# Skytap Provider

The Skytap provider is used to interact with the resources supported by Skytap. The provider needs to be configured with the proper credentials before it can be used.

Use the navigation to the left to read about the available resources.

A typical provider configuration will look something like:

```
provider "skytap" {
  username = "${var.skytap_username}"
  api_token = "${var.skytap_api_token}"
}

resource "skytap_environment" "env" {
    # ...
}
```

## Arguments Reference

The following arguments are supported:

- `username` - (Required) This is the Skytap username. This can also be specified with the `SKYTAP_USERNAME` shell environment variable.
- `api_token` - (Required) This is the Skytap API token. This can also be specified with the `SKYTAP_API_TOKEN` shell environment variable.

# skytap_project

Get information on a project. This data source provides the id, name, summary, auto_add_role_name and show_project_members properties of a project as configured on your Skytap account. This is useful in order to retrieve a project's id via its name.

An error is triggered if: 1. No projects can be retrieved. 2. The project does not exist. 3. More than one project matches the name.

## Example Usage

Get the project:

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

## Argument Reference

The following arguments are supported:

- `name` - (Required) The name of project.

## Attributes Reference

The following attributes are exported:

- `id` : The ID of the project.
- `summary` : The summary description of the project.
- `auto_add_role_name` : The role automatically assigned to every new user added to the project.
- `show_project_members` : Whether project members can view a list of the other project members.

# skytap_template

Get information on a template. This data source provides the id and name of a template as configured on your Skytap account. This is useful in order to retrieve a template's id via its name. The name field takes a regular expression to facilitate the matching process.

An error is triggered if: 1. No templates can be retrieved. 2. The template does not exist. 3. More than one template matches the name and the `most_recent` flag is not set.

If more than one templates are retrieved the `most_recent` can be set. This will sort the results in descending order according to the creation date. The newest template will be used.

## Example Usage

Get the template:

```
data "skytap_template" "example" {
    name = "18.04"
    most_recent = true
}
```

## Argument Reference

The following arguments are supported:

- `name` - (Required) A regular expression on the name of a template.
- `most_recent` - (Optional) Use the most recently created template from the returned list.

## Attributes Reference

The following attributes are exported:

- `id` : The ID of the template.

# skytap_environment

Provides a Skytap Environment resource. An environment consists of one or more virtual machines, networks, and associated settings and metadata. Unlike a template, an environment can be run and have most of its settings modified. When an environment is created all of its VMs will be run.

## Example Usage

```
# Create a new environment
resource "skytap_environment" "environment" {
  template_id = "123456"
  name = "Terraform Example"
  description = "Skytap terraform provider example environment."
}
```

## Argument Reference

The following arguments are supported:

- `template_id` - (Required, Force New) ID of the template you want to create an environment from. If updating with a new one then the environment will be recreated.
- `name` - (Required) User-defined name of the environment. Limited to 255 characters. UTF-8 character type. Will default to source template's name if null is provided.
- `description` - (Required) User-defined description of the environment. Limited to 1000 characters. Null allowed. UTF-8 character type.
- `outbound_traffic` - (Optional) Indicates whether networks in the environment can send outbound traffic.
- `routable` - (Optional) Indicates whether networks within the environment can route traffic to one another.
- `suspend_on_idle` - (Optional) The number of seconds an environment can be idle before it is automatically suspended. Valid range: 300 to 86400 seconds (5 minutes to 1 day).
- `suspend_at_time` - (Optional) The date and time that the environment will be automatically suspended. Format: yyyy/mm/dd hh:mm:ss. By default, the suspend time uses the UTC offset for the time zone defined in your user account settings. Optionally, a different UTC offset can be supplied (for example: 2018/07/20 14:20:00 -0000). The value in the API response is converted to your time zone.
- `shutdown_on_idle` - (Optional) The number of seconds an environment can be idle before it is automatically shut down. Valid range: 300 to 86400 seconds (5 minutes to 1 day).
- `shutdown_at_time` - (Optional) The date and time that the environment will be automatically shut down. Format: yyyy/mm/dd hh:mm:ss. By default, the suspend time uses the UTC offset for the time zone defined in your user account settings. Optionally, a different UTC offset can be supplied (for example: 2018/07/20 14:20:00 -0000). The value in the API response is converted to your time zone.

**NOTE:** * If `suspend_on_idle` and `suspend_at_time` are both null, automatic suspend is disabled. * If `shutdown_on_idle` and `shutdown_at_time` are both null, automatic shut down is disabled. * An environment cannot be set to automatically suspend and shut down. Only one of the following settings can take effect: `suspend_on_idle`, `suspend_at_time`, `shutdown_on_idle`, or `shutdown_at_time`. * When you send a request that updates one of the four suspend or shutdown options, the other three options are automatically set to null by the REST API. * If multiple suspend or shut down options are sent in the same request, the `suspend_type` field determines which setting Skytap Cloud will honor.

## Attributes Reference

The following attributes are exported:

- `id` : The ID of the environment.

# skytap_network

Provides a Skytap Network resource. Networks are not top-level elements of the Skytap API. Rather, they are elements properly contained within an environment. Operations on them are implicitly on the containing environment.

## Example Usage

```
# Create a new network
resource "skytap_network" "network" {
  environment_id = "123456"
  name = "my network"
  domain = "domain.com"
  subnet = "1.2.3.4/16"
  gateway = "1.2.3.254"
  tunnelable = true
}
```

## Argument Reference

The following arguments are supported:

- `environment_id` - (Required, Force New) ID of the environment you want to attach the network to. If updating with a new one then the network will be recreated.
- `name` - (Required) User-defined name of the network. Limited to 255 characters. UTF-8 character type.
- `domain` - (Required) Domain name for the Skytap network. Limited to 64 characters.

**NOTE:** Valid characters are lowercase letters, numbers, and hyphens. Cannot be blank, must not begin or end with a period, and must start and end with a letter or number. This field can be changed only when all virtual machines in the environment are stopped (not suspended or running).

- `subnet` - (Required) Defines the subnet address and subnet mask size in CIDR format (for example, 10.0.0.0/24). IP addresses for the VMs are assigned from this subnet and standard network services (DNS resolution, CIFS share, routes to Internet) are defined appropriately for it.

**NOTE:** The subnet mask size must be between 16 and 29. Valid characters are lowercase letters, numbers, and hyphens. Cannot be blank, must not begin or end with a period, and must start and end with a letter or number.

- `gateway` - (Optional, Computed) Gateway IP address.
- `tunnelable` - (Optional) If true, this network can be connected to other networks.

## Attributes Reference

The following attributes are exported:

- `id` : The ID of the network.

# skytap_project

Provides a Skytap Project resource. Projects are an access permissions model used to share environments, templates, and assets with other users.

## Example Usage

```
# Create a new project
resource "skytap_project" "project" {
  name = "Terraform Example"
  summary = "Skytap terraform provider example project."
  show_project_members = false
  auto_add_role_name = "participant"
}
```

## Argument Reference

The following arguments are supported:

- `name` - (Required) User-defined project name.
- `summary` - (Optional) User-defined description of project.
- `auto_add_role_name` - (Optional) If this field is set to `viewer`, `participant`, `editor`, or `manager`, new users added to your Skytap account are automatically added to this project with the specified project role. Existing users aren't affected by this setting. If the field is set to `null`, new users aren't automatically added to the project. For additional details, see Automatically adding new users to a project (https://help.skytap.com/csh-project-automatic-role.html).
- `show_project_members` - (Optional) Determines whether projects members can view a list of other project members. False by default.

## Attributes Reference

The following attributes are exported:

- `id` : The ID of the project.

# skytap_vm

Provides a Skytap Virtual Machine (VM) resource. The environment VM resource represents an image of a single virtual machine.

**NOTE:** * VMs do not exist outside of environments or templates. * An environment or template can have multiple VMs. * Each VM is a unique resource. Therefore, a VM in a template will have a different ID than a VM in an environment created from that template. * The VM will be run immediately after creation.

## Example Usage

```
# Create a new vm
resource "skytap_vm" "vm" {
  template_id = 123
  vm_id = 456
  environment_id = 789
  name = "my vm"
  cpu = 4
  ram = 4096

  os_disk_size = 40000

  disk = {
    name = "my disk"
    size = 4096
  }
  disk = {
      name = "my other disk"
      size = 4096
  }

  network_interface = {
     interface_type = "vmxnet3"
     network_id = "${skytap_network.my_network.id}"
     ip = "10.0.0.1"
     hostname = "myhost"

    published_service = {
      name = "ssh"
      internal_port = 22
    }
  }
}

# Will work after VM resource is created
output "ssh_ip" {
  value = "${skytap_vm.vm.service_ips.ssh}"
}
output "ssh_port" {
  value = "${skytap_vm.vm.service_ports.ssh}"
}
```

# Argument Reference

The following arguments are supported:

- `environment_id` - (Required, Force New) ID of the environment you want to add the VM to. If updating with a new one then the VM will be recreated.
- `template_id` - (Required, Force New) ID of the template you want to create the vm from. If updating with a new one then the VM will be recreated.
- `vm_id` - (Required, Force New) ID of the VM you want to create the VM from. If updating with a new one then the VM will be recreated.
- `name` - (Optional, Computed) User-defined name. Limited to 100 characters.
- `cpus` - (Optional, Computed) Number of CPUs allocated to this virtual machine. Valid range is 1 to 12. Maximum limit depends on the `max_cpus` setting.
- `ram` - (Optional, Computed) Amount of RAM allocated to this VM. Valid range is 256 and 131,072 (MB). Maximum limit depends on `max_ram` setting.
- `os_disk_size` - (Optional, Computed) The size of the OS disk. The disk size is in MiB; it will be converted to GiB in the Skytap UI. The maximum disk size is 2,096,128 MiB (1.999 TiB).
- `disk` - (Optional) Array of virtual disks within the VM. The disk size is in MiB; it will be converted to GiB in the Skytap UI. The maximum disk size is 2,096,128 MiB (1.999 TiB).
    - `name` - (Required) A unique name for the disk.
    - `size` - (Required) Specify the size of the disk. The new disk's size should be provided in MiB. The minimum disk size is 2048 MiB; the maximum is 2096128 MiB (1.999 TiB).

**NOTE:** The name will be truncated to 33 UTF-8 characters after saving. If a name is not provided then the source VM's name will be used.

- `network_interface` - (Optional, Computed, ForceNew) A Skytap network adapter is a virtualized network interface card (also known as a network adapter). It is logically contained in a VM and attached to a network.
    - `interface_type` - (Required, Force New) Type of network that this network adapter is attached to.
    - `network_id` - (Required, Force New) ID of the network that this network adapter is attached to.
    - `ip` - (Required, Force New) The IP address (for example, 10.1.0.37). Skytap will not assign the same IP address to multiple interfaces on the same network.
    - `hostname` - (Required, Force New) Limited to 32 characters. Valid characters are lowercase letters, numbers, and hyphens. Cannot begin or end with hyphens. Cannot be `gw`.
    - `published_service` - (Optional, Force New) Generally, a published service represents a binding of a port on a network interface to an IP and port that is routable and accessible from the public Internet. This mechanism is used to selectively expose ports on the guest to the public Internet.

**NOTE:** Published services exist and are managed as aspects of network interfaces—that is, as part of the overall environment element.

```
* `name` - (Required, Force New) A unique name for the published service.
* `internal_port` - (Required, Force New) The port that is exposed on the interface. Typically this will be dictated by standard usage (e.g., port 80 for http traffic, port 22 for SSH).
```

## Attributes Reference

The following attributes are exported:

- `id` : The ID of the VM.
- `max_cpus` : Maximum settable CPUs for this VM.
- `max_ram` : RAM settable CPUs for this VM.
- `disk` : The disks.
  - `id` : The ID for the disk.
  - `type` : The type of disk.
  - `controller` : The disk controller.
  - `lun` : The logical unit number of the disk (LUN).
- `network_interface` : The network adapters.
  - `id` : The network adapter's ID.
- `published_service` : The published services.
  - `id` : The published service's ID.
  - `external_ip` : The published service's external IP.
  - `external_port` : Each published service's external port.
- `service_ips` : A map of external IP addresses. The key is the name of a published service - as defined in the `published_service` block.
- `service_ports` : A map of external ports. The key is the name of a published service - as defined in the `published_service` block.