = "~> 1.0"
}
```

Usage:

```
$ export BRIGHTBOX_USER_NAME="someone@example.com"
$ export BRIGHTBOX_PASSWORD="secretpassword"
$ export BRIGHTBOX_ACCOUNT="acc-diffr"
$ terraform plan
```

Static Environment variables

You can provide your api client id and secret via the `BRIGHTBOX_CLIENT` and `BRIGHTBOX_CLIENT_SECRET` environment variables. This will operate on the account that issued the client id.

```
provider "brightbox" {
  version = "~> 1.0"
}
```

Usage:

```
$ export BRIGHTBOX_CLIENT="cli-testy"
$ export BRIGHTBOX_CLIENT_SECRET="secretcode"
$ terraform plan
```

Argument Reference

The following arguments are supported:

- `apiclient` - (optional) This is the Brightbox client id for an account. This can also be specified with the `BRIGHTBOX_CLIENT` shell environment variable.
- `apisecret` - (optional) This is the Brightbox client secret. This can also be specified with the `BRIGHTBOX_CLIENT_SECRET` shell environment variable.
- `username` - (optional) This is the Brightbox user logon. This can also be specified with the `BRIGHTBOX_USER_NAME` shell environment variable.
- `password` - (optional) This is the Brightbox user logon password. This can also be specified with the `BRIGHTBOX_PASSWORD` shell environment variable.
- `account` - (optional) This is the Brightbox account you wish to operate upon. This can also be specified with the `BRIGHTBOX_ACCOUNT` shell environment variable.
- `apiurl` - (Optional) Use this to override the default endpoint URL constructed for the region. It's typically used to connect to custom Brightbox endpoints.

NOTE: At least one of `username` or `apiclient` must be specified.

brightbox_database_type

Use this data source to get the ID of a Brightbox Database Type for use in other resources.

Example Usage

```
data "brightbox_database_type" "4gb" {
    name = "^SSD 4GB$"
}
```

Argument Reference

- `name` - (Optional) A regex string to apply to the Database Type list returned by Brightbox Cloud.
- `description` - (Optional) A regex string to apply to the Database Type list returned by Brightbox Cloud.

> **NOTE:** arguments form a conjunction. All arguments must match to select an image.

> **NOTE:** If more or less than a single match is returned by the search, Terraform will fail. Ensure that your search is specific enough to return a single image only, or use `most_recent` to choose the most recent one.

Attributes Reference

`id` is set to the ID of the found Database Type. In addition, the following attributes are exported:

- `disk_size` - The disk size of the database server for this type
- `ram` - The memory size of the database server for this type

brightbox_image

Use this data source to get the ID of a Brightbox Image for use in other resources.

Example Usage

```
data "brightbox_image" "ubuntu_lts" {
    name = "^ubuntu-xenial.*server$"
    arch = "x86_64"
    official = true
    most_recent = true
}
```

Argument Reference

- `most_recent` - (Optional) If more than one result is returned, use the most recent image based upon the `created_at` time.
- `name` - (Optional) A regex string to apply to the Image list returned by Brightbox Cloud.
- `description` - (Optional) A regex string to apply to the Image list returned by Brightbox Cloud.
- `source_type` - (Optional) Either `upload` or `snapshot`.
- `owner` - (Optional) The account id that owns the image. Matches exactly.
- `arch` - (Optional) The architecture of the image: either `x86_64` or `i686`.
- `public` - (Optional) Boolean to select a public image.
- `official` - (Optional) Boolean to select an official image.
- `compatibility_mode` - (Optional) Boolean to match the compatibility mode flag.
- `username` - (Optional) The username used to logon to the image. Matches exactly.
- `ancestor_id` - (Optional) The image id of the parent of the image you are looking for.
- `licence_name` - (Optional) The name of the licence for the image. Matches exactly.

NOTE: arguments form a conjunction. All arguments must match to select an image.

NOTE: If more or less than a single match is returned by the search, Terraform will fail. Ensure that your search is specific enough to return a single image only, or use `most_recent` to choose the most recent one.

Attributes Reference

`id` is set to the ID of the found Image. In addition, the following attributes are exported:

- `status` - The state the image is in. Usually `available`, `deprecated` or `deleted`.
- `created_at` - The time and date the image was created/registered (UTC)
- `locked` - true if image has been set as locked and can not be deleted
- `virtual_size` - The virtual size of the disk image "container" in MB
- `disk_size` - The actual size of the data within the Image in MB

brightbox_server_group

Use this data source to get the ID of a Brightbox Server Group for use in other resources.

Example Usage

```
data "brightbox_server_group" "defaul" {
    name = "^default$"
}
```

Argument Reference

- `name` - (Optional) A regex string to apply to the Server Group list returned by Brightbox Cloud.
- `description` - (Optional) A regex string to apply to the Server Group list returned by Brightbox Cloud.

> **NOTE:** arguments form a conjunction. All arguments must match to select an image.

> **NOTE:** If more or less than a single match is returned by the search, Terraform will fail. Ensure that your search is specific enough to return a single image only, or use `most_recent` to choose the most recent one.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Server

brightbox_cloudip

Provides a Brightbox CloudIP resource.

Example Usage

```
resource "brightbox_cloudip" "web-public" {
  target = "${brightbox_server.web.interface}"
  name = "web-1 public address"
}

resource "brightbox_server" "web" {
  image  = "img-testy"
  name   = "web-1"
  zone = "gb1a"
  type = "512mb.ssd"
  server_groups = [ "grp-testy" ]
}
```

Argument Reference

The following arguments are supported:

- `target` - (Required) The CloudIP mapping target. This is the interface from a server, or the id of a load balancer or cloud sql resource
- `name` - (Optional) a label to assign to the CloudIP
- `reverse_dns` - (Optional) The reverse DNS entry for the CloudIP

Attributes Reference

The following attributes are exported:

- `id` - The ID of the CloudIP
- `fqdn` - Fully Qualified Domain Name of the CloudIP
- `public_ip` - the public IPV4 address of the CloudIP
- `status` - Current state of the CloudIP: `mapped` or `unmapped`
- `username` - The username used to log onto the server

Import

CloudIPs can be imported using the `id`, e.g.

```
terraform import brightbox_cloudip.mycloudip cip-vsalc
```

brightbox_container

Provides a Brightbox Container resource. This can be used to create, modify, and delete Containers in Orbit.

Example Usage

```
# Example Container
resource "brightbox_container" "initial" {
  name = "initial"
  description = "Initial database snapshots"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) A label assigned to the Container
- `description` - (Optional) A further description of the Container
- `orbit_url` - (Optional) The Orbit URL you wish to talk to. This defaults to either `https://orbit.brightbox.com/v1/` or the contents of the `BRIGHTBOX_ORBIT_URL` environment variable if set.

Attributes Reference

The following attributes are exported:

- `auth_user` - the api client id used to access the container
- `auth_key` - the client secret used to access the container
- `account_id` - the account under which the container is stored

brightbox_database_server

Provides a Brightbox Database Server resource. This can be used to create, modify, and delete Database Servers.

Example Usage

```
resource "brightbox_database_server" "default" {
    name = "Default DB"
    description = "Default DB used by servers"
    database_engine = "mysql"
    database_version = "5.6"
    database_type = "${data.brightbox_database_type.4gb.id}"
    maintenance_weekday = 5
    maintenance_hour = 4
    allow_access = [
        "${brightbox_server_group.barfoo.id}",
        "${brightbox_server.foobar.id}",
        "158.152.1.65/32"
    ]
}

data "brightbox_database_type" "4gb" {
    name = "^SSD 4GB$"
}

resource "brightbox_server" "foobar" {
    name = "database access"
    image = "img-testy"
    server_groups = [ "${brightbox_server_group.barfoo.id}" ]
}

resource "brightbox_server_group" "barfoo" {
    name = "database access group"
}
```

Argument Reference

The following arguments are supported:

- `allow_access` (Required) - A list of server group ids, server ids or IPv4 address references the database server should be accessible from. There must be at least one entry in the list
- `name` - (Optional) A label assigned to the Database Server
- `description` - (Optional) A further description of the Database Server
- `maintenance_weekday` - (Optional) Numerical index of weekday (0 is Sunday, 1 is Monday...) to set when automatic updates may be performed. Default is 0 (Sunday).
- `maintenance_hour` - (Optional) Number representing 24hr time start of maintenance window hour for x:00-x:59 (0-23). Default is 6
- `database_engine` - (Optional) Database engine to request. Default is mysql.

- `database_version` - (Optional) Database version to request. Default is 5.5.
- `database_type` - (Optional) ID of the Database Type required.
- `snapshot` (Optional) - Database snapshot id to build from
- `zone` - (Optional) The handle of the zone required (`gb1-a`, `gb1-b`)

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Database Server
- `admin_username` - The username used to log onto the database
- `admin_password` - The password used to log onto the database
- `status` - Current state of the database server, usually `active` or `deleted`
- `locked` - True if database server has been set to locked and cannot be deleted

brightbox_firewall_policy

Provides a Brightbox Firewall Policy resource.

Example Usage

```
resource "brightbox_server_group" "default" {
  name = "Terraform"
}

resource "brightbox_firewall_policy" "default" {
  name         = "Terraform"
  server_group = "${brightbox_server_group.default.id}"
}
```

Argument Reference

The following arguments are supported:

- `server_group` - (Optional) The ID of the Server Group the policy will be applied to
- `name` - (Optional) A label to assign to the Firewall Policy
- `description` - (Optional) A further description of the Firewall Policy

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Firewall Policy

brightbox_firewall_rule

Provides a Brightbox Firewall Rule resource.

Example Usage

```
resource "brightbox_server_group" "default" {
  name = "Terraform"
}

resource "brightbox_firewall_policy" "default" {
  name         = "Terraform"
  server_group = "${brightbox_server_group.default.id}"
}

resource "brightbox_firewall_rule" "default_ssh" {
  destination_port = 22
  protocol         = "tcp"
  source           = "any"
  description      = "SSH access from anywhere"
  firewall_policy  = "${brightbox_firewall_policy.default.id}"
}
```

Argument Reference

The following arguments are supported:

- `firewall_policy` - (Required) The ID of the firewall policy this rule belongs to
- `protocol` - (Optional) Protocol Number or one of `tcp`, `udp`, `icmp`
- `source` - (Optional) Subnet, ServerGroup or ServerID. `any`,`10.1.1.23/32` or `srv-4ktk4`
- `source_port` - (Optional) single port, multiple ports or range separated by `-` or `:`; upto 255 characters. Example - `80`, `80,443,21` or `3000-3999`
- `destination` - (Optional) Subnet, ServerGroup or ServerID. `any`,`10.1.1.23/32` or `srv-4ktk4`
- `destination_port` - (Optional) single port, multiple ports or range separated by `-` or `:`; upto 255 characters. Example - `80`, `80,443,21` or `3000-3999`
- `icmp_type_name` - (Optional) ICMP type name. `echo-request`, `echo-reply`. Only allowed if protocol is `icmp`.
- `description` - (Optional) A further description of the Firewall Rule

NOTE: Only one of `source` or `destination` can be specified

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Firewall Rule

brightbox_load_balancer

Provides a Brightbox Load Balancer resource. This can be used to create, modify, and delete Load Balancers.

Example Usage

```
resource "brightbox_load_balancer" "lb" {
  name = "Terraform weblayer example"

  listener {
    protocol = "https"
    in       = 443
    out      = 8080
  }

  listener {
    protocol = "http"
    in       = 80
    out      = 8080
    timeout  = 10000
  }

  listener {
    protocol = "http+ws"
    in       = 81
    out      = 81
    timeout  = 10000
  }

  healthcheck {
    type = "http"
    port = 8080
  }

  nodes = [
    "${brightbox_server.server2.id}",
    "${brightbox_server.server1.id}",
  ]

  certificate_pem = <<EOF
-----BEGIN CERTIFICATE-----
MIIDBzCCAe+gAwIBAgIJAPD+BTBqIVp6MA0GCSqGSIb3DQEBBQUAMBoxGDAWBgNV
BAMMD3d3dy5leGFtcGxlLmNvbTAeFw0xNjAzMDIxMTU0MDFaFw0yNjAyMjgxMTU0
MDFaMBoxGDAWBgNVBAMMD3d3dy5leGFtcGxlLmNvbTCCASIwDQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBANuA/TLmuCbZdHcMKUwFadRpNnjg3S3PuP9AECDu+mIC
rOBmNqeZ66dEkzJqNMq4pEo30L9ZlZXl7fAvsIZTPYLEb0ieYGyTTdqAKrHi8GPP
ZeC+iAySKXnTKjpnciTWFv2T8R9tLsgPrsv54okM59bYC5mSnD7pL6RR/aQ0oi4f
X2eJex5fpfFlcxm9HvvVEdWq9/CQNoCOpGhLT911MRVMUl3S10BmzTG8Q87P76ji
Axt3t5piPg8JGiSBHTUJmKw/jxcwhybWHaf/217RmSmeoTo40wMCB2b05RqdSOm5
39qLotrjt2w3nFKzm423cVok3y2w55hLkDCbDlxUK1kCAwEAAaNQME4wHQYDVR0O
BBYEFCX20aoQddqjbga66nppwRlJdvB8MB8GA1UdIwQYMBaAFCX20aoQddqjbga6
6nppwRlJdvB8MAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADggEBAJkFZvAL
joeAiWaEItIPr8+98OJam7Pnta29HoKu4jAHkiunzXxNTQutUMMx1WhBF8OJJX1P
pHhKEfK47W8z4PbsM/hudZfm2xXlFMfvYNAusptJxOhMKNJgJz+gjY5FaTCGD9Ao
JkcshhUgXQ9zvu2Ol390qo0zlxMvnlVacRgKGY/I6hJaktrbdXm7qcReZp06Pw3a
adoKmzXeUlPvlbb+8KLXSD7hgUaojLDEgOLpAE++muiAAuwOP2UX3XJOPUQZdicB
sbrBMXO6F253YTqZiwAg9hgEHTHdXgqrd3TQT9P9mazrHxskqk9uWmIgN8oolHjp
OsWSdvMP2tRS8Oo=
-----END CERTIFICATE-----
EOF
```

```
  certificate_private_key = <<EOF
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA24D9Mua4Jtl0dwwpTAVp1Gk2eODdLc+4/0AQIO76YgKs4GY2
p5nrp0STMmo0yrikSjfQv1mVleXt8C+whlM9gsRvSJ5gbJNN2oAqseLwY89l4L6I
DJIpedMqOmdyJNYW/ZPxH20uyA+uy/niiQzn1tgLmZKcPukvpFH9pDSiLh9fZ4l7
Hl+l8WVzGb0e+9UR1ar38JA2gI6kaEtP3XUxFUxSXdLXQGbNMbxDzs/vqOIDG3e3
mmI+DwkaJIEdNQmYrD+PFzCHJtYdp//bXtGZKZ6hOjjTAwIHZvTlGp1I6bnf2oui
2uO3bDecUrObjbdxWiTfLbDnmEuQMJsOXFQrWQIDAQABAoIBAHzvoC42sB48q1OP
Mno4opHqCL0oj/uhPdTa69My8oSSrT9ULkubCkw8deO+G6o/ChPMTR58qO2W36VU
H491FY+2qviUXKGv/iIdzS9O0jCdPYl8KQeusbjLfj+b3ZYl3RQb/qQ6iuQIOR+U
bWJAXD0m3wNcNV6Bb0KCAHJUGvNQjiueMMVEND1Pvb9WogFWY7yvteoxv9ASFiRv
1N2LDlm/199/Tpmb9a9vVrIuT8pZfAtmVfZ5HhwV8xU1q2qbys1j9DpZPggHnT4l
CzIw7pALbaE8/sG17h6+icl13cKLpgp63HyJFgik1v1NDnCmzckrNAiSW4lZsgzM
BV3m9hkCgYEA7qboVDv6FvwwwyILbd3aYjLjCqNjDzpvngJrOl6/cDDQR34NQPzI
3ePYO1p99xRYmQe0FJ7ZuJtOQHJOdeLEJqeo6lNMI9T+FhKnqk7Gy7ZQI0PNP2x6
tpfoa27emeDblu+AVSBIZjByS+Cpf/Mnf4/DhhofAMdT4TFyng/JbbMCgYEA63XA
tHE8BwxY/6NxR/pGlRi0AbZfjfU40/q+309NNGrGyDZfoYpbG9I6Wo09Rc+QDhEq
2+zk59ubO1jkgh9eIOBm6+5yHjcbwftBsxesQQdabAg34ppFYMfvKsLgDnejvYEW
pfLmMAvcmIFGWid9hX5/ShbjjkJnIKSbu/vN9MMCgYEAoFhyZw45NTJSjPkV1sal
0S7Bj0dB6lxn3DFh3EEGvRl/B1nxC8YMK9HHWfGuCtGXyZH8c5JbVIa8p95lSx2G
jzY87tvyn2yfHzN/hZUSSpL++wK2J7P6Ky6bkXtXguoqgBoBDrD3E/nfAY48NGSq
GDH+u95XEE3c1MRFb1/KBbMCgYEAo2VgqBdYR6/a5vPd/cwBRSASconDf7inifsc
j8zxT6m1bmTFMk3X8dOOqR4QYiyq1Ag3zMx1AS0VaTbDxETORlRTN/CNgshNW+zn
Z8fKwom+xu9hEMBr2sCECRGY+JEvsKcvN1P7R2ZD3BUB5Dg5U/U3kguWODd+Z1mz
tN0FzI8CgYBx9giIe7aAItxl43p6tPsMW6ROlXEjWit2XBlaDdY5t48k8KJ2clk/
IHu8B12R2mN+lMn9mkOa4mSb9MrVQZ2FGg4lUAQro519NVBcVqoRsEDn1kHd+hhl
L6c41r4AZ3Iyvr3MYoSohogBbAnd6TW14NjvBHceREhAqvmIWlWmAQ==
-----END RSA PRIVATE KEY-----
EOF
}
```

Argument Reference

The following arguments are supported:

- `name` - (Optional) A label assigned to the Load Balancer
- `policy` - (Optional) Method of load balancing to use, either `least-connections` or `round-robin`
- `certificate_pem` - (Optional) A X509 SSL certificate in PEM format. Must be included along with `certificate_key`. If intermediate certificates are required they should be concatenated after the main certificate
- `certificate_private_key` - (Optional) The RSA private key used to sign the certificate in PEM format. Must be included along with `certificate_pem`
- `sslv3` - (Optional) Allow SSL v3 to be used. Default is `false`
- `buffer_size` - (Optional) Buffer size in bytes
- `nodes` - (Optional) An array of Server IDs
- `listener` - (Required) An array of listener blocks. The Listener block is described below
- `healthcheck` - (Required) A healthcheck block. The Healthcheck block is described below

Listener (`listener`) supports the following: * `protocol` - (Required) Protocol of the listener. One of `tcp`, `http`, `https`, `http+ws`, `https+wss` * `in` - (Required) Port to listen on * `out` - (Required) Port to pass through to * `timeout` - (Optional) Timeout of connection in milliseconds. Default is 50000

Health Check (`healthcheck`) supports the following: * `type` - (Required) Type of health check required: `tcp` or `http` * `port` - (Required) Port to connect to to check health * `request` - (Optional) Path used for HTTP check * `interval` - (Optional) Frequency of checks in milliseconds * `timeout` - (Optional) Timeout of health check in milliseconds * `threshold_up` - (Optional) Number of checks that must pass before connection is considered healthy * `threshold_down` - (Optional) Number of checks that must fail before connection is considered unhealthy

Attributes Reference

The following attributes are exported

- `id` - The ID of the Load Balancer
- `status` - Current state of the load balancer. Usually `creating` or `active`
- `locked` - True if the database server has been set to locked and cannot be deleted

brightbox_server

Provides a Brightbox Server resource. This can be used to create, modify, and delete Servers. Servers also support provisioning (/docs/provisioners/index.html).

Example Usage

```
# Create a new 512Mb SSD Web Server in the gb1a zone
resource "brightbox_server" "web" {
  image  = "img-testy"
  name   = "web-1"
  zone = "gb1a"
  type = "512mb.ssd"
  server_groups = [ "grp-testy" ]
}
```

Argument Reference

The following arguments are supported:

- `image` - (Required) The Server image ID
- `server_groups` (Required) - An array of server group ids the server should be added to. At least one server group must be specified.
- `name` - (Optional) The Server name
- `type` - (Optional) The handle of the server type required (`1gb.ssd`, etc)
- `zone` - (Optional) The handle of the zone required (`gb1-a`, `gb1-b`)
- `user_data` (Optional) - A string of the desired User Data for the Server.
- `user_data_base64` (Optional) - Already encrypted User Data - for use with the template provider.

NOTE: Only one of `user_data` or `user_data_base64` can be specified

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Server
- `fqdn` - Fully Qualified Domain Name of server
- `hostname` - short name of server, usually the same as the `id`
- `interface` - the id reference of the network interface. Used to target cloudips.

- `ipv4_address_private` - The RFC 1912 address of the server
- `ipv6_address` - the IPv6 address of the server
- `ipv6_hostname` - the FQDN of the IPv6 address
- `public_hostname` - the FQDN of the public IPv4 address. Appears if a cloud ip is mapped
- `ipv4_address` - the public IPV4 address of the server. Appears if a cloud ip is mapped
- `locked` - True if server has been set to locked and cannot be deleted
- `status` - Current state of the server, usually `active`, `inactive` or `deleted`
- `username` - The username used to log onto the server

Import

Servers can be imported using the server `id`, e.g.

```
terraform import brightbox_server.myserver srv-ojy3o
```

brightbox_server_group

Provides a Brightbox Server Group resource. This can be used to create, modify, and delete Server Groups.

Example Usage

```
# Default Server Group
# the instances over SSH and HTTP
resource "brightbox_server_group" "default" {
  name = "Terraform controlled servers"
}

# Create a new 512Mb SSD Web Server in the gb1a zone
resource "brightbox_server" "web" {
  image  = "img-testy"
  name   = "web-1"
  zone = "gb1a"
  type = "512mb.ssd"
  server_groups = ["${brightbox_server_group.default.id}"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Optional) A label assigned to the Server Group
- `description` - (Optional) A further description of the Server Group

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Server

Import

Server Groups can be imported using the server group `id`, e.g.

```
terraform import brightbox_server_group.default grp-ok8vw
```