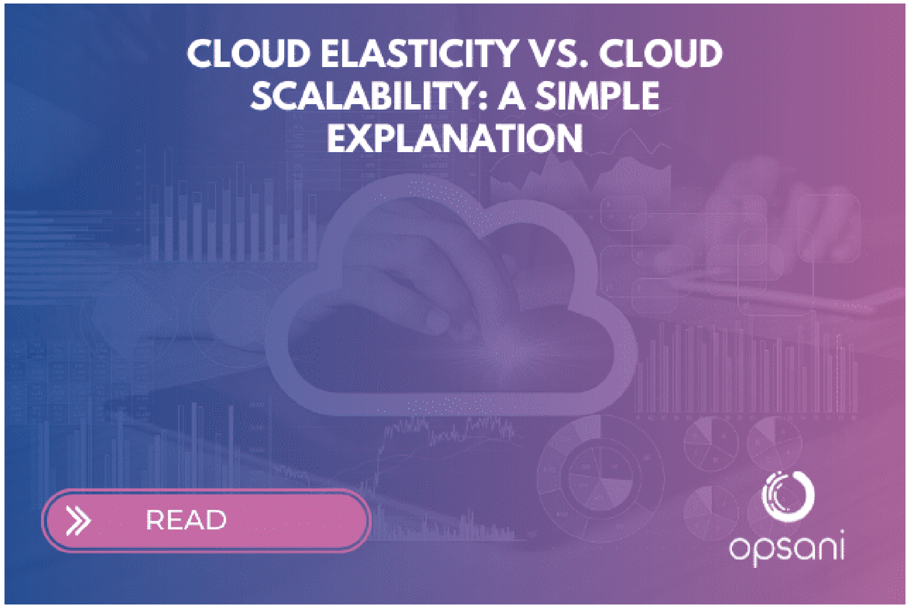
[A R T I C L E]

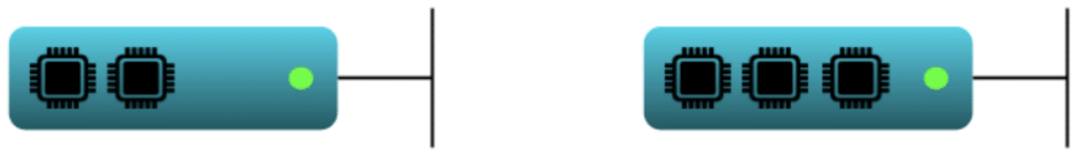
Cloud Elasticity vs. Cloud Scalability: A Simple Explanation


CLOUD ELASTICITY VS. CLOUD SCALABILITY: A SIMPLE EXPLANATION
READ
opsani

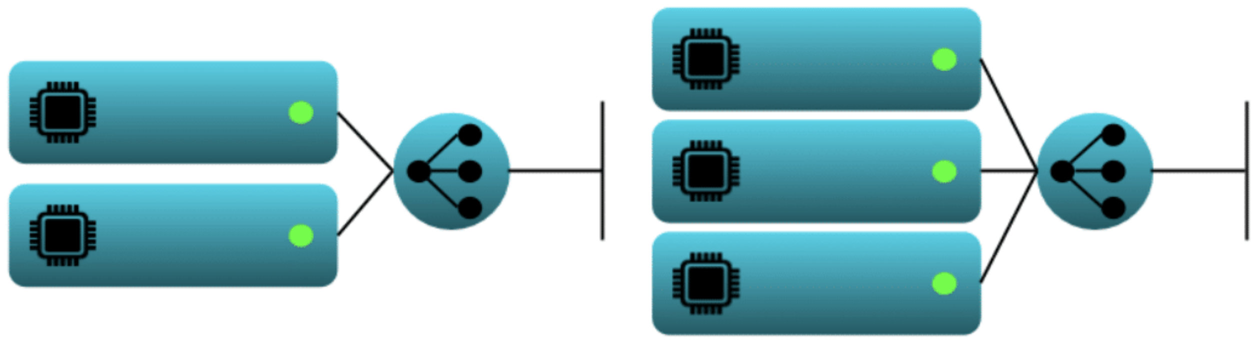
Cloud elasticity and cloud scalability seem like terms that should be possible to use interchangeably. Indeed, ten years after the US NIST provided a clear and concise definition of the term cloud computing, it is still common to hear cloud elasticity and cloud scalability treated as equivalent. While both are important and fundamental aspects of cloud computing systems, their actual functionality is related but not the same. In the 2011 NIST cloud computing definition, cloud elasticity is listed as a fundamental characteristic of cloud computing, while scalability is not. Yet, elasticity is not possible without scalability. The following quote from the NIST definition's clarification of the essential characteristic of rapid elasticity:

"*Rapid elasticity*. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time."

This means that the ability to scale a system, which is the ability to increase or decrease resources, is required before a system can be elastic. Elasticity is the system's ability to take advantage of that scaling ability appropriately and rapidly to demand. So, a system can be scalable without being elastic. However, if you are running a system that is scalable but not elastic, then you are, by definition, not running a cloud. Note that the system need not make use of this capability. It just needs to have the capability.



Scaling up from two to three CPUs



Scaling out from two to three CPUs

Scaling up and Scaling out

In the figure above, we can see the difference between scaling up and scaling out to increase a system's resources, in this case, CPU capacity. The converse would be scaling down or scaling in when shrinking resources. The scaling up/down terminology (aka vertical scaling) refers to scaling a single resource by increasing or decreasing its capacity to perform. As in our example, this could be the number of CPU cores – real or virtual – available in a single server. The scaling up or out concept (aka horizontal scaling), again illustrated here with CPUs, is a matter of adding replicas of resources to address demand. This could just as easily be envisioned as spinning up additional containers or VMs on a single server as the CPU example we are using.

Again, from a cloud definition, it is not what resource is being scaled, rather understanding how resource capacity is being increased or decreased. Confusion has crept into how the insistence of some uses cloud scaling and cloud elasticity. Each somehow refers to either a characteristic specific to infrastructure or an application. Elasticity is often artificially tied to infrastructure and scalability to applications. If we return to the NIST definition of elasticity, it does not explicitly call out infrastructure or applications and instead refers to capabilities. These capabilities are explicitly less critical than the overall system's ability to adjust to changing needs rapidly.

In truth, what is important to the end-user is not the means but the end. Depending on how much change in demand a system experiences, it is quite possible that adding or deleting application instances can provide the rapid elasticity needed.   The explosion in popularity of Linux containers such as Docker and Serverless/Function-as-a-Service (FaaS) solutions means that applications can be incredibly and quickly elastic without an absolute need to provision additional hardware, real or virtual. Continued improvement and automation of how hardware is provisioned and de-provisioned – even physical hardware – make integrating the hardware and software to provide even better elasticity increasingly functional and common.

Moving from "Cloud Scaling" to "Cloud Elasticity"

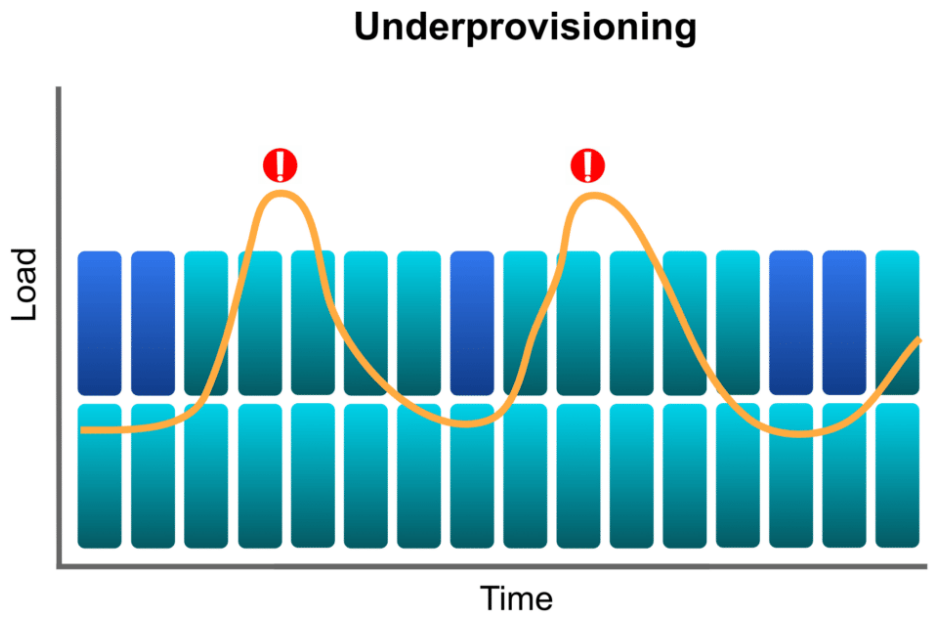
The extreme of scaling over or under compensates against the realities of production load. As an example, let's assume we've joined a company that just moved a significant legacy application to the cloud. While the engineering team has done some work to make the app cloud-friendly, such as breaking the app into containerized microservices, we've been tasked to optimize its performance. We've received some performance data, but not much, and based on the limited data, let's assume we've estimated that our necessary capacity is two servers, each costing at $0.05/hour or $1.20/day or $438/year. We've also implemented a more robust monitoring system to provide feedback on parameters such as application performance and server utilization. Unfortunately, we find that our initial static capacity estimate results in one server sitting idle during certain times of the day, costing us $6.00 per day or $2190 per year of excess resource costs. Furthermore, we did not estimate our daily load well enough, and we consistently see outages twice a day. This could cost us even more in lost revenue than the net cost of infrastructure through failed transactions and lost customers. What has happened here is a case of underprovisioning resources compared to our actual demand.

32 instances at
$0.05/hr = **$38.20/day**

5 idle instances at
$0.05/hr = **$6.00/day**

Outages = **2**

🟩 Active 🟦 Idle



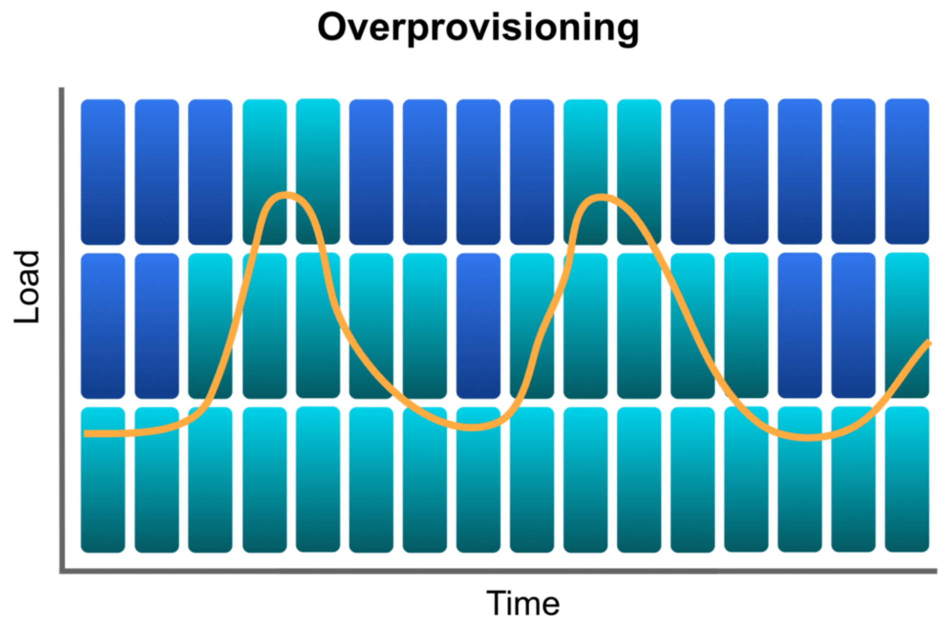
In a traditional IT infrastructure, the logical step would be to increase capacity. And as our CEO and head of engineering see performance initially more critical than cost, we look at scaling the system. You chose whether you wish to scale up or out, but the result is that we increase our capacity to three servers available to our system at all times. Now that we have scaled our system, we've eliminated our daily outages and, unfortunately, increased our overall system cost and substantially increased our wasted spending on idle servers to $20.40/day or $7446/year.

48 instances at
$0.05/hr = **$57.60/day**

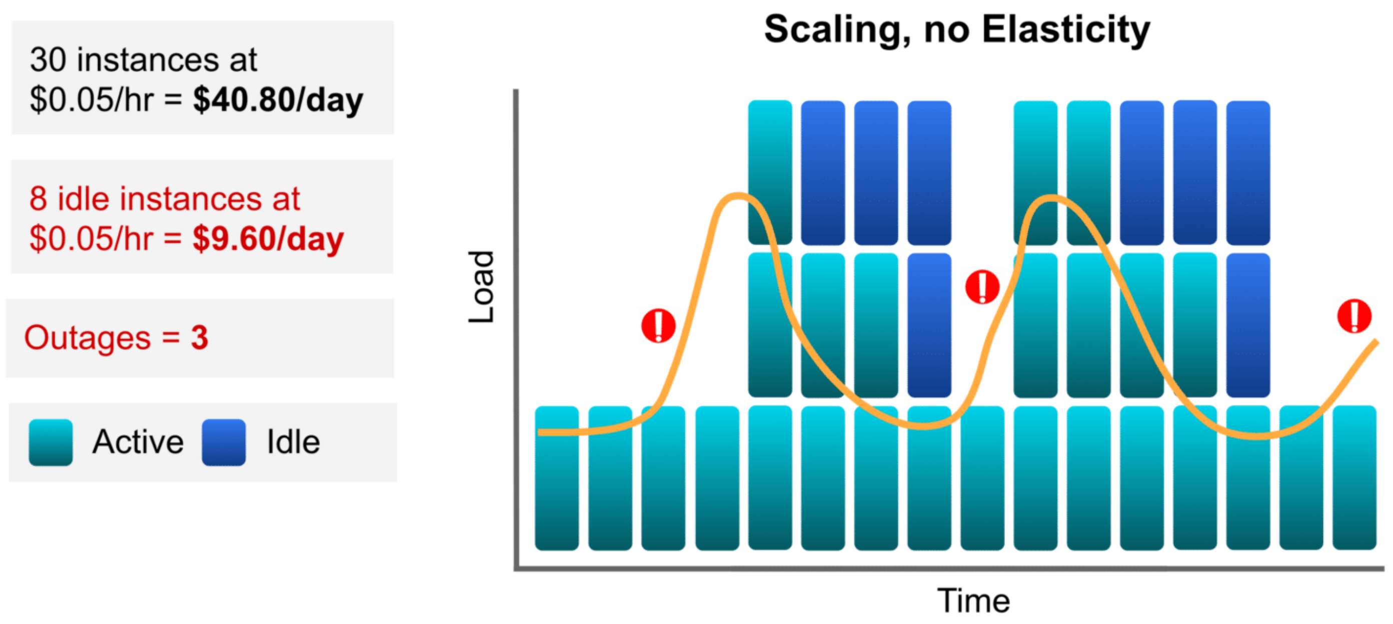
17 idle instances at
$0.05/hr = **$20.40/day**

Outages = **0**

🟦 Active 🟦 Idle



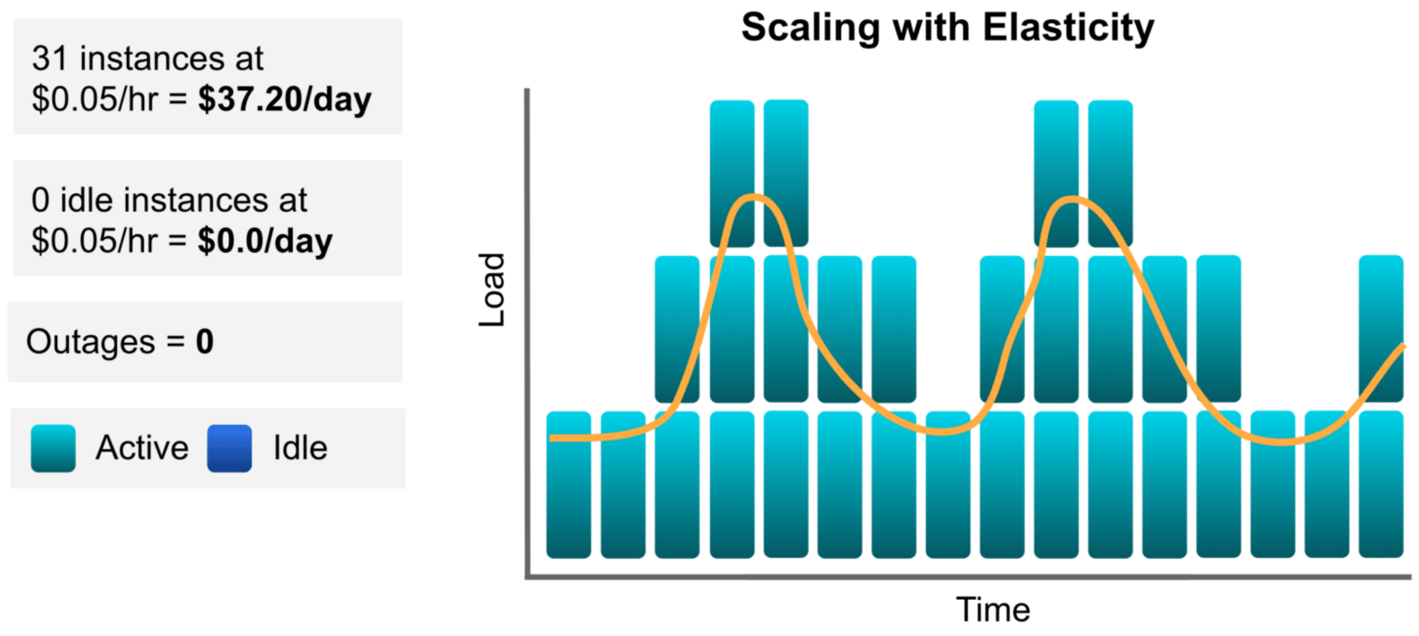
It just happens that our company hired a CFO that is really into FinOps, and she realizes that we are treating our infrastructure like it is a traditional IT resource, not a cloud. So we give scaling in response to changing load a try. Knowing that most of our system's load was covered by two servers, we scale back down (or in) to that level and set an alarm to page an engineer to scale our infrastructure to meet demand. Unfortunately, demand spikes and drops rapidly. By the time our very competent engineer has the additional servers online, there have been outages, and it also takes a while to scale back down.



FO is pleased we've cut our idle infrastructure cost in half; she still sees some cost savings that should be attainable. On top of that, our head of engineering and CEO is not pleased that we are again in a state where we are having outages and the work of manually scaling up and down in response to system changes is tedious work. We have achieved cloud scaling, but are not yet at a point of true cloud elasticity.

Cloud Elasticity to the Rescue

Finally, our team points out that our cloud provider has several automation tools that could tie into our monitoring system and automate the necessary rapid scaling responses to truly let us achieve cloud elasticity rather than merely cloud scaling. The outcome makes the CEO, CFO, and head of engineering happy with the entire team and further has eliminated the toil for your team of manually responding to load changes.



Because the process is automated, the response to changing loads is appropriate and rapid, resulting in eliminating outages and idle servers. Now that things look automated and stable, the CFO points out that there are times where server capacity is not optimal, and it might be time to look at that, but that will need to wait for another post.

Is Cloud Elasticity Required?

Now early in this article, I noted that not just elasticity, but "rapid elasticity" is required, by definition, for a cloud actually to be a cloud. Does this mean that your system MUST be elastic? In truth, no, it just needs to have the ability to be elastic to be a cloud system.

If you are running a service tied to retail sales, and seasonal events such as Valentine's Day, Christmas, or Black Friday/Cyber Monday spike the demands on your systems. This alone might warrant making sure your system has its cloud elasticity functionality ready to go. If, on the other hand, you are serving business software to small companies that have predictable growth and use rates throughout the year, elasticity may be less of a concern. Indeed the question might further be, do you need to run your system on a cloud?

Still, the point of cloud computing can be distilled down to another one of the NIST "essential characteristics" of cloud computing – self-service, on-demand access to resources. The uncertainty of the on-demand requirement makes cloud elasticity – and rapid elasticity at that – necessary. If your service has an outage because of insufficient resources, you've failed your end-users, and having elasticity working on your system is the prudent choice.

Conclusions: Cloud Scalability AND Cloud Elasticity

Hopefully, you are now clear on how your system's ability to scale is fundamental but different from the ability to quickly respond – be elastic – to the demand on resources. Being able to scale has no implications about how fast your system responds to changing demands. Being elastic, especially in the context of cloud computing, requires that the scaling occur rapidly in response to changing demands. A system that exhibits true cloud elasticity will need to have scalability and will likely be automated to avoid the toil of manual action and to take advantage of the responsiveness provided by computer-aided processes.

Related Posts

Evolving APM for the Cloud-Native Era

[Webinar] November 2021

Beyond APM: achieving the benefits of CI/CD and cloud native

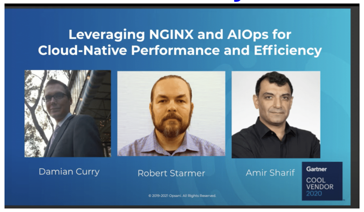
Jay Lyman
Senior Analyst
451 Research, part of S&P Global Market Intelligence

451 Research
S&P Global
Market Intelligence

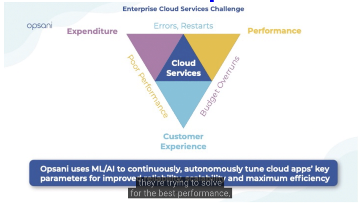
Presented with Opsani

Copyright © 2021 S&P Global Market Intelligence. Permission to reprint or distribute any content from this presentation requires the prior written approval of S&P Global Market Intelligence.

Leveraging NGINX and AIOps for Cloud-Native Performance and Efficiency


Leveraging NGINX and AIOps for Cloud-Native Performance and Efficiency
Damian Curry
Robert Starmer
Amir Sharif
Gartner COOL VENDOR 2020

Autonomous Optimization: The Future of AIOps



Opsani uses ML/AI to continuously, autonomously tune cloud apps' key parameters for improved reliability, scalability and maximum efficiency

they're trying to solve
for the best performance,