

Overview

Note: This document is about the OpenStack provider. Terraform supports a multitude of others. This document is NOT intended to teach users Hashicorp Terraform, but rather how to create an HPCC cluster in an instant.

HPCC-Terraform allows you to, reliably and quickly, create, change, and improve an HPCC infrastructure in the cloud using Hashicorp Terraform, an open source tool that codifies APIs into declarative configuration files that can be shared amongst team members, treated as code, edited, reviewed, and versioned.

HPCC-Terraform is IaC, infrastructure as code. It allows developers to maximize their productivity and increase transparency.

In this current HPCC-Terraform version, the cloud.init file and volume backup feature are not included. Normally, the cloud.init file plays the same role as the **RC File**, which we use in this version. It allows you to define your provider and set your credentials in order to connect to your provider through its API. That file might be needed if you wish to use a CI / CD Pipeline. In the other hand, the volume backup feature would allow you to back up your volumes so that you can recover from them when necessary.

Infrastructures are built using 3 types of blocks: Provider, Data and Resource blocks.

The **provider block** is used to define the provider and set your credentials.

The **data blocks** are used to query data from your provider.

The **resource blocks** are used to create infrastructures.

All of our Terraform codes are contained in files of this form: **filename.tf**

In addition, a script file is included within the **Root Module** that allows us to provision our infrastructure during init time.

The **RC File** can be downloaded from the OpenStack UI. It contains your OpenStack information with a password prompt.

Terraform comes with an initial module called the **Root Module**. Other modules can be created for better management and abstraction.

The **Root Module** is where you run your Terraform commands, such as `Terraform validate` and so on. It is also the directory Terraform uses to store the state and backup files.

In other words, the **provider block**, **data blocks**, and **resource blocks** are contained in .tf file(s) while **.tf files** are contained in **module(s)**.

Getting Started

Note: These instructions assume that you have access to an OpenStack, and a Linux machine (preferably Ubuntu).

HPCC-Terraform is comprised of a high level of abstraction to allow developers to easily and quickly create highly configurable HPCC clusters without having to worry about what goes behind the scene.

Installing Hashicorp Terraform

Download package from here: `https://www.terraform.io/downloads.html`

Unzip the package: `unzip <package>`

Move Terraform binary to your chosen directory: `mv <binary file> <to this location>`

Add the binary path to the PATH variable: `echo "export PATH=\"$PATH:<binary dir>\"" | sudo tee -a ~/.profile`

Apply changes immediately: `source ~/.profile`

Test Installation:

```
terraform -v
```

HPCC-Terraform Setup

This HPCC-Terraform version has 2 modules: The **Root Module** and what we call the **Abstracted Module**.

The **Root Module** is defined above.

The **Abstracted Module** is the remote module.

The **Root Module** can be found here:

<https://github.com/gfortil/HPCC-OSTRM>

The **Abstracted Module** can be found here:

<https://github.com/gfortil/HPCC-OSTAM>

1. Fork the **Root Module**.

2. Clone your fork on your local machine:

```
git clone https://github.com/<username>/HPCC-OSTRM
```

3. Change directory to HPCC-OSTRM:

```
cd HPCC-OSTRM
```

4. Create a new git branch

5. Copy **main.tf.template** as **main.tf**:

```
cp main.tf.template main.tf
```

6. Modify the **main.tf** file per your specifications (See below for the variable definitions)

7. Download the **RC File** from your OpenStack Account.

- a. Navigate to the upper right corner of your OpenStack account while you are logged in.
- b. Click on the dropdown
- c. Click on the **RC File OpenStack V3** to download it

8. Source the RC File:

```
source <RC File>
```

- a. Enter your password

Run HPCC-Terraform

As you have your main.tf configured and your RC File sourced, now it's time to run HPCC-Terraform.

1. Validate your settings:

```
terraform validate
```

2. Review the plan to be executed:

```
terraform plan
```

3. Apply your settings to create your cluster:

```
terraform apply
```

4. Configure your cluster (See HPCC Admin Documentation)

5. Copy the configured environment.xml to your **Root Module** as backup

6. Version your HPCC-OSTRM
 - a. Add changes to the git stage area
 - b. Commit
 - c. Tag
 - d. Push

Note: It is possible for some of your resources to be failed due to network errors. Running **terraform apply** again should create the rest of the resources that failed.

HPCC Version Upgrade

To upgrade to a new HPCC version, simply:

1. Change directory to HPCC-OSTRM
2. Create a new branch
3. Modify these variables on the main.tf file:

hpcc_version (Required) –String that represent the HPCC version (ex. "7.4.20")

hpcc_release (Required) –String that represents the HPCC release (ex. "rc1" or "1" for gold)

hpcc_package (Required), (Case insensitive) –String that represents the HPCC package (ex. "LN" or "CE")

4. Apply your settings to upgrade your cluster:

terraform apply

5. Version your HPCC-OSTRM
 - a. Add changes to the git stage area
 - b. Commit
 - c. Tag
 - d. Push

Variables

Name	Type	Description	Required
source	string	Path to the HPCC-OSTAM	Yes
hpcc_version	string	HPCC version	Yes
hpcc_release	string	rc number or 1 for gold	Yes
hpcc_package	string	Type of HPCC package. LN or CE	Yes
build_server	string	IP to the server that has the build. Internal use only.	Yes
image_name	string	Name of the OpenStack OS image	Yes
image_id	string	ID of the OpenStack OS image	Yes
key_pair	string	Name of the OpenStack key	Yes
subnet_name	string	Name of the subnet	Yes
network_name	string	Name of the network	Yes
timezone	string	Name of the Linux timezone	No
mydropzone_folder_names	list of string	Names of the folder	No
from_port	number	Open port from this port	No
to_port	number	Close port up to this port	No
device	string	Device to mount to. Default to /dev/vdb	Yes
mountpoint	string	Point to mount the device to. Default to /mnt/vdb	Yes

master_disk	number	Master node disk space in GB	No
slave_disk	number	Slave node disk space in GB	No
esp_disk	number	ESP node disk space in GB	No
backup_disk	number	Support node disk space in GB	No
lzone_disk	number	Landing Zone node disk space in GB	No
dali_disk	number	Dali node disk space in GB	No
master_flavor_name	string	Master node flavor name	Yes
slave_count	number	Number of slave nodes. Set to 0 for none.	No
slave_flavor_name	string	Slave flavor name	Yes
backup_count	number	Number of support nodes	No
backup_flavor_name	string	Support node flavor name	Yes
dali_flavor_name	string	Dali node flavor name	Yes
lzone_flavor_name	string	Landing Zone flavor name	Yes
esp_flavor_name	string	ESP flavor name	Yes