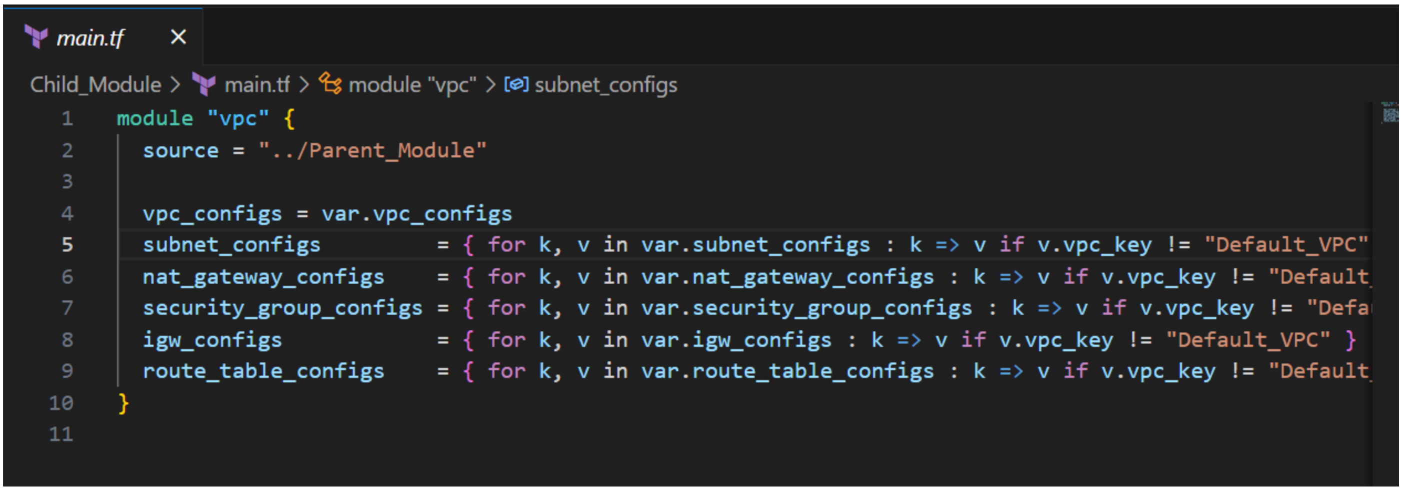
Terraform Module for Creation & Import

Task 1.8 → Terraform import using Module that supports for each concept

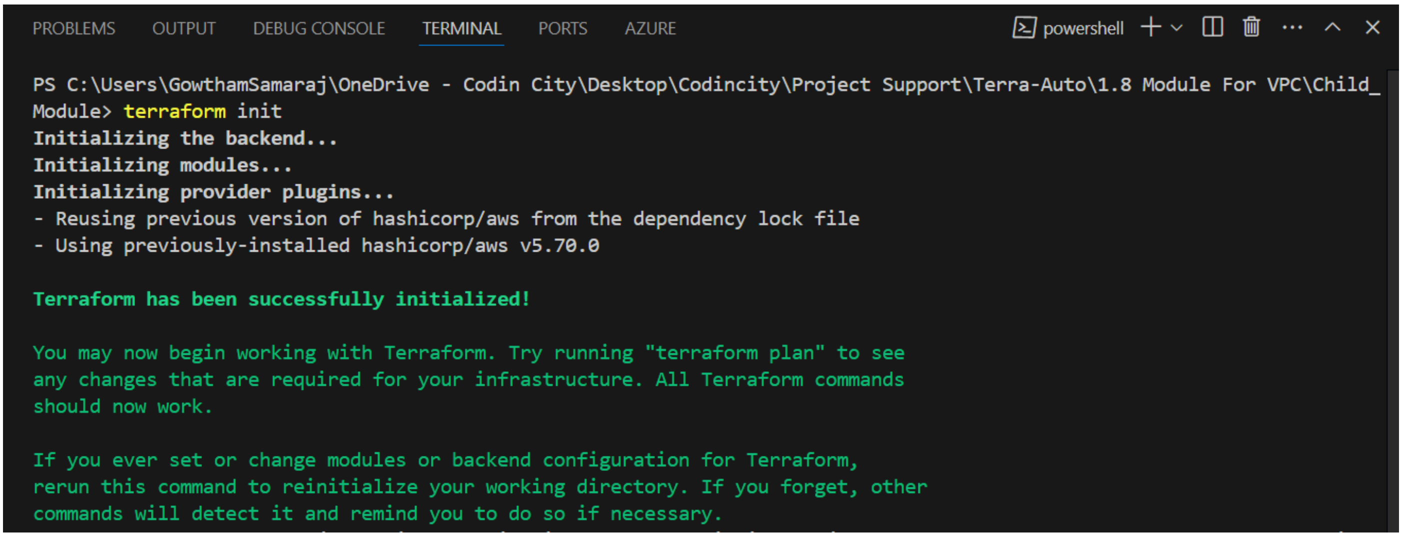
Step:1 I created two folders for two modules **Child-Module, Parent-Module,** Inside the module I created **main.tf, variables.tf & terraform.tfvars**



```
module "vpc" {
  source = "../Parent_Module"

  vpc_configs = var.vpc_configs
  subnet_configs          = { for k, v in var.subnet_configs : k => v if v.vpc_key != "Default_VPC"
  nat_gateway_configs     = { for k, v in var.nat_gateway_configs : k => v if v.vpc_key != "Default_
  security_group_configs = { for k, v in var.security_group_configs : k => v if v.vpc_key != "Defa
  igw_configs             = { for k, v in var.igw_configs : k => v if v.vpc_key != "Default_VPC" }
  route_table_configs     = { for k, v in var.route_table_configs : k => v if v.vpc_key != "Default_
}
```

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



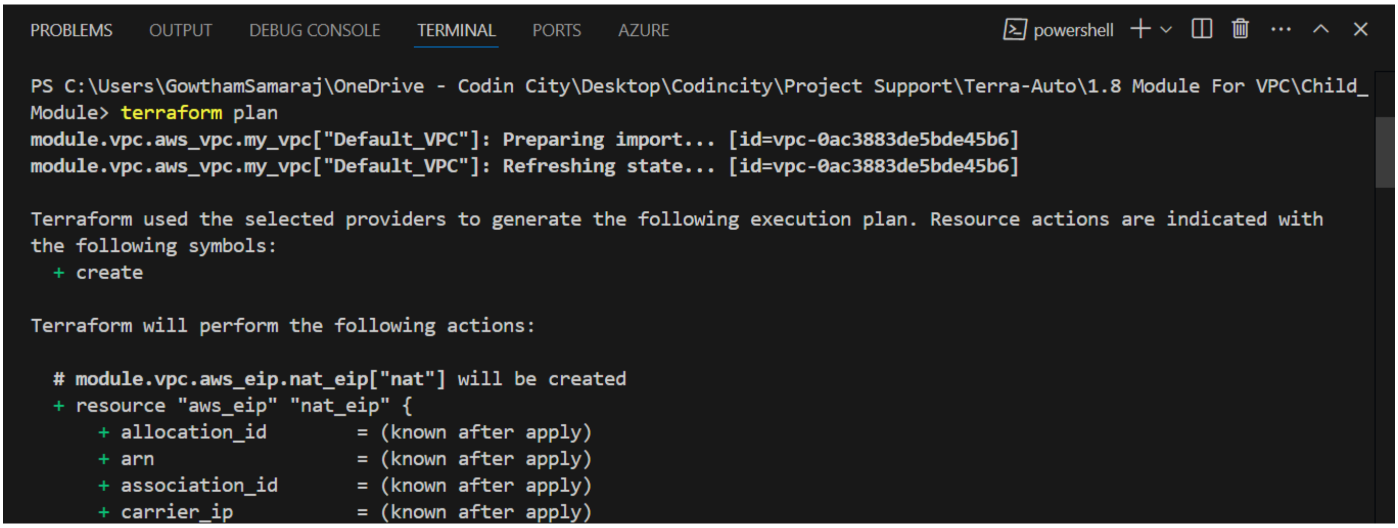
```
PS C:\Users\GowthamSamaraj\OneDrive - Codin City\Desktop\Codincity\Project Support\Terra-Auto\1.8 Module For VPC\Child_
Module> terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.70.0

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**



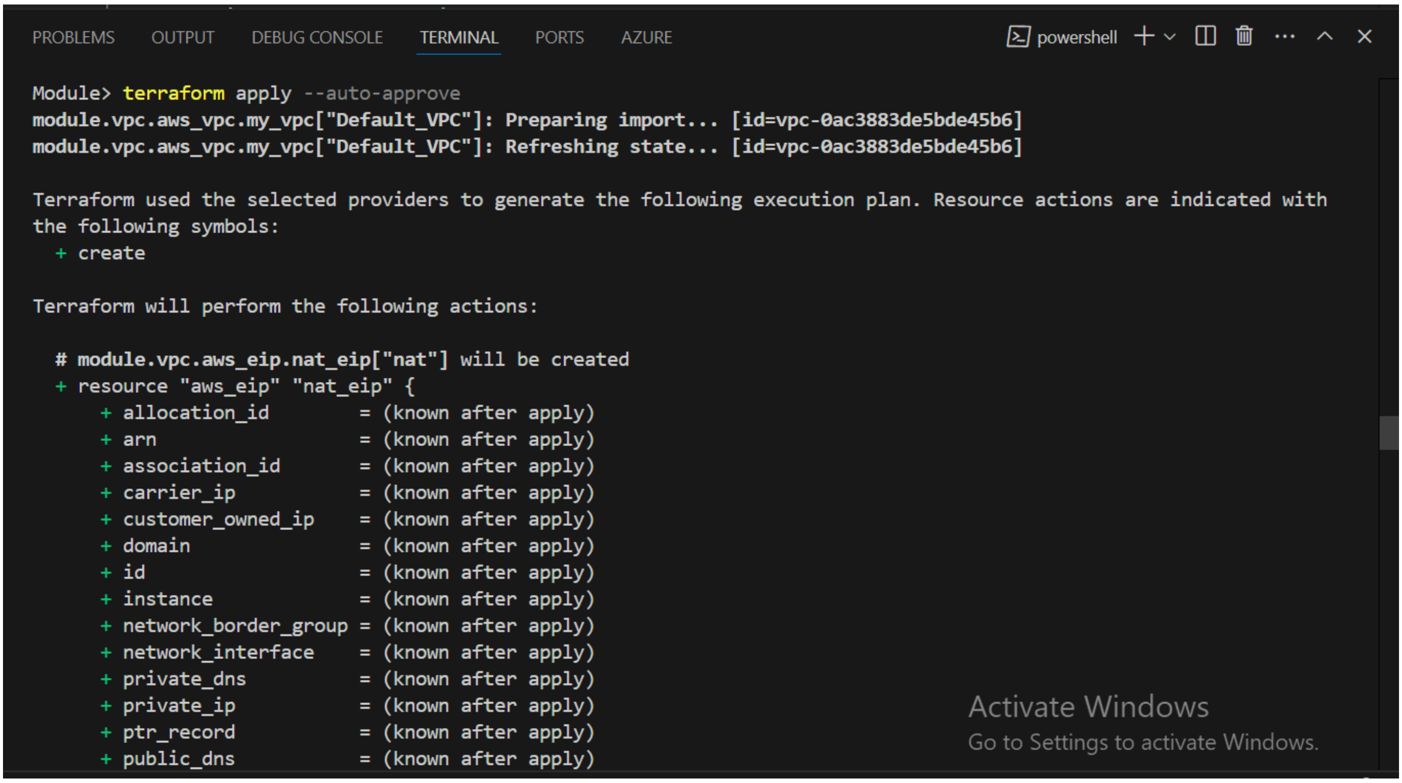
```
PS C:\Users\GowthamSamaraj\OneDrive - Codin City\Desktop\Codincity\Project Support\Terra-Auto\1.8 Module For VPC\Child_
Module> terraform plan
module.vpc.aws_vpc.my_vpc["Default_VPC"]: Preparing import... [id=vpc-0ac3883de5bde45b6]
module.vpc.aws_vpc.my_vpc["Default_VPC"]: Refreshing state... [id=vpc-0ac3883de5bde45b6]

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:

  # module.vpc.aws_eip.nat_eip["nat"] will be created
  + resource "aws_eip" "nat_eip" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
```

Step:4 → terraform apply –auto-approve



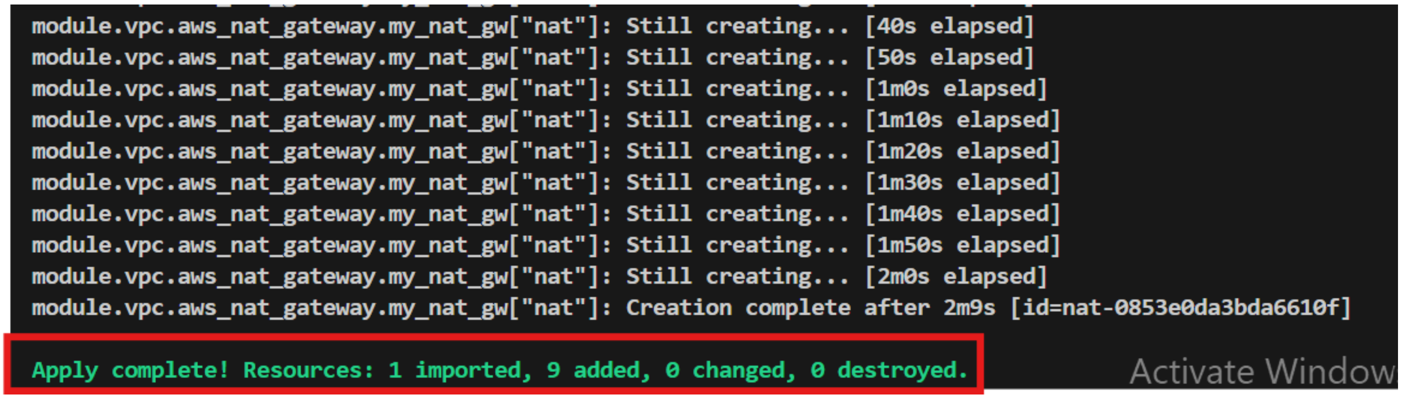
```
Module> terraform apply --auto-approve
module.vpc.aws_vpc.my_vpc["Default_VPC"]: Preparing import... [id=vpc-0ac3883de5bde45b6]
module.vpc.aws_vpc.my_vpc["Default_VPC"]: Refreshing state... [id=vpc-0ac3883de5bde45b6]

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:

  # module.vpc.aws_eip.nat_eip["nat"] will be created
  + resource "aws_eip" "nat_eip" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = (known after apply)
      + id                   = (known after apply)
      + instance             = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + ptr_record           = (known after apply)
      + public_dns           = (known after apply)
```

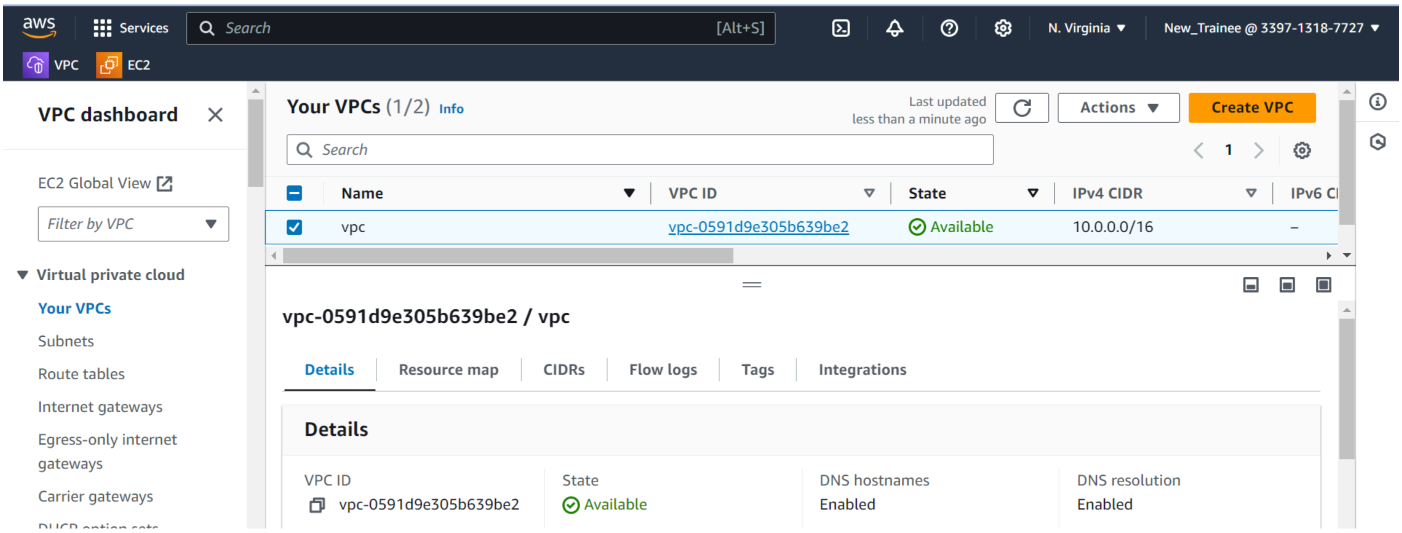
Step:5 As per the scenario one VPC has been imported, also new VPC, Subnet, IGW, NAT, SG created



```
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [40s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [50s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [1m0s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [1m10s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [1m20s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [1m30s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [1m40s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [1m50s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Still creating... [2m0s elapsed]
module.vpc.aws_nat_gateway.my_nat_gw["nat"]: Creation complete after 2m9s [id=nat-0853e0da3bda6610f]

Apply complete! Resources: 1 imported, 9 added, 0 changed, 0 destroyed.
```

Step:6 The VPC has been created



Your VPCs (1/2) Info

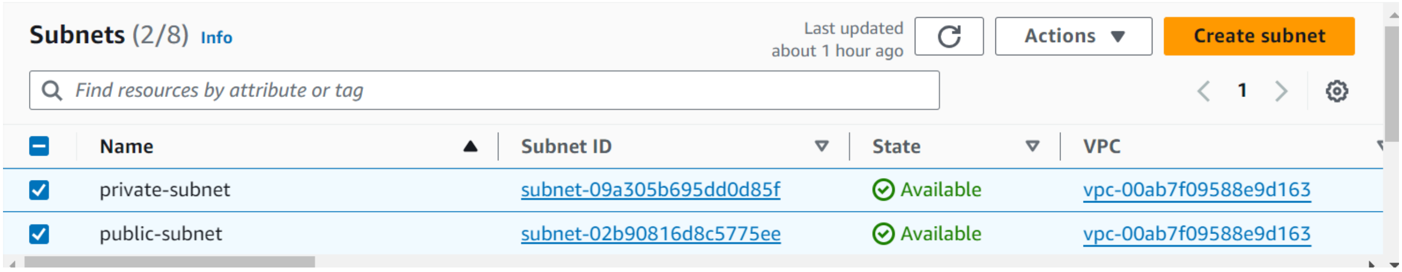
☐	Name	VPC ID	State	IPv4 CIDR	IPv6 CI
☑	vpc	vpc-0591d9e305b639be2	✓ Available	10.0.0.0/16	–

vpc-0591d9e305b639be2 / vpc

Details

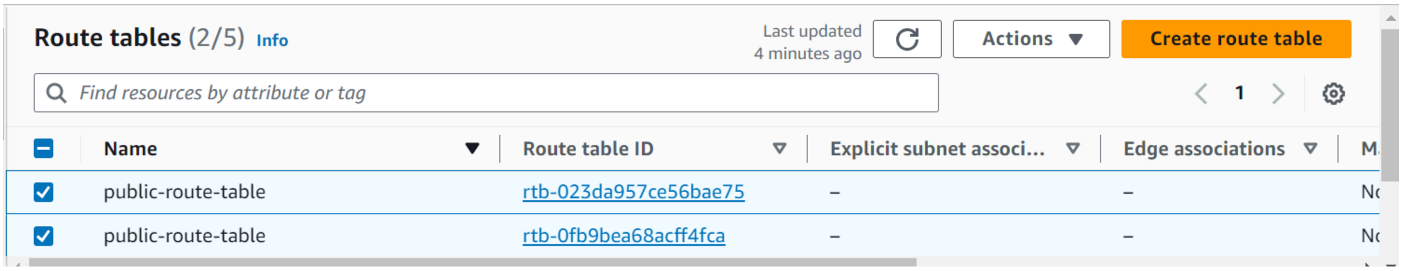
VPC ID	State	DNS hostnames	DNS resolution
vpc-0591d9e305b639be2	✓ Available	Enabled	Enabled

Step:7 Created one public subnet, one private subnet


Subnets (2/8) Info
Last updated about 1 hour ago
Actions ▼
Create subnet
Find resources by attribute or tag
1

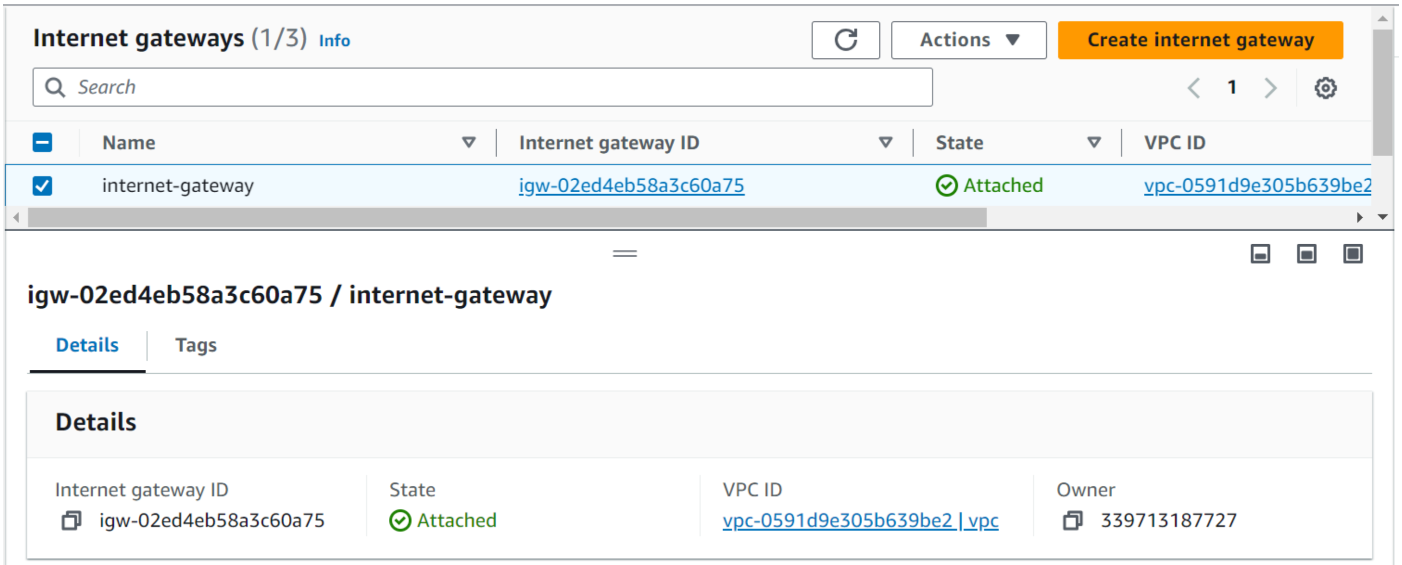
☐	Name ▲	Subnet ID	State	VPC
☑	private-subnet	subnet-09a305b695dd0d85f	✓ Available	vpc-00ab7f09588e9d163
☑	public-subnet	subnet-02b90816d8c5775ee	✓ Available	vpc-00ab7f09588e9d163

Step:8 Route table has been created & associated with corresponding subnet


Route tables (2/5) Info
Last updated 4 minutes ago
Actions ▼
Create route table
Find resources by attribute or tag
1

☐	Name ▼	Route table ID	Explicit subnet associ...	Edge associations	M
☑	public-route-table	rtb-023da957ce56bae75	–	–	N
☑	public-route-table	rtb-0fb9bea68acff4fca	–	–	N

Step:9 Internet Gateway has been created & associated with public subnet



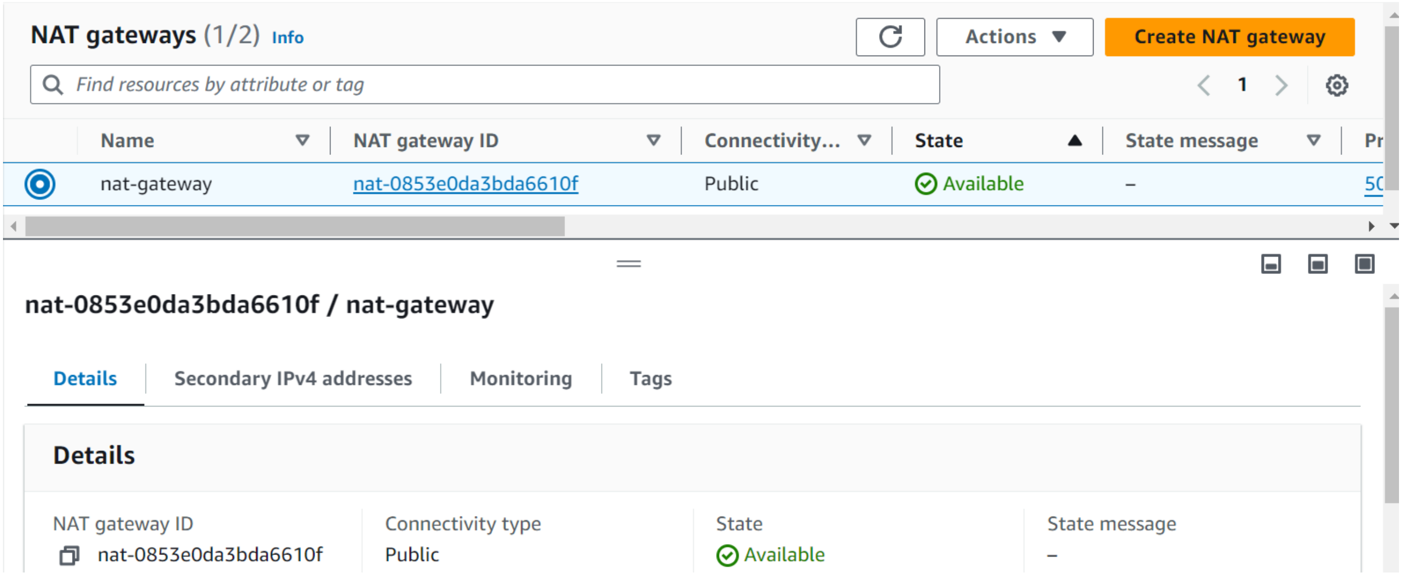
☐	Name	Internet gateway ID	State	VPC ID
☑	internet-gateway	igw-02ed4eb58a3c60a75	✓ Attached	vpc-0591d9e305b639be2

igw-02ed4eb58a3c60a75 / internet-gateway

Details

Internet gateway ID	State	VPC ID	Owner
igw-02ed4eb58a3c60a75	✓ Attached	vpc-0591d9e305b639be2 | vpc	339713187727

Step:10 NAT Gateway has been created & associated with private subnet



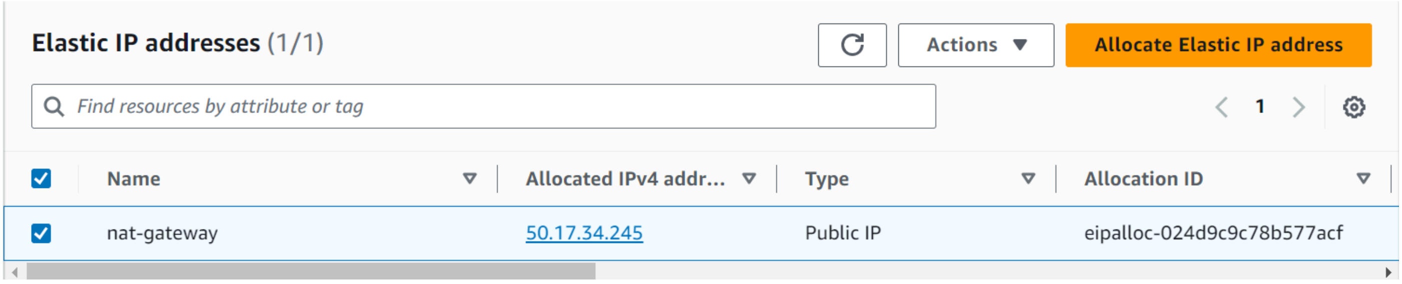
	Name	NAT gateway ID	Connectivity...	State ▲	State message	Pr
●	nat-gateway	nat-0853e0da3bda6610f	Public	✓ Available	–	5(

nat-0853e0da3bda6610f / nat-gateway

Details

NAT gateway ID	Connectivity type	State	State message
nat-0853e0da3bda6610f	Public	✓ Available	–

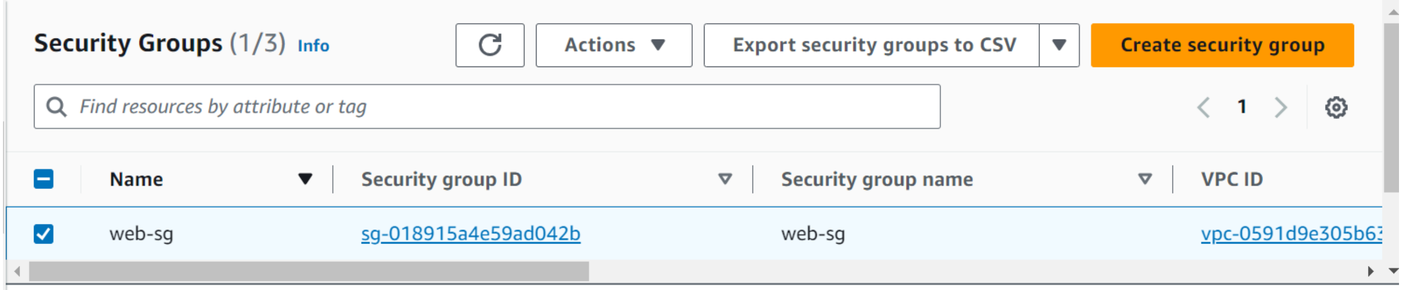
Step:11 Elastic IP has been created for NAT Gateway


Actions
Allocate Elastic IP address
Find resources by attribute or tag
1

Elastic IP addresses (1/1)

☑	Name	Allocated IPv4 addr...	Type	Allocation ID
☑	nat-gateway	50.17.34.245	Public IP	eipalloc-024d9c9c78b577acf

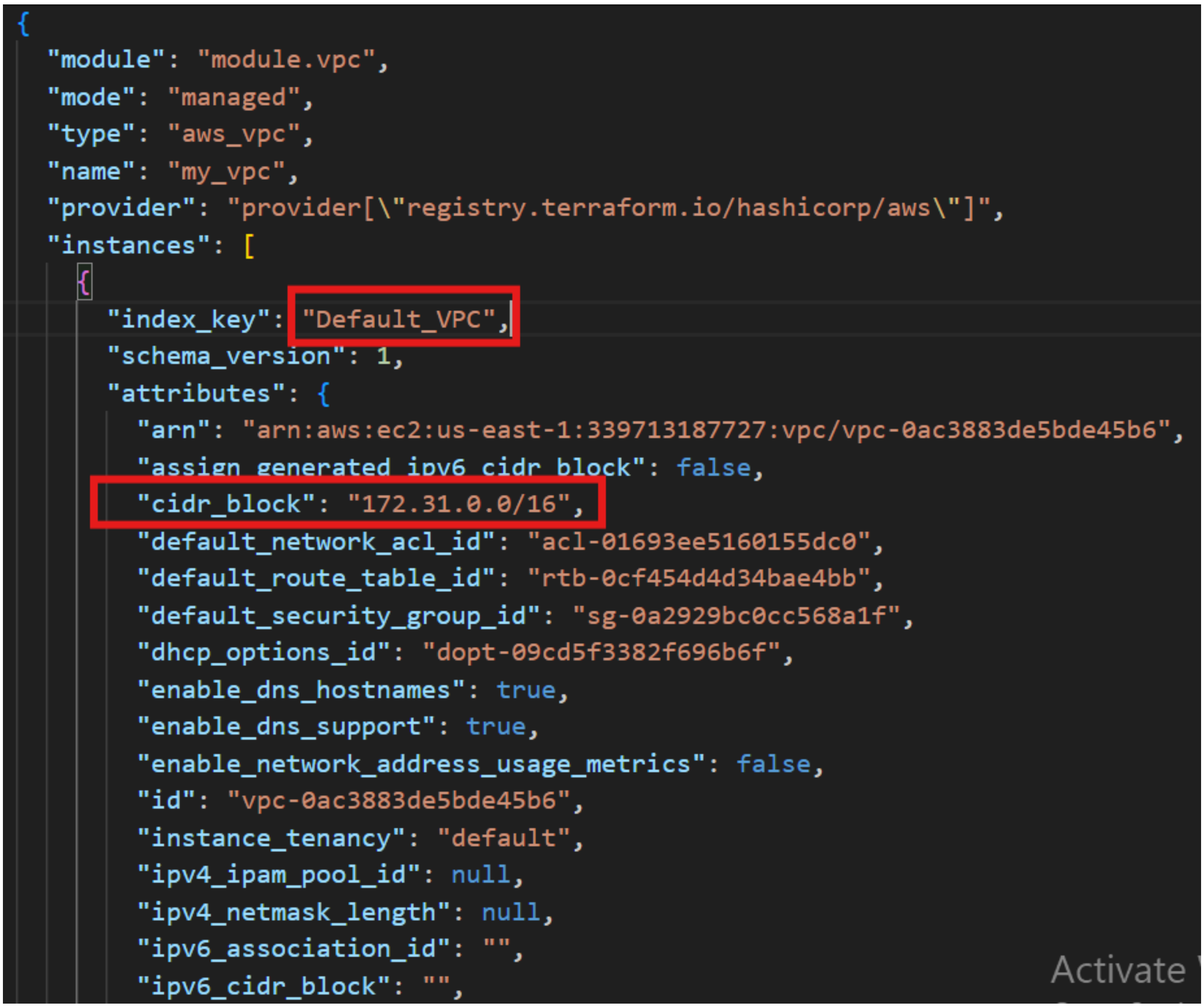
Step:12 Security Group has been created


Info
Actions
Export security groups to CSV
Create security group
Find resources by attribute or tag
1

Security Groups (1/3)

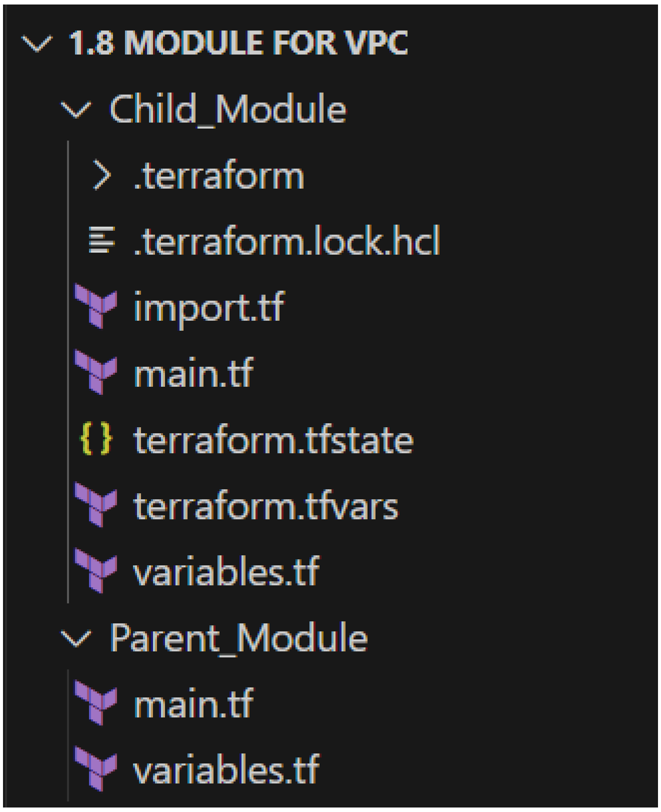
	Name	Security group ID	Security group name	VPC ID
☑	web-sg	sg-018915a4e59ad042b	web-sg	vpc-0591d9e305b63

Step:13 Now We can see that state file has been imported the Default_VPC & Others also created



```
{
  "module": "module.vpc",
  "mode": "managed",
  "type": "aws_vpc",
  "name": "my_vpc",
  "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
  "instances": [
    {
      "index_key": "Default_VPC",
      "schema_version": 1,
      "attributes": {
        "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",
        "ipv4_ipam_pool_id": null,
        "ipv4_netmask_length": null,
        "ipv6_association_id": "",
        "ipv6_cidr_block": "",
```

Step:14 This is the flow of using modules



```
1.8 MODULE FOR VPC
  Child_Module
    .terraform
    .terraform.lock.hcl
    import.tf
    main.tf
    terraform.tfstate
    terraform.tfvars
    variables.tf
  Parent_Module
    main.tf
    variables.tf
```