**🚀 Terraform Project: Deploying EC2 with Remote Backend (S3 + DynamoDB)**

**📌 Project Objective**

To provision an **AWS EC2 instance** using **Terraform** with a **Remote Backend** configured via **S3** (for storing the terraform.tfstate file) and **DynamoDB** (for state locking to prevent concurrent runs). This setup ensures **secure, collaborative, and scalable infrastructure management**.

**1 . Create Ec2 :**

**main.tf :**  
  
provider "aws" {

    region = "ap-south-1"

}

resource "aws\_instance" "ec2" {

    instance\_type = "t2.micro"

    ami = "ami-02521d90e7410d9f0"

    tags = {

      name ="terraform-ec2"

    }

}

**2. Create S3:**  
  
**s3.tf :**

resource "aws\_s3\_bucket" "s3\_bucket" {

    bucket = "bucket-for-remote-backend"

    tags = {

      name = "bucket-for-remote-backend"

    }

}

**3. create dynamodb table :**  
  
**dynamodb.tf :**

resource "aws\_dynamodb\_table" "basic-dynamodb-table" {

  name           = "dynamo\_remote\_backend\_table"

  billing\_mode   = "PAY\_PER\_REQUEST"                         #PROVISIONED = always active therefore cost is high

                                                             #PAY\_PER\_REQUEST = active only when requires less costing

  hash\_key       = "LockID"                                  # requires for lock and release mechanism

attribute {

  name = "LockID"

  type = "S"                                                   # S = String

}

  tags = {

    Name        = "dynamo\_remote\_backend\_table"

  }

}

**4. Then give command** : 1 . terraform init

2 . terraform plan

3 . terraform apply

Now your EC2 , S3 bucket , Dynamodb table is created. Now go to 5th step for lock mechanism.

**5. Create remote backend with locking mechanism:**  
  
**backend.tf :**

terraform {

  backend "s3" {

    bucket = "bucket-for-remote-backend"

    key    = "krishna/terraform.tfstate"

    region = "ap-south-1"

    dynamodb\_table = "dynamo\_remote\_backend\_table"

  }

}

6. **. Then give command** : 1 . terraform init

2 . terraform plan

3 . terraform apply