Terraform: Infrastructure as Code

Modules

Code Organization

The Terraform language uses configuration files that are named with the .tf file extension. There is also a JSON-based variant of the language that is named with the .tf.json file extension.

A module is a collection of .tf or .tf.json files kept together in a directory. The root module is built from the configuration files in the current working directory when Terraform is run, and this module may reference child modules in other directories, which can in turn reference other modules, etc.

The simplest Terraform configuration is a single root module containing only a single .tf file.

Modules

Modules help solve the problems:

- **Organize configuration** - Modules make it easier to navigate, understand, and update your configuration by keeping related parts of your configuration together.
- **Encapsulate configuration** - Another benefit of using modules is to encapsulate configuration into distinct logical component.
- **Re-use configuration** - Writing all of your configuration from scratch can be time consuming and error prone.
- **Provide consistency and ensure best practices** - It helps to ensure that best practices are applied across all of your configuration.

Module structure

A typical file structure:

```
$ tree minimal-module/
.
├── LICENSE # the license under which your module will be distributed.
├── README.md # documentation
├── main.tf # the main set of configuration
├── variables.tf # variable definitions
├── outputs.tf # output definitions
```

`gitignore`:

- `terraform.tfstate` and `terraform.tfstate.backup`: Terraform state
- `.terraform`: modules and plugins

Modules: nested

```
$ tree complete-module/
.
├── README.md
├── main.tf
├── variables.tf
├── outputs.tf
├── ...
├── modules/
│   ├── nestedA/
│   │   ├── README.md
│   │   ├── variables.tf
│   │   ├── main.tf
│   │   ├── outputs.tf
│   ├── nestedB/
│   ├── .../
```

Modules: Local

```
module "consul" {
  source = "./consul"
}
```

Modules: Registry

Registry source address: `<NAMESPACE>/<NAME>/<PROVIDER>`

```
module "consul" {
  source = "hashicorp/consul/aws"
  version = "0.1.0"
}
```

Modules: GitHub

```
module "consul" {
  source = "github.com/hashicorp/example"
}
```

Modules: Git

```
module "vpc" {
  source = "git::https://example.com/vpc.git"
}

module "storage" {
  source = "git::ssh://username@example.com/storage.git"
}
```

terraform-aws-modules

```
$ cat terraform-aws-modules/terraform-aws-vpc/main.tf

######
# VPC
######
resource "aws_vpc" "this" {
  count = var.create_vpc ? 1 : 0

  cidr_block                       = var.cidr
  instance_tenancy                 = var.instance_tenancy
  enable_dns_hostnames             = var.enable_dns_hostnames
  enable_dns_support               = var.enable_dns_support
  enable_classiclink               = var.enable_classiclink
  enable_classiclink_dns_support   = var.enable_classiclink_dns_support
  assign_generated_ipv6_cidr_block = var.enable_ipv6

  tags = merge(
    {
      "Name" = format("%s", var.name)
    },
    var.tags,
    var.vpc_tags,
  )
}
```

terraform-aws-modules: example

`main.tf`:

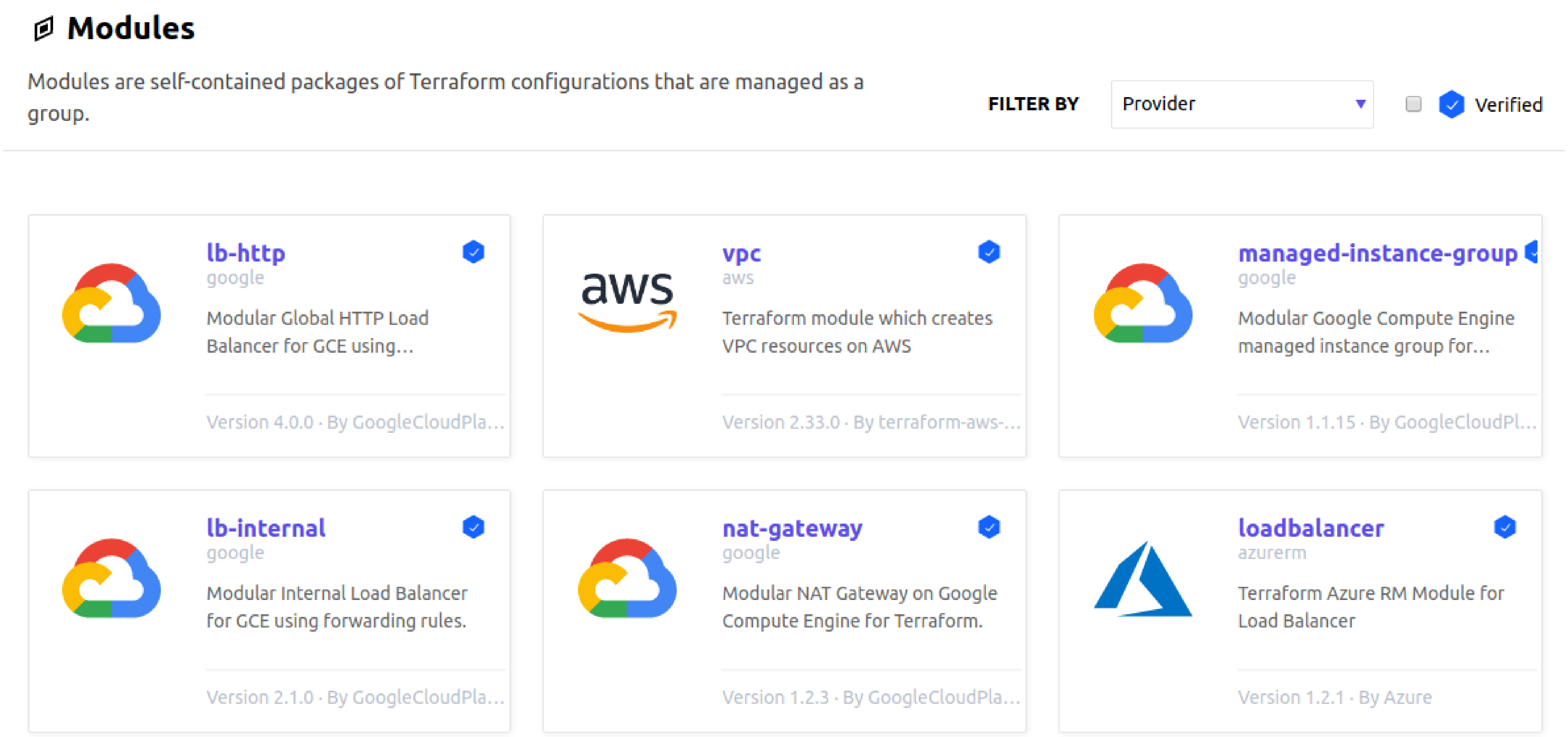
```
provider "aws" {
  region = "eu-west-1"
}

module "vpc" {
  source  = "terraform-aws-modules/vpc/aws"
  version = "2.33.0"

  create_vpc = false

  manage_default_vpc               = true
  default_vpc_name                 = "default"
  default_vpc_enable_dns_hostnames = true
}
```

Registry: Modules



Modules

Modules are self-contained packages of Terraform configurations that are managed as a group.

FILTER BY Provider ▾ □ ✓ Verified

lb-http ✓
google
Modular Global HTTP Load Balancer for GCE using…
Version 4.0.0 · By GoogleCloudPla…

vpc ✓
aws
Terraform module which creates VPC resources on AWS
Version 2.33.0 · By terraform-aws-…

managed-instance-group
google
Modular Google Compute Engine managed instance group for…
Version 1.1.15 · By GoogleCloudPl…

lb-internal ✓
google
Modular Internal Load Balancer for GCE using forwarding rules.
Version 2.1.0 · By GoogleCloudPla…

nat-gateway ✓
google
Modular NAT Gateway on Google Compute Engine for Terraform.
Version 1.2.3 · By GoogleCloudPla…

loadbalancer ✓
azurerm
Terraform Azure RM Module for Load Balancer
Version 1.2.1 · By Azure

Registry: Requirements

- GitHub. The module must be on GitHub and must be a public repo
- Named `terraform-<PROVIDER>-<NAME>`
- Repository description
- Standard module structure. The module must adhere to the standard module structure
- x.y.z tags for releases