

01

Escalando nossas instâncias do EC2

Transcrição

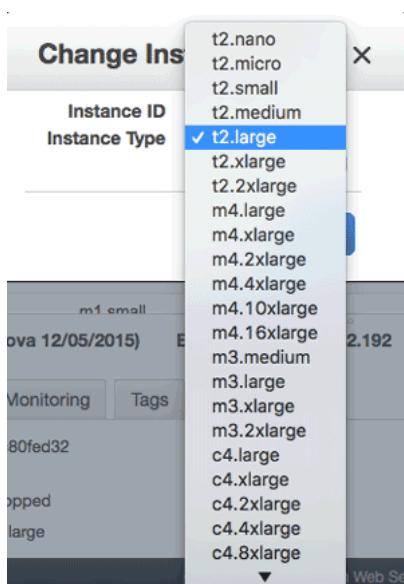
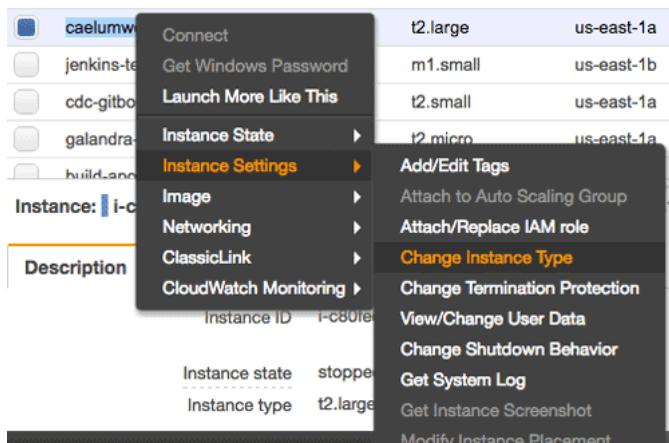
No capítulo anterior terminamos as configurações no EC2 para usar o AWS RDS. O nosso banco de dados está funcionando agora em uma máquina separada só para ele, num ambiente escalável e bem configurado pela Amazon. Como vimos, isso é um exemplo do SaaS (Software as a Service), pois não estamos mais responsáveis pela instalação do banco, apenas configuramos pela interface web.

Mais ainda tem uma questão referente a escalabilidade: Será que nosso `catalogo-teste` vai sempre aguentar qualquer número de requisições? Quando a demanda da nossa aplicação cresce será que essa máquina consegue ainda atender bem os nossos clientes?

Falando sobre escalabilidade, sempre há duas formas de escalar: vertical e horizontal

Escalabilidade vertical ou horizontal?

Vertical significa melhorar a máquina em si, por exemplo a CPU, Memória RAM ou HD. Isso é **limitado** e também pode ficar muito **caro**. Escalabilidade vertical é sempre limitado mas também é o mais simples, pois basta melhorar o Hardware da máquina. Isso é feito no Amazon através do *instance type*:



Para trocar o tipo da instância a máquina deve estar desligada e, novamente, quando gravamos o curso apenas o ipo `t2.micro` é gratuito.

Imagen com AMI

No nosso caso queremos continuar com a faixa gratuita da Amazon EC2 e por isso escalaremos a nossa aplicação horizontalmente. Isso significa que poderíamos criar uma segunda instância como já aprendemos. No entanto, não basta só criar a instância, também é preciso instalar todo o software nela, no nosso caso Apache Tomcat junto com a aplicação. Ideal seria criar uma nova instância a partir dessa instalação já feito. Isso é possível no Amazon AWS através das imagens que inclusive já usamos no curso. Essas imagens tem o nome do **Amazon Machine Instance (AMI)** e já existem vários disponíveis vimos elas no primeiro passo da criação da máquina:

A diferença é que criaremos a nossa própria imagem (ou template) baseado na instância que já configuramos!

Criando imagem

Para criar uma nova imagem basta selecionar a uma instância e clicar no `Image -> Create Image`

No formulário preenchemos o nome e descrição e basta confirmar para criar a imagem.

Create Image

Instance ID	i-084cf691fc5043989					
Image name	catalogo					
Image description	imagem do catalogo					
No reboot	<input type="checkbox"/>					
Instance Volumes						
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)
Root	/dev/sda1	snap-066a4d67938024361	8	General Purpose SSD (GP2)	100 / 3000	N/A
Add New Volume						
Total size of EBS Volumes: 8 GiB When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.						

A criação pode demorar alguns minutos e depois disso a nova imagem deve estar disponível no menu *IMAGES -> AMIs*. Com a nossa imagem disponível podemos iniciar uma nova instância a partir dela. Só vamos renomear para "catalogo-imagem":

Launch Actions	
Owned by me <input type="button" value="search : ami-8e681d98"/> Add filter	
Name	AMI Name
catalogo-imagem	catalogo

Nova instância

Agora vamos realmente criar uma nova instância a partir da nossa. No menu *INSTANCES -> Instances* vamos clicar no botão *Launch instance* e escolher a nossa imagem (menu *My AMIs*):

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start	Search my AMIs	Cancel and Exit
My AMIs	catalogo - ami-8e681d98	
AWS Marketplace	catalogo	Select
Community AMIs	Root device type: ebs Virtualization type: hvm Owner: 970399446805	64-bit

Como já vimos anteriormente, o tipo da instância deve ser novamente aquele que faz parte da faixa gratuita da Amazon (*Free Tier*), o `t2.micro`. Todos os outros passos que já aprendemos. Importante é que você escolhe o mesmo `keypair` que já definimos antes (claro, pode mudar tbm). Dessa forma podemos usar o mesmo arquivo `catalogo.pem` para ambos as máquinas:

Select an existing key pair or create a new key pair

A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

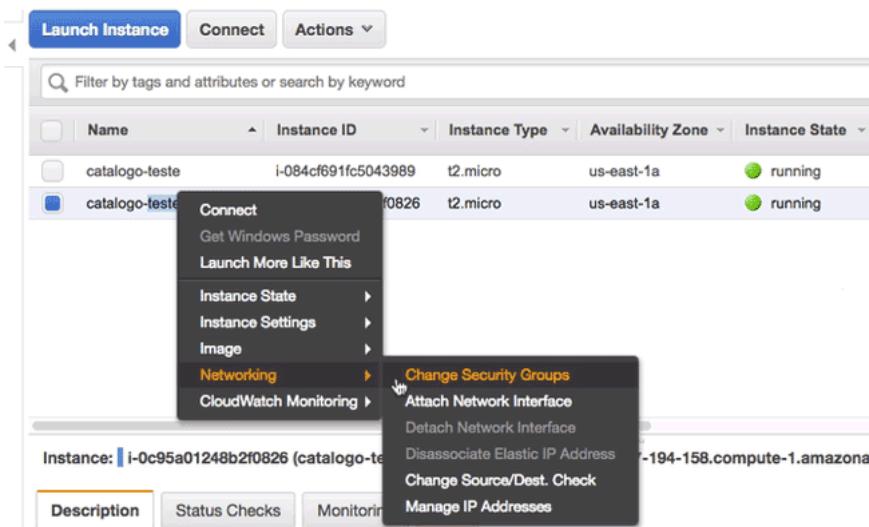
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair
Select a key pair
<input checked="" type="checkbox"/> catalogo
<input type="checkbox"/> I acknowledge that I have access to the selected private key file (catalogo.pem), and that without this file, I won't be able to log into my instance.
Cancel Launch Instances

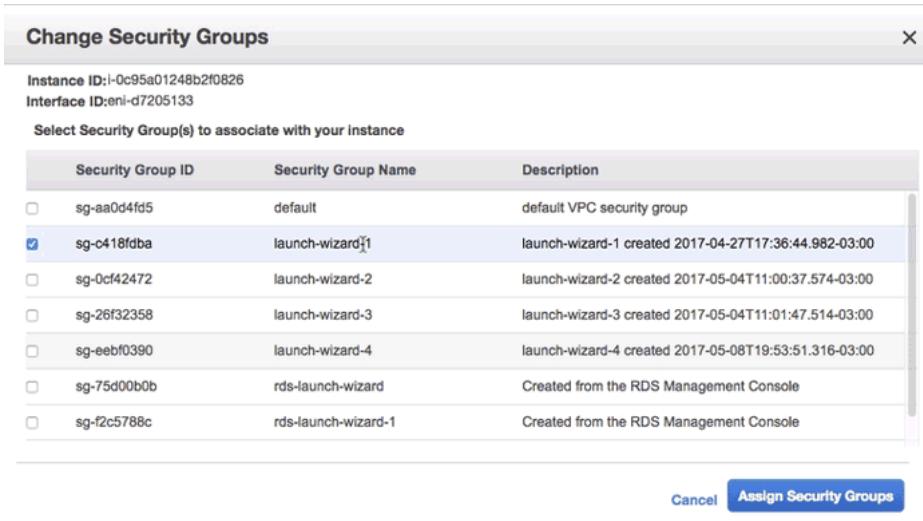
Ajustando Security Group

A instância já está configurado e rodando mas ainda não temos acesso na parte `8080` (e o servidor Tomcat usa por padrão essa porta). A configuração das portas não é da imagem e sim um serviço externo da Amazon e por isso devemos liberar

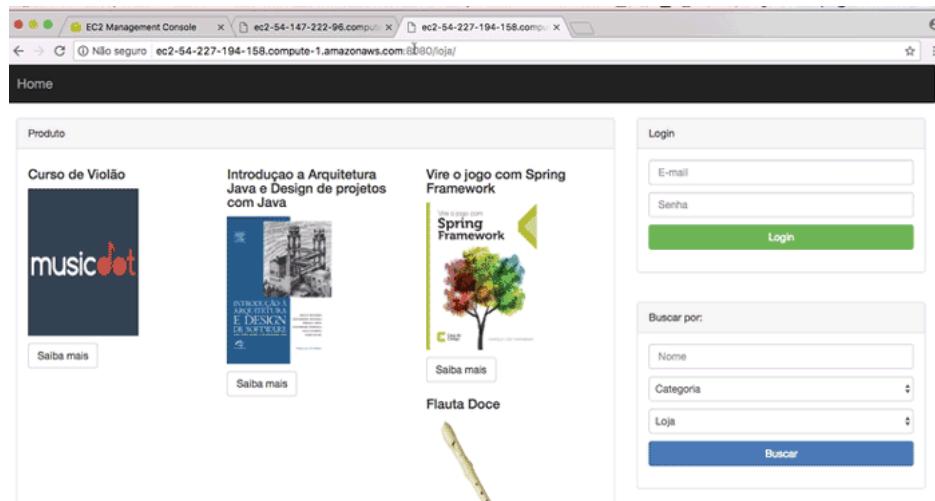
ainda a porta. Para simplificar e unificar a configuração vamos usar a mesma *Security Group* já configurada da outra instância. Para selecionar a nova instância e clique no botão direito *Instance Settings* -> *Change Security Group*:



Na nova janela desabilita o grupo atual e escolhe o grupo da primeira máquina configurada (no nosso caso `launch-wizard1`):



Uma vez configurado o mesmo primeiro *security group* para ambos as máquinas podemos testar o acesso direto à segunda instância.



Agora já podemos testar e qualquer alteração dos dados deve ser visível em ambos as aplicações. Isto é pois as duas instâncias estão usando o mesmo banco de dados.