
HashiCorp
Terraform
NOTIONS AVANCÉES
Par ULRICH MONJI

PLAN

- Rappel sur les bases
- Gestion des providers
- Count et Count Index
- Les expressions conditionnelles
- Les variables locales
- Les fonctions Terraform
- Les blocs dynamiques
- Tainting de ressource
- Plan file
- Terraform import
- Gestion du multi-environnement avec le workspace
- Terraform Cloud

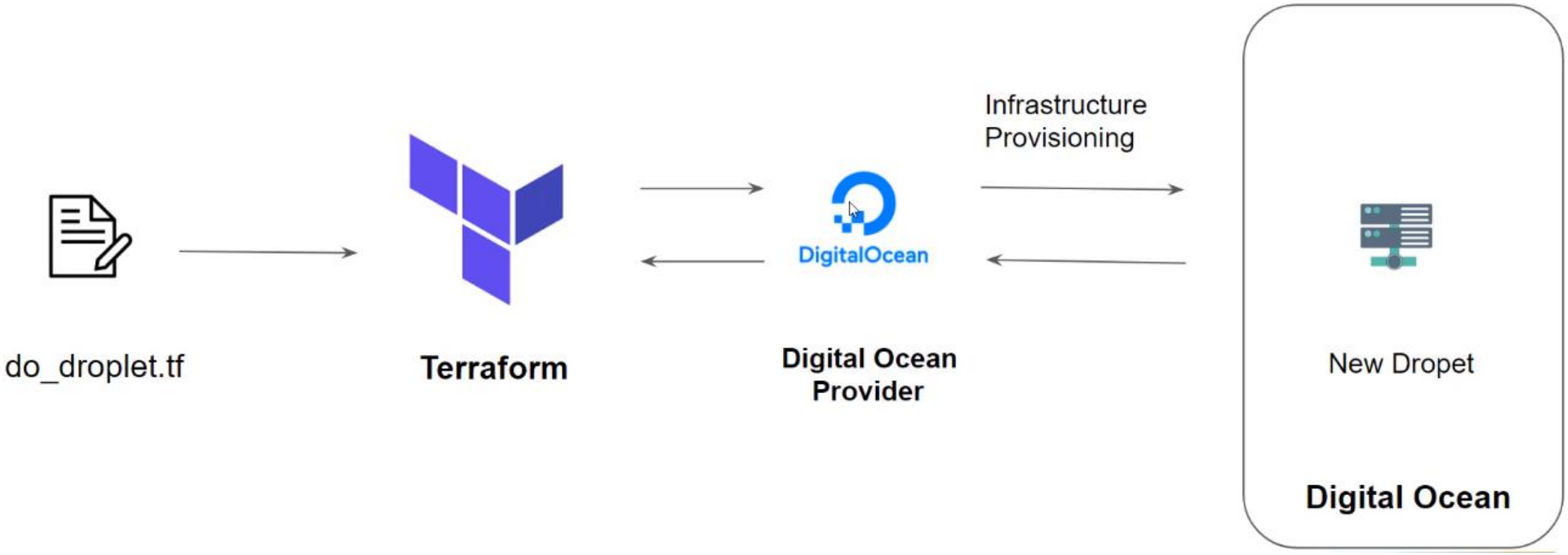
Prise en main de l’écosysteme de Tp

Lab 0

Révision: Création d’un module simple

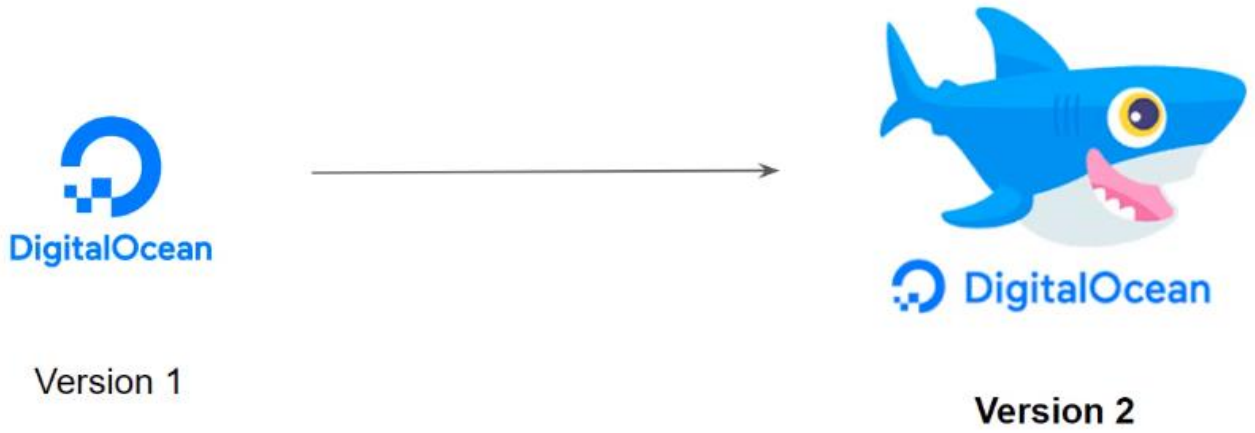
Lab 1

Gestion des providers: rôle



Gestion des providers: Versioning

- Les providers sont versionné indépendamment des versions de Terraform
- Chaque provider possède plusieurs versions de son plugin



Gestion des providers : Versioning

Version Number Arguments	Description
>=1.0	Greater than equal to the version
<=1.0	Less than equal to the version
~>2.0	Any version in the 2.X range.
>=2.10,<=2.30	Any version between 2.10 and 2.30

```
provider "aws" {
  region     = "us-west-2"
  version    = "2.7"
}
```

Gestion des providers

Lab 2

Count et Count Index

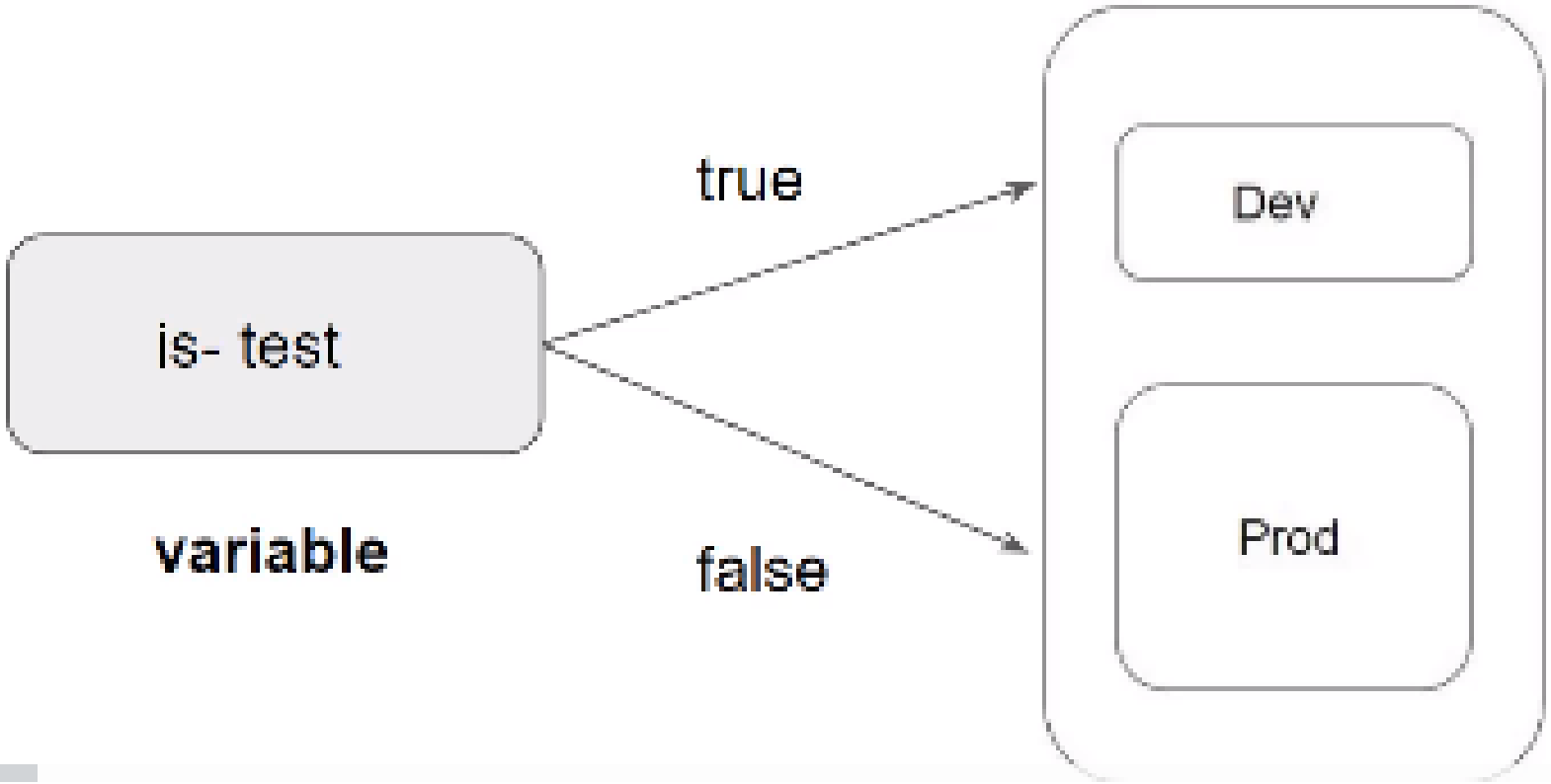
```
resource "aws_iam_user" "lb" {
  name = "loadbalancer.${count.index}"
  count = 5
  path = "/system/"
}
```

- Réduire la quantité de ligne de code
- Regrouper les actions similaires
- Rendre son code plus lisible

Count et Count Index

Lab 3

Expressions conditionnelles



Expressions conditionnelles

Lab 4

Local Values

```
locals {
  common_tags = {
    Owner = "DevOps Team"
    service = "backend"
  }
}
```

```
resource "aws_instance" "app-dev" {
  ami = "ami-082b5a644766e0e6f"
  instance_type = "t2.micro"
  tags = local.common_tags
}
```

```
resource "aws_ebs_volume" "db_ebs" {
  availability_zone = "us-west-2a"
  size              = 8
  tags = local.common_tags
}
```

```
locals {
  name_prefix = "${var.name != "" ? var.name : var.default}"
}
```

Local Values

Lab 5

Fonctions Terraform

- Function (argument1, argument2)
- On ne peut pas définir nos propres fonctions
- Terraform console

- Numeric
- String
- Collection
- Encoding
- Filesystem
- Date and Time
- Hash and Crypto
- IP Network
- Type Conversion

Fonctions Terraform

Lab 6

Dynamic Bloc

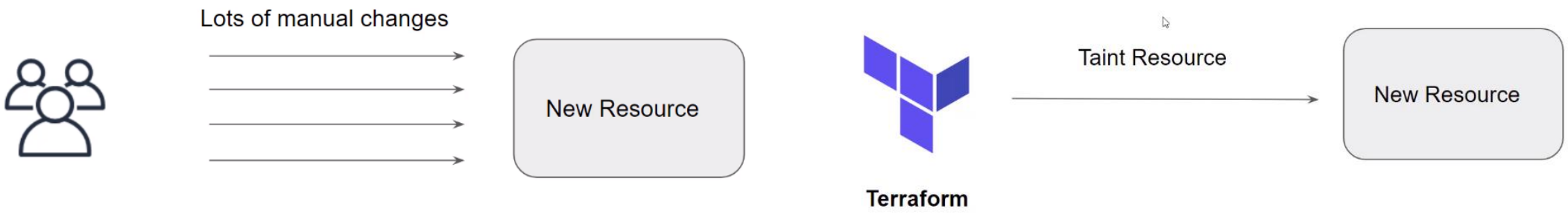
- Utile lorsque vous souhaitez répéter un bloc dans une section
- Permet de réduire la quantité de code
- Facilité la lisibilité de votre code

```
variable "sg_ports" {
  type        = list(number)
  description = "list of ingress ports"
  default     = [8200, 8201,8300, 9200, 9500]
}

resource "aws_security_group" "dynamicsg" {
  name        = "dynamic-sg"
  description = "Ingress for Vault"

  dynamic "ingress" {
    for_each = var.sg_ports
    content {
      from_port   = ingress.value
      to_port     = ingress.value
      protocol    = "tcp"
      cidr_blocks = ["0.0.0.0/0"]
    }
  }
```

Dynamic Bloc

Lab 7

Tainting de resources



Tainting de resources

Lab 8

Terraform plan file

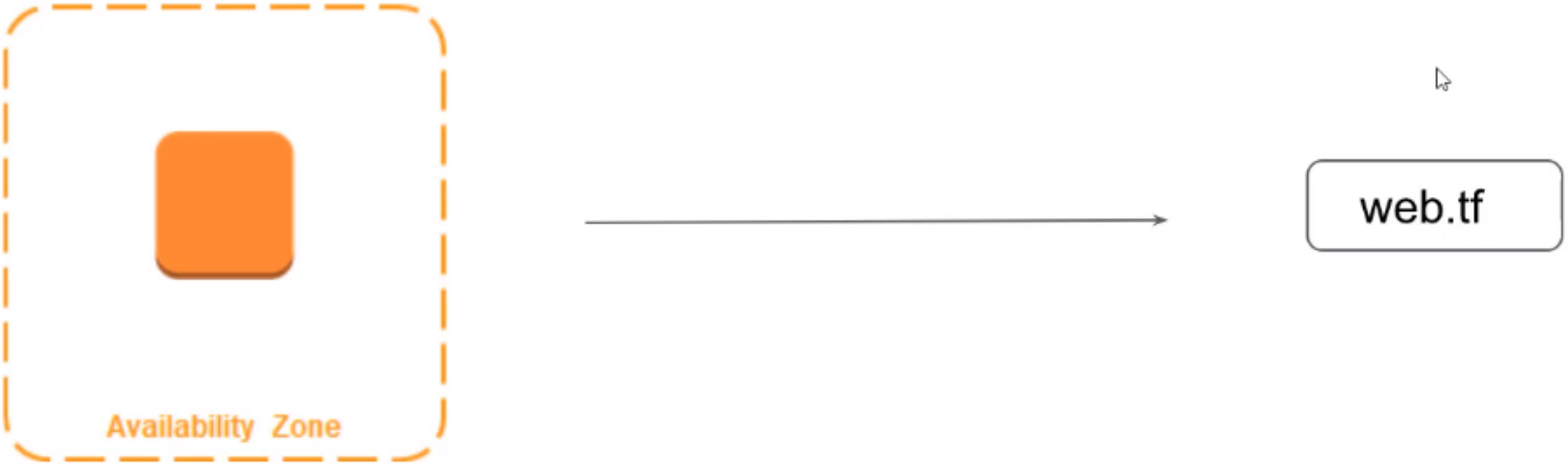
- terraform plan –out=<path>
- Terraform apply <path>

Terraform plan file

Lab 9

Terraform Import

- Rajouter une ressource créée manuellement dans terraform
- Pour importer il faut renseigner les caractéristiques de la VM dans un fichier .tf
- Pour finaliser l'importation il faut fournir l'id de la ressource à importer



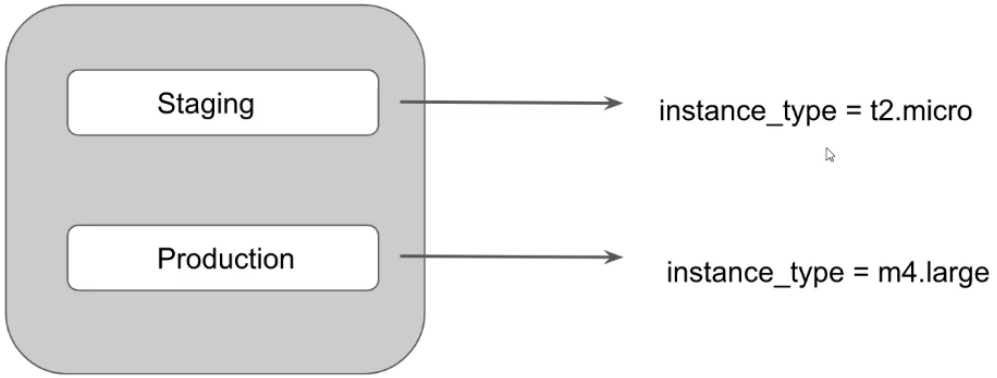
Terraform Import

Lab 10

Terraform Workspace

- Terraform workspace
- Terraform workspace show
- Il permet d'avoir plusieurs environnements de travail sur la même machine (staging, production)

```
provider "aws" {
  region     = "us-west-2"
  access_key = "YOUR-ACCESS-KEY"
  secret_key = "YOUR-SECRET-KEY"
}

resource "aws_instance" "myec2" {
   ami = "ami-082b5a644766e0e6f"
   instance_type = lookup(var.instance_type,terraform.workspace)
}

variable "instance_type" {
  type = "map"

  default = {
    default = "t2.nano"
    dev     = "t2.micro"
    prd     = "t2.large"
  }
}
```



Terraform Workspace

Lab 11

Terraform Multiregion

eip.tf

```
resource "aws_eip" "myeip" {
  vpc = "true"
}

resource "aws_eip" "myeip01" {
  vpc = "true"
  provider = "aws.aws02"
}
```

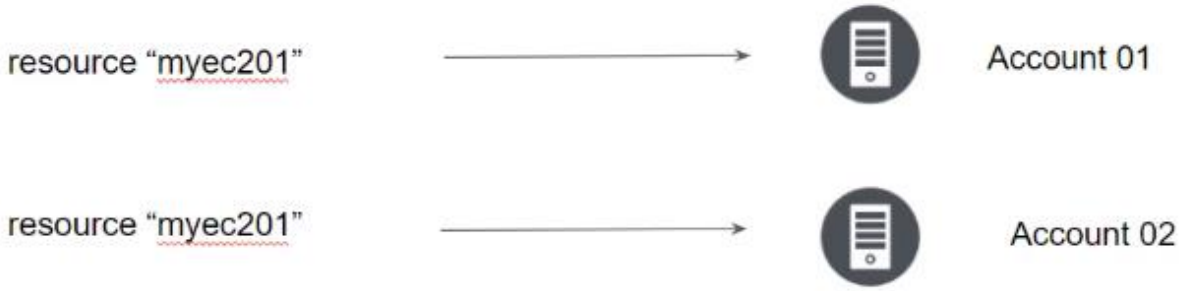
1st EIP -- one region

2nd EIP -- second region

providers.tf

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

provider "aws" {
  alias      =  "aws02"
  region     =  "ap-south-1"
  profile    =  "account02"
}
```

Multiple profile




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

provider "aws" {
  alias      =  "aws02"
  region     =  "ap-south-1"
  profile    =  "account02"
}
```

```
resource "aws_eip" "myeip" {
  vpc = "true"
}

resource "aws_eip" "myeip01" {
  vpc = "true"
  provider = "aws.aws02"
}
```

Terraform Multiregion:

Lab 12

Terraform Cloud Présentation

- Terraform managé
- Access control
- Private registry
- Policy controls
- Evaluation des coûts
- Et plus encore

Terraform Cloud

Lab 13

Merci pour votre attention !