Terraform

Date de création	@25 janvier 2024 00:42
Créée par	Alexon
Type	Information

Terraform

Terraform is an open-source, Infrastructure-as-Code tool that helps you manage your infrastructure at any time, deploy it/delete it in just one click, and work with other developers on your projects.

A common pattern is to use Terraform to set up base infrastructure, including networking, VM instances, and other foundational resources.

How to organize your files properly

Now we are going to create four different files:

- main.tf: will contain the main set of configurations for your project. Here, it will be our instance
- provider.tf: Terraform relies on plugins called “providers” to interact with remote systems
- backend.tf: each Terraform configuration can specify a backend, which defines where the state file of the current infrastructure will be stored. Thanks to this file, Terraform keeps track of the managed resources. This state can be stored locally or remotely. Configuring a remote backend allows multiple people to work on the same infrastructure
- variables.tf: will contain the variable definitions for your project. Since all Terraform values must be defined, any variables that are not given a default value will become required arguments
- terraform.tfvars: allows you to set the actual value of the variables

We have the choice between local vs remote backend

Terraform will create terraform.tfstate and terraform.tfstate.backup: these files contain the Terraform state specific to a specific environment. Thanks to this file, Terraform keeps track of the managed resources. This state can be stored locally or remotely.

Managing Terraform state files is crucial for infrastructure as code (IaC), and the choice between local and remote state storage is an important consideration. Storing Terraform state files in Git (or any version control system) is generally discouraged for several reasons:

Security Risks: These files may contain sensitive information stored in plain text from the previous deployment, such as passwords, secret keys, and other secrets. Storing these files in Git, especially public or shared repositories, can expose sensitive information to unauthorized users.

Team Work : The combinaison of a backend + a remote state! Where local state is great for an isolated developer, remote state is absolutely necessary for a team, as each member will need to share the infrastructure state whenever there is a change.

In our case, we will set up an S3 bucket to store our backend.

Manage Different Environments By Terraform

We can use Terraform Workspace and Each workspace maintains its own state, allowing you to manage different environments (like staging, production) within the same Terraform configuration.

```
terraform init
terraform workspace new testing
terraform workspace new staging
terraform workspace new production
terraform workspace list
terraform workspace select testing
```

In your bucket s3, it will create your **terraform.state** file inside the folder like **env:/testing**

More about Terraform Workspaces

Sources

Terraform Registry

https://registry.terraform.io/providers/scaleway/scaleway/latest/docs

Terraform: how to init your infrastructure

If you want to quickly and easily set up a cloud infrastructure, one of the best ways to do it is to create a Terraform repository. Learn the basics to start your infrastructure on Terraform.

https://www.scaleway.com/en/blog/terraform-how-to-init-your-infrastructure/

GitHub - julesmartin76000/Scaleway_how_to_terraform

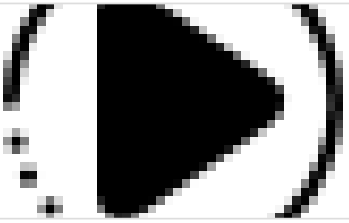
Contribute to julesmartin76000/Scaleway_how_to_terraform development by creating an account on GitHub.

https://github.com/julesmartin76000/Scaleway_how_to_terraform/tree/master

Scaleway + Terraform = ❤

Datatask a fait le choix de porter sa plateforme sur le cloud provider Scaleway. Au delà de nous permettre de satisfaire à des contraintes fortes de ...

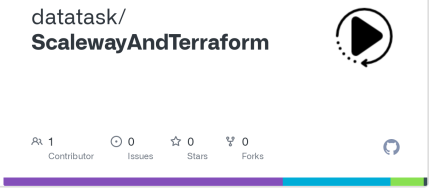
https://datatask.io/blog/scaleway-terraform/



GitHub - datatask/ScalewayAndTerraform

Contribute to datatask/ScalewayAndTerraform development by creating an account on GitHub.

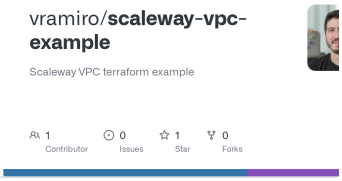
https://github.com/datatask/ScalewayAndTerraform/tree/main



GitHub - vramiro/scaleway-vpc-example: Scaleway VPC terraform example

Scaleway VPC terraform example. Contribute to vramiro/scaleway-vpc-example development by creating an account on GitHub.

https://github.com/vramiro/scaleway-vpc-example/tree/main



20 Terraform Best Practices to Improve your TF workflow

Learn some best practices that will assist you in pushing your Terraform skills to the next level. See how they can make your IaC management easier.

https://spacelift.io/blog/terraform-best-practices

Using Terraform and Ansible Together

In this tutorial, you'll learn how to use Terraform and Ansible together. See how Spacelift can greatly simplify and elevate your workflow for both tools.

https://spacelift.io/blog/using-terraform-and-ansible-together