

# 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 steps are to determine what to move first and how to make the move as smooth as possible. After 12 months of migration processes, Microsoft has completely migrated its SAP instance to Azure. Our SAP landscape consisting of 16 TB of compressed data (50 TB uncompressed) is in the public cloud, Azure.

We migrated our SAP infrastructure using both horizontal and vertical strategies. The horizontal strategy—where we first moved low-risk environments like our sandboxes—gave us Azure migration experience without affecting critical business functions. The vertical strategy—where we moved an entire low-impact 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 companies, Microsoft uses SAP—the enterprise resource planning (ERP) software solution—to run most of our business operations. SAP provides mission-critical business functions for finance, human resources, and global trade. In today's business world, rising costs, new processes and requirements, and a huge influx of data make it challenging to be agile. With an agile infrastructure, you minimize downtime, risk, and costs, and improve employee efficiency. [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 drive the power the digital transformation.

At Microsoft, our SAP Basis team has partnered with the company's Azure Customer Advisory Team to overcome these challenges. By moving our SAP systems to Microsoft Azure, we have:

- **Increased our cost savings.** We've seen an approximately 15 percent cost savings when moving from our on-premises physical and virtual servers to Azure.
- **Increased agility and scalability, with maximized system uptime.** In the cloud, we can allocate [virtual machines](#), change virtual machine sizes, and initiate failover processes within minutes.
- **Learned more about how to efficiently run our processes and operations in Azure.** We migrated SAP to Azure to create a more efficient environment for SAP and to improve our overall SAP operations metrics, while keeping our data secure.

We're running SAP—the backbone of our business processes—on Azure technology that we trust for our mission-critical systems. If you'd like to learn more about our cloud-adoption approach and how we optimize our servers, resources, and costs in Azure, see [Optimizing SAP for Azure](#).

# Strategies we used to move our SAP systems to Azure

When we decided what SAP systems to move to Azure, we used horizontal and vertical strategies. Figure 1 shows part of the SAP landscape at Microsoft.

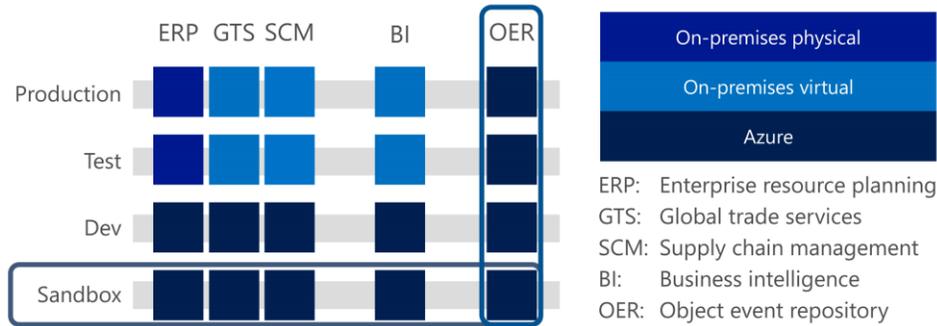


Figure 1. The simplified SAP landscape at Microsoft

In Figure 1, 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 enterprise resource planning (ERP), global trade, business intelligence (BI), and others. Within those systems are environments like sandbox, development, test, and production.
- Each horizontal row in the figure is an environment. Most companies have sandbox, development, test, and production environments, and possibly an environment for 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 lower-risk environments and are less critical. Those toward the top are higher risk and more critical. As you move up the stack, there's more risk in the migration process. So, production is our most critical environment, and the environment for user acceptance testing (Test)—which we also use for business continuity—is our second-most critical.
- The systems at the bottom are smaller, in that they have fewer computing resources, lower availability and size requirements, and less throughput. However, they have the same amount of storage as the production database.

## Horizontal strategy

We started with a horizontal strategy from the bottom of the stack 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. These processes will change as you move to Azure. Here's how the strategy works:

- To limit risk, start with low-impact sandbox or training systems. If something goes wrong, there's very little danger of affecting many users or mission-critical business functions.
- 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.
- 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 in parallel to the horizontal strategy. This also gave us a chance to adjust our internal processes for Azure and train team members. It's a great way to spot any issues in production early on. Here's how the strategy works:

- 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 governance, risk, and compliance system and then the object event repository (OER) system. Then we moved the higher risk ones, like BI and ERP.
- When you have a new SAP system, start in Azure by default rather than putting it on-premises and moving it later. In the diagram, OER is an example of this. At the time, OER was a new, low-risk system. After moving some of our other systems into Azure with the horizontal strategy, we deployed the entire OER vertical stack to Azure, end-to-end—from sandbox all the way up to production.
- Don't move your most critical system first. The last system we moved was the highest risk, most mission-critical system—our ERP production system. We needed the most performance-intensive virtual machine SKUs and the largest storage.
- 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 moved them together.
- If you have several SAP systems, look for upstream and downstream dependencies from one SAP system to the other, or from SAP to apps outside the SAP ecosystem. Examine traffic patterns and areas with high sensitivity to latency.
- If you have tightly connected systems, do a performance analysis to see what effect moving them will have. In our case, if there wasn't much impact, we moved them separately to Azure (for example Business Warehouse independent of ERP). Otherwise, we created migration groups and moved them together.
- In some cases, consider waiting. Sometimes we didn't move certain systems to Azure right away. This could be related to sizing requirements, when the processing requirements were so high that the virtual machines weren't yet big enough. We ran tests to ensure that moving these systems wasn't going to affect our SLAs with customers.

## Where we are today

Figure 2 shows the progress we've made since we began moving our systems to Azure in 2014, which we completed in February 2018, four to six months ahead of our original timelines. Azure now supports 100 percent of our SAP infrastructure, and all SAP Systems have migrated.

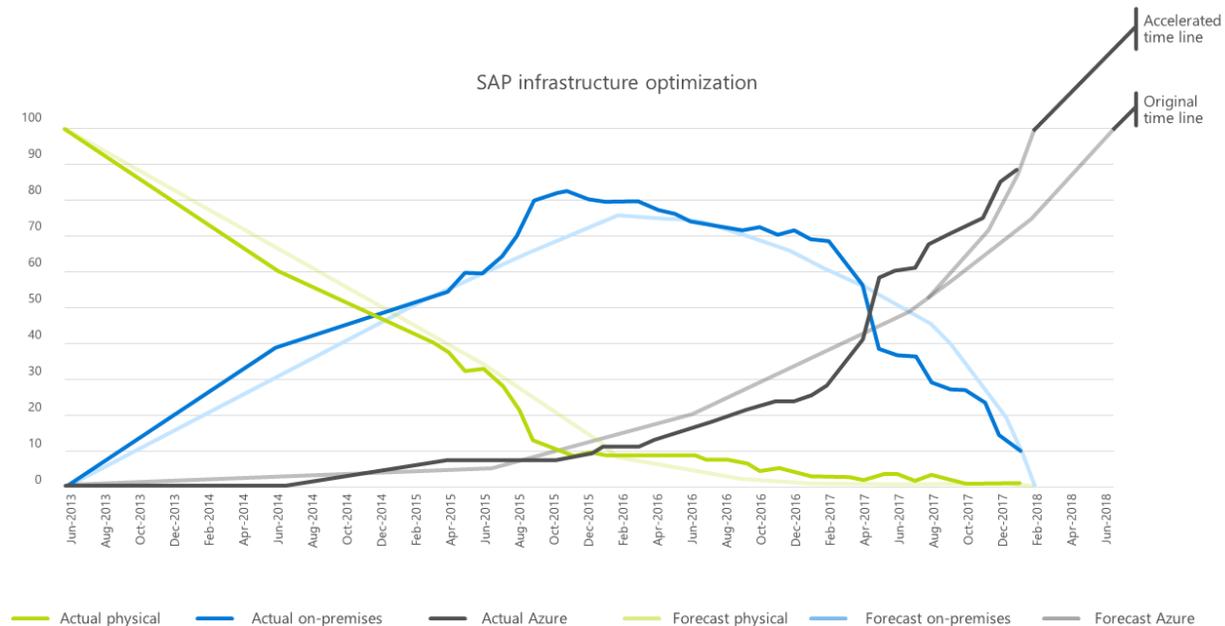


Figure 2. Timeline for SAP Infrastructure optimization

## 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 due to limited on-premise resources. We have to shut down a 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 needed 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 an approximately 15 percent cost savings when moving from our on-premises physical and virtual servers to Azure. Azure allows SAP to run in an optimized, performance-first environment that scales with our needs. We pay for only the resources we use, when we use them. It doesn't cost a lot of money if we try something and decide to do it differently later. As soon as we decommission a 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 SAP on Azure, we used the following technologies and features in our hardware implementation:

- Azure (IaaS) services and components. Our SAP systems are hosted in Azure IaaS virtual machines, which provide native high-availability and scalability. The Azure IaaS services we use include:
  - Azure virtual machines.
  - Network services in Azure (including ExpressRoute for fast speed and low latency connectivity to Azure).
  - Azure Storage.
  - Azure Resource Manager JavaScript Object Notation template for unified deployment of virtual machines and landscapes.
- SQL Server 2016 on Windows Server 2016. SQL Server 2016 is the default data storage provider for SAP.
- SQL Server always-on; Windows Server failover clustering in Azure.
- Microsoft Excel. We used Excel to show the number of on-premises physical and virtual servers on Azure, and how many we plan to move.
- A [third-party tool](#) to create logical shared drives in Azure.
- PowerShell to script and automate the system and server migrations.

## Technical considerations

While implementing SAP on Azure, 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 more indirect network path can cause slowdowns because the data isn't streamed all at once. To prevent slowdowns, we built file servers for systems that we moved to Azure right away, while still maintaining our on-premise file server for systems that remained on-premises.

Another example is bundling our tightly coupled systems from a business process standpoint and moving them together. This ensures that tightly connected systems will have no network latency issues. For daily work, don't move an SAP system tightly connected with US-based, on-premises apps to Azure on another continent—although it may be fine for business continuity.

## Technical implementation and technical capabilities

Figure 3 shows our SAP ERP/ECC production system. Our entire SAP environment is now 100 percent hosted in Azure. 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. We equip 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.

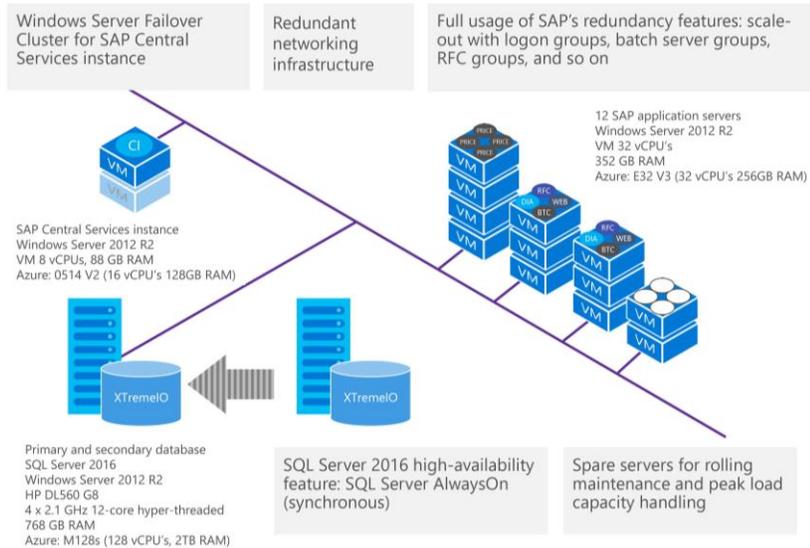


Figure 3. SAP production system in Azure

### 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 connects 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 failover 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 using Azure File Storage and Windows Server storage. And to minimize downtime, we're using virtual server names in Windows Server.

### Azure file storage

At the beginning of our journey to Azure, we were excited about Azure File Storage (files shared in the cloud) and were planning to use it for SAP transport directories. But after we implemented the solution in the first systems, we made the decision not to use it—Azure storage had too many limits on how to access the transport files easily, which made support for SAP transports and troubleshooting difficult. We reverted from Azure File Storage to normal disks.

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.

### Storage Spaces

We're using [Storage Spaces](#) for all systems that require higher I/O and throughput and need to store more data in a single drive on the operating system level.

With Storage Spaces, we can combine multiple virtual hard drives on an Azure virtual machine into a single drive. This helps us to easily grow drives and gives us better performance than a single Azure virtual hard drive. The first implementation of Azure Storage Spaces was our archiving system, where we needed a single 11 TB-drive on Storage Spaces to store intermediary files between two systems. Standard disks, as well as Premium disks, can be used for storage spaces. Depending on performance requirements, we decide what disks should be used; for example, Standard disks for backup spaces and Premium disks for data drives for the SQL Server database.

### Azure features

During our journey to Azure, many new features have been released. For example, Managed Disks became available. With this feature, storage design for higher throughput and I/O is easier and growing disk drives that are attached to virtual machines is simple. We switched our template design to use Managed Disks as soon as the feature was available on Azure. Other features are becoming available all the time. It's important to stay up to date on capabilities to ensure the best performance for applications, where needed.

New features that we implemented after the migration are the following:

- **Accelerated Networking** with up to 25 Gbps of networking throughput. This feature is very useful for virtual machines with high network load.
- **Write Accelerator** offers increased write performance on premium disk drives. We're using this for our large database servers for better performance of HANA log or SQL Server log drives.
- **Load Balancer Standard** provides many additional benefits over the Basic Load Balancer. Examples include more diagnostics to help with daily operations and troubleshooting, High-Availability Ports.

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

### 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 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 knows only 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 a new operating system and install the software in parallel with another machine. Then we move the virtual server name and IP address over and retire the older virtual machine.

This is 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 it for five years on the same operating system until the next server upgrade to avoid the risk and downtime associated with upgrades. Today, with Azure 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 in 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.

## Proven 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. 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 more automation. But our recommendations are to:

- Keep it simple. The configuration in our business continuity site mimics our 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 AlwaysOn. If we're in the US West region, we set up another region like US West 2, and then use SQL Server AlwaysOn 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 targets a higher level than we send to technical teams. We explain our horizontal and vertical strategies, with graphs like the one shown in Figure 2, and burndown charts of how much of the SAP landscape is physical, virtualized on-premises, or Azure, like 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 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.
- **Consider all SAP-related systems.** Make sure SAP-related systems such as tax calculation engines are Azure-certified or have sufficient test periods in your schedule.
- **Archive and compress data.** An Azure migration is a perfect opportunity to push for additional archiving and data compression (on SQL Server, for example) to lower your infrastructure costs in Azure.
- **Technology advances.** Azure technology and available virtual machine sizes and features always advance. Keep up to date with new capabilities and use them to achieve the best possible benefits for your business.

## Looking ahead

We will take advantage of more Azure benefits and share our experiences to help customers do the same. For example, we plan to:

- Focus on continuing optimization of our SAP landscape on Azure by:
  - Expanding scope for "Default snoozing" (VMs that're stopped by default and are started only on demand for a specific period of time) to include more systems to create greater cost savings
  - Using more aggressive tight sizing where we can.
  - Capture Azure resource change history through the API to monitor ongoing cost and usage
- Moving our production systems to Azure Availability Zones (Azure Availability Zones offer better SLA for VMs 99.99% vs 99.95 for Availability Set)

- 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 auto scale out/in and up/down.

## For more information

[microsoft.com/ITShowcase](https://microsoft.com/ITShowcase)

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

[Optimizing SAP for Azure](#)

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

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

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

[Azure Accelerated Networking](#)

[Azure Load Balancer Standard](#)

[Create your Azure free account today](#)

[Azure Availability Zones](#)

[Azure resource change history API](#)

© 2018 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.