

Implantação de Aplicação WordPress em uma Instância EC2 com Docker ou Containerd no AWS, RDS MySQL, EFS e Load Balancer

Configuração AWS

- **VPC:**

- Passo 1: Organização da VPC

Nossa VPC nesta atividade é organizada em duas tabelas de rotas. Uma tabela é dedicada às subnets privadas, e a outra às subnets públicas.

- Passo 2: Subnets Privadas e Públicas

Temos duas subnets privadas e duas subnets públicas.

As subnets privadas são criadas para hospedar um contêiner Docker que executa o WordPress. Isso permite que o WordPress seja acessado por meio de um endereço IP privado, em vez de um endereço público.

As subnets públicas são configuradas para permitir que o balanceador de carga (Load Balancer) acesse a internet a partir de duas Zonas de Disponibilidade (AZs) diferentes. Isso aumenta a disponibilidade do serviço.

- Passo 3: Roteamento de Tráfego

A tabela de rotas das subnets públicas inclui um Internet Gateway, o que permite que as instâncias nas subnets públicas acessem a internet.

A tabela de rotas das subnets privadas inclui um NAT Gateway. O NAT Gateway permite que as instâncias nas subnets privadas acessem a internet apenas para o tráfego de saída, mas não permite que a internet acesse diretamente as instâncias nas subnets privadas.

Essa organização da VPC garante que o WordPress seja acessível por meio de um IP privado e que o tráfego de saída esteja habilitado, enquanto a segurança das subnets privadas é mantida.

- **Grupo de Segurança (Security Group)**

Para esta atividade, foram criados três grupos de segurança, cada um com regras específicas para permitir ou negar o tráfego de rede para as instâncias correspondentes:

- Grupo de Segurança para EC2 (SG-EC2):
- Tipo: Security Group (Grupo de Segurança)
 - Regras:
 - SSH (Protocolo TCP, Porta 22) - Permitido para qualquer origem (0.0.0.0/0)
 - HTTP (Protocolo TCP, Porta 80) - Permitido para qualquer origem (0.0.0.0/0)
 - HTTPS (Protocolo TCP, Porta 443) - Permitido para qualquer origem (0.0.0.0/0)
- Grupo de Segurança para o Balanceador de Carga (SG-LoadBalancer):
- Tipo: Security Group (Grupo de Segurança)
 - Regras:
 - HTTPS (Protocolo TCP, Porta 443) - Permitido para qualquer origem (0.0.0.0/0)
 - HTTP (Protocolo TCP, Porta 80) - Permitido para qualquer origem (0.0.0.0/0)
- Grupo de Segurança para EFS (SG-EFS):
- Tipo: Security Group (Grupo de Segurança)
 - Regra:
 - NFS (Protocolo TCP, Porta 2049) - Permitido para qualquer origem (0.0.0.0/0)
- Grupo de Segurança para RDS (SG-RDS):
- Tipo: Security Group (Grupo de Segurança)
 - Regras:
 - MYSQL/Aurora (Protocolo TCP, Porta 3306) - Permitido para o SG-EC2 (Grupo de Segurança SG-EC2) e o IP 177.37.230.130/32.

Este passo a passo detalha os grupos de segurança criados para gerenciar o tráfego de rede nas instâncias EC2, no balanceador de carga, no sistema de arquivos elástico (EFS) e no banco de dados RDS, garantindo que o acesso seja concedido apenas de acordo com as regras especificadas.

- **Configurando o Load Balancer**

- Passo 1: Escolher o Load Balancer

Para esta atividade, optou-se pelo uso do Application Load Balancer devido aos benefícios em relação ao Load Balancer Classic, que incluem o uso de grupos de destinos.

- Passo 2: Criar o Load Balancer

A configuração do Load Balancer envolve as seguintes etapas:

- Acesse a seção de "Load Balancers" na AWS.
- Clique em "Criar Load Balancer".
- Selecione o tipo "Application Load Balancer".
- Insira o nome desejado para o Load Balancer.
- Escolha o esquema "Voltado para Internet".
- Selecione o tipo de endereço IP como "IPv4".
- Mapeie as subnets públicas de cada Zona de Disponibilidade (AZ).
- Configure o grupo de segurança para o Load Balancer.
- Na seção de "Listeners" e roteamento, selecione o grupo de destino criado.

- Passo 3: Configurar o Grupo de Destino (Target Group)

Para configurar o Target Group, siga estas etapas:

- Acesse a seção de "Grupos de Destino" na AWS.
- Clique em "Criar Grupo de Destino".
- Na configuração básica, escolha "Instâncias".
- Insira um nome para o grupo de destino.
- Selecione o protocolo e a porta desejados, como "HTTP" e "80", respectivamente.
- Escolha o tipo de endereço IP como "IPv4".
- Selecione a VPC onde as instâncias EC2 estão localizadas.
- Defina o caminho de verificação de integridade e o protocolo, como "/" e "HTTP", respectivamente.
- Registre as instâncias no grupo de destino, o que também pode ser feito posteriormente, retornando ao menu de grupos de destino, clicando em "Ações" e, em seguida, "Registrar Destinos".

Este passo a passo descreve como criar e configurar um Load Balancer usando o Application Load Balancer na AWS, juntamente com o seu grupo de destino. A configuração visa otimizar o roteamento de tráfego e a distribuição das solicitações entre as instâncias. Lembre-se de personalizar as configurações de acordo com as necessidades específicas do seu aplicativo.

- **Auto Scaling**

- Passo 1: Preparar o Launcher Template

Para configurar o Auto Scaling, comece criando um Launcher Template que utilize o script `user_data.sh`.

- Passo 2: Configurar o Auto Scaling

A configuração do Auto Scaling envolve os seguintes passos:

- Acesse a seção "Grupos do Auto Scaling" na AWS.
- Clique em "Criar Grupo do Auto Scaling".
- Insira um nome para o grupo.
- Selecione o Launcher Template que você criou.
- Escolha a VPC apropriada.
- Selecione as subnets privadas onde as instâncias serão lançadas.
- No que diz respeito ao Balanceador de Carga, você pode optar por selecionar um já existente ou criar um posteriormente.
- Opcionalmente, habilite a coleta de métricas de grupo no CloudWatch.
- Escolha a capacidade desejada, mínima e máxima, que no caso é definida como 2.

- Launcher Template

Para o Launcher Template, foi escolhida a seguinte configuração dentro do nível gratuito:

- Sistema Operacional: Amazon Linux 2
- Tipo de Instância: t2.micro
- Utilize um "Par de Chaves" criado especificamente para esta atividade.
- Configure o Grupo de Segurança conforme especificado nas seções anteriores.
- Escolha um armazenamento do tipo GP2 com capacidade de 8GB.
- O Launcher Template utiliza um script e `user_data`.

Este passo a passo descreve como configurar o Auto Scaling na AWS usando um Launcher Template com uma instância Amazon Linux 2, seguindo as configurações e parâmetros definidos para este ambiente.

- **EFS**

- Passo: Configurar o Elastic File System (EFS)

Para criar um Elastic File System, siga estas etapas:

- Acesse a seção "EFS" na AWS.
- Clique em "Criar Sistema de Arquivos".
- Insira um nome para o sistema de arquivos EFS.
- Escolha a VPC na qual o sistema de arquivos será implantado.
- Selecione o sistema de arquivos criado.
- Clique em "Visualizar Detalhes".
- Acesse a seção "Redes".
- Substitua o grupo de segurança pelo grupo de segurança previamente definido nas seções anteriores.

Este passo a passo simplifica a criação de um sistema de arquivos elástico (EFS) na AWS, permitindo que você configure e integre o EFS na sua VPC e com as configurações de segurança adequadas.

- **RDS**

- Passo 1: Configurar o Amazon RDS (Relational Database Service)

A configuração do Amazon RDS segue estas etapas:

- Acesse a seção "RDS" na AWS.
- Vá para o menu "Banco de Dados".
- Clique em "Criar Banco de Dados".
- Escolha o método de criação padrão.
- Selecione o mecanismo MySQL.
- Escolha o modelo de nível gratuito.
- Insira um nome de identificação para a instância.
- Configure o nome de usuário.
- Configure a senha do usuário.
- Selecione a configuração da instância, que é "db.t3.micro".
- Escolha o armazenamento GP2.
- Na seção de conectividade, selecione a opção "Não se conectar a um recurso de computação do EC2".
- Escolha o tipo de rede como IPv4.
- Selecione a VPC na qual estarão localizadas as instâncias e os outros componentes da infraestrutura.
- Escolha o grupo de sub-redes.
- Crie um grupo de segurança para o RDS.

- Deixe a zona de disponibilidade como "Sem preferência".
- Deixe a autoridade de certificação como padrão.
- Selecione a opção de autenticação com senha.
- Vá para as configurações adicionais.
- Insira o nome do RDS.
- As configurações de backup, monitoramento, criptografia e manutenção são opcionais.

Este passo a passo simplifica a configuração do Amazon RDS, permitindo que você crie uma instância do banco de dados MySQL com as configurações desejadas. Lembre-se de ajustar as configurações de acordo com as necessidades específicas do seu aplicativo.

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://github.com/getuliompinho/Pr-tica-Docker/blob/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 automática 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
```

- **Imagem .yaml**

```
version: '3.7'
```

```
services:
```

```
  wordpress:
```

```
    image: wordpress
```

```
    ports:
```

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

```
    environment:
```

```
      WORDPRESS_DB_HOST:
```

```
dockerproject.cdopevmnjizw.us-east-1.rds.amazonaws.com
```

```
      WORDPRESS_DB_USER: admin
```

```
      WORDPRESS_DB_PASSWORD: admin123
```

```
      WORDPRESS_DB_NAME: DockerProject
```

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