Scalability

What is Scalability ?

Scalability is the measure of a system’s ability to increase or decrease in performance and cost in response to changes in application and system processing demands.Examples would include how well a hardware system performs when the number of users is increased, how well a database withstands growing numbers of queries, or how well an operating system performs on different classes of hardware

What Is Horizontal Scaling?

Horizontal scaling (aka scaling out) refers to adding additional [nodes](https://medium.com/coinmonks/what-are-nodes-nodes-in-a-nutshell-f5d567bc9662) or machines to your [infrastructure](https://www.cloudzero.com/blog/cloud-infrastructure) to cope with new demands. If you are hosting an application on a server and find that it no longer has the capacity or capabilities to handle traffic, adding a server may be your solution.

It is quite similar to delegating workload among several employees instead of one. However, the downside of this may be the added complexity of your operation. You must decide which machine does what and how your new machines work with your old machines.

You can consider this the opposite of vertical scaling.

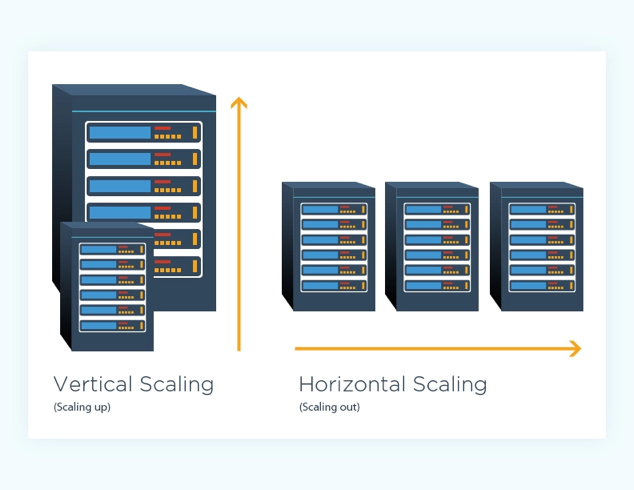
What Is Vertical Scaling?

Vertical scaling (aka scaling up) describes adding additional resources to a system so that it meets demand. How is this different from horizontal scaling?

While horizontal scaling refers to adding additional nodes, vertical scaling describes adding more power to your current machines. For instance, if your server requires more processing power, vertical scaling would mean upgrading the CPUs. You can also vertically scale the memory, storage, or network speed.

Additionally, vertical scaling may also describe replacing a server entirely or moving a server’s workload to an upgraded one.

Horizontal Vs. Vertical Scaling



Once again, the biggest central functional difference between the two is that horizontal scaling often forces you to rework how you implement your services or layers. For instance, let’s look at simple [three-tier architecture](https://www.techopedia.com/definition/24649/three-tier-architecture).

You have your presentation tier (user interface/client), logic tier (virtual server/services), and data tier (storage/databases). In the case of horizontal scaling, you can delegate each tier (or the functions responsible for them) to a different node.

However, you may already be running these tiers on different servers but find that one of these servers is underperforming or no longer meets demands. Once again, you can choose to scale this server vertically or horizontally. You may upgrade it with more resources or add another server to share the workload.

For further illustration, let’s consider databases. If you host your database on a single dedicated server and it gets too large, horizontal scaling would mean adding a new node, partitioning, and sharing the data between the old server and the new.

In our “lifting weights” analogy, horizontal scaling would mean buying new clothes while vertical scaling would be modifying your old clothes to handle your new gains.

With that being said, let’s look at a simple breakdown of the advantages and disadvantages of vertical and horizontal scaling.

Advantages of horizontal scaling

**Scaling is easier from a hardware perspective** - All horizontal scaling requires you to do is add additional machines to your current pool. It eliminates the need to analyze which system specifications you need to upgrade.

**Fewer periods of downtime** - Because you’re adding a machine, you don’t have to switch the old machine off while scaling. If done effectively, there may never be a need for downtime and clients are less likely to be impacted.

**Increased resilience and fault tolerance** - Relying on a single node for all your data and operations puts you at a high risk of losing it all when it fails. Distributing it among several nodes saves you from losing it all.

**Increased performance** - If you are using horizontal scaling to manage your network traffic, it allows for more endpoints for connections, considering that the load will be delegated among multiple machines.

Disadvantages of horizontal scaling

**Increased complexity of maintenance and operation** - Multiple servers are harder to maintain than a single server is. Additionally, you will need to add software for [load balancing](https://kemptechnologies.com/what-is-load-balancing/) and possibly [virtualization](https://searchservervirtualization.techtarget.com/definition/virtualization). Backing up your machines may also become a little more complex. You will need to ensure that nodes synchronize and communicate effectively.

**Increased Initial costs**- Adding new servers is far more expensive than upgrading old ones.

Advantages of vertical scaling

**Cost-effective** - Upgrading a pre-existing server costs less than purchasing a new one. Additionally, you are less likely to add new backup and virtualization software when scaling vertically. Maintenance costs may potentially remain the same too.

**Less complex process communication** - When a single node handles all the layers of your services, it will not have to synchronize and communicate with other machines to work. This may result in faster responses.

**Less complicated maintenance** - Not only is maintenance cheaper but it is less complex because of the number of nodes you will need to manage.

**Less need for software changes** - You are less likely to change how the software on a server works or how it is implemented.

Disadvantages of vertical scaling

**Higher possibility for downtime** - Unless you have a backup server that can handle operations and requests, you will need some considerable downtime to upgrade your machine.

**Single point of failure** - Having all your operations on a single server increases the risk of losing all your data if a hardware or software failure was to occur.

**Upgrade limitations** - There is a limitation to how much you can upgrade a machine. Every machine has its threshold for RAM, storage, and processing power.

Which Should You Choose And When?

Both horizontal and vertical scaling have their own benefits and limitations. Since there isn’t a one-size-fits-all solution for organizations, you need to scale according to your needs and resources. Here are a few factors to consider along with which type of scaling suits the situation best:

**Cost**- Initial hardware costs for horizontal upgrades are higher. If you are working on a tight budget and need to add more resources to your infrastructure quickly and cheaply, then vertical scaling may be the best option for you.

**Future-proofing** - Adding additional updated machines through horizontal scaling will increase the overall performance threshold of your organization. There is a limit to how much you can vertically scale a single node and it may not be able to handle the demands of the future.

**Topographic distribution**- If you plan to have nationwide or global clients, it is unreasonable to expect them all to access your services from a single machine in a single location. In a situation like this, you’ll need to horizontally scale your resources to maintain your [service level agreement](https://www.cio.com/article/2438284/outsourcing-sla-definitions-and-solutions.html) (SLA).

**Reliability** - Horizontal scaling may offer you a more reliable system. It increases redundancy and ensures that you are not relying on a single machine. If one machine fails, another may be able to pick up the slack temporarily.

**Upgradability and flexibility** - If you are running your application’s tiers on individual machines, they are easier to decouple and upgrade without any downtime.

**Performance and complexity** - Performance will depend on how your services work and how they are interconnected. Simple straightforward applications won’t benefit much from being run on multiple machines. In fact, it may degrade its quality. Sometimes it’s better to leave the application as is and upgrade the hardware to meet demand. Horizontal scaling may require you to rewrite the code or add a virtual machine that unifies all the servers.

On-Premise Vs. Cloud Scaling

For the majority of this guide, we’ve chosen to keep things simple by using on-premise non-cloud scaling for our examples.  However, cloud scaling works much the same.

A cloud service provider (CSP) may implement [hyper-converged infrastructure](https://www.nutanix.com/hyperconverged-infrastructure)-based horizontal scaling or choose to use virtual distributed services.

The former is quite common among private and hybrid cloud solutions. In most cases, your cloud provider will handle the scaling. This means you or your IT management won’t have to worry as much about what new hardware is required to meet new demands.

Service providers such as Azure and AWS have automatic scaling.

They can increase and decrease resources according to your requirements at any given time. They can scale up or out when traffic to your application is at its peak and scale down when demand is lessened. This provides organizations with more efficient and cost-effective scaling. This is another reason to consider [cloud migration](https://www.cloudzero.com/blog/cloud-migration).

Cost: The Grand Determinant

Despite your aspirations or organization’s needs, what may determine your decision, in the end, is cost. While horizontal scaling sounds great from a functional standpoint, you may not be able to afford it (right now). Nevertheless, it is still important to note that on-premise vertical and horizontal scaling may not be the only options available to you.

You can integrate both or migrate your organization’s infrastructure to a cloud service provider and allow them to handle scaling for you. The latter may be more financially and pragmatically feasible for you, especially in the long run.