ProfitBricks Provider

The ProfitBricks provider gives the ability to deploy and configure resources using the ProfitBricks Cloud API.

Use the navigation to the left to read about the available data sources and resources.

Usage

The provider needs to be configured with proper credentials before it can be used.

```
$ export PROFITBRICKS_USERNAME="profitbricks_username"
$ export PROFITBRICKS_PASSWORD="profitbricks_password"
$ export PROFITBRICKS_API_URL="profitbricks_cloud_api_url"
```

Or you can provide your credentials in a `.tf` configuration file as shown in this example.

Example Usage

```
provider "profitbricks" {
  username = "profitbricks_username"
  password = "profitbricks_password"
  endpoint = "profitbricks_cloud_api_url"
}

resource "profitbricks_datacenter" "main" {
  # ...
}
```

Note: The credentials provided in a `.tf` file will override the credentials from environment variables.

Configuration Reference

The following arguments are supported:

- `username` - (Required) If omitted, the `PROFITBRICKS_USERNAME` environment variable is used. The username is generally an e-mail address in 'username@domain.tld (mailto:username@domain.tld)' format.
- `password` - (Required) If omitted, the `PROFITBRICKS_PASSWORD` environment variable is used.
- `endpoint` - (Optional) If omitted, the `PROFITBRICKS_API_URL` environment variable is used, or it defaults to the current Cloud API release.
- `retries` - (Deprecated) Number of retries while waiting for a resource to be provisioned. Default value is 50. **Note**: This argument has been deprecated and replaced by the implementation of resource timeouts described below.

Resource Timeout

Individual resources may provide a `timeouts` block to configure the amount of time a specific operation is allowed to take before being considered an error. Each resource may provide configurable timeouts for the `create`, `update`, and `delete` operations. Each resource that supports timeouts will have or inherit default values for that operation. Users can overwrite the default values for a specific resource in the configuration.

The default `timeouts` values are:

- create - (Default 60 minutes) Used for creating a resource.
- update - (Default 60 minutes) Used for updating a resource .
- delete - (Default 60 minutes) Used for destroying a resource.
- default - (Default 60 minutes) Used for every other action on a resource.

An example of overwriting the `create`, `update`, and `delete` timeouts:

```
resource "profitbricks_server" "example" {
  name              = "server"
  datacenter_id     = "${profitbricks_datacenter.example.id}"
  cores             = 1
  ram               = 1024
  availability_zone = "ZONE_1"
  cpu_family        = "AMD_OPTERON"

  volume {
    name           = "new"
    image_name     = "${var.ubuntu}"
    size           = 5
    disk_type      = "SSD"
    ssh_key_path   = "${var.private_key_path}"
    image_password = "test1234"
  }

  nic {
    lan             = "${profitbricks_lan.example.id}"
    dhcp            = true
    ip              = "${profitbricks_ipblock.example.ip}"
    firewall_active = true

    firewall {
      protocol         = "TCP"
      name             = "SSH"
      port_range_start = 22
      port_range_end   = 22
    }
  }

  timeouts {
    create = "30m"
    update = "300s"
    delete = "2h"
  }
}
```

Valid units of time should be expressed in "s", "m", "h" for "seconds", "minutes", and "hours" respectively.

Individual resources must opt-in to providing configurable `timeouts`, and attempting to configure values for a resource that does not support `timeouts`, or overwriting a specific action that the resource does not specify as an option, will result in an error.

Note: Terraform does not automatically rollback in the face of errors. Instead, your Terraform state file will be partially updated with any resources that successfully completed.

Support

You are welcome to contact us with questions or comments at ProfitBricks DevOps Central (https://devops.profitbricks.com/).

profitbricks_datacenter

The data centers data source can be used to search for and return an existing Virtual Data Center. You can provide a string for the name and location parameters which will be compared with provisioned Virtual Data Centers. If a single match is found, it will be returned. If your search results in multiple matches, an error will be generated. When this happens, please refine your search string so that it is specific enough to return only one result.

Example Usage

```
data "profitbricks_datacenter" "dc_example" {
  name     = "test_dc"
  location = "us"
}
```

Argument Reference

- `name` - (Required) Name or part of the name of an existing Virtual Data Center that you want to search for.
- `location` - (Optional) Id of the existing Virtual Data Center's location.

Attributes Reference

- `id` - UUID of the Virtual Data Center

profitbricks_image

The images data source can be used to search for and return an existing image which can then be used to provision a server.

Example Usage

```
data "profitbricks_image" "image_example" {
  name     = "Ubuntu"
  type     = "HDD"
  version  = "14"
  location = "location_id"
}
```

Argument Reference

- `name` - (Required) Name or part of the name of an existing image that you want to search for.
- `version` - (Optional) Version of the image (see details below).
- `location` - (Optional) Id of the existing image's location.
- `type` - (Optional) The image type, HDD or CD-ROM.

If both "name" and "version" are provided the plugin will concatenate the two strings in this format [name]-[version].

Attributes Reference

- `id` - UUID of the image

profitbricks_location

The locations data source can be used to search for and return an existing location which can then be used elsewhere in the configuration.

Example Usage

```
data "profitbricks_location" "loc1" {
  name    = "karlsruhe"
  feature = "SSD"
}
```

Argument Reference

- `name` - (Required) Name or part of the location name to search for.
- `feature` - (Optional) A desired feature that the location must be able to provide.

Attributes Reference

- `id` - UUID of the location

profitbricks_resource

The resource data source can be used to search for and return any existing ProfitBricks resource and optionally their group associations. You can provide a string for the resource type (datacenter,image,snapshot,ipblock) and/or resource id parameters which will be queries against available resources. If a single match is found, it will be returned. If your search results in multiple matches, an error will be generated. When this happens, please refine your search string so that it is specific enough to return only one result.

Example Usage

```
data "profitbricks_resource" "res" {
  resource_type = "datacenter"
  resource_id="datacenter uuid"
}
```

Argument Reference

- `resource_type` - (Optional) The specific type of resources to retrieve information about.
- `resource_id` - (Optional) The ID of the specific resource to retrieve information about.

Attributes Reference

- `id` - UUID of the Resource

profitbricks_snapshot

The snapshots data source can be used to search for and return an existing snapshot which can then be used to provision a server.

Example Usage

```
data "profitbricks_snapshot" "snapshot_example" {
  name     = "my snapshot"
  size     = "2"
  location = "location_id"
}
```

Argument Reference

- `name` - (Required) Name or part of the name of an existing snapshot that you want to search for.
- `location` - (Optional) Id of the existing snapshot's location.
- `size` - (Optional) The size of the snapshot to look for.

Attributes Reference

- `id` - UUID of the snapshot

profitbricks_datacenter

Manages a Virtual Data Center on ProfitBricks.

Example Usage

```
resource "profitbricks_datacenter" "example" {
  name        = "datacenter name"
  location    = "us/las"
  description = "datacenter description"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required)[string] The name of the Virtual Data Center.
- `location` - (Required)[string] The regional location where the Virtual Data Center will be created.
- `description` - (Optional)[string] Description for the Virtual Data Center.

Import

Resource Datacenter can be imported using the `resource id`, e.g.

```
terraform import profitbricks_datacenter.mydc {datacenter uuid}
```

profitbricks_firewall

Manages a set of firewall rules on ProfitBricks.

Example Usage

```
resource "profitbricks_firewall" "example" {
  datacenter_id    = "${profitbricks_datacenter.example.id}"
  server_id        = "${profitbricks_server.example.id}"
  nic_id           = "${profitbricks_server.example.primary_nic}"
  protocol         = "TCP"
  name             = "test"
  port_range_start = 1
  port_range_end   = 2
}
```

Argument reference

- `datacenter_id` - (Required)[string] The Virtual Data Center ID.
- `server_id` - (Required)[string] The Server ID.
- `nic_id` - (Required)[string] The NIC ID.
- `protocol` - (Required)[string] The protocol for the rule: TCP, UDP, ICMP, ANY.
- `name` - (Optional)[string] The name of the firewall rule.
- `source_mac` - (Optional)[string] Only traffic originating from the respective MAC address is allowed. Valid format: aa:bb:cc:dd:ee:ff.
- `source_ip` - (Optional)[string] Only traffic originating from the respective IPv4 address is allowed.
- `target_ip` - (Optional)[string] Only traffic directed to the respective IP address of the NIC is allowed.
- `port_range_start` - (Optional)[string] Defines the start range of the allowed port (from 1 to 65534) if protocol TCP or UDP is chosen.
- `port_range_end` - (Optional)[string] Defines the end range of the allowed port (from 1 to 65534) if the protocol TCP or UDP is chosen.
- `icmp_type` - (Optional)[string] Defines the allowed type (from 0 to 254) if the protocol ICMP is chosen.
- `icmp_code` - (Optional)[string] Defines the allowed code (from 0 to 254) if protocol ICMP is chosen.

Import

Resource Firewall can be imported using the `resource id`, e.g.

```
terraform import profitbricks_firewall.myfwrule {datacenter uuid}/{server uuid}/{nic uuid}/{firewall uuid
}
```

profitbricks_group

Manages groups and group privileges on ProfitBricks.

Example Usage

```
resource "profitbricks_group" "group" {
  name = "my group"
  create_datacenter = true
  create_snapshot = true
  reserve_ip = true
  access_activity_log = false
  user_id="user_id"
}
```

Argument reference

- `access_activity_log` - (Required) [Boolean] The group will be allowed to access the activity log.
- `create_datacenter` - (Optional) [Boolean] The group will be allowed to create virtual data centers.
- `create_snapshot` - (Optional) [Boolean] The group will be allowed to create snapshots.
- `name` - (Optional) [string] A name for the group.
- `reserve_ip` - (Optional) [Boolean] The group will be allowed to reserve IP addresses.
- `user_id` - (Optional) [string] The ID of the specific user to add to the group.

profitbricks_ipblock

Manages IP Blocks on ProfitBricks. IP Blocks contain reserved public IP addresses that can be assigned servers or other resources.

Example Usage

```
resource "profitbricks_ipblock" "example" {
  location = "${profitbricks_datacenter.example.location}"
  size     = 1
}
```

Argument reference

- `location` - (Required)[string] The regional location for this IP Block: us/las, us/ewr, de/fra, de/fkb.
- `size` - (Required)[integer] The number of IP addresses to reserve for this block.
- `ips` - (Computed)[integer] The list of IP addresses associated with this block.

Import

Resource Ipblock can be imported using the `resource id`, e.g.

```
terraform import profitbricks_ipblock.myipblock {ipblock uuid}
```

profitbricks_ipfailover

Manages IP Failover groups on ProfitBricks.

Example Usage

```
resource "profitbricks_ipfailover" "failovertest" {
  datacenter_id = "datacenterId"
  lan_id="lanId"
  ip ="reserved IP"
  nicuuid= "nicId"
}
```

Argument reference

- `datacenter_id` - (Required)[string] The ID of a Virtual Data Center.
- `ip` - (Required)[string] The reserved IP address to be used in the IP failover group.
- `lan_id` - (Required)[string] The ID of a LAN.
- `nicuuid` - (Required)[string] The ID of a NIC.

profitbricks_lan

Manages a LAN on ProfitBricks.

Example Usage

```
resource "profitbricks_lan" "example" {
  datacenter_id = "${profitbricks_datacenter.example.id}"
  public        = true
}
```

Argument reference

- `datacenter_id` - (Required)[string] The ID of a Virtual Data Center.
- `name` - (Optional)[string] The name of the LAN.
- `public` - (Optional)[Boolean] Indicates if the LAN faces the public Internet (true) or not (false).

Import

Resource Lan can be imported using the `resource id`, e.g.

```
terraform import profitbricks_lan.mylan {datacenter uuid}/{lan id}
```

profitbricks_loadbalancer

Manages a Load Balancer on ProfitBricks.

Example Usage

```
resource "profitbricks_loadbalancer" "example" {
  datacenter_id = "${profitbricks_datacenter.example.id}"
  nic_ids        = ["${profitbricks_nic.example.id}"]
  name          = "load balancer name"
  dhcp          = true
}
```

Argument reference

- `name` - (Required)[string] The name of the load balancer.
- `datacenter_id` - (Required)[string] The ID of a Virtual Data Center.
- `nic_ids` - (Required)[list] A list of NIC IDs that are part of the load balancer.
- `dhcp` - (Optional)[Boolean] Indicates if the load balancer will reserve an IP using DHCP.
- `ip` - (Optional)[string] IPv4 address of the load balancer.

profitbricks_nic

Manages a NIC on ProfitBricks.

Example Usage

```
resource "profitbricks_nic" "example" {
  datacenter_id = "${profitbricks_datacenter.example.id}"
  server_id     = "${profitbricks_server.example.id}"
  lan           = 2
  dhcp          = true
  ip            = "${profitbricks_ipblock.example.ip}"
}
```

Argument reference

- `datacenter_id` - (Required)[string] The ID of a Virtual Data Center.
- `server_id` - (Required)[string] The ID of a server.
- `lan` - (Required)[integer] The LAN ID the NIC will sit on.
- `name` - (Optional)[string] The name of the LAN.
- `dhcp` - (Optional)[Boolean] Indicates if the NIC should get an IP address using DHCP (true) or not (false).
- `ip` - (Optional)[string] IP assigned to the NIC.
- `firewall_active` - (Optional)[Boolean] If this resource is set to true and is nested under a server resource firewall, with open SSH port, resource must be nested under the NIC.
- `nat` - (Optional)[Boolean] Boolean value indicating if the private IP address has outbound access to the public internet.
- `ips` - (Computed) The IP address or addresses assigned to the NIC.

Import

Resource Nic can be imported using the `resource id`, e.g.

```
terraform import profitbricks_nic.mynic {datacenter uuid}/{server uuid}/{nic uuid}
```

profitbricks_server

Manages a Server on ProfitBricks.

Example Usage

This resource will create an operational server. After this section completes, the provisioner can be called.

```
resource "profitbricks_server" "example" {
  name              = "server"
  datacenter_id     = "${profitbricks_datacenter.example.id}"
  cores             = 1
  ram               = 1024
  availability_zone = "ZONE_1"
  cpu_family        = "AMD_OPTERON"
  image_password    = "test1234"
  ssh_key_path      = "${var.private_key_path}"
  boot_image        = "${var.ubuntu}"

  volume {
    name           = "new"
    size           = 5
    disk_type      = "SSD"
  }

  nic {
    lan             = "${profitbricks_lan.example.id}"
    dhcp            = true
    ip              = "${profitbricks_ipblock.example.ip}"
    firewall_active = true
  }
}
```

Argument reference

- `name` - (Required)[string] The name of the server.
- `datacenter_id` - (Required)[string] The ID of a Virtual Data Center.
- `cores` - (Required)[integer] Number of server CPU cores.
- `ram` - (Required)[integer] The amount of memory for the server in MB.
- `availability_zone` - (Optional)[string] The availability zone in which the server should exist.
- `licence_type` - (Optional)[string] Sets the OS type of the server.
- `cpu_family` - (Optional)[string] Sets the CPU type. "AMD_OPTERON" or "INTEL_XEON". Defaults to "AMD_OPTERON".
- `volume` - (Required) See the Volume section.
- `nic` - (Required) See the NIC section.

- `boot_volume` - (Computed) The associated boot volume.
- `boot_cdrom` - (Computed) The associated boot drive, if any.
- `boot_image` - [string] The image or snapshot UUID. May also be an image alias. It is required if `licence_type` is not provided.
- `primary_nic` - (Computed) The associated NIC.
- `primary_ip` - (Computed) The associated IP address.
- `image_password` - (Computed) The associated IP address.
- `ssh_key_path` - (Required)[list] List of paths to files containing a public SSH key that will be injected into ProfitBricks provided Linux images. Required for ProfitBricks Linux images. Required if `image_password` is not provided.
- `image_password` - [string] Required if `sshkey_path` is not provided.

Import

Resource Server can be imported using the `resource id`, e.g.

```
terraform import profitbricks_server.myserver {datacenter uuid}/{server uuid}/{primary_nic uuid}
# or
terraform import profitbricks_server.myserver {datacenter uuid}/{server uuid}/{primary_nic uuid}/{firewal
l uuid}
```

profitbricks_share

Manages shares and list shares permissions granted to the group members for each shared resource.

Example Usage

```
resource "profitbricks_share" "share" {
  group_id = "groupId"
  resource_id = "resourceId"
  edit_privilege = true
  share_privilege = false
}
```

Argument reference

- `edit_privilege` - (Required)[Boolean] The group has permission to edit privileges on this resource.
- `group_id` - (Required)[string] The ID of the specific group containing the resource to update.
- `resource_id` - (Required)[string] The ID of the specific resource to update.
- `share_privilege` - (Required)[Boolean] The group has permission to share this resource.

profitbricks_snapshot

Manages snapshots on ProfitBricks.

Example Usage

```
resource "profitbricks_snapshot" "test_snapshot" {
  datacenter_id = "datacenterId"
  volume_id = "volumeId"
  name = "my snapshot"
}
```

Argument reference

- `datacenter_id` - (Required)[string] The ID of the Virtual Data Center.
- `name` - (Required)[string] The name of the snapshot.
- `volume_id` - (Required)[string] The ID of the specific volume to take the snapshot from.

profitbricks_user

Manages users and list users and groups associated with that user.

Example Usage

```
resource "profitbricks_user" "user" {
  first_name = "terraform"
  last_name = "test"
  email = "%s"
  password = "abc123-321CBA"
  administrator = false
  force_sec_auth= false
}
```

Argument reference

- `administrator` - (Required)[Boolean] The group has permission to edit privileges on this resource.
- `email` - (Required)[string] An e-mail address for the user.
- `first_name` - (Required)[string] A first name for the user.
- `force_sec_auth` - (Required)[Boolean] Indicates if secure (two-factor) authentication should be enabled for the user (true) or not (false).
- `last_name` - (Required)[string] A last name for the user.
- `password` - (Required)[string] A password for the user.

profitbricks_volume

Manages a volume on ProfitBricks.

Example Usage

A primary volume will be created with the server. If there is a need for additional volumes, this resource handles it.

```
resource "profitbricks_volume" "example" {
  datacenter_id = "${profitbricks_datacenter.example.id}"
  server_id     = "${profitbricks_server.example.id}"
  image_name    = "${var.ubuntu}"
  size          = 5
  disk_type     = "HDD"
  ssh_key_path  = "${var.private_key_path}"
  bus           = "VIRTIO"
}
```

Argument reference

- `datacenter_id` - (Required)[string] The ID of a Virtual Data Center.
- `server_id` - (Required)[string] The ID of a server.
- `disk_type` - (Required)[string] The volume type: HDD or SSD.
- `bus` - (Required)[Boolean] The bus type of the volume: VIRTIO or IDE.
- `size` - (Required)[integer] The size of the volume in GB.
- `ssh_key_path` - (Required)[list] List of paths to files containing a public SSH key that will be injected into ProfitBricks provided Linux images. Required for ProfitBricks Linux images. Required if `image_password` is not provided.
- `sshkey` - (Computed) The associated public SSH key.
- `image_password` - [string] Required if `sshkey_path` is not provided.
- `image_name` - [string] The image or snapshot UUID. May also be an image alias. It is required if `licence_type` is not provided.
- `licence_type` - [string] Required if `image_name` is not provided.
- `name` - (Optional)[string] The name of the volume.
- `availability_zone` - (Optional)[string] The storage availability zone assigned to the volume: AUTO, ZONE_1, ZONE_2, or ZONE_3.