Hands-on with AWS and Terraform

Terraform Configuration File (main.tf)

```
terraform { required_providers { aws = { source = "hashicorp/aws" version = "~> 4.16" } } required_version = ">= 1.2.0" } provider "aws" { region = "us-east-1" } resource "aws_s3_bucket" "s3_bucket" { bucket = "tcb-app-qa-jr" } resource "aws_s3_bucket_public_access_block" "s3_block" { bucket = aws_s3_bucket.s3_bucket.id block_public_acls = true block_public_policy = true ignore_public_acls = true restrict_public_buckets = true }
```

Steps

- Create folder cloud-warmup
- Create folder live2-terraform-aws
- Open the cloud-warmup folder on VS Code
- Create the file main.tf
- Create the configuration to deploy an S3 bucket

```
terraform { required_providers { aws = { source = "hashicorp/aws" version = "~> 4.16" } } required_version = ">= 1.2.0" } provider "aws" { region = "us-east-1" } resource "aws_s3_bucket" "s3_bucket" { bucket = "tcb-app-qa-jr" }
```

- Login to AWS, open the Cloud Shell
- Install Terraform https://learn.hashicorp.com/tutorials/terraform/install-cli?in=terraform/aws-get-started
- Upload the main.tf file to the Cloud Shell
- Init, plan and apply to create the S3

```
terraform init terraform plan terraform apply
```

- Add the configuration to prevent public access on S3

```
resource "aws_s3_bucket_public_access_block" "s3_block" { bucket =
aws_s3_bucket.s3_bucket.id block_public_acls = true block_public_policy =
true ignore_public_acls = true restrict_public_buckets = true }
```

Troubleshooting:

If the Cloud Shell AWS CLI token session expires, run any AWS CLI to refresh it. Example: aws s3 ls s3://tcb-app-qa-jr