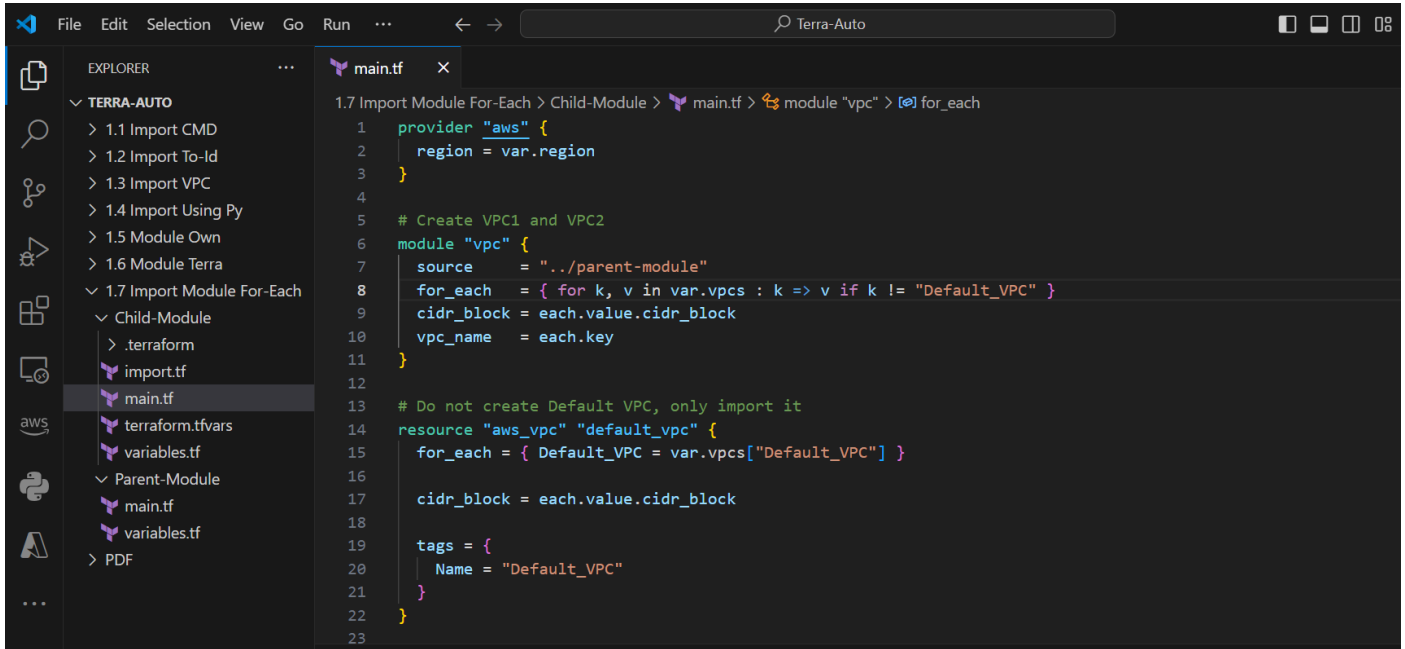


Terraform import using Module

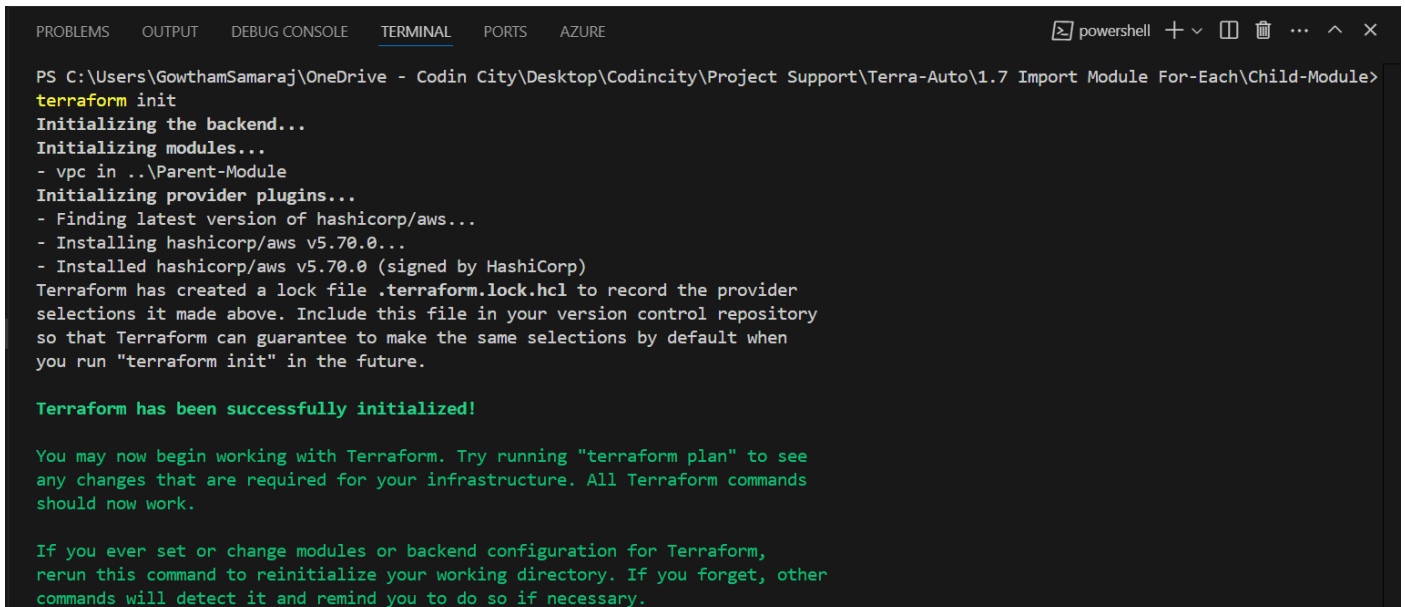
Task 1.7 → Terraform import using Module that supports for each concept

Step:1 I created two folders for two modules **Child-Module**, **Parent-Module**, Inside the module I created **main.tf**, **variables.tf** & **terraform.tfvars**



```
1.7 Import Module For-Each > Child-Module > main.tf > module "vpc" > for_each
1  provider "aws" {
2      region = var.region
3  }
4
5  # Create VPC1 and VPC2
6  module "vpc" {
7      source      = "../parent-module"
8      for_each    = { for k, v in var.vpcs : k => v if k != "Default_VPC" }
9      cidr_block  = each.value.cidr_block
10     vpc_name    = each.key
11 }
12
13 # Do not create Default VPC, only import it
14 resource "aws_vpc" "default_vpc" {
15     for_each = { Default_VPC = var.vpcs["Default_VPC"] }
16
17     cidr_block = each.value.cidr_block
18
19     tags = {
20         Name = "Default_VPC"
21     }
22 }
23
```

Step:2 After creating module I will initializing the terraform file → terraform init



```
PS C:\Users\GowthamSamaraj\OneDrive - Codin City\Desktop\Codincity\Project Support\Terra-Auto\1.7 Import Module For-Each\Child-Module>
terraform init
Initializing the backend...
Initializing modules...
- vpc in ../Parent-Module
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.70.0...
- Installed hashicorp/aws v5.70.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Step:3 → terraform plan

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE powershell + - [ ] [ ] ... ^ X

terraform plan
aws_vpc.default_vpc["Default_VPC"]: Preparing import... [id=vpc-0ac3883de5bde45b6]
aws_vpc.default_vpc["Default_VPC"]: Refreshing state... [id=vpc-0ac3883de5bde45b6]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
+ create

Terraform will perform the following actions:

# aws_vpc.default_vpc["Default_VPC"] will be imported
resource "aws_vpc" "default_vpc" {
  arn                  = "arn:aws:ec2:us-east-1:339713187727:vpc/vpc-0ac3883de5bde45b6"
  assign_generated_ipv6_cidr_block = false
  cidr_block           = "172.31.0.0/16"
  default_network_acl_id = "acl-01693ee5160155dc0"
  default_route_table_id = "rtb-0cf454d4d34bae4bb"
  default_security_group_id = "sg-0a2929bc0cc568a1f"
  dhcp_options_id       = "dopt-09cd5f3382f696b6f"
  enable_dns_hostnames  = true
  enable_dns_support    = true
  enable_network_address_usage_metrics = false
  id                    = "vpc-0ac3883de5bde45b6"
  instance_tenancy      = "default"
  ipv6_association_id   = null
  ipv6_cidr_block       = null
  ipv6_cidr_block_network_border_group = null
  ipv6_ipam_pool_id     = null
  ipv6_netmask_length   = 0
}
```

Step:4 → terraform apply -auto-approve

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE powershell + - [ ] [ ] ... ^ X

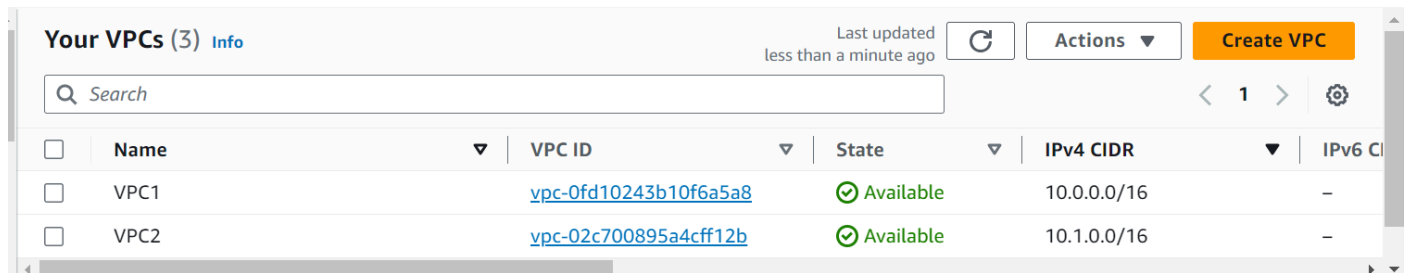
PS C:\Users\GowthamSamaraj\OneDrive - Codin City\Desktop\CodinCity\Project Support\Terra-Auto\1.7 Import Module For-Each\Child-Module>
terraform apply --auto-approve
aws_vpc.default_vpc["Default_VPC"]: Preparing import... [id=vpc-0ac3883de5bde45b6]
aws_vpc.default_vpc["Default_VPC"]: Refreshing state... [id=vpc-0ac3883de5bde45b6]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
+ create

Terraform will perform the following actions:

# aws_vpc.default_vpc["Default_VPC"] will be imported
resource "aws_vpc" "default_vpc" {
  arn                  = "arn:aws:ec2:us-east-1:339713187727:vpc/vpc-0ac3883de5bde45b6"
  assign_generated_ipv6_cidr_block = false
  cidr_block           = "172.31.0.0/16"
  default_network_acl_id = "acl-01693ee5160155dc0"
  default_route_table_id = "rtb-0cf454d4d34bae4bb"
  default_security_group_id = "sg-0a2929bc0cc568a1f"
  dhcp_options_id       = "dopt-09cd5f3382f696b6f"
  enable_dns_hostnames  = true
  enable_dns_support    = true
  enable_network_address_usage_metrics = false
  id                    = "vpc-0ac3883de5bde45b6"
  instance_tenancy      = "default"
  ipv6_association_id   = null
  ipv6_cidr_block       = null
  ipv6_cidr_block_network_border_group = null
  ipv6_ipam_pool_id     = null
}
```

Step:5 The VPC has been created



<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	VPC1	vpc-0fd10243b10f6a5a8	Available	10.0.0.0/16	-
<input type="checkbox"/>	VPC2	vpc-02c700895a4cff12b	Available	10.1.0.0/16	-

Step:6 As per the scenario two VPC has been created, One VPC has been imported

```
aws_vpc.default_vpc["Default_VPC"]: Importing... [id=vpc-0ac3883de5bde45b6]
aws_vpc.default_vpc["Default_VPC"]: Import complete [id=vpc-0ac3883de5bde45b6]
module.vpc["VPC2"].aws_vpc.my_vpc: Creating...
module.vpc["VPC1"].aws_vpc.my_vpc: Creating...
module.vpc["VPC1"].aws_vpc.my_vpc: Still creating... [10s elapsed]
module.vpc["VPC2"].aws_vpc.my_vpc: Still creating... [10s elapsed]
module.vpc["VPC2"].aws_vpc.my_vpc: Creation complete after 18s [id=vpc-02c700895a4cff12b]
module.vpc["VPC1"].aws_vpc.my_vpc: Creation complete after 18s [id=vpc-0fd10243b10f6a5a8]

Apply complete! Resources: 1 imported, 2 added, 0 changed, 0 destroyed.
```

Step:7 This is the flow of using modules

