

How to use the load balancer

Contents

I.	General explanation.....	2
A.	Definition.....	2
B.	Prerequisites	2
II.	Implementing a load balancer	3
A.	Configure Leap GIO Public	3
B.	Configure the VMs	6
C.	The load balancer in action.....	7

I. General explanation

A. Definition

- The **load balancer** is used to distribute the traffic load between a minimum of two or more connections, to optimize the traffic, maximize the throughput, minimize the response time, and avoid over heavy load on only one of the connection paths
- The **load balancer** can also be described as distributing the traffic load between two or more VMs. Therefore, on Leap GIO Public, a minimum of two VMs need to be created before the load balancer can be implemented

B. Prerequisites

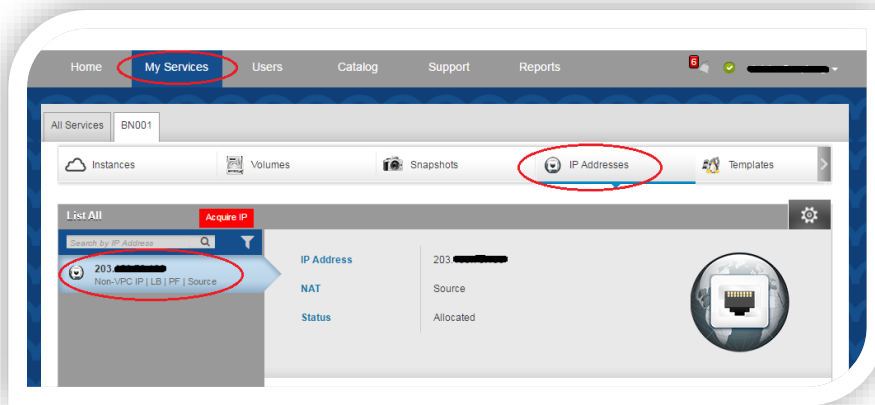
- A **minimum of two VMs** need to be created (preferably using the same OS, but it is up to your requirements)
- The VMs need to **belong to the same network** before a load balancer can be implemented
- Work needs to be done **on Leap GIO Public** and also **on the VMs** that would be used for load balancing for the load balancer feature to work

II. Implementing a load balancer

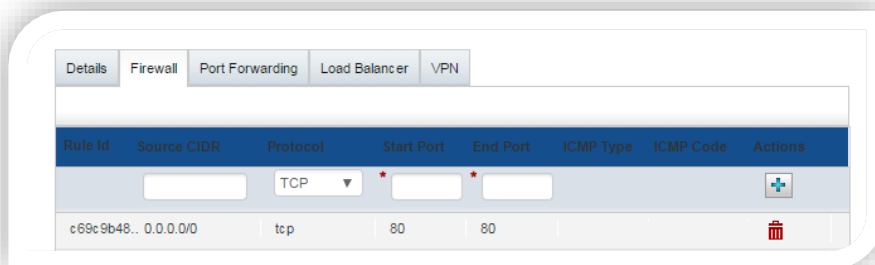
- To demonstrate how the load balancer works on Leap GIO Public, we will use an example using **two CentOS 7 VMs** that belong to the **same network**. It is to be noted that **the configuration on Leap GIO Public** for any type of OS should be **similar**. However, depending on the OS used, **the configuration of the VMs** might be **different** from OS to OS and even from release to release.

A. Configure Leap GIO Public

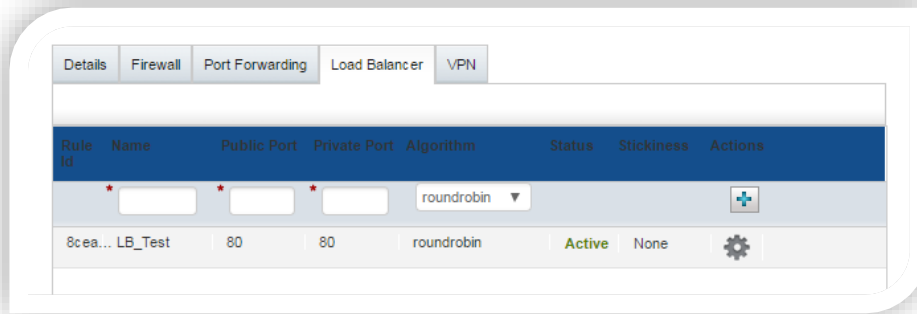
- To summarize, we will allow port 80 for access from a web browser. From that browser and through an HTML page configured on both VMs, we will be able to see the load balancing between these two VMs using the round robin algorithm
- We assume that two VMs using CentOS 7 have already been created (server02 and server05 in our example)
- Go to the GIOCloud Portal => My Services => Manage Resources => IP Addresses
- If you have more than one Public IP, select the Public IP of the network which has the two VMs



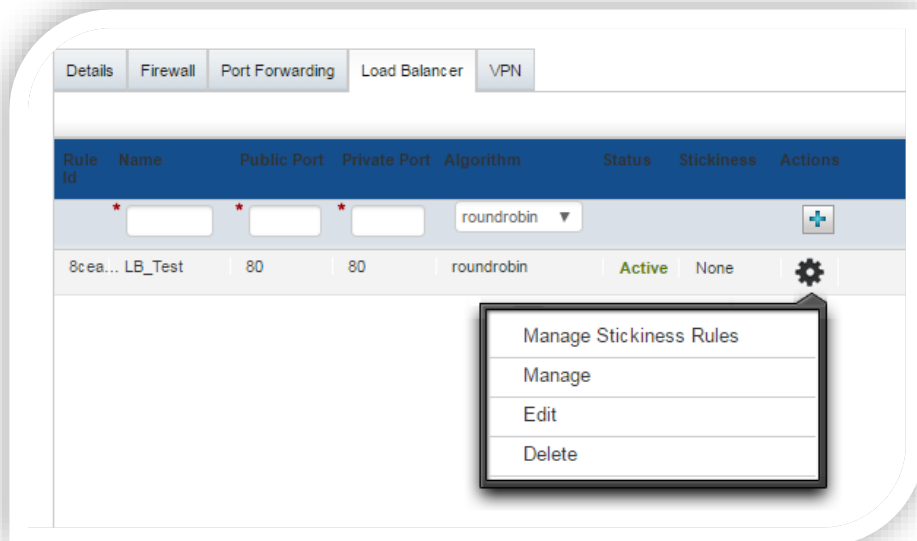
- Click on the **Firewall** tab and add a new rule with below parameters (using the usual parameters to make it simple to understand, but these can be configured in a different way for security reasons)
 - o **Source CIDR:** 0.0.0.0/0
 - o **Protocol:** TCP
 - o **Start Port:** 80
 - o **End Port:** 80



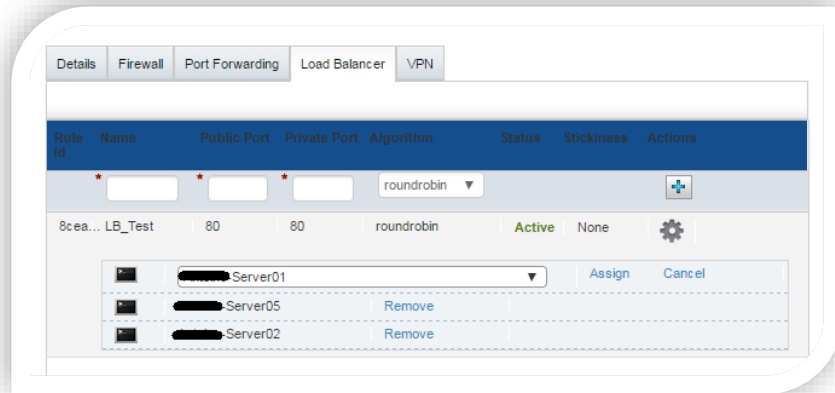
- Click on the **Load Balancer** tab and add a new rule with below parameters
 - o **Name:** A name for the Load Balancer rule
 - o **Public Port:** The externally accessible port that you would like to balance
 - o **Private Port:** The corresponding private port on a VM
 - o **Algorithm:** The process by which this balance will occur. Choose from the drop-down options
 - **Roundrobin:** Let's say you have 2 servers waiting for requests behind your load balancer. Once the first request arrives, the load balancer will forward that request to the 1st server. When the 2nd request arrives, that request will then be forwarded to the 2nd server and so on in a cyclical fashion.
 - **Leastconn:** This algorithm takes into consideration the number of current connections each server has. When a client attempts to connect, the load balancer will try to determine which server has the least number of connections and then assign the new connection to that server.
 - **Source:** Source IP Hash load balancing uses an algorithm that takes the source and destination IP address of the client and server and combines them to generate a unique hash key. This key is used to allocate the client to a particular server. As the key can be regenerated if the session is broken this method of load balancing can ensure that the client request is directed to the same server that it was using previously.



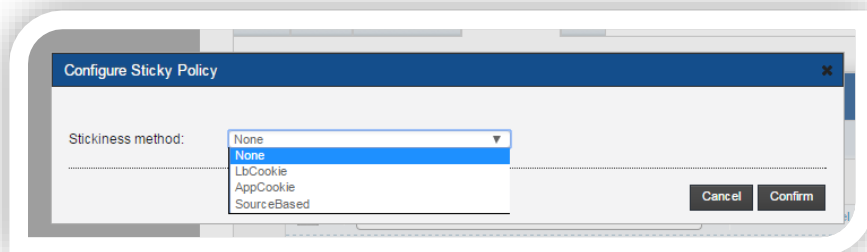
- Click on the **gear icon** and select **Manage**



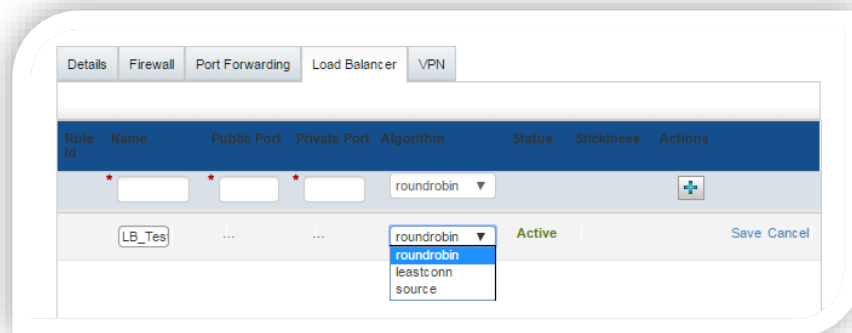
- Pick the **first** server that needs to be load balanced and click on **Assign**. Then, repeat the same process for the **second** server that needs to be load balanced. In our case we have assigned our two servers with CentOS 7.



- When clicking on the **gear icon of a rule**, you can also do the below (We leave those by default for our example):
 - o **Manage Stickiness Rules:** Sticky sessions are used in Web-based applications to ensure continuous availability of information across the multiple requests in a user's session. For example, if a shopper is filling a cart, you need to remember what has been purchased so far. The stickiness method could be **load balancer-generated cookie**, **application-generated cookie**, or **source-based**. In the source-based method, the source IP address is used to identify the user and locate the user's stored data. In the other methods, cookies are used. The cookie generated by the load balancer or application is included in request and response URLs to create persistence. The cookie name can be specified by the administrator or automatically generated. A variety of options are provided to control the exact behavior of cookies, such as how they are generated and whether they are cached.



- o **Edit:** Edit the name or the algorithm method



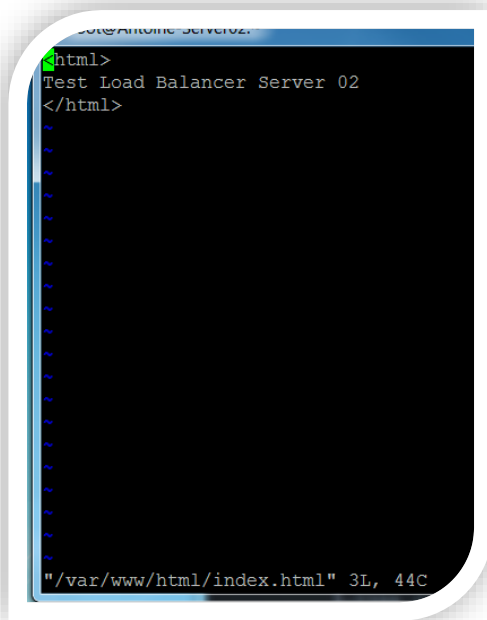
- **Delete:** Delete the rule
- Once the above steps have been completed, we are done with configuring the Leap GIO Public part

B. Configure the VMs

- Below is the configuration for CentOS 7 VMs. Different configuration or commands may apply for other OS or releases of CentOS.
- On **EACH** of the CentOS 7 VM:
 - Install httpd: **yum -y install httpd**
 - Enable httpd: **systemctl enable httpd**
 - Start httpd: **systemctl start httpd**
 - Create index.html: **vi /var/www/html/index.html** then save and exit
 - For VM Server02:


```
<html>
Test Load Balancer Server 02
</html>
```
 - For VM Server 05:

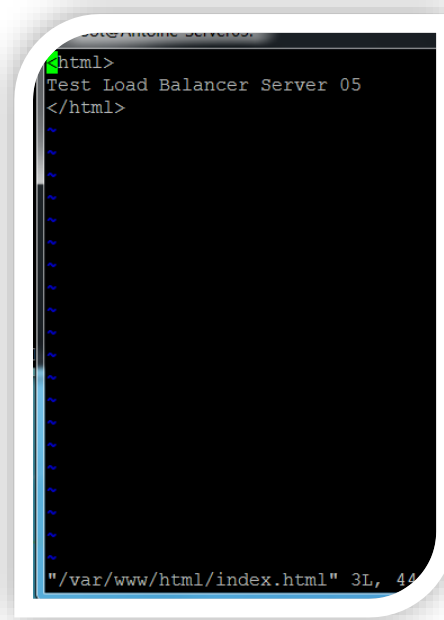

```
<html>
Test Load Balancer Server 05
</html>
```



```

<html>
Test Load Balancer Server 02
</html>
~/var/www/html/index.html" 3L, 44C

```



```

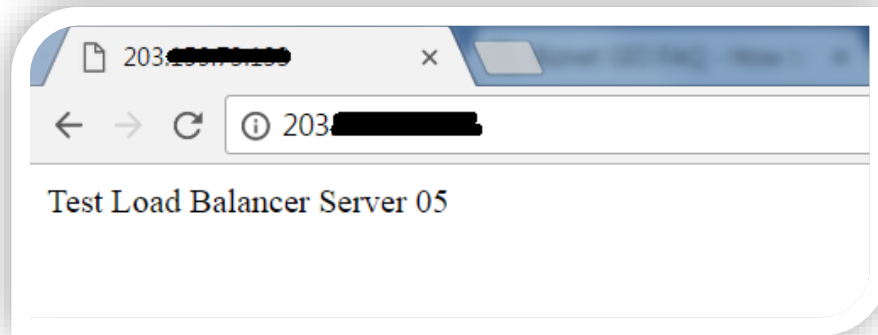
<html>
Test Load Balancer Server 05
</html>
~/var/www/html/index.html" 3L, 44C

```

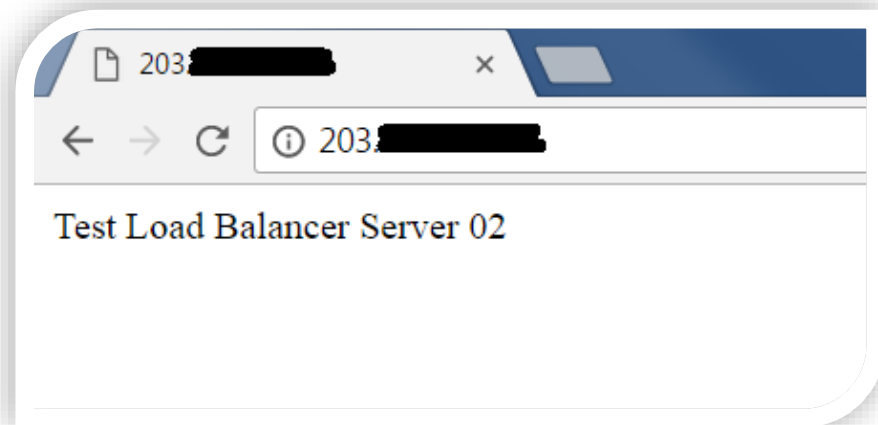
- Allow port 80 on the firewall:
 - **firewall-cmd --add-port=80/tcp --permanent**
 - **systemctl stop firewalld**
 - **systemctl start firewalld**
- Once the configuration of both VMs is completed and the configuration of the Leap GIO Public part is also completed, the load balancer is ready to work

C. The load balancer in action

- By opening a web browser on any machine that has access to internet, type the public IP of the network
- If everything was configured correctly on Leap GIO Public and on the 2 CentOS 7 VMs, you should be able to see the below



- After a refresh, you should then be able to see the below



- And so on after each refresh. This example is showing that using the round robin algorithm in our case, the web browser is accessing the html page on Server02 then Server05 and so on in a cyclical way after each refresh

If you have any questions please check our FAQ section. If you still cannot find what you are looking for or believe that there is a careless mistake in this document, please contact our support at support@leapsolutions.co.th or send us your inquiry through our [Inquiry Form](#) located on your Web Portal.