# **Documentação de Criação e Configuração**

## **Docker - Versão 1.0**

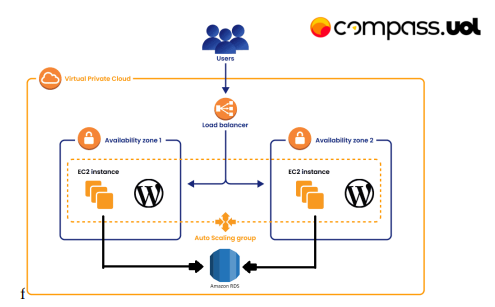
*Ana Maria Cavasin Data de Atualização: 30/10/2023*

### **Visão Geral**

A apresente documentação fornece instruções passo a passo para a instalação e configuração o de uma aplicação WordPress com RDS database MySQL entro 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 do 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 sugestão de utilizar o EFS (Elastic File Sistem).
* Usar Dockerfile ou Dockercompose.
* Aplicação Wordpress rodando na porta 80 ou 8080.

### **Desenvolvimento da Tarefa**

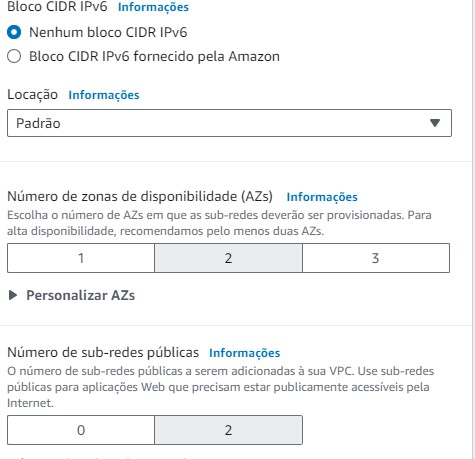
Siga estas etapas para realizar as configurações. Em cada passo será apresentado uma instrução escrita, bem como o tutorial em vídeo com o passo a passo.

**Passo 1: Criar uma nova VPC**

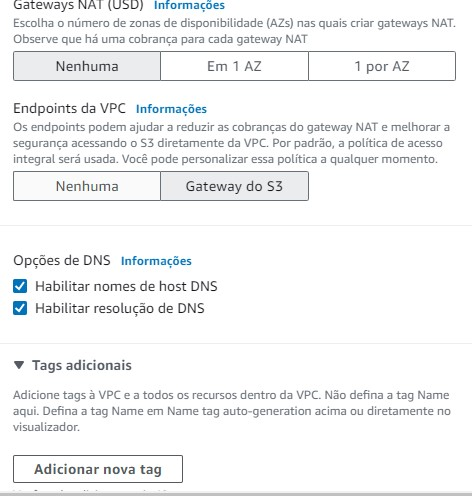
* No Console de Gerenciamento da AWS, acesse a opção **VPC**.
* Agora busque a opção **Suas VPCs**
* Após isso, clique em **Criar VPC**
* Selecione a opção **VPC e muito mais** e insira um **nome para VPC.**



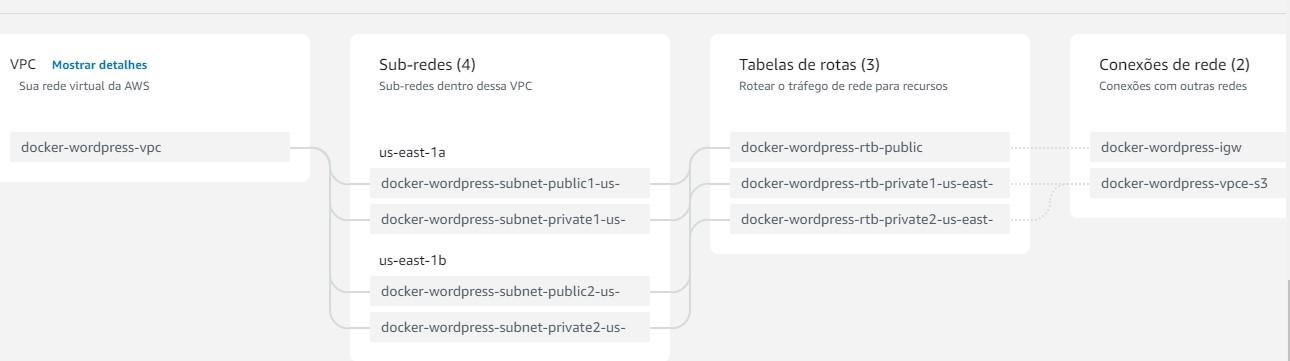
* Escolha o **Número de zonas de disponibilidade (AZs)** e o **Número de sub-redes públicas**.



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



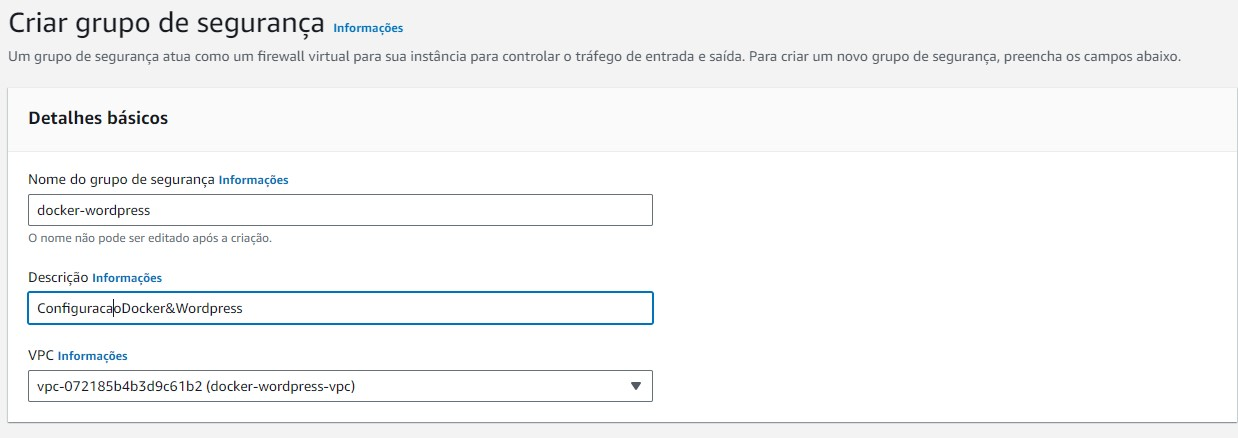
* No final, sua VPC deve ficar parecida com a seguinte.



* Agora clique em **Criar VPC**

**Passo 2: Criar um Grupo de Segurança**

* No Console de Gerenciamento da AWS, acesse **EC2**.
* Acesse a opção **Segurity 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



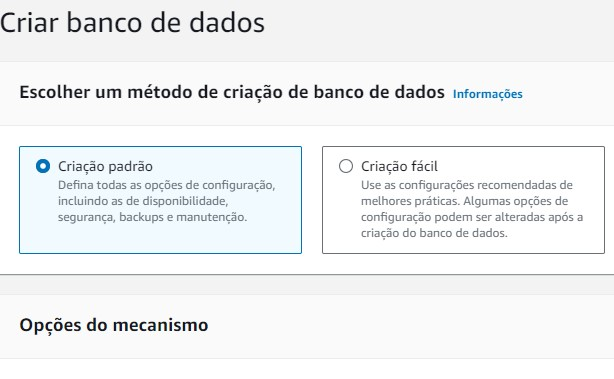
* Em **Regras de Entrada** defina as seguintes portas



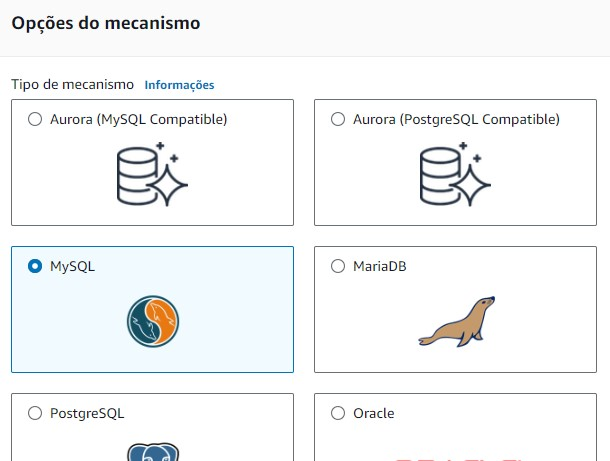
* Confime a criação do **Segurity Group**

**Passo 3: Criar o RDS database MySQL**

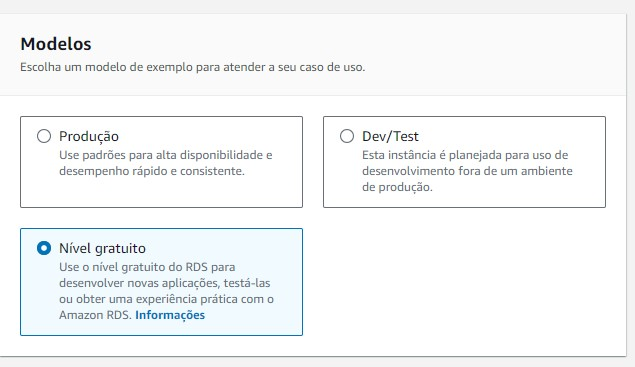
* No Console de Gerenciamento da AWS, acesse **RDS**.
* Clique em **Banco de Dados** e depois em **Criar Banco de Dados.**
* Escolha o método de **criação padrão**.



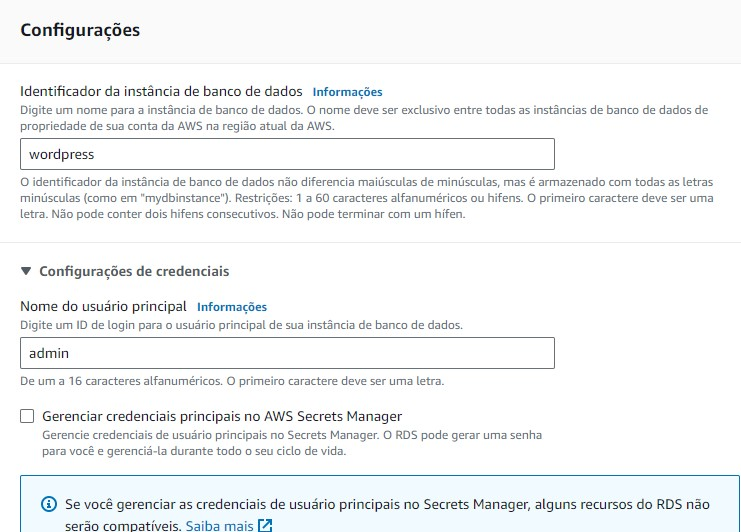
* Em **Opções do mecanismo** escolha o **MySQL.**



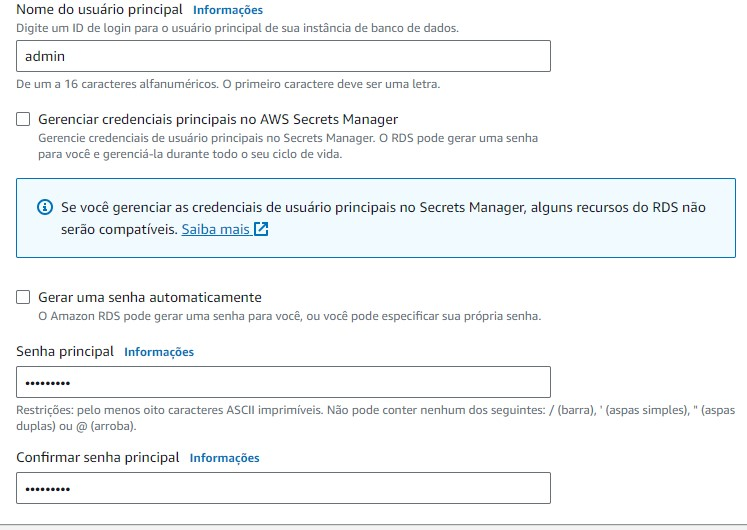
* No **Modelos** escolha a opção **Nível gratuito**.



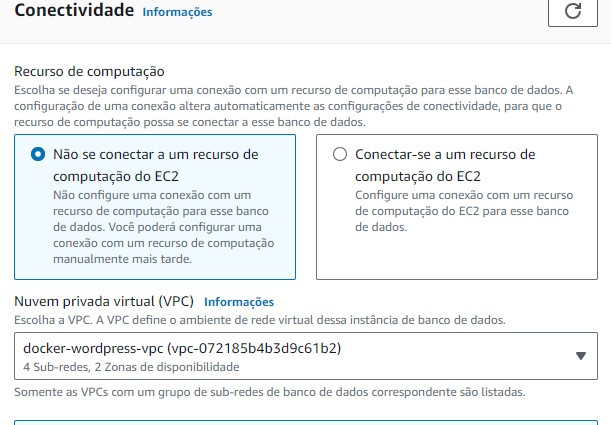
* Agora em **Configurações** escreva um **identificador da instancia do banco de dados.**



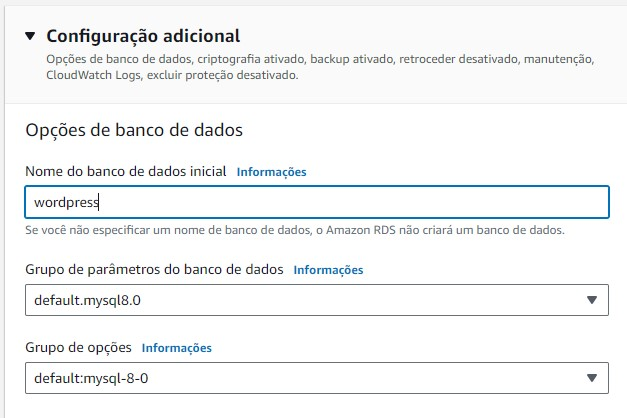
* Em **configurações de credencial**, escolha o **Nome do usuário principal** e a **senha principal.**



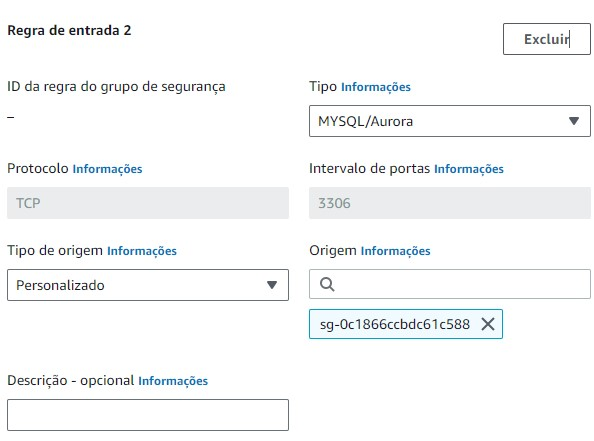
* Em **Conectividade**, selecione **Não se conectar a um recurso de computação EC2** e **selecione a VPC criada anteriormente** no passo 1.



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



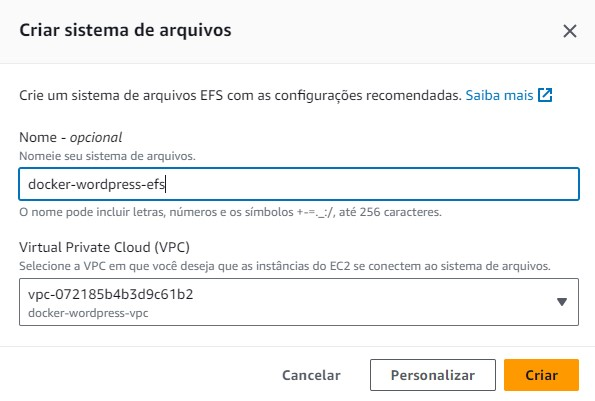
* As **demais opções ficam padrão.**
* Agora clique em **Criar banco de dados**
* Verifique se o **grupo de segurança** do banco de dados nas regras de entrada está liberado a seguinte porta



* Agora guarde as seguintes informações que serão uteis nos próximos passos

**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.



* Agora guarde a seguinte informação que iremos utilizar posteriormente
* 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 ulizamos 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 habilidação 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**
  + echo "[DNS do EFS AWS]:/ /mnt/efs nfs defaults 0 0" >> /etc/fstab #adiciona o monto de montagem no fstab
  + mount –a #faz a montagem do sistema de arquivos
* Agora vamos fazer a i**nstalação 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.**
* 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

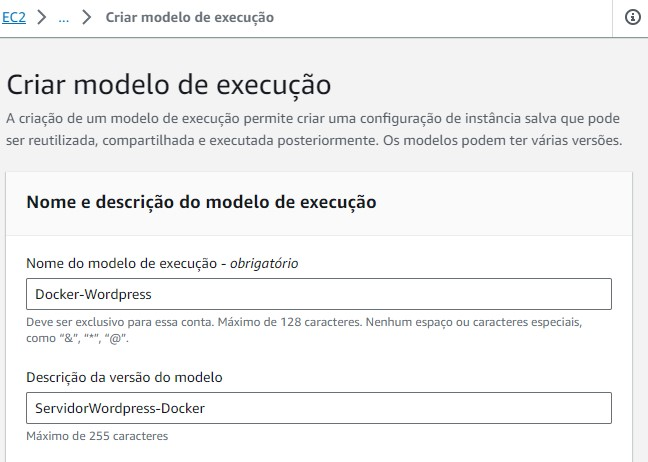
EOF

cd /Docker

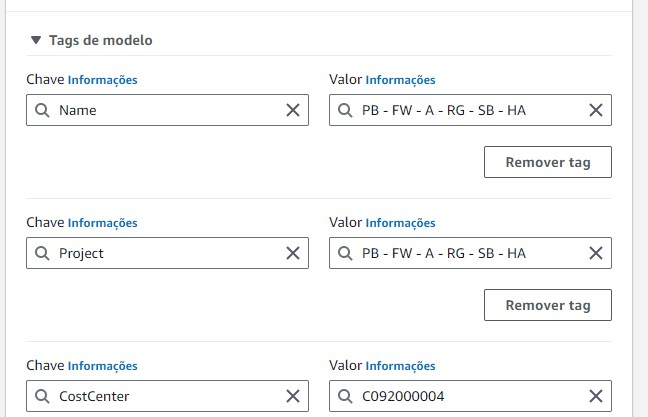
docker-compose up -d

**Passo 6: Criar o modelo de execução**

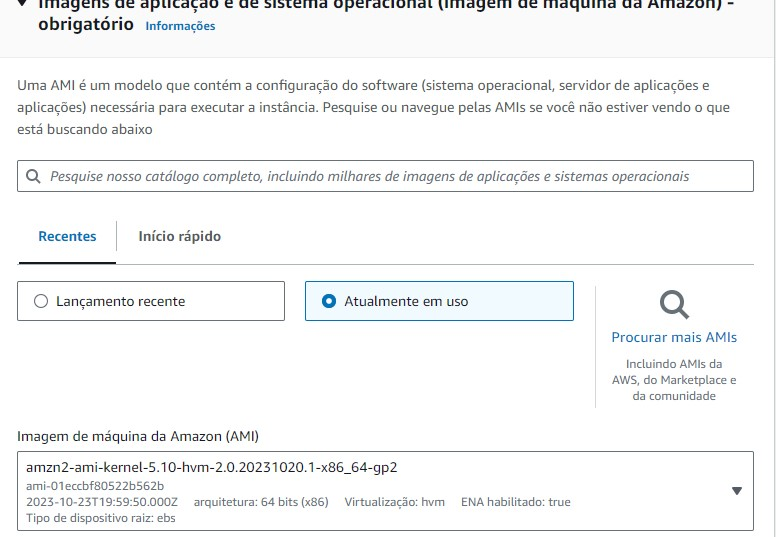
* No Console de Gerenciamento da AWS, acesse **EC2**.
* Vá na sessão Instancias e depois em **Modelos de Execução.**
* Click em **Criar modelo de execução**.
* Defina um **nome do modelo de execução** e a **descrição** da versão do modelo.



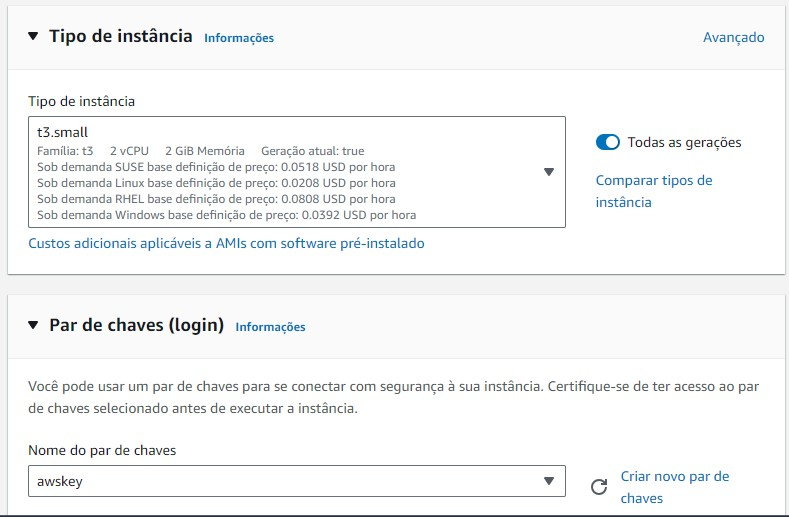
* Em **Tags do modelo**, adicione **todas as Tags padrões**.



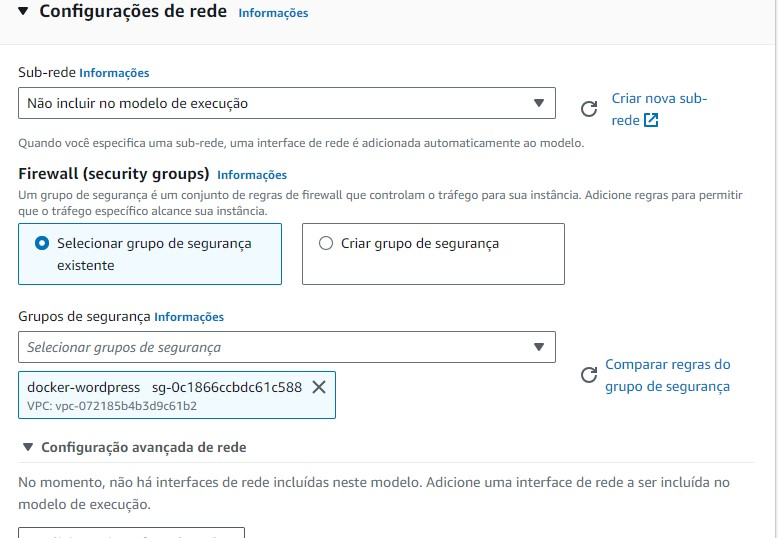
* Depois escolha a **Imagem da aplicação**, utilize a **Amazon Linux 2**.



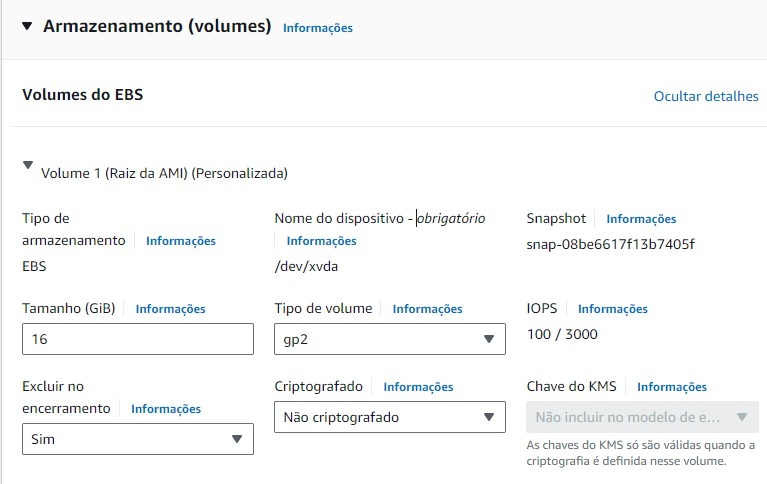
* Selecione o **tipo de instancia e o par de chaves para login.** O **tipo da instância deve ser o t3.smal**l e o **par de chaves pode ser um criado anteriormente ou um novo**.



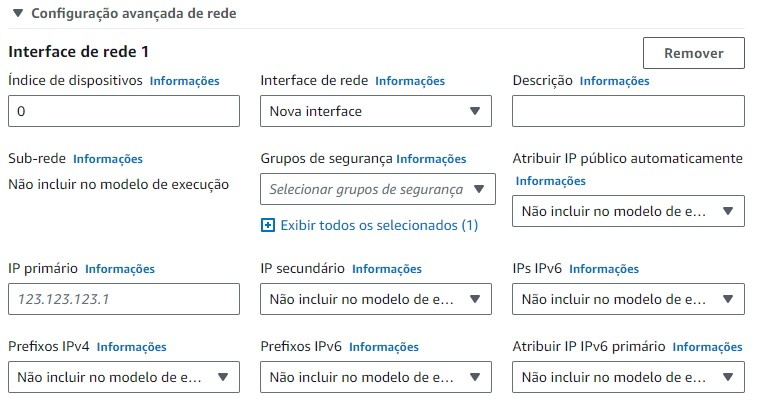
* Em **Configurações de rede**, no **Firewall(segurity groups)** escolha o **Segurity Group criado no passo 2**.



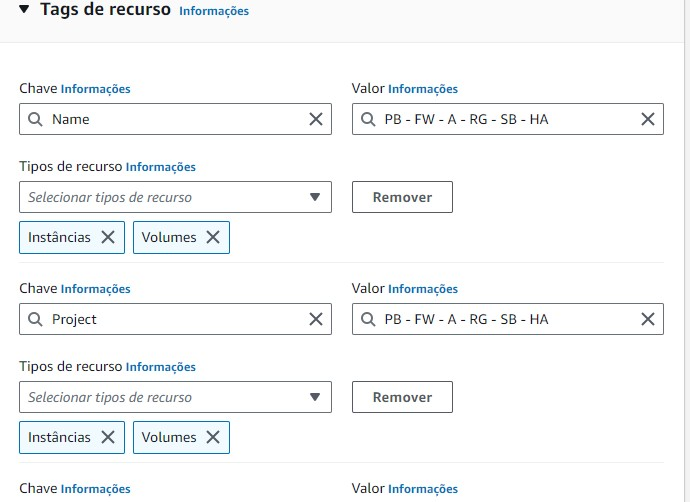
* No **Armazenamento**, configure o **Tamanho para 16 GiB.**



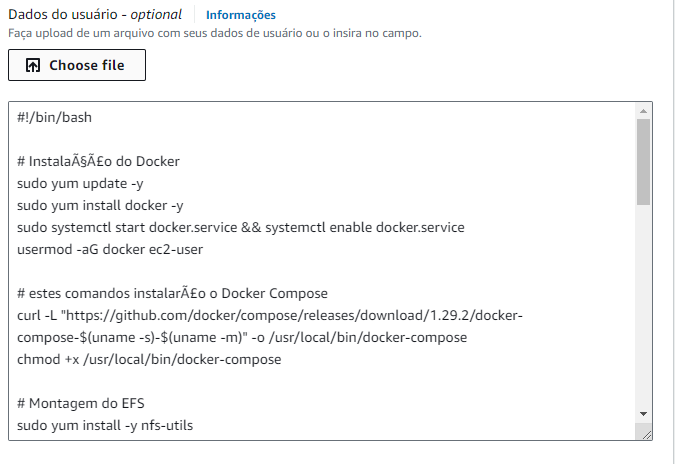
* Ainda em **Rede**, adicione uma **interface de rede conforme a imagem**.



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



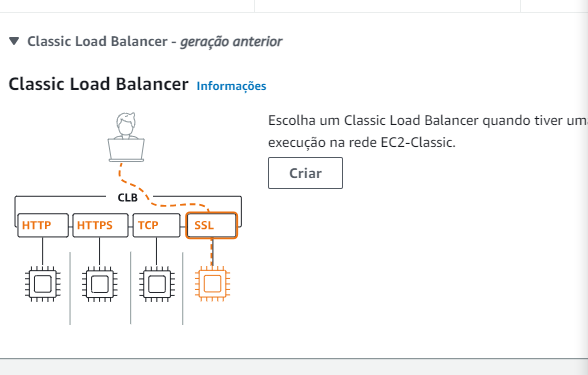
* Em **Detalhes Avançados**, em **Dados do usuário**, adicione o **arquivo user\_data.sh** criado no passo 5.



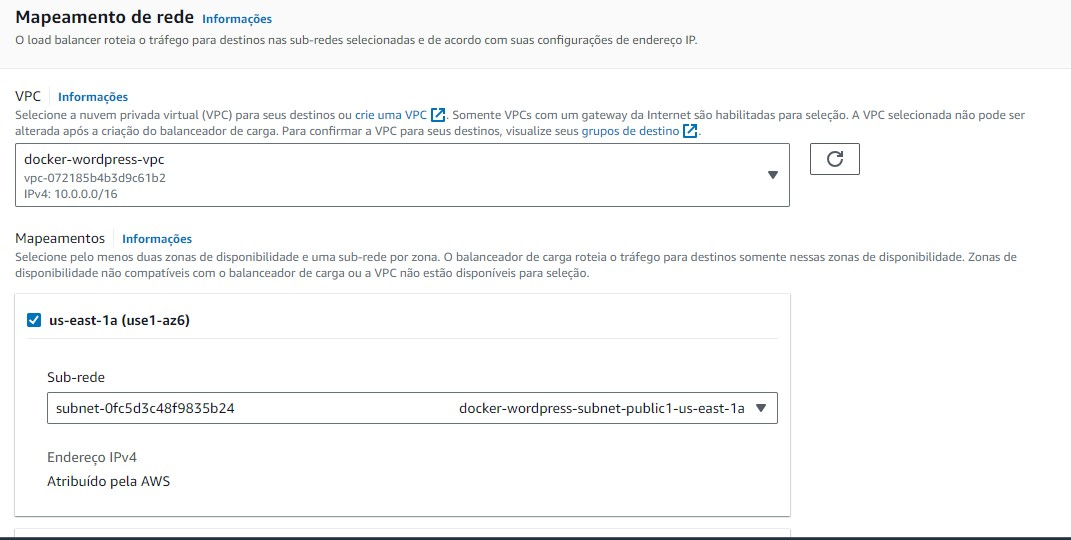
* Confira as informações e clique em **Criar versão do modelo.**

**Passo 7: Criar Load Balacing**

* 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.**

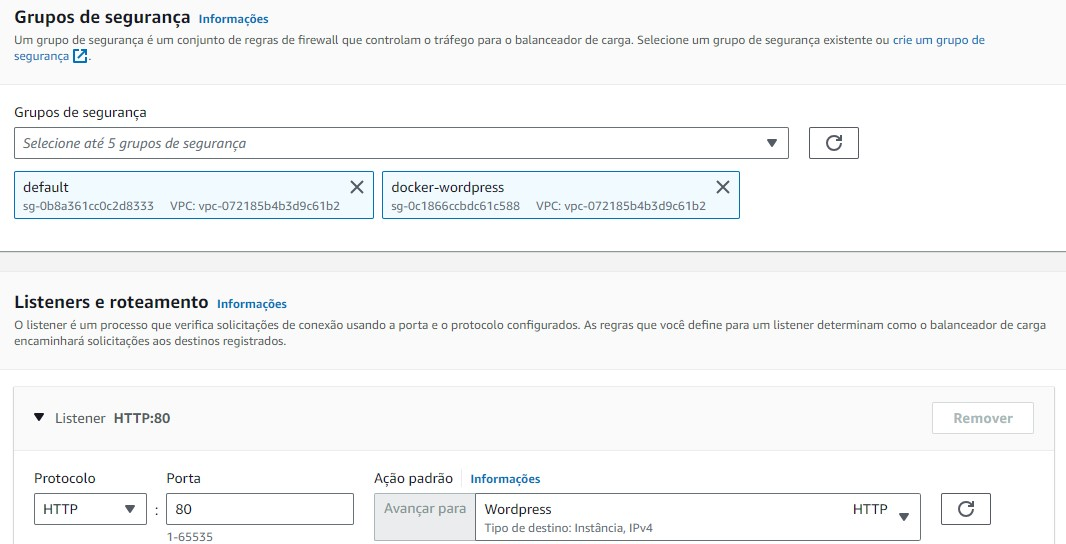


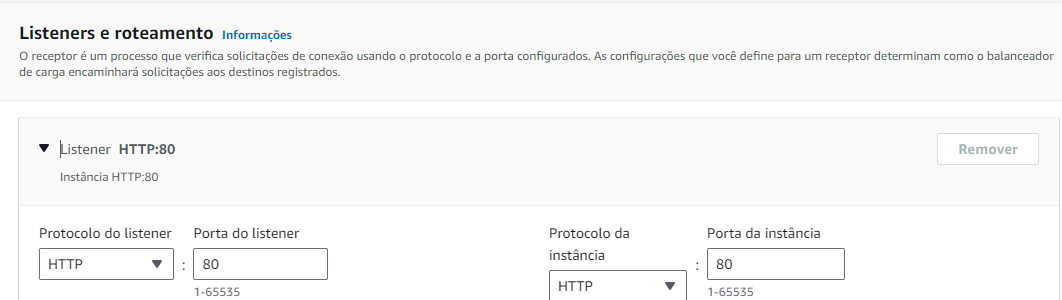
* Depois defina o **Nome do load balancer** e o **Esquema Voltado para a internet**.
* Em **Mapeamento de rede**, selecione a **VPC criada no passo 1** e no **Mapeamento selecione as duas AZs e as respectivas sub-redes públicas.**





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

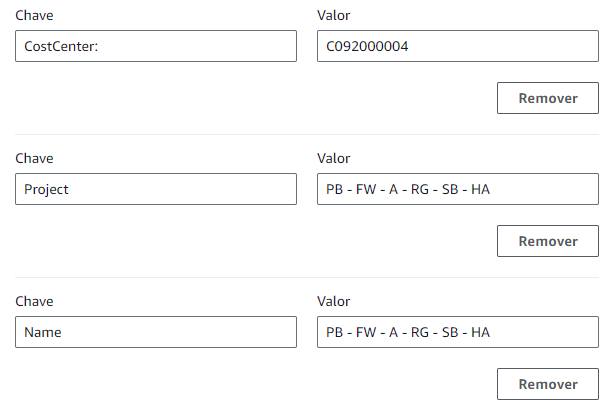




* Em **verificação de integridade**, deixe o caminho conforme a imagem.



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



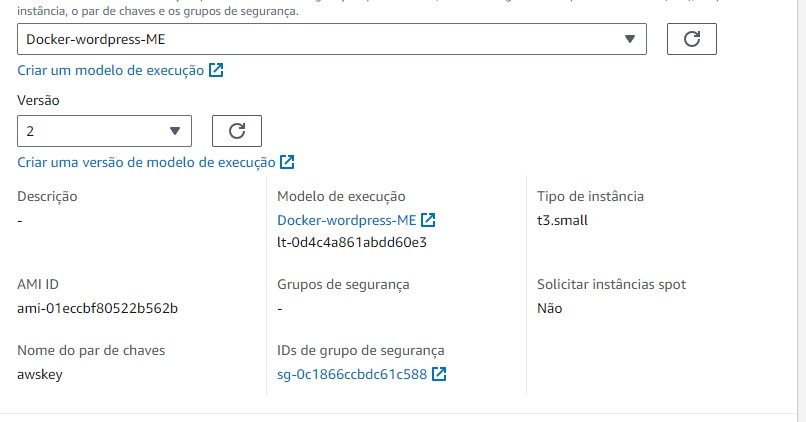
* **As demais, configurações deixe padrão**, agora clique em **Criar Load Balancer**.

**Passo 8: Criar Auto Scaling**

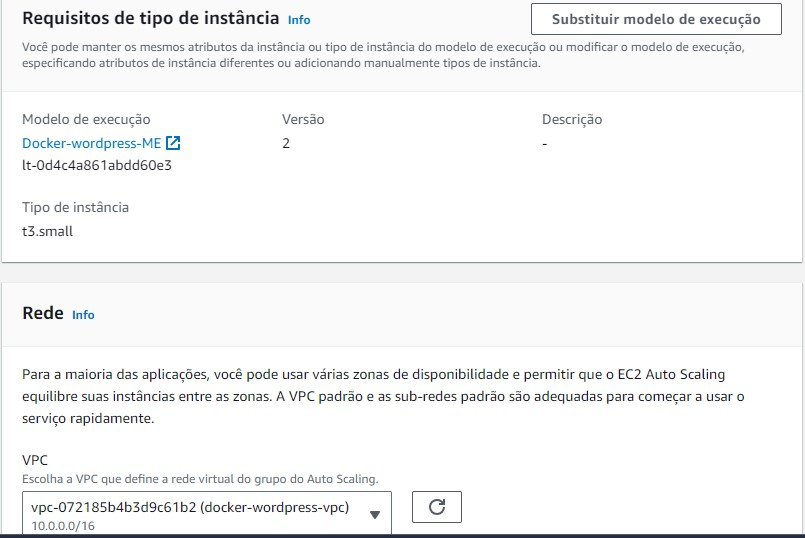
* No Console de Gerenciamento da AWS, acesse **EC2**.
* Vá na sessão Auto Scaling e depois em **Grupos de Auto Scaling.**
* Click em Criar **Grupos de Auto Scaling**.
* Escolha o **Nome do Auto Scaling.**



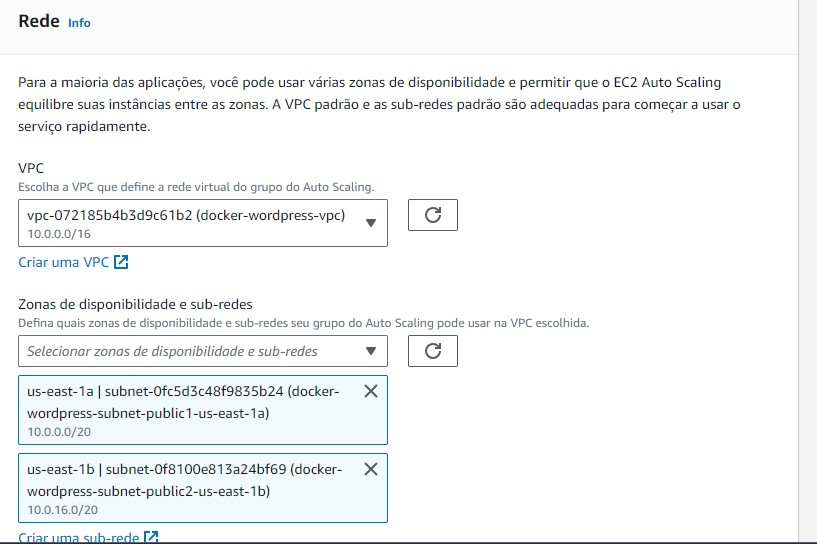
* Agora selecione o **modelo de execução criado anteriormente**.



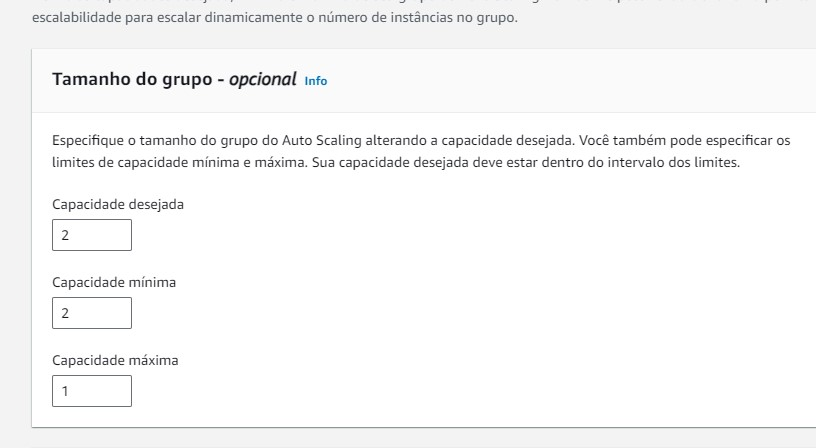
* Verifique se os Requisitos de tipo de instancia estão corretos.



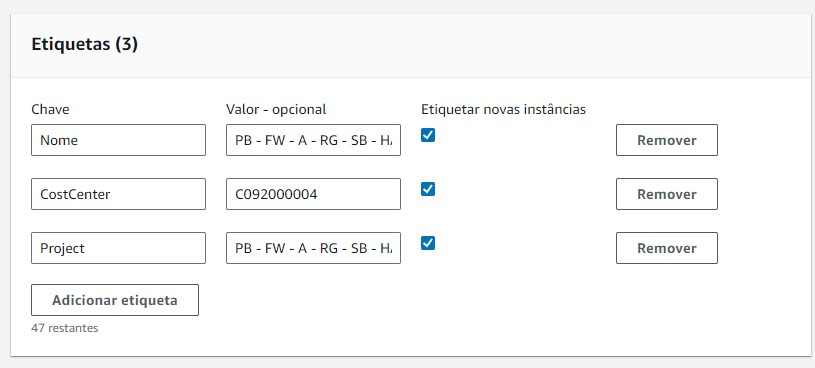
* Em **Rede**, selecione a **VPC criada anteriormente e selecione as Zonas de disponibilidade e as sub-redes públicas**.



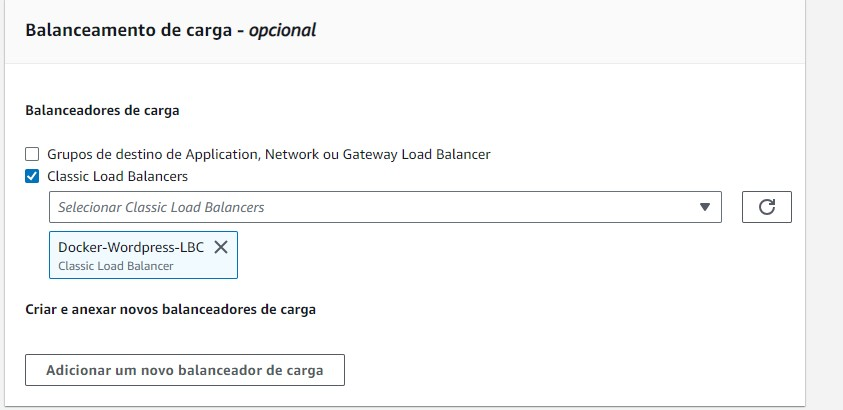
* Em **Tamanho de grupo**, configure a **capacidade desejada.**



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



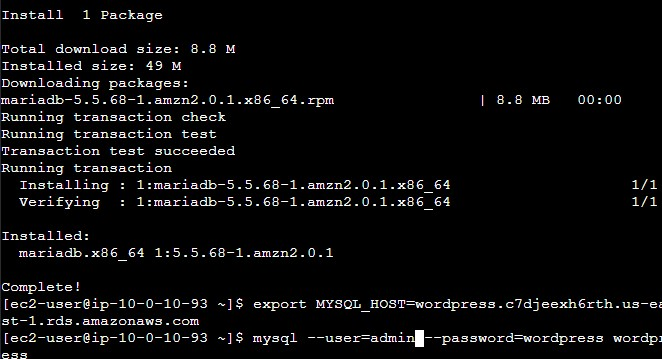
* Em **Balanceamento de carga**, selecione o **Load Balancer criado anteriormente.**



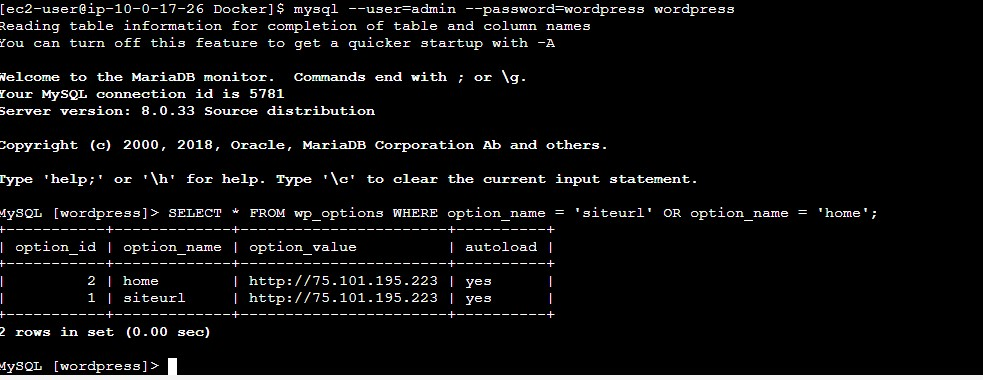
* **Crie** o seu **Auto Scaling e aguarde a criação das instâncias**.

**Passo 8: Configuração da Saída do Wordpress pelo Load Balancer**

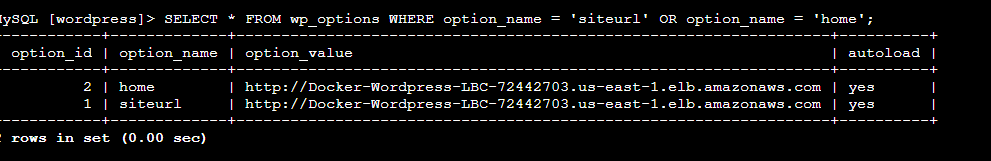
* Acesse uma das i**nstâncias criadas anteriormente via SSH**.
* Acesse o **banco de dados RDS MySQL pelos seguintes comandos**
  + export MYSQL\_HOST=[Endpoint do RDS criado]
  + mysql --user=[Usuario] --password=[senha] [nome da base de dados]



* Agora execute esse comando para **visualizar o endereço do Wordpress**
  + SELECT \* FROM wp\_options WHERE option\_name = 'siteurl' OR option\_name = 'home';

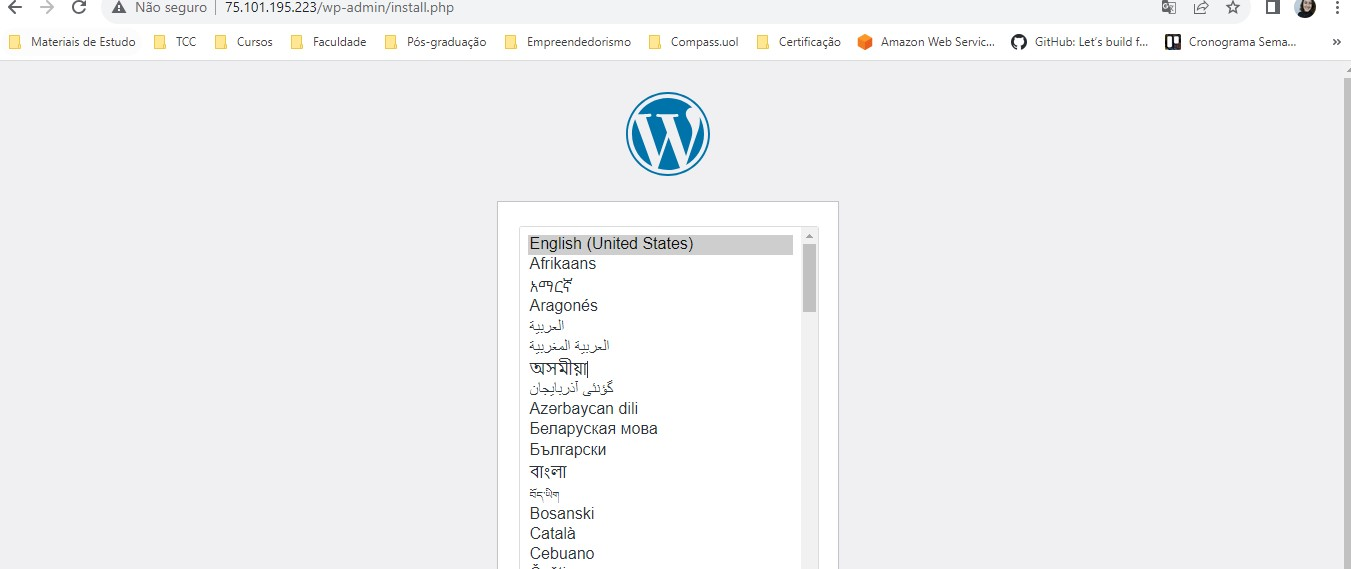


* Agora **altere o endereço para o DNS do load balancer** com esse comando
  + UPDATE wpoptions SET option\_value = 'http://[ DNS do Load Balancer]' WHERE option\_name = 'siteurl' OR option\_name = 'home'; = 'home';
* Execute novamente esse comando para **visualizar se o endereço do Wordpress foi alterado** 
  + SELECT \* FROM wp\_options WHERE option\_name = 'siteurl' OR option\_name = 'home';

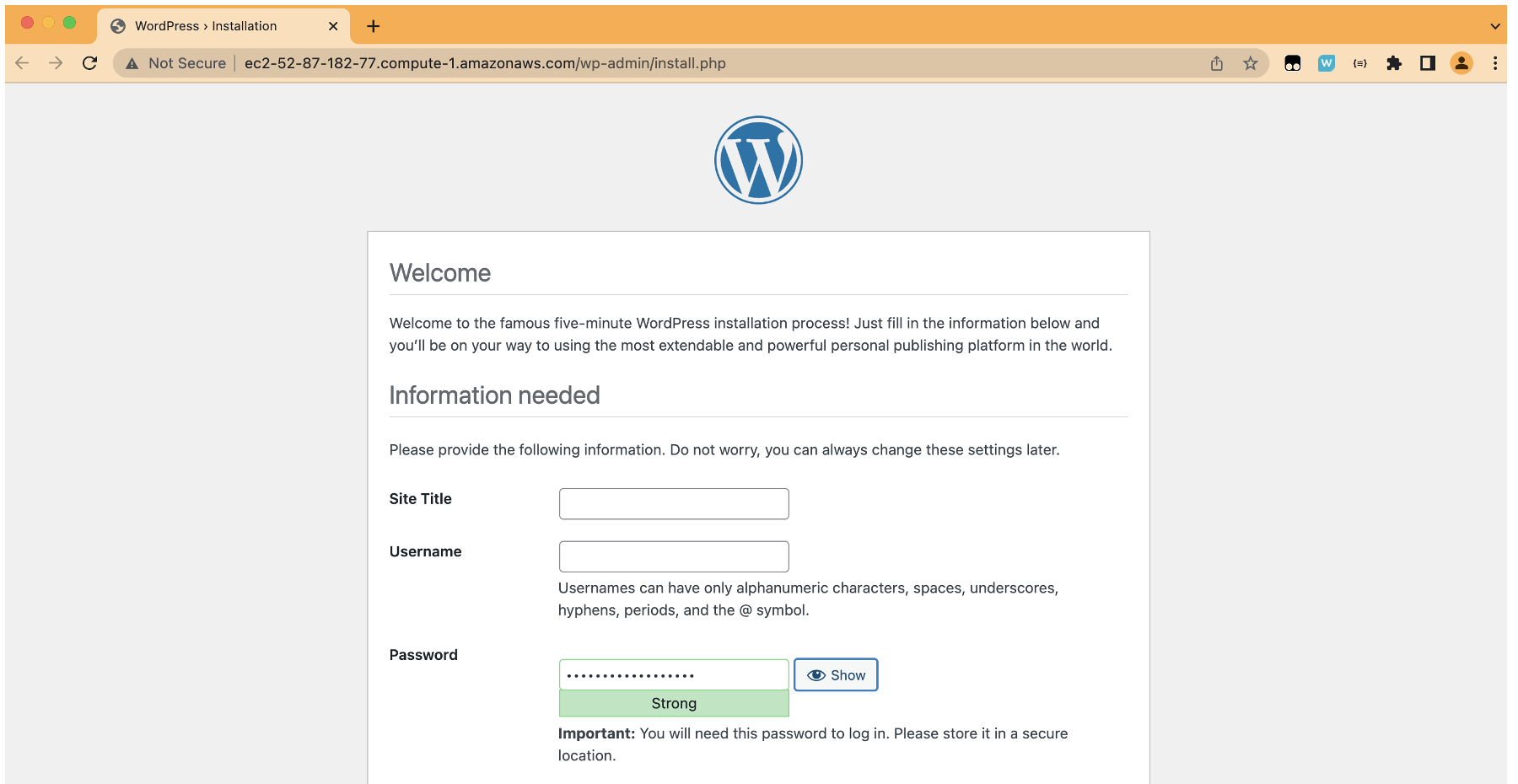


**Passo 9: Configuração do Wordpress**

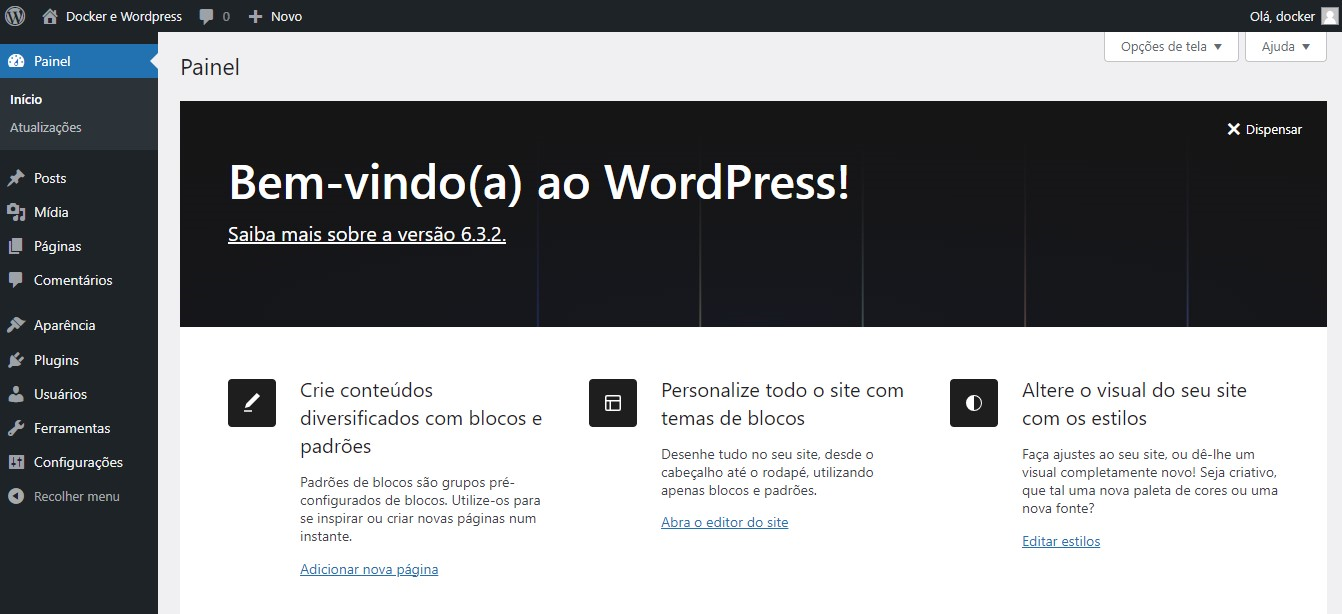
* Acesse o **endereço DNS do Load Balancer** configurado anteriormente



* 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>