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

Workspace in terraform

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

To manage infrastructure for multiple environment we will use terraform workspace concept

When we use workspace concept it will maintain separate state file for every environment

We can execute the same script for multiple environments

# terraform workspace show

# terraform workspace new dev

# terraform workspace new sit

# terraform workspace new uat

# terraform workspace new pilot

# terraform workspace new prod

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

Working with terraform workspace

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

Step-1: Create Terraform Project

Step-2: Create provider.tf file and configure provider details

Step-3: Create input variables files based on environments and configure variable values.

Ex :

dev.tfvars

qa.tfvars

prod.tfvars

Note: Both tfvars and tf are same. any name can be used with tf extension

Step-4 : Create main resources script file

Step-5 : Create outputs variable file

Step-6 : Create Workspaces for each environment

$ terraform workspace new dev

$ terraform workspace new sit

$ terraform workspace new uat

$ terraform workspace new pilot

$ terraform workspace new prod

Step-7 : Select workspace

$ terraform workspace select dev

Step-8 : Run script and check state files for dev env

$ terraform apply --var-file=dev.tfvars

Step-9 : switch to sit workspace and run the script

$ terraform workspace select sit

Step-10 : Run script and check state files for sit env

$ terraform apply --var-file=sit.tfvars

=======

main.tf

=======

variable "instance\_type" {

description = "it represents the type of instance"

}

resource "aws\_instance" "scopedindiavm" {

ami = "ami-0fd05997b4dff7aac"

instance\_type = var.instance\_type

key\_name = "terraformkeypair"

security\_groups = ["default"]

tags = {

Name = "linuxvmtvm"

}

}

===========

provider.tf

===========

provider "aws" {

region = "ap-south-1"

access\_key = "AKIAWOAVSUU7MZO24GGV"

secret\_key = "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"

}

==========

output.tf

==========

output "ec2\_public\_ip" {

value = aws\_instance.scopedindiavm.public\_ip

}

==========

dev.tfvars

==========

instance\_type = "t2.micro"

==========

sit.tfvars

==========

instance\_type = "t2.micro"

===========

uat.tfvars

===========

instance\_type = "t2.micro"

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

pilot.tfvars

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

instance\_type = "t2.medium"

===========

prod.tfvars

===========

instance\_type = "t2.xlarge"

# terraform apply --var-file=sit.tfvars

When we run the apply command a separate tfstate file will be created for sit environment

to track the sit environment. similarly when we run apply command using pro.tfvars file another tfstate file will be generated

So workspace feature in terraform will keep separate tfstate for tracking all of our environments