

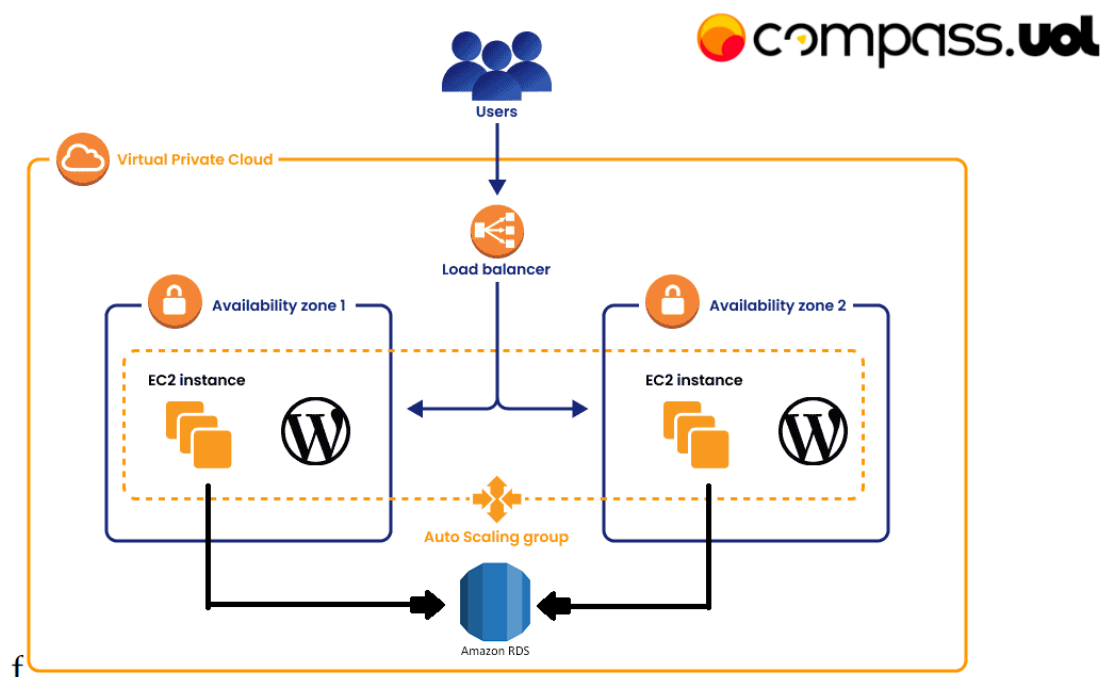
**ATIVIDADE REFERENTE A CONHECIMENTOS GERAIS
ADQUIRIDOS NOS CURSOS DE AWS, LINUX E CONTAINERS NA
TRILHA DE APRENDIZADO DO ESTÁGIO**

AWS & DEVSECOPS COMPASS UOL

Bolsista: Igor Henrique Buzatto

FREDERICO WESTPHALEN – RS

1 - ATIVIDADE REFERENTE A PARTE PRÁTICA SOBRE DOCKER

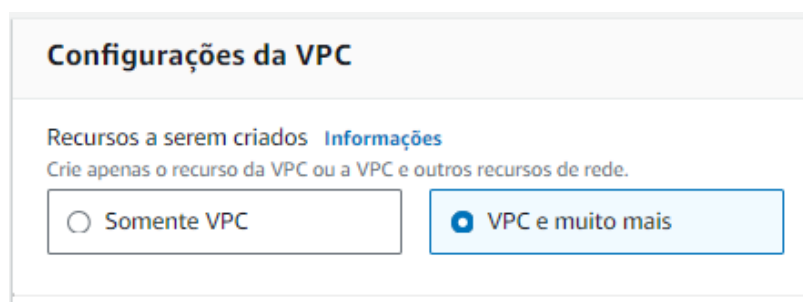


Seguindo o desenho da topologia disposta devemos realizar todas as configurações necessárias para que tudo funcione conforme solicitado.

1.1 – CONFIGURAÇÃO INICIAL DO AMBIENTE AWS

Inicialmente devemos realizar a configuração de uma estrutura de rede para que possamos criar todos os serviços e possibilitar a comunicação entre eles.

Iniciamos com a configuração de uma VPC (virtual private cloud) que é como se fosse a rede interna de uma empresa, onde as instancias e serviços que utilizarem essa VPC irão se comunicar. No menu de criação da VPC selecionamos criar uma “**VPC e Muito mais**”, conforme a figura abaixo.



Essa configuração permitirá a criação de toda a estrutura de uma só vez, contendo a VPC, subnets pública e privadas, tabelas de rotas e algumas outras configurações.

1.2 – CONFIGURAÇÃO DOS BANCOS DE DADOS EFS E RDS PARA ARMAZENAMENTO DO WORDPRESS

Nesta etapa devemos criar os bancos de dados para o armazenamento do nosso site wordpress, são necessários 2 tipos de banco de dados o Amazon RDS, que servirá para a configuração do banco de dados MYSQL, e o EFS, que servirá para o armazenamento de pastas públicas e estáticos do wordpress.

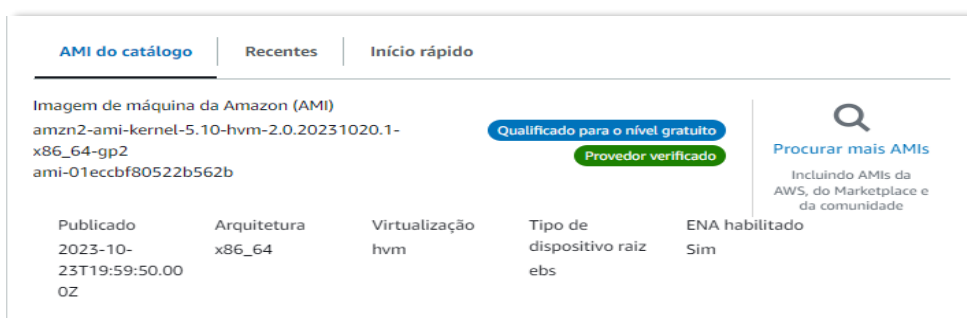
Acessando o menu **RDS** na AWS, selecionamos a opção “**Criar banco de dados**”, então a página de criação irá aparecer. Nas primeiras opções selecionamos o método de criação padrão, e o banco de dados **MySQL**.

Na aba “**modelos**” selecionamos o modelo de acordo com a necessidade, no meu caso selecionei o modelo gratuito para uso no laboratório. NA aba “**configurações**” criamos um nome para o banco de dados, um usuário e uma senha. Na aba “**conectividade**” selecionamos a VPC criada anteriormente, um grupo de segurança, que é criado automaticamente. Então é só revisar as configurações e criar o banco de dados.

Para a criação do banco de dados EFS, devemos acessar a página do EFS e clicar na opção “**criar sistema de arquivos**”. Digitamos um nome para o EFS e selecionamos a VPC criada. Então ao clicar em criar o sistema EFS estará criado.

1.3 –CRIANDO CONFIGURAÇÃO DA INSTANCIA COM CONTAINER WORDPRESS

Primeiro devemos executar uma instancia com as configurações fornecidas, tais como, tags, versão da AMI, tipo de instancia e volumes de armazenamento, visando testar todas as configurações para garantir um bom funcionamento futuro.



▼ 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

Volumes do EBS

Ocultar detalhes

▶ Volume 1 (Modelo e raiz da AMI) (16 GiB, EBS, SSD de uso geral (gp2))

Os volumes da AMI não são incluídos no modelo, a menos que sejam modificados

Adicionar novo volume

▼ Tags de recurso

Informações

Chave

Informações

Q Name X

Valor

Informações

Q PB - FW - A - RG X

Tipos de recurso

Informações

Selecione tipos d... ▼

Instâncias X

Volumes X

Remover

Chave

Informações

Q CostCenter X

Valor

Informações

Q C092000004 X

Tipos de recurso

Informações

Selecione tipos d... ▼

Instâncias X

Volumes X

Remover

Chave

Informações

Q Project X

Valor

Informações

Q PB - FW - A - RG X

Tipos de recurso

Informações

Selecione tipos d... ▼

Instâncias X

Volumes X

Remover

Adicionar nova tag

Você pode adicionar até mais 47 etiquetas.

Após a configuração das opções devemos então criar a instancia. Para iniciar a configuração acessamos a instancia e iniciamos com a execução de alguns comandos.

1. **“yum update -y”**: Isso atualiza todos os programas no sistema.
2. **“sudo amazon-linux-extras install docker -y”**: Aqui, estamos instalando o Docker.

3. **"sudo systemctl start docker"**: Inicia o Docker, tornando-o operacional.
4. **"sudo systemctl enable docker"**: Isso faz com que o Docker seja iniciado automaticamente sempre que o sistema for reiniciado.
5. **"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.
6. **"sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose"**: Torna o Docker Compose executável.
7. **"sudo mkdir /wordpress-config/efs"**: Cria um caminho contendo as pastas /wordpress-config e /efs.
8. **"sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-0930255a353754b99.efs.us-east-1.amazonaws.com:/ /wordpress-config/efs"**: Monta na pasta /wordpress-config/efs o sistema EFS que criamos anteriormente.

Agora com tudo montado, instalado e executando, devemos criar um arquivo chamado **"docker-compose.yml"** na pasta **"/wordpress-config"**, esse arquivo contém toda a informação necessária para a criação, conexão e execução do container wordpress. O arquivo ficará dessa forma:

```
version: '2.2'
services:
  wordpress:
    image: wordpress
    volumes:
      - /wordpress-config/efs/website:/var/www/html
    ports:
      - 80:80
    restart: always
    environment:
      WORDPRESS_DB_HOST: wp-rds.cexp47lsdaww.us-east-1.rds.amazonaws.com
      WORDPRESS_DB_USER: admin
      WORDPRESS_DB_PASSWORD: Admin123
      WORDPRESS_DB_NAME: wordpress
      WORDPRESS_TABLE_CONFIG: wp_
```

Após a criação do arquivo, acessamos a pasta em que ele foi criado e executamos o comando **"docker-compose up -d"**. Esse comando irá iniciar o container com wordpress.

Para ter certeza de que tudo funcionou devemos acessar o IP publico da instancia e a tela de instalação do wordpress deverá aparecer. Se a tela aparecer, significa que o container está iniciado e conectado ao banco de dados e sistema de arquivos.

1.4 – CRIANDO O SCRIPT USERDATA E MODELO DE EXECUÇÃO

Após verificar que tudo está funcionando podemos criar um script para inserir no userdata, para que durante a criação da instancia todos os comandos e instalações sejam executados automaticamente. Nosso userdata ficará assim:

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install docker -y
sudo service docker start
sudo usermod -a -G docker ec2-user
sudo chkconfig docker on
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-config
cd /wordpress-config
sudo mkdir efs
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrns=2,noresvport fs-021c572c794566fd3.efs.us-east-1.amazonaws.com:/ /wordpress-config/efs

echo "version: '2.2'"
services:
  wordpress:
    image: wordpress
    volumes:
      - /wordpress-config/efs/website:/var/www/html
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: wp-rds.cexp471sdaww.us-east-1.rds.amazonaws.com
      WORDPRESS_DB_USER: admin
      WORDPRESS_DB_PASSWORD: Admin123
      WORDPRESS_DB_NAME: wordpress
      WORDPRESS_TABLE_CONFIG: wp_" | sudo tee /wordpress-config/docker-compose.yml
docker-compose up -d
```

O script começa com o comando “**#!/bin/bash**”, dizer ao sistema que o script deve ser executado usando o interpretador de comandos “**bash**”. Após isso são inseridos todos os comandos citados anteriormente. Apenas ao criar o arquivo “**docker-compose.yml**” usamos o comando “**echo**”, que irá escrever o comando no local determinado. Então o comando “**docker-compose up -d**” é executado para iniciar o container wordpress.

Com o script criado podemos criar agora um modelo de execução para iniciar as instâncias com o autoscaling posteriormente. No menu Modelos de execução selecionamos a opção “**Criar Modelo de Execução**”, então inserimos todas as configurações citadas na seção 1.3, a única diferença é inserir o script userdata na aba “**detalhes avançados**”, na opção “**dados do usuário**”.

Dados do usuário - optional [Informações](#)

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

☐ Os dados do usuário já foram codificados em base64

1.5 – CRIAÇÃO DO AUTOSCALING E LOAD BALANCER

Agora devemos criar um grupo de autoscaling, para fazer a escalabilidade da nossa aplicação, esse grupo funcionará juntamente com um load balancer, que irá balancear a carga entre as instâncias do autoscaling.

Na aba “**Grupos do autoscaling**” selecionamos a opção “**Criar grupo do autoscaling**”, então criamos um nome para o grupo e selecionamos o template criado no capítulo anterior. No menu seguinte selecionamos a VPC criada anteriormente e duas subnets públicas. Então a seguir devemos criar uma configuração de balanceador de carga, seguindo as imagens a seguir.

Balanceamento de carga [Info](#)

Use as opções abaixo para anexar seu grupo do Auto Scaling a um balanceador de carga existente ou a um novo balanceador de carga definido por você.

☐ Nenhum balanceador de carga
O tráfego para seu grupo do Auto Scaling não será liderado por um balanceador de carga.

☐ Anexar a um balanceador de carga existente
Escolha entre seus balanceadores de carga existentes.

☒ Anexar a um novo balanceador de carga
Crie rapidamente um balanceador de carga básico para anexar ao seu grupo do Auto Scaling.

Anexar a um novo balanceador de carga

Defina um novo balanceador de carga para criar a fim de anexar a esse grupo do Auto Scaling.

Tipo de balanceador de carga

Escolha um dos tipos de balanceador de carga oferecidos abaixo. Não é possível alterar a seleção de tipo depois que o balanceador de carga for criado. Se você precisar de um tipo de balanceador de carga diferente dos oferecidos aqui, [acesse o console do Load Balancing](#). [↗](#)

☒ Application Load Balancer
HTTP, HTTPS

☐ Network Load Balancer
TCP, UDP, TLS

Nome do balanceador de carga

O nome não pode ser alterado após a criação do balanceador de carga.

wordpress

Esquema do balanceador de carga

O esquema não pode ser alterado após a criação do balanceador de carga.

☐ Internal

☒ Internet-facing

VPC

vpc-0ab3a8c103d4e3843 [🔗](#)

aws-controltower-VPC

Zonas de disponibilidade e sub-redes

Você deve selecionar uma só sub-rede para cada zona de disponibilidade habilitada. Somente sub-redes públicas estão disponíveis para seleção visando viabilizar a compatibilidade com a resolução de DNS.

<input checked="" type="checkbox"/> us-east-1b	subnet-0d54d4895768c3be6 ▼
<input checked="" type="checkbox"/> us-east-1d	subnet-0b38330dac68a133b ▼
<input type="checkbox"/> us-east-1c	Selecionar uma sub-rede ▼
<input type="checkbox"/> us-east-1e	Selecionar uma sub-rede ▼
<input type="checkbox"/> us-east-1f	Selecionar uma sub-rede ▼
<input type="checkbox"/> us-east-1a	Selecionar uma sub-rede ▼

Listeners e roteamento

Se precisar de listeners seguros ou de vários listeners, você poderá configurá-los no [Console do Load Balancing](#) [🔗](#) depois que o balanceador de carga for criado.

Protocolo	Porta	Roteamento padrão (encaminhar para)
HTTP	80	Criar um grupo de destino ▼
Nome do novo grupo de destinos		
Um grupo de destino de instância com configurações padrão será criado.		
wordpress		

Devemos ter cuidado com as opções de “**Esquema do balanceador de carga**”, selecionando “**internet-facing**”, e subnets, que deverão ser selecionadas apenas subnets públicas.

1.6 – INSALANDO O WORDPRESS

Para configurar o wordpress devemos acessar o endereço DNS do Load balancer no navegador. Ao acessar a tela de instalação do wordpress deverá aparecer, então selecionamos o idioma, usuário, senha e inserimos um email. Após isso o wordpress será instalado.

Agora devemos realizar uma configuração dentro do banco de dados RDS para que tudo funcione corretamente.

1.7 – CONFIGURANDO O DNS DO LOAD BALANCER DENTRO DO BANCO DE DADOS RDS

Para que consigamos acessar os serviços publicados por nosso load balancer, devemos configurar seu dns dentro do banco de dados RDS, para que seja possível acessar e gerenciar tudo.

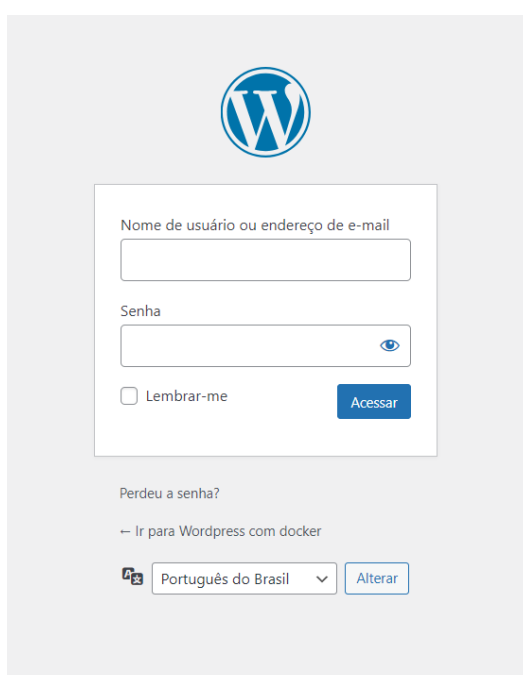
Conectamos via SSH em uma das instâncias criadas pelo autoscaling, então instalamos o mysql. Após a instalação conectamos ao nosso banco de dados RDS, usando o comando **"mysql -u admin -p -h wp-rds.cexp47lsdaww.us-east-1.rds.amazonaws.com -P 3306 -D wordpress"**, o comando contém o usuário do banco de dados, seu endpoint, sua porta e o nome do banco de dados.

Então executamos o comando **"SELECT * FROM wpoptions WHERE option_name = 'siteurl' OR option_name = 'home';"** Este comando irá selecionar todas as linhas da tabela wpoptions onde o nome da opção é igual a siteurl ou home. Ali estão contidas as informações de URL do site.

Após isso o comando **"UPDATE wpoptions SET option_value = 'http://wp-rds.cexp47lsdaww.us-east-1.rds.amazonaws.com' WHERE option_name = 'siteurl' OR option_name = 'home';"** Este comando irá alterar as linhas contidas na tabela "wpoptions", substituindo os valores antigos pelo dns do seu Load Balancer.

1.8 – RELIZANDO TESTES NO AMBIENTE

Após toda a configuração, ao acessar o endereço DNS do load balancer a tela inicial de demonstração do wordpress deverá aparecer. Ao acessar o endereço, seguido de **"/wp-login.php"**, devemos acessar a tela de login do wordpress.



Após fazer o login a tela de administração irá aparecer. Então tudo está feito, o ambiente está funcionando corretamente e pronto para receber o site desejado no wordpress.

