missioncloud.com

Horizontal Vs. Vertical Scaling: Which Is Right For Your App?

By

7-8 minutes

Think About Long-Term Viability

When demand for your application is soaring, you'll quickly recognize the need to maintain an app's accessibility, uptime, and capacity in the face of increased load. Do you scale up or scale out?

In other words, **is horizontal scaling or vertical scaling the correct strategy for your business?**

The difference between these two types of scaling comes from the way that computing resources are added to your infrastructure. With vertical scaling ("scaling up"), you're adding more compute power to your existing instances/nodes. In horizontal scaling ("scaling out"), you get the additional capacity in a system by adding more instances to your environment, sharing the processing and memory workload across multiple devices.

A useful analogy for understanding this distinction is to think about scaling as if it were upgrading your car. Vertical scaling is like retiring your Toyota and buying a Ferrari when you need more horsepower. With your super-fast car, you can zoom around at high speed with the windows down and look amazing. But, while Ferraris are awesome, they're not very practical- they're expensive,

and at the end of the day, they can only take you so far before they're out of gas (not to mention, only two seats!).

Horizontal scaling works similarly in that it gets you that added horsepower, but it doesn't mean ditching the Toyota for the Ferrari. Instead, it's like adding another vehicle to a fleet. You can think of horizontal scaling like several vehicles you can drive all at once. Maybe none of these machines is a Ferrari, but no *one* of them needs to be: across the fleet, you have all the horsepower you need.

Why Scaling Out Is Better Than Up

When you're choosing between horizontal scaling and vertical scaling, you also have to consider what's at stake when you scale up versus scale out.

In the Toyota-for-Ferrari trade-in scenario, you're replacing a slower server with a bigger, faster one.

When you do this, though, you're throttling yourself while the machine is taken offline for the upgrade. And, what happens down the road when your traffic is on the rise again and you have to repeat the upgrades? There are only a finite number of times you can go about solving your problem by "scaling up" in this manner.

Horizontal scaling is almost always more desirable than vertical scaling because you don't get caught in a resource deficit. Instead of taking your server offline while you're scaling up to a better one, horizontal scaling lets you keep your existing pool of computing resources online while adding more to what you already have. **When your app is scaled horizontally, you have the benefit of elasticity.**

You can do exactly this when your infrastructure is hosted in a Managed Cloud environment. When you scale out to the cloud, you enjoy more options for building and deploying apps.

Amazon Elastic Compute Cloud (EC2), for example, acts as a virtual server with unlimited capacity. You choose the processor, storage capacity, networking options, and operating system you need and adjust your capacity as your computing needs grow.

When you containerize your apps, you can use Amazon Elastic Container Service (ECS) or Amazon Elastic Kubernetes Service (EKS). You can use container orchestration services to deploy, manage, and scale your apps. These services automate node provisioning, patching, and updating so you can focus on other aspects of your application.

AWS Lambda's serverless functions also provide flexibility as you scale out. These functions enable you to run your code without provisioning or managing servers and automatically scale computing power as needed.

If you develop your applications using GraphQL application programming interfaces (APIs), AWS AppSync connects to Lambda and other data sources. AppSync automatically scales up and down depending on request volumes.

Scaling out to the cloud enables you to combine any and all of these cloud services, and more, with flexibility to meet your changing app configuration and use.

Other benefits of scaling out in a cloud environment include:

- Instant and continuous availability
- No limit to hardware capacity
- Cost can be tied to use
- You're not stuck always paying for peak demand
- Built-in redundancy
- Easy to size and resize properly to your needs

How To Achieve Effective Horizontal Scaling

There are important best practices to keep in mind to make your service offering compatible with horizontal scaling.

The first is to **make your application stateless on the server side as much as possible**. Any time your application has to rely on server-side tracking of what it's doing at a given moment, that user session is inextricably tied to that particular server. If, on the other hand, all session-related specifics are stored browser-side, that session can be passed seamlessly across literally hundreds of servers. The ability to hand a single session (or thousands or millions of single sessions) across servers interchangeably is the very epitome of horizontal scaling.

The second goal to keep square in your sights is to develop your app with a service-oriented architecture. **The more your app is comprised of self-contained but interacting logical blocks, the more you'll be able to scale each of those blocks independently as your use load demands.** Be sure to develop your app with independent web, application, caching and database tiers. This is critical for realizing cost savings — because without this microservice architecture, you're going to have to scale up each component of your app to the demand levels of services tiers

getting hit the hardest.

Enjoy the Best of Both Solutions

Scaling up versus scaling out is not necessarily an either/or choice. While you're splitting your monolithic apps into microservices, you can scale up, too, to handle increased load in the meantime.

Start by splitting out the parts of your app with the highest load. That way, you can scale out those microservices as needed.

When you approach your scaling this way, what's left of the original app won't need to scale up as far. You can gradually complete the transition from monolithic app to microservices while still scaling up if demand spikes.

Your Cloud-Friendly, Autoscaling App and DevOps

You've worked hard developing your app. Now it's time to get it on the market and ready to handle staggering growth. A Managed AWS Cloud service and a team of highly skilled cloud architects who can implement DevOps automation is *the* most effective way to ensure your app scales to success.

Get started on solving your scaling challenges by scheduling a 30-minute SA On-Demand where you can talk to one of our engineers about the steps you need to take to get ready with autoscaling!