middleware

The Ultimate Guide to Auto Scaling Your Application Infrastructure

www.middleware.io

Table of contents

A handy overview of autoscaling to help organizations prioritize their business needs and meet growing traffic. The what, why, and how explained by our subject-matter experts. Read on to bust some myths and gain deep insights into this (autoslcaling your application infrastructure) overarching process.

Scaling the application infrastructure based on changing demand is essential. Many computing engines allow the infrastructure to be autoscaled to save costs. For this, you need to devise ways to help you autoscale, especially if you don't have the necessary resources.

To prepare for high website traffic, scaling up is a sensible first step. You risk downtime on your busiest, most business-critical sales if you ignore the complexities of autoscaling. Let us help you learn more about this trending process, the types, and the various challenges and benefits.

What is autoscaling?

Automatically adjusting a system's processing power to the current resource load is known as autoscaling.

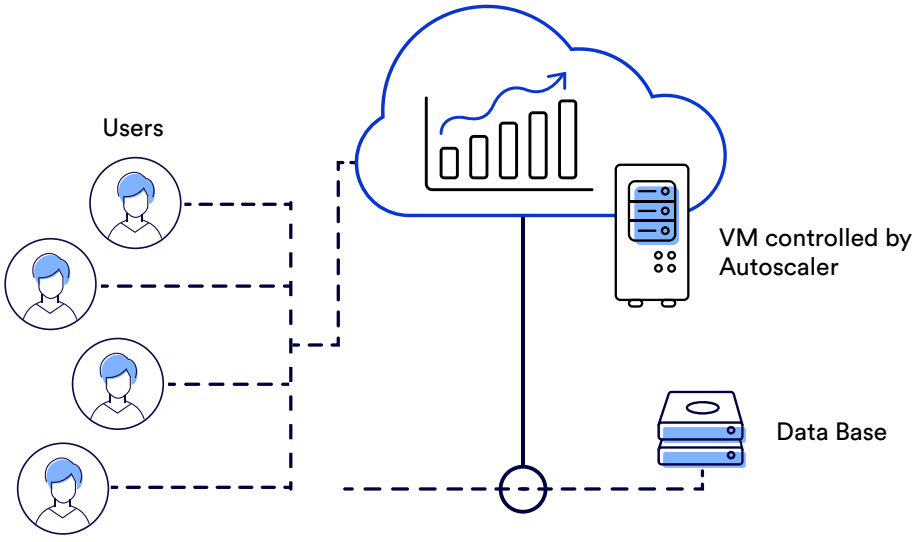
To serve more visitors, web applications require additional computing power from the servers. This power is used to scale an application's or system's components, allowing it to expand while managing additional traffic. More visitors to your website mean more demand on your servers. Autoscaling horizontally scales up servers when demand is high and scales them down when demand is low.

Why you should autoscale your infrastructure

Each website has a physical server or group of servers, known as a server farm. When visitors open your website in their browsers, servers provide computational power to provide search data to these visitors.

As the number of visitors grows, your website sees a surge in traffic that your servers may or may not handle well. Manually scaling them is a huge task. This is why you need autoscaling. Autoscaling automatically expands your server's ability to handle the surge in traffic and prevents it from being delayed or completely down.

How autoscaling works



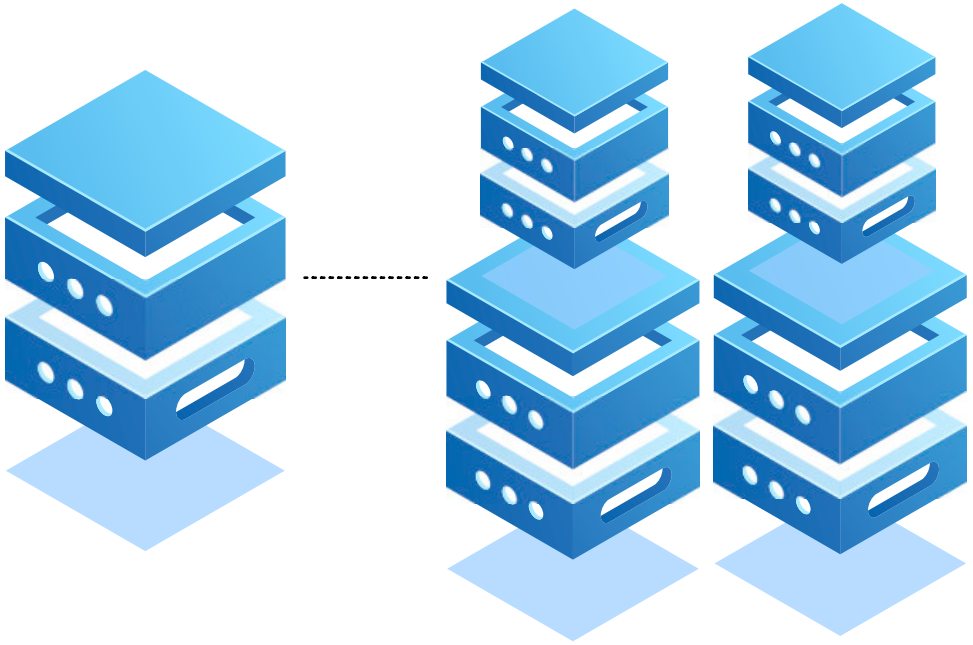
Users need to limit the number of resources (CPU cores and RAM) to deploy their workloads across many public and private cloud systems. At the same time, human operators demand a higher limit than the job requires, risking system failure and delaying or dropping end-user requests. This leads to a substantial collective waste of resources at scale. As a result, many tech giants have developed automated solutions to autoscale the systems and avoid all these scaling issues.

Google, for example, uses Autopilot to automatically configure resources, altering the number of concurrent processes in a job (horizontal scaling) and changing the CPU/memory limits for individual jobs (vertical scaling). Middleware is a similar solution that reduces slack and improves scalability.

Middleware (just like human operators) mitigates slack – the difference between the limit and actual resource utilization while limiting the likelihood that a job will be killed by an out-of-memory (OOM) error or performance degradation due to CPU throttling.

To walk the line, Middleware applies machine learning algorithms to historical data about previous job executions and a number of finely-tuned heuristics, enabling you to scale both vertically and horizontally as needed.

Horizontal scaling



Horizontal scaling, also known as scaling out, involves adding more nodes or machines to your infrastructure to meet the increased demand. If you're hosting an application on a server that no longer has the capacity or ability to handle traffic, adding another server may be the solution.

It's like delegating tasks to multiple employees, not just one. However, the added complexity of your operation is a disadvantage here. You need to figure out which machine does what and how your new machines interact with your old ones.

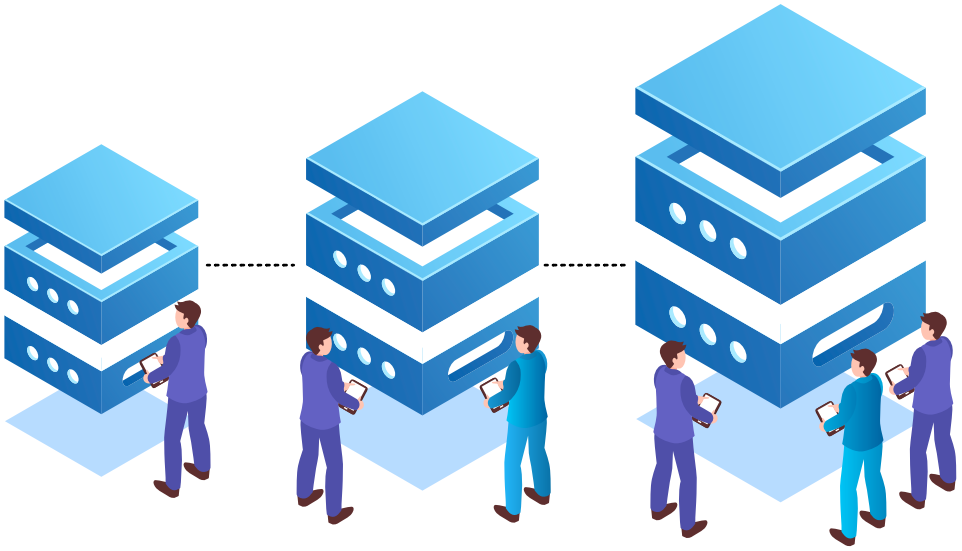
Advantages of horizontal scaling

- **Easy scaling :** From a hardware standpoint, scaling is simple. All you have to do is add more machines to your current pool to scale horizontally. This eliminates the need to find out which system specifications need upgrades.
- **Less downtime :** Because you're adding a machine, you don't need to turn off the old machine while scaling. If done correctly, you can reduce downtime and the resulting impact on clients.
- **Increased fault tolerance and resilience :** When you rely on a single node for all of your data and processes, you risk losing everything in the event of a failure. With horizontal scaling, you can avoid losing important data by distributing it over multiple nodes.
- **Increased performance :** Horizontal scaling allows you to manage network traffic and connect more endpoints because the load is distributed across multiple workstations.

Disadvantages of horizontal scaling

- **Increased maintenance and operational complexity :** Maintaining several servers is more difficult than maintaining a single server with horizontal scaling as you need additional load balancing or virtualization software. Backing up your equipment can also be more difficult. You need to make sure the nodes are synchronized and communicate properly.
- **Initial prices are higher :** Adding new servers is far more expensive than upgrading existing servers.

Vertical scaling



Vertical scaling, also known as scaling up, adds more resources to a system to meet demand. How is it different from horizontal scaling?

Horizontal scaling involves adding new nodes, and vertical scaling increases the power or performance of your current machines. For example, vertical scaling upgrades the CPUs if your server requires additional processing capacity. You may also scale memory, storage, and network speed vertically.

Vertical scaling can also entail completely replacing a server or shifting an existing server's workload to a newer one.

Advantages of vertical scaling

- **Upgrading an existing server is less expensive than purchasing a new one :** When scaling vertically, you hardly need to add new backup and virtualization software. The maintenance prices also remain the same.
- **Less complicated process communication :** A single node managing all the services' layers doesn't need to synchronize or connect with other machines to function. This leads to faster responses.

  Maintenance is less complicated : Because there are fewer nodes to handle maintenance is not only cheaper but also less complicated.
- **Software changes are less likely :** You're less likely to change how a server's software works or how it's implemented.

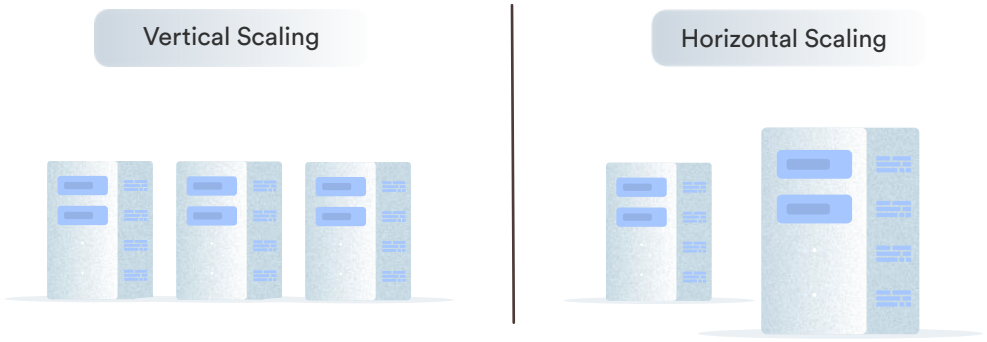
Disadvantages of vertical scaling

- **Higher risk of downtime :** Unless you have a backup server to manage operations and requests, upgrading your machine requires significant downtime.

- **Single point of failure :** If all your activities run on a single server, you can lose all your data in the event of a hardware or software failure.

- **Limitations on machine upgrades :** There is a limit to how much you can improve a machine. Every machine has a limit on RAM, storage, and computing power.

Horizontal vs. vertical scaling: which one should be used when?



Both horizontal and vertical scaling have advantages and disadvantages. You should scale based on your needs and resources because there is no one-size-fits-all solution here. Below are some factors to consider when deciding the type of scaling ideal for your situation:

1. Cost : Horizontal upgrades have higher initial hardware costs. Vertical scaling is a better solution if you're on a budget and need to add more resources to your infrastructure quickly and cheaply.

2. Future-proofing : By horizontally scaling additional updated machines, you can increase your organization's total performance threshold. On the other hand, there is a limit to how much a single node can vertically scale and may not satisfy future needs.

3. Topographic distribution: It's unrealistic to expect all of your clients to access your services from a single computer in a single area if you aim for a national or global customer base. To maintain your service level agreement (SLA), you need to scale your resources horizontally.

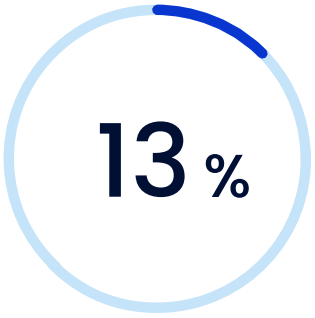
4. Reliability: Horizontal scaling provides you with a more reliable system. It adds redundancy to ensure that you don’t rely on a single machine for your scaling needs. If one machine fails, another can temporarily pick up the slack.

5. Upgradeability and flexibility: If your application's tiers run on separate machines, it's easier to decouple and update them without downtime.

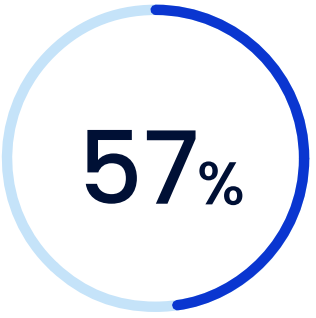
6. Performance and complexity: Horizontal scaling may need code rewriting or adding a virtual machine uniting all the servers. However, vertical scaling may not require any of these complex steps, making it a better choice in performance and ease of use.

Autoscaling issues in cloud environments

The state of DevOps 2021 report shows that cloud and automation correlate with higher DevOps evolution. Sixty-five percent of mid-evolution organizations report using the public cloud, and only 20% use the cloud to its full potential, compared to 57% of high-evolution organizations.



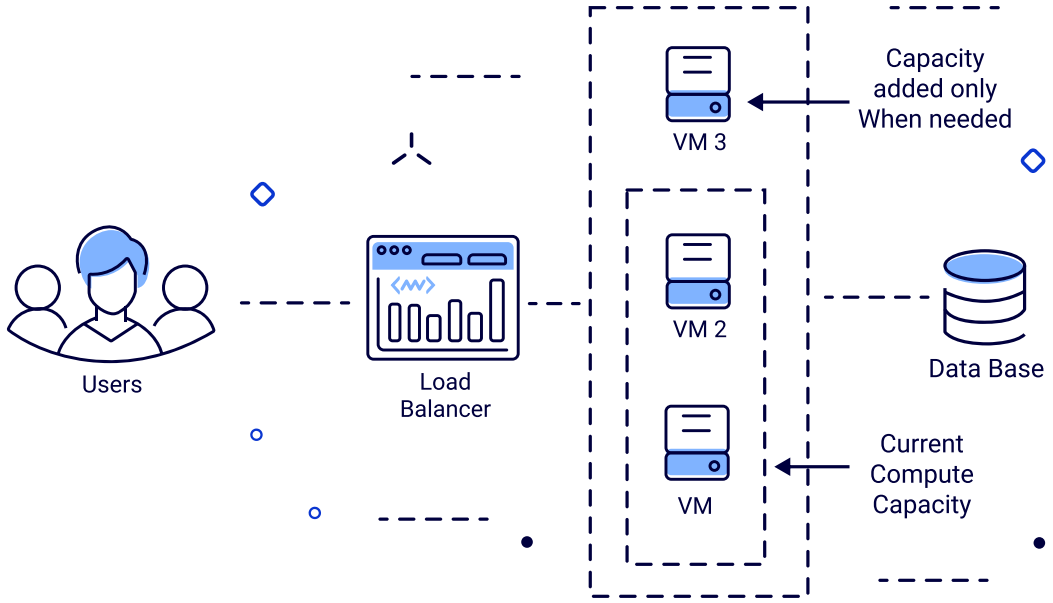
13% of mid-evolution organizations use a public cloud totheir fullest capacity.



57% of mid-evolution organizations concluded they use their public cloud to its fullest capacity.

Managing high availability even at extreme load conditions (Black Friday sales, marketing campaigns) and keeping an eye on the cost are prevalent roadblocks in cloud automation technology like autoscaling.

How Middleware helps overcome these autoscaling issues



Middleware Auto Scaler enables service discovery, AI load balancing across different services and instances, proxy servers for handling cross-network traffic, and standard protocols for infrastructural autoscaling. Some notable benefits of autoscaling with Middleware are:

1. High availability and scalability. : Middleware Auto Scaler uses AI to maintain high availability and scalability by monitoring the cloud infrastructure and floating resources in the background.

2. More cost savings : When resources are not in use, Middleware's Auto Scaler puts them to sleep, potentially saving up to 60% on costs without sacrificing performance.

3. Easy scaling : Middleware Auto Scaler allows you to scale resources horizontally and vertically to manage your application on the cloud and across geographic locations.

4. Faster deployment: The auto scaler comes in a plug-and-play package with a no-code infrastructure that can develop services once they're running on the cloud.

Middleware predictive autoscaling

Middleware is an end-to-end infrastructure solution that can manage everything from load balancing to service discovery and scaling. You just need to connect your microservice container to it, which then deploys a server in your data center depending on the data traffic.

The Middleware Auto Scaler automatically generates and deletes virtual machines based on the load on your application. It scans your infrastructure for patterns and prepares services in advance to ensure that your application is always up and running during peak hours. It also puts many of a machine's components to sleep during low traffic, saving you a lot of money on cloud costs.

Why should companies opt to autoscale?

Autoscaling allows you to keep track of your hardware costs by shrinking or expanding yourapplication infrastructure on demand. The cost savings come from using a cloud utility-based pricing approach, where you only pay for the resources you use.

Take a look at Amazon Web Service's pricing for the varying sizes of its t3a instances:

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
t3a.nano	2	Variable	0.5 GiB	EBS Only	$0.0047 per Hour
t3a.micro	2	Variable	1 GiB	EBS Only	$0.0094 per Hour
t3a.small	2	Variable	2 GiB	EBS Only	$0.0188 per Hour
t3a.medium	2	Variable	4 GiB	EBS Only	$0.0376 per Hour
t3a.large	2	Variable	8 GiB	EBS Only	$0.0752 per Hour
t3a.xlarge	4	Variable	16 GiB	EBS Only	$0.1504 per Hour
t3a.2xlarge	8	Variable	32 GiB	EBS Only	$0.3008 per Hour

As you can see, the price is largely proportional to the memory capacity of each instance.Using Middleware, you only need to turn it on. Our AI will automatically scale up and scale down for you.

Using a utility-based pricing approach is different from the traditional technique of deploying hardware on-premises to meet peak real-time consumption, which results in improper resource utilization and a lesser return on investment (ROI).

Autoscaling also ensures that your applications are always available and resilient. With this method, you can create a solution that quickly detects failed server instances and automatically replaces them with the healthy ones transparent to the application.

Before autoscaling your application, you should always weigh the cost of deploying such a solution against the expected savings. When deploying an autoscaling solution, consider the following costs:

Development work

Software licenses

Software maintenance

You also need to account for the solution's continued maintenance to keep up with the many changes in an application. It’s best to think about how much work it’ll take to implement an autoscaling solution with your application.

Some current apps are designed to loosely connect to make the autoscaling process easier. This can help reduce development time and the cost of implementing the solution.

As a result, traditional systems not built for dynamic reconfiguration would require additional development time. This may not be worth the cost of using the solution. Once you estimate the cost of deploying an autoscaling solution, you need to compare it with the anticipated savings to see if the cost is justified.

Is autoscaling complex?

One step you should take to prepare your website for higher demand is to scale up your servers. Once you reach a higher vertical scale level (upgrade to higher-end servers), prices skyrocket.

Horizontal scaling (connecting to many smaller servers) is more cost-effective in terms of hardware. Still, it is a challenge to integrate all of the servers into one cohesive system. This requires specialized personnel and is expensive to administer.

Scaling your website entails much more than simply expanding server capacity; you'll run into issues with search, concurrency, consistency, and performance.

Server scaling doesn't address website bottlenecks such as payment gateways. Building performance into a web application necessitates a comprehensive approach. Therefore, if you ignore the difficulty of scaling servers and websites, you run the risk of a highly impactful server failure.

When it comes to meeting the growing demand, scaling your website is imperative. But there are also major limitations that can make autoscaling complex and tedious.

Find out more about Middleware Auto Scaler to simplify all your autoscaling needs.