

## Transcrição

[00:00] Nós temos agora dois servidores rodando na Amazon com aplicação na Casa do Código, e precisamos encontrar uma forma de dividir as requisições, os acessos dos usuários entre esses dois servidores que nós temos rodando na Amazon.

[00:16] A ideia é a seguinte, vamos voltar na apresentação. Temos esse segundo servidor que configuramos através das políticas do "auto scaling", onde colocamos aquelas métricas que caso ultrapasse 0,5% de utilização do CPU do Servidor, pediríamos para a Amazon instanciar um segundo servidor.

[00:35] Agora que temos esses dois servidores, temos que fazer com que esses usuários que vão acessar aplicação da caixa do código acessem um平衡ador de carga, que em inglês é conhecido como "load balancer", por isso temos essas iniciais aqui "LB".

[00:52] A ideia é que esses vários usuários que querem acessar a aplicação da Casa do Código vão primeiro acessar esse平衡ador de carga e esse平衡ador de carga é que vai se encarregar de acessar esses servidores fazendo a distribuição desses vários acessos, dessas várias requisições de usuários entre os servidores.

[01:12] A nossa etapa agora é fazer a configuração desse平衡ador de cargas, para que os usuários que queiram acessar a aplicação da Casa do Código acessem pelo "load balancer". Vamos fazer isso então?

[01:27] Eu vou voltar para o painel de console da Amazon. E temos que vir aqui, só confirmando que estamos na parte do EC2 que vamos ter que fazer essa configuração. Se não estiver aqui basta colocar na aba "service" e procurar pelo EC2, e já vamos ser redirecionado. Só confirmando as instâncias que estão rodando, e volta para o cenário onde já tinha a visualização dos dois servidores que estão rodando.

[01:54] Uma vez certificado que estamos aqui, vamos no menu lateral-esquerdo, e vamos na aba do "load balancer" para que possamos criar esse平衡ador de carga. Vamos aqui e colocamos a opção de "load balancer", e não temos nenhum平衡ador configurado, temos que criar esse平衡ador. Quando clicamos nessa opção, perceba que existem três opções de平衡ador de carga que a Amazon oferece para nós escolhermos.

[02:23] Essas informações estão na documentação da Amazon. Nessa documentação tem algumas descrições de quando que é melhor, mais vantajoso usar um平衡ador em comparação ao outro. Temos até uma tabela comparativa de que um平衡ador oferece em comparação ao outro.

[02:45] Em nosso projeto vamos utilizar o "classic load balancer", que vai realizar a distribuição igualitária entre esses dois servidores. Com o "classic load balancer", ora a requisição vai ser mandada para um servidor, hora vai ser mandada para o outro servidor, e já vai atender a nossa necessidade de balanceamento dessas várias requisições, desses vários usuários entre esses dois servidores.

[03:09] Então vamos voltar e dizer que vamos trabalhar com o "classic load balancer". Vamos clicar em "create", e vamos dar o nome para o nosso平衡ador de carga, o nome que quisermos para a identificação. Eu vou colocar, por exemplo, como sendo "balanceador-casadocodigo". Então esse é o nome desse nosso平衡ador que estamos criando. Agora aqui embaixo, perceba que temos algumas informações que devemos preencher.

[03:09] Temos duas informações de portas de comunicação, temos que ter um pouco de cuidado. Essa primeira configuração dessa porta, o "load balancer port", é a porta que os usuários da internet vão utilizar para poder acessar

esse "load balancer", esse balanceador que estamos criando. Por padrão, ele já coloca a porta "80", que é a porta utilizada pelo protocolo http para comunicação web. Podemos deixar aqui mesmo a porta de comunicação "80" a porta padrão da comunicação web do protocolo http.

[04:11] Agora temos que ter cuidado. Esse balanceador de carga vai ter que redirecionar essas requisições para esses dois servidores. Então ele tem que encontrar uma forma de acessar a porta de comunicação desses dois servidores, onde a aplicação da Casa do Código está rodando.

[04:32] Só que não podemos esquecer que a nossa aplicação da Casa do Código está rodando no Tomcat, e o Tomcat utiliza a porta de comunicação "8080". Então tenho que falar para esse balanceador de carga que quando ele for redirecionar essa requisição para os servidores, ele tem que acessar a porta "8080" dos servidores. Tenho que falar que a porta da instância, a porta do servidor que tenho que acessar é a porta "8080", cuidado aqui. A porta "8080" é a que tem o Tomcat rodando a aplicação da Casa do Código.

[05:04] Agora temos que fazer o seguinte, temos os dois servidores, um está na localidade "B", e o outro na localidade "C". Então tenho que falar que esse balanceador vai ter que trabalhar nessas duas localidades, na localidade "B", e na localidade "C". E para isso temos que selecionar essa opção "enable advanced VPC configuration", para que possamos indicar para esse balanceador quais são as localidades que ele vai ter que atuar.

[05:28] Então checamos essa opção. Aqui temos todas as localidades da região do norte da Virgínia. Vamos aqui, e colocamos o símbolo "+" nessas duas localidades a "B" e a "C" para que o balanceador possa atuar nessas duas localidades.

[05:45] Vamos colocar o símbolo "+" no "B" e o símbolo do "+" no "C" também. Com isso, o nosso balanceador vai passar a atuar nessas duas localidades. Então vamos clicar em "next", e assim como nas demais configurações dos servidores que subimos no banco de dados, precisamos colocar um grupo de segurança para acessar esse nosso balanceador. Então ainda não temos nenhum grupo de segurança previamente configurado, para acessar um balanceador de carga. Temos que criar um grupo de segurança.

[06:17] Então vamos em "create a new security group", para criar esse grupo segurança. Vamos colocar um nome mais descriptivo, por exemplo, grupo de segurança, abreviação em inglês é "SG" para acessar o "load balancer" o "LB". Então ser um grupo de segurança para acessar o balanceador de cargas o "LB". O que temos que fazer nesse grupo de segurança? Temos que liberar a porta "80", que é a porta que os usuários da internet vão acessar para poder chegar nesse balanceador de cargas. Vamos liberar a porta para todos os endereços IP.

[06:52] Feito isso, vamos seguir colocando "next", esse balanceador de carga, ele vai ficar checando, periodicamente, como que está o status desses servidores, se eles estão ativos, se estão respondendo. Essa etapa agora seria nessa configuração que vamos ter que fazer, indicando de quanto em quanto tempo esse balanceador vai fazer uma comunicação com os servidores, com o objetivo de verificar se eles estão ativos, se estão aptos a receber as requisições.

[07:26] Porque se o servidor ficará indisponível o balanceador vai parar de transmitir as requisições para ele, só vai mandar para o que estiver ativo. Para isso ele tem que fazer essa verificação constante para ver se os servidores estão ativos.

[07:37] Então vamos fazer um teste de conectividade do balanceador para esses servidores. Vou colocar no "ping" para utilizar o protocolo "TCP", e vamos fazer o teste do "ping" nos servidores na porta "8080", que é a porta que está rodando o Tomcat, e ela está aberta no grupo de segurança do EC2 para que seja possível ser acessada.

[08:01] Aqui já falamos que o nosso balanceador vai fazer essa verificação com os servidores da aplicação da Casa do Código, através do "ping", colocando a porta "8080", para verificar se essa porta "8080" está disponível. E falta agora

indicar quais são os servidores que vão estar vinculados a esse balanceador de carga. Precisamos colocar os nossos dois servidores, o da localidade "B" e o da localidade "C".

[08:28] Vamos escolher esses dois servidores, o da localidade "B" e o da localidade "C", para estarem vinculados a esse balanceador de cargas. Colocamos "next", se quisermos colocar alguma "tag" de identificação para esse balanceador, nesse caso não precisa, só temos esse único balanceador, não precisa ficar se preocupando com essa questão de gerenciamento. Basta vir aqui e colocar "review" e verifica que estamos criando um balanceador com esse nome.

[09:17] E estamos falando que a porta "80" desse nosso balanceador é que vai estar aberta para os usuários da internet acessarem, e o balanceador vai redirecionar para a porta "8080" das instâncias dos servidores que tem a aplicação na Casa do Código. Aqui temos a verificação de quanto em quanto tempo esse balanceador vai verificar o status de servidor se eles estão ativos. Aparentemente está tudo ok. Vamos pedir para criar esse nosso balanceador de carga.

[09:41] Temos a informação da Amazon que o nosso balanceador de cargas foi criado. Então se voltarmos, temos essa aba do "status" indicando que ainda temos zero de duas instâncias em serviço. Isso porque leva um tempo para esse processo de anexação de servidores no balanceador de carga seja concluído.

[10:05] Esperamos que esse status, depois dessa verificação ser realizada, ele indique que esse balanceador está vinculado com essas duas instâncias, com esses dois servidores, e vamos testar a nossa aplicação. Então vou só fazer agora uma pausa, porque esse processo pode demorar alguns minutos, e volto para fazer essa análise e testar a nossa aplicação para ver se conseguimos acessar pelo "load balancer", eu volto na sequência.

[10:28] Então agora temos as duas instâncias que tínhamos anexado nesse balanceador, e estão indicando que estão em serviço. Se for até a aba "instances", vemos que temos os dois servidores, um na localidade "C" e o outro na localidade "B", e que estão com o "status" em serviço, e anexados a esse balanceador de cargas e que tudo está funcionando. Então agora o que devemos fazer? Vamos verificar se conseguimos acessar aplicação da Casa do Código, através do balanceador de cargas.

[11:01] Vamos voltar para aba "descriptions", e vamos copiar esse DNS público desse balanceador de cargas, com o objetivo de verificar se conseguimos acessar a aplicação da Casa do Código. Então vou abrir uma outra aba no "browser" para colocar o DNS público.

[11:15] Agora eu tinha falado que esse balanceador de carga vai ser acessado pela porta "80", que é a porta padrão utilizada pelas comunicações web. Não preciso vir aqui e colocar ":80", até posso colocar aqui, mas não tem necessidade. Só para mostrar que ele funciona tanto colocando ou não, vou colocar ":80", vou colocar a aplicação da Casa do Código, e temos acesso a aplicação. Podemos tanto colocar aqui ":80", ou não, porque a porta "80" é a porta padrão utilizada pelas comunicações web, eu posso colocá-la ou não.

[11:54] Então agora vemos que conseguimos acessar a aplicação da Casa do Código aqui pelo balanceador de cargas, pelo "load balancer". E será que esse balanceamento entre os servidores está sendo realizado? Aqui no projeto da Casa do Código, já tínhamos configurado uma "JSP", para pegar um endereço IP de qual máquina, de qual servidor está respondendo a requisição. Vamos só confirmar qual é o servidor que está respondendo a essas requisições? Para ver se esse balanceamento está sendo feito?

[12:26] Vou colocar aqui a "JSP", que é "load balancer". E aqui temos que essa requisição foi respondida pelo servidor com o endereço IP "172.31.35.84". Esse valor deve ficar sendo alterado entre os dois. Se eu fizer uma requisição, quem respondeu agora "172.31.31.76", se eu fizer uma nova requisição, deve mudar para o outro servidor.

[12:52] Vamos ver, mudou para "84", vamos fazer mais uma, voltou para o "76". Então, perceba que as requisições que vamos fazendo vão sendo alternados entre esses dois servidores. O balanceamento de acesso de requisições está sendo dividido entre esses dois servidores com a aplicação da Casa do Código.

[13:12] Aparentemente está tudo funcionando. O balanceamento está sendo realizado. Agora, eu como usuário da Casa do Código, o que queremos fazer? Queremos comprar esse livro do "asp.net". Então venho aqui, e eu quero comprar esse livro do "asp.net". Vou vir no livro do "ap.net", no "impresso" e quero comprar esse livro. Vou vir aqui e clicar no "comprar".

[13:36] Vamos esperar, mas o que aconteceu? Cadê o nosso livro do "asp.net" no carrinho de compras? Ele sumiu, por que isso aconteceu? Vamos tentar entender na sequência o que está acontecendo, e como conseguimos resolver esse problema.