

# Atividade Docker - PB UFC 2023

Ana Karine Nobre Bezerra

## Modelo de execução:

- Amazon linux 2;
- t2.micro
- Volume de 16g
- Script subindo as orientações

## A VPC desta atividade está dividida entre:

- Duas tabelas de rotas, em que uma é dedicada para as subnets privadas e a outra para as subnets públicas
- Duas subnets privadas e duas públicas, no caso as duas privadas é para conter o container docker com o wordpress.
- As duas subnets públicas estão criadas para permitir que o load balancer consiga conectar-se à internet a partir das duas zonas de disponibilidade.
- A tabela de rotas públicas possui um internet gateway para acessar a internet.
- A tabela de rotas privadas possui um NAT para permitir o tráfego apenas de saída

## Security Group:

EC2:

Tipo	Protocolo	Porta	Origem
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
HTTP S	TCP	443	0.0.0.0/0

Load Balancer:

Tipo	Protocolo	Porta	Origem
HTTPS	TCP	443	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0

EFS:

Tipo	Protocolo	Porta	Origem
NFS	TCP	2049	0.0.0.0/0

RDS:

Tipo	Protocolo	Porta	Origem
MYSQL/Aurora	TCP	3306	Security Group -EC2
MYSQL/Aurora	TCP	3306	Meu IP

### Load Balancer : Criação

- Na seção do load balancer da aws.
- Selecionar o tipo de Application Load Balancer.
- Nomear o load balancer.
- Selecionar o esquema para identificar que ele é voltado para internet.
- Selecionar tipo de endereço IP como IPv4.
- Mapear as subnets públicas de cada zona.
- Criar o grupo de segurança para o load balancer.
- Observar os listeners e os roteamentos, selecionar o target group criado

### TARGET GROUP:

- Na seção de target group, crie um grupo de destino e configure o básico das instâncias, coloque nome e tags.
- Selecionar o protocolo HTTP e a porta 80.
- O tipo de endereço IP: IPv4
- Selecionar a VPC que estarão nas instâncias EC2.
- Selecionar o caminho para verificação de integridade "/", juntamente do protocolo HTTP.
- Depois basta registrar as instâncias.

### Auto scaling:

Usando o script user\_data.sh, que foi usado para configurar as instalações.:

- Na seção do auto scaling, crie um grupo, insira o nome e os demais detalhes básicos.
- Selecionar o script criado, após selecionar a VPC.
- Selecionar as subnets privadas.
- Selecionar o Load balance.
- Selecionar a capacidade mínima e máxima como 2.

### EFS:

- Na seção de EFS na AWS.
- Crie um sistema de arquivos, configure os detalhes básicos como nome, a VPC que vai pertencer.
- Utilize o grupo de segurança do EFS.
-

## RDS:

- Na seção de RDS, abra o menu de Banco de Dados, crie um.
- Selecionar o Mysql, no nível gratuito.
- Inserir o nome da instância
- Configurar nome e senha do usuário
- Configuração de instância foi db.t3.micro.
- Escolher armazenamento gp2.
- Em conectividade, utilize a opção "não se conectar a um recurso de computação do EC2".
- Tipo de rede IPv4.
- Utilize a VPC que estão nas instâncias e os componentes da infraestrutura.
- Selecionar grupo de sub-redes.
- Utilize grupo de segurança para o RDS
- Não usar zona de preferência pode deixar sem.
- Deixar a autoridade de certificação como padrão
- Selecionar opção de autenticação com senha
- Inserir nome do RDS

## Arquivos

- **Script user\_data.sh**

```
#!/bin/bash
```

```
# Atualiza o sistema  
sudo yum update -y
```

```
# Instala o Docker  
sudo yum install docker -y
```

```
# Inicializa e habilita o Docker no início da instância  
sudo systemctl start docker  
sudo systemctl enable docker
```

```
# Instala o Docker Compose curl -L  
"https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  
chmod +x /usr/local/bin/docker-compose
```

```
# Baixa o arquivo .yaml do GitHub curl -sL  
"https://raw.githubusercontent.com/Ana408/atv_docker.PBAWS/main/docker-compose.yaml" --output "/home/ec2-user/docker-compose.yaml"
```

```
# Instala o cliente NFS  
sudo yum install nfs-utils -y
```

```
# Cria um diretório para montagem
```

```
sudo mkdir /mnt/efs
```

```
# Define permissões no diretório para leitura, escrita e execução
```

```
sudo chmod 777 /mnt/efs
```

```
# Monta o sistema de arquivos com o EFS
```

```
sudo mount -t nfs4 -o
```

```
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport
```

```
fs-07c68e847f4ea9744.efs.us-east-1.amazonaws.com:/ /mnt/efs
```

```
# Habilita a montagem auto mática na inicialização
```

```
echo "fs-07c68e847f4ea9744.efs.us-east-1.amazonaws.com:/ /mnt/efs nfs defaults 0  
0" | sudo tee -a /etc/fstab.
```

```
# Adiciona o usuário atual ao grupo do Docker
```

```
sudo usermod -aG docker ${USER}
```

```
# Dá permissões de leitura e escrita no docker.sock
```

```
sudo chmod 666 /var/run/docker.sock
```

```
# Cria o contêiner com o Docker Compose usando a imagem do .yaml
```

```
docker-compose -f /home/ec2-user/docker-compose.yaml up -d
```

- **SCRIPT E ARQUIVO .YAML**

```
version: '3.7'
```

```
services:
```

```
  wordpress:
```

```
    image: wordpress
```

```
    ports:
```

```
      - "80:80"
```

```
  environment:
```

```
    WORDPRESS_DB_HOST: "endpoint do seu RDS"
```

```
    WORDPRESS_DB_USER: admin
```

```
    WORDPRESS_DB_PASSWORD: "senha do rds"
```

```
    WORDPRESS_DB_NAME: database-1
```

```
  volumes:
```

```
    - /mnt/efs/wordpress:/var/www/html
```