» cherryservers_server

Provides a CherryServers Server resource. This can be used to create, modify, and delete Servers. Servers also support provisioning.

» Example Usage

```
variable "region" {
  default = "EU-East-1"
}
variable "image" {
  default = "Ubuntu 18.04 64bit"
}
variable "plan_id" {
  default = "86"
}
# Create a new Web Server in the EU-East-1 region
# Optional provisioning is specified as well.
resource "cherryservers_server" "my-server" {
  project_id = "${cherryservers_project.myproject.id}"
  region = "${var.region}"
  hostname = "production_server"
  image = "${var.image}"
  plan_id = "${var.plan_id}"
  ssh_keys_ids = [
    "${cherryservers_ssh.mykey.id}"]
  ip_addresses_ids = [
    "${cherryservers_ip.my-ip.id}"]

  # Upload your setup script
  provisioner "file" {
    source = "my_setup_script.sh"
    destination = "/tmp/my_setup_script.sh"

    connection {
      type = "ssh"
      user = "root"
      host = "${self.primary_ip}"
      private_key = "${file(var.private_key)}"
      timeout = "20m"
    }
  }
  # Make setup script executable and run it
  provisioner "remote-exec" {
```

```
    inline = [
      "chmod +x /tmp/my_setup_script.sh",
      "/tmp/my_setup_script.sh",
    ]
    connection {
      type = "ssh"
      user = "root"
      host = "${self.primary_ip}"
      private_key = "${file(var.private_key)}"
      timeout = "20m"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The Project ID to deploy the IP in.
- `region` - (Required) The region to deploy in. You can find these on CherryServers API.
- `image` - (Required) The Server image slug. You can find these on CherryServers API, after finding your plan_id.
- `hostname` - (Required) The Server name.
- `plan_id` - (Required) The unique slug that indentifies the type of Server. You can find a list of available slugs on CherryServers API.
- `ssh_key_ids` - (Optional) A list of SSH IDs to enable in the format `[12345, 123456]`.
- `ip_addresses_ids` - (Optional) A list of IP Address IDs to enable in the format `[12345, 123456]`.
- `user_data` - (Optional) Base64 encoded User-Data blob. It should be either bash or cloud-config script.
- `tags` - (Optional) Key/value metadata for server tagging.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Server
- `hostname` - The hostname of the Server
- `name`- The name of the Server
- `region` - The region of the Server
- `image` - The image of the Server
- `primary_ip` - The primary IPv4 address of the server. Servers will always have a primary ID in addition to any attached reserverd IPs

- `private_ip` - The private IPv4 address of the server
- `state` - The state of the server, such as "Pending"
- `power_state` - The power state of the server, such as "Powered off"
- `price` - The Server hourly price

» Import

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

```
terraform import cherryservers_server.myserver 100823
```

» cherryservers_ip

Provides a CherryServers IP resource. This can be used to create, modify, and delete IP.

» Example Usage

```
# Optionally configure the Region to launch in as a variable or specify inline below
variable "region" {
  default = "EU-East-1"
}

# To see how to configure a project, see the Cherryservers_project documentation.
# You will need to have an existing project in order to reserve an IP resource. You do not r
resource "cherryservers_ip" "my_ip_address" {
  project_id = "${cherryservers_project.myproject.id}"
  region = "${var.region}"
  routed_to_ip = "127.0.0.1" # Optional
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The Project ID to deploy the IP in.
- `region` - (Required) The region to deply in.
- `routed_to_hostname` - (Optional) A hostname to route network traffic to
- `routed_to_ip` - (Optional) An IP address to route network traffic to

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the IP Address
- `address` - The address of the IP eg. 127.0.0.1
- `cidr`- The CIDR block of the IP
- `gateway` - The Gateway address for the IP
- `a_record` - Public A record which points to the assigned IP address
- `ptr_record` - The PTR record that resolves to the IP address
- `type` - The type of IP
- `routed_to` - The hostname or IP that the IP routes to
- `region` - The region of the IP

» Import

IPs can be imported using the IP `id`, e.g.

```
terraform import cherryservers_ip.my_ip_address 123
```

» cherryservers_project

Provides a CherryServers Project resource. This can be used to create, modify, and delete Projects.

» Example Usage

```
# You must find your team_id for your account by login into the CherryServers portal: [https
variable "team_id" {
  default = "12345"
}
# Specify a name for your project
variable "project_name" {
  default = "My Cool New Project"
}

# Create a new Project for your team
resource "cherryservers_project" "myproject" {
  team_id = "${var.team_id}"
  name = "${var.project_name}"
}
```

» Argument Reference

The following arguments are supported:

- `team_id` - (Required) The Server image ID or slug.
- `name` - (Required) The Server name.

» Attributes Reference

The following attributes are exported:

- `team_id` - The ID of team that owns the Project
- `name`- The name of the Project
- `project_id` - The computed ID of the Project, to be used with other Resources

» Import

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

```
terraform import cherryservers_project.myproject 8123
```

» cherryservers_ssh

Provides a CherryServers SSH Key resource. This can be used to create, and delete SSH Keys associated with your account.

» Example Usage

```
# Create a new SSH Key for your account

# (Optionally) specify the path to your key as a terraform variable

variable "private_key" {
  default = "~/.ssh/cherry"
}

resource "cherryservers_ssh" "mysshkey" {
  name   = "mykey"
  public_key = "${file("${var.private_key}.pub")}" # The public key contents can also be sto
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The Server name.
- `public_key` - (Required) The public key contents as a string to be added to your account

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the SSH Key. Use this attribute when associating an SSH Key to a cherryservers_server resource
- `name`- The name of the SSH Key
- `fingerprint` - The fingerprint of your SSH Public key
- `created` - The date when this Key was added
- `updated` - The date when this Key was modified

» Import

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

```
terraform import cherryservers_ssh.mysshkey 900
```