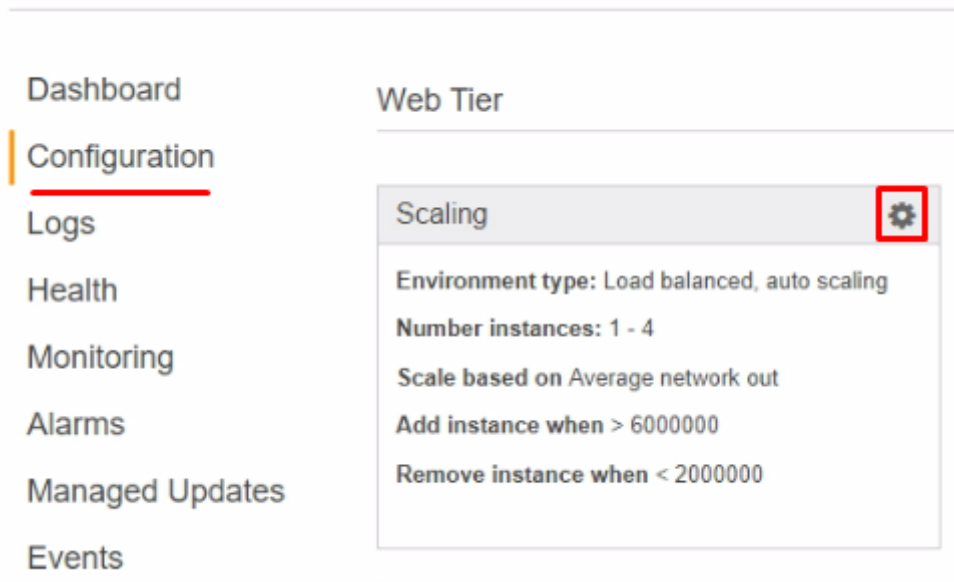


Vamos agora verificar como podemos realizar o escalonamento de nossa aplicação para os casos em que tivermos uma alta demanda de acesso. Para isso, no painel de console do serviço do *Elastic Beanstalk* clique na opção **Configuration** e posteriormente clique na engrenagem de configuração referente ao escalonamento (*Scaling*)



Posteriormente, clique na opção referente a configuração do auto escalonamento e configure o *Elastic Beanstalk* para realizar o escalonamento para no máximo 2 servidores visando esses casos em que tivermos uma grande demanda de acesso

### ▼ Auto Scaling

Use the following settings to control auto scaling behavior. [Learn more](#)

Minimum instance count:  Minimum number of instances to run.

Maximum instance count:  Maximum number of instances to run.

Feito isso, devemos configurar quais serão as métricas que deverão ser analisadas com o objetivo de configurar esse segundo servidor. Vamos configurar o escalonamento de nossa aplicação para considerar a soma das quantidades de requisições feitas para nossa aplicação, para isso, altere a opção *Trigger measurement* para o valor *RequestCount*, *Trigger statistic* para o valor *Sum* e a opção *Unit of measurement* para o valor *Count*

### ▼ Scaling Trigger

Trigger measurement:	<input type="text" value="RequestCount"/>	The measure name associated with the metric the trigger uses.
Trigger statistic:	<input type="text" value="Sum"/>	The statistic that the trigger uses when fetching metrics statistics to examine.
Unit of measurement:	<input type="text" value="Count"/>	The standard unit that the trigger uses when fetching metric statistics to examine.

Na sequência, configure a quantidade de requisições que será considerada para realizarmos esse escalonamento, vamos considerar uma somatória de 150 requisições, ou seja, se em um período de 5 minutos for detectado um pico de acesso que totalize 150 requisições, vamos pedir para a Amazon escalonar nossa aplicação para termos dois servidores.

Nos casos que tivermos uma baixa quantidade de acessos, vamos pedir para a Amazon remover esse servidor adicional, para isso, vamos considerar por exemplo um valor de 20 requisições.

Upper threshold:	<input type="text" value="150"/>	The upper limit for the metric. If the data points exceed the threshold for the period set as the breach duration, the trigger is activated.
Upper breach scale increment:	<input type="text" value="1"/>	The incremental amount to use when performing scaling activities when the upper threshold has been breached. Must be an integer, optionally followed by a % sign.
Lower threshold:	<input type="text" value="20"/>	The lower limit for the metric. If the data points are below this threshold for the period set as the breach duration, the trigger is activated.
Lower breach scale increment:	<input type="text" value="-1"/>	The incremental amount to use when performing scaling activities when the lower threshold has been breached. Must be an integer, optionally followed by a % sign.

Uma vez que configuramos as políticas de escalonamento clique no botão **Apply** para que essas mudanças surtam efeito nesse ambiente.