Terraform AWS EC2 with VPC and Subnet Data Sources - Detailed Documentation

This document explains the Terraform configuration provided for deploying an AWS EC2 instance using existing VPC and Subnet information retrieved via Terraform **data sources**.

1. AWS VPC Data Source:
The `data` block for `aws_vpc` retrieves details about an existing VPC based on its tag name. This is useful when you don't want Terraform to create a new VPC, but instead use an already provisioned one.

```
data "aws_vpc" "demo-vpc" {
    filter {
      name   = "tag:Name"
      values = ["my-vpc"]
    }
}
```

2. AWS Subnet Data Source:
The `data` block for `aws_subnet` retrieves details about an existing subnet using its tag name. This allows the EC2 instance to be placed in a specific subnet without hardcoding its ID.

```
data "aws_subnet" "subnet_1" {
    filter {
      name   = "tag:Name"
      values = ["subnet-1"]
    }
}
```

3. AWS EC2 Instance Resource:
This block creates an EC2 instance inside the retrieved subnet.
- `ami`: The Amazon Machine Image ID.
- `instance_type`: The type of instance.
- `subnet_id`: Uses the `id` attribute from the `aws_subnet` data source.
- `tags`: Key-value pairs for instance metadata.

```
resource "aws_instance" "my-ec2" {
    ami           = "ami-084e7e1456028650e"
    instance_type = "t2.micro"
    subnet_id     = data.aws_subnet.subnet_1.id
    tags = {
      "Name": "first-ec2"
    }
}
```

Notes:
1. Ensure that the VPC and Subnet already exist in AWS and have the correct tags.
2. Data sources in Terraform are used for reading information without creating or modifying the resource.
3. The `subnet_id` in the EC2 resource is dynamically populated from the data source, making the configuration more reusable.
4. Commands to execute:
- `terraform init`
- `terraform plan`
- `terraform apply`