

School of Computer Science
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN, UTTARAKHAND



**System Monitoring And Configuration
Management**

Lab File

(2023-2024)

for

6th Semester

Submitted To:

Dr. Hitesh Kumar Sharma

Submitted By:

Prakhar Gupta

B. Tech. CSE DevOps

[6^h Semester]

Sap id- 500093012

Batch 2

R2142210572

LAB EXERCISE 8

Aim: Creating a VPC in Terraform Objective

Step 1: Create a main.tf file

```
LAB_8 > main.tf
1 provider "aws" {
2   region = "us-east-1"
3   access_key = "AKIA2UC27CLCKMWF5RA"
4   secret_key = "f5AEpq0QFLngq+WzxzMLL3a5SVpsH2FQ6IGoxRo"
5 }
6
7 variable "iam_users" {
8   type = list(string)
9   default = ["user1", "user2", "user3"]
10 }
11
12 resource "aws_vpc" "my_vpc" {
13   cidr_block = "10.0.0.0/16"
14   enable_dns_support = true
15   enable_dns_hostnames = true
16   tags = {
17     Name = "MyVPC"
18   }
19 }
20
21 resource "aws_subnet" "my_subnet" {
22   count = 2
23   vpc_id = aws_vpc.my_vpc.id
24   cidr_block = "10.0.${count.index + 1}.0/24"
25   availability_zone = "us-east-1a"
26   map_public_ip_on_launch = true
27   tags = {
28     Name = "MySubnet-${count.index + 1}"
29   }
30 }
31
```

Step 2: Create a vpc.tf file

```
ew Go Run Terminal Window Help
vpc.tf — SPCM

main.tf vpc.tf X
LAB_8 > vpc.tf
1 Click here to ask Blackbox to help you code faster
2 resource "aws_vpc" "gfg-vpc" {
3   cidr_block = "10.0.0.0/16"
4 }
5
6 resource "aws_subnet" "gfg-subnet" {
7   vpc_id = aws_vpc.gfg-vpc.id
8   cidr_block = "10.0.1.0/24"
9
10   tags = {
11     Name = "gfg-subnet"
12   }
13 }
14
15 resource "aws_internet_gateway" "gfg-gw" {
16   vpc_id = aws_vpc.gfg-vpc.id
17
18   tags = {
19     Name = "gfg-IG"
20   }
21 }
22
23 resource "aws_route_table" "gfg-rt" {
24   vpc_id = aws_vpc.gfg-vpc.id
25
26   route {
27     cidr_block = "0.0.0.0/0"
28     gateway_id = aws_internet_gateway.gfg-gw.id
29   }
30
31   tags = {
32     Name = "GFG-Route-Table"
33   }
34 }
35
36 resource "aws_route_table_association" "gfg-rta" {
37   subnet_id = aws_subnet.gfg-subnet.id
38   route_table_id = aws_route_table.gfg-rt.id
39 }
40
41 resource "aws_security_group" "gfg-sg" {
42   name = "my-gfg-sg"
43   vpc_id = aws_vpc.gfg-vpc.id
44
45   ingress {
46     description = "TLS from VPC"
47     from_port = 20
48     to_port = 20
49     protocol = "tcp"
50     cidr_blocks = ["0.0.0.0/0"]
51     ipv6_cidr_blocks = [":::/0"]
52   }
53
54   egress {
55     from_port = 0
56     to_port = 0
57     protocol = "-1"
58     cidr_blocks = ["0.0.0.0/0"]
59     ipv6_cidr_blocks = [":::/0"]
60   }
61
62   tags = {
63     Name = "my-gfg-sg"
64   }
65 }
```

Step 3: Now run terraform init command to Initialize.

```
PrakharGupta@192 LAB_8 % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.37.0

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.
PrakharGupta@192 LAB_8 % terraform validate
Success! The configuration is valid.

PrakharGupta@192 LAB_8 % terraform plan

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_internet_gateway.gfg-gw will be created
+ resource "aws_internet_gateway" "gfg-gw" {
+   arn           = (known after apply)
+   id           = (known after apply)
+   owner_id     = (known after apply)
+   tags         = {
+     + "Name" = "gfg-IG"
+   }
+   tags_all     = {
+     + "Name" = "gfg-IG"
+   }
+   vpc_id       = (known after apply)
}
```

Step 5: Now run terraform plan command

```
F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpc_8>terraform plan

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_subnet.my_subnet[0] will be created
+ resource "aws_subnet" "my_subnet" {
+   arn                             = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone               = "us-east-1a"
+   availability_zone_id            = (known after apply)
+   cidr_block                      = "10.0.1.0/24"
+   enable_dns64                   = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id                             = (known after apply)
+   ipv6_cidr_block_association_id = (known after apply)
+   ipv6_native                    = false
+   map_public_ip_on_launch        = true
+   owner_id                       = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   tags                           = {
+     + "Name" = "MySubnet-1"
+   }
+   tags_all                       = {
+     + "Name" = "MySubnet-1"
+   }
+   vpc_id                         = (known after apply)
}

# aws_subnet.my_subnet[1] will be created
+ resource "aws_subnet" "my_subnet" {
+   arn                             = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone               = "us-east-1a"
+   availability_zone_id            = (known after apply)
+   cidr_block                      = "10.0.2.0/24"
+   enable_dns64                   = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id                             = (known after apply)
+   ipv6_cidr_block_association_id = (known after apply)
+   ipv6_native                    = false
+   map_public_ip_on_launch        = true
+   owner_id                       = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   tags                           = {
+     + "Name" = "MySubnet-1"
+   }
+   tags_all                       = {
+     + "Name" = "MySubnet-1"
+   }
+   vpc_id                         = (known after apply)
}
```

```
C:\Windows\System32\cmd.exe
}
+ vpc_id = (known after apply)
}

# aws_vpc.my_vpc will be created
+ resource "aws_vpc" "my_vpc" {
+   arn = (known after apply)
+   cidr_block = "10.0.0.0/16"
+   default_network_acl_id = (known after apply)
+   default_route_table_id = (known after apply)
+   default_security_group_id = (known after apply)
+   dhcp_options_id = (known after apply)
+   enable_dns_hostnames = true
+   enable_dns_support = true
+   enable_network_address_usage_metrics = (known after apply)
+   id = (known after apply)
+   instance_tenancy = "default"
+   ipv6_association_id = (known after apply)
+   ipv6_cidr_block = (known after apply)
+   ipv6_cidr_block_network_border_group = (known after apply)
+   main_route_table_id = (known after apply)
+   owner_id = (known after apply)
+   tags = {
+     "Name" = "MyVPC"
+   }
+   tags_all = {
+     "Name" = "MyVPC"
+   }
}

Plan: 3 to add, 0 to change, 0 to destroy.
```

Step 6: Now run the terraform apply command.

```
F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpc_8\terraform apply
aws_vpc.my_vpc: Refreshing state... [id=vpc-0df261dd0c6a4a5c3]

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_subnet.my_subnet[0] will be created
+ resource "aws_subnet" "my_subnet" {
+   arn = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone = "ap-south-1a"
+   availability_zone_id = (known after apply)
+   cidr_block = "10.0.1.0/24"
+   enable_dns64 = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id = (known after apply)
+   ipv6_cidr_block_association_id = (known after apply)
+   ipv6_native = false
+   map_public_ip_on_launch = true
+   owner_id = (known after apply)
+   private_dns_hostname_type_on_launch = (known after apply)
+   tags = {
+     "Name" = "MySubnet-1"
+   }
+   tags_all = {
+     "Name" = "MySubnet-1"
+   }
+   vpc_id = "vpc-0df261dd0c6a4a5c3"
}

# aws_subnet.my_subnet[1] will be created
+ resource "aws_subnet" "my_subnet" {
+   arn = (known after apply)
+   assign_ipv6_address_on_creation = false
+   availability_zone = "ap-south-1a"
+   availability_zone_id = (known after apply)
+   cidr_block = "10.0.2.0/24"
+   enable_dns64 = false
+   enable_resource_name_dns_a_record_on_launch = false
+   enable_resource_name_dns_aaaa_record_on_launch = false
+   id = (known after apply)
+   ipv6_cidr_block_association_id = (known after apply)
+   ipv6_native = false
+   map_public_ip_on_launch = true
+   owner_id = (known after apply)
}
```

```

+ tags_all
+ "Name" = "MySubnet-2"
}
+ vpc_id
}
}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

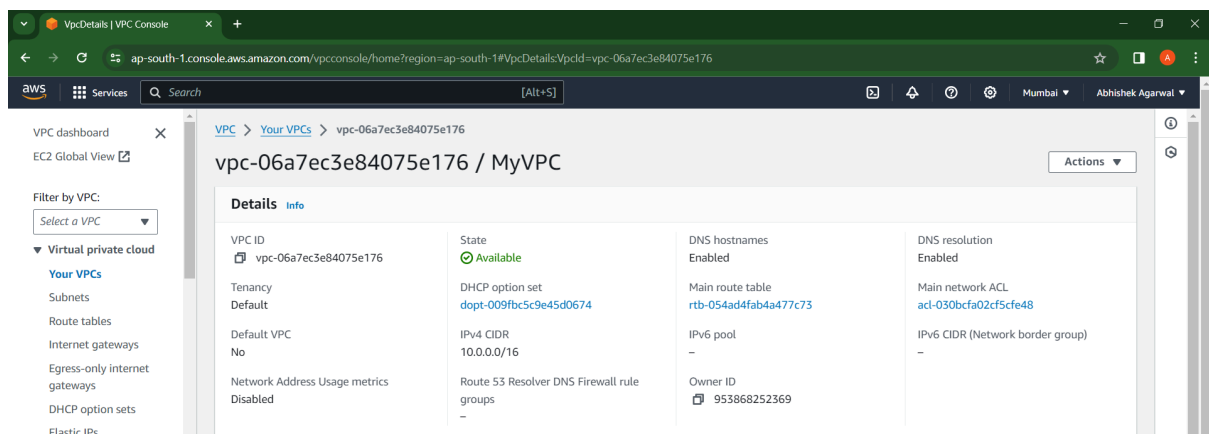
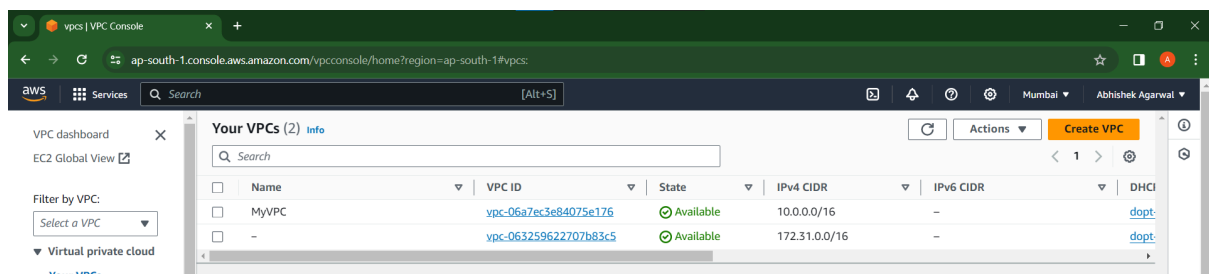
  Enter a value: yes

aws_subnet.my_subnet[1]: Creating...
aws_subnet.my_subnet[0]: Creating...
aws_subnet.my_subnet[1]: Still creating... [10s elapsed]
aws_subnet.my_subnet[0]: Still creating... [10s elapsed]
aws_subnet.my_subnet[0]: Creation complete after 11s [id=subnet-00fecef33341c030f]
aws_subnet.my_subnet[1]: Creation complete after 11s [id=subnet-01447d7d04cc270bf]

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

F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpc_8>

```



Step 7: Now run the terraform destroy to destroy the VPC created.

```

C:\Windows\System32\cmd.exe

F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpctest\terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-0df261dd0c6a4a5c3]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-00fecef33341c030f]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-01447d7d04cc270bf]

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

Terraform will perform the following actions:

# aws_subnet.my_subnet[0] will be destroyed
- resource "aws_subnet" "my_subnet" {
  arn = "arn:aws:ec2:ap-south-1:953868252369:subnet/subnet-00fecef33341c030f" -> null
  assign_ipv6_address_on_creation = false -> null
  availability_zone = "ap-south-1a" -> null
  availability_zone_id = "aps1-az1" -> null
  cidr_block = "10.0.1.0/24" -> null
  enable_dns64 = false -> null
  enable_in_iat_device_index = 0 -> null
  enable_resource_name_dns_a_record_on_launch = false -> null
  id = "subnet-00fecef33341c030f" -> null
  ipv6_native = false -> null
  map_customer_owned_ip_on_launch = false -> null
  map_public_ip_on_launch = true -> null
  owner_id = "953868252369" -> null
  private_dns_hostname_type_on_launch = "ip-name" -> null
  tags = {
    "Name" = "MySubnet-1"
  } -> null
  tags_all = {
    "Name" = "MySubnet-1"
  } -> null
  vpc_id = "vpc-0df261dd0c6a4a5c3" -> null
}

# aws_subnet.my_subnet[1] will be destroyed
- resource "aws_subnet" "my_subnet" {
  arn = "arn:aws:ec2:ap-south-1:953868252369:subnet/subnet-01447d7d04cc270bf" -> null
  assign_ipv6_address_on_creation = false -> null
  availability_zone = "ap-south-1a" -> null
  availability_zone_id = "aps1-az1" -> null
  cidr_block = "10.0.2.0/24" -> null
  enable_dns64 = false -> null
  enable_in_iat_device_index = 0 -> null
  enable_resource_name_dns_a_record_on_launch = false -> null
  id = "subnet-01447d7d04cc270bf" -> null
  ipv6_native = false -> null
}

```

```

C:\Windows\System32\cmd.exe

- "Name" = "MySubnet-2"
} -> null
- vpc_id = "vpc-0df261dd0c6a4a5c3" -> null
}

# aws_vpc.my_vpc will be destroyed
- resource "aws_vpc" "my_vpc" {
  arn = "arn:aws:ec2:ap-south-1:953868252369:vpc/vpc-0df261dd0c6a4a5c3" -> null
  assign_generated_ipv6_cidr_block = false -> null
  cidr_block = "10.0.0.0/16" -> null
  default_network_acl_id = "acl-01533c343bbe12124" -> null
  default_route_table_id = "rtb-00d3cb62d2f041bb6" -> null
  default_security_group_id = "sg-02702f55bd1e2aaba" -> null
  dhcp_options_id = "dopt-009fbc5c9e45d0674" -> null
  enable_dns_hostnames = true -> null
  enable_dns_support = true -> null
  enable_network_address_usage_metrics = false -> null
  id = "vpc-0df261dd0c6a4a5c3" -> null
  instance_tenancy = "default" -> null
  ipv6_netmask_length = 0 -> null
  main_route_table_id = "rtb-00d3cb62d2f041bb6" -> null
  owner_id = "953868252369" -> null
  tags = {
    "Name" = "MyVPC"
  } -> null
  tags_all = {
    "Name" = "MyVPC"
  } -> null
}

Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

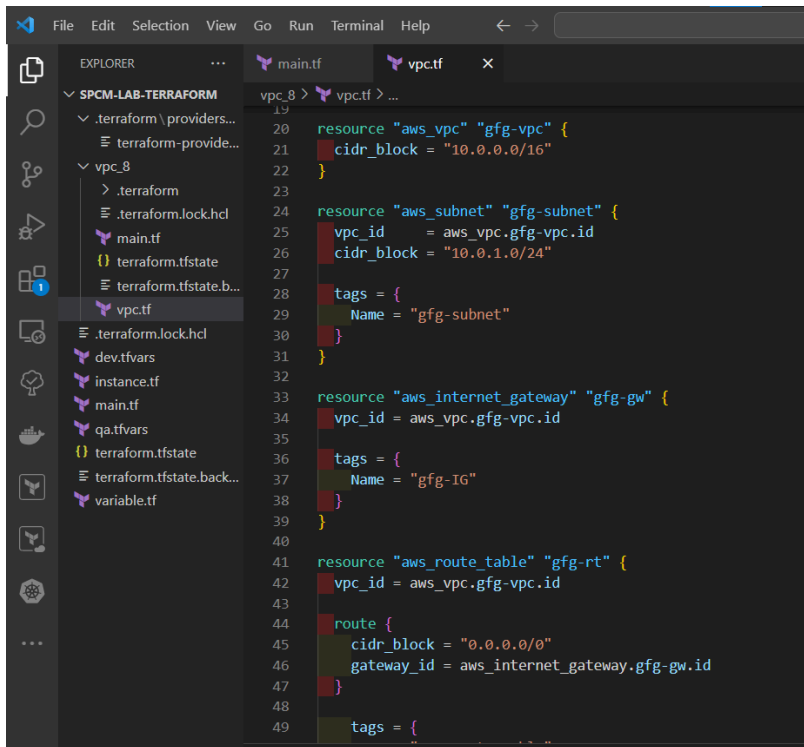
aws_subnet.my_subnet[0]: Destroying... [id=subnet-00fecef33341c030f]
aws_subnet.my_subnet[1]: Destroying... [id=subnet-01447d7d04cc270bf]
aws_subnet.my_subnet[1]: Destruction complete after 0s
aws_subnet.my_subnet[0]: Destruction complete after 0s
aws_vpc.my_vpc: Destroying... [id=vpc-0df261dd0c6a4a5c3]
aws_vpc.my_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpctest\

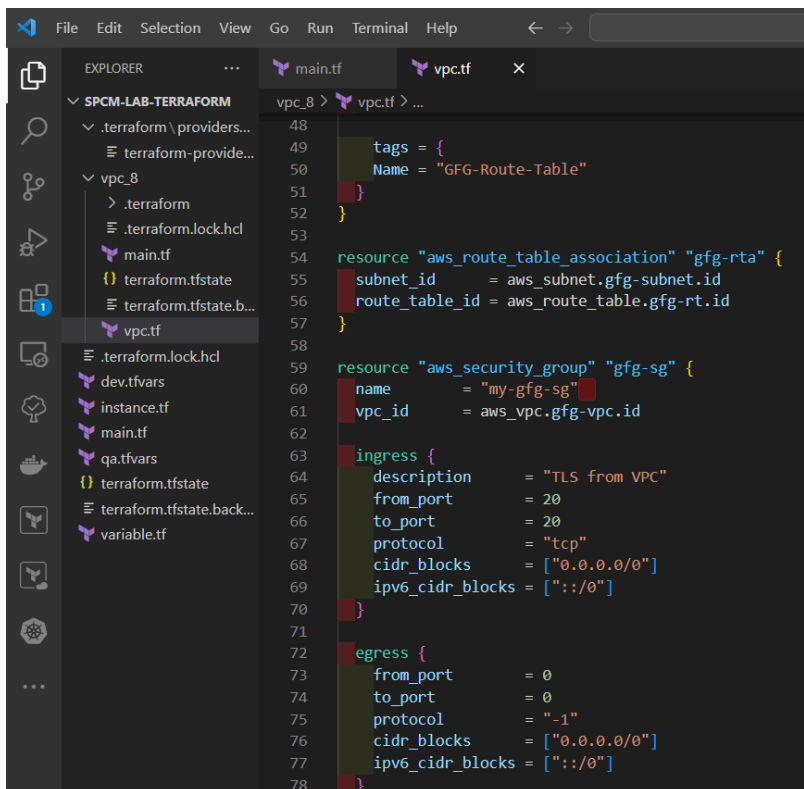
```

Step 8: Now update the vpc.tf file.



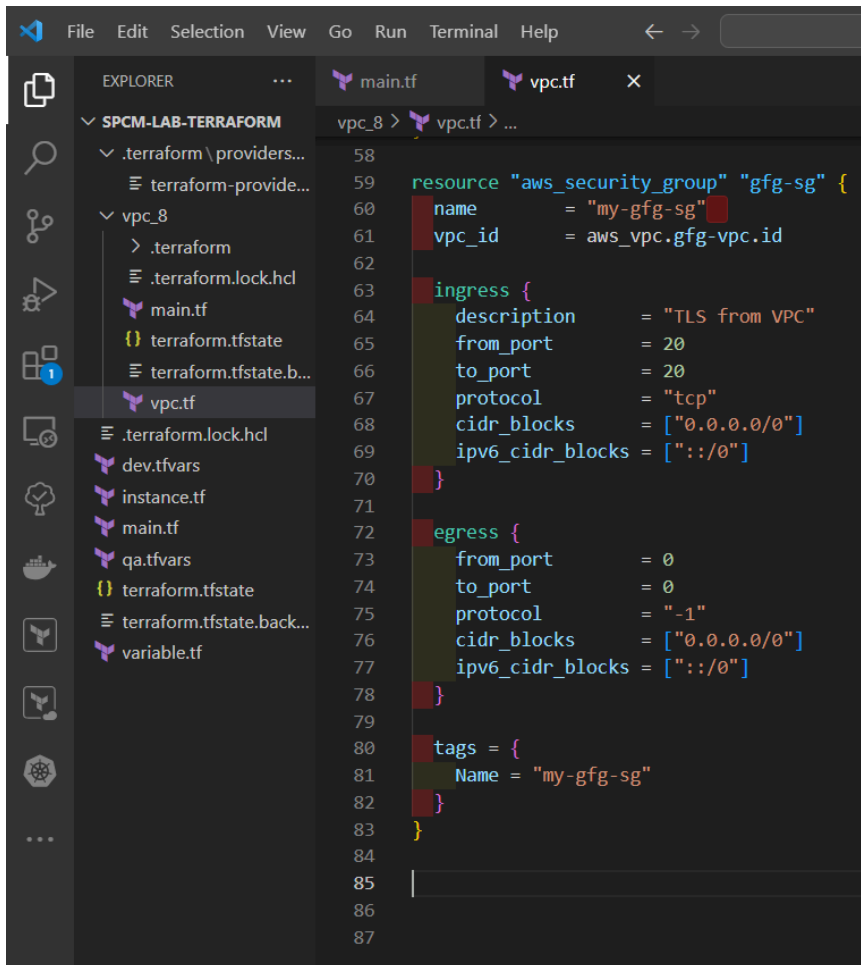
This screenshot shows the VS Code editor with the file explorer on the left displaying a project structure for 'SPCM-LAB-TERRAFORM'. The 'vpc.tf' file is selected and open in the editor. The code defines several AWS resources: an AWS VPC, an AWS Subnet, an AWS Internet Gateway, an AWS Route Table, and an AWS Route Table Association. The configuration includes CIDR blocks, VPC and Subnet IDs, and tags for each resource.

```
19 resource "aws_vpc" "gfg-vpc" {
20   cidr_block = "10.0.0.0/16"
21 }
22
23 resource "aws_subnet" "gfg-subnet" {
24   vpc_id     = aws_vpc.gfg-vpc.id
25   cidr_block = "10.0.1.0/24"
26 }
27
28 tags = {
29   Name = "gfg-subnet"
30 }
31
32 resource "aws_internet_gateway" "gfg-gw" {
33   vpc_id = aws_vpc.gfg-vpc.id
34 }
35
36 tags = {
37   Name = "gfg-IG"
38 }
39
40 resource "aws_route_table" "gfg-rt" {
41   vpc_id = aws_vpc.gfg-vpc.id
42 }
43
44 route {
45   cidr_block     = "0.0.0.0/0"
46   gateway_id     = aws_internet_gateway.gfg-gw.id
47 }
48
49 tags = {
```



This screenshot shows the continuation of the 'vpc.tf' file in the VS Code editor. The code defines an AWS Route Table Association and an AWS Security Group. The configuration includes subnet and route table IDs for the association, and a name, VPC ID, and ingress/egress rules for the security group.

```
48   tags = {
49     Name = "GFG-Route-Table"
50   }
51
52 resource "aws_route_table_association" "gfg-rta" {
53   subnet_id     = aws_subnet.gfg-subnet.id
54   route_table_id = aws_route_table.gfg-rt.id
55 }
56
57 resource "aws_security_group" "gfg-sg" {
58   name     = "my-gfg-sg"
59   vpc_id   = aws_vpc.gfg-vpc.id
60 }
61
62 ingress {
63   description     = "TLS from VPC"
64   from_port       = 20
65   to_port         = 20
66   protocol        = "tcp"
67   cidr_blocks     = ["0.0.0.0/0"]
68   ipv6_cidr_blocks = [ ":::/0" ]
69 }
70
71 egress {
72   from_port       = 0
73   to_port         = 0
74   protocol        = "-1"
75   cidr_blocks     = ["0.0.0.0/0"]
76   ipv6_cidr_blocks = [ ":::/0" ]
77 }
78 }
```



Step 9: Now again run the terraform run and validate command.

```
C:\Windows\System32\cmd.exe

F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpc_8>terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.31.0

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 10: Now again run the terraform plan command.


```

C:\Windows\System32\cmd.exe
PS:\sem 6\SPCH_LAB\spcm-lab-terraform\vpc_8\terraform plan
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_internet_gateway.gfg-ig will be created
+ resource "aws_internet_gateway" "gfg-ig" {
  + arn = (known after apply)
  + id = (known after apply)
  + owner_id = (known after apply)
  + tags = {
    + "Name" = "gfg-ig"
  }
  + tags_all = {
    + "Name" = "gfg-ig"
  }
  + vpc_id = (known after apply)
}

# aws_route_table.gfg-rt will be created
+ resource "aws_route_table" "gfg-rt" {
  + arn = (known after apply)
  + id = (known after apply)
  + owner_id = (known after apply)
  + propagating_vpus = (known after apply)
  + route = [
    {
      + carrier_gateway_id = ""
      + cidr_block = "0.0.0.0/0"
      + core_network_arn = ""
      + destination_prefix_list_id = ""
      + egress_only_gateway_id = ""
      + gateway_id = (known after apply)
      + ipv6_cidr_block = ""
      + local_gateway_id = ""
      + nat_gateway_id = ""
      + network_interface_id = ""
      + transit_gateway_id = ""
      + vpc_endpoint_id = ""
      + vpc_peering_connection_id = ""
    }
  ]
  + tags = {
    + "Name" = "GFG-Route-Table"
  }
  + tags_all = {
    + "Name" = "GFG-Route-Table"
  }
}

```

```

C:\Windows\System32\cmd.exe
+ tags = {
+   + "Name" = "GFG-Route-Table"
+ }
+ tags_all = {
+   + "Name" = "GFG-Route-Table"
+ }
+ vpc_id = (known after apply)

# aws_route_table_association.gfg-rta will be created
+ resource "aws_route_table_association" "gfg-rta" {
  + id = (known after apply)
  + route_table_id = (known after apply)
  + subnet_id = (known after apply)
}

# aws_security_group.gfg-sg will be created
+ resource "aws_security_group" "gfg-sg" {
  + arn = (known after apply)
  + description = "Managed by Terraform"
  + egress = [
    {
      + cidr_blocks = [
        + "0.0.0.0/0",
      ]
      + description = ""
      + from_port = 0
      + ipv6_cidr_blocks = [
        + "::/0",
      ]
      + prefix_list_ids = []
      + protocol = "-1"
      + security_groups = []
      + self = false
      + to_port = 0
    }
  ]
  + id = (known after apply)
  + ingress = [
    {
      + cidr_blocks = [
        + "0.0.0.0/0",
      ]
      + description = "TLS from VPC"
      + from_port = 20
      + ipv6_cidr_blocks = [
        + "::/0",
      ]
      + prefix_list_ids = []
      + protocol = "tcp"
    }
  ]
}

```

```

C:\Windows\System32\cmd.exe
}

# aws_subnet.gfg-subnet will be created
+ resource "aws_subnet" "gfg-subnet" {
  + arn = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone = (known after apply)
  + availability_zone_id = (known after apply)
  + cidr_block = "10.0.1.0/24"
  + enable_dns64 = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id = (known after apply)
  + ipv6_cidr_block_association_id = (known after apply)
  + ipv6_native = false
  + map_public_ip_on_launch = false
  + owner_id = (known after apply)
  + private_dns_hostname_type_on_launch = (known after apply)
  + tags = {
    + "Name" = "gfg-subnet"
  }
  + tags_all = {
    + "Name" = "gfg-subnet"
  }
  + vpc_id = (known after apply)
}

# aws_vpc.gfg-vpc will be created
+ resource "aws_vpc" "gfg-vpc" {
  + arn = (known after apply)
  + cidr_block = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id = (known after apply)
  + enable_dns_hostnames = (known after apply)
  + enable_dns_support = true
  + enable_network_address_usage_metrics = (known after apply)
  + id = (known after apply)
  + instance_tenancy = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id = (known after apply)
  + tags_all = (known after apply)
}

Plan: 6 to add, 0 to change, 0 to destroy.

```

Step 11: Now run the terraform apply command.

```

C:\Windows\System32\cmd.exe

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpc_8>terraform apply

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_internet_gateway.fg-gw will be created
+ resource "aws_internet_gateway" "fg-gw" {
  + arn = (known after apply)
  + id = (known after apply)
  + owner_id = (known after apply)
  + tags = {
    + "Name" = "fg-gw"
  }
  + tags_all = {
    + "Name" = "fg-gw"
  }
  + vpc_id = (known after apply)
}

# aws_route_table.fg-gw will be created
+ resource "aws_route_table" "fg-gw" {
  + arn = (known after apply)
  + id = (known after apply)
  + owner_id = (known after apply)
  + propagating_vpcs = (known after apply)
  + route = [
    + {
      + carrier_gateway_id = ""
      + cidr_block = "0.0.0.0/0"
      + core_network_arn = ""
      + destination_prefix_list_id = ""
      + egress_only_gateway_id = ""
      + gateway_id = (known after apply)
      + ipv6_cidr_block = ""
      + local_gateway_id = ""
      + nat_gateway_id = ""
      + network_interface_id = ""
      + transit_gateway_id = ""
      + vpc_endpoint_id = ""
      + vpc_peering_connection_id = ""
    },
  ]
  + tags = {
    + "Name" = "fg-gw-Route-Table"
  }
}

```

```

C:\Windows\System32\cmd.exe

+ vpc_id = (known after apply)

# aws_vpc.fg-gw will be created
+ resource "aws_vpc" "fg-gw" {
  + arn = (known after apply)
  + cidr_block = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id = (known after apply)
  + enable_dns_hostnames = (known after apply)
  + enable_dns_support = true
  + enable_network_address_usage_metrics = (known after apply)
  + id = (known after apply)
  + instance_tenancy = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id = (known after apply)
  + tags_all = (known after apply)
}

Plan: 6 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.fg-gw: Creating...
aws_vpc.fg-gw: Creation complete after 1s [id=vpc-037ceb0d28d510d10]
aws_internet_gateway.fg-gw: Creating...
aws_subnet.fg-gw: Creating...
aws_security_group.fg-gw: Creating...
aws_internet_gateway.fg-gw: Creation complete after 0s [id=igw-0017c36c9e5f43f2c]
aws_route_table.fg-gw: Creating...
aws_subnet.fg-gw: Creation complete after 0s [id=subnet-0fdcac6fc73c0dc8]
aws_route_table.fg-gw: Creation complete after 1s [id=rtb-012da00a9c1cec2f]
aws_route_table_association.fg-gw: Creating...
aws_route_table_association.fg-gw: Creation complete after 0s [id=rtbassoc-079ea0909bc41e8ee]
aws_security_group.fg-gw: Creation complete after 2s [id=sg-00923de6f5a1674b5]

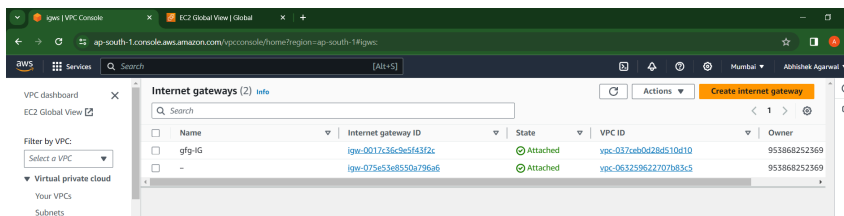
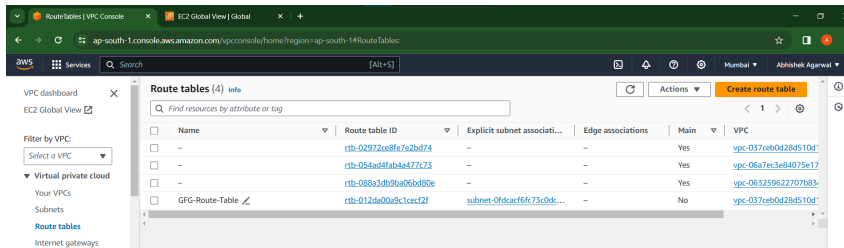
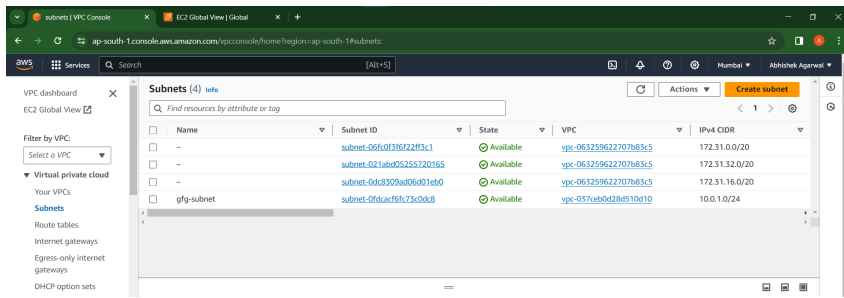
Apply complete! Resources: 6 added, 0 changed, 0 destroyed.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpc_8>

```

The screenshot shows the AWS VPC console interface. On the left, there's a sidebar with navigation links like 'VPC dashboard', 'EC2 Global View', 'Subnets', 'Route tables', and 'Internet gateways'. The main area is titled 'Your VPCs (3)' and contains a table with the following data:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
MyVPC	vpc-06a7ec3e84075e176	Available	10.0.0.0/16	-	dopt
-	vpc-037ceb0d28d510d10	Available	10.0.0.0/16	-	dopt
-	vpc-063259622707b83c5	Available	172.31.0.0/16	-	dopt



Step 12: Now run terraform destroy command.

```
C:\Windows\System32\cmd.exe

F:\sem 6\SPM LAB\spcm-lab-terraform\vpcc>terraform destroy -auto-approve
aws_vpc.gfg-vpc: Refreshing state... [id=vpc-037cebd28d510d10]
aws_internet_gateway.gfg-gw: Refreshing state... [id=igw-0017c36c9e5f43f2c]
aws_subnet.gfg-subnet: Refreshing state... [id=subnet-0fdcac6fc73cd0d8]
aws_security_group.gfg-sg: Refreshing state... [id=sg-00923de6f5a1674b5]
aws_route_table.gfg-rt: Refreshing state... [id=rtb-012da00a9c1cecf2f]
aws_route_table_association.gfg-rtas: Refreshing state... [id=rtbasoc-079ea0909b0c41e8ee]

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

Terraform will perform the following actions:

# aws_internet_gateway.gfg-gw will be destroyed
resource "aws_internet_gateway" "gfg-gw" {
  arn      = "arn:aws:ec2:ap-south-1:953868252369:internet-gateway/igw-0017c36c9e5f43f2c" -> null
  id       = "igw-0017c36c9e5f43f2c" -> null
  owner_id = "953868252369" -> null
  tags     = {
    "Name" = "gfg-IG"
  } -> null
  tags_all = {
    "Name" = "gfg-IG"
  } -> null
  vpc_id   = "vpc-037cebd28d510d10" -> null
}

# aws_route_table.gfg-rt will be destroyed
resource "aws_route_table" "gfg-rt" {
  arn      = "arn:aws:ec2:ap-south-1:953868252369:route-table/rtb-012da00a9c1cecf2f" -> null
  id       = "rtb-012da00a9c1cecf2f" -> null
  owner_id = "953868252369" -> null
  propagating_vpcs = [] -> null
  route      = [
    {
      carrier_gateway_id = ""
      cidr_block         = "0.0.0.0/0"
      core_network_arn   = ""
      destination_prefix_list_id = ""
      egress_only_gateway_id = ""
      gateway_id         = "igw-0017c36c9e5f43f2c"
      ipv6_cidr_block     = ""
      local_gateway_id    = ""
      nat_gateway_id      = ""
      network_interface_id = ""
      transit_gateway_id   = ""
      vpc_endpoint_id     = ""
    }
  ]
}
```

```

C:\Windows\System32\cmd.exe
- vpc_peering_connection_id = ""
} -> null
tags = {
  "Name" = "GFG-Route-Table"
} -> null
tags_all = {
  "Name" = "GFG-Route-Table"
} -> null
vpc_id = "vpc-037ceb0d28d510d10" -> null
}

# aws_route_table_association.gfg-rt will be destroyed
resource "aws_route_table_association" "gfg-rt-a" {
  id = "rtbassoc-079ea0909bc41e8ee" -> null
  route_table_id = "rtb-012da00a9c1cecf2f" -> null
  subnet_id = "subnet-0fdcac6fc73c0dc8" -> null
}

# aws_security_group.gfg-sg will be destroyed
resource "aws_security_group" "gfg-sg" {
  arn = "arn:aws:ec2:ap-south-1:953868252369:security-group/sg-00923de6f5a1674b5" -> null
  description = "Managed by Terraform" -> null
  egress = [
    {
      cidr_blocks = [
        "0.0.0.0/0",
      ]
      description = ""
      from_port = 0
      ipv6_cidr_blocks = [
        ":::/0",
      ]
      prefix_list_ids = []
      protocol = "tcp"
      security_groups = []
      self = false
      to_port = 0
    }
  ]
  id = "sg-00923de6f5a1674b5" -> null
  ingress = [
    {
      cidr_blocks = [
        "0.0.0.0/0",
      ]
      description = "TLS from VPC"
      from_port = 20
      ipv6_cidr_blocks = [
        ":::/0",
      ]
    }
  ]
}

```

```

C:\Windows\System32\cmd.exe
- tags = {
  "Name" = "gfg-subnet"
} -> null
tags_all = {
  "Name" = "gfg-subnet"
} -> null
vpc_id = "vpc-037ceb0d28d510d10" -> null
}

# aws_vpc.gfg-vpc will be destroyed
resource "aws_vpc" "gfg-vpc" {
  arn = "arn:aws:ec2:ap-south-1:953868252369:vpc/vpc-037ceb0d28d510d10" -> null
  assign_generated_ipv6_cidr_block = false -> null
  cidr_block = "10.0.0.0/16" -> null
  default_network_acl_id = "acl-016075efe8fd21be9" -> null
  default_route_table_id = "rtb-02972ce8fe7e2bd74" -> null
  default_security_group_id = "sg-0bb8c0682dc6e309d" -> null
  dhcp_options_id = "dopt-009fbc5c9e45d0674" -> null
  enable_dns_hostnames = false -> null
  enable_dns_support = true -> null
  enable_network_address_usage_metrics = false -> null
  id = "vpc-037ceb0d28d510d10" -> null
  instance_tenancy = "default" -> null
  ipv6_netmask_length = 0 -> null
  main_route_table_id = "rtb-02972ce8fe7e2bd74" -> null
  owner_id = "953868252369" -> null
  tags = {} -> null
  tags_all = {} -> null
}

```

```

Plan: 0 to add, 0 to change, 6 to destroy.
aws_route_table_association.gfg-rt-a: Destroying... [id=rtbassoc-079ea0909bc41e8ee]
aws_security_group.gfg-sg: Destroying... [id=sg-00923de6f5a1674b5]
aws_route_table_association.gfg-rt-a: Destruction complete after 0s
aws_subnet.gfg-subnet: Destroying... [id=subnet-0fdcac6fc73c0dc8]
aws_route_table.gfg-rt: Destroying... [id=rtb-012da00a9c1cecf2f]
aws_security_group.gfg-sg: Destruction complete after 0s
aws_subnet.gfg-subnet: Destruction complete after 0s
aws_route_table.gfg-rt: Destruction complete after 0s
aws_internet_gateway.gfg-gw: Destroying... [id=igw-0017c36c9e5f43f2c]
aws_internet_gateway.gfg-gw: Destruction complete after 1s
aws_vpc.gfg-vpc: Destroying... [id=vpc-037ceb0d28d510d10]
aws_vpc.gfg-vpc: Destruction complete after 0s

```

Destroy complete! Resources: 6 destroyed.

F:\sem 6\SPCM_LAB\spcm-lab-terraform\vpc_8>