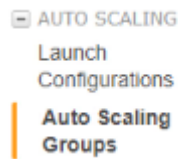
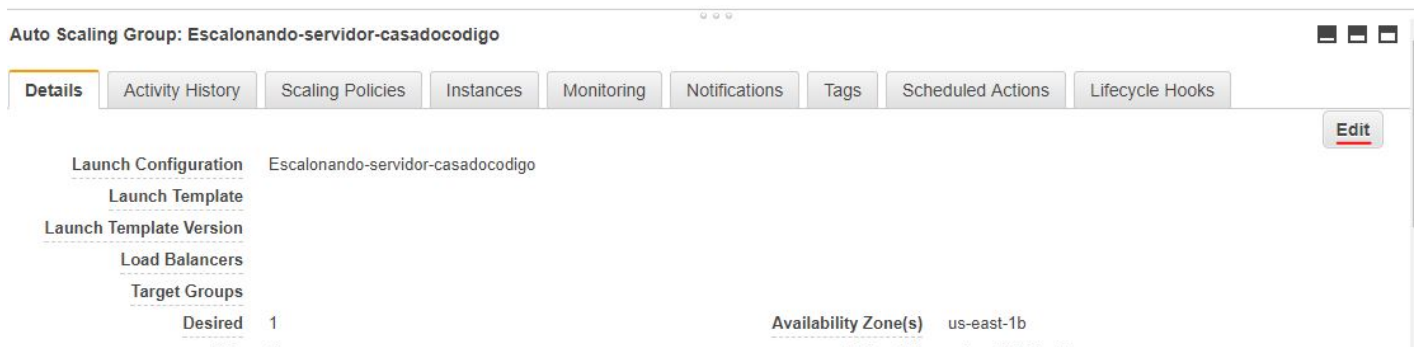


Devemos definir agora quais serão as políticas utilizadas para poder fazer esse escalonamento. No painel de console da Amazon, pesquisar pelo serviço EC2, feito isso, no menu lateral esquerdo pesquise pela aba Auto Scaling e clique em Auto Scaling Groups.



Na sequência, clique no grupo de auto-escalonamento criado e vá até a aba Details. Vamos primeiro indicar que os demais servidores deverão utilizar a configuração customizada que criamos, Clique no botão Edit e coloque na opção Launch Configuration a configuração customizada, imagem-customizada-casadocodigo. No projeto, vamos configurar para que o mínimo de servidores ativos seja 1 e nesses casos, que teremos uma grande demanda de acessos que seja configurado um segundo servidor, Min: 1 e Max:2 servidores.



Para finalizar, especifique as localidades (zonas de disponibilidade) nas quais iremos configurar esses servidores. O primeiro servidor configuramos na zona de disponibilidade us-east-1b, a fim de garantir sempre uma disponibilidade de nossos serviços, vamos configurar uma segunda zona de disponibilidade para o segundo servidor do grupo de auto-escalonamento, por exemplo a zona de disponibilidade us-east-1c. No fim, salve essas alterações da aba Detalhes, clicando no botão **Save**

The screenshot shows the 'Launch Configuration' page in the AWS Management Console. The 'Launch Configuration' dropdown is set to 'imagem-customizada-casadocodigo'. The 'Launch Template' and 'Launch Template Version' sections are empty. The 'Load Balancers' and 'Target Groups' sections are also empty. The 'Desired' instance count is set to 1, and the 'Min' instance count is set to 1. The 'Max' instance count is set to 2. The 'Health Check Type' is set to 'EC2'. The 'Availability Zone(s)' is set to 'us-east-1b'. The 'Subnet(s)' dropdown is open, showing two options: 'subnet-6cd4b808(172.31.16.0/20)' and 'subnet-9928f6b6(172.31.32.0/20)'. The 'Default Cooldown' is set to 300 seconds. The 'Placement Group' is empty.

Agora devemos configurar as políticas de escalonamento para que a Amazon "suba" esse segundo servidor. Vá até a aba **Scaling Policies** e crie uma política com o nome **Aumentando servidores** e clique na opção **Create a scaling policy with steps**.

The screenshot shows the 'Create Scaling policy' dialog box. The 'Name' field is set to 'Aumentando-servidores'. The 'Metric type' is set to 'Average CPU Utilization'. The 'Target value' field is empty. The 'Instances need' field is set to 300, with the text 'seconds to warm up after scaling' next to it. The 'Disable scale-in' checkbox is unchecked. At the bottom, there are two links: 'Create a simple scaling policy' and 'Create a scaling policy with steps'. A mouse cursor is hovering over the 'Create a scaling policy with steps' link.

Feito isso, clique na opção **Create new alarm** para que configuremos assim a política de escalonamento. A política que iremos criar será para analisar o consumo de CPU do servidor que temos no momento, se esse valor for ultrapassado, a Amazon irá subir o segundo servidor com as configurações que fizemos. Para evitar um longo processo de espera, coloque um consumo baixo, por exemplo 0.5% por um período de 1 minuto:

Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.
To edit an alarm, first choose whom to notify and then define when the notification should be sent.

☐ Send a notification to:

Whenever: of ☐ ☐

Is: Percent

For at least: consecutive period(s) of

Name of alarm:

Para finalizar, devemos especificar que ação tomar quando esse limite for ultrapassado, vamos querer justamente fazer o escalonamento para "subir" uma instância, chegando ao valor máximo que estipulamos de 2 instâncias e clique no botão **Create** para finalizar a criação da política de escalonamento.

Create Scaling policy

Name:

Execute policy when:

breaches the alarm threshold: CPUUtilization >= 0.5 for 60 seconds
for the metric dimensions AutoScalingGroupName = Escalonando-servidor-casadocodigo

Take the action: instances when <= CPUUtilization < +Infinity

[Add step](#) ⓘ

Agora só falta nós fazermos o teste, para isso vamos utilizar o Apache Bench para simular várias requisições de usuários, se estiver no Windows abra o Prompt e mude para o diretório onde teremos a pasta bin do XAMPP que por padrão estará em **C:\xampp\apache\bin** e faça o teste simulando por exemplo 500 requisições que já deverá ser suficiente para ultrapassar o limiar estipulado de 0.5% e deveremos ter assim a criação da segunda instância (Esse processo poderá demorar alguns minutos)

```
ab.exe -n 500 [IP Público instância EC2 us-east1-b]:8080/casadocodigo
```

COPIAR CÓDIGO

Caso esteja no Linux ou no Mac abra o terminal e coloque (Esse processo poderá demorar alguns minutos):

```
ab -n 500 [IP Público instância EC2 us-east1-b]:8080/casadocodigo
```

[COPIAR CÓDIGO](#)

Feito isso, volte para o painel de console com as instâncias do EC2, faça uma atualização e veja se o segundo servidor foi criado. Tente acessar a aplicação da Casa do Código nesse segundo servidor. Qual é o resultado? Você consegue acessar a aplicação da Casa do Código nesse servidor que fazia parte do grupo de auto-escalonamento?