

Documentação de Criação e Configuração Docker - Versão 1.0

Ana Maria Cavasin

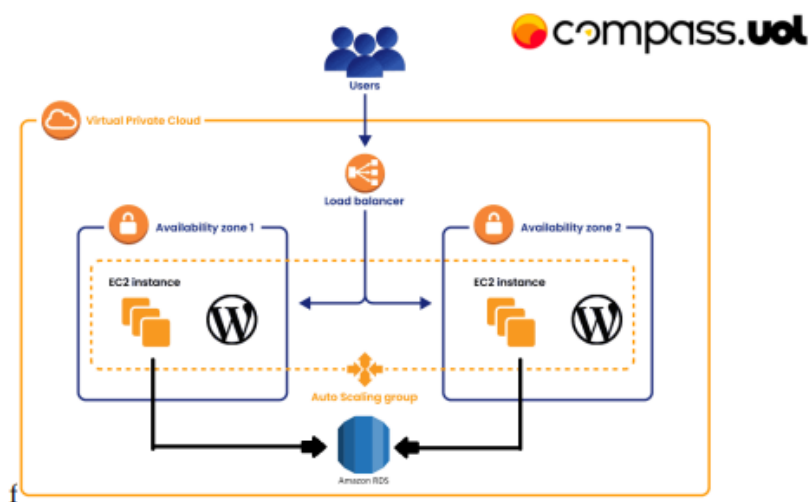
Data de Atualização: 30/10/2023

Visão Geral

A presente documentação fornece instruções passo a passo para a instalação e configuração de uma aplicação WordPress com RDS database MySQL dentro de um contêiner Docker em uma instância EC2 na AWS. A seguir serão apresentados os tópicos que serão apresentados nessa documentação:

Arquitetura:

O modelo a seguir representa a arquitetura desenvolvida nesse projeto.



Descrição do projeto:

- Instalação e configuração do DOCKER ou CONTAINERD no host EC2;
- Utilizar a instalação via script de Start Instance (user_data.sh).
- Efetuar Deploy de uma aplicação Wordpress com: container de aplicação RDS database Mysql.
- Configuração da utilização do serviço EFS AWS para estáticos do container de aplicação Wordpress.
- Configuração do serviço de Load Balancer AWS para a aplicação Wordpress.

Pontos de atenção:

- Não utilizar ip público para saída dos serviços WP (Evitem publicar o serviço WP via IP Público).
- Sugestão para o tráfego de internet sair pelo LB (Load Balancer Classic).
- As pastas públicas e estáticos do wordpress sugerem de utilizar o EFS (Elastic File System).
- Usar Dockerfile ou Dockercompose.
- Aplicação Wordpress rodando na porta 80 ou 8080.

Desenvolvimento da Tarefa

Siga estas etapas para realizar as configurações. Em cada passo será apresentado uma instrução escrita, bem como o tutorial em vídeo com o passo a passo.

Passo 1: Criar uma nova VPC

- No Console de Gerenciamento da AWS, acesse a opção **VPC**.
- Agora busque a opção **Suas VPCs**
- Após isso, clique em **Criar VPC**
- Selecione a opção **VPC e muito mais** e insira um **nome para VPC**.

- Escolha o **Número de zonas de disponibilidade (AZs)** e o **Número de sub-redes públicas**.

- Agora escolha o **Gateways Nat (USD)** e os **Endpoints da VPC**.

Gateways NAT (USD) [Informações](#)

Escolha o número de zonas de disponibilidade (AZs) nas quais criar gateways NAT. Observe que há uma cobrança para cada gateway NAT.

Nenhuma	Em 1 AZ	1 por AZ
---------	---------	----------

Endpoints da VPC [Informações](#)

Os endpoints podem ajudar a reduzir as cobranças do gateway NAT e melhorar a segurança acessando o S3 diretamente da VPC. Por padrão, a política de acesso integral será usada. Você pode personalizar essa política a qualquer momento.

Nenhuma	Gateway do S3
---------	---------------

Opções de DNS [Informações](#)

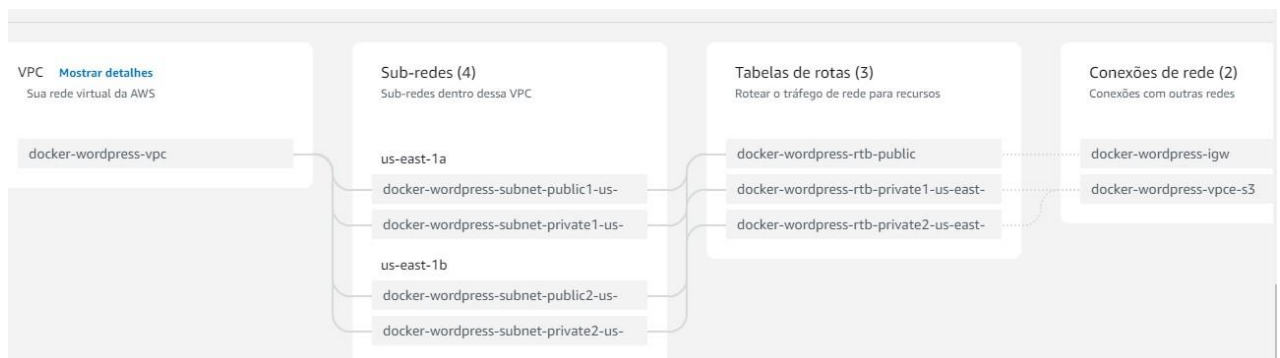
- ☒ Habilitar nomes de host DNS
- ☒ Habilitar resolução de DNS

▼ Tags adicionais

Adicione tags à VPC e a todos os recursos dentro da VPC. Não defina a tag Name aqui. Defina a tag Name em Name tag auto-generation acima ou diretamente no visualizador.

[Adicionar nova tag](#)

- No final, sua VPC deve ficar parecida com a seguinte.



- Agora clique em **Criar VPC**

Passo 2: Criar um Grupo de Segurança

- No Console de Gerenciamento da AWS, acesse **EC2**.
- Acesse a opção **Security group** na sessão **Redes e segurança**
- Clique em **Criar grupo de segurança**
- Em **Detalhes básicos** defina o **Nome do Grupo** e a **VPC** criada anteriormente

Criar grupo de segurança [Informações](#)

Um grupo de segurança atua como um firewall virtual para sua instância para controlar o tráfego de entrada e saída. Para criar um novo grupo de segurança, preencha os campos abaixo.

Detalhes básicos

Nome do grupo de segurança [Informações](#)

O nome não pode ser editado após a criação.

Descrição [Informações](#)

VPC [Informações](#)

- Em **Regras de Entrada** defina as seguintes portas

ID da regra do grupo de segurança	Tipo Informações	Protocolo Informações	Intervalo de portas Informações	Origem Informações	Descrição - opcional Informações	
sgr-012abf22fd72ac015	SSH	TCP	22	Per... <input type="text" value="0.0.0.0/0"/>		Excluir
sgr-0b6b5ae7cc8a07056	MYSQL/Aurora	TCP	3306	Per... <input type="text" value="0.0.0.0/0"/>		Excluir
sgr-07c95935b736141e2	HTTP	TCP	80	Per... <input type="text" value="0.0.0.0/0"/>		Excluir
-	NFS	TCP	2049	Per... <input type="text" value="0.0.0.0/0"/>		Excluir

- Confirme a criação do **Security Group**

Passo 3: Criar o RDS database MySQL

- No Console de Gerenciamento da AWS, acesse **RDS**.
- Clique em **Banco de Dados** e depois em **Criar Banco de Dados**.
- Escolha o método de **criação padrão**.

Criar banco de dados

Escolher um método de criação de banco de dados [Informações](#)

☒ Criação padrão

Defina todas as opções de configuração, incluindo as de disponibilidade, segurança, backups e manutenção.



☐ Criação fácil

Use as configurações recomendadas de melhores práticas. Algumas opções de configuração podem ser alteradas após a criação do banco de dados.

- Em **Opções do mecanismo** escolha o **MySQL**.

Opções do mecanismo

Tipo de mecanismo [Informações](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input type="radio"/> Aurora (PostgreSQL Compatible) 
<input checked="" type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 

- No **Modelos** escolha a opção **Nível gratuito**.

Modelos

Escolha um modelo de exemplo para atender a seu caso de uso.

<input type="radio"/> Produção Use padrões para alta disponibilidade e desempenho rápido e consistente.	<input type="radio"/> Dev/Test Esta instância é planejada para uso de desenvolvimento fora de um ambiente de produção.
<input checked="" type="radio"/> Nível gratuito Use o nível gratuito do RDS para desenvolver novas aplicações, testá-las ou obter uma experiência prática com o Amazon RDS. Informações	

- Agora em **Configurações** escreva um **identificador da instancia do banco de dados**.

Configurações

Identificador da instância de banco de dados [Informações](#)

Digite um nome para a instância de banco de dados. O nome deve ser exclusivo entre todas as instâncias de banco de dados de propriedade de sua conta da AWS na região atual da AWS.

O identificador da instância de banco de dados não diferencia maiúsculas de minúsculas, mas é armazenado com todas as letras minúsculas (como em "mydbinstance"). Restrições: 1 a 60 caracteres alfanuméricos ou hífens. O primeiro caractere deve ser uma letra. Não pode conter dois hífens consecutivos. Não pode terminar com um hífen.

- Em **configurações de credencial**, escolha o **Nome do usuário principal** e a **senha principal**.

Nome do usuário principal [Informações](#)

Digite um ID de login para o usuário principal de sua instância de banco de dados.

admin

De um a 16 caracteres alfanuméricos. O primeiro caractere deve ser uma letra.

☐ Gerenciar credenciais principais no AWS Secrets Manager

Gerencie credenciais de usuário principais no Secrets Manager. O RDS pode gerar uma senha para você e gerenciá-la durante todo o seu ciclo de vida.

 Se você gerenciar as credenciais de usuário principais no Secrets Manager, alguns recursos do RDS não serão compatíveis. [Saiba mais](#)

☐ Gerar uma senha automaticamente

O Amazon RDS pode gerar uma senha para você, ou você pode especificar sua própria senha.

Senha principal [Informações](#)

Restrições: pelo menos oito caracteres ASCII imprimíveis. Não pode conter nenhum dos seguintes: / (barra), ' (aspas simples), " (aspas duplas) ou @ (arroba).

Confirmar senha principal [Informações](#)

- Em **Conectividade**, selecione **Não se conectar a um recurso de computação EC2** e selecione a VPC criada anteriormente no passo 1.

Conectividade [Informações](#)



Recurso de computação

Escolha se deseja configurar uma conexão com um recurso de computação para esse banco de dados. A configuração de uma conexão altera automaticamente as configurações de conectividade, para que o recurso de computação possa se conectar a esse banco de dados.

☒ Não se conectar a um recurso de computação do EC2

Não configure uma conexão com um recurso de computação para esse banco de dados. Você poderá configurar uma conexão com um recurso de computação manualmente mais tarde.

☐ Conectar-se a um recurso de computação do EC2

Configure uma conexão com um recurso de computação do EC2 para esse banco de dados.

Nuvem privada virtual (VPC) [Informações](#)

Escolha a VPC. A VPC define o ambiente de rede virtual dessa instância de banco de dados.

docker-wordpress-vpc (vpc-072185b4b3d9c61b2)

4 Sub-redes, 2 Zonas de disponibilidade

Somente as VPCs com um grupo de sub-redes de banco de dados correspondente são listadas.

- Em **configuração adicional**, dentro **opções de banco de dados**, escolha **nome do banco de dados inicial** e a **versão do banco de dados**.

▼

Configuração adicional

Opções de banco de dados, criptografia ativado, backup ativado, retroceder desativado, manutenção, CloudWatch Logs, excluir proteção desativado.

Opções de banco de dados

Nome do banco de dados inicial

Informações

wordpress

Se você não especificar um nome de banco de dados, o Amazon RDS não criará um banco de dados.

Grupo de parâmetros do banco de dados

Informações

default.mysql8.0

Grupo de opções

Informações

default:mysql-8-0

- As **demais opções ficam padrão.**
- Agora clique em **Criar banco de dados**
- Verifique se o **grupo de segurança** do banco de dados nas regras de entrada está liberado a seguinte porta

Regra de entrada 2

Excluir

ID da regra do grupo de segurança

Tipo

Informações

-

MYSQL/Aurora

Protocolo

Informações

TCP

Intervalo de portas

Informações

3306

Tipo de origem

Informações

Personalizado

Origem

Informações

Q

sg-0c1866ccbd61c588

X

Descrição - opcional

Informações

- Agora guarde as seguintes informações que serão úteis nos próximos passos

Segurança e conexão

Endpoint e porta

Endpoint

wordpress.c7djeexh6rth.us-east-1.rds.amazonaws.com

Porta

3306

Redes

Zona de disponibilidade

us-east-1b

VPC

[docker-wordpress-vpc \(vpc-072185b4b3d9c61b2\)](#)

Grupo de sub-redes

default-vpc-072185b4b3d9c61b2

Sub-redes

[subnet-088828af88a5e77ce](#)

Passo 4: Criar o EFS – Sistema de Arquivos

- No Console de Gerenciamento da AWS, acesse **EFS**.
- Clique em **Sistemas de Arquivos** e depois em **Criar Sistemas de Arquivos**.
- Escolha o **Nome** e a **VPC criada anteriormente** e clique em criar.

Criar sistema de arquivos

Crie um sistema de arquivos EFS com as configurações recomendadas. [Saiba mais](#)

Nome - *opcional*

Nomeie seu sistema de arquivos.

docker-wordpress-efs

O nome pode incluir letras, números e os símbolos +-=._:/, até 256 caracteres.

Virtual Private Cloud (VPC)

Selecione a VPC em que você deseja que as instâncias do EC2 se conectem ao sistema de arquivos.

vpc-072185b4b3d9c61b2

docker-wordpress-vpc

Cancelar

Personalizar

Criar

- Agora guarde a seguinte informação que iremos utilizar posteriormente

docker-wordpress-efs (fs-0e23f87bdba2182bd) Excluir Anexar

Geral Editar

<p>Modo de desempenho</p> <p>Uso geral</p> <p>Modo de taxa de transferência</p> <p>Elastic</p> <p>Gerenciamento de ciclo de vida</p> <p>Transição para IA: 30 dias desde o último acesso</p> <p>Transição fora do IA: Nenhum</p> <p>Zona de disponibilidade</p> <p>Padrão</p>	<p>Backups automáticos</p> <p>✔ Habilitado</p> <p>Criptografado</p> <p>e1bc0745-cdec-40dd-9299-aef61bd41c88 (aws/elasticfilesystem)</p> <p>Estado do sistema de arquivos</p> <p>✔ Disponível</p> <p>Nome de DNS</p> <p>fs-0e23f87bdba2182bd.efs.us-east-1.amazonaws.com</p>
---	---

- Lembre-se de verificar se os **grupos de segurança** dos **destinos de montagens** é o que criamos no passo 2.

Passo 5: Criar o arquivo user_data

- Na **pasta do projeto em sua máquina**, crie o arquivo **user_data.sh**. Esse arquivo será responsável pela instalação e configuração automática das instancias na sua criação.
- A seguir será explicado cada etapa que está no arquivo e após isso será disponibilizado na integra.
- Na primeira parte temos a **instalação do Docker**, para isso utilizamos os seguintes comandos.
 - `sudo yum update -y` #Atualização dos pacotes de instalação
 - `sudo yum install docker -y` #Instalação do Docker
 - `sudo systemctl start docker.service && systemctl enable docker.service` #Inicialização e habilitação do Docker
 - `usermod -aG docker ec2-user` #Adiciona o usuario ao grupo docker
- Em seguida, fizemos a **instalação do Docker Compose**.
 - `curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose` #instala o docker compose
 - `chmod +x /usr/local/bin/docker-compose` #permite a execução do docker compose
- Agora vamos fazer a **montagem do EFS**
 - `sudo yum install -y nfs-utils` # instala o nfs
 - `sudo mkdir /mnt/efs` #cria a pasta onde o efs será montado
 - Acesse o **EFS criado anteriormente** e busque o **DNS do EFS**

docker-wordpress-efs (fs-0e23f87bdba2182bd) Excluir Anexar

Geral Editar

<p>Modo de desempenho Uso geral</p> <p>Modo de taxa de transferência Elastic</p> <p>Gerenciamento de ciclo de vida Transição para IA: 30 dias desde o último acesso Transição fora do IA: Nenhum</p> <p>Zona de disponibilidade Padrão</p>	<p>Backups automáticos ✔ Habilitado</p> <p>Criptografado e1bc0745-cdec-40dd-9299-aef61bd41c88 (aws/elasticfilesystem)</p> <p>Estado do sistema de arquivos ✔ Disponível</p> <p>Nome de DNS fs-0e23f87bdba2182bd.efs.us-east-1.amazonaws.com</p>
--	---

- echo "[DNS do EFS AWS]:/mnt/efs nfs defaults 0 0" >> /etc/fstab #adiciona o montno de montagem no fstab
- mount -a #faz a montagem do sistema de arquivos
- Agora vamos fazer a **instalação do MySQL que será utilizado posteriormente.**
 - sudo yum install -y mysql
- Agora fizemos a **criação do diretório Docker** onde será criado o docker-compose.yml
 - sudo mkdir /Docker
- Agora precisamos buscar algumas informações para a configuração
 - Busque o **Endpoint do Banco de Dados RDS, bem como o nome do banco e o usuário e senha.**

Segurança e conexão

<p>Endpoint e porta</p> <p>Endpoint wordpress.c7djeexh6rth.us-east-1.rds.amazonaws.com</p> <p>Porta 3306</p>	<p>Redes</p> <p>Zona de disponibilidade us-east-1b</p> <p>VPC docker-wordpress-vpc (vpc-072185b4b3d9c61b2)</p> <p>Grupo de sub-redes default-vpc-072185b4b3d9c61b2</p> <p>Sub-redes subnet-088828af88a5e77ce</p>
---	---

-
- Com isso concluído, fizemos a **criação do arquivo de configuração docker-compose.yml**
 - sudo tee /Docker/docker-compose.yml <<EOF #criação do arquivo na pasta

```

version: '3.7'

services:
  wordpress:
    image: wordpress
    restart: always

```

ports:

- 80:80

environment:

WORDPRESS_DB_HOST: [Endpoint do RDS]

WORDPRESS_DB_NAME: [nome do banco de dados]

WORDPRESS_DB_USER: [usuario]

WORDPRESS_DB_PASSWORD: [senha do usuario]

volumes:

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

EOF

- Agora vamos subir o **Container da aplicação Wordpress**
 - cd /Docker # Acessa o diretório Docker
 - docker-compose up -d # Inicializa os contêineres Docker com o Docker Compose
- A seguir será apresentado o **arquivo na integra**.

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

```
sudo yum update -y
```

```
sudo yum install docker -y
```

```
sudo systemctl start docker.service && systemctl enable docker.service
```

```
usermod -aG docker ec2-user
```

```
curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

```
sudo yum install -y nfs-utils
```

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

```
echo "fs-0e23f87bdba2182bd.efs.us-east-1.amazonaws.com:/ /mnt/efs nfs defaults 0 0" >> /etc/fstab
```

```
mount -a
```

```
sudo yum install -y mysql
```

```
sudo mkdir /Docker
```

```
sudo tee /Docker/docker-compose.yml <<EOF
```

```
version: '3.7'
```

```
services:
```

```
  wordpress:
```

```
    image: wordpress
```

```
    restart: always
```

```
    ports:
```

```
      - 80:80
```

```
    environment:
```

```
      WORDPRESS_DB_HOST: [Endpoint do RDS]
```

```
      WORDPRESS_DB_NAME: [nome do banco de dados]
```

```
      WORDPRESS_DB_USER: [usuario]
```

```
      WORDPRESS_DB_PASSWORD: [senha do usuario]
```

```
    volumes:
```

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

```
EOF
```

```
cd /Docker
```

```
docker-compose up -d
```

Passo 6: Criar o modelo de execução

- No Console de Gerenciamento da AWS, acesse **EC2**.
- Vá na sessão Instancias e depois em **Modelos de Execução**.
- Click em **Criar modelo de execução**.
- Defina um **nome do modelo de execução** e a **descrição** da versão do modelo.

Criar modelo de execução

A criação de um modelo de execução permite criar uma configuração de instância salva que pode ser reutilizada, compartilhada e executada posteriormente. Os modelos podem ter várias versões.

Nome e descrição do modelo de execução

Nome do modelo de execução - *obrigatório*

Deve ser exclusivo para essa conta. Máximo de 128 caracteres. Nenhum espaço ou caracteres especiais, como "&", "*", "@".

Descrição da versão do modelo

Máximo de 255 caracteres

- Em **Tags do modelo**, adicione **todas as Tags padrões**.

▼ Tags de modelo

Chave **Informações**



Valor **Informações**



Remover tag

Chave **Informações**



Valor **Informações**



Remover tag

Chave **Informações**



Valor **Informações**



- Depois escolha a **Imagem da aplicação**, utilize a **Amazon Linux 2**.

▼ **Imagens de aplicação e de sistema operacional (imagem de máquina da Amazon) - obrigatório** [Informações](#)

Uma AMI é um modelo que contém a configuração do software (sistema operacional, servidor de aplicações e aplicações) necessária para executar a instância. Pesquise ou navegue pelas AMIs se você não estiver vendo o que está buscando abaixo

🔍 *Pesquise nosso catálogo completo, incluindo milhares de imagens de aplicações e sistemas operacionais*

Recentes | **Início rápido**

☐ Lançamento recente ☒ Atualmente em uso

Procurar mais AMIs
Incluindo AMIs da AWS, do Marketplace e da comunidade

Imagem de máquina da Amazon (AMI)

amzn2-ami-kernel-5.10-hvm-2.0.20231020.1-x86_64-gp2
ami-01eccbf80522b562b
2023-10-23T19:59:50.000Z arquitetura: 64 bits (x86) Virtualização: hvm ENA habilitado: true
Tipo de dispositivo raiz: ebs

- **Selecione o tipo de instância e o par de chaves para login. O tipo da instância deve ser o t3.small e o par de chaves pode ser um criado anteriormente ou um novo.**

▼ **Tipo de instância** [Informações](#) Avançado

Tipo de instância

t3.small
Família: t3 2 vCPU 2 GiB Memória Geração atual: true
Sob demanda SUSE base definição de preço: 0.0518 USD por hora
Sob demanda Linux base definição de preço: 0.0208 USD por hora
Sob demanda RHEL base definição de preço: 0.0808 USD por hora
Sob demanda Windows base definição de preço: 0.0392 USD por hora

☒ Todas as gerações

[Comparar tipos de instância](#)

[Custos adicionais aplicáveis a AMIs com software pré-instalado](#)

▼ **Par de chaves (login)** [Informações](#)

Você pode usar um par de chaves para se conectar com segurança à sua instância. Certifique-se de ter acesso ao par de chaves selecionado antes de executar a instância.

Nome do par de chaves

awskey

[Criar novo par de chaves](#)

- **Em Configurações de rede, no Firewall(security groups) escolha o Security Group criado no passo 2.**

- No **Armazenamento**, configure o **Tamanho** para **16 GiB**.

- Ainda em **Rede**, adicione uma **interface de rede conforme a imagem**.

▼ Configuração avançada de rede

Interface de rede 1 Remover

Índice de dispositivos Informações <input type="text" value="0"/>	Interface de rede Informações <input type="text" value="Nova interface"/>	Descrição Informações <input type="text"/>
Sub-rede Informações Não incluir no modelo de execução	Grupos de segurança Informações <input type="text" value="Selecionar grupos de segurança"/> Exibir todos os selecionados (1)	Atribuir IP público automaticamente Informações <input type="text" value="Não incluir no modelo de e..."/>
IP primário Informações <input type="text" value="123.123.123.1"/>	IP secundário Informações <input type="text" value="Não incluir no modelo de e..."/>	IPs IPv6 Informações <input type="text" value="Não incluir no modelo de e..."/>
Prefixos IPv4 Informações <input type="text" value="Não incluir no modelo de e..."/>	Prefixos IPv6 Informações <input type="text" value="Não incluir no modelo de e..."/>	Atribuir IP IPv6 primário Informações <input type="text" value="Não incluir no modelo de e..."/>

- Em **Tags de recursos**, adicione as mesmas **Tags (Name, Project e CostCenter)** nos tipos de recursos **Instâncias e Volumes**.

▼ Tags de recurso [Informações](#)

Chave Informações <input type="text" value="Name"/>	Valor Informações <input type="text" value="PB - FW - A - RG - SB - HA"/>
Tipos de recurso Informações <input type="text" value="Selecionar tipos de recurso"/>	
<input type="button" value="Instâncias"/> <input type="button" value="Volumes"/>	
<input type="button" value="Remover"/>	


Chave Informações <input type="text" value="Project"/>	Valor Informações <input type="text" value="PB - FW - A - RG - SB - HA"/>
Tipos de recurso Informações <input type="text" value="Selecionar tipos de recurso"/>	
<input type="button" value="Instâncias"/> <input type="button" value="Volumes"/>	
<input type="button" value="Remover"/>	

Chave Informações	Valor Informações
-----------------------------------	-----------------------------------

- Em **Detalhes Avançados**, em **Dados do usuário**, adicione o arquivo **user_data.sh** criado no passo 5.

Dados do usuário - *optional* | [Informações](#)

Faça upload de um arquivo com seus dados de usuário ou o insira no campo.

 Choose file

```
#!/bin/bash

# Instalação do Docker
sudo yum update -y
sudo yum install docker -y
sudo systemctl start docker.service && systemctl enable docker.service
usermod -aG docker ec2-user

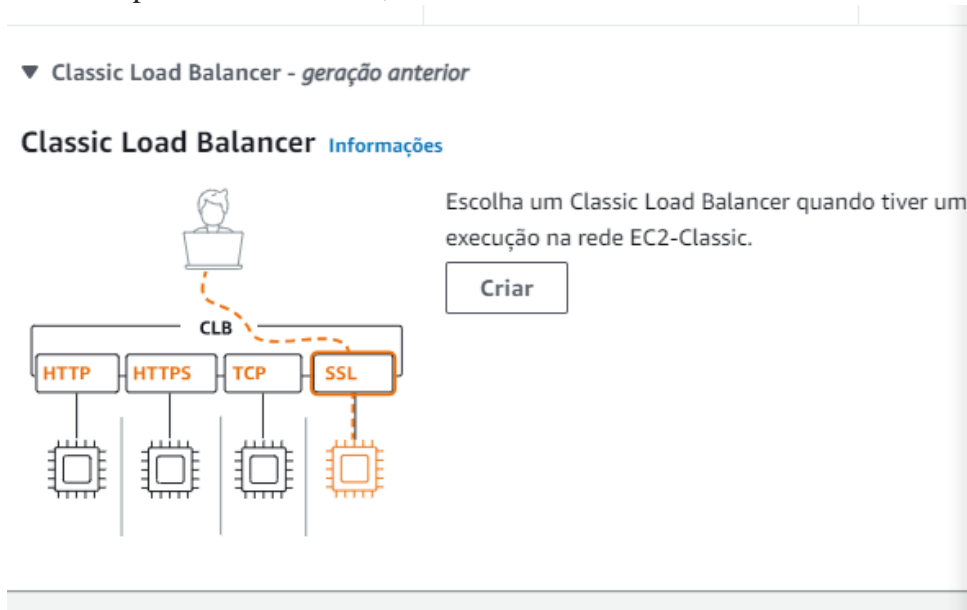
# estes comandos instalarão o Docker Compose
curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
chmod +x /usr/local/bin/docker-compose

# Montagem do EFS
sudo yum install -y nfs-utils
```

- Confira as informações e clique em **Criar versão do modelo**.

Passo 7: Criar Load Balancing

- No Console de Gerenciamento da AWS, acesse **EC2**.
- Vá na sessão Balanceamento de Carga e depois em **Load Balancers**.
- Click em **Criar load balancer**.
- Escolha o tipo de load balancer, no nosso caso será o **Classic Load Balancer**.



- Depois defina o **Nome do load balancer** e o **Esquema Voltado para a internet**.
- Em **Mapeamento de rede**, selecione a **VPC criada no passo 1** e no **Mapeamento** selecione as duas **AZs** e as respectivas **sub-redes públicas**.

Mapeamento de rede [Informações](#)

O load balancer roteia o tráfego para destinos nas sub-redes selecionadas e de acordo com suas configurações de endereço IP.

VPC [Informações](#)

Selecione a nuvem privada virtual (VPC) para seus destinos ou crie uma VPC [🔗](#). Somente VPCs com um gateway da Internet são habilitadas para seleção. A VPC selecionada não pode ser alterada após a criação do balanceador de carga. Para confirmar a VPC para seus destinos, visualize seus [grupos de destino](#) [🔗](#).

docker-wordpress-vpc
vpc-072185b4b3d9c61b2
IPv4: 10.0.0.0/16



Mapeamentos [Informações](#)

Selecione pelo menos duas zonas de disponibilidade e uma sub-rede por zona. O balanceador de carga roteia o tráfego para destinos somente nessas zonas de disponibilidade. Zonas de disponibilidade não compatíveis com o balanceador de carga ou a VPC não estão disponíveis para seleção.

☒ us-east-1a (use1-az6)

Sub-rede

subnet-0fc5d3c48f9835b24

docker-wordpress-subnet-public1-us-east-1a ▼

Endereço IPv4

Atribuído pela AWS

☒ us-east-1b (use1-az1)

Sub-rede

subnet-0f8100e813a24bf69

docker-wordpress-subnet-public2-us-east-1b ▼

Endereço IPv4

Atribuído pela AWS

- **Selecione o grupo de segurança criado no passo 2.**

Grupos de segurança [Informações](#)

Um grupo de segurança é um conjunto de regras de firewall que controlam o tráfego para o balanceador de carga. Selecione um grupo de segurança existente ou crie um grupo de segurança [🔗](#).

Grupos de segurança

Selecione até 5 grupos de segurança



default

sg-0b8a361cc0c2d8333 VPC: vpc-072185b4b3d9c61b2



docker-wordpress

sg-0c1866ccbd61c588 VPC: vpc-072185b4b3d9c61b2



Listeners e roteamento [Informações](#)

O receptor é um processo que verifica solicitações de conexão usando o protocolo e a porta configurados. As configurações que você define para um receptor determinam como o balanceador de carga encaminhará solicitações aos destinos registrados.

▼ Listener HTTP:80

Instância HTTP:80

Remover

Protocolo do listener

HTTP ▼

Porta do listener

80

1-65535

Protocolo da

instância

HTTP ▼

Porta da instância

80

1-65535

- **Em verificação de integridade, deixe o caminho conforme a imagem.**

Verificações de integridade [Informações](#)

Seu balanceador de carga executa automaticamente verificações de integridade para testar a disponibilidade de todas as instâncias registradas. O tráfego é roteado somente para instâncias íntegras, o que é determinado em sua resposta à verificação de integridade.

Destino do ping
O ping de verificação de integridade é enviado usando o protocolo e a porta especificados. Se o protocolo HTTP/HTTPS estiver sendo usado, você também deverá fornecer o caminho de destino.

Protocolo ping

HTTP

▼

:

Porta ping

80

Caminho de ping

/

1-65535

▶ Configurações avançadas de verificação de integridade

- Em **Load Balancer Tags**, adicione as **Tags padrão do Projeto**.

Chave	Valor	
CostCenter:	C092000004	Remover
Chave	Valor	
Project	PB - FW - A - RG - SB - HA	Remover
Chave	Valor	
Name	PB - FW - A - RG - SB - HA	Remover

- As demais, configurações deixe padrão, agora clique em **Criar Load Balancer**.

Passo 8: Criar Auto Scaling

- No Console de Gerenciamento da AWS, acesse **EC2**.
- Vá na sessão Auto Scaling e depois em **Grupos de Auto Scaling**.
- Click em Criar **Grupos de Auto Scaling**.
- Escolha o **Nome do Auto Scaling**.

Escolher o modelo de execução [Info](#)

Especifique um modelo de execução que contenha configurações comuns a todas as instâncias do EC2 executadas por esse grupo do Auto Scaling.

Nome

Nome do grupo do Auto Scaling

Insira um nome para identificar o grupo.

O nome deve ser exclusivo para essa conta na região atual e não pode ter mais de 255 caracteres.

Modelo de execução [Info](#)

i Para contas criadas após 31 de maio de 2023, o console do EC2 só oferece suporte à criação de grupos do Auto Scaling com modelos de execução. A criação de grupos do Auto Scaling com configurações de execução não é recomendada, mas ainda está disponível por meio da CLI e da API até 31 de dezembro de 2023.

- Agora selecione o **modelo de execução criado anteriormente**.

instância, o par de chaves e os grupos de segurança.

[Criar um modelo de execução](#)

Versão

[Criar uma versão de modelo de execução](#)

Descrição	Modelo de execução	Tipo de instância
-	Docker-wordpress-ME lt-0d4c4a861abdd60e3	t3.small
AMI ID	Grupos de segurança	Solicitar instâncias spot
ami-01eccbf80522b562b	-	Não
Nome do par de chaves	IDs de grupo de segurança	
awskey	sg-0c1866ccbd61c588	

- Verifique se os Requisitos de tipo de instancia estão corretos.

Requisitos de tipo de instância [Info](#)

Você pode manter os mesmos atributos da instância ou tipo de instância do modelo de execução ou modificar o modelo de execução, especificando atributos de instância diferentes ou adicionando manualmente tipos de instância.

Modelo de execução	Versão	Descrição
Docker-wordpress-ME lt-0d4c4a861abdd60e3	2	-
Tipo de instância		
t3.small		

- Em **Rede**, selecione a **VPC criada anteriormente** e selecione as **Zonas de disponibilidade** e as **sub-redes públicas**.

Rede [Info](#)

Para a maioria das aplicações, você pode usar várias zonas de disponibilidade e permitir que o EC2 Auto Scaling equilibre suas instâncias entre as zonas. A VPC padrão e as sub-redes padrão são adequadas para começar a usar o serviço rapidamente.

VPC
Escolha a VPC que define a rede virtual do grupo do Auto Scaling.

vpc-072185b4b3d9c61b2 (docker-wordpress-vpc) 10.0.0.0/16

[Criar uma VPC](#)

Zonas de disponibilidade e sub-redes
Defina quais zonas de disponibilidade e sub-redes seu grupo do Auto Scaling pode usar na VPC escolhida.

Selecionar zonas de disponibilidade e sub-redes

us-east-1a | subnet-0fc5d3c48f9835b24 (docker-wordpress-subnet-public1-us-east-1a) 10.0.0.0/20

us-east-1b | subnet-0f8100e813a24bf69 (docker-wordpress-subnet-public2-us-east-1b) 10.0.16.0/20

[Criar uma sub-rede](#)

- Em **Tamanho de grupo**, configure a **capacidade desejada**.

escalabilidade para escalar dinamicamente o número de instâncias no grupo.

Tamanho do grupo - opcional [Info](#)

Especifique o tamanho do grupo do Auto Scaling alterando a capacidade desejada. Você também pode especificar os limites de capacidade mínima e máxima. Sua capacidade desejada deve estar dentro do intervalo dos limites.

Capacidade desejada
2

Capacidade mínima
2

Capacidade máxima
1

- Em **Etiquetas**, adicione as **Tags** padrões do projeto.

Etiquetas (3)

Chave	Valor - opcional	Etiquetar novas instâncias	
Nome	PB - FW - A - RG - SB - H.	<input checked="" type="checkbox"/>	Remover
CostCenter	C092000004	<input checked="" type="checkbox"/>	Remover
Project	PB - FW - A - RG - SB - H.	<input checked="" type="checkbox"/>	Remover

[Adicionar etiqueta](#)

47 restantes

- Em **Balanceamento de carga**, selecione o **Load Balancer** criado anteriormente.

Balanceamento de carga - opcional

Balanceadores de carga

☐ Grupos de destino de Application, Network ou Gateway Load Balancer

☒ Classic Load Balancers

Selecionar Classic Load Balancers ▼

Docker-Wordpress-LBC X
Classic Load Balancer

Criar e anexar novos balanceadores de carga

Adicionar um novo balanceador de carga

- Crie o seu **Auto Scaling** e aguarde a criação das instâncias.

Passo 8: Configuração da Saída do Wordpress pelo Load Balancer

- Acesse uma das instâncias criadas anteriormente via SSH.
- Acesse o **banco de dados RDS MySQL** pelos seguintes comandos
 - `export MYSQL_HOST=[Endpoint do RDS criado]`
 - `mysql --user=[Usuario] --password=[senha] [nome da base de dados]`

```
Install 1 Package

Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm | 8.8 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1
  Verifying : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[ec2-user@ip-10-0-10-93 ~]$ export MYSQL_HOST=wordpress.c7djeexh6rth.us-east-1.rds.amazonaws.com
[ec2-user@ip-10-0-10-93 ~]$ mysql --user=admin --password=wordpress wordpress
```

- Agora execute esse comando para **visualizar o endereço do Wordpress**
 - `SELECT * FROM wp_options WHERE option_name = 'siteurl' OR option_name = 'home';`

```
[ec2-user@ip-10-0-17-26 Docker]$ mysql --user=admin --password=wordpress wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 5781
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [wordpress]> SELECT * FROM wp_options WHERE option_name = 'siteurl' OR option_name = 'home';
+----+-----+-----+-----+
| option_id | option_name | option_value | autoload |
+----+-----+-----+-----+
| 2 | home | http://75.101.195.223 | yes |
| 1 | siteurl | http://75.101.195.223 | yes |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL [wordpress]>
```

- Agora altere o endereço para o DNS do load balancer com esse comando

- UPDATE wptions SET option_value = 'http://[DNS do Load Balancer]' WHERE option_name = 'siteurl' OR option_name = 'home';
- Execute novamente esse comando para **visualizar se o endereço do Wordpress foi alterado**
 - SELECT * FROM wp_options WHERE option_name = 'siteurl' OR option_name = 'home';

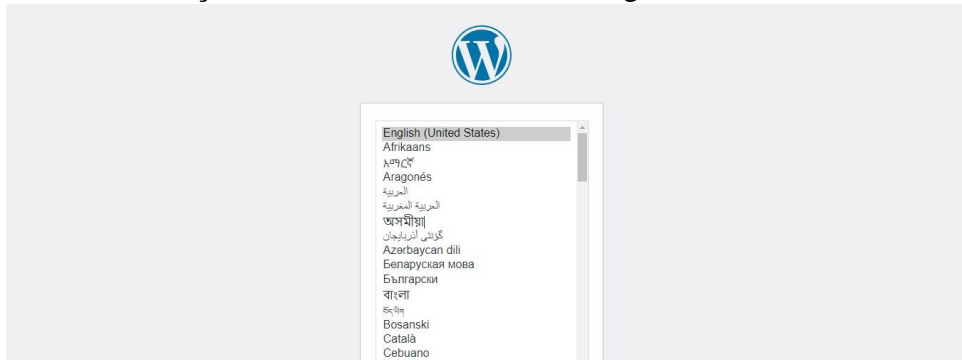
```
mysql [wordpress]> SELECT * FROM wp_options WHERE option_name = 'siteurl' OR option_name = 'home';
```

option_id	option_name	option_value	autoload
2	home	http://Docker-Wordpress-LBC-72442703.us-east-1.elb.amazonaws.com	yes
1	siteurl	http://Docker-Wordpress-LBC-72442703.us-east-1.elb.amazonaws.com	yes

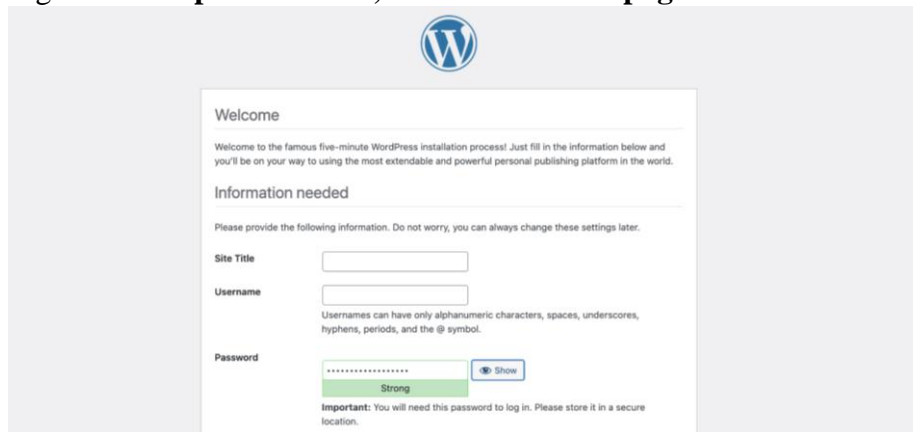
```
rows in set (0.00 sec)
```

Passo 9: Configuração do Wordpress

- Acesse o **endereço DNS do Load Balancer** configurado anteriormente



- Configure os **campos de e-mail, senha e nome da página**



- Pronto! Agora sua aplicação já está funcionando corretamente



Referências para a criação

- <https://aws.amazon.com/pt/tutorials/deploy-wordpress-with-amazon-rds/>
- <https://npaluri2.medium.com/building-a-scalable-wordpress-website-with-aws-ec2-rds-and-apache-webserver-c697d448296c>
- https://docs.aws.amazon.com/pt_br/efs/latest/ug/wt1-test.html
- <https://aws.amazon.com/pt/getting-started/hands-on/deploy-wordpress-with-amazon-rds/2/>
- https://docs.aws.amazon.com/pt_br/elasticloadbalancing/latest/classic/elb-getting-started.html