**Implantação do Wordpress com Docker e AWS**

**Descrição**

O projeto tem como objetivo a criação de uma aplicação Wordpress em uma instância Amazon EC2 com ambiente baseado em Linux utilizando Docker-Compose junto de um sistema livre para gestão de conteudo na internet, o Wordpress em uma instância RDS com o banco de dados MYSQL. Além da instância principal, o Docker-Compose e o Wordpress, também foram necessários criar uma instância a mais para funcionar como Bastion Host para o acesso da principal, uma VPC para as instâncias e um Load Balancer para a organização do tráfego. Também foi usado EFS para o armazenamento estático dos arquivos.

**Pré-requisitos**

* Possuir uma conta na AWS com permissões suficientes para criar e configurar os serviços mencionados.
* Conhecimento básico de Docker e containers.
* Conhecimento básico de Git e repositórios.

**Instalação e Configuração do Docker na instância**

1. Criação de uma instância EC2 com ambiente AWS Linux 2.
2. Durante a criação da instância, no campo "User data", foi adicionado o script abaixo para instalação automática do Docker:

#!/bin/bash

sudo yum update -y

sudo yum install -y docker

sudo systemctl docker start

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

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

mv /usr/local/bin/docker-compose /bin/docker-compose

**Upload no GitHub**

1. Criação de um repositório destinado para o trabalho.
2. Download do Git na instância através do comando “sudo yum install git”.
3. Inicialização do Git usando o comando “git init”.
4. Comando “git status” para verificar se há arquivo(s) não adicionado(s) e comando “git add “[nome do arquivo]”” para adicioná-lo(s).
5. Configuração do git com os comandos “git config --global user.email “[email]”” e “git config --global user.name “[nome]””.
6. Commit através do comando “git commit -m “nome do commit””.

**Uso do Docker**

1. Execução do script através do comando “docker-compose up -d” em segundo plano.
2. Script:

#!/bin/bash

version: "3"

services:

db:

image: mysql:5.7

restart: always

environment:

MYSQL\_HOST: database-wordpress.cdhzsu6mbxkf.us-east-1.rds.amazonaws.com

MYSQL\_ROOT\_PASSWORD: senhabanco

MYSQL\_DATABASE: bancoDeDados

MYSQL\_USER: admin

MYSQL\_PASSWORD: senhabanco

wordpress:

depends\_on:

- db

image: wordpress:latest

restart: always

ports:

- "80:80"

environment:

WORDPRESS\_DB\_HOST: database-wordpress.cdhzsu6mbxkf.us-east-1.rds.amazonaws.com

WORDPRESS\_DB\_USER: admin

WORDPRESS\_DB\_PASSWORD: senhabanco

WORDPRESS\_DB\_NAME: bancoDeDados

volumes:

- "/var/www/html:/var/www/html"

volumes:

mysql: {}

1. Acesso ao endereço IP da instância EC2 no navegador para verificar se o Wordpress está funcionando.

**Configuração da VPC**

1. Acesso ao serviço VPC;
2. Criação da VPC;
3. Criação das subnets em duas zonas de disponibilidades diferentes, uma privada e uma pública na zona 1a e 1b;
4. Criação do internet gateway e NAT gateway;
5. Criação da tabela de rotas.

**Configuração do EFS**

1. Criação e configuração do EFS.
2. Criação e configuração do NFS.

**Configuração do Load Balancer**

1. Acesso ao serviço EC2.
2. Criação do Classic Load Balancer:
   1. Configuração do LB (nome, VPC, security groups...)
   2. Ativar as verificações de integridade de instâncias.
   3. Criação do target group;
   4. Seleção das instâncias que o LB irá atuar e habilitar o balanceamento de carga.
   5. Configuração do LB para permitir a saída tráfego de internet.
   6. Adição de tags.