

Optimizing SAP for Azure

SAP is the backbone of our digital transformation. Like many enterprises, Microsoft uses SAP—the enterprise resource planning (ERP) software solution—to run the majority of our business operations. And we’re the only enterprise that runs SAP on our own platform. At Microsoft we’re running SAP on Azure, the preferred platform for SAP and the best platform for digital transformation.

We’ve optimized our SAP on Azure environment to gain business and operational benefits that make our SAP environment agile, efficient, and able to grow and change with our business. Optimizing our Azure environment has allowed us to:

- Increase cost saving by using our Azure infrastructure more efficiently.
- Create a more agile, scalable and flexible SAP on Azure solution.

[As of February 2018, Microsoft’s instance of SAP is 100% migrated to Azure.](#) By optimizing SAP on Azure, we’re positioning our SAP environment to grow and change with our business needs. Additionally, we’re positioned to lead our digital transformation and empower everyone in our organization to achieve more. Azure makes SAP better.

SAP at Microsoft

Like many enterprises, Microsoft uses SAP—the global enterprise resource planning (ERP) software solution—to run various business operations. SAP has systems and apps for mission-critical business functions like finance, human resources, supply chain management, and others. [SAP on Azure](#) is your trusted path to innovation in the cloud. It provides an agile infrastructure, minimizing downtime, risks, costs, and improves employee efficiencies to power the digital transformation.

Each SAP system or app in the overall SAP landscape uses servers and hardware, computing resources (like CPU and memory), and storage resources. Each system also has separate environments, like sandbox and production. The resources needed to run SAP can be costly in an on-premises model, where you have physical or virtualized servers that often go unused.

Consider a typical on-premises system. The IT industry often sizes on-premises servers and storage infrastructure for the next three to five years, based on the expected maximum utilization and workload during the life span of an asset. But often, the full capacity of the hardware isn’t used outside of peak periods—or isn’t needed at all. Maintaining these on-premises systems is costly.

With Azure, we combat infrastructure underutilization and overprovisioning. We quickly and easily scale up and scale down our SAP systems for current and short-term needs, not for maximum load over the next three to five years.

Capacity management decreases our costs and increases our agility

By engaging in capacity management and sizing our SAP systems for Azure:

- We have a lower total cost of ownership—we pay for only what we need, when we need it. We save on costs of unused hardware and ongoing server maintenance.
- We cut the core counts (number of CPUs in a machine) nearly in half—from 64-core physical machines to 32-core virtual machines for almost every server that we moved.
- We’re much more agile. We size for now, and easily add/change our setup as needed to handle new functionality. For example, in minutes, we changed an 8-CPU virtual machine to 16 CPUs, doubled the memory, and added Azure Premium Storage for our short-term needs. Later, to save costs, we easily reverted to the original setup.

What does optimizing involve?

Optimizing involves calculating our hardware requirements like CPU resources, storage space, memory, input/output (I/O), and network bandwidth. When we optimize, we size for today. We assess our infrastructure, resources, and costs, and then size our systems as small as possible. And we ensure that there's enough space to run business processes without performance issues during expected events like product releases or quarterly financial reporting. This capability provides Microsoft the ability to right-size and tight-size our storage and computing power, giving us flexibility and agility on-demand.

Tips for sizing

Sizing is an ongoing task because the load, business requirements, and behavior patterns can change at any time. Here are some considerations and tips, based on the process that we use:

- **Design for easy scale-up and scale-out.** Upsize only when needed, rather than scaling up or out months or quarters ahead of an actual business need. Start with the smallest possible computing capacity. It's easy to add capacity later. And resize before business processes change or before new processes go live in the environment. Autoscaling up and out brings additional benefits, because it's an automatic response to current conditions and usage patterns.
- **Figure out how many virtual machines a system needs.** Our production and user acceptance testing (UAT) systems have multiple virtual machines. But for sandbox and quality assurance, we usually have single virtual machines. Sometimes our SAP app instances and database instance are on the same virtual machine.
- **Don't size for only CPUs and memory.** Size for storage I/O and throughput requirements, too.
- **Consider upstream/downstream dependencies in data movement and in app-to-app communication.** Say that you move an app into a public cloud. Adding 20 to 40 milliseconds in communication between on-premises and public cloud apps can affect dependencies, and it can affect customers or your SLAs with business partners.
- **Decide whether all your systems need Azure Premium Storage.** For our archiving system, we used Azure Standard Storage and small virtual machines. When we loaded data into the system, we temporarily doubled the memory and CPU, and added Azure Premium Storage for log file drives in the database. To save costs, after we loaded the data, we made the virtual machine smaller again and removed the Premium Storage drives.
- **Decide if all apps have to run continuously.** Can some apps run eight hours a day, Monday through Friday? Save costs by [snoozing](#). At night or on weekends or holidays, you can often snooze development and sandbox systems if they aren't in use. Also consider test systems and potentially even some production systems for snoozing. Create a snooze schedule and provide key personnel with the ability to unsnooze systems on demand. If you have a separate business continuity system and you snooze hardware for it, you pay for only storage, not for compute consumption. Or use the smallest size feasible. If there's a disaster—resize to a bigger size before you start the production system—for example, the database server for business continuity.
- **Keep monitoring and managing system and resource capacity.** Make changes before issues occur. Monitor storage use, growth rates, CPU, network utilization, and memory resources that are used on virtual machines. Again, consider autoscaling up and out. If monitoring indicates that a system is consistently over-sized, then adjust downward.

Two common strategies for sizing SAP systems

We used two common strategies for sizing SAP systems, each in a different way and at different points in the optimization process. We used the SAP Quick Sizer at the start of our optimization process because it had a simple, web-based interface and it allowed us to get sizing strategies prepared right away. We used Reference sizing later on, once we had determined some context around our virtual machine sizing and we could provide virtual machines in Azure for reference.

SAP Quick Sizer

If you don't yet have systems or workloads in Azure, start with SAP Quick Sizer—it's an online app that guides you on sizing requirements, based on your business needs. Quick Sizer is good for capacity and budget planning. The app has a questionnaire where you indicate the number of SAP users, how many transactions you expect, and other details. The SAP system recommends a number for the SAP Application Performance Standard (SAPS)—a measurement of processing requirements—that you need, such as for a database server. If the recommended number is 80,000, you find servers with SAPS that add up to 80,000. You can find more information about SAPS for Azure virtual machines in [SAP Note #1928533 SAP Applications on Azure: Supported Products and Azure VM types](#) (SAP logon required).

But there are a few points that you should keep in mind with Quick Sizer. There can be customization and variations of SAP systems, depending on business processes, which could change system behavior. Or, say you have capabilities enabled for new SAP deployments or custom code for which there's no Quick Sizer. Also, in the past, hardware vendors guided customers on what servers they needed and how to install them. With Azure, customers make their own decisions—for example, how to grow storage as data volume grows, or how to adjust CPU compute resources.

Reference sizing

After you have one or two systems in Azure, reference sizing is the recommended method. With this approach, you look at the performance of systems you've already moved to Azure that are similar to the systems you want to move. This comparison helps you estimate your sizing requirements fairly accurately. Say that you have an on-premises system that you want to move to Azure, and it's three times as big as one of the systems you already have on Azure. Adjust the sizing based on systems you've already deployed in Azure, and then deploy the new system.

And what if your estimate wasn't accurate? It's much easier and quicker to adjust CPU and memory resources in Azure by choosing a different virtual machine size. Adjusting the database on-premises is much harder because you might need to buy servers with more CPU and memory. For on-premises, you have to look at what you have, add a buffer, and consider what additional load you'll have in the next few years.

Some key technical considerations

When we integrate SAP with Windows Server and SQL Server, our main considerations are cost of ownership and low complexity. When you plan your integration and reference architecture, make sure that the technical landscape is easy and inexpensive to operate. With business-critical systems, it's hard to scale when you have an architecture where you need highly skilled people, or when there are emergencies where you need business continuity.

For easy administration and operations, we use the same app design in all SAP production systems. We only change virtual machine sizes and number of virtual machines for the SAP application layer. It's very inexpensive to run SAP systems on SQL Server—in the last 23 years, Microsoft has benefited a lot from this.

Also, to avoid issues for customers who run SAP workloads on Azure, Microsoft certifies only certain Azure virtual machine types. These must meet memory, CPU, and ratio requirements, and they must support defined throughputs. To learn more about Azure virtual machine types certified for SAP, visit [SAP certifications and configurations running on Microsoft Azure](#).

Technical implementation and technical capabilities

Figure 1 shows our SAP ERP/ECC production system in Azure. By moving to Azure, we've gained agility and scalability on the application layer. We can scale up and down by increasing and decreasing the size and number of the virtual machines. The design and architecture have high availability measures against single points of failure. So, if we need to update Windows Server or SQL Server, perform infrastructure maintenance, or make other system changes, it

doesn't require much—if any—downtime. We implement infrastructure in Azure for our production systems with standard SAP, SQL Server, and Windows Server high availability features.

High availability

Eliminates single points of failure.

Guards against planned and unplanned downtime with high redundancy.

Dual databases

Allows staggered releases.

Rolls back releases instantly.

Offers local "hot" standby.

Automates failover.

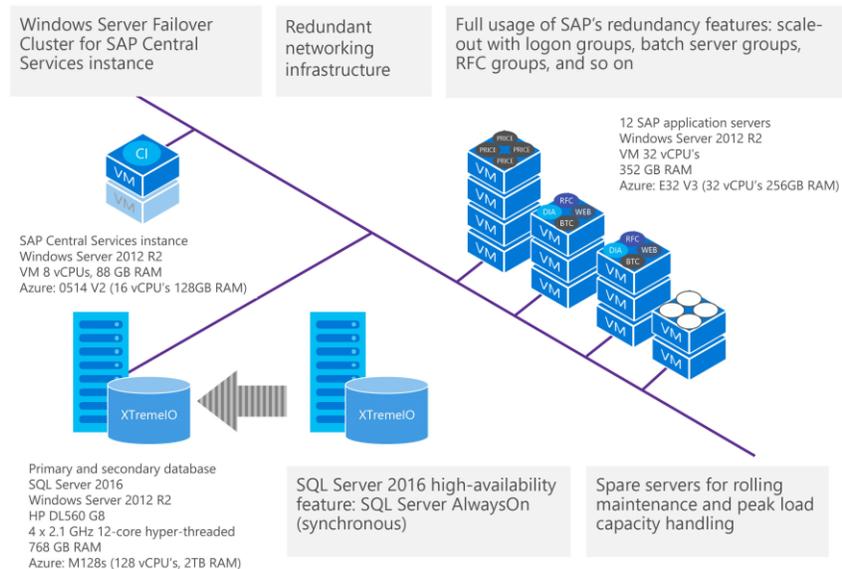


Figure 1. Current SAP ERP/ECC production system in Azure.

High availability and scalability

For high availability, SQL Server Always On is a standard method. We have two database servers where we use SQL Server Always On with a synchronous commit. If one database server goes down or is undergoing maintenance, we don't lose data. This is because the data is committed on both database servers, and SAP automatically reconnects to the other database. Because we can use the secondary database, we can upgrade software and SQL Server, roll back to previous releases, and do automatic failovers with no or minimal risk.

Also, for high availability, we have an SAP Central Services instance that runs on Windows Server Failover Clustering. The two cluster nodes share the data image.

For scalability and high availability of the SAP application layer, multiple SAP app server instances are assigned to SAP redundancy features like logon groups and batch server groups. Those app server instances are configured on different Azure virtual machines for high availability. SAP automatically dispatches the workload to multiple app server instances per the group definitions. If an app server instance isn't available, business processes can still run via other SAP app server instances that are part of the same group.

Rolling maintenance

The scale-out logic of SAP app server instances is also used for rolling maintenance. We remove one virtual machine (and SAP app server instances running on it) from the SAP system without affecting production. After we finish our work, we add back the virtual machine, and the SAP system automatically uses the instances again.

If there's high load and we need to scale out, we add spare virtual machines to our SAP systems. And when we're doing rolling maintenance, we also use the spares to replace a server without reducing overall resources.

Lessons learned

We keep learning and iterating as we optimize. Here are some important lessons that we've learned:

- Ensure that you don't over-provision your virtual machines, yet make sure that you provision enough resources so that you don't have to keep increasing your system weekly.

- Design and architect your infrastructure and storage in Azure so that it can scale.
- Even for our development and test systems, we decided to use Azure Premium Storage for its low latency. Also, during project implementation, there are often multiple developers who simultaneously use the development systems.
- The types of virtual machine storage and Azure networking that we use are influenced by lessons learned about functionality. Azure CAT drove several Azure load balancing features based on customer feedback and deployment. This gave us better performing and more efficient SAP system configurations.
- Design for high availability in your production systems with Windows Server Failover clustering, SQL Server Always On, and SAP features like logon groups, remote function call groups, and batch server groups.

Looking ahead

We're excited about the decreased costs and increased agility that we've seen so far. In the future, we'd like to:

- Share more lessons learned from our SAP migration to Azure. For information, see [Strategies for migrating SAP systems to Microsoft Azure](#).
- Automate the sizing of our simpler systems and environments and develop auto-scale. Automation and auto-scale apply more to the middle tier—the SAP application layer—but we'd also like to auto-scale up and down for the database layer and file servers. We want systems to auto-scale based on current conditions.
- Add more automation for business continuity. Right now, we use the same semi-automated business continuity process in Azure as we did on-premises. If there's a disaster, production fails over to a different Azure region.
- Explore new business continuity strategies and technology options as they apply to Azure.
- Enable more people in Microsoft to use self-serve features—for example, snoozing and unsnoozing non-production servers by using a snooze portal.
- Add more automation for snoozing and unsnoozing systems.
- Help customers with SAP scenarios like Azure backup or Azure encryption of data at rest, and with questions like:
 - Which policies do I apply in the SAP landscape?
 - What do I encrypt? Do I use disk encryption or database encryption?
 - Do I need the same backup methods for a 50GB database as for an 10TB database?
- Add and use new Azure capabilities. We want to enable more SAP scenarios to run in Azure—better and faster storage, larger virtual machines, better network connectivity, and more Azure operational guidance.

For more information

microsoft.com/itshowcase

[Hello Azure: Unpacking how Microsoft moved its SAP workload to the cloud](#)

[Running SAP Applications on the Microsoft Platform](#)

[Using Azure for hosting and running SAP workload scenarios](#)

[Strategies for migrating SAP systems to Microsoft Azure](#)

[Building an agile and trusted SAP environment on Microsoft Azure](#)

[Azure Accelerated Networking](#)

[Azure Load Balancer Standard](#)

[Create your Azure free account today](#)

© 2019 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.