terraform

Este projeto utiliza o Terraform para provisionar uma infraestrutura básica na AWS de forma automatizada. Ele cria uma VPC (Virtual Private Cloud), uma instância EC2 e um bucket S3. A estrutura modular do projeto facilita a organização, reutilização e personalização do código, seguindo as melhores práticas de Infraestrutura como Código (IaC).

Funcionalidades Principais

1. **VPC (Virtual Private Cloud)**:
    - Cria uma VPC com CIDR customizável.
    - Define sub-redes públicas em zonas de disponibilidade selecionadas.
    - Implementa uma tabela de roteamento para acesso à internet via Internet Gateway.
2. **EC2 (Elastic Compute Cloud)**:
    - Provisiona uma instância EC2 dentro da VPC.
    - Utiliza a AMI especificada para inicialização da instância.
    - Define tipo de instância e chave SSH para acesso seguro.

    Funcionalidades Adicionais:

    - **Apache e PHP em Docker na Instância EC2:**
        - O Docker é instalado na instância EC2.
        - Um container Docker é executado com Apache e PHP pré-configurados.
        - A porta 80 é aberta no grupo de segurança da instância para acesso ao servidor web.
    - **PostgreSQL 13 na Instância EC2:**
        - O PostgreSQL 13 é instalado na instância EC2.

    - A porta 5432 é aberta no grupo de segurança da instância para acesso ao banco de dados.

3. **S3 (Simple Storage Service)**:
    - Cria um bucket S3.

Uso e Execução

Passos para utilizar o projeto:

1. **Inicialização do Terraform**:
    - Execute `terraform init` no diretório raiz para inicializar o ambiente e baixar plugins.
2. **Planejamento das Alterações**:
    - Execute `terraform plan` para visualizar as modificações que serão feitas na infraestrutura AWS.
3. **Aplicação das Alterações**:
    - Execute `terraform apply` para criar os recursos na sua conta AWS. Confirme digitando `yes`.
4. **Gerenciamento da Infraestrutura**:
    - Utilize `terraform destroy` para remover todos os recursos provisionados, quando não forem mais necessários.

Resultado Final

```
Apply complete! Resources: 10 added, 0 changed, 0 destroyed.

Outputs:

bucket_name = "backup-mariusso-teste"
instance_id = "i-0311301bc75303f21"
instance_public_ip = "52.87.221.133"
public_subnet_ids = [
  "subnet-0e2c6854c8440fda0",
  "subnet-0e544b0ff58aebcfc",
]
vpc_id = "vpc-0639e951d3bbb2706"
➜  aws-infra
```

```
ubuntu@ip-10-0-1-50:~$ sudo service postgresql status
● postgresql.service - PostgreSQL RDBMS
     Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)
     Active: active (exited) since Fri 2024-06-14 21:21:12 UTC; 53s ago
   Main PID: 3296 (code=exited, status=0/SUCCESS)
        CPU: 2ms

Jun 14 21:21:12 ip-10-0-1-50 systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Jun 14 21:21:12 ip-10-0-1-50 systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
ubuntu@ip-10-0-1-50:~$ sudo docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                   NAMES
8f795c7578fb   apache-php-image   "docker-php-entrypoi…"   20 seconds ago   Up 20 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   distracted_brattain
ubuntu@ip-10-0-1-50:~$
```