Types Of Scalability

In this lesson, we will explore the two types of scaling: Vertical and Horizontal Scaling.

We'll cover the following

- What is Vertical Scaling?
- What is Horizontal Scaling?
- Cloud Elasticity

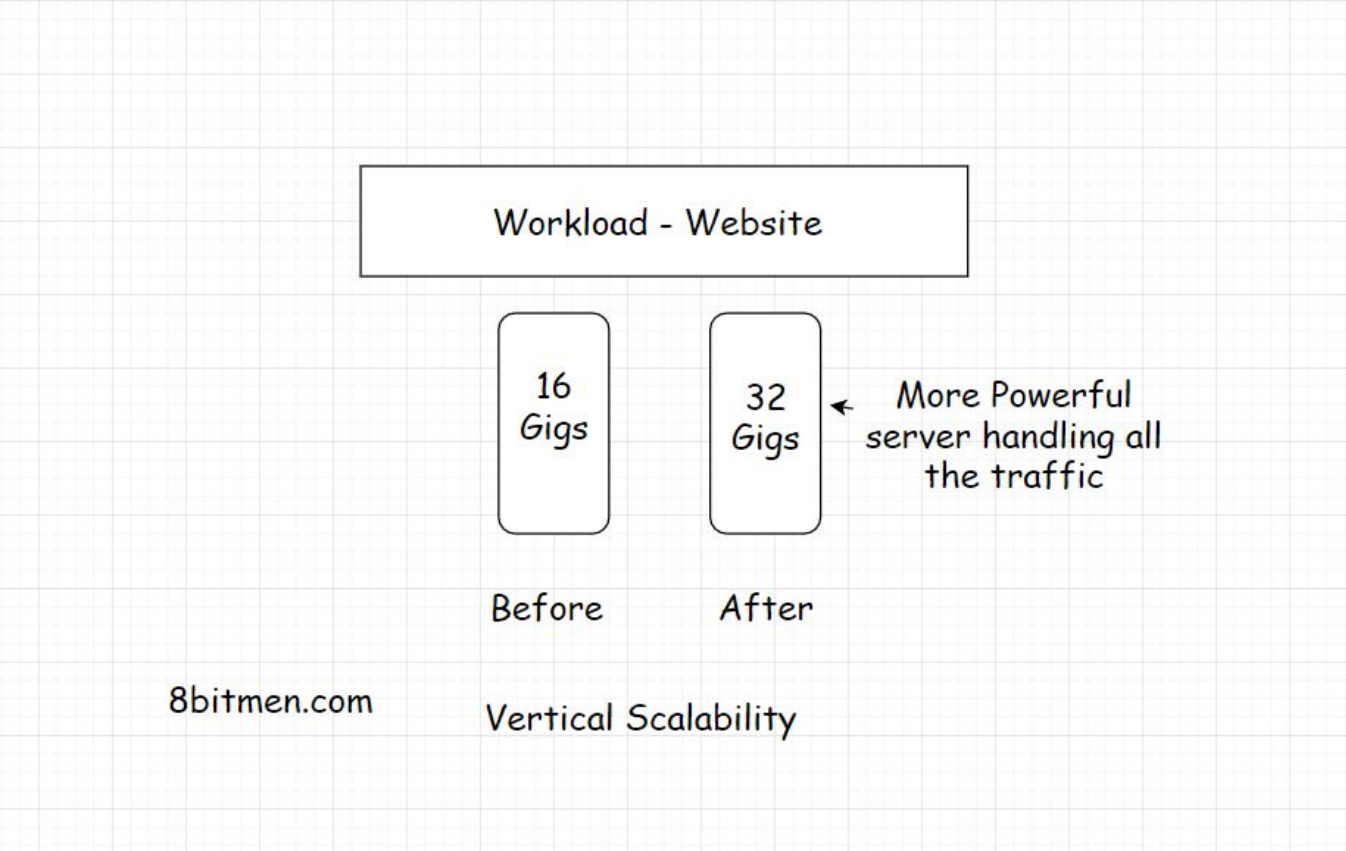
An application to scale well needs solid computing power. The servers should be powerful enough to handle increased traffic loads.

There are two ways to scale an application:

1. Vertical Scaling
2. Horizontal Scaling

What is Vertical Scaling?

Vertical scaling means adding more power to your server. Let’s say your app is hosted by a server with 16 Gigs of RAM. To handle the increased load you increase the RAM to 32 Gigs. You have vertically scaled the server.



Ideally, when the traffic starts to build upon your app the first step should be to scale vertically. Vertical scaling is also called *scaling up*.

In this type of scaling we increase the power of the hardware running the app. This is the simplest way to scale since it doesn't require any code refactoring, not making any complex configurations and stuff. I'll discuss further down the lesson, why code refactoring is required when we horizontally scale the app.

But there is only so much we can do when scaling vertically. There is a limit to the capacity we can augment for a single server.

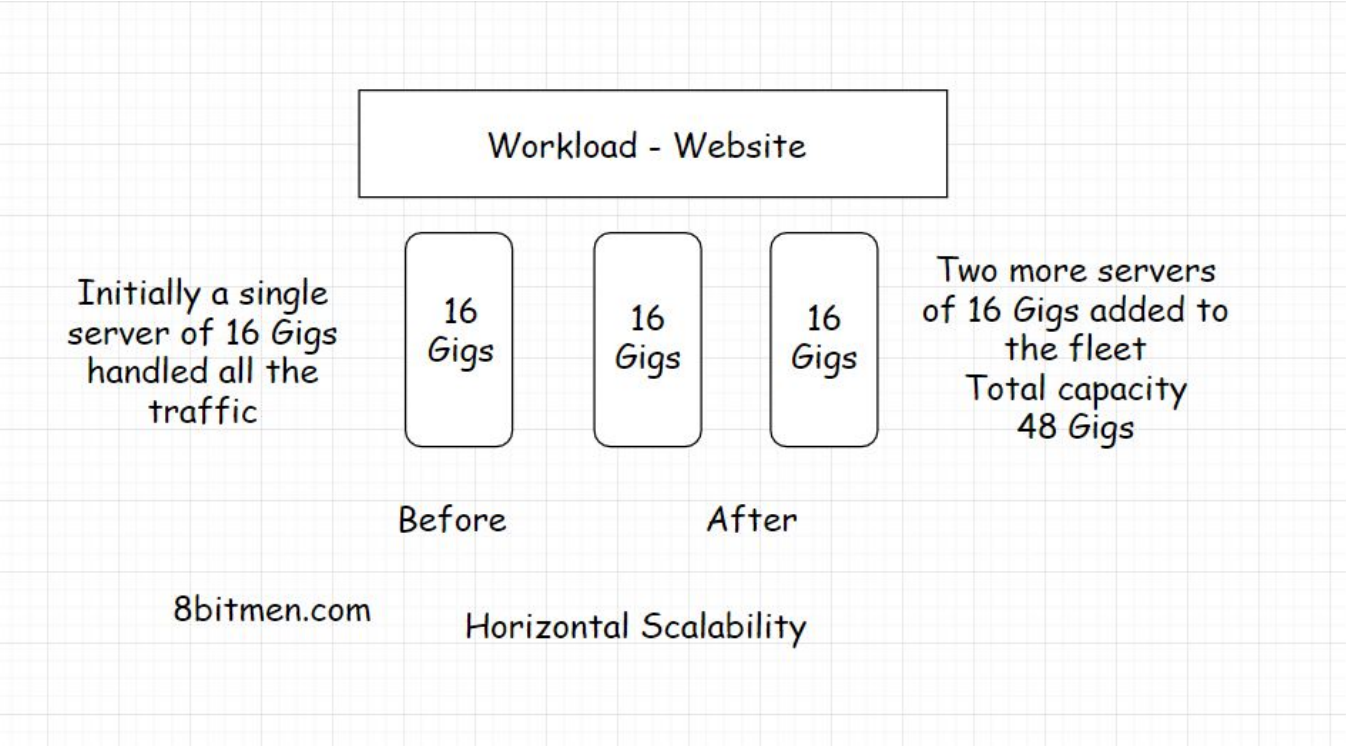
A good analogy would be to think of a multi-story building we can keep adding floors to it but only upto a certain point. What if the number of people in need of a flat keeps rising? We can't scale up the building to the moon, for obvious reasons.

Now is the time to build more buildings. This is where *Horizontal Scalability* comes in.

When the traffic is just too much to be handled by single hardware, we bring in more servers to work together.

What is Horizontal Scaling?

Horizontal scaling, also known as *scaling out*, means adding more hardware to the existing hardware resource pool. This increases the computational power of the system as a whole.



Now the increased traffic influx can be easily dealt with the increased computational capacity & there is literally no limit to how much we can scale horizontally assuming we have infinite resources. We can keep adding servers after servers, setting up data centres after data centres.

Horizontal scaling also provides us with the ability to dynamically scale in real-time as the traffic on our website increases & decreases over a period of time as opposed to vertical scaling which requires pre-planning & a stipulated time to be pulled off.

Cloud Elasticity

The biggest reason why *cloud computing* got so popular in the industry is the ability to scale up & down dynamically. The ability to use & pay only for the resources required by the website became a trend for obvious reasons.

If the site has a heavy traffic influx more server nodes get added & when it doesn't the dynamically added nodes are removed.

This approach saves businesses bags of money every single day. The approach

is also known as *cloud elasticity*. It indicates the stretching & returning to the original infrastructural computational capacity.

Having multiple server nodes on the backend also helps with the website staying alive online all the time even if a few server nodes crash. This is known as *High Availability*. We'll get to that in the upcoming lessons.