

TP 3 — Modules locaux et boucles

Terraform

Objectif : créer un module, réutiliser du code, gérer une liste d'éléments avec `for_each`.

Provider utilisé : `local`.

1. Préparation

```
mkdir tp2
cd tp2
```

2. Créer un module `file_generator`

Créer l'arborescence :

```
tp2/
  main.tf
  modules/
    file_generator/
      main.tf
      variables.tf
      outputs.tf
```

3. Contenu du module

variables.tf

```
variable "files" {
  type          = map(string)
  description = "Map de nom -> contenu"
}
```

main.tf

```
resource "local_file" "generated" {
  for_each = var.files

  filename = "${path.module}/${each.key}.txt"
```

```
    content = each.value
}
```

outputs.tf

```
output "generated_files" {
  value = [for f in local_file.generated : f.filename]
}
```

4. Utiliser le module dans main.tf

```
terraform {
  required_providers {
    local = {
      source = "hashicorp/local"
    }
  }
}

provider "local" {}

module "docs" {
  source = "../modules/file_generator"

  files = {
    "hello" = "Bonjour depuis Terraform !"
    "config" = "param1 = true\nparam2 = false"
    "user"   = "admin"
  }
}

output "fichiers" {
  value = module.docs.generated_files
}
```

5. Workflow Terraform

```
terraform init
terraform plan
terraform apply -auto-approve
```

6. Vérification

```
ls modules/file_generator
```

7. Nettoyage

```
terraform destroy -auto-approve
```