AWS vs. Azure: 9 Benefits of Hosting Sitecore on Azure PaaS

ENGAGENCY.COM

Website | Schedule a Meeting

In recent years, a growing number of businesses have moved their Sitecore deployments from physical hardware to the cloud.

When making this decision, many of our clients at Engagency have asked us, “Which is the better hosting environment for Sitecore: Azure PaaS or Amazon Web Services?”

Both Azure PaaS and Amazon Web Services (AWS) offer significant benefits to businesses, but for the majority of situations, Azure PaaS is the one we recommend. Why?

Azure PaaS is easier to set up, it’s easier to scale, and it can be significantly less expensive.

But that’s not all. To learn more about the various benefits Azure PaaS can offer your organization over AWS, read on.

1. Azure PaaS is faster to set up

The benefits of Azure PaaS over AWS start at the very beginning.

If you set up your Sitecore environment on AWS instead of Azure PaaS, you should plan for it to take at least 10 times longer—and cost up to $15K (or more) in additional consulting hours.

On Azure PaaS, set up shrinks down to just a few hours. Here’s how that’s possible:

- In the Azure PaaS setup wizard, you’ll plug in a few key answers, including your current traffic numbers and future traffic predictions.
- Then, you can go grab a cup of coffee while you let the wizard do its thing.
- About an hour later, you’ll have a Sitecore environment.
- Spend a few more hours tweaking things to your ideal configuration, and you’re done.

Set up your Sitecore environment on AWS, and you have to plan out the exact traffic requirements for every instance you’re going to spin up. Don’t forget the databases, virtual machines, load balancer, and more. That’s a lot of infrastructure to spin up and connect.

Depending on the size of your environment, your Sitecore partner may charge you up to $15K to do this from scratch—and that’s just the execution part of it. That doesn’t include the time your Sitecore partner has to spend planning the implementation with your internal teams to make sure it's done correctly the first time.

2. Scale resources in real-time

When it comes to scalability, there's no contest: Azure PaaS allows for near real-time scalability. Experience a traffic surge on AWS, and your site can slow down for hours, even going offline altogether as your team scrambles to bring up an extra virtual machine.

With Azure PaaS, you can optimize Sitecore performance by adding and removing resources as needed to handle short-term traffic spikes. Azure PaaS was designed to make this easy. Simply login to the Azure PaaS portal, and drag a slider. In a matter of clicks, you can scale from 3 instances to 10 and your site is ready for Black Friday in minutes.

On AWS, you'll have to bring up an extra virtual machine, deploy Sitecore to it, configure it on your load balancer, and then tear it down once the surge ends. You'd have to repeat this process for every virtual machine, introducing more opportunity for error—especially if your team is working in a panicked state.

At best, this process could take a few hours. At worst, it could take a few days. Either way, you've lost the opportunity to capitalize on that traffic. Your site could slow to a halt or not be reachable at all, losing your customers' trust—and their sales.

3. Scale automatically

You can even program Azure PaaS to scale automatically. Set up triggers and Azure PaaS will make the changes in minutes.

For example, you might have 3 instances available serving your production site during business hours. After you close, you drop it down to 2, and then down to 1 during the midnight hours. You could also create auto-scaling triggers based on other metrics, such as spinning up another resource if CPU usage exceeds 70%.

Theoretically, you could set up AWS to scale comparably with Azure PaaS, but it involves maintenance of your Amazon Machine Images (AMI). If you have a perfectly up-to-date AMI (and we do mean perfectly), you can auto-scale and it might take 10 to 15 minutes to provision. This is still a bit longer than the minutes Azure PaaS promises, but it's close enough.

However, this describes a best-case scenario where you have a dedicated DevOps team spending 25% to 50% of their time managing your environment. For most companies, that's simply not feasible.

So while it is possible to auto scale with AWS, why would you? Use Azure PaaS instead, and you'll reduce your risk and make life easier.

4. Reduce risk with blue/green deployments

No matter how you slice it, deployments are stressful. Even with the best QA and UAT testing, unanticipated problems can crop up during and after deployments.

Part of what makes deploying on AWS complicated are the AMIs we mentioned earlier. AWS works on Amazon Machine Images (AMI). Whenever you spin up a new environment, you have to keep your AMIs up to date so you have a template that's ready to go for your VMs. Otherwise, you won't be able to auto-scale and deployments will take much longer. That's why most companies who go the AWS route have a dedicated DevOps engineer or team who must maintain the AMIs.

If you follow a blue/green deployment strategy, AWS is still tricky. You'll need to have two identical virtual machines running at the same time. Then, you can configure the load balancer to point to one, while you deploy to the other one. Once you've deployed to the first VM, you'll have to switch and configure the load balancer to point to the other one. You'll also have to pay an additional cost for the second VM, while the cost of a staging server is included in your Azure PaaS subscription.

Of course, this is all assuming you provisioned and configured your Sitecore environment to support blue/green deployments in the first place. If you didn't, and you're on AWS, you'll have to go back and redo everything, which may take weeks and up to $18K in consulting time.

Blue/green or not, deploying on Azure PaaS is a matter of navigating to the Swap screen in the portal and clicking "Go." Azure PaaS handles all the reconfiguration for you.

5. Say goodbye to Windows updates and reboots

For the vast majority of system administrators, dealing with reboots and ensuring that regular software updates don't brick the servers or cause unexpected issues, is one of the least appealing aspects of their job.

When you deploy on Azure PaaS, both you and your IT team can have peace of mind that the resources supporting your Sitecore environment are up-to-date. That's because Microsoft handles operating system-level updates to the infrastructure behind the scenes with no disruption. Azure PaaS automatically spins up a new machine, directs your traffic over there, and your site visitors are none the wiser.

Reboots on AWS, on the other hand, require careful planning so you don't patch all your machines at once. It's possible to do, but the time you have to spend planning could be better used for other, more exciting projects.

What's more, once you go live, there's a chance the Windows update could mess with something on your environment. Then, you'd have to roll back and start over.

Are you running a single instance of Sitecore on AWS with no redundancy? You're even more at risk during Windows updates. Your site could go down completely for 10 minutes while you wait for the server to reboot. If you can't afford the downtime, you can spin up a new environment for redundancy first, but that takes up more time—easily 2+ hours. That's precious time that could be used more effectively.

With Azure PaaS, everyone is happy. Your IT team doesn't have to worry about risky reboots. Your marketing team can rest assured the site is up and working properly, all the time. And there are no angry executives knocking down doors to find someone to blame for unexpected downtime.

6. Upgrading Sitecore is just as easy

Sitecore upgrades are also much simpler if you're already on Azure PaaS, particularly when moving to a new point release. Simply create a new instance with the current version of Sitecore, move your code over to it, upgrade the databases and you're done.

With AWS, the process is much more involved:

- First, you'll take a copy of your production database, and point 2 or 3 of your live servers toward it.
- Next, take one of the live servers pointing to your old database offline.
- Then update your old database and run the Sitecore upgrade.
- Now, you'll have to wait until a lower-traffic time in the day when you can risk repeating the process for your second server.
- Put that second server back on, rinse, and repeat.

Your IT team will be professional jugglers by the end of this.

Don't forget that your team has to spend time documenting these processes for AWS, too, taking up more of their time and delaying your Sitecore upgrade as a result. Whereas, with Azure PaaS, the process is so intuitive and streamlined, there's no documentation required.

Based on what we've seen with our own clients, we'd estimate the average Sitecore upgrade takes a third of the time on Azure PaaS, as it does on AWS.

7. Create multiple environments with facilitated geo-distribution

To keep customers around, your website needs to load fast, every time. If your company supports users in multiple countries or continents, you need geo-distribution to achieve the performance everyone expects.

Deploying a geo-distributed Sitecore environment with Azure PaaS is as easy as using Azure Traffic Manager to route visitors to the appropriate app service for their location, and using Azure SQL Geo Replication to keep the data close to the user. Your databases, servers, and resources can all stick together in a geo-distributed environment, serving users in the US, Europe, Africa, or wherever they may be.

On the other hand, to set up multiple geo-diverse environments with AWS, you'd have to manually create a whole new set of databases, servers, and resources for each new location—as you watch your costs and support hours tick up.

8. Exponential benefits for years to come

As your site grows, each of these Azure PaaS benefits becomes more beneficial.

With Azure PaaS it's just as easy to scale for a one-time traffic surge as it is for multiple. It's easy to create a single Sitecore environment, and just as easy to create multiple. And as Sitecore and Azure PaaS continue to release new services, the functionality of your site can blossom while deploying such functionality remains easy and intuitive.

Of course, you can have a fully supported environment on both Azure PaaS and AWS. But with AWS, you're introducing more difficulties into your life. With each new AWS environment you have to build, and each traffic surge you have to respond to, things get more challenging to manage. The more your traffic grows, the more time your team has to dedicate to avoiding and responding to possible infrastructure failures.

AWS also limits the future potential of your site. You have fewer options for add-on services, as Sitecore continues to optimize for Microsoft's cloud infrastructure over Amazon's.

Azure PaaS is easier to set up, it's cheaper to maintain, and it only gets better with time. Choosing Azure PaaS to host Sitecore will ensure you've set your website up for maximum performance and minimum total cost of ownership, well into the future.

9. Forget unexpected licensing costs and get more for your money

Speaking of costs, Azure PaaS allows you to completely eliminate Windows and SQL Server licensing management and costs. Because Azure PaaS is a Microsoft product, all the licensing costs of additional Windows Server and SQL Server products are baked into the costs of hosting on Azure PaaS.

With both physical and non-Azure PaaS virtual cloud servers like AWS, your company is on the hook for additional Windows Server and SQL Server licensing costs. It's also your responsibility to ensure that each machine is allocated an appropriate number of client access licenses.

Azure PaaS simplifies things by removing that requirement, so you only pay for the resources your environment consumes. It's a fully-managed cloud solution, so there is no additional cost for server management or licensing costs. With Azure PaaS, your cost remains relatively fixed based on your usage.

Where do those consulting estimates come from?

We threw a lot of numbers around in this article. We estimated these hours based on typical bill rates and the average time it would take a Sitecore expert with years of experience. If you opt to go the DIY route and do things internally – and something goes wrong – it can cost even more.

How Engagency can help

If you are considering moving your Sitecore solution to Azure PaaS, the Engagency team can help. For over a decade, we've helped a wide range of clients with Sitecore upgrades, infrastructure planning, data migration, and best practice implementation.

It all starts with a comprehensive Sitecore Implementation Audit, where we take a deep dive into the state of your code, content, and overall environment — including pre-upgrade issues and optimization opportunities, which can be addressed seamlessly through one of our Managed or Dedicated Services plans.

Intrigued? Contact us now to find out how Engagency can help your organization reap the full benefits of Sitecore, the world's most powerful experience platform.