

Lab Exercise 8

Creating a VPC in Terraform

1. Create a Terraform Directory:

```
PS E:\> mkdir terraform-vpc

Directory: E:\

Mode                LastWriteTime         Length Name
----                -
d-----          07-02-2024    09:48             terraform-vpc

PS E:\> cd .\terraform-variables\
PS E:\terraform-variables> |
```

2. Create a Terraform Configuration File:

```
main.tf  X  vpc.tf

main.tf > resource "aws_subnet" "my_subnet"
1  provider "aws" {
2    region = "us-east-2"
3    access_key = "AKIAVRUVV37F66GBPTT4"
4    secret_key = "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"
5  }
6  resource "aws_vpc" "my_vpc" {
7    cidr_block = "10.0.0.0/16"
8    enable_dns_support = true
9    enable_dns_hostnames = true
10   tags = {
11     Name = "MyVPC"
12   }
13 }
14 resource "aws_subnet" "my_subnet" [
15   count = 2
16   vpc_id = aws_vpc.my_vpc.id
17   cidr_block = "10.0.${count.index + 1}.0/24"
18   availability_zone = "us-east-2a"
19   map_public_ip_on_launch = true
20   tags = {
21     Name = "MySubnet-${count.index + 1}"
22   }
23 ]
```

```

vpc.tf > resource "aws_vpc" "gfg-vpc"
1  resource "aws_vpc" "gfg-vpc" {
2    cidr_block = "10.0.0.0/16"
3  }
4
5  resource "aws_subnet" "gfg-subnet" {
6    vpc_id     = aws_vpc.gfg-vpc.id
7    cidr_block = "10.0.1.0/24"
8
9    tags = {
10     Name = "gfg-subnet"
11   }
12 }
13
14 resource "aws_internet_gateway" "gfg-gw" {
15   vpc_id = aws_vpc.gfg-vpc.id
16
17   tags = {
18     Name = "gfg-IG"
19   }
20 }
21
22 resource "aws_route_table" "gfg-rt" {
23   vpc_id = aws_vpc.gfg-vpc.id
24
25   route {
26     cidr_block = "0.0.0.0/0"
27     gateway_id = aws_internet_gateway.gfg-gw.id
28   }
29
30   tags = {
31     Name = "GFG-Route-Table"
32   }
33 }
34
35 resource "aws_route_table_association" "gfg-rta" {
36   subnet_id     = aws_subnet.gfg-subnet.id
37   route_table_id = aws_route_table.gfg-rt.id

```

3. Initialize and Apply:

```
PS E:\terraform-vpc> terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.35.0

```
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.
```

```
Windows PowerShell
PS E:\terraform-vpc> terraform apply
aws_vpc.gfg-vpc: Refreshing state... [id=vpc-09dae9480d70e3d6e]
aws_vpc.my_vpc: Refreshing state... [id=vpc-0f425492c15f287b7]
aws_internet_gateway.gfg-gw: Refreshing state... [id=igw-0a11a3cc518a3f507]
aws_subnet.gfg-subnet: Refreshing state... [id=subnet-0c94b7f813e5d6084]
aws_security_group.gfg-sg: Refreshing state... [id=sg-01a863bf1ce13d188]
aws_route_table.gfg-rt: Refreshing state... [id=rtb-09782cc807b3539bb]
aws_route_table_association.gfg-rta: Refreshing state... [id=rtbassoc-09addaa6e656eebdc]

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-2a"
  + 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-0f425492c15f287b7"
}
```

Your VPCs (3) Info

Q Search

< 1 > ⚙

⏻

Actions ▾

Create VPC

<input type="checkbox"/>	Name ▾	VPC ID ▾	State ▾	IPv4 CIDR ▾	IPv6 CIDR ▾	DHCP
<input type="checkbox"/>	-	vpc-09dae9480d70e3d6e	Available	10.0.0.0/16	-	dopt-
<input type="checkbox"/>	-	vpc-00fbc92b09849315e	Available	172.31.0.0/16	-	dopt-
<input type="checkbox"/>	MyVPC	vpc-0f425492c15f287b7	Available	10.0.0.0/16	-	dopt-

4. Clean Up:

```
Windows PowerShell
- ipv6_netmask_length           = 0 -> null
- main_route_table_id          = "rtb-0f24e16e7b8882fe5" -> null
- owner_id                     = "381492256715" -> null
- tags                         = {
  - "Name" = "MyVPC"
} -> null
- tags_all                     = {
  - "Name" = "MyVPC"
} -> null
}

Plan: 0 to add, 0 to change, 9 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_route_table_association.gfg-rta: Destroying... [id=rtbassoc-09addaa6e656eebdc]
aws_subnet.my_subnet[1]: Destroying... [id=subnet-017c02736417362b9]
aws_security_group.gfg-sg: Destroying... [id=sg-01a863bf1ce13d188]
aws_subnet.my_subnet[0]: Destroying... [id=subnet-0fbf8561c46687cf6]
aws_route_table_association.gfg-rta: Destruction complete after 3s
aws_subnet.my_subnet[0]: Destruction complete after 3s
aws_subnet.gfg-subnet: Destroying... [id=subnet-0c94b7f813e5d6084]
aws_route_table.gfg-rt: Destroying... [id=rtb-09782cc807b3539bb]
aws_subnet.my_subnet[1]: Destruction complete after 3s
aws_vpc.my_vpc: Destroying... [id=vpc-0f425492c15f287b7]
aws_security_group.gfg-sg: Destruction complete after 3s
aws_subnet.gfg-subnet: Destruction complete after 1s
aws_route_table.gfg-rt: Destruction complete after 1s
aws_internet_gateway.gfg-gw: Destroying... [id=igw-0a11a3cc518a3f507]
aws_vpc.my_vpc: Destruction complete after 2s
aws_internet_gateway.gfg-gw: Destruction complete after 2s
aws_vpc.gfg-vpc: Destroying... [id=vpc-09dae9480d70e3d6e]
aws_vpc.gfg-vpc: Destruction complete after 2s

Destroy complete! Resources: 9 destroyed.
PS E:\terraform-vpc> |
```

Your VPCs (1) Info

Q Search

< 1 > ⚙

▼	DHCP option set ▼	Main route table ▼	Main network ACL ▼	Tenancy ▼	Default VPC ▼	Owner ID ▼
	dopt-0cb162946e046f3...	rtb-037599cfc1712efe6	acl-025607b0d1978a2f0	Default	Yes	381492256715