Strategies for migrating SAP systems to Microsoft Azure

You’ve studied the benefits of moving your SAP systems to Azure and have decided to make the big move. The next logical step is to determine what to move first and how to make the move as smooth as possible. At Microsoft IT, we moved our SAP systems using horizontal and vertical strategies. The horizontal strategy—where we first moved low-risk environments like our sandboxes—gave us Azure experience without affecting critical business functions. The vertical strategy—where we moved an entire system from sandbox to production—gave us experience with production systems on Azure. For both strategies, we moved our lowest-risk SAP resources before more critical ones.

SAP at Microsoft

Like many enterprises, Microsoft uses SAP—the enterprise resource planning (ERP) software solution—to run many business operations. SAP has systems for mission-critical business functions like finance, human resources, and global trade. In today’s business world, rising costs, new processes and requirements, and more data can make it challenging to be agile. With an agile infrastructure, you minimize downtime, risk, and costs, and improve employee efficiency.

At Microsoft IT, our SAP Basis team partners with the company’s Azure Customer Advisory Team to combat these challenges. By moving our SAP systems to Microsoft Azure, we:

- Maximize our cost savings. We’ve seen about 20 percent cost savings when moving from our on-premises physical and virtual servers to Azure.
- Increase agility and scalability, while maximizing system uptime—in the cloud, we can allocate machines and change sizes within minutes.
- Learn more about how to efficiently run our processes and operations in Azure without incurring risk.

We’re running our SAP mission-critical systems on Azure technology. Similarly, you can trust Azure for your own SAP systems, while reaping all the benefits of the cloud. To learn more about our cloud-adoption approach and how we optimize our servers, resources, and costs, see Right-size and tight-size SAP for Azure.

Strategies we use to move our SAP systems to Azure

What systems in our overall SAP landscape do we move to Azure first? To help us decide, we use two strategies—horizontal and vertical. Figure 1 shows part of the SAP landscape in Microsoft.
In the figure, the rows, columns, and blocks illustrate the horizontal and vertical strategies that we use for our SAP landscape. Here are some things to note:

- Typically, enterprises have SAP systems for business functions like ERP, global trade, business intelligence (BI), and others. Within those systems are environments like sandbox, test, and others—all the way up to production.
- Each row (the horizontal dimension) in the figure is an environment. Most companies likely have sandbox, development, test, and production environments, and possibly business continuity. Larger companies might have more.
- Each column (the vertical dimension) is an SAP system for a business function (for example, ERP and BI).
- The rows or layers at the bottom are environments of lower risk and criticality. Those toward the top are higher risk and criticality. As you move up the stack, there’s more risk. So, production is our most critical environment, and user acceptance testing (UAT)—which we also use for business continuity—is second in criticality.
- The systems at the bottom are smaller, in that they have fewer computing resources, lower availability and size requirements, and less throughput. But they have the same amount of storage as the production database.

**Horizontal strategy**

We recommend starting with a horizontal strategy because it’s a safe way to experiment and gain experience with Azure. It’s also a good strategy to use while you redefine your operational, deployment, and approval processes because these will change as you move to Azure. Here’s how the strategy works:

1. To be as low risk as possible, start with low-impact sandbox or training systems. If something goes wrong, there’s very little danger of affecting lots of users or mission-critical business functions.
2. Then, as you gain experience with running, hosting, and administering SAP systems in Azure, apply what you’ve learned to the next layer of systems up the stack.
3. For each layer, estimate costs, potential money saved, performance, and optimization potential—and adjust if needed.

**Vertical strategy**

To get experience with production systems on Azure, we used a vertical strategy with low-risk systems. It also gave us a chance to adjust our internal processes for Azure and to train team members. It’s a great way to spot any issues in production early on. Here’s how the strategy works and our suggestions:

- Look at the impact on cost, customers, service level agreements (SLAs), and legal requirements. We first moved systems (from sandbox up to production) that have the lowest risk, the object event repository (OER) system. Next, we moved the higher risk ones (like BI and ERP, to the left in the figure).
- When you have a new SAP system, put it in Azure right away rather than putting it on-premises and moving it later. In the diagram, OER is an example. At the time, OER was a new, low-risk system. After moving some of our other systems into Azure with the horizontal strategy, we moved the entire OER vertical stack to Azure, end-to-end—from sandbox all the way up to production—simultaneously.
- Don’t move your most critical system first. The last system we move is the highest risk, most mission-critical system—our ERP production system. We need the maximum size virtual machine and the highest performance.
- Move standalone systems first. Some systems are closely joined with other systems—for example, our ERP and GTS systems. There’s a lot of synchronous, real-time traffic between the two. If we move ERP to Azure, but keep GTS on-premises, it will affect performance because of network latency, so we move them together.
- If you have several SAP systems, see if there are upstream and downstream dependencies from one SAP system to the other, or from SAP to apps outside the SAP ecosystem. Look at traffic patterns and areas with high latency.
- If you have tightly connected systems, do a performance analysis to see what impact moving them will have. In our case, if there isn’t much impact, we move them separately to Azure. Otherwise, we move them together.
• In some cases, consider waiting. Sometimes we don’t move certain systems to Azure right away. This could be related to sizing requirements—where the processing requirements are so high that the virtual machines aren’t big enough yet. We need to test to ensure that moving the systems won’t affect our SLAs with customers.

Where we are today

Figure 2 shows the progress we’ve made since 2014 in moving our systems to Azure. And Figure 3 shows the actual number of physical servers, on-premises virtual machines, and Azure virtual machines versus what we forecasted.

In 2014:
520 servers
61% physical hardware
39% on-premises virtual machines

Goal by June 2017:
1% physical hardware
49% on-premises virtual machines
50% Azure

Figure 2. Our path to virtualization and Azure

Figure 3. Actual versus forecasted number of physical servers, on-premises virtual machines, and Azure virtual machines
Benefits we’ve gained

We’ve seen many benefits from moving SAP to the cloud, including:

- **Minimum risk and downtime.** With on-premises, we can’t build up virtual machines in parallel. We have to shut down the server, reconfigure it, and bring it up again—which causes production downtime. With Azure, we just bring up another virtual machine, temporarily duplicate the virtual machine, do any required installations or upgrades on the new virtual machine, and remove the old virtual machine. If we need the old virtual machine, we can use it and decommission it later. We can quickly switch between the old and new virtual machines with virtual server names in Windows Server. The SAP application layer knows only the virtual server/alias name, and it doesn’t have to be reconfigured when the name is moved between virtual machines.

- **More agility and time savings.** We can deploy a system architecture with one or more virtual machines, storage, and virtual networks, and quickly adjust sizing. When we adjusted the size of our virtual machine for our archiving system, we did it in minutes instead of the weeks it would take to set up on-premises hardware. We quickly scale up for high performance requirements—and afterward, we rapidly scale down again to save costs.

- **More self-sufficient.** We don’t have to rely on other teams for hardware or resources. We quickly add virtual machines and adjust resources as we need them.

- **Lower costs.** We’ve seen about 20 percent cost savings when moving from our physical and virtual servers on-premises to Azure. It doesn’t cost a lot of money if we try something and decide to do it differently later. As soon as we decommission the virtual machine and release the storage, there are no longer any costs.

- **Easier processes.** Maintaining our SAP apps in the cloud has simplified many of our processes. For example, we don’t wait weeks for physical hardware or on-premises virtual machines.

Technologies we used

For this solution, we used the following technologies and features in our hardware implementation:

- Azure (IaaS) services and components:
  - Azure virtual machines
  - Network services in Azure (including ExpressRoute for fast speed and low latency connectivity into Azure)
  - Azure Storage
  - Azure Resource Manager JavaScript Object Notation template for unified deployment of virtual machines and landscapes

- SQL Server 2016 on Windows Server 2012 R2
- SQL Server Always On; Windows Server Failover Clustering in Azure
- Microsoft Excel (shows a count of physical and virtual servers, servers on Azure, and how many we plan to move)
- A [third-party tool] to create logical shared drives in Azure

Technical considerations

While implementing this solution, we took a few technical considerations into account. For example, most of our systems have some interfaces where they write files to a file server. With the move to Azure, writing files over a longer distance can cause slowdowns because the data isn’t streamed all at once. To prevent slowdowns, we’re building file servers for systems that we move to Azure right away, and we’re keeping some other file servers on-premises.

Another consideration is that you can use Azure as another datacenter, so that you don’t have to worry about maintenance and getting hardware. But there may be network latency, depending on the Azure location you select. Know your business processes and be sure that tightly coupled systems don’t have to communicate over a long distance. Bundle them, and move them together. For daily work, don’t move an SAP system tightly connected with US-based, on-premises apps to Azure on another continent, although it might be fine for business continuity.
Technical implementation and technical capabilities

Figure 4 shows the on-premises hardware setup of our biggest SAP production systems. Almost 90 percent of the setup is virtualized—for example, the SAP Central Instances layer and the application layer. We plan to move this setup to Azure. By moving to Azure, we can be agile and scale out on the application layer. And we can scale up and down by increasing and decreasing the sizes 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, do hardware maintenance, or make other system changes, it doesn’t require much—if any—downtime.

**High availability**

- Elimination of single points of failure
- High degree of redundancy guards against planned and unplanned downtime

**Dual databases**

- Allows staggered upgrades
- Instant release rollback
- Local “hot” standby
- Automated failover

System is highly virtualized (87.5%)

---

**Figure 4. Current SAP production system on-premises, where almost 90 percent of the setup is virtualized**

We implement hardware for our production systems with standard SAP, SQL Server, and Windows Server features.

**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 connects to the other database. Because we can use the secondary database, we can upgrade software and work 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 instances are assigned to SAP redundancy features like logon groups and batch server groups. Those app instances are configured on different Azure virtual machines for high availability. SAP automatically dispatches the workload to multiple instances per the group definitions. If an instance isn’t available, business processes can still run via other SAP app instances that are part of the same group.

**Rolling maintenance**

The scale-out logic of SAP app instances is also used for rolling maintenance. We remove one virtual machine (and SAP 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.
Other Azure and Windows Server capabilities

For our storage design, we’re exploring Azure File storage and Windows Server Storage Spaces. And for minimizing downtime, we’re using virtual server names in Windows Server.

Azure File storage

Part of Azure Storage, Azure File storage is an Azure platform as a service offering that allows file shares in the cloud. We’re thinking of using it for our SAP transport directories—these directories contain SAP transports with software changes that need to be moved for SAP systems. We tried Azure File storage on our non-production systems. The benefit is that all our SAP systems connect to the same transport directories. We don’t have to mount a drive to the server. The downside is that it affects how we copy files and the data that people can see. It makes it harder to troubleshoot because this directory isn’t as readily available as other drives—the user who needs access must use special commands to add it.

For SAP, we support Azure Standard Storage and Azure Premium Storage. For scalability and I/O-intensive workloads, we recommend Premium Storage for the database layer and Standard Storage for the application layer. For details about configuring storage for optimal performance, see Running SAP applications on the Microsoft platform.

Storage Spaces

We’re exploring whether to use Storage Spaces for high availability of file servers and for our larger databases. We can use it inside an Azure virtual machine to combine multiple virtual hard drives into a single drive. This helps us to easily grow drives and gives better performance than a single Azure virtual hard drive. For our archiving system, we set up a file server and 11-TB drive on Storage Spaces to store intermediary files between two systems.

Again, the benefit of Azure is that, even if a setup doesn’t work, it’s easy to reconfigure, without a big cost impact.

Virtual server names

For less risk and downtime, we use virtual server names (also called server alias names). Here’s how it works:

- The “physical” SAP Central Instance server—saptstserver01—is the actual name of the server/virtual machine. It’s the name that the datacenter uses for server performance monitoring, and nightly, weekly, or monthly backups.
- There’s a registry entry that we can use to assign a virtual name to the physical server/virtual machine. In this case, sapalias01 is the assigned virtual name. This name is used for SAP app instances installed on the server/virtual machine, and by all users.
- The SAP app only knows the virtual server name. We change the physical server/virtual machine name as needed, without affecting the system. Business continuity failover, server exchanges, and system moves are easy.

How we upgrade

In the past, when we upgraded an operating system, we flattened machines and rebuilt them, which caused downtime. Today, we bring up a new virtual machine with the new operating system and install the software in parallel to what’s on another machine. Then we move over the virtual server name and IP address, and retire the older virtual machine.

This a good example of the flexibility that we get from virtualization—it’s not just the machine that’s virtualized, but also the operating system installation on that virtual machine. In the past, many customers bought a server, installed the operating system, and ran for five years on the same operating system until the next server upgrade because of risk and downtime associated with the upgrades. Today, with Azure and virtualized on-premises servers, there isn’t any new hardware. Instead, virtual machines are moved to new servers—and with the move, they keep their old operating system image. Now everyone who runs on a virtualized environment has to think about how to upgrade operating systems. Using the virtual server name is an easy way to minimize risk and downtime for this activity.

Best practices for security

If you have ExpressRoute connectivity between on-premises systems and Azure, you don’t need a public port open to Remote Desktop Services, and Terminal Services doesn’t have to connect to virtual machines via a public IP address.
For high availability, there are several architectures where you need a load balancer in your SAP landscape. Use internal load balancers that don’t have a publicly exposed surface.

For your Internet proxy, don’t go directly from Azure virtual machines to the Internet. Instead, make sure that all your traffic goes through the proxy that’s set up on-premises (the company proxy) because it has a firewall and rules.

When you’re planning your architecture, use Azure Resource Manager security groups to define who can access, administer, and perform operations on a virtual machine.

**Best practices for business continuity**

Smaller companies sometimes have trouble running a business continuity site because they have only one datacenter. But with Azure, it’s easy because you have all the virtual machines that you’d have in a datacenter. Azure offers many regions, so it’s easy to set up business continuity. We’re still refining our business continuity strategy and want to add automation. But our recommendations are to:

- Keep it simple. The configuration in our business continuity site mimics configuration in production.
- At least once a year, conduct business continuity failover testing.
- To minimize downtime, use virtual names. If there’s a disaster, and a production server goes offline, the support team doesn’t have to remove the server alias of the test server and replace it with sapalias01 (the name of our production server). SAP can run regardless of the name of the server that we install the app on.
- On the SAP application layer, use Azure Site Recovery services. Replicate the content of the virtual machines. On the database layer, use database functionality like SQL Server Always On. If we’re in the US West region, we set up another region like US West 2, and then use SQL Server Always On to get the database content there.
- If you have an ExpressRoute from on-premises into US West as the primary app location, think about how you connect into a business continuity region like US West 2. You might need another ExpressRoute connection for business continuity failover. The ExpressRoute that goes to your primary location could have a disaster, too.

**Communicating our strategy across Microsoft**

We have two strategies for informing teams, executives, and other stakeholders in Microsoft about our SAP migration work, and we’ve received positive feedback. The communications that we send are tailored to one of two audiences:

- **Technical teams, developers, and testers.** In this monthly update, we communicate what we’re moving, the impact, and any possible downtime or slow performance.
- **Chief information officers, company executives, and stakeholders.** This quarterly update is higher level than what we send to technical teams. We explain our horizontal and vertical strategies, with pie charts like the ones shown earlier in Figure 2, and burndown charts of how much of the SAP landscape is physical, virtualized on-premises, or Azure like the one shown earlier in Figure 3.

**Lessons learned**

Here are some examples of what we’ve learned or changed based on our experience:

- **Consider moving low risk systems to Azure with the vertical strategy right away.** When we started, we planned to use the horizontal strategy and then the vertical strategy. But because one of our end-to-end systems was low risk, we used it as a test case for the vertical strategy to get experience with a production environment in Azure.
- **Consider building new systems in Azure from the start.** When we built a new system, we weren’t sure whether to put it on-premises and then move it, or to build it in Azure from the get-go. It was low business impact, so we built it in Azure. We saved money, and learned about cluster setups and production environments in Azure.
• **Balance security needs with the ability to troubleshoot.** In Azure, we don’t open all ports on the cluster installation—only the ones that are really needed. We’re still refining this strategy based on our experience. We want to have it somewhat open to help with troubleshooting, but we don’t want it to be too open, either.

• **Predict known business events.** Don’t move systems when they’re highly critical. We schedule around events like product releases, quarterly financial reporting, and big projects that go live in the production environment.

• **Communicate strategy often.** Stakeholders like to know what’s in progress, what we’re moving next, expected downtime, and possible performance impact. Advance notice means fewer tickets and issue escalations.

**Looking ahead**

We’d like to take advantage of more Azure benefits, and to help our customers do the same. For example, we plan to:

• Move the remaining production servers to Azure.

• Decide whether, and to what extent, we’ll use Storage Spaces and Azure File storage.

• Provide scenario-based guidance to customers on how they can move their SAP systems to Azure.

• Enable more SAP scenarios to run in Azure. For example, better and faster storage, larger virtual machines, better network connectivity, and Azure operational guidance.

• Refine our processes to benefit more from Azure capabilities—for example, snoozing non-production systems over the weekend. For the SAP application layer, we want to autoscale out/in and up/down.

**For more information**

**Microsoft IT**

[Microsoft IT Showcase](microsoft.com/ITShowcase)

**Right-size and tight-size SAP for Azure**

**Video: Running SAP on Azure reduces costs and increases scale**

**Running SAP Applications on the Microsoft Platform**

**On-demand webinar: Running a cost-optimized and agile SAP system at Microsoft IT**

**On-demand webinar: Running SAP workloads on Azure**

**Optimizing resource efficiency in Microsoft Azure**

**Using SAP on Azure Linux Virtual Machines (VMs)**

**IT Showcase data analytics content**

© 2017 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.