

Resumo Docker Swarm

Lucas Sepeda

Março de 2023

1 Configuração na AWS

1.1 Configurando VPC

1. Abra o menu da VPC na AWS e clique em criar VPC
2. As suas configurações devem ser as seguintes

Criar VPC [Informações](#)

Uma VPC é uma parte isolada da Nuvem AWS preenchida por objetos da AWS, como instâncias do Amazon EC2.

Configurações da VPC

Recursos a serem criados [Informações](#)
Crie apenas o recurso da VPC ou a VPC e outros recursos de rede.

☒ Somente VPC ☐ VPC e muito mais

Tag de nome - opcional
Cria uma tag com uma chave de "Nome" e um valor que você especifica.

swarm-vpc

Bloco CIDR IPv4 [Informações](#)
☒ Entrada manual de CIDR IPv4
☐ Bloco CIDR IPv4 alocado por IPAM

CIDR IPv4
10.0.0.0/24

Bloco CIDR IPv6 [Informações](#)
☒ Nenhum bloco CIDR IPv6
☐ Bloco CIDR IPv6 alocado por IPAM
☐ Bloco CIDR IPv6 fornecido pela Amazon
☐ CIDR IPv6 de minha propriedade

Localização [Informações](#)
Padrão

1.2 Configurando subnet na VPC

1. No menu de VPC selecione Sub-redes
2. Clique em criar sub-rede
3. Selecione a VPC que foi criada na passo anterior no campo ID da VPC
4. Suas configurações de sub-rede devem ser as seguintes

Configurações de sub-rede

Especifique os blocos CIDR e a zona de disponibilidade para a sub-rede.

Sub-rede 1 de 1

Nome da sub-rede
Crie uma tag com a chave 'Nome' e um valor que você especificar.

swarm-vpc-subnet

O nome pode ter até 256 caracteres.

Zona de disponibilidade [Informações](#)
Escolha a zona na qual sua sub-rede residirá ou deixe que a Amazon escolha uma para você.

América do Sul (São Paulo) / sa-east-1a

Bloco CIDR IPv4 [Informações](#)

10.0.0.0/24

▼ **Tags - opcional**

Chave	Valor - opcional	
Name	swarm-vpc-subnet	Remove

[Adicionar nova tag](#)

Você pode adicionar mais 49 tags.

[Remover](#)

[Adicionar nova sub-rede](#)

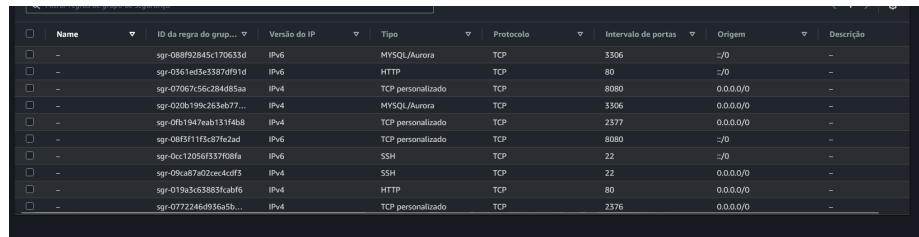
Cancelar [Criar sub-rede](#)

1.3 Configurando Internet Gateway e Routing Tables

1. No menu VPC, selecione Gateways da Internet e clique em criar um novo gateway
2. Aqui é só necessário criar um nome, como por exemplo "swarm-gateway"
3. Após criado, clique com o botão direito do mouse em cima do gateway criado e depois em "Associar VPC". Selecione a VPC do swarm
4. No menu tabela de rotas, procure pela tabela de rotas associada a nossa VPC do swarm. É possível identificar isso pela propriedade VPC.
5. Clique em editar rota, e adicione a seguinte rota. Destino: 10.0.0.0/24 e selecione como alvo um gateway da internet e selecione o gateway criado

1.4 Configurando security groups do EC2

1. Acesse o menu do EC2 na AWS
2. Clique em Security Groups na parte de Redes e Segurança
3. Crie um novo grupo
4. Na parte de detalhes básicos crie um nome ("swarm-sg") e tenha certeza de selecionar a VPC que criamos.
5. Agora, configure as regras de entrada. As regras são as seguintes



<input type="checkbox"/>	Name	ID da regra do grup...	Versão do IP	Tipo	Protocolo	Intervalo de portas	Origem	Descrição
<input type="checkbox"/>	-	sg-088f92845c170633d	IPv6	MySQL/Aurora	TCP	3306	:/0	-
<input type="checkbox"/>	-	sg-0361e3a3387d91d	IPv6	HTTP	TCP	80	:/0	-
<input type="checkbox"/>	-	sg-07037c56294d85ea	IPv4	TCP personalizado	TCP	8080	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-020b199c263a077...	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-0fb1947eab1314ab8	IPv4	TCP personalizado	TCP	2377	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-08f3f11f3cd71e2ad	IPv6	TCP personalizado	TCP	8080	:/0	-
<input type="checkbox"/>	-	sg-0xc12056f337f08fa	IPv6	SSH	TCP	22	:/0	-
<input type="checkbox"/>	-	sg-09ca87a02ce4c0f3	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-019a3c63883fcbf6	IPv4	HTTP	TCP	80	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-0772246d936a5b...	IPv4	TCP personalizado	TCP	2376	0.0.0.0/0	-

Com isso concluímos nossa configuração na AWS

2 Criando um Swarm

Com as configurações da AWS feitas, podemos criar máquinas a partir do `docker-machine`.

2.1 Criando o nosso manager

Para rodar vamos precisar de algumas variáveis.

1. `--amazonec2-vpc-id` - Procure o ID da VPC na AWS

2. `--amazon-subnet-id` - Procure o ID da Subnet na AWS
3. `--amazonec2-region` - O valor utilizado aqui deve ser `sa-east-1`
4. `--amazonec2-zone` - O valor utilizado aqui deve ser `a`
5. `--amazonec2-instance-type` - O tipo da instância. O valor utilizado aqui é `t2.micro`
6. `--amazonec2-security-group` - Nome do grupo de segurança criado na AWS
7. `--amazonec2-ami` - Qual imagem de sistema operacional será instalada na máquina. Aqui o nosso valor é `ami-019e9884d49f70483`
8. Por final passamos o nome que a máquina deverá ser criada

O comando para criar o manager fica, por exemplo:

```
docker-machine create -d amazonec2 --amazonec2-vpc-id vpc-0fb3cf003a3ab3d15 \
--amazonec2-subnet-id subnet-0208792f3e2adfb24 \
--amazonec2-region sa-east-1 \
--amazonec2-zone a --amazonec2-instance-type t2.micro \
--amazonec2-security-group swarm-cluster-sg \
--amazonec2-ami ami-019e9884d49f70483 manager01
```

Isso deve iniciar nossa máquina no EC2. Com máquina criada podemos executar `docker-machine ssh manager01`. Isso fará nosso terminal se conectar à nossa máquina no EC2.

Dentro da nossa máquina no EC2 podemos executar o comando `docker swarm init` para iniciar nosso swarm. Isso nos dará as credenciais para entrarmos como worker a partir de outra máquina.

Agora desconecte o terminal e retorne a sua máquina local com o comando `exit`.

Repita os procedimentos acima para criar workers 1 e 2. E com o token de acesso inicie essas máquinas como Workers.

Na nossa máquina local, podemos executar `docker-machine ls` para listar as máquinas disponíveis. Caso queira eliminar uma máquina rode `docker-machine rm [nomeDaMaquina]`

Lembre que enquanto não desligarmos as máquinas, as instâncias do EC2 continuaram rodando e possíveis cobranças podem ser aplicadas.