» hcloud_server

Provides an Hetzner Cloud server resource. This can be used to create, modify, and delete Servers. Servers also support provisioning.

» Example Usage

```
# Create a new server running debian
resource "hcloud_server" "node1" {
  name = "node1"
  image = "debian-9"
  server_type = "cx11"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the server to create (must be unique per project and a valid hostname as per RFC 1123).
- `server_type` - (Required) Name of the server type this server should be created with.
- `image` - (Required) Name or ID of the image the server is created from.
- `location` - (Optional) The location name to create the server in.
- `datacenter` - (Optional) The datacenter name to create the server in.
- `user_data` - (Optional) Cloud-Init user data to use during server creation
- `ssh_keys` - (Optional) SSH key IDs or names which should be injected into the server at creation time
- `keep_disk` - (Optional) If true, do not upgrade the disk. This allows downgrading the server type later.
- `backup_window` - (Optional) Enable and configure backups for a server. Time window (UTC) in which the backup will run, choices: `22-02 02-06 06-10 10-14 14-18 18-22`
- `iso` - (Optional) Name of an ISO image to mount.
- `rescue` - (Optional) Enable and boot in to the specified rescue system. This enables simple installation of custom operating systems. `linux64 linux32` or `freebsd64`

» Attributes Reference

The following attributes are exported:

- `id` - Unique ID of the server.
- `name` - Name of the server.

- `server_type` - Name of the server type.
- `image` - Name or ID of the image the server was created from.
- `location` - The location name.
- `datacenter` - The datacenter name.
- `backup_window` - The backup window of the server, if enabled.
- `iso` - Name of the mounted ISO image.
- `ipv4_address` - The IPv4 address.
- `ipv6_address` - The IPv6 address.
- `status` - The status of the server.

» Import

Servers can be imported using the server `id`:

```
terraform import hcloud_server.myserver <id>
```

» hcloud_ssh_key

Provides a Hetzner Cloud SSH key resource to manage SSH keys for server access.

» Example Usage

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

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the SSH key.
- `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`:

```
terraform import hcloud_ssh_key.mykey <id>
```

» hcloud_floating_ip

Provides a Hetzner Cloud Floating IP to represent a publicly-accessible static IP addresses that can be mapped to one of your Servers.

» Example Usage

```
resource "hcloud_server" "node1" {
  name = "node1"
  image = "debian-9"
  server_type = "cx11"
}

resource "hcloud_floating_ip" "master" {
  type = "ipv4"
  server_id = "${hcloud_server.node1.id}"
}
```

» Argument Reference

- `type` - (Required) Type of the Floating IP. `ipv4` `ipv6`
- `server_id` - (Optional) Server to assign the Floating IP to.
- `home_location` - (Optional) Home location (routing is optimized for that location). Optional if server_id argument is passed.
- `description` - (Optional) Description of the Floating IP.

» Attributes Reference

- `id` - Unique ID of the Floating IP.
- `type` - Type of the Floating IP.
- `server_id` - Server to assign the Floating IP is assigned to.

- `home_location` - Home location.
- `description` - Description of the Floating IP.
- `ip_address` - IP Address of the Floating IP.

» Import

Floating IPs can be imported using its `id`:

```
terraform import hcloud_floating_ip.myip <id>
```