Terraform Basic

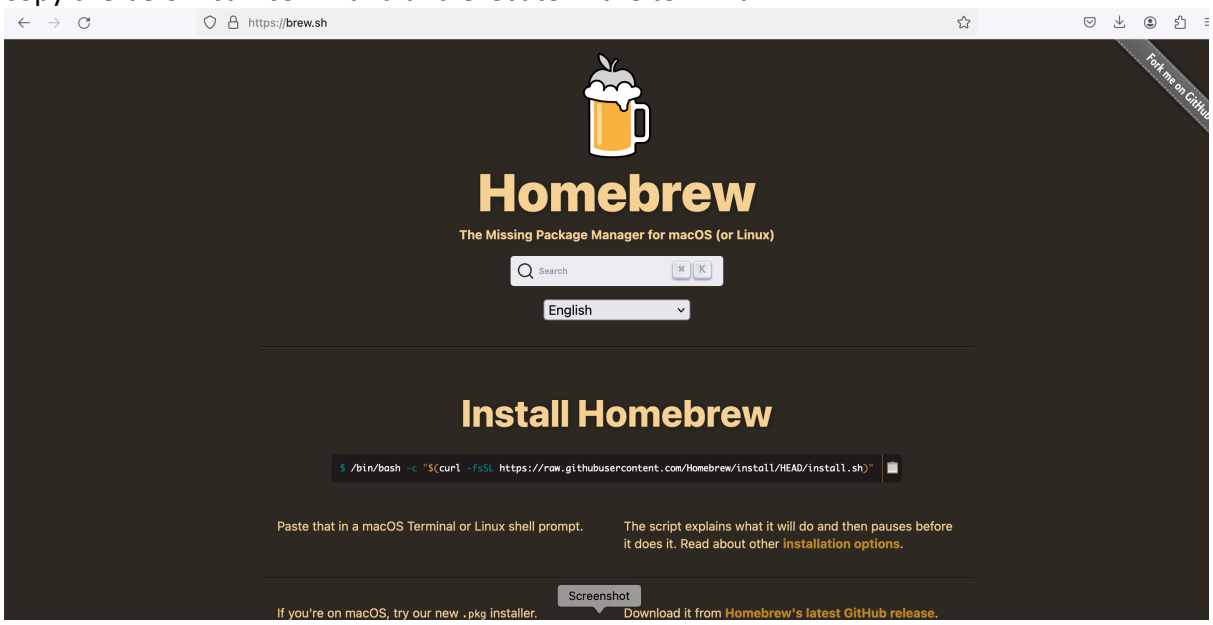
In Mac

Open the terminal in the mac

Go to browser

brew.sh

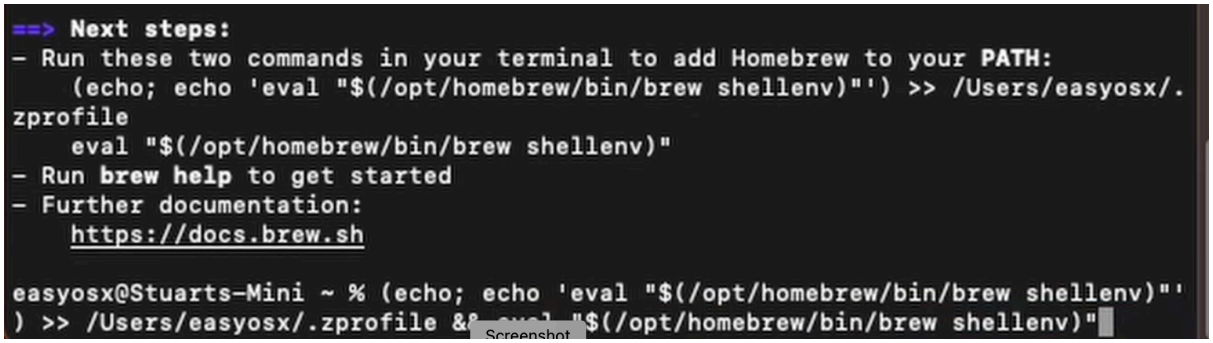
copy the below curl command and execute in the terminal



search for terraform and use the below command to install terraform

brew install iam-policy-json-to-terraform

Use the below two commands to go forward



Note: (echo ….. && eval…) and execute the command as shown in SS

check for terraform version

terraform -version

use the below command to access key and password

aws configure

Acces key

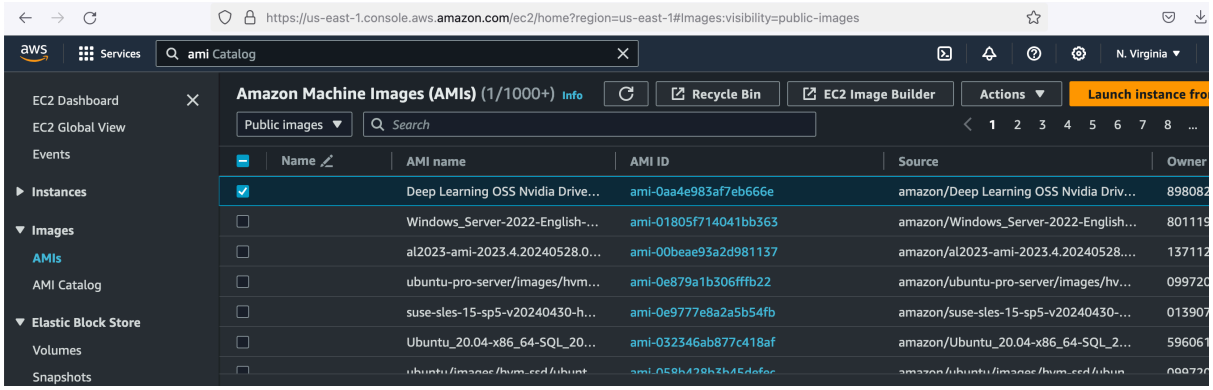
[redacted]

Create an mkdir file and switch to it
Mkdir terraform
Create an vi main.tf use the below

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.16"
    }
  }

  required_version = ">= 1.2.0"
}

provider "aws" {
  region  = "us-east-1"
}

resource "aws_instance" "app_server" {
  ami           = "ami-0aa4e983af7eb666e"
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform_Demo"
  }
}
```

Note: Need to change the below

Region as per the aws console
Ami available below



Copy them as per the required and use them

Use the below commands to create terraform

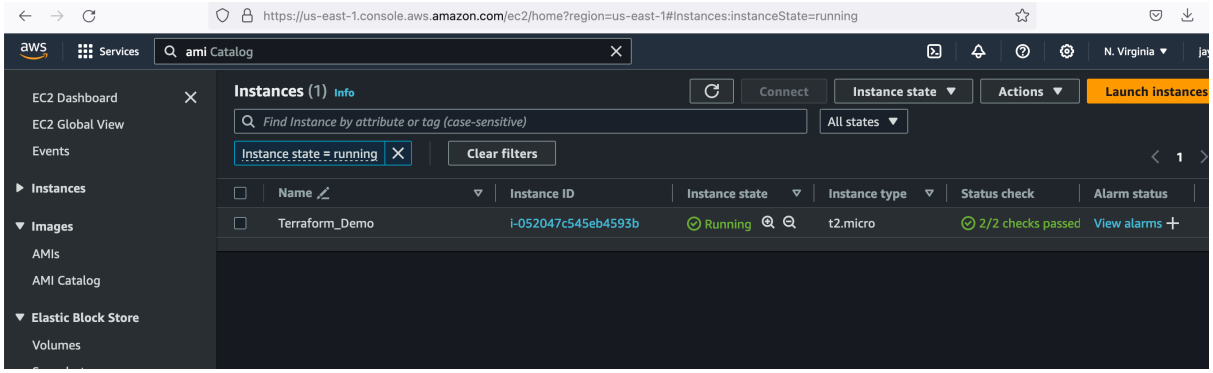
Terraform init

```
mohan@Mohans-MacBook-Air terraform % terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.67.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
mohan@Mohans-MacBook-Air terraform %
```

Terraform plan

Terraform apply

Go to aws console and check for instance



Terraform destroy

The above command will terminate all the instance which are created

Done

Terraform Advance

By me

mkdir terraform-s3-dynamodb
cd terraform-s3-dynamodb

Created 3 files

```
mohan@Mohans-MacBook-Air terraform-s3-dynamodb % ls -ltr
total 40
-rw-r--r--  1 mohan  staff   244 12 Jun 09:47 outputs.tf
-rw-r--r--  1 mohan  staff   489 12 Jun 10:06 main.tf
-rw-r--r--  1 mohan  staff   407 12 Jun 10:07 variables.tf
```

Main.tf

Note: Please change the region as per the required

```
provider "aws" {
  region = "us-west-2"
}

# S3 Bucket
resource "aws_s3_bucket" "my_bucket" {
  bucket = "my-unique-bucket-name-12345"
  acl    = "private"

  tags = {
    Name        = "MyBucket"
```

```
    Environment = "Dev"
  }
}

# DynamoDB Table
resource "aws_dynamodb_table" "my_table" {
  name           = "my-table"
  billing_mode   = "PAY_PER_REQUEST"
  hash_key       = "ID"
  attribute {
    name = "ID"
    type = "S"
  }

  tags = {
    Name        = "MyDynamoDBTable"
    Environment = "Dev"
  }
}

output "s3_bucket_id" {
  value = aws_s3_bucket.my_bucket.id
}

output "dynamodb_table_name" {
  value = aws_dynamodb_table.my_table.name
}
```

Variable.tf

Note: Please change the region as per the required

```
variable "region" {
  description = "The AWS region to deploy resources"
  type        = string
  default     = "us-west-2"
}

variable "s3_bucket_name" {
  description = "The name of the S3 bucket"
  type        = string
  default     = "my-unique-bucket-name-12345"
}

variable "dynamodb_table_name" {
  description = "The name of the DynamoDB table"
  type        = string
```

```
  default     = "my-table"
}
```

outputs.tf

Note: Please change the region as per the required

```
output "s3_bucket_id" {
  description = "The ID of the S3 bucket"
  value       = aws_s3_bucket.my_bucket.id
}

output "dynamodb_table_name" {
  description = "The name of the DynamoDB table"
  value       = aws_dynamodb_table.my_table.name
}
```

Use the below to proceed further

terraform init

terraform validate

terraform plan

terraform apply

Done

End of the session use

terraform destroy