» digitalocean_image

Get information on an snapshot images. The aim of this datasource is to enable you to build droplets based on snapshot names.

An error is triggered if zero or more than one result is returned by the query.

» Example Usage

Get the data about a snapshot:

```
data "digitalocean_image" "example1" {
  name = "example-1.0.0"
}
```

Reuse the data about a snapshot to create a droplet:

```
data "digitalocean_image" "example1" {
  name = "example-1.0.0"
}
resource "digitalocean_droplet" "example1" {
  image  = "${data.digitalocean_image.example1.image}"
  name   = "example-1"
  region = "nyc2"
  size   = "512mb"
}
```

» Argument Reference

The following arguments are supported:

- `name` - The name of the image.

» Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.
- `image` - The id of the image.
- `min_disk_size`: The minimum 'disk' required for the image.
- `private` - Is image a public image or not. Public images represents Linux distributions or Application, while non-public images represent snapshots and backups and are only available within your account.
- `regions`: The regions that the image is available in.
- `size_gigabytes`: The size of the image in gigabytes.
- `type`: Type of the image. Can be "snapshot" or "backup".

» digitalocean_certificate

Provides a DigitalOcean Certificate resource that allows you to manage certificates for configuring TLS termination in Load Balancers. Certificates created with this resource can be referenced in your Load Balancer configuration via their ID.

» Example Usage

```
# Create a new TLS certificate
resource "digitalocean_certificate" "cert" {
  name              = "Terraform Example"
  private_key       = "${file("/Users/terraform/certs/privkey.pem")}"
  leaf_certificate  = "${file("/Users/terraform/certs/cert.pem")}"
  certificate_chain = "${file("/Users/terraform/certs/fullchain.pem")}"
}

# Create a new Load Balancer with TLS termination
resource "digitalocean_loadbalancer" "public" {
  name        = "secure-loadbalancer-1"
  region      = "nyc3"
  droplet_tag = "backend"

  forwarding_rule {
    entry_port      = 443
    entry_protocol  = "https"

    target_port     = 80
    target_protocol = "http"

    certificate_id  = "${digitalocean_certificate.cert.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the certificate for identification.
- `private_key` - (Required) The contents of a PEM-formatted private-key corresponding to the SSL certificate.
- `leaf_certificate` - (Required) The contents of a PEM-formatted public TLS certificate.

- `certificate_chain` - (Optional) The full PEM-formatted trust chain between the certificate authority's certificate and your domain's TLS certificate.

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID of the certificate
- `name` - The name of the certificate
- `not_after` - The expiration date of the certificate
- `sha1_fingerprint` - The SHA-1 fingerprint of the certificate

» digitalocean_domain

Provides a DigitalOcean domain resource.

» Example Usage

```
# Create a new domain
resource "digitalocean_domain" "default" {
  name       = "www.example.com"
  ip_address = "${digitalocean_droplet.foo.ipv4_address}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the domain
- `ip_address` - (Required) The IP address of the domain. This IP is used to created an initial A record for the domain. It is required upstream by the DigitalOcean API.

» Attributes Reference

The following attributes are exported:

- `id` - The name of the domain

» Import

Domains can be imported using the `domain name`, e.g.

```
terraform import digitalocean_domain.mydomain mytestdomain.com
```

» digitalocean_droplet

Provides a DigitalOcean Droplet resource. This can be used to create, modify, and delete Droplets. Droplets also support provisioning.

» Example Usage

```
# Create a new Web Droplet in the nyc2 region
resource "digitalocean_droplet" "web" {
  image  = "ubuntu-14-04-x64"
  name   = "web-1"
  region = "nyc2"
  size   = "512mb"
}
```

» Argument Reference

The following arguments are supported:

- `image` - (Required) The Droplet image ID or slug.
- `name` - (Required) The Droplet name
- `region` - (Required) The region to start in
- `size` - (Required) The instance size to start
- `backups` - (Optional) Boolean controlling if backups are made. Defaults to false.
- `monitoring` - (Optional) Boolean controlling whether monitoring agent is installed. Defaults to false.
- `ipv6` - (Optional) Boolean controlling if IPv6 is enabled. Defaults to false.
- `private_networking` - (Optional) Boolean controlling if private networks are enabled. Defaults to false.
- `ssh_keys` - (Optional) A list of SSH IDs or fingerprints to enable in the format `[12345, 123456]`. To retrieve this info, use a tool such as `curl` with the DigitalOcean API, to retrieve them.
- `resize_disk` - (Optional) Boolean controlling whether to increase the disk size when resizing a Droplet. It defaults to `true`. When set to `false`, only the Droplet's RAM and CPU will be resized. **Increasing a Droplet's**

disk size is a permanent change. Increasing only RAM and CPU is reversible.
- `tags` - (Optional) A list of the tags to label this droplet. A tag resource must exist before it can be associated with a droplet.
- `user_data` (Optional) - A string of the desired User Data for the Droplet.
- `volume_ids` (Optional) - A list of the IDs of each block storage volume to be attached to the Droplet.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Droplet
- `name`- The name of the Droplet
- `region` - The region of the Droplet
- `image` - The image of the Droplet
- `ipv6` - Is IPv6 enabled
- `ipv6_address` - The IPv6 address
- `ipv6_address_private` - The private networking IPv6 address
- `ipv4_address` - The IPv4 address
- `ipv4_address_private` - The private networking IPv4 address
- `locked` - Is the Droplet locked
- `private_networking` - Is private networking enabled
- `price_hourly` - Droplet hourly price
- `price_monthly` - Droplet monthly price
- `size` - The instance size
- `disk` - The size of the instance's disk in GB
- `vcpus` - The number of the instance's virtual CPUs
- `status` - The status of the droplet
- `tags` - The tags associated with the droplet
- `volume_ids` - A list of the attached block storage volumes

» Import

Droplets can be imported using the droplet `id`, e.g.

```
terraform import digitalocean_droplet.mydroplet 100823
```

» digitalocean_firewall

Provides a DigitalOcean Cloud Firewall resource. This can be used to create, modify, and delete Firewalls.

» Example Usage

```
resource "digitalocean_droplet" "web" {
  name      = "web-1"
  size      = "512mb"
  image     = "centos-7-x64"
  region    = "nyc3"
}

resource "digitalocean_firewall" "web" {
  name = "only-22-80-and-443"

  droplet_ids = ["${digitalocean_droplet.web.id}"]

  inbound_rule = [
    {
      protocol           = "tcp"
      port_range         = "22"
      source_addresses   = ["192.168.1.0/24", "2002:1:2::/48"]
    },
    {
      protocol           = "tcp"
      port_range         = "80"
      source_addresses   = ["0.0.0.0/0", "::/0"]
    },
    {
      protocol           = "tcp"
      port_range         = "443"
      source_addresses   = ["0.0.0.0/0", "::/0"]
    },
  ]

  outbound_rule = [
    {
      protocol                = "tcp"
      port_range              = "53"
      destination_addresses   = ["0.0.0.0/0", "::/0"]
    },
    {
      protocol                = "udp"
      port_range              = "53"
      destination_addresses   = ["0.0.0.0/0", "::/0"]
    },
  ]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The Firewall name
- `droplet_ids` (Optional) - The list of the IDs of the Droplets assigned to the Firewall.
- `tags` (Optional) - The names of the Tags assigned to the Firewall.
- `inbound_rule` - (Optional) The inbound access rule block for the Firewall. The `inbound_rule` block is documented below.
- `outbound_rule` - (Optional) The outbound access rule block for the Firewall. The `outbound_rule` block is documented below.

`inbound_rule` supports the following:

- `protocol` - (Optional) The type of traffic to be allowed. This may be one of "tcp", "udp", or "icmp".
- `port_range` - (Optional) The ports on which traffic will be allowed specified as a string containing a single port, a range (e.g. "8000-9000"), or "1-65535" to open all ports for a protocol.
- `source_addresses` - (Optional) An array of strings containing the IPv4 addresses, IPv6 addresses, IPv4 CIDRs, and/or IPv6 CIDRs from which the inbound traffic will be accepted.
- `source_droplet_ids` - (Optional) An array containing the IDs of the Droplets from which the inbound traffic will be accepted.
- `source_tags` - (Optional) An array containing the names of Tags corresponding to groups of Droplets from which the inbound traffic will be accepted.
- `source_load_balancer_uids` - (Optional) An array containing the IDs of the Load Balancers from which the inbound traffic will be accepted.

`outbound_rule` supports the following:

- `protocol` - (Optional) The type of traffic to be allowed. This may be one of "tcp", "udp", or "icmp".
- `port_range` - (Optional) The ports on which traffic will be allowed specified as a string containing a single port, a range (e.g. "8000-9000"), or "1-65535" to open all ports for a protocol.
- `destination_addresses` - (Optional) An array of strings containing the IPv4 addresses, IPv6 addresses, IPv4 CIDRs, and/or IPv6 CIDRs to which the outbound traffic will be allowed.
- `destination_droplet_ids` - (Optional) An array containing the IDs of the Droplets to which the outbound traffic will be allowed.
- `destination_tags` - (Optional) An array containing the names of Tags corresponding to groups of Droplets to which the outbound traffic will be allowed. traffic.
- `destination_load_balancer_uids` - (Optional) An array containing the IDs of the Load Balancers to which the outbound traffic will be allowed.

» Attributes Reference

The following attributes are exported:

- `id` - A unique ID that can be used to identify and reference a Firewall.
- `status` - A status string indicating the current state of the Firewall. This can be "waiting", "succeeded", or "failed".
- `created_at` - A time value given in ISO8601 combined date and time format that represents when the Firewall was created.
- `pending_changes` - An list of object containing the fields, "droplet_id", "removing", and "status". It is provided to detail exactly which Droplets are having their security policies updated. When empty, all changes have been successfully applied.
- `name` - The name of the Firewall.
- `droplet_ids` - The list of the IDs of the Droplets assigned to the Firewall.
- `tags` - The names of the Tags assigned to the Firewall.
- `inbound_rules` - The inbound access rule block for the Firewall.
- `outbound_rules` - The outbound access rule block for the Firewall.

» Import

Firewalls can be imported using the firewall `id`, e.g.

```
terraform import digitalocean_firewall.myfirewall b8ecd2ab-2267-4a5e-8692-cbf1d32583e3
```

» digitalocean_floating_ip

Provides a DigitalOcean Floating IP to represent a publicly-accessible static IP addresses that can be mapped to one of your Droplets.

» Example Usage

```
resource "digitalocean_droplet" "foobar" {
  name               = "baz"
  size               = "1gb"
  image              = "centos-5-8-x32"
  region             = "sgp1"
  ipv6               = true
  private_networking = true
}

resource "digitalocean_floating_ip" "foobar" {
  droplet_id = "${digitalocean_droplet.foobar.id}"
```

```
  region     = "${digitalocean_droplet.foobar.region}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Required) The region that the Floating IP is reserved to.
- `droplet_id` - (Optional) The ID of Droplet that the Floating IP will be assigned to.

NOTE: A Floating IP can be assigned to a region OR a droplet_id. If both region AND droplet_id are specified, then the Floating IP will be assigned to the droplet and use that region

» Attributes Reference

The following attributes are exported:

- `ip_address` - The IP Address of the resource

» Import

Floating IPs can be imported using the `ip`, e.g.

```
terraform import digitalocean_floating_ip.myip 192.168.0.1
```

» digitalocean_loadbalancer

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

» Example Usage

```
resource "digitalocean_droplet" "web" {
  name      = "web-1"
  size      = "512mb"
  image     = "centos-7-x64"
  region    = "nyc3"
}

resource "digitalocean_loadbalancer" "public" {
  name = "loadbalancer-1"
```

```
  region = "nyc3"

  forwarding_rule {
    entry_port = 80
    entry_protocol = "http"

    target_port = 80
    target_protocol = "http"
  }

  healthcheck {
    port = 22
    protocol = "tcp"
  }

  droplet_ids = ["${digitalocean_droplet.web.id}"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The Load Balancer name
- `region` - (Required) The region to start in
- `algorithm` - (Optional) The load balancing algorithm used to determine which backend Droplet will be selected by a client. It must be either `round_robin` or `least_connections`. The default value is `round_robin`.
- `forwarding_rule` - (Required) A list of `forwarding_rule` to be assigned to the Load Balancer. The `forwarding_rule` block is documented below.
- `healthcheck` - (Optional) A `healthcheck` block to be assigned to the Load Balancer. The `healthcheck` block is documented below. Only 1 healthcheck is allowed.
- `sticky_sessions` - (Optional) A `sticky_sessions` block to be assigned to the Load Balancer. The `sticky_sessions` block is documented below. Only 1 sticky_sessions block is allowed.
- `redirect_http_to_https` - (Optional) A boolean value indicating whether HTTP requests to the Load Balancer on port 80 will be redirected to HTTPS on port 443. Default value is `false`.
- `droplet_ids` (Optional) - A list of the IDs of each droplet to be attached to the Load Balancer.
- `droplet_tag` (Optional) - The name of a Droplet tag corresponding to Droplets to be assigned to the Load Balancer.

`forwarding_rule` supports the following:

- `entry_protocol` - (Required) The protocol used for traffic to the Load

Balancer. The possible values are: `http`, `https`, or `tcp`.

- `entry_port` - (Required) An integer representing the port on which the Load Balancer instance will listen.
- `target_protocol` - (Required) The protocol used for traffic from the Load Balancer to the backend Droplets. The possible values are: `http`, `https`, or `tcp`.
- `target_port` - (Required) An integer representing the port on the backend Droplets to which the Load Balancer will send traffic.
- `certificate_id` - (Optional) The ID of the TLS certificate to be used for SSL termination.
- `tls_passthrough` - (Optional) A boolean value indicating whether SSL encrypted traffic will be passed through to the backend Droplets. The default value is `false`.

`sticky_sessions` supports the following:

- `type` - (Required) An attribute indicating how and if requests from a client will be persistently served by the same backend Droplet. The possible values are `cookies` or `none`. If not specified, the default value is `none`.
- `cookie_name` - (Optional) The name to be used for the cookie sent to the client. This attribute is required when using `cookies` for the sticky sessions type.
- `cookie_ttl_seconds` - (Optional) The number of seconds until the cookie set by the Load Balancer expires. This attribute is required when using `cookies` for the sticky sessions type.

`healthcheck` supports the following:

- `protocol` - (Required) The protocol used for health checks sent to the backend Droplets. The possible values are `http` or `tcp`.
- `port` - (Optional) An integer representing the port on the backend Droplets on which the health check will attempt a connection.
- `path` - (Optional) The path on the backend Droplets to which the Load Balancer instance will send a request.
- `check_interval_seconds` - (Optional) The number of seconds between between two consecutive health checks. If not specified, the default value is `10`.
- `response_timeout_seconds` - (Optional) The number of seconds the Load Balancer instance will wait for a response until marking a health check as failed. If not specified, the default value is `5`.
- `unhealthy_threshold` - (Optional) The number of times a health check must fail for a backend Droplet to be marked "unhealthy" and be removed from the pool. If not specified, the default value is `3`.
- `healthy_threshold` - (Optional) The number of times a health check must pass for a backend Droplet to be marked "healthy" and be re-added to the pool. If not specified, the default value is `5`.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Load Balancer
- `ip`- The ip of the Load Balancer

» digitalocean_record

Provides a DigitalOcean DNS record resource.

» Example Usage

```
# Create a new domain
resource "digitalocean_domain" "default" {
  name       = "www.example.com"
  ip_address = "${digitalocean_droplet.foo.ipv4_address}"
}

# Add a record to the domain
resource "digitalocean_record" "foobar" {
  domain = "${digitalocean_domain.default.name}"
  type   = "A"
  name   = "foobar"
  value  = "192.168.0.11"
}
```

» Argument Reference

The following arguments are supported:

- `type` - (Required) The type of record
- `domain` - (Required) The domain to add the record to
- `value` - (Optional) The value of the record
- `name` - (Optional) The name of the record
- `weight` - (Optional) The weight of the record, for SRV records.
- `port` - (Optional) The port of the record, for SRV records.
- `priority` - (Optional) The priority of the record, for MX and SRV records.
- `ttl` - (Optional) The time to live for the record, in seconds.

» Attributes Reference

The following attributes are exported:

- `id` - The record ID
- `fqdn` - The FQDN of the record

» digitalocean_ssh_key

Provides a DigitalOcean SSH key resource to allow you to manage SSH keys for Droplet access. Keys created with this resource can be referenced in your droplet configuration via their ID or fingerprint.

» Example Usage

```
# Create a new SSH key
resource "digitalocean_ssh_key" "default" {
  name       = "Terraform Example"
  public_key = "${file("/Users/terraform/.ssh/id_rsa.pub")}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the SSH key for identification
- `public_key` - (Required) The public key. If this is a file, it can be read using the file interpolation function

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID of the key
- `name` - The name of the SSH key
- `public_key` - The text of the public key
- `fingerprint` - The fingerprint of the SSH key

» Import

SSH Keys can be imported using the `ssh key id`, e.g.

```
terraform import digitalocean_ssh_key.mykey 263654
```

» digitalocean_tag

Provides a DigitalOcean Tag resource. A Tag is a label that can be applied to a droplet resource in order to better organize or facilitate the lookups and actions on it. Tags created with this resource can be referenced in your droplet configuration via their ID or name.

» Example Usage

```
# Create a new tag
resource "digitalocean_tag" "foobar" {
  name = "foobar"
}

# Create a new droplet in nyc3 with the foobar tag
resource "digitalocean_droplet" "web" {
  image  = "ubuntu-16-04-x64"
  name   = "web-1"
  region = "nyc3"
  size   = "512mb"
  tags   = ["${digitalocean_tag.foobar.id}"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the tag

» Attributes Reference

The following attributes are exported:

- `id` - The id of the tag
- `name` - The name of the tag

» Import

Tags can be imported using the `name`, e.g.

```
terraform import digitalocean_tag.mytag tagname
```

» digitalocean_volume

Provides a DigitalOcean Block Storage volume which can be attached to a Droplet in order to provide expanded storage.

» Example Usage

```
resource "digitalocean_volume" "foobar" {
  region      = "nyc1"
  name        = "baz"
  size        = 100
  description = "an example volume"
}

resource "digitalocean_droplet" "foobar" {
  name       = "baz"
  size       = "1gb"
  image      = "coreos-stable"
  region     = "nyc1"
  volume_ids = ["${digitalocean_volume.foobar.id}"]
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Required) The region that the block storage volume will be created in.
- `name` - (Required) A name for the block storage volume. Must be lowercase and be composed only of numbers, letters and "-", up to a limit of 64 characters.
- `size` - (Required) The size of the block storage volume in GiB.
- `description` - (Optional) A free-form text field up to a limit of 1024 bytes to describe a block storage volume.
- `droplet_ids` - (Computed) A list of associated droplet ids

» Attributes Reference

The following attributes are exported:

- `id` - The unique identifier for the block storage volume.

» Import

Volumes can be imported using the `volume id`, e.g.

```
terraform import digitalocean_volume.volumea 506f78a4-e098-11e5-ad9f-000f53306ae1
```