



Today we are going to talk about the Azure **Load Balancer (LB)** service.

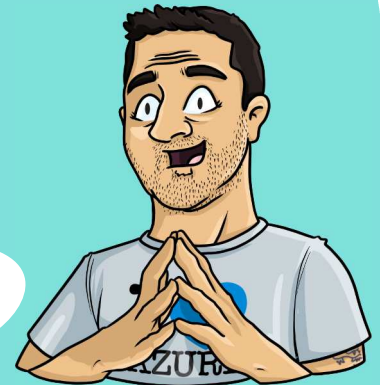
It is therefore part of the traffic distribution offer, just like Front Door, Traffic Manager and Application Gateway?!

Exactly.

LB operates on the **Transport layer (4)** of the OSI model, so it does **not offer SSL offloading**.

By default, LB offers a **distribution mode** based on **hash**, i.e. the combination of different elements such as source IP, source port, destination IP, destination port and type of protocol.

These elements are called **tuples**.



Is there another method of distribution?

Yes, that of **session affinity** with 2 options (IP or IP+protocol), in which each user is always redirected to the same instance, often for performance reasons.

In addition to the 2 distribution modes, LB is offered in 2 types:

The 1st one is the **public LB**, which is exposed on the internet, and therefore allows traffic from external sources to be distributed to private resources on Azure, such as VMs or VMSS*.

The idea is obviously to reduce the attack surface.

The 2nd one is the **private LB** which distributes traffic from private sources to other private resources.



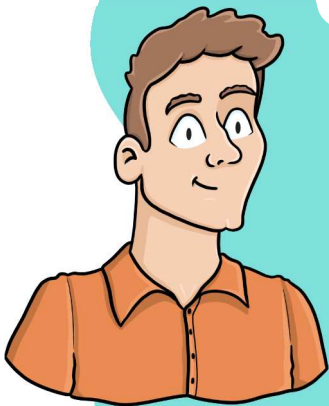
Oh yeah, that's pretty smart.

next



LB is available in 2 service tiers, **Basic** or **Standard**.

The **Basic tier** is the basic solution with few options, limited to VMs in an Availability Set, or to VMSSs. And above all, it does not offer **any SLA**, therefore to be preferred for tests or non-production environments. By default, traffic is allowed.



The **Standard tier** offers many options, such as distributing traffic within and between **regions** and **Availability Zones** for high resiliency. Inbound traffic is denied by default unless it is allowed at the NSG level.

But also the distribution of traffic to the VMs or VMSSs of a VNET. It is possible to configure incoming or outgoing **NAT traffic** rules, it supports **Private Link**, and above all it offers a **99.99% SLA**.



The difference in options between the 2 is huge, so you have to choose the right one from the beginning.

Yes it is preferable, knowing that it is **not possible to convert** an LB Basic into LB Standard, once created.

If needed, Microsoft provides a script to migrate from an LB Basic to an LB Standard. But in reality it just creates a new LB and copies the existing configuration to the new LB.



So it is better to identify from the beginning, the needs that we will have in order to choose the right service tier to avoid tinkering afterwards!



Yes, especially since there are **limitations** for each service tier, which can complicate this "migration", so you might as well avoid problems.



$$P=m \cdot V \quad T=2\pi \sqrt{\frac{l}{g}}$$



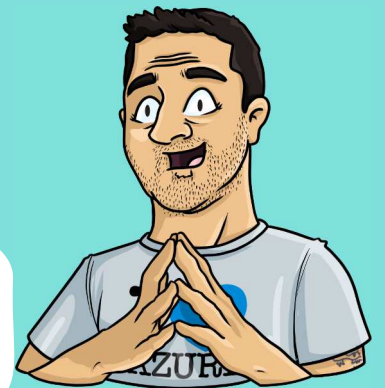
And setting up a LB isn't too complicated?

Not at all.

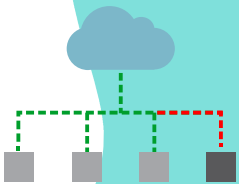
Once you have chosen your **LB type**, public or private, and the Basic or Standard **service tier**, you simply have to follow these steps:

We obviously start by defining a **Frontend IP**, it is simply the IP of your LB, your front door.

Then you configure your **Backend Pool**, i.e. the instances to which the traffic must be sent, this is the exit door.



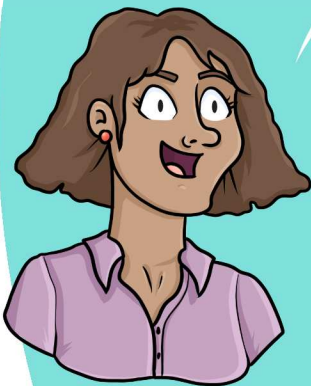
An important step is the implementation of **Health Probes**, which will make it possible to evaluate the status of an instance and remove it from the pool if it fails. The probe simply consists of the protocol and port of the instance that will be monitored.



And we end with a **Rule** that associates all the elements: the frontend IP, the backend pool, and the probe, and that's it!

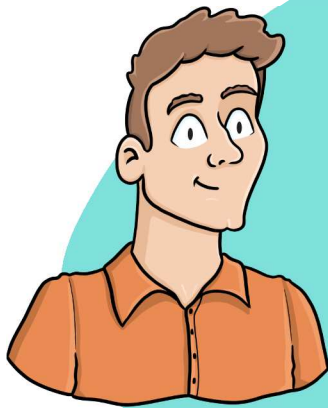
You mentioned **incoming NAT rules** earlier, can you tell us more?

Yes, this allows you to redirect traffic to a **different port** than the one exposed to clients.



The Public LB also offers management of **outgoing NAT rules**, which allows all instances in the backend pool to use the LB's public IP to go out on the Internet.

Cool. So if we need to provide our outgoing IPs to a partner in case he must authorize them on his side, it is actually easier to manage.



And in your opinion, which service tier, should we use?

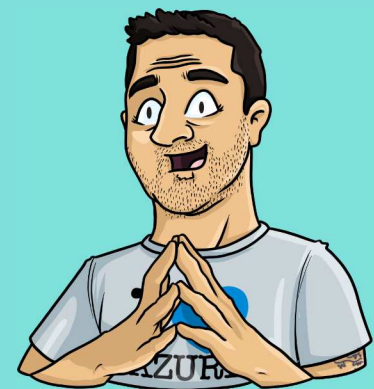
As often, it will depend on your **use case**, and the **features** you are looking for.

But with all we said earlier, it is preferable to use the **Basic tier** for tests or in a non-production environment. But it is strongly discouraged to use it on a Production environment knowing that it does **not offer SLA**.

With customers, I often come across the Standard tier, but again it will depend on your use case, and also on the price. We didn't talk about the cost of the LB, but **Standard tier is billed** unlike Basic, which is free, hence the missing of SLA.



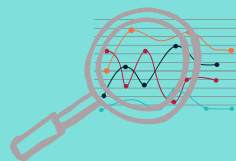
As you remind regularly, it is **strongly discouraged** to use **services without SLA** or in **Preview** for **Production** environments to avoid possible problems during incidents.



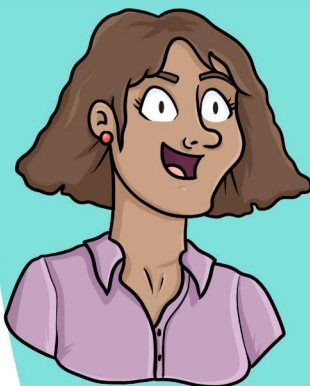
It is important to remember

In terms of integration with other Azure services, LB is natively supported by **Azure Monitor**.

You can export performance metrics to **Log Analytics Workspace** to perform analysis via **KQL** requests.



The Insights feature of Azure Monitor provides information such as the status of your LB, or the availability of your instances in the backend pool.



So to sum up, LB is the go-to solution when looking for an ordinary traffic distribution solution without SSL offloading!

Thank you!



If you want to continue **learning** in a fun way about the **Azure ecosystem**, and not miss any of our illustrations ...

... Feel free to subscribe at:



<https://aka.ms/grow-una>



<https://tinyurl.com/youtube-growuna>

If you like our work, please share it ;o)

See you soon!

