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

In terraform we have two types of variables are available

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

1 Input variables - Input variables are used to supply the values to the script at the runtime

2 Output variables - Output variables are used to get the values from terraform script after script execution

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

Input variable terraform script

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

# input\_vars.tf

variable "ami" {

description = "Amazon machine image id"

default = "ami-0e53db6fd757e38c7"

}

variable "instance\_type" {

description = "Represens EC2 instance type"

default = "t2.micro"

}

variable "key\_name" {

description = ""

default = "awslab"

}

# provider.tf

provider "aws" {

region = "ap-south-1"

access\_key = "AKIA23WHUC7VOLG46UX6"

secret\_key = "duA6Wxsh0J+j9XfY5H49hqgX+ZLrPzXu/8b4/t/i2"

# main.tf

resource "aws\_instance" "vinod\_ec2\_vm" {

ami = "${var.ami}"

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

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

security\_groups = ["default"]

tags = {

Name = "Linux-VM"

}

}

"aws\_instance" = aws resource name

"vinod\_ec2\_vm" = Local name / Terraform reference name / Terraform resource name

We have created aws ec2 instance using our script and we used "aws\_instance" as aws resource name and that can be access using our local name or terraform referance name

After the script execution all the resources information will be pointed or available in our local name or terraform referance name ("vinod\_ec2\_vm" )

By using this Terraform reference name ie. "vinod\_ec2\_vm" we can access the resources information by using output variables

Output variable syntax

output <"outputvariable name"> {

value = <aws resource name.terrafrom reference name.information attribute keyword>

}

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

Output variable terraform script

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

# output\_vars.tf

output "ec2\_vm\_public\_ip" {

value = aws\_instance.vinod\_ec2\_vm.public\_ip

}

output "ec2\_vm\_privte\_ip" {

value = aws\_instance.vinod\_ec2\_vm.private\_ip

}

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

Write a terraform script to create an ec2 instance using variable concept (input variables) and get some information using output variables

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

-----------------------

input\_vars.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"

}

------------------------

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"

}

}

--------------------------------------------

# output\_vars.tf

output "ec2\_vm\_public\_ip" {

value = aws\_instance.vinod\_ec2\_vm.public\_ip

}

output "ec2\_vm\_privte\_ip" {

value = aws\_instance.vinod\_ec2\_vm.private\_ip

}

output "ec2\_vm\_subnet\_id" {

value = aws\_instance.scopedindiavm.subnet\_id

}

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

To get full information of our Terraform resource

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

# output\_fullinfo\_vars.tf

output "ec2\_vm" {

value = aws\_instance.scopedindiavm

}

$ terraform init

Initialize terraform script

$ terraform validate

Verify terraform script syntax is valid or not

$ terraform fmt

Format terraform script indent spacing

$ terraform plan

Create execution plan for terraform script

# terraform apply --auto-approve

Create the actual resource (ec2 Instance)

# terraform destroy --auto-approve

Remove the resources which we had created using the script