



Article

Horizontal scaling vs vertical scaling: Choosing your strategy



By [Sujatha R](#)
Technical Writer

Published: February 1, 2024 • 9 min read

← Back to All Articles

Table of contents

What is horizontal scaling?

What is vertical scaling?

Horizontal scaling vs vertical scaling

10 factors for choosing between horizontal vs vertical scaling

Horizontal Scaling vs Vertical Scaling FAQs

Scale confidently with DigitalOcean

The narrative is shifting from “Should we embrace the cloud?” to “How can we thrive in the cloud-centric future?” As [Gartner](#) predicts, global spending on public cloud services will hit \$679 billion by 2024. As investments in cloud services soar, the ability to effectively manage [cloud scalability](#) is key for businesses aiming to succeed.

When it comes to cloud scalability, there are two main strategies: horizontal scaling vs vertical scaling. Choosing the right scaling method requires evaluating factors like application architecture, traffic patterns, resource utilization needs, and budget constraints. This article explains horizontal and vertical scaling in-depth, including real-world examples, so you can determine the best fit for your business needs as you scale your operations in the cloud.

Key takeaways:

- Horizontal scaling (scaling out) means adding more machines or nodes to distribute load (for example, adding more servers to handle increased web traffic), whereas vertical scaling (scaling up) means increasing the resources (CPU, RAM, etc.) of an existing machine to handle more load.
- Horizontal scaling offers greater long-term scalability and fault tolerance (since if one node fails, others still serve the system), and is a foundation of cloud-native design, while vertical scaling is limited by the maximum hardware capacity and can lead to downtime during upgrades.
- The choice of strategy depends on the application and growth pattern: vertical scaling is simpler and useful for quick, moderate growth, but horizontal scaling is usually preferred for building highly available systems and handling massive or unpredictable demand by simply adding more commodity servers.

What is horizontal scaling?

Horizontal scaling is the process of adding more machines or nodes to a resource pool in a system to distribute the workload. This approach is synonymous with scalability through replication. In simpler terms, if your organization scales horizontally, it means adding more servers to your existing infrastructure to handle an increased load.

Horizontal scaling is like adding more lanes to a highway to accommodate more traffic. Each lane represents a server and the overall capacity increases as more lanes are added.

What is vertical scaling?

Vertical scaling is the process of increasing the power of individual machines in the system. Instead of adding more servers, organizations employing vertical scaling

Share



Cloud Education

Try DigitalOcean for free

Click below to sign up and get \$200 of credit to try our products over 60 days!

[Sign up →](#)

improve the capabilities of existing servers. Vertical scaling upgrades involve upgrading hardware components, such as CPU, RAM, and network speed, adding more resources or storage, to handle larger workloads.

Vertical scaling is akin to expanding the capacity of a single bridge to accommodate more traffic. By making the bridge stronger and wider, it can handle increased load without the need for additional lanes.

Quick fact: 🌟 Netflix uses both horizontal and vertical scaling, showcasing the effectiveness of a hybrid approach in handling global content distribution and massive datasets.

Horizontal scaling vs vertical scaling

Horizontal scaling and vertical scaling are key strategies for improving system capacity and performance. Here's how these approaches align and differ in how they manage increases in workload and resources:

Benefits of horizontal scaling

- **Enhanced load distribution:** Horizontal scaling distributes workload across multiple servers or nodes, reducing the burden on individual machines. This ensures efficient utilization of resources and prevents bottlenecks, leading to improved system performance and reliability.
- **On-demand scalability:** Horizontal scaling allows businesses to easily accommodate increasing workload demands by adding more servers to their resource pool. This scalability on-demand ensures flexibility and agility in scaling infrastructure to meet changing business needs without significant upfront investment.
- **Fault tolerance:** By distributing workload across multiple servers, horizontal scaling improves fault tolerance and system resilience. In the case of hardware failures or server downtime, redundant servers can seamlessly take over, ensuring uninterrupted service availability and minimal impact on operations.

Challenges of horizontal scaling

- **Complexity of load balancing:** Horizontal scaling introduces the complexity of [load balancing](#), as it requires efficient distribution of incoming traffic across multiple servers. Implementing and managing effective load balancing mechanisms, such as elastic load balancers, adds overhead and complexity to the infrastructure.
- **Data consistency concerns:** Maintaining data consistency and synchronization across distributed servers can be challenging in a horizontally scaled environment. As data is spread across multiple nodes, ensuring consistency and coherence becomes crucial, requiring robust data replication and synchronization strategies.
- **Scalability limits:** Despite its scalability benefits, horizontal scaling has inherent scalability limits determined by factors such as network bandwidth, communication overhead, and synchronization latency. Scaling beyond these limits may lead to diminishing returns or performance degradation, necessitating careful [cloud capacity planning](#) and optimization.

Horizontal scaling examples

- **Load balancers:** Load Balancers scale horizontally by distributing incoming application traffic across multiple instances. As demand increases, new instances can be added to the pool, and the load balancer distributes the load and processing power among several servers in a system. This ensures a balanced distribution of traffic to multiple machines.
- **Kubernetes:** Kubernetes is a container orchestration platform that facilitates horizontal scaling. It can dynamically adjust the number of running instances based on resource utilization and demand, ensuring optimal performance.

Benefits of vertical scaling

- **Increased processing power:** Vertical scaling improves the processing power and computing capacity of individual servers by upgrading hardware components such as CPU, memory, and storage. This enables servers to handle larger workloads and resource-intensive applications more efficiently.
- **Simplified resource management:** Vertical scaling simplifies resource management by consolidating workload on a single, powerful server. This leads to less complexity in infrastructure management and eliminates the need for load balancing and distributed data management, streamlining operational processes.

- **Cost-efficient for specific workloads:** Vertical scaling can be cost-effective for specific workloads or applications that require high-performance computing or specialized hardware configurations. By investing in vertical scalability, businesses can optimize resource utilization and achieve better cost-efficiency for targeted use cases.

Challenges of vertical scaling

- **Limited scalability:** Vertical scaling has inherent scalability limits determined by the maximum capacity of individual servers. Scaling beyond these limits may require costly hardware upgrades or replacement, leading to potential downtime and operational disruptions.
- **Single point of failure:** Vertical scaling increases the risk of a single point of failure, as all workload is concentrated on a single server. Hardware failures or system crashes can have a significant impact on service availability and may result in downtime until the issue is resolved.
- **Complexity of upgrades:** Upgrading hardware components in vertically scaled environments can be costly and complex, requiring downtime for server maintenance and upgrades. Managing hardware compatibility and ensuring seamless migration of workloads during upgrades adds to the operational overhead and complexity of vertical scaling.

Vertical scaling examples

- **Virtual Machines (VMs):** VMs scale vertically by allowing users to resize the virtual machine to a higher or lower unit. This involves changing the size of the VM to accommodate varying performance and resource requirements such as storage capacity, networking options, etc.,
- **Compute services:** Cloud computing services across various providers enable vertical scaling by allowing users to resize instances with different configurations and performance levels. This includes adding more CPU cores, memory, or storage capacity to accommodate increased workload requirements. Users can vertically scale their compute resources as needed to maintain performance levels and meet business demands.

10 factors for choosing between horizontal vs vertical scaling

Horizontal scaling is particularly effective for applications that can be easily distributed across multiple servers, such as web applications, content delivery networks (CDNs), and databases. On the other hand, the vertical scaling approach is suitable for applications that may not easily lend themselves to distribution across multiple servers, but require more processing power, such as certain legacy systems or resource-intensive databases.

You can also combine both of the strategies if that suits your business. We have distilled down a few parameters below, which will help you easily decide which scaling strategy suits your business the best.

1. Traffic patterns

Observe how demand fluctuates. Based on how the demand fluctuates in your business, there can be unpredictable spikes or dips in activity. In such a case, you may favor horizontal scaling. It adds or removes computing resources, adapting to varying traffic.

2. Resource efficiency

Assess resource optimization needs. Horizontal scaling keeps the existing pool of computing resources online and adds more to what is already present. For consistent, high-resource demand, vertical scaling may be efficient. It boosts existing capabilities without the need for added infrastructure.

3. Cost considerations

Evaluate budget flexibility. Considering the cost perspective, vertical scaling may offer budget stability but can be pricier for frequent adjustments. Horizontal scaling might offer a high-cost set-up environment but might be cost-effective in the long run.

4. Application architecture

Examine system design and dependencies. If your applications are designed to run on multiple servers, lean towards horizontal scaling. For applications reliant on a single robust server, vertical scaling may be more suitable.

5. Downtime tolerance

Determine tolerance for system interruptions. Horizontal scaling often facilitates less downtime, making it suitable for businesses requiring high availability. Vertical scaling may involve brief downtimes during upgrades.

6. Future growth forecast

Anticipate long-term scalability needs. If your business expects consistent and substantial growth, a combination of horizontal and vertical scaling (hybrid approach) may provide the necessary flexibility.

7. Cloud service integration

Consider third-party cloud services. Horizontal scaling aligns well, utilizing the distributed nature of cloud infrastructure. Vertical scaling can simplify operations when using a single robust cloud server.

8. Operational simplicity

Assess ease of management. Horizontal scaling is often simpler to manage, especially for distributed systems. Vertical scaling may involve more complex adjustments to a single server.

9. Application performance

Prioritize consistent performance. For applications where consistent performance is critical, vertical scaling ensures dedicated resources, improving overall system performance.

10. Workload distribution

Examine the nature of workloads. Horizontal scaling excels in distributing workloads across multiple nodes. Vertical scaling suits scenarios where a single machine can handle the entire workload efficiently.

Horizontal Scaling vs Vertical Scaling FAQs

What is the main difference between horizontal and vertical scaling?

Horizontal scaling (scaling out) involves adding more servers to distribute the workload across multiple machines. Vertical scaling (scaling up) involves increasing the resources like CPU, RAM, or storage on existing servers to handle more load.

When should you choose horizontal scaling over vertical scaling?

Horizontal scaling is ideal when you need to handle unpredictable traffic spikes and want fault tolerance across multiple servers. It's also better for applications that can be distributed across multiple nodes and when you want to avoid single points of failure.

What are the cost implications of each scaling approach?

Vertical scaling can be initially cheaper and simpler to implement but has hard limits and can become expensive at higher tiers. Horizontal scaling may require more upfront investment in architecture design but offers better long-term cost efficiency and virtually unlimited scaling potential.

Which scaling method is easier to implement for beginners?

Vertical scaling is generally easier for beginners as it requires minimal changes to application architecture. Horizontal scaling requires more complex application design considerations like load balancing, data synchronization, and stateless application patterns.

Scale confidently with DigitalOcean

DigitalOcean offers a suite of cloud infrastructure [products](#) designed to simplify cloud scaling, empowering developers and businesses to grow without complexity. The platform makes both horizontal and vertical scaling achievable through offerings tailored to accommodate fluctuating workloads.

Here's how you can harness DigitalOcean's products to achieve cloud scalability:

- [Droplets](#): You can seamlessly implement vertical scaling through Droplets. They are DigitalOcean's Linux-based virtual machines that run on top of virtualized hardware. They can be effortlessly customized to meet evolving requirements. With diverse performance types, including Standard, General Purpose, CPU-

Optimized, or Memory-Optimized Droplets, you can align your computing resources precisely with your application's demands.

- [Kubernetes](#): For the best horizontal scaling, we have DigitalOcean Kubernetes (DOKS). It streamlines container orchestration with a managed service that simplifies deployment, management, and scaling of containerized applications using the open-source Kubernetes platform, devoid of Kubernetes management complexities.
- [Spaces](#): Catering to vast amounts of unstructured data such as videos, images, and backups, Spaces offer scalable storage sans the intricacies of traditional storage system management, ensuring seamless scalability.
- [Volumes](#): DigitalOcean's block storage solution, Volumes, empowers you to scale your storage effortlessly by attaching additional storage to Droplets, with the flexibility to resize as necessary, ensuring storage scales alongside your application.
- [App Platform](#): As a scalable [platform-as-a-service \(PaaS\)](#) solution, App Platform automates infrastructure management tasks like updates, scaling, and security, supporting multiple programming languages and frameworks for quick app deployment and scaling.
- [Load Balancers](#): DigitalOcean's fully managed Load Balancers efficiently distribute traffic across your infrastructure, adeptly handling traffic spikes, and optimizing performance and computing power to uphold application availability and performance amid increasing demand.

Each of these products seamlessly integrates within DigitalOcean's ecosystem, delivering a unified experience that simplifies the scalability journey. With transparent [pricing](#) models and robust [community](#) support, DigitalOcean fosters an environment conducive to growth-focused businesses.

Seeking stress-free cloud scalability? DigitalOcean has your back—[experience effortless scaling today](#) and establish a robust cloud infrastructure that supports the ascent of your business.

About the author



Sujatha R Author
Technical Writer

[See author profile](#)

Sujatha R is a Technical Writer at DigitalOcean. She has over 10+ years of experience creating clear and engaging technical documentation, specializing in cloud computing, artificial intelligence, and machine learning. 🙌 She combines her technical expertise with a passion for technology that helps developers and tech enthusiasts uncover the cloud's complexity.

Related Resources



Articles

Cloud Migration Checklist: Your Pre- and Post-Migration Guide

[Read more →](#)



Articles

11 IT Cost Optimization Strategies for Scalable Savings in 2025

[Read more →](#)



Articles

What is Lift-and-Shift Migration? Your Fastest Route to the Cloud

[Read more →](#)

Get started for free

Sign up and get \$200 in credit for your first 60 days with DigitalOcean.*

[Get started](#)

*This promotional offer applies to new accounts only.

Company

About
Leadership
Blog
Careers
Customers
Partners
Referral Program
Affiliate Program
Press
Legal
Privacy Policy
Security
Investor Relations

Products

Overview
Droplets
Kubernetes
Functions
App Platform
Gradient™ AI GPU Droplets
Gradient™ AI Bare Metal GPUs
Gradient™ AI 1-Click Models
Gradient™ AI Platform
Load Balancers
Managed Databases
Spaces
Block Storage
API
Uptime
Identity and Access Management

Resources

Community Tutorials
Community Q&A
CSS-Tricks
Write for DOnations
Currents Research
Hatch Startup Program
Wavemakers Program
Compass Council
Open Source
Newsletter Signup
Marketplace
Pricing
Pricing Calculator
Documentation
Release Notes
Code of Conduct
Shop Swag

Solutions

Website Hosting
VPS Hosting
Web & Mobile Apps
Game Development
Streaming
VPN
SaaS Platforms
Cloud Hosting for Blockchain
Startup Resources
Migration Assistance

Contact

Support
Sales
Report Abuse
System Status
Share your ideas

