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



**System Monitoring and Configuration
Management**

Lab File

(2024)

for

6th Semester

Submitted To:

Dr. Hitesh Kumar Sharma

Submitted By:

Eksha Malhotra
B. Tech. CSE DevOps
[6^h Semester]
Sap id- 500091532
Batch 1
R2142210300

Lab Exercise 8

Aim: Creating a VPC in Terraform Objective:

1. Create a Terraform Directory:

```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\eksha>mkdir terraform-VPC
A subdirectory or file terraform-VPC already exists.

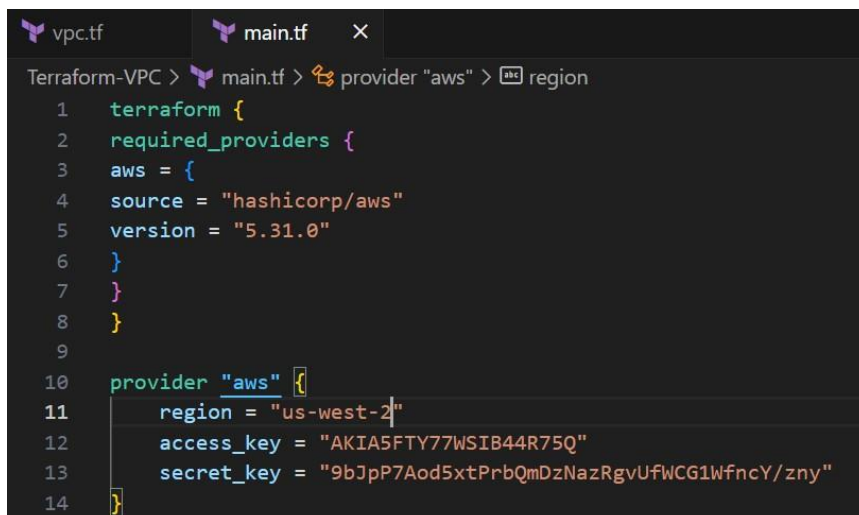
C:\Users\eksha>cd terraform-VPC

C:\Users\eksha\terraform-vpc>|
```

2. Create Terraform Configuration Files:

- Create a file named

main.tf:# main.tf



The screenshot shows a code editor with two tabs: 'vpc.tf' and 'main.tf'. The 'main.tf' tab is active, displaying the following Terraform configuration code:

```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "us-west-2"
12   access_key = "AKIA5FTY77WSIB44R75Q"
13   secret_key = "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"
14 }
```

#vpc.tf

```
vpc.tf  X  main.tf
Terraform-VPC > vpc.tf > resource "aws_subnet" "my_subnet" > cidr_block
1  resource "aws_vpc" "my_vpc" {
2  cidr_block = "10.0.0.0/16"
3  enable_dns_support = true
4  enable_dns_hostnames = true
5  tags = {
6  Name = "MyVPC"
7  }
8  }
9
10 resource "aws_subnet" "my_subnet" {
11     count = 2
12     vpc_id      = aws_vpc.my_vpc.id
13     cidr_block  = "10.0.${count.index + 1}.0/24"
14     availability_zone = "us-west-2a"
15     map_public_ip_on_launch = true
16
17     tags = {
18         Name = "MySubnet-${count.index + 1}"
19     }
20 }
```

3. Initialize and Apply:

```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-VPC> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.31.0"...
- Installing hashicorp/aws v5.31.0...
- Installed hashicorp/aws v5.31.0 (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.
```

```

PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-VPC> terraform apply
aws_vpc.my_vpc: Refreshing state... [id=vpc-083b0a0224fe987cd]

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-west-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"
  }
}

```

4. Verify Resources in AWS Console:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0ae7ae5898c67a19d	Available	172.31.0.0/16	-

Name	Subnet ID	State	VPC
-	subnet-0b2394b746b64bcd8	Available	vpc-0ae7ae5898c67a19d
-	subnet-0480d5ce483aa80b6	Available	vpc-0ae7ae5898c67a19d
-	subnet-0f4b77adc91c55a33	Available	vpc-0ae7ae5898c67a19d

5. Clean Up

```
PS C:\Desktop\DevOps\Sem6\SMCP\Lab Files\TERRAFORM LAB SCRIPTS\Terraform-VPC> terraform destroy
aws_vpc.my_vpc: Refreshing state... [id=vpc-083b0a0224fe987cd]
aws_subnet.my_subnet[1]: Refreshing state... [id=subnet-01b3c786513ee92ff]
aws_subnet.my_subnet[0]: Refreshing state... [id=subnet-06b85fe8723a16277]

Terraform used the selected providers to generate the following execution plan. Resource actions
- destroy

Terraform will perform the following actions:

# aws_subnet.my_subnet[0] will be destroyed
- resource "aws_subnet" "my_subnet" {
  - arn                                = "arn:aws:ec2:us-west-2:905418112420:sub
  - assign_ipv6_address_on_creation    = false -> null
  - availability_zone                  = "us-west-2a" -> null
  - availability_zone_id                = "usw2-az1" -> null
  - cidr_block                         = "10.0.1.0/24" -> null
  - enable_dns64                       = false -> null
  - enable_lni_at_device_index         = 0 -> null
  - enable_resource_name_dns_a_record_on_launch = false -> null
  - enable_resource_name_dns_aaaa_record_on_launch = false -> null
```