



Documentação de atividade AWS Docker

Bolsista : Ricardo Machado Nunes

Team : PB – FW – A – RG – SB – HA

Studio : Cloud & DevSecOps

DC : DCV Ijuí

Outubro 2023

1. Configurações iniciais

Para o desenvolvimento da atividade, configuraremos o ambiente da AWS para que nossa estrutura esteja alocada de forma organizada e funcional através do *checklist* abaixo.

- ✓ Criação de um *key pair*;
- ✓ Criação de uma *subnet* / *VPC*;
- ✓ Criação de um *Security Group*;
- ✓ Viabilizar o acesso via *SSH*;
 1. Criação de um *user AMI*;
 2. Criação de uma *Acess key*; e
- ✓ Configurar o acesso via *SSH*.

2. Configurações de apoio

Aqui serão configuradas as estruturas de apoio a aplicação, como o *auto scaling*, *load balancer*, *Amazon RDS* e *Amazon EFS*.

2.1 Auto Scaling

Para o funcionamento do *auto scaling* é necessária a criação de um modelo de execução, utilizaremos a instância inicial para determinar como padrão para a criação deste modelo.

Também foi criada uma *subnet* com nome de *subnet replacement*, na qual serão alocados os modelos criados pelo *auto scaling*.

Foi determinada a limitação de 3 instâncias por AZ, nas políticas de escalonamento do *auto scaling*.

wordpressAS

Detalhes

Atividade

Escalabilidade automática

Gerenciamento de instâncias

Monitoramento

Atualização da instância

Detalhes do grupo

Editar

Nome do grupo do Auto Scaling wordpressAS	Capacidade desejada 1	Status -	Nome de recurso da Amazon (ARN)  <code>arn:aws:autoscaling:us-east-1:044175277726:autoScalingGroup:1b666348-3d55-4d6b-9a66-83843f146026:autoScalingGroupName/wordpressAS</code>
Data de criação Tue Oct 24 2023 17:45:29 GMT-0300 (Horário Padrão de Brasília)	Capacidade mínima 1	Capacidade máxima 3	

Teste de integridade do *auto scaling*.

Histórico de atividades (22)			
<input type="text" value="Filtrar histórico de atividades"/>			
Status	Descrição	Causa	Horário inicial
✔ Êxito	Launching a new EC2 instance: i-089fa13fd7969c325	At 2023-10-24T21:46:14Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2023 October 24, 06:46:17 PM -03:00
✘ Com falha	Launching a new EC2 instance. Status Reason: There aren't sufficient free Ipv4 addresses or prefixes. Launching EC2 instance failed.	At 2023-10-24T21:36:48Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.	2023 October 24, 06:36:49 PM -03:00

2.2 Load Balancer

O *Load Balancer* tem o papel de garantir a distribuição de carga entre as instâncias incluídas em seu alcance, além de garantir a disponibilidade em caso de alguma falha ou indisponibilidade.

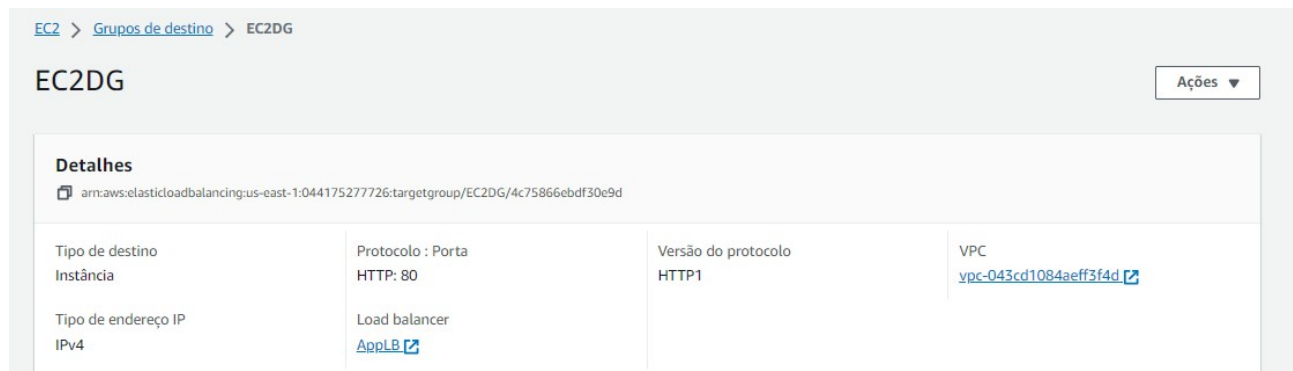
EC2 > Load balancers > wordpress			
wordpress			
▼ Detalhes			
Tipo de balanceador de carga Network	Status ✔ Ativo	VPC vpc-043cd1084aeff3f4d	Tipo de endereço IP IPv4
Esquema Internet-facing	Zona hospedada Z26RNL4JYFTOTI	Zonas de disponibilidade subnet-081b02744657d9460 us-east-1d (use1-az1) subnet-0921cc1bd4935e771 us-east-1b (use1-az4) subnet-0ac25fc78783f76e8 us-east-1a (use1-az2)	Data de criação 26 de outubro de 2023, 14:34 (UTC-03:00)
ARN do load balancer arn:aws:elasticloadbalancing:us-east-1:044175277726:loadbalancer/net/wordpress/b7f32ae9cb749ae6		Nome do DNS Informações wordpress-b7f32ae9cb749ae6.elb.us-east-1.amazonaws.com (Registro A)	

Na imagem acima, podemos observar o *load balancer* em uma configuração de *internet-facing*, ou seja, como uma porta, com sua atuação ocorrendo sob tráfego de formato *IPv4* e distribuindo entre as AZ 1d, 1b e 1ª.

2.2.1 Target Group

O *load balancer* não trabalha sozinho na função de distribuir o tráfego de acesso, ele funciona através de regras de roteamento, neste ponto entra o target group.

Ele direciona o tráfego entre as AZ que estão previamente disponíveis no *load balancer* através de uma regra (ou mais) de encaminhamentos.



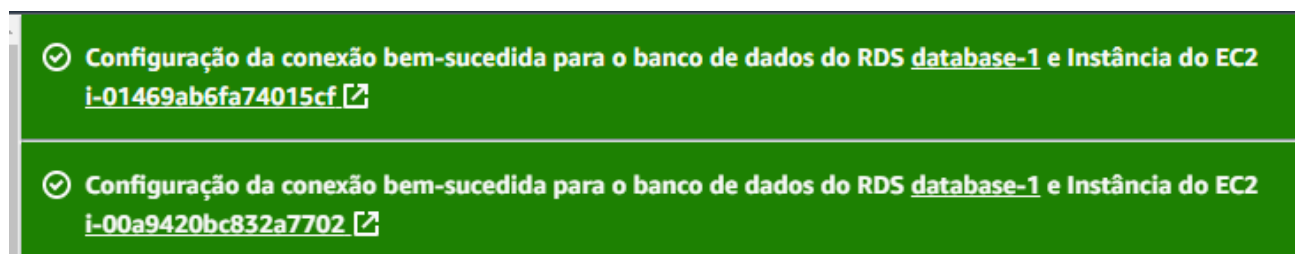
The screenshot shows the 'EC2DG' (EC2 Target Group) page in the AWS Management Console. The breadcrumb navigation is 'EC2 > Grupos de destino > EC2DG'. The page title is 'EC2DG'. There is an 'Ações' (Actions) button with a dropdown arrow. The 'Detalhes' (Details) section shows the ARN: `arn:aws:elasticloadbalancing:us-east-1:044175277726:targetgroup/EC2DG/4c75866ebdf30e9d`. Below this, there are four columns of details:

Tipo de destino	Protocolo : Porta	Versão do protocolo	VPC
Instância	HTTP: 80	HTTP1	vpc-043cd1084aeff3f4d
Tipo de endereço IP	Load balancer		
IPv4	AppLB		

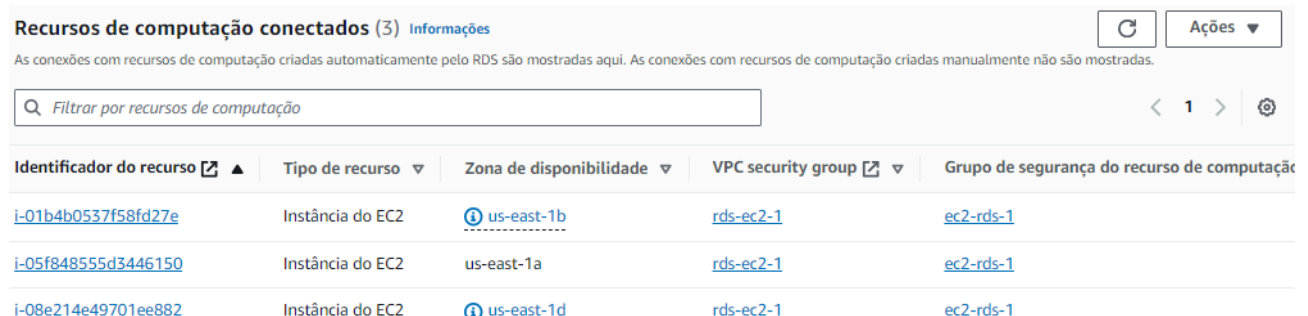
Aqui temos um *target group* atuando sob a porta 80 para conexões do tipo *IPV4* dentro de uma determinada *virtual private cloud (VPC)* com suas configurações atreladas a um *load balancer* na forma de regra de roteamento.

2.3 Amazon RDS

Como gestora da *database* de nossa aplicação, será utilizada a estrutura do *Amazon RDS*, que gerencia, automatiza e provê a escalabilidade necessária para um banco de dados relacional para aplicações em nuvem.



The screenshot shows two green notification banners with white text and checkmark icons. The first banner says: 'Configuração da conexão bem-sucedida para o banco de dados do RDS database-1 e Instância do EC2 i-01469ab6fa74015cf'. The second banner says: 'Configuração da conexão bem-sucedida para o banco de dados do RDS database-1 e Instância do EC2 i-00a9420bc832a7702'. Both banners have a small external link icon at the end of the instance ID.



The screenshot shows the 'Recursos de computação conectados (3)' (Connected compute resources (3)) page in the AWS Management Console. There is a search bar with the placeholder 'Filtrar por recursos de computação'. Below the search bar is a table with the following columns: 'Identificador do recurso', 'Tipo de recurso', 'Zona de disponibilidade', 'VPC security group', and 'Grupo de segurança do recurso de computação'.

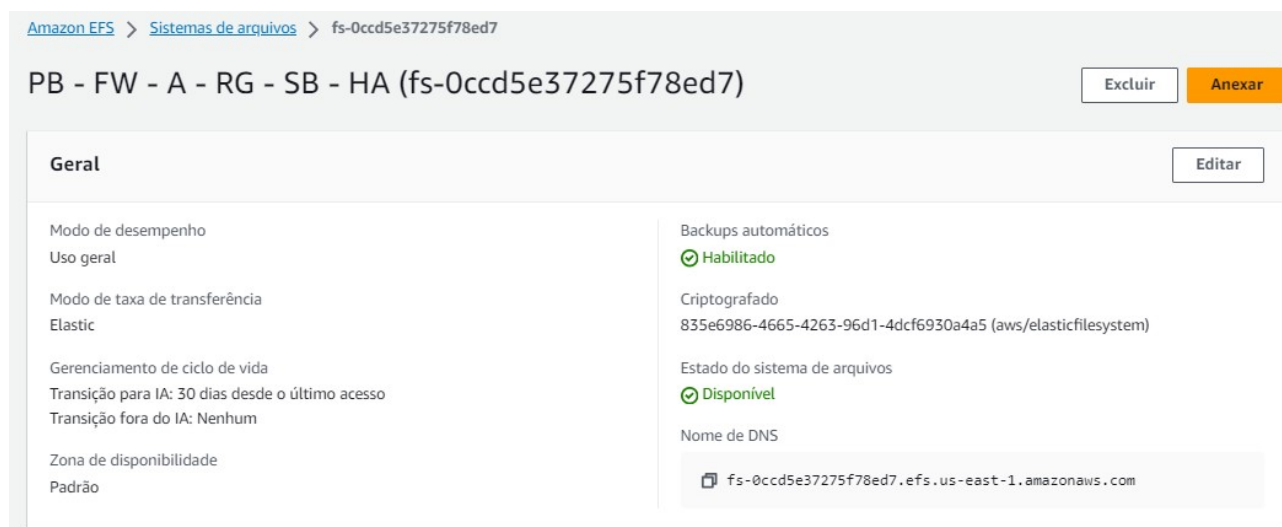
Identificador do recurso	Tipo de recurso	Zona de disponibilidade	VPC security group	Grupo de segurança do recurso de computação
i-01b4b0537f58fd27e	Instância do EC2	us-east-1b	rds-ec2-1	ec2-rds-1
i-05f848555d3446150	Instância do EC2	us-east-1a	rds-ec2-1	ec2-rds-1
i-08e214e49701ee882	Instância do EC2	us-east-1d	rds-ec2-1	ec2-rds-1

2.4 Amazon EFS

O *Amazon EFS* é uma ferramenta que opera no protocolo NFS de compartilhamento de banco de dados por diversas estruturas.

A ferramenta disponibiliza um comando de montagem automatizado (Anexar) possibilitando ser realizado tanto pelo *DNS* do *EFS* como por endereço de IP.


Visando a estabilidade da escalabilidade da aplicação, foi utilizada a montagem via *DNS* e anexado ao *script* de inicialização de nossas instâncias EC2.



Amazon EFS > Sistemas de arquivos > fs-0ccd5e37275f78ed7

PB - FW - A - RG - SB - HA (fs-0ccd5e37275f78ed7) Excluir Anexar

Geral Editar

Modo de desempenho	Backups automáticos
Uso geral	✔ Habilitado
Modo de taxa de transferência	Criptografado
Elastic	835e6986-4665-4263-96d1-4dcf6930a4a5 (aws/elasticfilesystem)
Gerenciamento de ciclo de vida	Estado do sistema de arquivos
Transição para IA: 30 dias desde o último acesso	✔ Disponível
Transição fora do IA: Nenhum	Nome de DNS
Zona de disponibilidade	 fs-0ccd5e37275f78ed7.efs.us-east-1.amazonaws.com
Padrão	

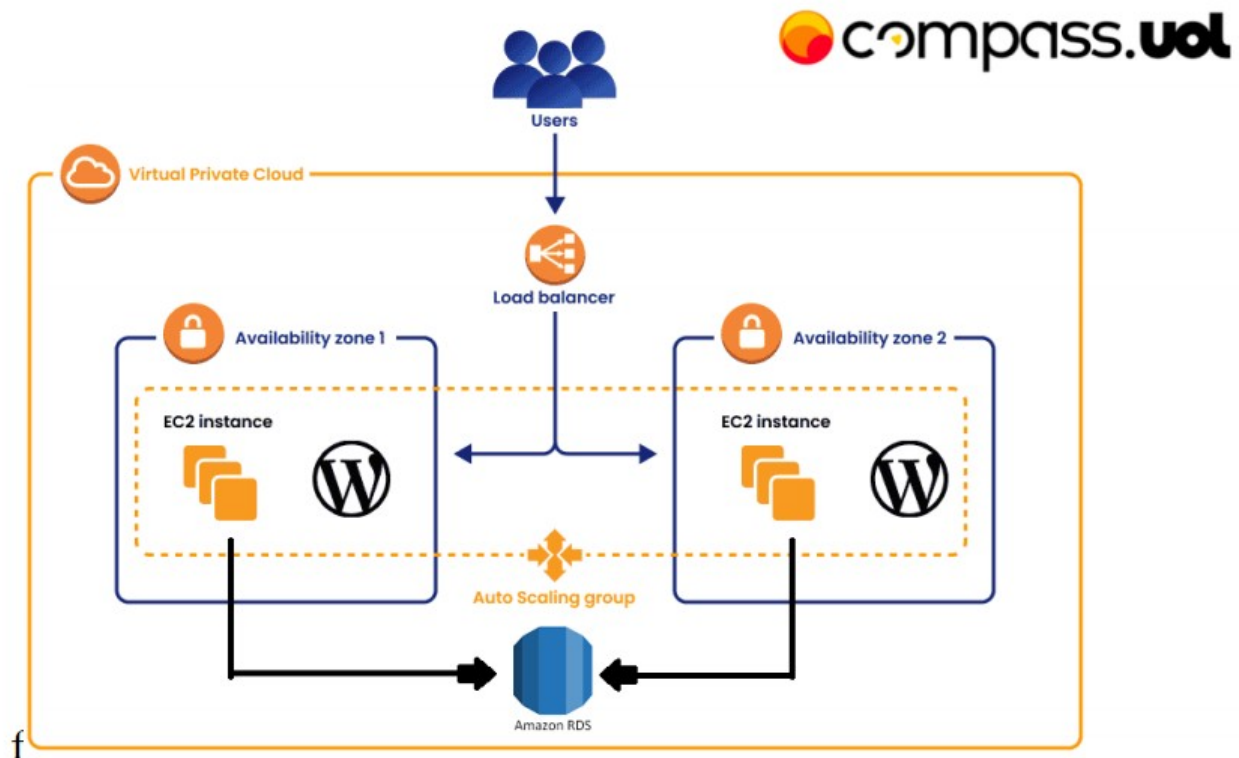
Acima podemos observar a disponibilidade do sistema de *EFS*.

```
[ec2-user@ip-10-0-0-94 ~]$ df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  954M        0   954M   0% /dev
tmpfs                     963M        0   963M   0% /dev/shm
tmpfs                     963M    648K   962M   1% /run
tmpfs                     963M        0   963M   0% /sys/fs/cgroup
/dev/nvme0n1p1            16G    2.9G    14G  18% /
tmpfs                     193M        0   193M   0% /run/user/0
tmpfs                     193M        0   193M   0% /run/user/1000
fs-0ccd5e37275f78ed7.efs.us-east-1.amazonaws.com:/ 8.0E        0   8.0E   0% /efs
[ec2-user@ip-10-0-0-94 ~]$
```

Sistema montado de forma padrão em uma instância EC2 através do *script* de inicialização (user_data.sh).

3. Deploy

O ambiente proposto deve ser organizado da seguinte forma:



Os seguintes pontos devem ser observados:

- ➔ Acesso via *web* através do *Load Balancer*;
- ➔ Ao menos duas instâncias em um mesmo *auto scaling group* com o propósito de manter a disponibilidade da aplicação permanentemente.
- ➔ Deve haver uma *database* comum gerida pelo *Amazon RDS* na qual o sistema de arquivos compartilhados pelo *Amazon EFS* na estrutura da *AWS*; e
- ➔ *Deploy* da aplicação *Wordpress* utilizando desta *database* e compartilhamento de arquivos previamente configurado.

Conforme previamente configurado no item 2.2 o acesso via *load balancer* está assegurado, bem como a configuração do *auto scaling group* no item 2.1 foi realizado com sucesso, então vamos aos demais pontos pendentes.

3.1 EC2

Para a atividade foi criada uma instância EC2 com as seguintes configurações:

Dados do usuário - *optional* [Informações](#)
Faça upload de um arquivo com seus dados de usuário ou o insira no campo.

```
#!/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 mkdir /wordpress
cd /wordpress
sudo mkdir efs
```

Imagem do software (AMI)
Amazon Linux 2 Kernel 5.10 AMI...[Ler mais](#)
ami-003ed94f11f7ecc7e

Tipo de servidor virtual (tipo de instância)
t3.small

Firewall (grupo de segurança)
wordpressSG

Armazenamento (volumes)
1 volume(s) - 16 GiB

Foi utilizado o *script* abaixo para a automação da implantação da instância (*user_data.sh*). Foi realizada a **atualização do SO**; instalação do *docker*, sua inicialização e fixação nos aplicativos iniciais da instância; **download do docker compose** e **permissão de execução root**; criação do diretório *wordpress* e a montagem do sistema *NFS* dentro do serviço *EFS*; criação da imagem *wordpress* em um contêiner no *docker compose*, com suas configurações de tráfego e armazenamento em um banco de dados virtual do *RDS*; retorno ao diretório inicial e por fim **inicializar os serviços docker compose** previamente configurados em *docker-compose.yml*

```
#!/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 mkdir /wordpress
cd /wordpress
sudo mkdir efs
sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport fs-
```

```
0ccd5e37275f78ed7.efs.us-east-1.amazonaws.com:/ /efs
```

```
echo "version: '3.7'
```

```
services:
```

```
  wordpress:
```

```
    image: wordpress
```

```
    volumes:
```

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

```
    ports:
```

```
      - "80:80"
```

```
    restart: always
```

```
    environment:
```

```
      WORDPRESS_DB_HOST: wordpressdb.czfezhkuxjky.us-east-1.rds.amazonaws.com
```

```
      WORDPRESS_DB_USER: admin2023
```

```
      WORDPRESS_DB_PASSWORD: admin2023
```

```
      WORDPRESS_DB_NAME: wordpress
```

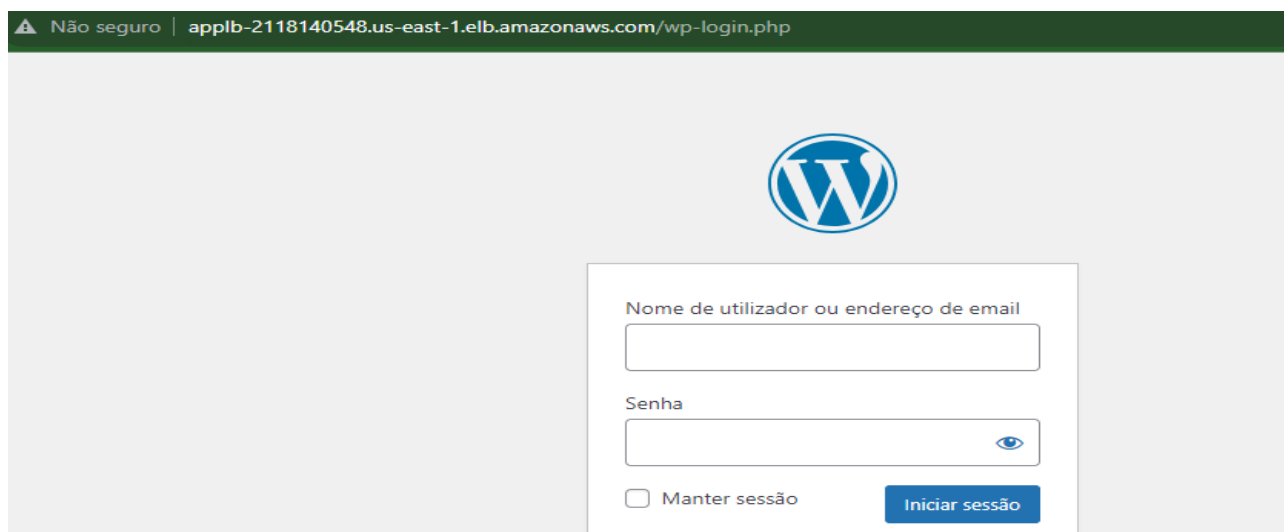
```
      WORDPRESS_TABLE_PREFIX: wp" | sudo tee /docker-compose.yml
```

```
cd /
```

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

4. Conclusão

Ao término da atividade, deve-se acessar via *web* a página inicial do *wordpress* com sucesso através do endereço que direciona ao *load balancer* configurado para distribuir os acessos nas instâncias multizoneadas com a aplicação disponibilizada pelo *Docker Composer*.



Configuração finalizada.