

Lab Exercise 11– Creating a

Objective:

Objective:

Learn how to use Terraform to create a basic Virtu

Prerequisites:

- Terraform installed on your machine.
- AWS CLI configured with the necessary creden

Steps:

1. Create a Terraform Directory:

```
[ujjwal@ujjwalmacbook Terraform-variables % mkdir Terraform-vpc  
[ujjwal@ujjwalmacbook Terraform-variables % cd Terraform-vpc  
[ujjwal@ujjwalmacbook Terraform-vpc % touch main8.tf  
[ujjwal@ujjwalmacbook Terraform-vpc % vim main8.tf
```

- Create Terraform Configuration Files:
- Create a file named main.tf:

```
resource "aws_vpc" "gfg-vpc" {
  cidr_block = "10.0.0.0/16"
  resource "aws_subnet" "gfg-subnet" {
    vpc_id = aws_vpc-gfg-vpc.id
    cidr_block = "10.0.1.0/24"
    tags = {
      Name = "gfg-subnet"
    }
    resource "aws_internet_gateway" "gfg-gw" {
      vpc_id = aws_vpc-gfg-vpc.id
      tags = {
        Name = "gfg-IG"
      }
    }
    resource "aws_route_table" "gfg-rt" {
      vpc_id = aws_vpc.gfg-vpd
    }
  }
}
```

Verify and apply

Terraform init terraform apply

```
(base) → terraform-vpc-lab11 git:(main) ✕ terraform i
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.68.0" ...
- Installing hashicorp/aws v5.68.0...
- Installed hashicorp/aws v5.68.0 (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 system so that Terraform can guarantee to make the same selections
you run "terraform init" in the future.
```

Terraform has been successfully initialized!

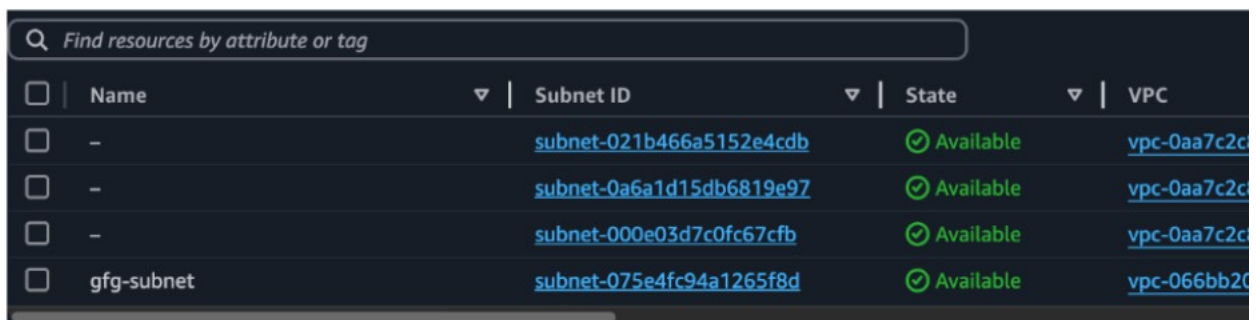
You may now begin working with Terraform. Try running `terraform apply` to make any changes that are required for your infrastructure. Terraform should now work.

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

```
(base) → terraform-vpc-lab11 git:(main) ✕
```

3. Verify Resources in AWS Console

- Log in to the AWS Management Console and navigate to the VPC console.
- Verify that the VPC and subnets with the specified tags are created.



The screenshot shows the AWS VPC console interface. At the top, there is a search bar with the placeholder text "Find resources by attribute or tag". Below the search bar is a table listing subnets. The table has four columns: "Name", "Subnet ID", "State", and "VPC". Each row represents a subnet and includes a checkbox for selection. The subnets listed are: a subnet with ID "subnet-021b466a5152e4cdb", a subnet with ID "subnet-0a6a1d15db6819e97", a subnet with ID "subnet-000e03d7c0fc67cfb", and a subnet named "gfg-subnet" with ID "subnet-075e4fc94a1265f8d". All subnets are in the "Available" state and are associated with the VPC "vpc-0aa7c2c".

<input type="checkbox"/>	Name	Subnet ID	State	VPC
<input type="checkbox"/>	-	subnet-021b466a5152e4cdb	Available	vpc-0aa7c2c
<input type="checkbox"/>	-	subnet-0a6a1d15db6819e97	Available	vpc-0aa7c2c
<input type="checkbox"/>	-	subnet-000e03d7c0fc67cfb	Available	vpc-0aa7c2c
<input type="checkbox"/>	gfg-subnet	subnet-075e4fc94a1265f8d	Available	vpc-066bb20