Terraform AWS EC2 with S3 Backend - Detailed Documentation

This document explains in detail the Terraform configuration provided for deploying an AWS EC2 instance and using an S3 bucket as the backend for storing the Terraform state file.

1. AWS Provider Configuration:
The `provider` block defines the AWS provider and specifies the region where resources will be created.
Example:

```
provider "aws" {
    region = "ap-south-1"
}
```

2. Terraform Backend Configuration:
The backend block configures where Terraform stores its state file. Here, the S3 backend is used.
- `bucket`: Name of the S3 bucket.
- `key`: Path within the bucket where the state file will be stored.
- `region`: AWS region of the S3 bucket.

```
terraform {
  backend "s3" {
    bucket = "8764rtyibf"
    key    = "resources/terraform.tfstate"
    region = "ap-south-1"
  }
}
```

3. AWS EC2 Instance Resource:
This block creates an AWS EC2 instance.
- `ami`: The Amazon Machine Image ID.
- `instance_type`: The type of instance (t2.micro is free-tier eligible).

```
resource "aws_instance" "second-ec2" {
    ami           = "ami-084e7e1456028650e"
    instance_type = "t2.micro"
}
```

Notes:
1. Ensure AWS credentials are configured via environment variables, shared credentials file, or AWS SSO.
2. The S3 bucket must exist before running `terraform init`.
3. Run the following commands in sequence:
- `terraform init` (Initialize Terraform and configure backend)
- `terraform plan` (Preview the changes)
- `terraform apply` (Apply the changes)
4. Using an S3 backend ensures state locking (with DynamoDB) and centralized state management.