=================

Terraform modules

=================

Terraform module is a set of configuration files available in a single directory

Using terraforms modules we can achieve re-usability

We will run terraform commands from root module and root module will invoke child modules for execution.

One module can contain one or more .tf files

01-Project

- main.tf

- inputs.tf

- outputs.tf

One module can have any no of child modules in terraform

vinod-ses

- ec2

- main.tf

- input.tf

- output.tf

- s3

- main.tf

- input.tf

- output.tf

- privider.tf

- input.tf

- output.tf

======================================

Terraform project setup with Modules

======================================

Create a sub directory or a child module inside "05\_Terraform\_Project" & name it as "modules" - This is an optional directory/child module creation.

Under "modules" directory create 2 more child modules - ec2 & s3

Under each child module create some config files along with root module config files

step-1

Create a project directory (root module)

eg: 05-app

Step-2

create a "module" directory inside the root module or project directory

eg: 05-app

- modules

Step-3

Inside the modules directory create a child module like ec2 and s3 directory

eg: 05-app

- modules

- ec2

- s3

Step-4

Create terraform scripts inside the ec2 directory

eg: 05-app

- modules

- ec2

input.tf

main.tf

output.tf

- s3

Step-5

Create terraform scripts inside the s3 directory

eg: 05-app

- modules

- ec2

input.tf

main.tf

output.tf

- s3

input.tf

main.tf

output.tf

Step-6

create "provider.tf" file in the root module,

create "main.tf" file in the root module and invoke the child module from the root module

whatever the output is available that I want to get from the output module for that i need to create "output.tf"

eg: 05-app

- modules

- ec2

input.tf

main.tf

output.tf

- s3

input.tf

main.tf

output.tf

- main.tf

- provider.tf

- output.tf

===============================

first configure the provider.tf

===============================

# provider.tf

provider "aws" {

region = "ap-south-1"

access\_key = "AKIAWOAVSUU7MZO24GGV"

secret\_key = "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"

}

=============================================

Now we need to write the ec2 module input.tf

=============================================

#input.tf

variable "ami" {

description = "amazon machine image"

default = "ami-0fd05997b4dff7aac"

}

variable "instance\_type" {

description = "what is the instance type"

default = "t2.micro"

}

variable "key\_name" {

description = "my keypair name"

default = "terraformkeypair"

}

============================================

Now we need to write the ec2 module main.tf

============================================

# main.tf

resource "aws\_instance" "scopedindiavm" {

ami = "${var.ami}"

instance\_type = "${var.instance\_type}"

key\_name = "${var.key\_name}"

security\_groups = ["default"]

tags = {

Name = "linuxvm"

}

}

==============================================

Now we need to write the ec2 module output.tf

==============================================

# output.tf

output "ec2\_vm\_public\_ip" {

value = aws\_instance.scopedindiavm.public\_ip

}

output "ec2\_vm\_private\_ip" {

value = aws\_instance.scopedindiavm.private\_ip

}

output "ec2\_vm\_subnet\_id" {

value = aws\_instance.scopedindiavm.subnet\_id

}

============================================

Now we need to write the root module main.tf

============================================

# main.tf

module "my-ec2" {

source = "./modules/ec2"

}

module "my-s3" {

source = "./modules/s3"

}

==============================================

Now we need to write the root module output.tf

==============================================

# output.tf

output "ec2\_vm\_public\_ip" {

value = module.my-ec2.ec2\_vm\_public\_ip

}

output "ec2\_vm\_private\_ip" {

value = module.my-ec2.ec2\_vm\_private\_ip

}

output "ec2\_vm\_subnet\_id" {

value = module.my-ec2.ec2\_vm\_subnet\_id

}

=================================================

Now we need to write the s3 child module input.tf

=================================================

# input.tf

variable "bucket\_name" {

default = "scopeindias3bucket1"

}

=================================================

Now we need to write the s3 child module main.tf

=================================================

# main.tf

resource "aws\_s3\_bucket" "scopeindia\_s3\_bucket" {

bucket = "${var.bucket\_name}"

versioning {

enabled = true

}

}