



[Portal](#) > [Knowledgebase](#) > [Technical](#) > [Networking](#) > [Load Balancing without Load Balancers](#)

Load Balancing without Load Balancers

Jem Camba - 2019-03-06 - in [Networking](#)

What is Load Balancing?

In general terms, Load Balancing refers to a form distribution of all incoming network traffic across two or more servers. In modern workloads, where you need to serve thousands, if not millions of incoming requests.

Setting up load balancing in front of your stack, helps by not only sharing this load between the servers, but also avoiding single points of failure.

Load Balancers can be of a hardware or a software form. They can vary a lot, depending on the specific use-cases and the amount or form of traffic that they need to work under.

There are a few commonly used load balancing algorithms, but on this guide, we will go through a simple Round Robin, where requests are served in turns by the available servers.

Why BGP?

BGP is a fundamental building block of the internet, and is the mechanism by which core backbone internet routers know where to route IP packets.

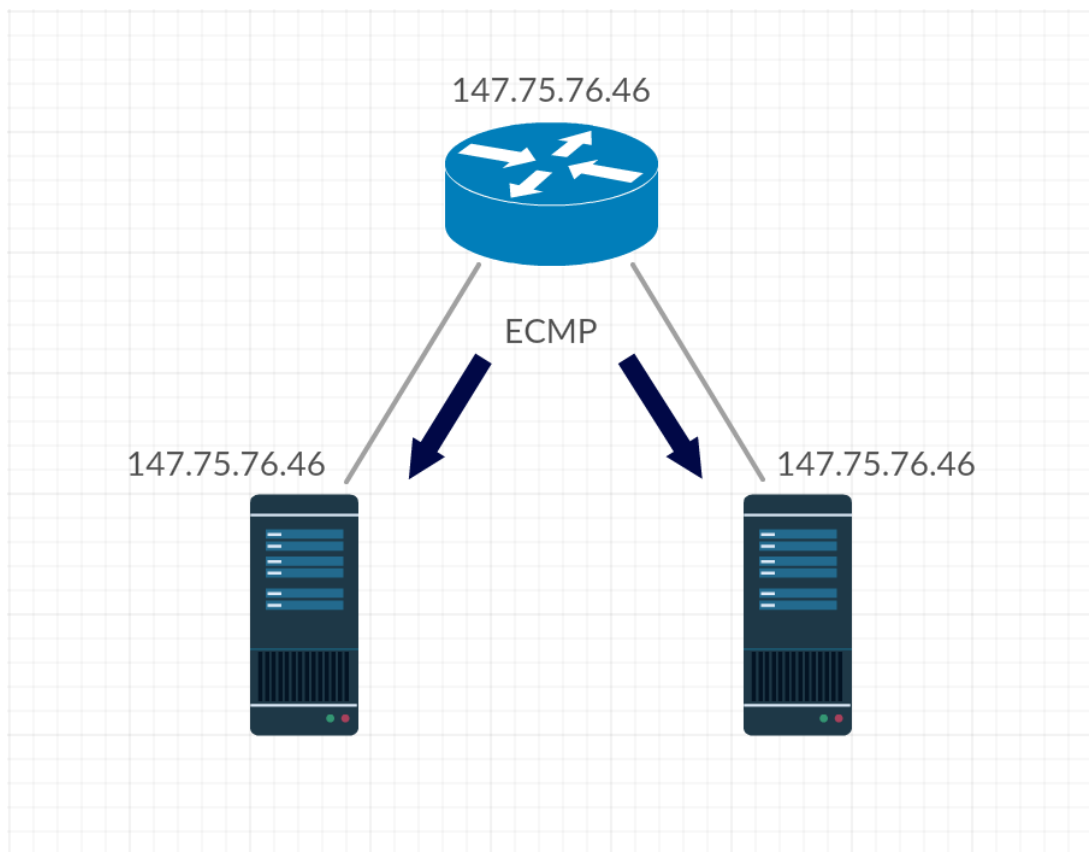
Part of Packet's value prop is automating core network/compute/storage to provide optimal value and performance to our users. To encourage this, we offer a unique self-serve BGP feature that allows customers to specify an active server instance as announcing BGP.

Local BGP & Load Balancing

BGP, on its own, is not able to do any sort of load balancing. But we will be able to take advantage of a great feature our core routers have, Equal-Cost Multi Path (ECMP), which will be better explained below.

For this guide, we'll be using Local BGP to announce an Elastic IP simultaneously from 2 different servers.

When a request for the IP 147.75.76.46 goes to our core router, it sees that it has two available paths / servers to send this traffic to. And so the router uses the ECMP logic to create a Round Robin load balancing, by choosing the available paths in turns. If any of these servers goes down, no problem, the other server will still be serving the IP 147.75.76.46 , and there will be no downtime.



Configuration

1. Have (2) two t1.small servers ready to work on.
2. Request an Elastic IP for the project, on the same facility as the above servers.
3. Enable Local BGP for the project.
4. Use Local BGP to announce the Elastic IP from both servers. (The process has been explained in detail [here](#))

If you successfully announced the elastic IP from both the servers, all requests to that IP will be served by them in a Round Robin way.

Here is an example of a simple website answering from 2 servers:

```
curl 147.75.76.46
```

```
<!DOCTYPE html>
```

```
...
```

```
Welcome to nginx 1 !
```

```
...
```

```
curl 147.75.76.46
```

```
<!DOCTYPE html>
```

```
...
```

```
Welcome to nginx 2 !
```

```
...
```

Tags

load balancing