Chapter 5: Terraform tips and tricks

loops

count

- usage:

```
resource "blablabla" "blablabla_name" {
count = 3
name  = "bla${count.index}"
}
```

- result: 3 elements ([0],[1],[2]) → resources array
- with `length()`:

```
variable "bla_names" {
  type    = list(string)
  default = ["bla_green","bla_yellow","bla_red"]
}
resource "blablabla" "blablabla_name" {
  count = length(var.bla_names)
  name  = bla_names[count.index]
}
```

- output
  - **[#]**: use cardinal to access a concrete element
  - **[*]**: all elements from resource list
- each element is assigned to their index in state, deletes o modifications implies «move» all elements after that.
- only usable for **resources**, no inline blocks

for_each

- lists (using **toset()** function), sets and maps
- for use in **resources** o **inline blocks**
- use **each.key** and **each.value**
- usage:

```
variable "bla_names" {
  type    = list(string)
  default = ["bla_green","bla_yellow","bla_red"]
}
resource "blablabla" "blablabla_name" {
for_each = toset(var.bla_names)
name  = each.value
}
```

```
variable "bla_names" {
  type    = map(string)
  default = {"green" = "bla_green","yellow" = "bla_yellow","red" = "bla_red"}
```

```
  #default = map("green","bla_green", "yellow","bla_yellow", "red","bla_red")
}
resource "blablabla" "blablabla_name" {
for_each = toset(var.bla_names)
name  = each.value
}
```

- **output**:
    - values(<resource_name>)[*].attribute

for

for string

conditionals

> **Pendiente**
> Pos 3500 aprox

count

```
variable "booleana" {
  type = bool
}
resource "resource" "resource_name" {
  count = var.booleana ? 1 : 0
  ...
}
variable "cadena" {
  type = string
  default = "aaa"
}
resource "resource" "resource_name {
  count = format("%.1s",var.cadena) == "a" ? 1 : 0
}
```

for_each

if string

Zero-Downtime Deployment

terraform gotchas

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/info:libros:terraform-upandrunning:cap5

Last update: **24/05/2020 00:06**