Scaleway Guide

Introduction

Scaleway is a cloud provider supported by Ansible, version 2.6 or higher via a dynamic inventory plugin and modules. Those modules are:

- :ref:`scaleway_sshkey_module`: adds a public SSH key from a file or value to the Packet infrastructure. Every subsequently-created device will have this public key installed in .ssh/authorized_keys.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 15); *backlink*

  Unknown interpreted text role "ref".

- :ref:`scaleway_compute_module`: manages servers on Scaleway. You can use this module to create, restart and delete servers.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 16); *backlink*

  Unknown interpreted text role "ref".

- :ref:`scaleway_volume_module`: manages volumes on Scaleway.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 17); *backlink*

  Unknown interpreted text role "ref".

Note

This guide assumes you are familiar with Ansible and how it works. If you're not, have a look at :ref:`ansible_documentation` before getting started.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 20); *backlink*

Unknown interpreted text role "ref".

Requirements

The Scaleway modules and inventory script connect to the Scaleway API using Scaleway REST API. To use the modules and inventory script you'll need a Scaleway API token. You can generate an API token via the Scaleway console here. The simplest way to authenticate yourself is to set the Scaleway API token in an environment variable:

```
$ export SCW_TOKEN=00000000-1111-2222-3333-444444444444
```

If you're not comfortable exporting your API token, you can pass it as a parameter to the modules using the `api_token` argument.

If you want to use a new SSH key pair in this tutorial, you can generate it to `./id_rsa` and `./id_rsa.pub` as:

```
$ ssh-keygen -t rsa -f ./id_rsa
```

If you want to use an existing key pair, just copy the private and public key over to the playbook directory.

How to add an SSH key?

Connection to Scaleway Compute nodes use Secure Shell. SSH keys are stored at the account level, which means that you can re-use the same SSH key in multiple nodes. The first step to configure Scaleway compute resources is to have at least one SSH key configured.

:ref:`scaleway_sshkey_module` is a module that manages SSH keys on your Scaleway account. You can add an SSH key to your account by including the following task in a playbook:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 56); *backlink*

Unknown interpreted text role "ref".

```
- name: "Add SSH key"
  scaleway_sshkey:
    ssh pub key: "ssh-rsa AAAA..."
    state: "present"
```

The `ssh_pub_key` parameter contains your ssh public key as a string. Here is an example inside a playbook:

```
- name: Test SSH key lifecycle on a Scaleway account
  hosts: localhost
  gather_facts: no
  environment:
    SCW_API_KEY: ""

  tasks:

    - scaleway_sshkey:
        ssh pub key: "ssh-rsa AAAAB...424242 developer@example.com"
        state: present
      register: result

    - assert:
        that:
          - result is success and result is changed
```

How to create a compute instance?

Now that we have an SSH key configured, the next step is to spin up a server! :ref:`scaleway_compute_module` is a module that can create, update and delete Scaleway compute instances:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 93); *backlink*

Unknown interpreted text role "ref".

```
- name: Create a server
  scaleway_compute:
    name: foobar
    state: present
    image: 00000000-1111-2222-3333-444444444444
    organization: 00000000-1111-2222-3333-444444444444
    region: ams1
    commercial_type: START1-S
```

Here are the parameter details for the example shown above:

- `name` is the name of the instance (the one that will show up in your web console).
- `image` is the UUID of the system image you would like to use. A list of all images is available for each availability zone.
- `organization` represents the organization that your account is attached to.
- `region` represents the Availability Zone which your instance is in (for this example, par1 and ams1).
- `commercial_type` represents the name of the commercial offers. You can check out the Scaleway pricing page to find which instance is right for you.

Take a look at this short playbook to see a working example using `scaleway_compute`:

```
- name: Test compute instance lifecycle on a Scaleway account
  hosts: localhost
  gather_facts: no
  environment:
    SCW_API_KEY: ""

  tasks:

    - name: Create a server
      register: server creation_task
      scaleway compute:
        name: foobar
        state: present
        image: 00000000-1111-2222-3333-444444444444
        organization: 00000000-1111-2222-3333-444444444444
        region: ams1
        commercial_type: START1-S
        wait: true
```

```
    - debug: var=server_creation_task

    - assert:
        that:
          - server_creation_task is success
          - server_creation_task is changed

    - name: Run it
      scaleway_compute:
        name: foobar
        state: running
        image: 00000000-1111-2222-3333-444444444444
        organization: 00000000-1111-2222-3333-444444444444
        region: ams1
        commercial_type: START1-S
        wait: true
        tags:
          - web server
      register: server_run_task

    - debug: var=server_run_task

    - assert:
        that:
          - server_run_task is success
          - server_run_task is changed
```

Dynamic Inventory Script

Ansible ships with :ref:`scaleway_inventory`. You can now get a complete inventory of your Scaleway resources through this plugin and filter it on different parameters (`regions` and `tags` are currently supported).

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst`, line 172); *backlink***
>
> Unknown interpreted text role "ref".

Let's create an example! Suppose that we want to get all hosts that got the tag web_server. Create a file named `scaleway_inventory.yml` with the following content:

```
plugin: scaleway
regions:
  - ams1
  - par1
tags:
  - web_server
```

This inventory means that we want all hosts that got the tag `web_server` on the zones `ams1` and `par1`. Once you have configured this file, you can get the information using the following command:

```
$ ansible-inventory --list -i scaleway_inventory.yml
```

The output will be:

```
{
    "_meta": {
        "hostvars": {
            "dd8e3ae9-0c7c-459e-bc7b-aba8bfa1bb8d": {
                "ansible_verbosity": 6,
                "arch": "x86_64",
                "commercial_type": "START1-S",
                "hostname": "foobar",
                "ipv4": "192.0.2.1",
                "organization": "00000000-1111-2222-3333-444444444444",
                "state": "running",
                "tags": [
                    "web_server"
                ]
            }
        }
    },
    "all": {
        "children": [
            "ams1",
            "par1",
            "ungrouped",
            "web_server"
        ]
    },
    "ams1": {},
    "par1": {
```

```
        "hosts": [
            "dd8e3ae9-0c7c-459e-bc7b-aba8bfa1bb8d"
        ]
    },
    "ungrouped": {},
    "web_server": {
        "hosts": [
            "dd8e3ae9-0c7c-459e-bc7b-aba8bfa1bb8d"
        ]
    }
}
```

As you can see, we get different groups of hosts. `par1` and `ams1` are groups based on location. `web_server` is a group based on a tag.

In case a filter parameter is not defined, the plugin supposes all values possible are wanted. This means that for each tag that exists on your Scaleway compute nodes, a group based on each tag will be created.

Scaleway S3 object storage

Object Storage allows you to store any kind of objects (documents, images, videos, and so on). As the Scaleway API is S3 compatible, Ansible supports it natively through the modules: :ref:`s3_bucket_module`, :ref:`aws_s3_module`.

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 249); *backlink***
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\scenario_guides\[ansible-devel][docs][docsite][rst][scenario_guides]guide_scaleway.rst, line 249); *backlink***
>
> Unknown interpreted text role "ref".

You can find many examples in the scaleway_s3 integration tests.

```
- hosts: myserver
  vars:
    scaleway_region: nl-ams
    s3_url: https://s3.nl-ams.scw.cloud
  environment:
    # AWS_ACCESS_KEY matches your scaleway organization id available at https://cloud.scaleway.com/#/accou
    AWS_ACCESS_KEY: 00000000-1111-2222-3333-444444444444
    # AWS SECRET KEY matches a secret token that you can retrieve at https://cloud.scaleway.com/#/credenti
    AWS SECRET KEY: aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee
  module_defaults:
    group/aws:
      s3_url: '{{ s3_url }}'
      region: '{{ scaleway_region }}'
  tasks:
   # use a fact instead of a variable, otherwise template is evaluate each time variable is used
    - set_fact:
        bucket_name: "{{ 99999999 | random | to_uuid }}"

    # "requester pays:" is mandatory because Scaleway doesn't implement related API
    # another way is to use aws_s3 and "mode: create" !
    - s3_bucket:
        name: '{{ bucket_name }}'
        requester_pays:

    - name: Another way to create the bucket
      aws_s3:
        bucket: '{{ bucket_name }}'
        mode: create
        encrypt: false
      register: bucket_creation_check

    - name: add something in the bucket
      aws s3:
        mode: put
        bucket: '{{ bucket_name }}'
        src: /tmp/test.txt  #  needs to be created before
        object: test.txt
        encrypt: false  # server side encryption must be disabled
```