

## lab9 - 应用程序-自动缩放功能

在实验9中，我学习了如何自动缩放OpenFaaS。具体来说，它具有与Prometheus的Alertmanager链接的机制。

开箱即用配置OpenFaaS，使其将根据 request per seconds 通过Prometheus测量的度量标准进行自动缩放。当流量通过API网关时会捕获此度量。如果 request per seconds 超过了定义的阈值，则AlertManager将触发。应将此阈值重新配置为适合生产使用的级别。

其实就是自动调整负载。

打开普罗米修斯并添加一个图：

在普罗米修斯上执行这个，`rate( gateway_function_invocation_total{code="200"} [20s])`

The screenshot shows the Prometheus web interface. At the top, there's a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below this, there's a search bar with the query `rate(gateway_function_invocation_total{code="200"} [20s])`. A blue 'Execute' button is visible. Below the query bar, there are tabs for 'Graph' and 'Console'. The 'Graph' tab is active, showing a 'Moment' view. Below the graph area, there's a list of elements, each representing a function invocation for a specific code ('200') and instance ('10.0.1.31:8082'). The elements are:

- `{code="200",function_name="env",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="figlet",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="hello-openfaas",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="issue-bot",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="long-task",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="markdown",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="nodeinfo",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="sentimentanalysis",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="show-html",instance="10.0.1.31:8082",job="gateway"}`
- `{code="200",function_name="sleep-for",instance="10.0.1.31:8082",job="gateway"}`

☐ Enable query history

rate( gateway\_function\_invocation\_total{code="200"} [20s])

Execute

- insert metric at cursor -

Graph

Console

- 1h +

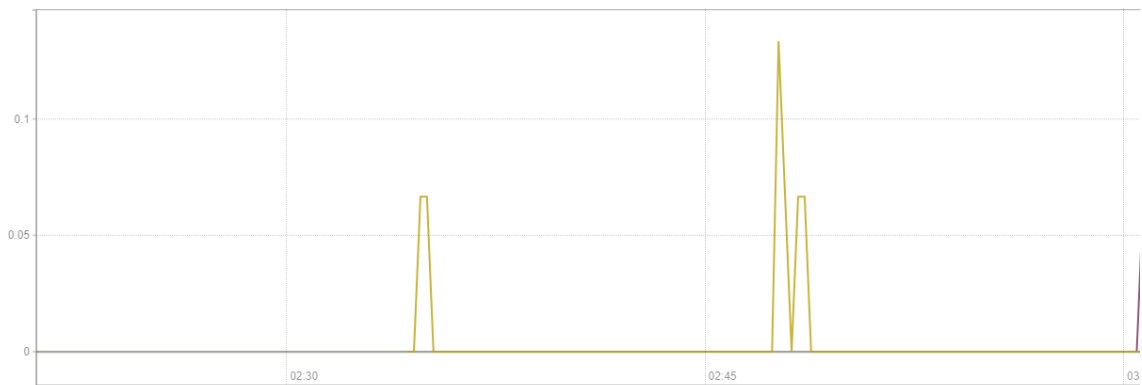
◀

Until

▶

Res. (s)

☐ stacked



<http://127.0.0.1:9090/alerts> 可以查看何时超过每秒请求的阈值。

# Alerts

☐ Show annotations

/etc/prometheus/alert.rules.yml > prometheus/alert.rules

**APIHighInvocationRate** (0 active)

**service\_down** (0 active)

## 触发警告测试

```
$ git clone https://github.com/alexellis/echo-fn \  
&& cd echo-fn \  
&& faas-cli template store pull golang-http \  
&& faas-cli deploy \  
  --label com.openfaas.scale.max=10 \  
  --label com.openfaas.scale.min=1
```



```
$ for i in {0..10000};
do
    echo -n "Post $i" | faas-cli invoke go-echo && echo;
done;
```

[Enable query history](#)

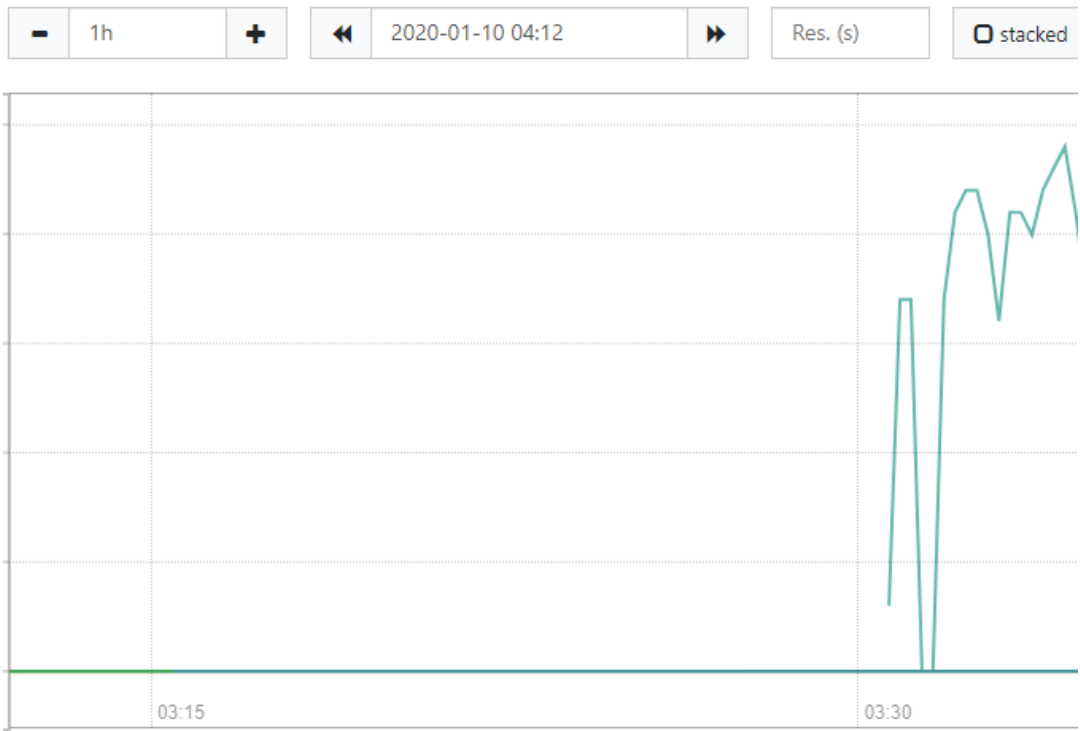
rate( gateway\_function\_invocation\_total{code="200"} [10s])

Execute

- insert metric at cursor -

Graph

Console



可以看到go-echo的调用次数一直增加。

