

» digitalocean__certificate

Get information on a certificate. This data source provides the name, type, state, domains, expiry date, and the sha1 fingerprint as configured on your DigitalOcean account. This is useful if the certificate in question is not managed by Terraform or you need to utilize any of the certificates data.

An error is triggered if the provided certificate name does not exist.

» Example Usage

Get the certificate:

```
data "digitalocean_certificate" "example" {  
  name = "example"  
}
```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The name of certificate.

» Attributes Reference

The following attributes are exported:

- **id**: The ID of the certificate.
- **type**: The type of the certificate.
- **state**: the current state of the certificate.
- **domains**: Domains for which the certificate was issued.
- **not_after**: The expiration date and time of the certificate.
- **sha1_fingerprint**: The SHA1 fingerprint of the certificate.

» digitalocean__domain

Get information on a domain. This data source provides the name, TTL, and zone file as configured on your DigitalOcean account. This is useful if the domain name in question is not managed by Terraform or you need to utilize TTL or zone file data.

An error is triggered if the provided domain name is not managed with your DigitalOcean account.

» Example Usage

Get the zone file for a domain:

```
data "digitalocean_domain" "example" {
  name = "example.com"
}

output "domain_output" {
  value = "${data.digitalocean_domain.example.zone_file}"
}

$ terraform apply
```

```
data.digitalocean_domain.example: Refreshing state...
```

```
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

Outputs:

```
domain_output = $ORIGIN example.com.
$TTL 1800
example.com. IN SOA ns1.digitalocean.com. hostmaster.example.com. 1516944700 10800 3600 604800
example.com. 1800 IN NS ns1.digitalocean.com.
example.com. 1800 IN NS ns2.digitalocean.com.
example.com. 1800 IN NS ns3.digitalocean.com.
www.example.com. 3600 IN A 176.107.155.137
db.example.com. 3600 IN A 179.189.166.115
jira.example.com. 3600 IN A 207.189.228.15
```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The name of the domain.

» Attributes Reference

The following attributes are exported:

- **ttl**: The TTL of the domain.
- **zone_file**: The zone file of the domain.

» digitalocean__droplet

Get information on a Droplet for use in other resources. This data source provides all of the Droplet's properties as configured on your DigitalOcean account. This is useful if the Droplet in question is not managed by Terraform or you need to utilize any of the Droplets data.

An error is triggered if the provided Droplet name does not exist.

» Example Usage

Get the Droplet:

```
data "digitalocean_droplet" "example" {  
  name = "web"  
}
```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The name of Droplet.

» Attributes Reference

The following attributes are exported:

- **id**: The ID of the Droplet.
- **region** - The region the Droplet is running in.
- **image** - The Droplet image ID or slug.
- **size** - The unique slug that identifies the type of Droplet.
- **disk** - The size of the Droplets disk in GB.
- **vcpus** - The number of the Droplets virtual CPUs.
- **memory** - The amount of the Droplets memory in MB.
- **price_hourly** - Droplet hourly price.
- **price_monthly** - Droplet monthly price.
- **status** - The status of the Droplet.
- **locked** - Whether the Droplet is locked.
- **ipv6_address** - The Droplets public IPv6 address
- **ipv6_address_private** - The Droplets private IPv6 address
- **ipv4_address** - The Droplets public IPv4 address
- **ipv4_address_private** - The Droplets private IPv4 address
- **backups** - Whether backups are enabled.
- **ipv6** - Whether IPv6 is enabled.
- **private_networking** - Whether private networks are enabled.

- **monitoring** - Whether monitoring agent is installed.
- **volume_ids** - List of the IDs of each volumes attached to the Droplet.
- **tags** - A list of the tags associated to the Droplet.

» digitalocean__floating_ip

Get information on a floating ip. This data source provides the region and Droplet id as configured on your DigitalOcean account. This is useful if the floating IP in question is not managed by Terraform or you need to find the Droplet the IP is attached to.

An error is triggered if the provided floating IP does not exist.

» Example Usage

Get the floating IP:

```
variable "public_ip" {}

data "digitalocean_floating_ip" "example" {
  ip_address = "${var.public_ip}"
}
```

» Argument Reference

The following arguments are supported:

- **ip_address** - (Required) The allocated IP address of the specific floating IP to retrieve.

» Attributes Reference

The following attributes are exported:

- **region**: The region that the floating IP is reserved to.
- **droplet_id**: The Droplet id that the floating IP has been assigned to.

» digitalocean__image

Get information on an images for use in other resources (e.g. creating a Droplet based on snapshot). This data source provides all of the image properties as

configured on your DigitalOcean account. This is useful if the image in question is not managed by Terraform or you need to utilize any of the image's data.

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   = "s-1vcpu-1gb"
}
```

Get the data about an official image:

```
data "digitalocean_image" "example2" {
  slug = "ubuntu-18-04-x64"
}
```

» Argument Reference

The following arguments are supported:

- **name** - (Optional) The name of the private image.
- **slug** - (Optional) The slug of the official image.

» Attributes Reference

The following attributes are exported:

- **id**: The ID of the image.
- **image** - The id of the image.
- **distribution** - The name of the distribution of the OS of the image.
- **min_disk_size**: The minimum 'disk' required for the image.

- **private** - Is image a public image or not. Public images represent Linux distributions or One-Click Applications, while non-public images represent snapshots and backups and are only available within your account.
- **regions**: The regions that the image is available in.
- **type**: Type of the image.

» digitalocean_loadbalancer

Get information on a load balancer for use in other resources. This data source provides all of the load balancers properties as configured on your DigitalOcean account. This is useful if the load balancer in question is not managed by Terraform or you need to utilize any of the load balancers data.

An error is triggered if the provided load balancer name does not exist.

» Example Usage

Get the load balancer:

```
data "digitalocean_loadbalancer" "example" {
  name = "app"
}
```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The name of load balancer.

» Attributes Reference

See the Load Balancer Resource for details on the returned attributes - they are identical.

» digitalocean_record

Get information on a DNS record. This data source provides the name, TTL, and zone file as configured on your DigitalOcean account. This is useful if the record in question is not managed by Terraform.

An error is triggered if the provided domain name or record are not managed with your DigitalOcean account.

» Example Usage

Get data from a DNS record:

```
data "digitalocean_record" "example" {
  domain = "example.com"
  name   = "test"
}

output "record_type" {
  value = "${data.digitalocean_record.example.type}"
}

output "record_ttl" {
  value = "${data.digitalocean_record.example.ttl}"
}

$ terraform apply
```

```
data.digitalocean_record.example: Refreshing state...
```

```
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

Outputs:

```
record_ttl = 3600
record_type = A
```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The name of the record.
- **domain** - (Required) The domain name of the record.

» Attributes Reference

The following attributes are exported:

- **id**: The ID of the record.
- **type**: The type of the DNS record.
- **data**: Variable data depending on record type. For example, the "data" value for an A record would be the IPv4 address to which the domain will be mapped. For a CAA record, it would contain the domain name of the CA being granted permission to issue certificates.
- **priority**: The priority for SRV and MX records.

- **port:** The port for SRV records.
- **ttl:** This value is the time to live for the record, in seconds. This defines the time frame that clients can cache queried information before a refresh should be requested.
- **weight:** The weight for SRV records.
- **flags:** An unsigned integer between 0-255 used for CAA records.
- **tag:** The parameter tag for CAA records.

» digitalocean_ssh_key

Get information on a ssh key. This data source provides the name, public key, and fingerprint as configured on your DigitalOcean account. This is useful if the ssh key in question is not managed by Terraform or you need to utilize any of the keys data.

An error is triggered if the provided ssh key name does not exist.

» Example Usage

Get the ssh key:

```
data "digitalocean_ssh_key" "example" {
  name = "example"
}
```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The name of the ssh key.

» Attributes Reference

The following attributes are exported:

- **id:** The ID of the ssh key.
- **public_key:** The public key of the ssh key.
- **fingerprint:** The fingerprint of the public key of the ssh key.

» **digitalocean__tag**

Get information on a tag. This data source provides the name as configured on your DigitalOcean account. This is useful if the tag name in question is not managed by Terraform or you need validate if the tag exists in the account.

An error is triggered if the provided tag name does not exist.

» **Example Usage**

Get the tag:

```
data "digitalocean_tag" "example" {  
  name = "example"  
}
```

» **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.

» **digitalocean__volume**

Get information on a volume for use in other resources. This data source provides all of the volumes properties as configured on your DigitalOcean account. This is useful if the volume in question is not managed by Terraform or you need to utilize any of the volumes data.

An error is triggered if the provided volume name does not exist.

» **Example Usage**

Get the volume:

```
data "digitalocean_volume" "example" {
  name    = "app-data"
  region  = "nyc3"
}
```

Reuse the data about a volume to attach it to a Droplet:

```
data "digitalocean_volume" "example" {
  name    = "app-data"
  region  = "nyc3"
}

resource "digitalocean_droplet" "example" {
  name      = "foo"
  size      = "s-1vcpu-1gb"
  image     = "ubuntu-18-04-x64"
  region    = "nyc3"
}

resource "digitalocean_volume_attachment" "foobar" {
  droplet_id = "${digitalocean_droplet.example.id}"
  volume_id  = "${data.digitalocean_volume.example.id}"
}
```

» Argument Reference

The following arguments are supported:

- **name** - (Required) The name of block storage volume.
- **region** - (Optional) The region the block storage volume is provisioned in.

» Attributes Reference

The following attributes are exported:

- **id**: The ID of the block storage volume.
- **size** - The size of the block storage volume in GiB.
- **description** - Text describing a block storage volume.
- **filesystem_type** - Filesystem type currently in-use on the block storage volume.
- **filesystem_label** - Filesystem label currently in-use on the block storage volume.
- **droplet_ids** - A list of associated Droplet ids.

» digitalocean__volume__snapshot

Volume snapshots are saved instances of a block storage volume. Use this data source to retrieve the ID of a DigitalOcean volume snapshot for use in other resources.

» Example Usage

Get the volume snapshot:

```
data "digitalocean_volume_snapshot" "snapshot" {
  name_regex = "^web"
  region     = "nyc3"
  most_recent = true
}
```

Reuse the data about a volume snapshot to create a new volume based on it:

```
data "digitalocean_volume_snapshot" "snapshot" {
  name_regex = "^web"
  region     = "nyc3"
  most_recent = true
}

resource "digitalocean_volume" "foobar" {
  region    = "nyc3"
  name      = "baz"
  size      = 100
  snapshot_id = "${data.digitalocean_volume_snapshot.snapshot.id}"
}
```

» Argument Reference

- **name** - (Optional) The name of the volume snapshot.
- **name_regex** - (Optional) A regex string to apply to the volume snapshot list returned by DigitalOcean. This allows more advanced filtering not supported from the DigitalOcean API. This filtering is done locally on what DigitalOcean returns.
- **region** - (Optional) A "slug" representing a DigitalOcean region (e.g. `nyc1`). If set, only volume snapshots available in the region will be returned.
- **most_recent** - (Optional) If more than one result is returned, use the most recent volume snapshot.

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 volume snapshot ID only, or use `most_recent` to choose the most recent one.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the volume snapshot.
- `created_at` - The date and time the volume snapshot was created.
- `min_disk_size` - The minimum size in gigabytes required for a volume to be created based on this volume snapshot.
- `regions` - A list of DigitalOcean region "slugs" indicating where the volume snapshot is available.
- `volume_id` - The ID of the volume from which the volume snapshot originated.
- `size` - The billable size of the volume snapshot in gigabytes.

» `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. The certificate can either be a custom one provided by you or automatically generated one with Let's Encrypt.

» Example Usage

```
# Create a new TLS certificate
resource "digitalocean_certificate" "cert" {
  name           = "Terraform Example"
  type           = "custom"
  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.
- **type** - (Optional) The type of certificate to provision. Can be either `custom` or `lets_encrypt`. Defaults to `custom`.
- **private_key** - (Optional) The contents of a PEM-formatted private-key corresponding to the SSL certificate. Only valid when type is `custom`.
- **leaf_certificate** - (Optional) The contents of a PEM-formatted public TLS certificate. Only valid when type is `custom`.
- **certificate_chain** - (Optional) The full PEM-formatted trust chain between the certificate authority's certificate and your domain's TLS certificate. Only valid when type is `custom`.
- **domains** - (Optional) List of fully qualified domain names (FQDNs) for which the certificate will be issued. The domains must be managed using DigitalOcean's DNS. Only valid when type is `lets_encrypt`.

» 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

» Import

Certificates can be imported using the certificate `id`, e.g.

```
terraform import digitalocean_certificate.mycertificate 892071a0-bb95-49bc-8021-3afd67a210b1
```

» digitalocean__domain

Provides a DigitalOcean domain resource.

» Example Usage

```
# Create a new domain
resource "digitalocean_domain" "default" {
  name      = "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** - (Optional) The IP address of the domain. If specified, this IP is used to create an initial A record for the domain.

» 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-18-04-x64"
  name   = "web-1"
  region = "nyc2"
  size   = "s-1vcpu-1gb"
}
```

» 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 unique slug that identifies the type of Droplet. You can find a list of available slugs on DigitalOcean API documentation.
- **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.

NOTE: If you use `volume_ids` on a Droplet, Terraform will assume management over the full set volumes for the instance, and treat additional volumes as a drift. For this reason, `volume_ids` must not be mixed with external `digitalocean_volume_attachment` resources for a given instance.

» 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      = "s-1vcpu-1gb"  
  image     = "ubuntu-18-04-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"]
    },
    {
      protocol      = "icmp"
      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"]
    },
    {
      protocol      = "icmp"
      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** - (Required) 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. Required for when protocol is tcp or udp.
- **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** - (Required) 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. Required for when protocol is tcp or udp.
- **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.

NOTE: Floating IPs can be assigned to a Droplet either directly on the `digitalocean_floating_ip` resource by setting a `droplet_id` or using the `digitalocean_floating_ip_assignment` resource, but the two cannot be used together.

» Example Usage

```
resource "digitalocean_droplet" "foobar" {
  name      = "baz"
  size      = "s-1vcpu-1gb"
```

```

    image            = "ubuntu-18-04-x64"
    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.

» 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__floating_ip__assignment

Provides a resource for assigning an existing DigitalOcean Floating IP to a Droplet. This makes it easy to provision floating IP addresses that are not tied to the lifecycle of your Droplet.

» Example Usage

```

resource "digitalocean_floating_ip" "foobar" {
  region = "sgp1"
}

```

```

resource "digitalocean_droplet" "foobar" {
  name           = "baz"
  size           = "s-1vcpu-1gb"
  image          = "ubuntu-18-04-x64"
  region        = "sgp1"
  ipv6           = true
  private_networking = true
}

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

```

» Argument Reference

The following arguments are supported:

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

» 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     = "s-1vcpu-1gb"
  image    = "ubuntu-18-04-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}"]
}

```

When managing certificates attached to the load balancer, make sure to add the `create_before_destroy` lifecycle property in order to ensure the certificate is correctly updated when changed. The order of operations will then be: Create new certificate -> Update loadbalancer with new certificate -> Delete old certificate. When doing so, you must also change the name of the certificate, as there cannot be multiple certificates with the same name in an account.

```

resource "digitalocean_certificate" "cert" {
    name          = "cert"
    private_key   = "${file("key.pem")}"
    leaf_certificate = "${file("cert.pem")}"

    lifecycle {
        create_before_destroy = true
    }
}

resource "digitalocean_droplet" "web" {
    name      = "web-1"
    size      = "s-1vcpu-1gb"
    image     = "ubuntu-18-04-x64"
    region    = "nyc3"
}

resource "digitalocean_loadbalancer" "public" {
    name = "loadbalancer-1"
    region = "nyc3"

    forwarding_rule {
        entry_port = 443
        entry_protocol = "https"
    }
}

```

```

    target_port = 80
    target_protocol = "http"

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

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 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

» Import

Load Balancers can be imported using the `id`, e.g.

```
terraform import digitalocean_loadbalancer.myloadbalancer 4de7ac8b-495b-4884-9a69-1050c67930
```

» `digitalocean__record`

Provides a DigitalOcean DNS record resource.

» Example Usage

```
# Create a new domain
resource "digitalocean_domain" "default" {
  name = "example.com"
}

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

# Output the FQDN for the record
output "fqdn" {
  value = "${digitalocean_record.www.fqdn}"
}
```

» Argument Reference

The following arguments are supported:

- `type` - (Required) The type of record. Must be one of `A`, `AAAA`, `CAA`, `CNAME`, `MX`, `NS`, `TXT`, or `SRV`.
- `domain` - (Required) The domain to add the record to.
- `value` - (Required) The value of the record.
- `name` - (Required) The name of the record.

- **port** - (Optional) The port of the record. Only valid when type is **SRV**. Must be between 1 and 65535.
- **priority** - (Optional) The priority of the record. Only valid when type is **MX** or **SRV**. Must be between 0 and 65535.
- **weight** - (Optional) The weight of the record. Only valid when type is **SRV**. Must be between 0 and 65535.
- **ttl** - (Optional) The time to live for the record, in seconds. Must be at least 0.
- **flags** - (Optional) The flags of the record. Only valid when type is **CAA**. Must be between 0 and 255.
- **tag** - (Optional) The tag of the record. Only valid when type is **CAA**. Must be one of **issue**, **wildissue**, or **iodef**.

» Attributes Reference

The following attributes are exported:

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

» Import

Records can be imported using the domain name and record **id** when joined with a comma. See the following example:

```
terraform import digitalocean_record.example_record example.com,12345678
```

» 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")}"
}
```

```
# Create a new Droplet using the SSH key
```

```
resource "digitalocean_droplet" "web" {
  image    = "ubuntu-18-04-x64"
  name     = "web-1"
  region  = "nyc3"
  size     = "s-1vcpu-1gb"
  ssh_keys = ["${digitalocean_ssh_key.default.fingerprint}"]
}
```

» 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-18-04-x64"
  name   = "web-1"
  region = "nyc3"
  size   = "s-1vcpu-1gb"
  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
}

```

```

    initial_filesystem_type = "ext4"
    description              = "an example volume"
  }

  resource "digitalocean_droplet" "foobar" {
    name      = "baz"
    size      = "s-1vcpu-1gb"
    image     = "ubuntu-18-04-x64"
    region    = "nyc1"
  }

  resource "digitalocean_volume_attachment" "foobar" {
    droplet_id = "${digitalocean_droplet.foobar.id}"
    volume_id  = "${digitalocean_volume.foobar.id}"
  }

```

You can also create a volume from an existing snapshot.

```

data "digitalocean_volume_snapshot" "foobar" {
  name = "baz"
}

resource "digitalocean_volume" "foobar" {
  region    = "lon1"
  name      = "foo"
  size      = "${data.digitalocean_volume_snapshot.foobar.min_disk_size}"
  snapshot_id = "${data.digitalocean_volume_snapshot.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. If updated, can only be expanded.
- **description** - (Optional) A free-form text field up to a limit of 1024 bytes to describe a block storage volume.
- **snapshot_id** - (Optional) The ID of an existing volume snapshot from which the new volume will be created. If supplied, the region and size will be limited on creation to that of the referenced snapshot

- `initial_filesystem_type` - (Optional) Initial filesystem type (`xfs` or `ext4`) for the block storage volume.
- `initial_filesystem_label` - (Optional) Initial filesystem label for the block storage volume.

» Attributes Reference

The following attributes are exported:

- `id` - The unique identifier for the block storage volume.
- `filesystem_type` - Filesystem type (`xfs` or `ext4`) for the block storage volume.
- `filesystem_label` - Filesystem label for the block storage volume.
- `droplet_ids` - A list of associated droplet ids.

» Import

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

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

» `digitalocean__volume__attachment`

Manages attaching a Volume to a Droplet.

NOTE: Volumes can be attached either directly on the `digitalocean_droplet` resource, or using the `digitalocean_volume_attachment` resource - but the two cannot be used together. If both are used against the same Droplet, the volume attachments will constantly drift.

» Example Usage

```
resource "digitalocean_volume" "foobar" {
  region          = "nyc1"
  name            = "baz"
  size            = 100
  initial_filesystem_type = "ext4"
  description     = "an example volume"
}

resource "digitalocean_droplet" "foobar" {
  name      = "baz"
  size     = "s-1vcpu-1gb"
```

```

    image      = "ubuntu-18-04-x64"
    region     = "nyc1"
  }

  resource "digitalocean_volume_attachment" "foobar" {
    droplet_id = "${digitalocean_droplet.foobar.id}"
    volume_id  = "${digitalocean_volume.foobar.id}"
  }

```

» Argument Reference

The following arguments are supported:

- `droplet_id` - (Required) ID of the Droplet to attach the volume to.
- `volume_id` - (Required) ID of the Volume to be attached to the Droplet.

» Attributes Reference

The following attributes are exported:

- `id` - The unique identifier for the volume attachment.

» `digitalocean__volume__snapshot`

Provides a DigitalOcean Volume Snapshot which can be used to create a snapshot from an existing volume.

» Example Usage

```

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

resource "digitalocean_volume_snapshot" "foobar" {
  name      = "foo"
  volume_id = "${digitalocean_volume.foobar.id}"
}

```

» Argument Reference

The following arguments are supported:

- **name** - (Required) A name for the volume snapshot.
- **volume_id** - (Required) The ID of the volume from which the volume snapshot originated.

» Attributes Reference

The following attributes are exported:

- **id** The ID of the volume snapshot.
- **created_at** - The date and time the volume snapshot was created.
- **min_disk_size** - The minimum size in gigabytes required for a volume to be created based on this volume snapshot.
- **regions** - A list of DigitalOcean region "slugs" indicating where the volume snapshot is available.
- **size** - The billable size of the volume snapshot in gigabytes.

» Import

Volume Snapshots can be imported using the `snapshot id`, e.g.

```
terraform import digitalocean_volume_snapshot.snapshot 506f78a4-e098-11e5-ad9f-000f53306ae1
```