

**Exercício Sobre Conhecimentos Gerais Adquiridos Nos Cursos
Aws, Docker & Linux**

AWS, DOCKER & LINUX

Criado e editado por: Luiz Felipe Lazarotto Pires

Versão final



Frederico Westphalen – RS

Nesta atividade usamos o User_data para fazer toda a configuração do modelo de instancia EC2 para o nosso site WordPress funcionar, abaixo vou listar e explicar o User_data e o que cada comando faz.

```
#!/bin/bash
yum update -y

sudo amazon-linux-extras install docker -y
sudo systemctl start docker
sudo systemctl enable docker

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

sudo mkdir /wordpress
cd /wordpress
sudo mkdir efs
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0a22cae163b271a35.efs.us-east-1.amazonaws.com:/ /wordpress/efs

echo "version: '3.7'"
services:
  wordpress:
    image: wordpress
    volumes:
      - /efs/website:/var/www/html
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: wordpress1.c3j6s4qzbagx.us-east-1.rds.amazonaws.com
      WORDPRESS_DB_USER: admin
      WORDPRESS_DB_PASSWORD: Luizfelipe28
      WORDPRESS_DB_NAME: wordpress
      WORDPRESS_TABLE_PREFIX: wp" | sudo tee /docker-compose.yml

cd /
docker-compose up -d
```

Acima está listado todos comandos utilizados no User_data;

1. `#!/bin/bash`: Esta linha diz ao sistema que o script deve ser executado usando o interpretador de comandos "bash".
2. `yum update -y`: Isso atualiza todos os programas no sistema.
3. `sudo amazon-linux-extras install docker -y`: Aqui, estamos instalando o Docker, uma plataforma para executar aplicativos em contêineres.
4. `sudo systemctl start docker`: Inicia o Docker, tornando-o operacional.
5. `sudo systemctl enable docker`: Isso faz com que o Docker seja iniciado automaticamente sempre que o sistema é reiniciado.
6. `sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose`: Baixa o Docker Compose, uma ferramenta que ajuda a gerenciar aplicativos em contêineres.
7. `sudo chmod +x /usr/local/bin/docker-compose`: Permite que o Docker Compose seja executado.
8. `sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose`: Cria um atalho para o Docker Compose, tornando-o mais fácil de usar.
9. `sudo mkdir /wordpress`: Cria uma pasta chamada "wordpress" na raiz do sistema.

10. ``cd /wordpress``: Move o diretório de trabalho atual para a pasta "wordpress".
11. ``sudo mkdir efs``: Cria uma subpasta chamada "efs" dentro da pasta "wordpress".
12. ``sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0a22cae163b271a35.efs.us-east-1.amazonaws.com:/ /wordpress/efs``: Conecta o sistema de arquivos remoto na pasta "efs", que será usada para armazenar os arquivos do WordPress.
- 13.

```
echo "version: '3.7'
services:
  wordpress:
    image: wordpress
    volumes:
      - /efs/website:/var/www/html
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: wordpress1.c3j6s4qzbagx.us-east-1.rds.amazonaws.com
      WORDPRESS_DB_USER: admin
      WORDPRESS_DB_PASSWORD: Luizfelipe28
      WORDPRESS_DB_NAME: wordpress
      WORDPRESS_TABLE_PREFIX: wp"| sudo tee /docker-compose.yml
```

"`sudo tee docker-compose.yml`": Cria um arquivo de configuração do Docker Compose chamado que especifica como o contêiner do WordPress deve ser configurado.

14. ``cd /``: Volta para a pasta raiz do sistema.
15. ``docker-compose up -d``: Inicia o contêiner do WordPress, permitindo que seu site seja executado. O "-d" significa que o contêiner será executado em segundo plano.

Este script automatiza a instalação do Docker, a configuração do Docker Compose e a montagem de um sistema de arquivos remoto para armazenar os arquivos do WordPress. Em seguida, ele inicia um contêiner do WordPress para executar o site.

Para acessarmos a página inicial do WordPress através do DNS do load balance devemos configurar dentro da instância, com os seguintes comandos:

1. ``mysql -u admin -p -h wordpress1.c3j6s4qzbagx.us-east-1.rds.amazonaws.com -P 3306 -D wordpress``: Este comando é usado para acessar um banco de dados MySQL hospedado em um servidor chamado "wordpress1.c3j6s4qzbagx.us-east-1.rds.amazonaws.com". Aqui está o que cada parte significa:
 - ``-u admin``: Isso especifica que você está se conectando como o usuário "admin".
 - ``-p``: Isso solicitará a senha do usuário após a execução do comando.
 - ``-h wordpress1.c3j6s4qzbagx.us-east-1.rds.amazonaws.com``: Isso indica o host ou servidor ao qual você está se conectando.
 - ``-P 3306``: Isso define a porta na qual o banco de dados está ouvindo (geralmente a porta padrão do MySQL é 3306).
 - ``-D wordpress``: Isso especifica que você está se conectando ao banco de dados chamado "wordpress".

2. ``SELECT * FROM wptions WHERE option_name = 'siteurl' OR option_name = 'home';``: Este é um comando SQL que está sendo executado no banco de dados. Ele está selecionando todas as linhas da tabela "wptions" onde o nome da opção é igual a 'siteurl' ou 'home'. Isso é usado para recuperar informações sobre a URL do site.

3. ``UPDATE wptions SET option_value = 'http://wordpress1.c3j6s4qzbagx.us-east-1.rds.amazonaws.com' WHERE option_name = 'siteurl' OR option_name = 'home';``: Este é outro comando SQL que está sendo executado no banco de dados. Ele está atualizando os valores das opções 'siteurl' e 'home' na tabela "wptions". Isso é feito para ajustar a URL do site para o novo local.