

# Sizing SAP workloads on Azure

Including general tips and recommendations for working with the SAP on Azure Pricing calculator to build SAP landscapes on Microsoft Azure

## Summary

This document presents some experiences from recent projects of planning and sizing SAP infrastructure on Azure. The aim is to provide some general rules of thumbs on how to design and size SAP landscapes on Azure, specifically when working with the SAP on Azure Pricing Calculator tool. The document is intended to cover most SAP on Azure scenarios, such as ECC6, SAP HANA, S4/HANA and BW4/HANA. It also has a section for sizing SAP systems using Azure NetApp Files as storage.

The document is offered “as-is” without any warranties or claims to authoritative advice. It is a working document which may or will change with future updates. Some sections may reflect the specific limitations or possibilities at time of writing and become outdated over time. I make no claim to cover all possible scenarios or situations.

If you are new to sizing SAP landscapes on Azure, or do not specifically work with SAP on Azure architecture, it is strongly recommended to seek guidance from people who have the necessary competence – like the SAP on Azure CSA community within Microsoft.

Finally, I don’t consider myself an expert (yet) in these matters. I have worked extensively with the excellent community of peers within Microsoft on the projects that have been used to produce these guidelines, and am forever grateful to them for most of the learnings. My contribution mostly lies in the recording of this knowledge. I hope you find it useful!

Comments & corrections: [trstroem@microsoft.com](mailto:trstroem@microsoft.com)

## Contents

Summary .....	1
Introduction.....	3
Common sizing principles (HANA and non-HANA).....	3
The components of an SAP landscape .....	4
Additional components .....	4
How to use Early Watch reports for sizing .....	5
SAP Application Performance Standard - SAPS.....	6
The 50/50 rule .....	6
Production vs non-production .....	7

Backup sizing .....	8
Sizing of non-HANA SAP systems (AnyDB) .....	9
Starting point.....	9
General recommendations for App/ASCS server sizing .....	9
Using SAPS to size VM's for non-HANA systems .....	9
General recommendations for DB server sizing.....	10
Recommendations for disk sizing.....	10
Recommended VM sizings for specific DBMS brands .....	10
Temporarily up-sizing VMs and disks during migration .....	11
Sizing example – non-HANA .....	11
Sizing of HANA systems .....	12
Starting point.....	12
Recommendations for App server sizing.....	12
Recommendations for DB server sizing .....	12
The 50/50 rule (again) .....	12
Recommendations for disk sizing.....	13
Sizing example - HANA .....	14
Sizing DR systems (reduced size, running QA/test instances on DR VM) .....	14
Downsized/skinny DR .....	14
“Reduced” vs “Skinny” DR – important considerations .....	15
Co-locating QA/test in DR VM .....	15
Saving costs: use of non-certified (“cost-conscious”) VM's for non-critical systems .....	16
Sizing with Azure Netapp Files .....	17
Sizing exercise for a 15 TiB HANA BW scaleout system .....	17
Variant 1: 3 TiB node sizing .....	18
Variant 2: 2 TiB node sizing .....	19
Further documentation on ANF .....	20
General tips and principles.....	20
Further cost-saving ideas .....	22
Use latest-generation VMs.....	22
Snooze idle VMs .....	22
Switch to lower cost storage during snoozing .....	22
Use non-certified VMs for non-critical workloads .....	22
Some words of warning on Standard SSD .....	22
External sources of information and tools .....	23

## Introduction

Sizing of SAP infrastructure on Azure mainly involves two categories of VM's: Application servers/ASCS servers, and DB. For HA/DR solutions, we may also need specific VM's like jumpboxes. Our main sizing efforts are usually focused on the DB servers; these in turn will differ radically between non-HANA (ECC6, or classic AnyDB systems) and HANA (including S/4). HANA systems are usually the simpler ones to size, since the main aim is selecting a VM with a memory similar or larger to the HANA DB footprint. For HANA, SAP recommends the "50/50" rule, stating that HANA VM's should have twice the memory of the net HANA DB size, but this is a practice rarely seen in our "live" sizing exercises, where the focus is as much on making our offer economically edible as practically possible. Add to that our t-shirt sizes for Azure VMs, which some times force us to size up by a considerable degree (eg. for an 8Tb HANA ERP system, you will have to select a 12Tb VM, unless you cannot shrink the DB by archiving or relegating some of the data to cool/cold storage).

The SAP on Azure Calculator tool – now released for partners and customers - will help tremendously with configuring VM's (especially for HANA), including setting up the disk footprint, but you will still need to decide on the size of that VM, and also make some decisions about the remaining VMs to go with your data base. To some extent, the "Config Guess" feature will help, based on the inputs for SAPS, CPU, RAM and so on. However, some basic principles of how to translate (sometimes sparsely articulated) requirements from our customers can be found in this document. The aim is to explain rules of thumb, techniques, and best practices, and how to use these in conjunction with information from EWA or HANA Sizing reports, SAP requirements, and other customer recommendations.

## Common sizing principles (HANA and non-HANA)

When asked to size an SAP landscape, you are usually faced with one of two distinct scenarios: Lift & shift, or migration to HANA (including S/4). Lift & shift is where the client have no intention of changing their setup. This is theoretically simple; just replace existing hardware on-premise or in another cloud with identical SKU's on Azure. However, given that many existing (on-premise) SAP landscapes are oversized (to cater for future growth or just because the hardware vendor managed to oversell), it still makes sense to request Early Watch reports – at least for the productive environments.

Migrations from ECC/AnyDB to HANA or S/4HANA usually provide more room for an innovative approach. HANA Sizing reports are indispensable here, as they more or less eliminate the need for complex conversion formulas to deduce the new DB size (and thereby, memory requirements). Also, our SAP Pricing Calculator has a wide range of pre-configured HANA SKU's complete with suggested disk layout, making this exercise much easier. (Note that Early Watch reports for AnyDB systems make less sense where the aim is to migrate to HANA or S/4, since the nature of a HANA in-memory DB system is fundamentally different).

Customers (or partners) will sometimes provide blueprints of what they would like to implement on Azure. Some times, these are based on existing hardware and do not take into account what is possible within the Azure t-shirt sized VMs (nor correspond to the SAP-certified VMs as per SAP note 1928533). For example, the explicit requirement may be for a HANA DB server with 256Gb memory

and 10 or 15 CPUs. In these cases, it is important to focus on the real capacity requirements (memory, SAPS, number of users etc.) in order to adopt the proposed landscape into Azure.

## The components of an SAP landscape

A minimal SAP installation will normally comprise of the following components:

- A DB server
- One or more Application servers
- An ASCS server (the ABAP Central Services)

For more complex installations, you may need additional servers as needed (Jumpbox, NFS, Java...), but the above is a minimum for a single-node productive installation. For non-productive landscapes, the App and ASCS (even sometimes the DB) can run on the same server (but this depends on system size and usage). In larger systems, you could consider bundling the ASCS component with one of the application servers, thus reducing the total number of servers (but slightly increasing the risk of failure).

In an HA scenario, where you use 2 (or more) DB servers (multi-node), you will have to multiply the above by two (or more). Also, for larger SAP systems (more users, more SAPS requirements – see section on SAPS below) you will probably use several App servers, perhaps across Availability Sets or Zones, in order to increase the availability of the SAP services.

Below is a screenshot from the SAP on Azure calculator with a dual-node productive installation with Disaster Recovery in a second region:

Server Name	SW Comp	Env	Instance Type	SID	CG	Region	No	VM	1	2	3	4	5	6	7	8	9	0	1	2	Model	Util	Azure OS
ECP-DB-HA	DB	PRD	DB	ECP		Region1	2	M208msv2	x												RI3Y	100%	SLES-SAP
ECP-App	App	PRD	App	ECP		Region1	2	D8asv4	x												RI3Y	100%	Linux
ECP-ASCS	ASCS	PRD	ASCS	ECP		Region1	2	D4asv4	x												RI3Y	100%	Linux
ECP-DB-DR	DB	DR	DB	ECP		Region2	1	M208sv2	x												PAYG	1%	SLES-SAP
ECP	Jumpbox	PRD	Other	ECP		Region2	1	E2sv3	x												RI3Y	100%	Linux

Here, we have 2 HANA servers in region 1; these are the HA dual-node DB VMs. In addition, there are 2 app servers, 2 ASCS servers, also in region 1. The DR setup consists of one DB VM which is sized at 50% of the main DB servers, plus one jumpbox. Note that here, no App or ASCS servers are specified for the DR region. This was done to save cost (albeit the cost savings are marginal). The argument is that these VMs will probably not have scarcity issues and can be provisioned as needed during a DR scenario. Also note that to drive cost down, the DR VM for the HANA DB has been set to PAYG with 1% utilization. This means the VM is reserved, but not started, something that does not guarantee its availability – but has been done to reduce cost of the estimation. The customer will need to be informed about this.

In short, you will need to look at the required SAPS to determine the number of application servers. The total number will rarely exceed 3 or 4 per HANA DB; VM SKU will therefore be determined by total number of application SAPS divided by between 1-4: 1 or 2 for non-productive installations, 3 or 4 for production installations).

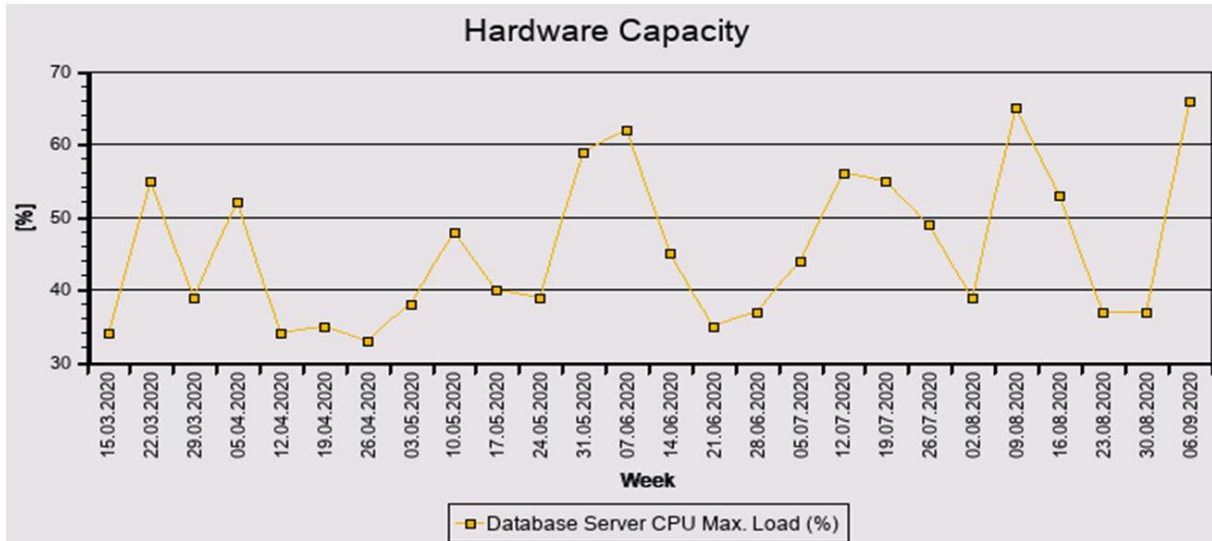
## Additional components

Your landscape on Azure will need additional components, such as gateways, web dispatchers, specific VMs for running workloads such as SAP Fiori and so on. This guide does not take these

components into account. Please make sure you include these components in the full landscape overview – for instance, when using the SAP on Azure pricing calculator.

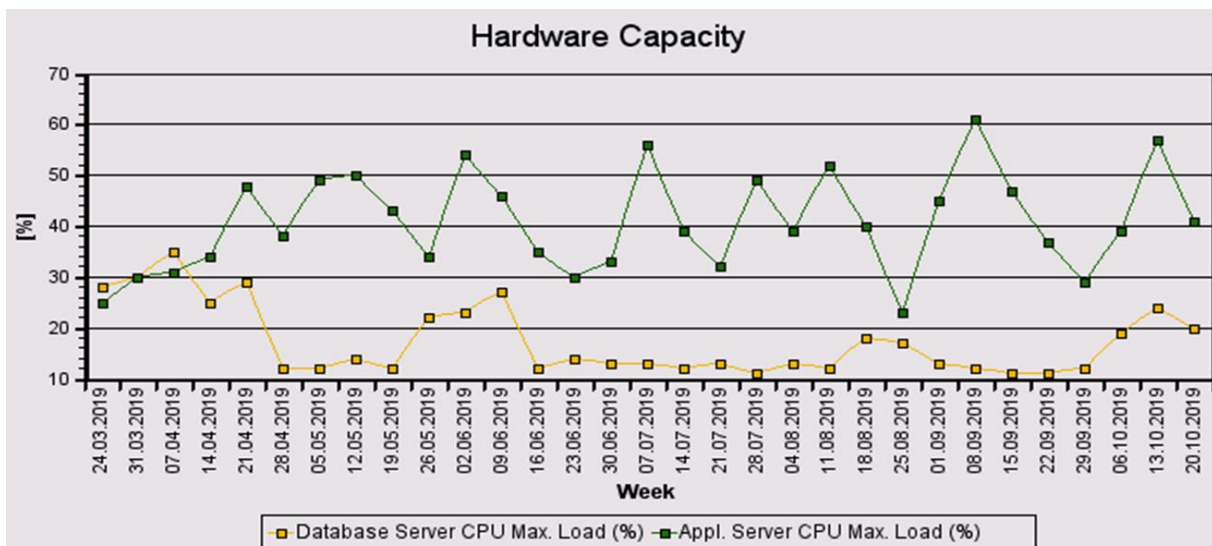
### How to use Early Watch reports for sizing

A useful measurement in the SAP EWA report is the Hardware Capacity section. Here, you can get an understanding of whether the existing hardware is oversized or not. As mentioned elsewhere, the ideal max CPU workload for a DB server according to SAP is around 65%. A system with the below workload would be close to properly sized:



Here, the DB CPU utilization oscillates between ~35% and ~55%, with spikes up towards 70%. The system is correctly sized without being “stressed out” at any time.

Another example shows a system where the DB server is possibly oversized: The average CPU load is around 10-15%, with sporadic spikes up to 30%):



The yellow curve shows the DB CPU load. For this system, we could consider sizing down the DB server. Note that this is a system running on SQLserver; for HANA systems, as mentioned, the server size is more or less given by the memory requirement.

Also note that the EWA report additionally provides information on **user activity** and **peak usage**, which also have to be considered when sizing DB and App servers. A system sized for everyday use may not be able to handle month-end or year-end workloads. Adding temporary App server capacity might be an option, but will require monitoring and some manual action from the customer's side (even if automated – in which case scripts will have to be created and executed). On the DB side, increasing DB VM size temporarily to handle peak workloads is something most customers will shy away from – particularly for production systems. A good starting point would be to opt for “right-sizing” with a PAYG model, then adjust within the first few months of usage and finally locking in reserved instances when the size is “nailed”.

### SAP Application Performance Standard - SAPS

The concept of SAPS is often a central part of the sizing discussions. SAPS stands for SAP Application Performance Standard, and is a measuring unit for computer processing performance in terms of running SAP workloads. 100 SAPS is defined as the capability to fully process 2000 SAP order line items in the timespan of one hour (as you can guess, it was derived from the SAP Sales & Distribution module, more specifically something called the SD Standard Application Benchmark). By “fully process” is meant executing a number of transactional steps, starting with the creation of a sales order, and ending with completing it – something involving a number of ABAP processes. For reference, your typical laptop in 2021 has processing power equivalent to roughly 10.000 SAPS.

According to SAP, using SAPS to derive the hardware needs should provide an accurate sizing. The SAPS number of an existing system is obtained using the SAP QuickSizer tool (link at the start of this document); selecting hardware with similar SAPS capability to the output of the report will (in an ideal world) result in a CPU utilization of ~65% for a productive system ([link to SAP documentation](#)). Note that this may not be ideal; customers may prefer running their CPU at lower levels of utilization.

(Also note that the SAP QuickSizer tool is very difficult to access and use; it requires experience and a large selection of input data. Many customers will be reluctant to run it, or will not even have heard of it).

On the other hand, other customers may very well provide the required SAPS upfront. If so, consider yourself lucky; this can be very valuable help during the sizing work.

The sections below on sizing for both HANA and ECC (non-HANA) landscapes discuss how to use the SAPS data to calculate App and DB server sizes.

### The 50/50 rule

This rule refers to SAP's recommendations to always size the memory for HANA systems twice as big as the net DB size. This would mean providing a 2Tb VM for a 1Tb HANA DB, a 4Tb VM for a 2Tb HANA DB and so on. Although this rule is an SAP recommendation, it may not be wise to follow it blindly. Azure's t-shirt sizes is a factor that may help us offset the impact of the rule; as an example, if the customer's net DB size according to HANA sizing is 4.5Tb, the next size up for Azure is 6Tb. Depending on whether this is an ERP S/4 or a BW system (the latter is more memory-intensive as a general rule), the 6Tb VM should be sufficient. Again, starting out with a smaller, more tight-sized VM is probably a better recommendation than blindly applying the 50/50 rule, and also provides a more attractive financial estimation. Whatever approach is used must of course be discussed and made clear to the customer.

## Production vs non-production

Sizing of non-production systems generally provide room for more flexibility than productive environments. This is also the area where it quickly becomes clear to which extent our customers are “cloud-savvy”, that’s to say, have an understanding of the advantages of the cloud.

Many customers are used to running QA, development or even sandbox systems as copies of production, with the same size and uptime. The tricky part is to convince them to adopt a “cloud mindset”, ie. apply right- and tight-sizing. Many customers, especially those with little or no experience with cloud-based IaaS, are reluctant to apply these approaches, instead focusing on reproducing their on-premise landscapes as closely as possible on Azure – to the extent that they often arrive armed with strict requirements of vCPU’s, memory and storage size. A major challenge in initial discussions is convincing them of the virtues and advantages of right- and tight-sizing. However, this is crucial, since pure lift & shift may not provide the cost savings expected from a move to the cloud.

Further, copying entire production systems to QA/test or even sandbox systems is easy, compared to the slice & dice approach – even if the latter would result in (sometimes radically) smaller footprints for these systems. 3<sup>rd</sup> party products exist to help with this; the challenge is convincing customers that investing in such tools could pay off compared to buying oversized VMs.

One option here could be to suggest using radically downsized SBX, DEV and QA systems for the daily development work; then have a dedicated Stress & Volume test system that is a full copy of production available at certain points in time, for instance during monthly import all testing, to save cost. Set it up as PAYG, and use it for 5 days a month. That’s roughly 50 hours out of 720 – with Pay-As-You-Go, it will cost you roughly 20% of a Reserved Instance the same size. The point is that Azure gives you this flexibility; there is no need to procure a piece of hardware – all you have to do is start a new VM and do a system replication (or restore from a backup), then turn the system off (or wipe it out completely) when no longer needed.

The choice is yours.

Again, if the hardware is your own and you already paid for it, there’s really no incentive for adopting a cloud mindset... 😊

When sizing non-production SAP systems, we should focus on the following ways of saving costs:

- Reduce DB size: this can achieve large cost savings compared to on-premise hardware. Ask the customer to consider deleting obsolete data; perform data cleansing exercises; archiving.
- Reduced uptime: pay-as-you-go models normally becomes more attractive below 230h/month. Customers should consider PAYG at least for the initial stages. If it turns out this is not a viable option, they can later settle on reserved instances.
- Scale compute power as needed: propose sizing up VM’s whenever really needed (like periodic stress & volume tests etc.), or if the system size really requires it.
- Combine DB and App servers into one VM (for non-critical systems like sandbox)
- Alternatively, combine App and ASCS into one VM – but beware of the increased risk of system unavailability. For example, with two app servers, one of which is also running the ASCS services, the chance of your system becoming unavailable if an app server goes down is 50%. With two app servers and a separate ASCS server, you have 67% chance of still reaching your system if one of those 3 servers drop...



- Use E-series VM's (preferably E\*v4) with Premium Storage for non-productive systems. This violates the SAP HANA certifications (which states that E-series should use ANF or Ultra disk for at least the /data/log area), but might still be acceptable for non-prod workloads, even though not yielding the same performance as with ANF/Ultra disk. As always, be transparent when advising customers of such non-certified options.

## Backup sizing

Backup may add considerable costs to a sizing exercise. Although Azure has elaborate built-in backup solutions for SAP, many customers already use 3<sup>rd</sup> party backup solutions like CommVault. Clarifying whether these will be used also on Azure can be vital to saving backup storage costs. The below example from the SAP on Azure pricing calculator shows the difference in cost between two identical HANA systems; both involving 4Tb M128ms VMs, one with Azure HANA backup and the other with 3<sup>rd</sup> party backup selected:

Backup Type	Backup Storage Type	Backup Change Rate	Backup Generations	Backup Size	Storage Capacity	VM Price Total	Storage Price Total	Load Balancer Price	ASR Costs	Backup Costs	Total Price
Azure-HANA	COOL-LRS	5%	28	198.912	15.294	€ 5.515.60	€ 1.565.27	€ 0.00	€ 0.00	€ 4.541.70	€ 11.622.57
3rd party	COOL-LRS	5%	28	17.050	15.294	€ 5.515.60	€ 1.565.27	€ 0.00	€ 0.00	€ 215.67	€ 7.296.54

As can be seen, the backup price for the first VM **almost matches the VM price**, whereas the second VM (where 3<sup>rd</sup> party backup is selected) is radically less expensive overall. Establishing whether customers are using their own backup solutions – or even just **making that assumption** (and clearly stating so in the sizing exercise for transparency) – can greatly reduce the pricing impact.

**Tip:** taking a “conservative” approach – by reducing the backup requirements (backup generations, change rate) will contribute to a reduced overall price. This could make an offer more enticing, although we must of course be completely transparent of how these parameters are set when providing the offer. Also note that the backup requirements for non-productive systems may be less stringent than for productive environments; clarify this with the customer and make sure it is reflected correctly in the SAP on Azure pricing calculator.

The SAP on Azure calculator now includes a “default” backup setting – available by clicking on the “Enter Backup Defaults” button. This simply enters pre-defined values into the backup columns, based on what most customers would select (30 daily backups retained, 2 weekly, 1 monthly and 1 yearly). This is probably “good enough” for most production scenarios, but you (or the customer) may want to scale it down for non-prod workloads. **Please note** that the default setting will apply the same backup rates and retention for all systems - I advise you to scale this down manually for non-productive workloads (eg. Dev, SBX, QA). Also note that productive workloads may need Zone- or Geo-redundant backup storage, not locally redundant as is the default setting. This also impacts pricing – discuss this with the customer upfront and make sure it is agreed which one to use.



## Sizing of non-HANA SAP systems (AnyDB)

### Starting point

For sizing non-HANA systems, we generally require the following information:

- SAPS required (individually for DB and App, or “total SAPS”). See the SAPS section further down for explanation of SAPS.
- DB size on disk
- DB growth rate
- Number of CPU’s currently used for DB and App servers
- Memory (RAM) of App and DB servers

Ideally, SAP’s Early Watch reports should be provided by the customer, for each SAP system involved. These reports contain the above information and also interesting info like system usage patterns and CPU load overviews. This can provide insights into whether the current system is over- or undersized.

### General recommendations for App/ASCS server sizing

Normally, we would recommend 4- or 8-vCPU VMs for the App servers. If App and ASCS runs together on the same VM, opt for 8- vCPU VMs; if not, select 4- or 8- vCPU for the App servers and 2-4 vCPU for the ASCS server. As for the number of App servers, this depends on whether the SAP system is a productive system or not; also, the total user load. If we have the number of application-related SAPS required, this gives us valuable input. Productive systems should be equipped with at least 2 App VMs, except if the system is of lower importance/volume and only based on a single VM (self-healing, 99.9% SLA).

### Using SAPS to size VM’s for non-HANA systems

The recommended number of SAPS can be retrieved from the SAP QuickSizer report. If this document is not available (which is usually the case), we can estimate the SAPS number to be ~300 per active user. Equipped with this number, we can use the below rules to split this between DB and App servers:

- For OLTP systems: 60% SAPS for the DB server, 40% for the App server(s)
- For OLAP systems: 33% for the DB server; 66% for the App server(s)

These formulas are starting points; depending on the SAP system and the nature of work they may need to be modified – however, the SAPS requirements for the DB servers (60% and 33% respectively) should be considered a minimum.

Once the DB SAPS has been determined, we can calculate the disk IOPS requirement. A generally accepted rule seems to be that OLTP systems need IOPS of roughly 0.6 times the DB SAPS, whereas OLAP will need around 0.9 times the DB SAPS. (Note again that DB SAPS is not identical to overall SAPS; see the conversion rule above).

## General recommendations for DB server sizing

For non-HANA servers, the VM size must reflect parameters such as DB size, number of users, and system load. This can be trickier to deduce than when sizing HANA systems. Existing on-premise hardware combined with Early Watch reports is usually a good starting point: start with finding the VM most closely resembling what the customer is currently running, and scale depending on the system load (max CPU usage from EW report). In most cases, this means scaling down to the nearest VM below the number of CPU's currently used – while keeping the RAM to the extent possible.

Example: we have a current AnyDB system running on a VM with 80 CPU's, 1 Tb RAM, DB size 15Tb. The EW report shows daily max CPU load of 5-10%, maxing out at 20% for 2-3 days per month. This is a fairly low system utilization; SAP recommends not going above ~60% CPU load but we're nowhere near this. In this case, it would make sense to look at the "closest" VM SKU *below* these specifications in terms of CPU, with the same memory: we get the M64s with 64 CPU's, 1Tb RAM.

On the other hand, with a CPU load of close to or above the SAP maximum recommendations, we might have opted for one step up: M128s.

Note that the option of "scaling up" means that customers could go live with one VM size in a PAYG model for, say, 1-3 months, and then simply scale up if they find that the VM selected is not up to scratch in terms of horsepower. Also note that, for Reserved Instances, scaling up/down works best (from an economical point of view) when done within the same Instance size flexibility group, since customers are free to "re-configure" their virtual CPU collections as long as they stay inside the same such group. Example: a 64-CPU VM can be "broken up" into 2 32-CPU VM's in the same Instance size flexibility group, or one 32-CPU and 4 8-CPU VM's. This does not work across Instance size flexibility groups.

Reference: [Virtual machine size flexibility -Azure Reserved VM Instances - Azure Virtual Machines | Microsoft Docs](#)

## Recommendations for disk sizing

When sizing disks for non-HANA systems, the following rules of thumb can be used initially:

- /data: roughly same size as DB size, plus 20% overhead
- /log: 50% of /data size for systems less than 512Gb; 512Gb for anything above that size (may want to increase to 1Tb for OLAP scenarios)
- /shared: roughly same size as /log
- /backup: approx. 60-70% of DB size

For more exact sizing, having an idea of the IOPS needed for the SAP system helps in deciding the disk denomination. As an example, the P40 disk has a size of 2Tb whereas the P50 is 4Tb. Both have an IOPS of 7500Mb/s. Thus, attaching 2 P40's to a VM will give you the same physical storage space as one P50, but you get twice the IOPS (15000Mb/s).

In principle, there should always be more than one single disk allocated for /data and /log areas.

## Recommended VM sizings for specific DBMS brands

The following Microsoft document landing page contains sub-chapters for various DBMS sizings on Azure – including specific recommendations for the disk layout of DB2, Oracle and SQL Server:

## [Considerations for Azure Virtual Machines DBMS deployment for SAP workload - Azure Virtual Machines | Microsoft Docs](#)

### Temporarily up-sizing VMs and disks during migration

In order to speed up migration of heterogeneous systems, both VMs and disks can be up-sized. A nice document about this is located [here](#). An article about increasing disk performance by modifying their performance tiers is [here](#). Note that these techniques are to be used during migration phases and will not reflect in the initial sizing of a customer's landscape.

### Sizing example – non-HANA

The below example shows an AnyDB system running on a 64-CPU VM with 512Gb RAM (an E64asv4), with a DB size of 3Tb. The OS and executable disks are sized as P10's (128 Gb each). The /data volume spans 3 P30 disks of 1Tb each. The /log and /shared volumes each consists of one P20 disk of 512Gb, thus following the rules in the previous section. Finally, the /backup volume is 2/3 of the DB size, with 2 P30 disks.

				Storage & Backup											
CPU2	MEM	SAPS		OS Disk		Disk 1 (EXE)		Disk 2 (DATA)		Disk 3 (LOG)		Disk 4 (SHRD)		Disk 5 (BCKP)	
		SAPS VM	SAPS total	Type0	Count0	Type1	Count1	Type2	Count2	Type3	Count3	Type4	Count4	Type5	Count5
64	512	96.700	96.700	P10	1	P10	1	P30	3	P20	1	P20	1	P30	2

## Sizing of HANA systems

### Starting point

If the SAP system we're migrating from is non-HANA ("AnyDB"), the sizing of SAP HANA VM's should come out of the HANA Sizing report, provided by the customer. If this report is not available, the general rule is to look at the existing non-HANA DB size and size the new HANA VM as per the formulas in the Recommendations section below.

### Recommendations for App server sizing

As for non-HANA systems, this is usually in the 4-8 CPU range, depending on the system load and whether we run 2- or 3-tier systems. For 3-tier systems (separate App and ASCS server), we should look at servers with 4 CPUs; for 2-tier systems where App and ASCS is combined in the same VM, we would normally need VMs with 8 CPUs.

The number of App servers will be based on the number of users – the SAPS requirements can help here. Usually, for productive systems, you will need at least two app servers; this is also related to availability SLA's. For non-productive systems where the user load is lower, one single app server is usually selected.

As for the sizing of App servers, see the section above for App server sizing for non-HANA systems.

### Recommendations for DB server sizing

For existing, on-premise HANA systems, the DB VM size should ideally be deduced from the existing HANA server size. We would normally select the next larger size if the DB size falls in between any of the pre-defined Azure VM sizes: for an existing 5Tb HANA DB, this would resolve to a 6Tb VM (M208s).

If the existing system is non-HANA, we should request the HANA Sizing report – it will advise on the future HANA DB size. If this information is not available, we can use the following rules of thumb to deduce the future HANA DB size, then consider applying the 50:50 rule as explained in the next section.

**For ERP/non-BW systems:**  $\text{RAM} = ((\text{data source footprint} * 2) / 7) * c$ , where  $c$  is the source database specific compression factor (this should be provided by the DB Admin). Normally,  $c$  will be somewhere between 1 and 2, probably closer to 2... as a rule of thumb, we can take existing AnyDB size multiplied by **0.6** to find the HANA DB memory size.

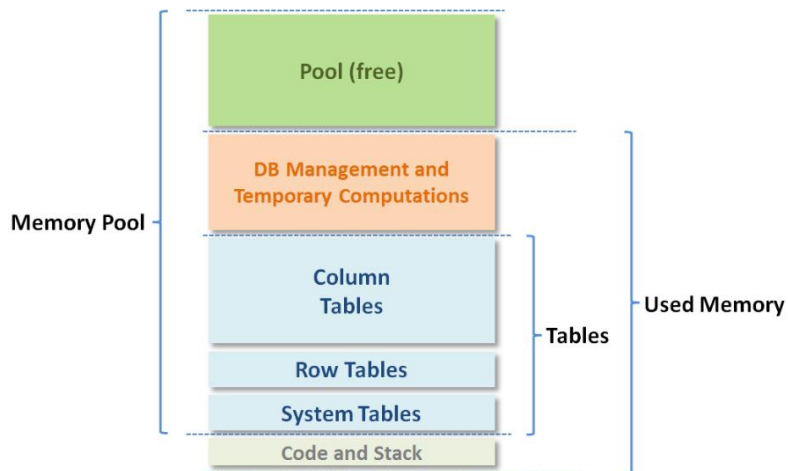
**Example:** for a 4Tb non-HANA DB with a compression factor of ~2, we can calculate HANA memory size as  $((4 * 2) / 7) * 2 = \mathbf{2.3Tb}$ .

**For BW systems:** these usually employ a higher degree of compression; here, the rule of thumb is ~**0.25** times the original (uncompressed) DB size.

### The 50/50 rule (again)

As already mentioned, the general recommendation from SAP is that the server on which HANA runs, should have a memory capacity (RAM) roughly **twice the size of the (compressed) DB size**. This is in order to ensure there is enough space for all of the HANA components; Code & Stack, System, Row

and Column tables, DB management and space for temporary computations, plus finally the memory pool:



### Recommendations for disk sizing

The SAP on Azure Calculator provides functionality for automatically configuring a HANA VM with disk layout, based on the initial memory requirement (DB size). Because of this, there is usually no need to size disks manually. However, if you really want to get your hands dirty, the following document provides sample disk layouts for the various HANA VM sizes: [SAP HANA Azure virtual machine storage configurations - Azure Virtual Machines | Microsoft Docs](#)

If you decide to perform a manual disk sizing, the normal recommendation for net data size on disk is roughly the same as the memory requirement plus 20%. The following general rules might be useful:

- **/data:** take the net disk size number from the HANA sizing report and add 20%.
- **/log:** if the total memory requirement is less than 512Gb, use a minimum of 50% of the memory size for the /log disk. If memory requirement is between 512Gb and 2Tb, use 512Gb. For HANA DB sizes above 2Tb, use 25% of DB size with a minimum of 1Tb. **Note:** there might be special requirements for increasing the log size; for instance, if the system is part of a High Availability setup (2 or more nodes) where other nodes are passive – this means log data is only periodically replicated to the secondary node(s). In this case, the main log disk will need to be larger, based on the temporary buildups of data.
- **/shared:** for single-node HANA: ideally same as memory size – but can be scaled down to 25% of DB size or even below to save cost. For scale-out: 1 times the memory size per 4 worker nodes (minimum the same as memory size). **Example:** in a 6-node system where each node is 2Tb, use 4Tb (2Tb for node 1, plus another 2Tb for the 5<sup>th</sup> node).

As for storage types, most scenarios require Premium disks. Standard SSD is a no-go for most SAP systems (especially HANA) – except for OS and EXE disks, where Standard SSD could be used for cost saving. Ultra disk, on the other hand, could be used when the customer requires specific capacity (IOPS), but this is normally not the case. An exception would be sizing HANA systems with non-certified VM's (see "Saving costs" section below).

## Sizing example - HANA

- 1 Choose VM type for SAP ASCS (\*), Application (based on SAPS) & HANA Database (based on MEM)
- 2 Determine Premium Storage type and # of disks for SAP HANA DB Files, based on requirements (TDI/HWCCT)
- 3 Set up high availability solution for SAP ASCS and SAP HANA Database, based on SLA

(\*) ASCS : ABAP SAP Central Services – part of SAP application servers running message/enqueue service, which is a single point of failure (SPoF)

message/enqueue service, which is a single point of failure (SPoF)

Requirements				Sizing for Azure								
	SAPS needed	Memory Needed	DB Storage	Azure VM Type	# of VMs	All Active ?	SAPS provided	Database File Disk Choice/Layout	Throughput DB Files	Storage size allocation for DB Files	Database Log Disk Choice/Layout	Latency for Log
ASCS (*)	17,000	-		D2_v3	2	Active/Passive	2,189	-				
Application		-		D8_v3	2	All Active	17,420	-				
Database	-	4 TB	TDI Best Practice	M128ms	2	Active/Passive	134,630	5 x P30 disks (1TB, 5000 IOPS/ disk)	1 GBps	5TB	2 x P20 disk (0.5TB, 300 MB/s Throughput)	< 1ms

In the above example, we have the requirements for a 4Tb S/4HANA system with 17000 SAPS for the App server. The DB server size translates as M128ms, whereas two D8\_v3 App servers provide the required SAPS. The ASCS servers have been assigned to the D2\_v3 SKU. On the storage side, Premium SSD is selected, with a total of 5Tb (as mentioned, we take the DB size and add 20%). Total IOPS will be  $5000 \times 5 = 25000$ . For the log files, we select 2 P20 disks totaling 1Tb. For the shared files (not shown), we would initially use the same size as the DB (4Tb), but to bring down costs, we could also scale this down to roughly 25% of the DB size (1Tb in our example).

## Sizing DR systems (reduced size, running QA/test instances on DR VM)

When providing customers with Disaster Recovery (DR) systems, you would normally size these similar to the primary VMs. This is the best way of ensuring (within our 99.9% SLA for a VM) that the DR VM will be available in the DR region in case of a failover.

## Downsized/skinny DR

However, there is an option of downsizing the DR system, which will lower cost. Such a strategy means, of course, that the customer may be forced to upscale the DR VM at short notice – something which may not be possible due to resource scarcity.

When you decide to propose a reduced DR VM in order to reduce cost, make sure the customer is aware of the risk that they may not be able to increase the VM size within an acceptable time window in case of a DR scenario. As many customers will be flocking to the (paired) DR region at the same time, we cannot guarantee the availability of a properly sized VM in such cases. Customers may want to equip their DR emergency procedures with “immediately reserve a properly sized VM” as the very first point on the DR agenda, even before they gather together to “make the go/no-go decision to switch to DR region”, but even this may not be executed in time to guarantee the availability of a properly sized VM. This is an important risk element.

That said, reduced-size DR may save considerable cost over time. Based on the log replication mechanism used, the DR VM size could be half of the main VM, or even lower. “Skinny DR”, with DR VM sizes down towards 10% of the main system, can be achieved by using delta\_datashipping with no preload. In these cases, you can even get away with smaller D-series VM’s. This document from SAP provides some guidance:

[https://help.sap.com/doc/c81e9406d08046c0a118c8bef71f6bdc/2.0.04/en-US/SAP\\_HANA\\_System\\_Replication\\_Guide\\_en.pdf](https://help.sap.com/doc/c81e9406d08046c0a118c8bef71f6bdc/2.0.04/en-US/SAP_HANA_System_Replication_Guide_en.pdf)

#### “Reduced” vs “Skinny” DR – important considerations

**Note:** the above guide also makes it clear (as does Microsoft documentation) that you cannot use both logreplay and delta\_datashipping between nodes in the same HANA system. This means that if the “skinny” DR VM is to be considered, which requires delta datashipping, there can be no synchronous HSR between the two HA nodes in region 1 (using logreplay). However, if logreplay is used for both the HA nodes and the DR node (without preload), a reduced DR node can still be selected – but probably not smaller than 50% of the main (primary) nodes.

Source: [SAP HANA availability across Azure regions - Azure Virtual Machines | Microsoft Docs](#)

#### Co-locating QA/test in DR VM

Another – and slightly less “risky” – approach to saving cost is to size the DR VM properly, and co-host a QA or test system on it (as opposed to having the DR VM just performing the reception of the asynchronously received data from the primary VM). In this scenario, bear in mind the below points, from the following SAP help page:

<https://help.sap.com/viewer/4e9b18c116aa42fc84c7dbfd02111aba/2.0.04/en-US/9d62b8108063497f9d6aab08902b2e04.html>

When running other systems on the secondary system’s hardware, keep in mind the following recommendations:

- We recommend using a separate storage infrastructure for each system. Since the secondary system requires the same I/O capacity as the primary, the additional systems could have a negative impact on the secondary’s I/O.
- The SIDs and instance numbers used for the additional systems running on the secondary hardware must be different from the system replication SID.
- To save memory resources, switch off the preload of tables on the secondary system using global.ini/[system\_replication]-> preload\_column\_tables=false. (The takeover process will take longer as no data is preloaded to memory on the secondary system.)
- The available memory on the secondary system’s hardware must be shared among the systems running there. However, the secondary system should receive the amount of memory it needs. Only the remaining memory can be shared among the DEV or QA systems.

How to calculate the minimum memory requirements for a HANA DR system:

**Operation Mode: delta\_datashipping:**



With preload Off: minimum: 64Gb, or row store size + 20 GB (if this sum is higher)

**Operation Mode: logreplay:**

With preload Off: size of loaded column tables (in-memory) + row store size + 50 GB

(Source: [https://help.sap.com/doc/c81e9406d08046c0a118c8bef71f6bdc/2.0.04/en-US/SAP\\_HANA\\_System\\_Replication\\_Guide\\_en.pdf](https://help.sap.com/doc/c81e9406d08046c0a118c8bef71f6bdc/2.0.04/en-US/SAP_HANA_System_Replication_Guide_en.pdf) )

**Saving costs: use of non-certified (“cost-conscious”) VM’s for non-critical systems**

For non-prod systems (and potentially also lower priority Prod systems), you can use non-certified VM’s such as DSv2 or Ev3/4 series with disk striping for the /hana/data and /hana/log volumes, combining these on the same premium disk set and using E disks for /shared, /root and /user/sap volumes. The following document (which also contains a wealth of other useful info on storage for HANA) describes this option: <https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/hana-vm-operations-storage#cost-conscious-azure-storage-configuration>

An excerpt of the cost-conscious VM configurations is shown here:

A less costly alternative for such configurations could look like:							
VM SKU	RAM	Max. VM I/O Throughput	/hana/data and /hana/log striped with LVM or MDADM	/hana/shared	/root volume	/usr/sap	comments
DS14v2	112 GiB	768 MB/s	4 x P6	1 x E10	1 x E6	1 x E6	Will not achieve less than 1ms storage latency <sup>1</sup>
E16v3	128 GiB	384 MB/s	4 x P6	1 x E10	1 x E6	1 x E6	VM type not HANA certified Will not achieve less than 1ms storage latency <sup>1</sup>

This could be a solution for smaller HANA systems as long as these are not production-critical instances. Please read the above document carefully, particularly with regards to SLA’s, throughput and latency. Also take note of specific disk type requirements - the same document states: "For the HANA certified VMs of the Azure Esv3 family and the Edsv4, you need to use ANF for the /hana/data and /hana/log volume. Or you need to leverage Azure Ultra disk storage instead of Azure premium storage only for the /hana/log volume."

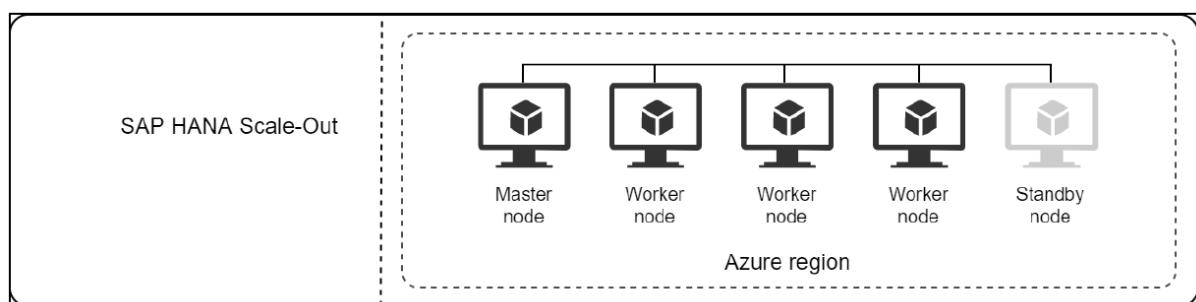
**Note:** if the workload is non-critical, like Dev or Sandbox systems, you can even skip the ANF/Ultra disk requirements (which are there for SAP certification purposes) and go with Premium disks. This, too, has to be communicated properly to the customer before proposed.

## Sizing with Azure Netapp Files

Azure Netapp Files (ANF) is a storage technology currently used mostly for larger systems, especially BW. These systems are often larger than what can be handled with one single VM (scale-up), either for memory or CPU utilization purposes. To alleviate this, we need to apply what is known as “scale-out”: the DB layer is spread out across several VM’s (nodes). This section describes sizing with ANF in such a scale-out scenario.

(Note: ANF can also be used in single-node systems. For now, this section focuses on scale-out scenarios).

One option when designing scale-out systems is to use a “standby” node; this means we achieve a kind of High Availability without using a mirror instance in another Availability Zone:



In order to use standby nodes with HANA on Azure, the use of ANF as storage medium is currently a requirement. The following section outlines a sizing exercise for such a system.

Doc: <https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/sap/sap-hana-high-availability-netapp-files-red-hat>

### Sizing exercise for a 15 TiB HANA BW scaleout system

We have a 15+3Tb system hosted on-premise (5 nodes of 3Tb each, plus one standby node of 3Tb). We would like to build this on Azure, using suitable VM’s and storage.

The system landscape has 5 tiers, with the following DB sizes (standby nodes in parenthesis):

- Prod: 15Tb (+ 3Tb)
- DR: 15Tb (+ 3Tb)
- QA: 15Tb (+ 3Tb)
- SBX: 15Tb (+ 3Tb)
- DEV: 3Tb

On the App server side, there are specific SAPS requirements as follows:

- Prod: 128K
- QA: 64K
- SBX: 8K
- DEV: 8K

### Variant 1: 3 TiB node sizing

(Note: for the sake of simplification, all sizes below are given as MiB and TiB, not MB/TB).

The starting point for all sizing using ANF is SAP's recommendations for storage throughput characteristics as follows:

- Enable read/write on /hana/log of at least 250 MiB/sec with 1 MiB I/O sizes
- Enable read activity of at least 400 MiB/sec for /hana/data for 16 MiB and 64 MiB I/O sizes
- Enable write activity of at least 250 MiB/sec for /hana/data with 16 MiB and 64 MiB I/O sizes

The Azure NetApp Files throughput limits per 1 TiB of volume quota are:

- Premium Storage Tier - 64 MiB/s
- Ultra Storage Tier - 128 MiB/s

Based on the above, we get the following best practices (as provided by ANF and SAP):

For /hana/log, we divide the 250 MiB/sec requirement from SAP by 64 MiB/s (which is the capacity of a 1 TiB volume) to get the number of 1TiB volumes we need: this is roughly 4 TiB (each 1 TiB will facilitate a throughput of 64 MiB/s and we need a total of 250 MiB/s). This is if using Premium storage; if we decide to go with Ultra SSD, we have a capacity of 128 MiB/s per 1 TiB volume, which means we can go with a 2 TiB volume (250 divided by 128).

For /hana/data, we have a read throughput requirement from SAP of 400 MiB/s. This means that for Premium SSD, we need a quota of  $(400 / 64) = 6.3$  TiB ANF volumes (the write request is lower; only 250 MiB/s, but we have to calculate based on the highest requirement which is the read throughput). For Ultra SSD, we arrive at ~3.2 TiB (half the size, since Ultra SSD has twice the throughput of Premium SSD).

If we relate the above to our sample BW system outlined earlier, we can make the following calculations (the below is edited from an e-mail exchange with one of our NetApps specialists):

To achieve 400 MiB/s throughput for HANA data, we require 3.2 TiB ANF Ultra (ANF-U):

- We oversize this to 4.5 TiB per node to store snapshots in-place (for faster restore) and also to guarantee some additional backup throughput for offloading backup to Azure blob
- No oversizing needed for SBX and DEV since we don't need to achieve the HANA KPIs there (this means we stay with the size of 3.2 TiB, not 4.5)
- SBX will use ANF Ultra to be easily scaled up to HANA KPIs if needed for support requests
- DEV will use ANF Premium for cost savings

To achieve 250MB/s throughput for HANA log, we require 2.0TB ANF Ultra (ANF-U):

- SBX and DEV do not need to achieve HANA KPIs, which means we can go with lower size volume; 0.5 TiB is OK here
- SBX uses ANF Ultra to be easily scaled up to HANA KPIs if needed for support requests
- DEV uses ANF Premium for cost savings

SAP SHRD should be 1x the memory for every 4 nodes, using ANF Premium (ANF-P)

- HANA DB PRD => 6TB (we have 5 nodes, so 3Tb for node 1 + another 3Tb for node 4+1)
- DR HANA DB => 6TB (same as for PRD)
- HANA DB QUAL => 6TB (same as for PRD)

- SBX => 6TB (same as for PRD)
- DEV => 3TB (only 1 node)

Backup volume twice the database size using ANF Standard (ANF-S)

- Used for file-based backups
- Should be scheduled regularly for an integrity-check
- If no backup is needed for an environment, Disk 5 can be removed from the line

### Variant 2: 2 TiB node sizing

The above calculations were made for a BW HANA scale-out system of 15 TiB in 5 + 1 nodes of 3 TiB each (total: 15 + 3 TiB which gives us 18 TiB including the scale-out node). This means the use of 6 M208sv2 VM's. Another variant – which might save cost – is to size the system differently using the 2 TiB M128s VM's. To keep the size requirement (15 TiB), we arrive at 8 + 1 nodes (16 TiB for the main DB plus one standby node). This has certain impacts on the size calculations, some of which may not be too obvious. Most important to notice is that the quota requirements for the /hana/log volume will go up, since we are required (in order to meet SAP's throughput KPI's) to have 2 TiB available for each node (because each node has its own log area). Since we now have a total of 8 production nodes (we don't count the standby node) instead of 5, this means a need for 16 TiB of /hana/log volume as compared with only 10 TiB for the M208sv2 landscape variant. The following table compares the two scenarios:

M208sv2	M128s	Environment	Storage M208sv2	Storage M128s
5+1	8+1	PRD (KPI)	5x 4.5TiB Data (ANF-U) 5x 2.0TiB Log (ANF-U) 1x 6TiB SHRD (ANF-P)  ⇒ 32.5TiB ANF-U ⇒ 6.0TiB ANF-P	8x 3.2TiB Data (ANF-U) 8x 2.0TiB Log (ANF-U) 1x 4TiB SHRD (ANF-P)  ⇒ 41.6 TiB ANF-U (28% more) ⇒ 4.0 TiB ANF-P (33% less)
5+0	8+0	DRSYNC (KPI)	5x 4.5TiB Data (ANF-U) 5x 2.0TiB Log (ANF-U) 1x 6TiB SHRD (ANF-P)  ⇒ 32.5TiB ANF-U ⇒ 6.0TiB ANF-P	8x 3.2TiB Data (ANF-U) 8x 2.0TiB Log (ANF-U) 1x 4TiB SHRD (ANF-P)  ⇒ 41.6 TiB ANF-U (28% more) ⇒ 4.0 TiB ANF-P (33% less)
5+1	8+1	QUAL (KPI)	5x 4.5TiB Data (ANF-U) 5x 2.0TiB Log (ANF-U) 1x 6TiB SHRD (ANF-P)  ⇒ 32.5TiB ANF-U ⇒ 6.0TiB ANF-P	8x 3.2TiB Data (ANF-U) 8x 2.0TiB Log (ANF-U) 1x 4TiB SHRD (ANF-P)  ⇒ 41.6TiB ANF-U (28% more) ⇒ 4.0TiB ANF-P (33% less)
5+0	8+0	SBX (no KPI)	5x 3.0TiB Data (ANF-U) 5x 0.5TiB Log (ANF-U) 1x 6TiB SHRD (ANF-P)  ⇒ 17.5 TiB ANF-U ⇒ 6.0 TiB ANF-P	8x 2.0TiB Data (ANF-U) 8x 0.5TiB Log (ANF-U) 1x 4TiB SHRD (ANF-P)  ⇒ 20.0TiB ANF-U (14% more) ⇒ 4.0TiB ANF-P (33% less)
1+0	1+0	DEV (no KPI)	1x 3.0TiB Data (ANF-P) 1x 0.5TiB Log (ANF-P) 1x 3TiB SHRD (ANF-P)  ⇒ 3.5 TiB ANF-P ⇒ 3 TiB ANF-P	1x 2.0TiB Data (ANF-P) 1x 0.5TiB Log (ANF-P) 1x 2TiB SHRD (ANF-P)  ⇒ 2.5 TiB ANF-P (28% less) ⇒ 2 TiB ANF-P (33% less)

Total	⇒ 115.0 TiB ANF-U	⇒ 147.3 TiB ANF-U (28% more)
	⇒ 30.5 TiB ANF-P	⇒ 20.5 TiB ANF-P (33% less)

If we need to calculate additional throughput for offloading the backups, the required storage for ANF-U increases even more, but temporarily resizing may also be an option here.

## Further documentation on ANF

NetApp TR-4746: <https://aka.ms/tr-4746>

HANA Snapshot Backup: <https://blog.netapp.com/azure-netapp-files-backup-sap-hana>

## General tips and principles

The very first thing (and often, the only thing) a customer will notice is the total cost of ownership of the SAP on Azure solution. In terms of infrastructure, this means the price of VMs and storage. Explaining to a customer (after an initial not-so-positive response to the IaaS price) that our offer is “bullet-proof” in terms of adequate sizing, performance, SLAs and DR capabilities might not make much of a difference as long as the only thing the customer wants to focus on is a price comparison with other providers. That these other providers some times make “assumptions” and shortcuts won’t help you if you fail to pass the initial hurdle. Hence, we should strive to make the best possible offer both in terms of quality and cost.

This section outlines some principles aimed at reducing cost by applying some of the tools provided by cloud flexibility. It is important to know the impact of such decisions, especially those that might alter important factors like RTO or SLA’s, and be fully transparent about them to the end customer. Nevertheless, you should know about them and carefully consider which ones might be relevant for your scenarios. It might be better to start out with a “bare bones” estimate, then add additional features during further discussions with the customer. Again, always state these assumptions up front and explain why they are selected.

- For HANA (including S/4) systems, the DB size will be the main factor to determine the VM size. Example: HANA DB size of 2.8Tb translates as M208sv2 (3 Tb). Normally, for initial estimates, we ignore the SAP 50/50 rule in order to land on the minimum viable VM size.
- SAP’s Early Watch report usually provides sufficient information for sizing (most customers will not be able/willing to use the much more complicated QuickSizer tool). Look for maximum CPU utilization of various servers to judge whether the existing system is over-sized. Average CPU utilization below ~10-20% means there is potential for reducing the size or number of servers.
- For migrations from anyDB to HANA, the HANA Sizing Report is indispensable. If not available, a very rough rule of thumb is: HANA DB size = ~ 60% of existing non-HANA DB size.
- When sizing Disaster Recovery (DR) HANA DB VM’s, you can usually opt for 50% of the size of the primary VM’s – regardless of the HSR method used. This is because the DR VM will normally only be used for a short period of time and not require all the HANA data to be loaded into memory. Note however that, during a failover, if the DR region is used for a

longer period of time (days or weeks), this will not be adequate. Always be transparent with the customer about such sizing and ensure there is mutual agreement.

- For HANA System Replication using delta\_datashipping, Disaster Recovery (DR) VM's can be minimized ("skinny DR") to save money (we have seen use cases where these HANA VM's have been sized down towards 10% of the main VM). Note that this will increase the RTO since the system will need to be upsized in case of a region failover. Customers must be informed of this potential risk. In case of a DR scenario, it would make sense to immediately increase the size of the DR VM – even before deciding whether to switch to the secondary region. Yes, you have to pay for that VM for a few hours even if you don't switch, but at least you have a chance of actually getting it...
- For smaller HANA systems, running multiple HANA instances on the same VM (HANA multi-tenant) might be an option. This is more complex technically, but can save money (especially if the HANA DB's are not matching our t-shirt VM sizes).
- Customers should consider archiving and deleting obsolete data. This can have a considerable impact on DB (and therefore HANA VM) size.
- A key point of the Cloud Mindset is flexibility. Systems can be scaled both up and down. If customers insist on one common size for all of their DEV/SBX/TST/PROD environments, one suggestion might be to start with full-sized PAYG VM for the non-PROD systems for a limited number of months, then scale down and move to reserved instances once the installation "stabilizes", eg. in 1-3 months' time.
- Keep in mind that customers are buying VM capacity (basically, a given number of CPU's) on Azure, not individual VM's. This can be important when sizing up/down – and even provide a way to change the size of reserved instances. VMs can be merged – or split – within the same family. This means a 32 CPU VM can be split into 2 VMs, each with 16 CPUs, or 4 VMs of 8 CPUs each. Likewise, 4 VMs of 16 CPU's each can be combined into one VM of 32 CPUs. This means that customers may change their VM sizes – even for reserved instances – as long as they stay within the same VM family. This adds flexibility to right- and tight-sizing scenarios.
- Stay within a limited number of VM families (see the previous point about flexibility). This simplifies the overall architecture and ensures customers can re-allocate their capacity by breaking up or merging VM's. Also look at using the latest generation SKU's (note that the SAP on Azure calculator does not necessarily provide these SKU's automatically; for example, when setting up an App server it might give you a D\*sv3. Amend this by simply changing the SKU afterwards – for instance to a D\*v4 or V5).
- Ask customers upfront about their preferred backup solution. Some will already be using 3<sup>rd</sup> party backup like Commvault. This will reduce the estimation amount – backup costs can be removed from the calculator.
- Select geographically paired regions – and always aim to provide customers with the least expensive region that makes sense for them, based on factors such as SKU availability, latency and legal/organizational requirements. As an example, Europe West is 10-15% more expensive than Europe North for certain VM's, whereas smaller regions like Switzerland can be even 20% more expensive than larger, more established ones. "Flipping regions" (eg. switching from Europe West to Europe North as primary region – or even placing non-production systems in the secondary (cheaper) region) can result in tangible price reductions. As an example, we had a customer using Europe West as primary, while Europe North was secondary (DR) region. Re-assigning the DEV, SBX and QA systems to the secondary region reduced overall costs by 10%, while the increase in latency was negligible. Storage prices, on the other hand, are relatively stable across regions.

## Further cost-saving ideas

### Use latest-generation VMs

Usually, when we switch from one generation VM to the next (example: D\*v3 to D\*v4), there is also a price drop. Keep this in mind (the SAP on Azure calculator may not always be updated in time).

### Snooze idle VMs

Snooze VMs that are not used. For some systems, like sandbox or volume testing, challenge customers to consider the PAYG model with regular up/downtime windows (example: uptime 9 hours/day, 5 days/week translates as 26%. This is already less expensive than RI3Y).

### Switch to lower cost storage during snoozing

Change disk types to less expensive storage during VM snoozing. For example, switch Premium disks to Standard SSD in order to save cost. This is only possible during VM downtime, but is a simple and effective way to further reduce cost. Documentation: [Convert managed disks storage between different disk types by using Azure PowerShell - Azure Virtual Machines | Microsoft Docs](#)

### Use non-certified VMs for non-critical workloads

Use E\*v4 series VMs with Premium SSD for non-critical (HANA) workloads. These will not be certified by SAP for productive HANA use (for that, you need Ultra disk storage for at least the /log area), but will still be considerably cheaper than the matching M-series VM.

### Some words of warning on Standard SSD

Beware of using Standard SSD, even for OS or exe disks. Standard SSD has a “flat” performance; all sizes have identical IOPS and throughput characteristics, which – for the smaller sizes – supersedes the Premium SSD. Thus, it might be enticing using these for smaller workloads. However, keep the following points in mind: performance is not guaranteed, like for Premium SSD. Also, Standard SSD cannot burst – something easily overlooked. Adding to this, Standard SSD has transaction costs, whereas Premium does not. To add to this, the performance guarantees of Standard SSD is (apparently) related to time-slicing, meaning that if a Standard SSD reaches the limit of 500 IOPS in less than one second, it will “freeze” until the next second starts. This could mean intermittent “interrupts” of activity, to keep with the limit of 500 IOPS/second... some customers have apparently reported issues due to this. Finally, our SLA of 99.9% for a VM is only valid when (at least) Premium SSD is used for all disks.



## External sources of information and tools

Note: some of the following resources are provided by 3<sup>rd</sup> parties. Use at your own discretion.

[Nick Morgan and Bartosz Jankowski's SAP on Azure Implementation Guide](#)

[Microsoft SAP on Azure Architecture documentation](#)

[SAP: Certified and supported hardware directory](#)

[SAP QuickSizer \(requires OSS login\)](#)

[SAP official Sizing landing page](#)

[SAP Blog: the ultimate guide to effective sizing of SAP HANA](#)

[SAP document: SAP HANA Storage Requirements](#)

[OSS Note 1928533: Supported VMs for SAP on Azure](#)

[New sizing report for SAP BW/4HANA](#)

[ABAP on HANA sizing report](#)

[Sizing Approaches for SAP HANA – Lessons Learned](#)

[SAP HANA Memory usage explained](#)

[Migrating SAP on Oracle workloads to Azure](#)

[SAP HANA on Azure VM and Storage configurations](#)

[SAP HANA System Replication](#)

[SAP HANA availability across Azure regions - Azure Virtual Machines | Microsoft Docs](#)

[Optimizing SAP for Azure](#)

[SAP Migrations to Azure – Performance Optimization Guide](#)