

Configurando o cluster na AWS

Mário Olímpio de Menezes

Criando as máquinas

Uma etapa fundamental ao criar as máquinas é utilizar um **security group** único para todas as máquinas. Isso é necessário para as configurações posteriores.

Depois de seleccionar a AML e o tipo de instância, chegamos nesta tela. **Selecione Edit security groups**

Step 7: Review Instance Launch

▼ Instance Type

Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups

Edit security groups

Security group name

launch-wizard-1

Description

launch-wizard-1 created 2020-04-03T23:55:41.910-03:00

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
This security group has no rules				

Dê um nome especial para o security group, para identificar o cluster

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a **new** security group
☐ Select an **existing** security group

Security group name:
Description:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH ▼	TCP	22	Custom ▼ 0.0.0.0/0	e.g. SSH for Admin Desktop
<div>Add Rule</div>				

Quando estiver criando as outras máquinas do cluster, reutilize-o como mostrado na figura a seguir:

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☐ Create a new security group
☒ Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-09a4d8448968bc987	balanceado_cargas	launch-wizard-9 created 2020-04-01T20:32:57.889-03:00	Copy to new
<input checked="" type="checkbox"/> sg-0b9e23e25335bb816	cluster	launch-wizard-9 created 2020-04-03T21:26:07.440-03:00	Copy to new
<input type="checkbox"/> sg-a4d748c7	default	default VPC security group	Copy to new
<input type="checkbox"/> sg-0ab0c532e83c2a3e0	launch-wizard-2	launch-wizard-2 created 2019-08-30T18:53:34.697-03:00	Copy to new
<input type="checkbox"/> sg-0bd5e08b337886690	launch-wizard-3	launch-wizard-3 created 2019-08-30T21:57:30.114-03:00	Copy to new

Abrindo portas TCP dentro do **security group** para o MPI

Para o MPI rodar, precisamos abrir as portas TCP livremente entre as máquinas do cluster. Como todas estão utilizando o mesmo **security group**, fazemos isso adicionando uma regra como abaixo:

Primeiro, vamos na opção **NETWORK & SECURITY**, e selecionamos **Security Groups**; depois, selecionamos o **security group** correto

The screenshot displays the AWS Management Console interface for the 'Security Groups' page. On the left, the navigation sidebar is visible, with 'NETWORK & SECURITY' expanded and 'Security Groups' selected. The main content area shows a list of security groups. The table has columns for checkboxes, Security group ID, Name, and VPC ID. The 'cluster' security group (sg-0b9e23e25335bb816) is highlighted in blue.

	Security group ID	Name	VPC ID
<input type="checkbox"/>	sg-09a4d8448968bc987	balanceado_cargas	vpc-c0
<input checked="" type="checkbox"/>	sg-0b9e23e25335bb816	cluster	vpc-c0
<input type="checkbox"/>	sg-0ab0c532e83c2a3e0	launch-wizard-2	vpc-c0
<input type="checkbox"/>	sg-0b5739fa4a...	launch-wizard-5	vpc-c0
<input type="checkbox"/>	sg-0bd5e08b337886690	launch-wizard-3	vpc-c0

E vamos editar as regras de entrada: **Edit Inbound Rules**:

Inbound rules					Edit inbound rules
Type	Protocol	Port range	Source	Description - optional	
SSH	TCP	22	0.0.0.0/0	-	

Agora, vamos Adicionar uma nova Regra **All TCP**

Inbo

Custom UDP

Custom ICMP - IPv4

Custom ICMP - IPv6

Custom Protocol

All TCP

All UDP

All ICMP - IPv4

Custom TCP ▲

the incoming traffic that's allowed to reach the instance.

Info

Protocol

Port range

Info

Source

Info

TCP

22

Cu... ▼

Q

0.0.0.0/0

TCP

0

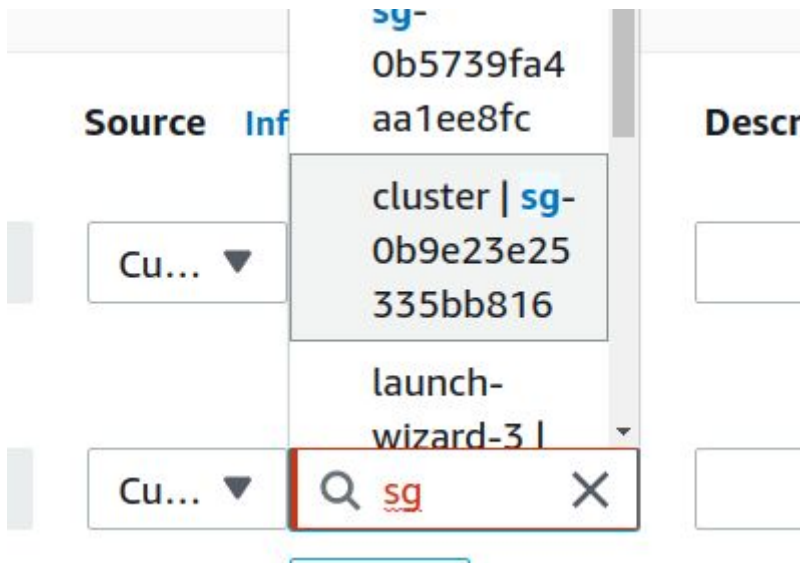
Cu... ▼

Q

Add rule

Depois de selecionar **All TCP** é preciso informar de onde podem vir as conexões (**Source**).

Neste campo, comece a digitar “**sg**” e vão aparecer as opções dos **security groups** para você poder selecionar:



Selecione o **security group** do seu cluster e salve as modificações.

Copiando a chave .pem para as máquinas

Quando uma máquina é criada, a AWS envia a chave **.pem** utilizada para conectar. Esta chave tem que ser enviada para cada máquina do cluster (utilize sempre a mesma chave).

Você deve utilizar o comando **scp** ou o **pscp** para fazer a cópia da chave: (o comando abaixo é em uma única linha)

```
$ scp -i ~/.ssh/mariomack.pem ~/.ssh/mariomack.pem  
ubuntu@ec2-18-223-28-95.us-east-2.compute.amazonaws.com:~/.ssh/
```

Agora precisamos adicionar nossa chave privada ao nosso agente de autenticação **ssh-agent**. **TEM QUE FAZER ISSO EM TODAS AS MÁQUINAS**

```
$ eval `ssh-agent`  
$ ssh-add mariomack.pem
```

Colocar todas as máquinas no `/etc/hosts` utilizando o IP privado.

Edite o arquivo `/etc/hosts` adicionando estas linhas logo abaixo da linha do **localhost** (1ª linha)

```
$ sudo nano /etc/hosts
```

```
172.31.35.142 master
172.31.31.44 slave1
172.31.22.225 slave2
```

(Para finalizar, digite CRTL+O para salvar o arquivo e depois CRTL+X para sair)

Depois, da máquina master executar um comando remoto com o **ssh** em cada slave para tornar a máquina conhecida (pelo IP privado):

```
$ ssh ubuntu@slave1 hostname
The authenticity of host 'slave1 (172.31.22.225)' can't be established.
ECDSA key fingerprint is
SHA256:9YrhJaRMtgTBm/leoDCM8qhdsnnnrPPJPcd81nvoZ9o.
Are you sure you want to continue connecting (yes/no)? yes
```

```
$ ssh ubuntu@slave2 hostname
The authenticity of host 'slave2 (172.31.24.225)' can't be established.
ECDSA key fingerprint is
SHA256:9YrhJaRMtgTBm/leoDCM8qhdsnnnrPPJPcd81nvoZ9o.
Are you sure you want to continue connecting (yes/no)? yes
```

Na segunda vez que executar o comando acima, ele não deverá mais mostrar esta informação de autenticidade da máquina, etc. Vai mandar o resultado direto da execução do comando **hostname**. Você deve rodar o comando acima para todas as **slaves** do cluster.