**MODULE - 01 INTRODUCTION**

Use same project for deploy on different cloud providers (resources on AWS and Azure).

Plan before applying changes (what if) – not possible with AWS Cloudformation.

**MODULE - 01 CONFIGURING TERRAFORM WITH AWS**

Create AWS account (Free tier).

Create TerraformUser on IAM Users with full access.

Install Terraform (<https://www.terraform.io/>) and configure System variables.

Create System Variable for AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY.

**MODULE - 02 CREATING AN AWS RESOURCE WITH TERRAFORM**

Code samples: <https://github.com/kevholditch/terraform-course-examples>

Sample code to create S3 bucket

Go to sample code folder (main.tf file) and Terraform init

Terraform plan

Terraform apply

Created S3 bucket rodolfomarra-myfirst-bucket on us-east-1

Terraform destroy

Deleted S3 bucket rodolfomarra-myfirst-bucket on us-east-1

**MODULE – 02 TERRAFORM RESOURCES**

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs>

Resources have arguments: required and optional

Resources could have Exported Attributes (ie. [ARN returned after having created S3 bucket](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/s3_bucket#attributes-reference))

Interpolation Syntax: replace information generated after creating AWS object (ie. pass ARN from S3 Bucket to IAM Role).

Interpolation syntax format:

“${<resource\_type>.<resource\_name>.<exported\_attribute>}”

Example:

**resource** “aws\_s3\_bucket” “my\_bucket” {

bucket = “rodolfomarra-myfirst-bucket”

}

Reference using: “$(aws\_s3\_bucket.my\_bucket.arn}”

Data types for resources attributes:

Int – defined using – port = 21

String – defined using – host = “localhost”

List – defined using – security\_groups = [“abc”,”def”]

Bool – defined using – enabled = false

Sample code to create S3 bucket and IAM Policy to list Bucket

Go to sample code folder (main.tf file) and Terraform init

Terraform apply

Created S3 bucket rodolfomarra-myfirst-bucket on us-east-1

Created IAM Policy my-bucket-policy on us-east-1

Terraform destroy

Deleted S3 bucket rodolfomarra-myfirst-bucket on us-east-1

Deleted IAM Policy my-bucket-policy on us-east-1

**MODULE – 03 TERRAFORM DATA SOURCES**

Data sources allow to reference objects from external Terraform projects or to resources already created on AWS

This will return many attributes from AWS resources to be reused on TF. To reuse attributes, you will need to use:

${**data**.<resource\_type>.<resource\_name>.<exported\_attribute>}

Example:

**data** “aws\_s3\_bucket” “my\_bucket” {

bucket = “rodolfomarra-myfirst-bucket”

}

Reference using: “$(**data**.aws\_s3\_bucket.my\_bucket.arn}”

Created S3 bucket rodolfomarra-already-here in us-east-1 **manually**

Sample code to reference S3 bucket that already exist and create IAM Policy to list Bucket

Go to sample code folder (main.tf file) and Terraform init

Terraform apply

Created IAM Policy my-bucket-policy on us-east-1

Terraform destroy

Deleted IAM Policy my-bucket-policy on us-east-1

Deleted S3 bucket rodolfomarra-already-here on us-east-1 **manually**

**MODULE – 04 TERRAFORM LOCALS**

Locals allow to assign a name to an expression (like variable)

Example of single local:

locals {

bucket\_name\_prefix = “rody-”

default\_instance\_tag = “my-instance”

}

Example of multiples locals:

locals {

bucket\_name\_prefix = “rody-”

}

locals {

default\_instance\_tag = “my-instance”

}

To reference locals you need to use the interpolation syntax:

“${local.<variable\_name>}”

Example:

locals {

bucket\_name\_prefix = “rody-”

default\_instance\_tag = “my-instance”

}

To reference bucket\_name\_prefix, you will need to use: “${**local**.bucket\_name\_prefix}”

Locals values can be combined to make more local values, example:

locals {

first = “rody”

last = “marra”

name = “$(local.first}-${local.last}”

}

The result of local.name will be “rody-marra”.

Locals can be a value of exported attribute, example:

resource “aws\_s3\_bucket” “my\_bucket” {

bucket = “rodolfomarra-myfirst-bucket”

}

locals {

bucket\_arn = “${aws\_s3\_bucket.my\_bucket.arn}”

}

Sample code to create S3 bucket using locals as S3 bucket name prefix

Go to sample code folder (main.tf file) and Terraform init

Terraform apply

Created S3 bucket rodolfomarra-myfirst-bucket on us-east-1

Modified main.tf changing local used to S3 bucket name prefix (from rodolfomarra to rodolfosoares)

Terraform apply

Deleted S3 bucket rodolfomarra-myfirst-bucket on us-east-1

Created S3 bucket rodolfosoares-myfirst-bucket on us-east-1

Terraform destroy

Deleted S3 bucket rodolfosoares-myfirst-bucket on us-east-1

**MODULE – 05 TERRAFORM OUTPUTS**

Tell TF which values are important so TF can output them on screen when we run “Terraform apply”

Example:

output “my\_value” {

value = “hello kevin”

}

Will give the output: my\_value = hello kevin

Output can be result of expressions:

output “my\_value” {

value = “${aws\_s3\_bucket.my\_bucket.arn}”

}

Output can be values of locals:

output “my\_value” {

value = “${local.bucket\_name}”

}

Outputs can be used to return values from a module

Sample code to print on screen a value of local

**MODULE – 06 TERRAFORM TEMPLATES AND FILES**

Terraform allow to use a file as a parameter to a resource (ie. a block of JSON for IAM Policy)

To use files, you will need to use: “${file(“<path\_to\_file>”)}”

Example:

resource “aws\_iam\_user\_policy” “my\_bucket\_policy” {

name = “my-bucket-policy”

user = “rodolfo-marra”

policy = “${file(“policy.json”)}”

}

<https://registry.terraform.io/providers/hashicorp/template/latest/docs>

To replace some value (ie."${bucket\_arn}") inside some TF template files (ie. policy.json) you need to use:

data "template\_file" "bucket\_policy" {

template = "${file("**policy.json**")}"

vars {

**bucket\_arn = "${aws\_s3\_bucket.my\_bucket.arn}"**

}

}

Every line on **policy.json** that is **"${bucket\_arn}"** will be replaced by S3 Bucket ARN.

To get and use the result of this replacement, you need to use: “${**data.template\_file.**<name>.**rendered**}”

Example:

“${**data**.**template\_file**.**bucket\_policy**.rendered}”

Sample code to create S3 bucket, IAM User and IAM Policy using external JSON template rendered with S3 bucket ARN

Go to sample code folder (main.tf file) and Terraform init

Terraform apply

Created S3 bucket rodolfomarra-bucket on us-east-1

Created IAM user Rodolfo-Marra on us-east-1

Created IAM User Policy my-policy on us-east-1

Terraform destroy

Deleted S3 bucket rodolfomarra-bucket on us-east-1

Deleted IAM user Rodolfo-Marra on us-east-1

Deleted IAM User Policy my-policy on us-east-1

**MODULE – 07 TERRAFORM PROVIDERS**

Terraform provider enables TF to talk to an API to manage resources (ie. aws, azure, googlecloud, etc).

Allow to manage resources on multiples clouds (or multiple regions of same cloud provider) in the single project.

You can specify the region and also access\_key and secret\_key when you define a provider (but is not recommended because it will be available on code), example:

provider "aws" {

region = "ca-central-1"

alias = "canada"

access\_key = “AAAAA”

secret\_key = “aASdasg”

}

We can pin a provider to certain version or add version requirements for a provider, example:

provider "aws" {

region = "ca-central-1"

alias = "canada"

version = “1.8”

}

provider "aws" {

region = "ca-central-1"

alias = "canada"

version = “~> 1.8”

}

<https://www.terraform.io/docs/configuration/providers.html>

If we don’t specify a provider on resource creation, default provider will be used

To specify another provider for specific resources, need to use: <provider>.<alias>

resource aws\_s3\_bucket "canada\_bucket" {

bucket = "rodolfomarra-canada"

**provider** = "**aws**.**canada**"

}

Sample code to create S3 bucket using default provider and another provider

Go to sample code folder (main.tf file) and Terraform init

Terraform apply

Created S3 bucket rodolfomarra-us-bucket on us-east-1

Created S3 bucket rodolfomarra-ca-bucket on ca-central-1

Terraform destroy

Deleted S3 bucket rodolfomarra-us-bucket on us-east-1

Deleted S3 bucket rodolfomarra-ca-bucket on ca-central-1

**MODULE – 08 Terraform Variables**

Variables serve as parameters to a TF module

When used at top level they enable to pass parameters into TF project

Properties of variables:

Type (optional) – string, list or map

Default (optional) – Default value

Description (optional) – Variable description

Default variables need to be a String (ie. Foo) or List (ie. [“a”,”b”]). Can’t be interpolation syntax.

Examples of variable:

variable “key” {

type = “string”

}

variable “instance\_size\_map” {

type = “map”

default = {

dev = “t2.micro”

test = “t2.medium”

prod = “m4.large”

}

}

variable “zones” {

type = “list”

default = [“us-east-1a”,”us-east-1b”]

}

Usage: sometimes we cannot store secret, password, access\_key externally from source codes. So for it, we use external variables with .tfvars files.