

Terraform Modules and Variables

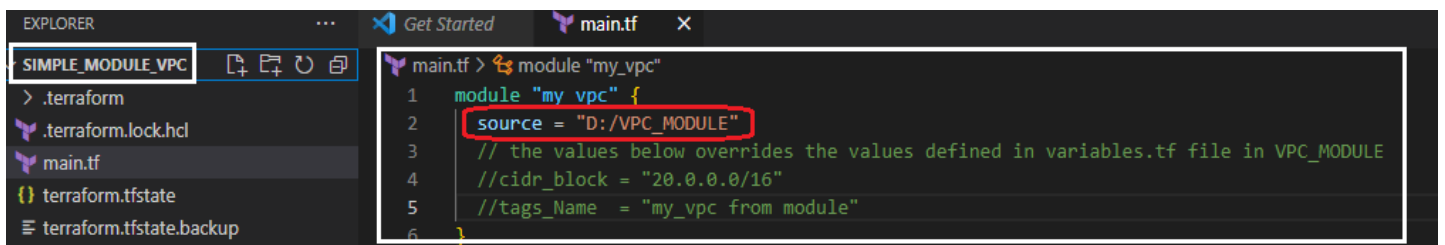
1) Why or What is a purpose of using Modules in terraform?

- ➔ Mainly Modules are used to reuse architecture or resources already created.
- ➔ Used to create similar architecture for Different environments like DEV, PRE_PROD and PROD.

Just we need to specify values that are required as per our environment needs.

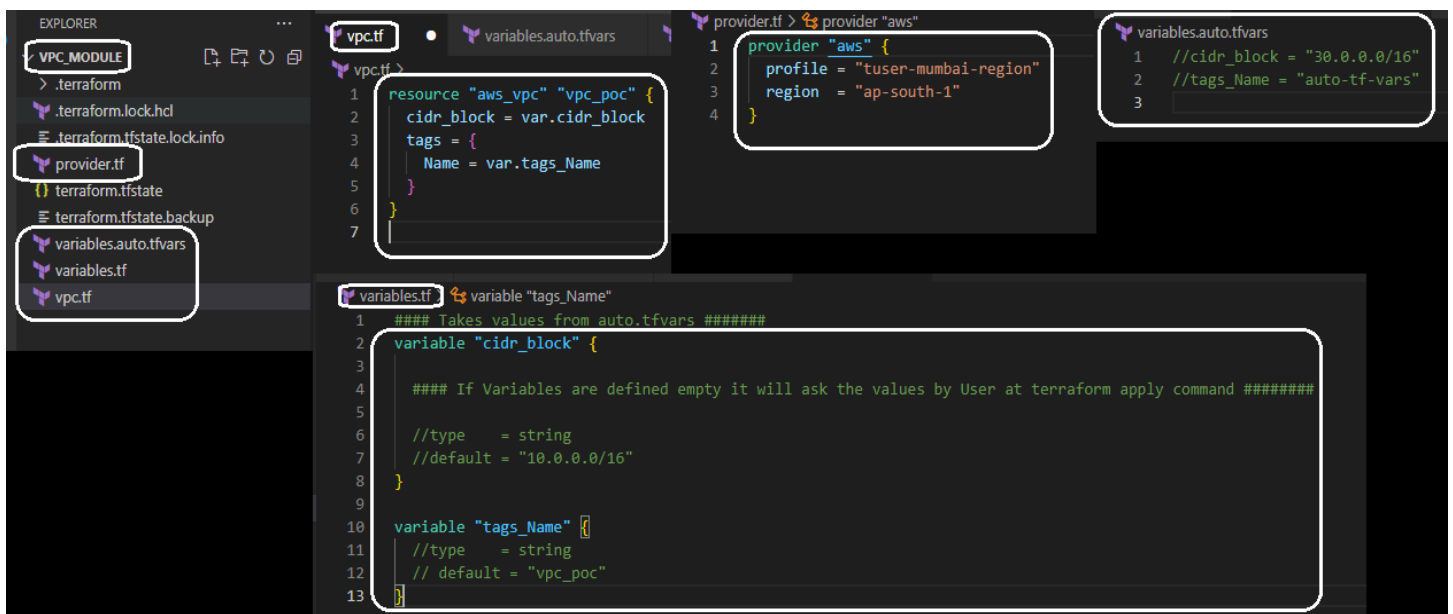
Below is simple example for creation of VPC for Different environments like DEV, PRE_PROD and PROD

Need to Define Module like Below and in Source we need to provide the folder in which resources or .tf values are saved



```
main.tf > module "my_vpc"
1 module "my_vpc" {
2   source = "D:/VPC_MODULE"
3   // the values below overrides the values defined in variables.tf file in VPC_MODULE
4   //cidr_block = "20.0.0.0/16"
5   //tags_Name = "my_vpc from module"
6 }
```

Below are the files that are stored in the Source folder(.tf files)



```
vpc.tf
1 resource "aws_vpc" "vpc_poc" {
2   cidr_block = var.cidr_block
3   tags = {
4     Name = var.tags_Name
5   }
6 }

provider.tf
1 provider "aws" {
2   profile = "tuser-mumbai-region"
3   region = "ap-south-1"
4 }

variables.tf
1 variable "tags_Name" {
2   type = string
3   default = "vpc_poc"
4 }

variables.auto.tfvars
1 //cidr_block = "30.0.0.0/16"
2 //tags_Name = "auto-tf-vars"
```

NOTE:

- 1) If we create VPC using Module without specifying any variable in it, it will take variables values from variable.tf file
Even we have declared the auto.tfvar file still it will take values from variables.tf
Auto.tfvar file is used when there is variable.tf file present

```
D: > SIMPLE_MODULE_VPC > main.tf > module "my_vpc"
1 module "my_vpc" {
2   source = "D:/VPC_MODULE"
3   // the values below overrides the values defined in variables.tf file in VPC_MODULE
4   //cidr_block = "20.0.0.0/16"
5   //tags_Name = "my_vpc from module"
6 }
```

```
variables.tf > variable "tags_Name"
1 ##### Takes values from auto.tfvars #####
2 variable "cidr_block" {
3   ##### If Variables are defined empty it will ask the values by User at terraform apply command #####
4   type = string
5   default = "10.0.0.0/16"
6 }
7
8 variable "tags_Name" {
9   type = string
10  default = "vpc_poc"
11 }
```

Console Output :

<input type="checkbox"/> vpc_poc	vpc-08012f3056b98a496	✓ Available	10.0.0.0/16
----------------------------------	-----------------------	-------------	-------------

- 2) If we create a VPC using Modules and specify the new values that I need to create a new Environment with Module itself then it will override all values declared in variables.tf file or any tfvar file.

```
D: > SIMPLE_MODULE_VPC > main.tf > module "my_vpc"
1 module "my_vpc" {
2   source = "D:/VPC_MODULE"
3   // the values below overrides the values defined in variables.tf file in VPC_MODULE
4   cidr_block = "20.0.0.0/16"
5   tags_Name = "my_vpc value from module"
6 }
```

Console Output :

<input type="checkbox"/> my_vpc value from module	vpc-03ab7057f3c206bd6	✓ Available	20.0.0.0/16
---	-----------------------	-------------	-------------

- 3) If we create a VPC using our resources or we can say direct .tf files if we are using auto.tfvars file like below then it will override our values which are present in variables.tf and apply the values which are there in auto.tfvars file

The screenshot shows three files in a Terraform project:

- vpc.tf**: Contains a resource `aws_vpc "vpc_poc"` with variables `cidr_block` and `tags_Name` assigned to `var.cidr_block` and `var.tags_Name` respectively.
- variables.tf**: Defines `variable "cidr_block"` with a default value of `"10.0.0.0/16"` and `variable "tags_Name"` with a default value of `"vpc_poc"`.
- variables.auto.tfvars**: Contains values `cidr_block = "30.0.0.0/16"` and `tags_Name = "auto-tf-vars"`, which override the defaults in `variables.tf`.

Console Output :

<input type="checkbox"/>	auto-tf-vars	Values added by auto.tfvars file	vpc-0deb468d9e87a8a6f	Available	30.0.0.0/16
--------------------------	--------------	----------------------------------	-----------------------	-----------	-------------

- 4) If we want to have a user-input at run time then declare a variables in variables.tf file without any values just an empty variable block like below.

The screenshot shows two parts:

- variables.tf**: Defines `variable "cidr_block"` and `variable "tags_Name"` as empty blocks, labeled **EMPTY BLOCKS**.
- Terminal Output**: Shows the command `terraform plan` being run. It prompts for `var.cidr_block` (entering `40.0.0.0/16`) and `var.tags_Name` (entering `user-input`), both labeled **USER ENTERED**.

Console Output:

<input type="checkbox"/>	user-input	OUTPUT	vpc-0cb4c82208a7c0ead	Available	40.0.0.0/16
--------------------------	------------	--------	-----------------------	-----------	-------------

- 5) When we are creating VPC directly using .tf files we can add variables on terraform apply command which will override the values that are present in variables.tf file
Syntax : `terraform apply -var="cidr_block=50.0.0.0/16"` :

<input type="checkbox"/>	vpc_poc	Defined in variable.tf file	CIDR Value updated on apply command		50.0.0.0/16
--------------------------	---------	-----------------------------	-------------------------------------	--	-------------