Terraform with AWS  
  
🔷 Provisioning on AWS is quite easy and straightforward with Terraform.  
  
🔷 AWS CLI installed  
The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.  
  
🛠 Below is an explanation of the key elements:  
  
🔸 1. AWS CLI Configuration:  
-----------------------------------------------------  
↪ export AWS\_ACCESS\_KEY\_ID=<access key>  
↪ export AWS\_SECRET\_ACCESS\_KEY=<secret access key>  
-----------------------------------------------------  
This exports your AWS access key ID and secret access key as environment variables. Ensure you replace ' <access key> ' and ' <secret access key> ' with your actual AWS access credentials.  
  
🔸 2. Terraform Configuration:  
-----------------------------------------------------  
terraform {  
 required\_providers {  
 aws = {  
 source = "hashicorp/aws"  
 version = "~> 4.16"  
 }  
 }  
 required\_version = ">= 1.2.0"  
}  
-----------------------------------------------------  
This specifies the required provider (AWS in this case) and its version.  
  
🔸 3. AWS Provider Configuration:  
-----------------------------------------------------  
provider "aws" {  
 region = "us-east-1"  
}  
-----------------------------------------------------  
This configures the AWS provider with the desired region (' us-east-1 ' in this case).  
  
🔸 4. Provisioning EC2 Instances:  
-----------------------------------------------------  
resource "aws\_instance" "aws\_ec2\_test" {  
 count     = 4  
 ami      = "ami-08c40ec9ead489470"  
 instance\_type = "t2.micro"  
 tags = {  
 Name = "TerraformTestServerInstance"  
 }  
}  
-----------------------------------------------------  
This Terraform script creates an AWS EC2 instance using the specified Amazon Machine Image (AMI), instance type (' t2.micro '), and tags. The 'count' attribute is set to 4, meaning it will create 4 instances.  
  
🛠 How to use this script:  
  
🔹 1. Save the script to a file, for example, ' main .tf '.  
  
🔹 2. Open a terminal and navigate to the directory containing the script.  
  
🔹 3. Run the following commands:  
-----------------------------------------------------  
↪ terraform init # Initialize the Terraform configuration  
↪ terraform apply # Apply the configuration to create AWS resources  
-----------------------------------------------------  
  
🔹 4. Terraform will prompt you to confirm the changes. Type ' yes ' to proceed.  
  
🔹 5. Once the instances are provisioned, you can check your AWS EC2 Console to see the created instances.  
  
💡 Remember to run ' terraform destroy ' when you're done testing to avoid unnecessary charges. This will destroy the created AWS resources.

