

Simulando AWS com LocalStack e TerraForm

Docker Compose básico para subir container do LocalStack:

```
services:
  localstack:
    image: localstack/localstack:latest
    container_name: local-aws-services
    environment:
      - DEBUG=${DEBUG:-0}
    ports:
      - '4566-4597:4566-4597'
    volumes:
      - './.localstack:/var/lib/localstack'
      - '/var/run/docker.sock:/var/run/docker.sock'
```

Comando para executar docker compose:

```
docker-compose up
```

Para o Podman o PATH AONDE O DOCKER-COMPOSE.YML FOI SALVO NÃO PODE TER " ", "-" E "_"

Comandos terraform:

- inicializa o terraform: **terraform init**
- exibe as alterações que serão realizadas: **terraform plan**
- aplica as alterações planejadas: **terraform apply -auto-approve**
- destrói toda a infraestrutura criada: **terraform destroy**

Comandos AWS:

Testar LocalStack, exibindo informações do usuário:

```
$ aws --endpoint http://localhost:4566 sts get-caller-identity
```

Resposta

```
{
  "UserId": "AKIAIOSFODNN7EXAMPLE",
  "Account": "000000000000",
  "Arn": "arn:aws:iam::000000000000:root"
}
```

Testar as Buckets

Listar buckets aws:

```
aws s3 ls --profile localstack
```

Criar um bucket

```
aws s3 mb s3://lduran-bucket --profile localstack
```

Copia um arquivo para o seu bucket

```
aws s3 cp testes.txt s3://lduran-bucket --profile localstack
```

Listar o conteúdo de um bucket

```
aws s3 ls s3://lduran-bucket/ --profile localstack
```

Comandos para testar o DynamoDB e os Lamdas no LocalStack (devem ser executados via Git Bash):

Listar tabelas do DynamoDB

```
aws dynamodb list-tables --endpoint-url=http://localhost:4566
```

Listar funções Lambda

```
aws lambda list-functions --endpoint-url=http://localhost:4566
```

Listar APIs no API Gateway

```
aws apigateway get-rest-apis --endpoint-url=http://localhost:4566
```

Testar a Tabela DynamoDB

Adicionar um item ao DynamoDB

```
aws dynamodb put-item \
  --table-name Books \
  --item '{"id": {"S": "1"}, "title": {"S": "Test Book"}, "author": {"S": "Author Name"}}' \
  --endpoint-url=http://localhost:4566
```

Listar os itens do DynamoDB

```
aws dynamodb scan --table-name Books --endpoint-url=http://localhost:4566
```

Recuperar um item específico do DynamoDB

```
aws dynamodb get-item \
  --table-name Books \
  --key '{"id": {"S": "1"}}' \
  --endpoint-url=http://localhost:4566
```

Excluir um item do DynamoDB

```
aws dynamodb delete-item \
  --table-name Books \
  --key '{"id": {"S": "1"}}' \
  --endpoint-url=http://localhost:4566
```

Testar as Funções Lambda

Invocar a função Lambda para listar livros

```
aws lambda invoke \
  --function-name BookFunction \
  --payload '{"httpMethod": "GET", "path": "/book"}' \
  --cli-binary-format raw-in-base64-out \
  response.json \
  --endpoint-url=http://localhost:4566
```

Invocar a função Lambda para criar um livro

```
aws lambda invoke \
  --function-name BookFunction \
  --payload '{"httpMethod": "POST", "path": "/book", "body": "{\"id\": \"2\", \"title\": \"New Book\", \"author\": \"New Author\"}"}' \
  --cli-binary-format raw-in-base64-out \
  response.json \
  --endpoint-url=http://localhost:4566
```

Verificação da Infraestrutura

```
aws apigateway get-rest-apis --endpoint-url=http://localhost:4566
```

Resposta

```
{
  "items": [
    {
      "id": "pgnbnhjjzc",
      "name": "BookFunction",
      "description": "API para gerenciar livros",
      "createdDate": "2024-11-16T15:39:53-03:00",
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "EDGE"
        ]
      },
      "disableExecuteApiEndpoint": false
    }
  ]
}
```

Testar o API Gateway

Para testar o API Gateway, você pode usar uma ferramenta como curl, Postman ou até mesmo a AWS CLI para enviar requisições HTTP para os endpoints criados.

Substitua {api-id} pelo ID da API retornado no passo de **Verificação da Infraestrutura**.

O api-id deve ser pego desta linha da saída do script do TerraForm: `aws_api_gateway_resource.book_id:`

Creation complete after 0s [id=oty6yjwvyw]

Listar todos os livros

```
curl -X GET "http://localhost:4566/restapis/{api-id}/prod/_user_request_/book"
```

Buscar um livro específico por ID

```
curl -X GET "http://localhost:4566/restapis/{api-id}/prod/_user_request_/book/{id}"
```