Load Balancers improve the availability of a system.  
  
But what if the load balancer goes down?  
  
This is a common point that I see coming up during design discussions.  
  
But it is often hand-waived away.  
  
"Oh - the cloud provider will take care of it!"  
  
And that’s mostly true.  
  
Modern cloud systems have reached a point where developers don't have to be concerned with such details.  
  
✅ But let's consider what a high-availability load balancer setup looks like.  
  
When you talk about a highly available load balancer, you are actually saying that you don't want the LB to become a single-point-of-failure.  
  
And how do we remove a single point of failure?  
  
By investing in redundancy, of course.  
  
As per the example screenshot below, we can have multiple load balancer instances behind a Static IP.  
  
👉 When a user accesses your website, the request goes through the external IP address to the active load balancer.  
  
👉 If that LB instance fails for some reason, the failover mechanism will detect it  
  
👉 It will automatically reassign the IP address to the passive LB instance that will take over.  
  
But how to setup this active-passive switch-over?  
  
One simple way is to use the keepalived process on both the instances of the load balancer.  
  
The keepalived process monitors the health of the load balancers and reassigns the IP address to the healthy instance.  
  
Having said that, what if the Static IP faces issues due to DNS resolution problems or network issues?  
  
Well - that's why 100% availability is elusive.  
  
Ultimately, things can and will go wrong somewhere and sometime.  
  
So - have you faced availability issues with Load Balancers?  
  
And do you take any specific measures for managing availability at the infra level?  
  
Also, if System Design discussions like this interest you, consider subscribing to my free weekly newsletter.  
  
It's far more detailed and organized.  
  
Here's the link:  
<https://lnkd.in/g-Y6Evsp>

Activate to view larger image,

