

SAP NetWeaver on Microsoft Azure Virtual Machine Services – Planning and Implementation Guide

Microsoft Corporation

Version: 3.00

Date of last change: 07/30/2015

Abstract

Microsoft Azure enables companies to acquire compute and storage resources in minimal time without lengthy procurement cycles. Azure Virtual Machines allow companies to deploy classical applications, like SAP NetWeaver based applications into Azure and extend their reliability and availability without having further resources available on-premises. Azure Virtual Machine Services also supports cross-premises connectivity, which enables companies to actively integrate Azure Virtual Machines into their on-premises domains, their Private Clouds and their SAP System Landscape.

This white paper describes the fundamentals of Microsoft Azure Virtual Machine and provides a walkthrough of planning and implementation considerations for SAP NetWeaver installations in Azure and as such should be the document to read before starting actual deployments of SAP NetWeaver on Azure.

The paper complements the SAP Installation Documentation and SAP Notes which represent the primary resources for installations and deployments of SAP software on given platforms.

Copyright Information

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet website references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Microsoft, Active Directory, Hyper-V, SQL Server, Windows PowerShell, Windows, Microsoft Azure and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

© 2014 Microsoft Corporation. All rights reserved.

Contents

[1 Summary 7](#_Toc426396883)

[1.1 Definitions upfront 7](#_Toc426396884)

[1.2 Resources 8](#_Toc426396885)

[2 Possible Scenarios 10](#_Toc426396886)

[2.1 Cloud-Only - Virtual Machine deployments into Azure without dependencies on the on-premises customer network 12](#_Toc426396887)

[2.2 Cross-Premise - Deployment of single or multiple SAP VMs into Azure with the requirement of being fully integrated into the on-premises network 14](#_Toc426396888)

[2.3 Supported OS and Database Releases 15](#_Toc426396889)

[3 Microsoft Azure Virtual Machine Services 16](#_Toc426396890)

[3.1 Azure Regions 17](#_Toc426396891)

[3.2 The Microsoft Azure Virtual Machine Concept 18](#_Toc426396892)

[3.2.1 Fault Domains 18](#_Toc426396893)

[3.2.2 Upgrade Domains 18](#_Toc426396894)

[3.2.3 Azure Availability Sets 19](#_Toc426396895)

[3.2.4 Azure Affinity Groups – Changed recommendations!!! 19](#_Toc426396896)

[3.3 Storage: Microsoft Azure Storage and Data Disks 19](#_Toc426396897)

[3.3.1 Azure Standard Storage 21](#_Toc426396898)

[3.3.2 Azure Premium Storage 21](#_Toc426396899)

[3.3.3 Azure Storage Accounts 22](#_Toc426396900)

[3.4 Microsoft Azure Networking 23](#_Toc426396901)

[3.4.1 Cloud Services and Virtual Networks 23](#_Toc426396902)

[3.4.2 Site-to-Site Connectivity 26](#_Toc426396903)

[3.4.3 Point-to-Site VPN 27](#_Toc426396904)

[3.4.4 Multi-Site VPN 27](#_Toc426396905)

[3.4.5 VNet to VNet Connection 28](#_Toc426396906)

[3.4.6 Private Connection to Azure – ExpressRoute 28](#_Toc426396907)

[3.4.7 Forced tunneling in case of Cross-Premise 28](#_Toc426396908)

[3.4.8 Summary of Azure Networking 29](#_Toc426396909)

[3.5 Quotas in Azure Virtual Machine Services 29](#_Toc426396910)

[4 Managing Azure Assets 33](#_Toc426396911)

[4.1 Microsoft Azure Portal 33](#_Toc426396912)

[4.2 Management via Microsoft Azure PowerShell cmdlets 35](#_Toc426396913)

[5 Different ways to deploy VMs for SAP in Azure 36](#_Toc426396914)

[5.1 Deployment of VMs for SAP 36](#_Toc426396915)

[5.2 Preparing VMs with SAP for Azure 37](#_Toc426396916)

[5.2.1 Preparation for moving a VM from on-premises to Azure with a non-generalized disk 37](#_Toc426396917)

[5.2.2 Preparation for deploying a VM with a customer specific image for SAP 38](#_Toc426396918)

[5.3 Transferring VMs and VHDs between on-premises to Azure 39](#_Toc426396919)

[5.3.1 Difference Between an Azure Disk and Azure Image 41](#_Toc426396920)

[5.3.2 Uploading a VHD from on-premises to Azure 41](#_Toc426396921)

[5.3.3 Deployment of a VM Image 42](#_Toc426396922)

[5.3.4 Downloading VHDs to on-premises 42](#_Toc426396923)

[5.4 Transferring VMs and VHDs within Azure 43](#_Toc426396924)

[5.4.1 Copying SAP systems within Azure 43](#_Toc426396925)

[5.4.2 Copying disks between Azure Storage Accounts 44](#_Toc426396926)

[5.5 Disk Handling 45](#_Toc426396927)

[5.5.1 VM/VHD structure for SAP deployments 45](#_Toc426396928)

[5.5.2 Disk Handling 47](#_Toc426396929)

[5.5.3 Setting automount for attached disks 48](#_Toc426396930)

[5.6 Final Deployment 48](#_Toc426396931)

[6 Accessing SAP systems running within Azure VMs 49](#_Toc426396932)

[6.1 Remote Access to SAP systems 49](#_Toc426396933)

[6.1.1 Configuration of the SAP System and SAP GUI connectivity for Cloud-Only scenario 50](#_Toc426396934)

[6.1.2 Changing Firewall Settings within VM 50](#_Toc426396935)

[6.1.3 Security recommendations 51](#_Toc426396936)

[6.2 Connecting SQL Server Graphical User Interface Tools to SQL Server in Azure VMs 51](#_Toc426396937)

[7 Concepts of Cloud-Only deployment of SAP instances 52](#_Toc426396938)

[7.1 Single VM with SAP NetWeaver demo/training scenario 52](#_Toc426396939)

[7.2 Implement a set of VMs which need to communicate within Azure 54](#_Toc426396940)

[7.2.1 Cloud Service and Virtual Machine naming 54](#_Toc426396941)

[7.2.2 Setup Network for communication between the different VMs 54](#_Toc426396942)

[7.2.3 Gateway/Endpoint configuration 58](#_Toc426396943)

[8 Deploying SAP VMs with Corporate Network Connectivity (Cross-Premises) 59](#_Toc426396944)

[8.1 Scenario of an SAP landscape 59](#_Toc426396945)

[8.1.1 Security considerations 60](#_Toc426396946)

[8.2 Dealing with different Virtual Machine Series 61](#_Toc426396947)

[8.2.1 Printing on a local network printer from SAP instance in Azure 61](#_Toc426396948)

[8.2.2 Integration of SAP Azure Systems into Correction and Transport System (TMS) in Cross-Premises 62](#_Toc426396949)

[8.2.3 Including SAP Systems in the Transport Domain 63](#_Toc426396950)

[8.2.4 RFC traffic between SAP instances located in Azure and on-premises (Cross-Premises) 65](#_Toc426396951)

[8.2.5 Accessing ‘local’ fileshares from SAP instances located in Azure or vice versa 65](#_Toc426396952)

[9 Supportability 66](#_Toc426396953)

[9.1 Azure Monitoring Solution for SAP 66](#_Toc426396954)

[9.1.1 Solution design 66](#_Toc426396955)

[9.2 Integration of Azure located SAP instance into SAProuter 68](#_Toc426396956)

[10 SAP NetWeaver AS Java 70](#_Toc426396957)

[10.1 SAP Enterprise Portal 70](#_Toc426396958)

[11 High Availability (HA) and Disaster Recovery (DR) for SAP NetWeaver running on Azure Virtual Machines 72](#_Toc426396959)

[11.1 Overview 72](#_Toc426396960)

[11.2 Overview of High Availability 72](#_Toc426396961)

[11.3 Azure Infrastructure High Availability 72](#_Toc426396962)

[11.3.1 Virtual Machine (VM) High Availability 73](#_Toc426396963)

[11.3.2 Azure Storage Redundancy 73](#_Toc426396964)

[11.3.3 Utilizing Azure Infrastructure VM Restart to Achieve “Higher Availability” of SAP Applications 73](#_Toc426396965)

[11.4 SAP Application High Availability on Azure IaaS 75](#_Toc426396966)

[11.4.1 High Availability for SAP Application Servers 75](#_Toc426396967)

[11.4.2 High Availability for the SAP (A)SCS instance 76](#_Toc426396968)

[11.4.3 High Availability for the SAP database instance 77](#_Toc426396969)

[11.4.4 End-to-End High Availability for the Complete SAP System 78](#_Toc426396970)

[11.5 Using Autostart for SAP instances 79](#_Toc426396971)

[11.6 Larger 3-Tier SAP systems 80](#_Toc426396972)

[11.6.1 Location of 3-Tier SAP configurations 80](#_Toc426396973)

[11.7 Offline Backup of SAP systems 80](#_Toc426396974)

[11.8 Online backup of an SAP system 81](#_Toc426396975)

[11.9 Azure as DR site for production SAP landscapes 81](#_Toc426396976)

[11.10 Summary 81](#_Toc426396977)

# Summary

Cloud Computing is a widely used term which is gaining more and more importance within the IT industry, from small companies up to large and multinational corporations.

Microsoft Azure is the Cloud Services Platform from Microsoft which offers a wide spectrum of new possibilities. Now customers are able to rapidly provision and de-provision applications as Cloud-Services, so they are not limited to technical or budgeting restrictions. Instead of investing time and budget into hardware infrastructure, companies can focus on the application, business processes and its benefits for customers and users.

With Microsoft Azure Virtual Machine Services, Microsoft offers a comprehensive **I**nfrastructure **a**s **a** **S**ervice (IaaS) platform. SAP NetWeaver based applications are supported on Azure Virtual Machines (IaaS). This whitepaper will describe how to plan and implement SAP NetWeaver based applications within Microsoft Azure as the platform of choice.

The paper itself will focus on two main aspects:

* The first part will describe two supported deployment patterns for SAP NetWeaver based applications on Azure. It will also describe general handling of Azure with SAP deployments in mind.
* The second part will detail implementing the two different scenarios described in the first part.

For additional resources see [chapter 1.2](#_Resources) in this document.

## Definitions upfront

Throughout the document we will use the following terms:

* IaaS: Infrastructure as a Service.
* PaaS: Platform as a Service.
* SaaS: Software as a Service.
* SAP Component: an individual SAP application such as ECC, BW, Solution Manager or EP.  SAP components can be based on traditional ABAP or Java technologies or a non-NetWeaver based application such as Business Objects.
* SAP Environment: one or more SAP components logically grouped to perform a business function such as Development, QAS, Training, DR or Production.
* SAP Landscape: This refers to the entire SAP assets in a customer’s IT landscape. The SAP landscape includes all production and non-production environments.
* SAP System: The combination of DBMS layer and application layer of e.g. an SAP ERP development system, SAP BW test system, SAP CRM production system, etc. In Azure deployments it is not supported to divide these two layers between on-premises and Azure. This means an SAP system is either deployed on-premises or it is deployed in Azure. However, you can deploy the different systems of an SAP landscape into either Azure or on-premises. For example, you could deploy the SAP CRM development and test systems in Azure but the SAP CRM production system on-premises.
* Cloud-Only deployment: A deployment where the Azure subscription is not connected via a site-to-site or ExpressRoute connection to the on-premises network infrastructure. In common Azure documentation these kinds of deployments are also described as ‘Cloud-Only’ deployments. Virtual Machines deployed with this method are accessed through the internet and public internet endpoints assigned to the VMs in Azure. The on-premises Active Directory (AD) and DNS is not extended to Azure in these types of deployments. Hence the VMs are not part of the on-premises Active Directory. **Note: Cloud-Only deployments in this document is defined as complete SAP landscapes are running exclusively in Azure without extension of Active Directory or name resolution from on-premises into public cloud. Cloud-Only configurations are not supported for production SAP systems or configurations where SAP STMS or other on-premises resources need to be used between SAP systems hosted on Azure and resources residing on-premises.**
* Cross-Premises: Describes a scenario where VMs are deployed to an Azure subscription that has site-to-site, multi-site or ExpressRoute connectivity between the on-premises datacenter(s) and Azure. In common Azure documentation, these kinds of deployments are also described as Cross-Premises scenarios. The reason for the connection is to extend on-premises domains, on-premises Active Directory and on-premises DNS into Azure. The on-premises landscape is extended to the Azure assets of the subscription. Having this extension, the VMs can be part of the on-premises domain. Domain users of the on-premises domain can access the servers and can run services on those VMs (like DBMS services). Communication and name resolution between VMs deployed on-premises and Azure deployed VMs is possible. This is the scenario we expect most SAP assets to be deployed in. See more information here: <http://msdn.microsoft.com/en-us/library/azure/jj156075.aspx>. N**ote: Cross-Premises deployments of SAP systems where Azure Virtual Machines running SAP systems are members of an on-premises domain are supported for production SAP systems. Cross-Premises configurations are supported for deploying parts or complete SAP landscapes into Azure. Even running the complete SAP landscape in Azure requires having those VMs being part of on-premises domain and ADS. In former versions of the documentation we talked about Hybrid-IT scenarios, where the term ‘Hybrid’ is rooted in the fact that there is a cross-premises connectivity between on-premises and Azure. Plus, the fact that the VMs in Azure are part of the on-premises Active Directory.**

Some Microsoft documentation describes Cross-Premises scenarios a bit differently, especially for DBMS HA configurations. In the case of the SAP related documents, the Cross-Premises scenario just boils down to having a site-to-site or private (ExpressRoute) connectivity and the fact that the SAP landscape is distributed between on-premises and Azure.

## Resources

The following additional guides are available for the topic of SAP deployments on Azure:

* SAP NetWeaver on Microsoft Azure Virtual Machine Services - Deployment Guide
* DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services
* SAP NetWeaver: Building a Microsoft Azure–based Disaster Recovery Solution
* Clustering SAP ASCS Instance using Windows Server Failover Cluster on Azure with SIOS DataKeeper

The guides can be downloaded under the section ‘SAP’ here: <http://go.microsoft.com/fwlink/p/?LinkId=397566>

**Wherever possible a link to the referring SAP Installation Guide is used (Reference InstGuide-01, see** [**http://service.sap.com/instguides**](http://service.sap.com/instguides)**). When it comes to the prerequisites and installation process, the SAP NetWeaver Installation Guides should always be read carefully, as this document only covers specific tasks for SAP NetWeaver systems installed in a Microsoft Azure Virtual Machine.**

The following SAP Notes are related to the topic of SAP on Azure:

|  |  |
| --- | --- |
| Note number | Title |
| 1928533 | SAP Applications on Azure: Supported Products and Sizing |
| 2015553 | SAP on Microsoft Azure: Support Prerequisites |
| 1999351 | Enhanced Azure Monitoring for SAP |
| 2178632 | Key Monitoring Metrics for SAP on Microsoft Azure |
| 1409604 | Virtualization on Windows: Enhanced Monitoring |

General default limitations and maximum limitations of Azure subscriptions can be found here:

<http://azure.microsoft.com/en-us/documentation/articles/azure-subscription-service-limits/#subscription>

# Possible Scenarios

SAP is often seen as one of the most mission-critical applications within enterprises. The architecture and operations of these applications is mostly very complex and ensuring that you meet requirements on availability and performance is important.

Thus enterprises have to think carefully about which applications can be run in a Public Cloud environment, independent of the chosen Cloud provider.

Possible system types for deploying SAP NetWeaver based applications within public cloud environments are listed below:

1. **Medium sized production systems**
2. **Development systems**
3. **Testing systems**
4. **Prototype systems**
5. **Learning / Demonstration systems**

In order to successfully deploy SAP systems into either Azure IaaS or IaaS in general, it is important to understand the significant differences between the offerings of traditional outsourcers or hosters and IaaS offerings. Whereas the traditional hoster or outsourcer will adapt infrastructure (network, storage and server type) to the workload a customer wants to host, it is instead the customer’s responsibility to choose the right workload for IaaS deployments.

As a first step, customers need to verify the following items:

* The SAP supported VM types of Azure and
* The SAP supported products/releases on Azure and
* The supported OS and DBMS releases for the specific SAP releases in Azure and
* SAPS throughput provided by different Azure SKUs.

The answers to these questions can be read in SAP Note [1928533 – SAP Applications on Azure: Supported Products and Sizing](http://service.sap.com/sap/support/notes/1928533).

As a second step, Azure resource and bandwidth limitations need to be compared to actual resource consumption of on-premises systems. Therefore, customers need to be familiar with the different capabilities of the Azure types supported with SAP in the area of:

* CPU and memory resources of different VM types and
* IOPS bandwidth of different VM types and
* Network capabilities of different VM types.

Most of that data can be found here: <http://msdn.microsoft.com/en-us/library/windowsazure/dn197896.aspx>

Keep in mind that the limits listed in the link above are upper limits. It does not mean that the limits for any of the resources, e.g. IOPS can be provided under all circumstances. The exceptions though are the CPU and memory resources of a chosen VM type. For the VM types supported by SAP, the CPU and memory resources are reserved and as such available at any point in time for consumption within the VM.

The Microsoft Azure platform like other IaaS platforms is a multi-tenant platform. This means that Storage, Network and other resources are shared between tenants. Intelligent throttling and quota logic is used to prevent one tenant from impacting the performance of another tenant (noisy neighbor) in a drastic way. Though logic in Azure tries to keep variances in bandwidth experienced small, highly shared platforms tend to introduce larger variances in resource/bandwidth availability than a lot of customers are used to in their on-premises deployments. As a result, you might experience different levels of bandwidth in regards to networking or storage I/O (the volume as well as latency) from minute to minute. The probability that an SAP system on Azure could experience larger variances than in an on-premises system needs to be taken into account.

A last step is to evaluate availability requirements. It can happen, that the underlying Azure infrastructure needs to get updated and requires the hosts running VMs to be rebooted. In these cases, VMs running on those hosts would be shut down and restarted as well. The timing of such maintenance is done during non-core business hours for a particular region but the potential window of a few hours during which a restart will occur is relatively wide. There are various technologies within the Azure platform that can be configured to mitigate some or all of the impact of such updates. Future enhancements of the Azure platform, DBMS and SAP application are designed to minimize the impact of such restarts.

In order to successfully deploy a SAP system onto Azure, the on-premises SAP system(s) Operating System, Database and SAP applications must appear on the SAP Azure support matrix, fit within the resources the Azure infrastructure can provide and which can work with the Availability SLAs Microsoft Azure offers. As those systems are identified, you need to decide on one of the following two deployment scenarios.

## Cloud-Only - Virtual Machine deployments into Azure without dependencies on the on-premises customer network

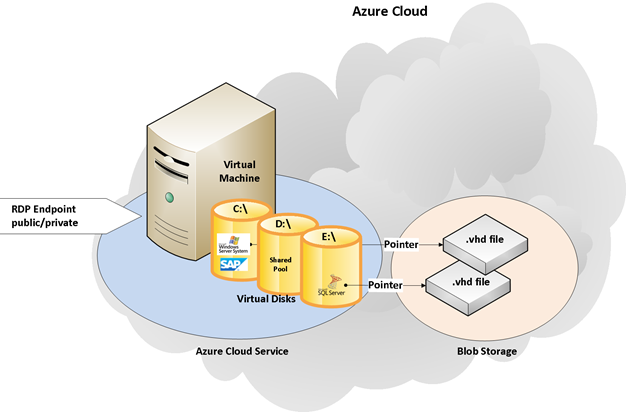


Figure 1: Single VM with SAP demo or training scenario deployed in Azure

This scenario is typical for trainings or demo systems, where all the components of SAP and non-SAP software are installed within a single VM. Production SAP systems are not supported in this deployment scenario. In general, this scenario meets the following requirements:

* The VMs themselves are accessible over the public network. Direct network connectivity for the applications running within the VMs to the on-premises network of either the company owning the demos or trainings content or the customer is not necessary.
* In case of multiple VMs representing the trainings or demo scenario, network communications and name resolution needs to work between the VMs. But communications between the set of VMs need to be isolated so that several sets of VMs can be deployed side by side without interference.
* Internet connectivity is required for the end user to utilize terminal services in order to connect to the VMs hosted in Azure. Terminal Services/RDS is used to access the VM to either fulfill the training tasks or perform the demos. If SAP ports such as 3200, 3300 & 3600 can also be exposed the SAP application instance can be accessed from any Internet connected desktop.
* The SAP system(s) (and VM(s)) represent a standalone scenario in Azure which only requires public internet connectivity for end user access and does not require a connection to other VMs in Azure.
* SAPGUI and Internet Explorer are installed and run directly on the VM.
* A fast reset of a VM to the original state and new deployment of that original state again is required.
* In the case of demo and trainings scenarios which are realized in multiple VMs, an Active Directory and/or DNS service is required for each set of VMs.

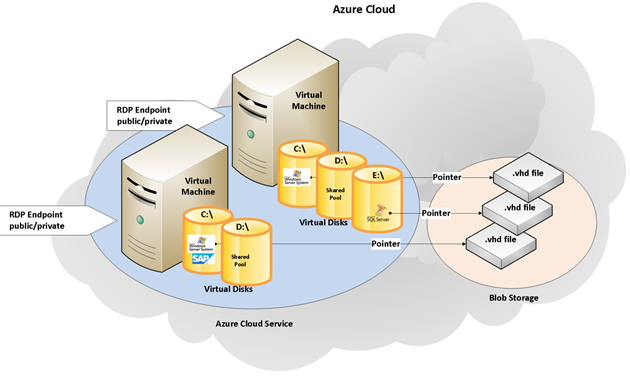


Figure 2: Group of VM's representing one demo or training scenario in an Azure Cloud Service

It is important to keep in mind that the VM(s) in each of the sets need to be deployed in parallel, where the VM names in each of the set are the same.

## Cross-Premise - Deployment of single or multiple SAP VMs into Azure with the requirement of being fully integrated into the on-premises network

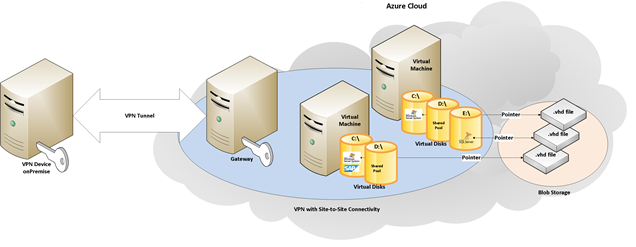


Figure 3: VPN with Site-To-Site Connectivity (Cross-Premise)

This scenario is a Cross-Premises scenario with many possible deployment patterns. It can be described as simply as running some parts of the SAP landscape on-premises and other parts of the SAP landscape on Azure. All aspects of the fact that part of the SAP components are running on Azure should be transparent for end users. Hence the SAP Transport Correction System (STMS), RFC Communication, Printing, Security (like SSO), etc. will work seamlessly for the SAP systems running on Azure. But the Cross-Premises scenario also describes a scenario where the complete SAP landscape runs in Azure with the customer’s domain and DNS extended into Azure.

**Note:** This is the deployment scenario which is supported for running productive SAP systems.

**IMPORTANT**: When we are talking about Cross-Premises scenarios between Azure and on-premises customer deployments, we are looking at the granularity of whole SAP systems. Scenarios which are not supported for Cross-Premises scenarios are:

* Running different layers of SAP applications in different deployment methods. E.g. running the DBMS layer on-premises, but the SAP application layer in VMs deployed as Azure VMs or vice versa.
* Some components of an SAP layer in Azure and some on-premises. E.g. splitting Instances of the SAP application layer between on-premises and Azure VMs.
* Distribution of VMs running SAP instances of one system over multiple Azure Regions is not supported.

The reason for these restrictions is the requirement for a very low latency high performance network within one SAP system, especially between the application instances and the DBMS layer of an SAP system.

## Supported OS and Database Releases

* Microsoft server software supported for Azure Virtual Machine Services is listed in this article: <http://support.microsoft.com/kb/2721672>.
* Supported operating system releases, database system releases supported on Azure Virtual Machine Services in conjunction with SAP software are documented in SAP Note [1928533 – SAP Applications on Azure: Supported products and Sizing](http://service.sap.com/sap/support/notes/1928533).
* SAP applications and releases supported on Azure Virtual Machine Services are documented in SAP Note [1928533 – SAP Applications on Azure: Supported products and Sizing](http://service.sap.com/sap/support/notes/1928533).
* Only 64Bit images are supported to run as Guest VMs in Azure for SAP scenarios. This also means that only 64-bit SAP applications and databases are supported.

# Microsoft Azure Virtual Machine Services

The Microsoft Azure platform is an internet-scale cloud services platform hosted and operated in Microsoft data centers. The platform includes the Microsoft Azure Virtual Machine Services (Infrastructure as a Service, or IaaS) and a set of rich Platform as a Service (PaaS) capabilities.

The Azure platform reduces the need for up-front technology and infrastructure purchases. It simplifies maintaining and operating applications by providing on-demand compute and storage to host, scale and manage web application and connected applications. Infrastructure management is automated with a platform that is designed for high availability and dynamic scaling to match usage needs with the option of a pay-as-you-go pricing model.

Azure handles load balancing and resource management and automatically manages the life cycle of a service based on requirements that the owner of the service established. Services are called “Hosted Services” within the Microsoft Azure Platform Management Portal.

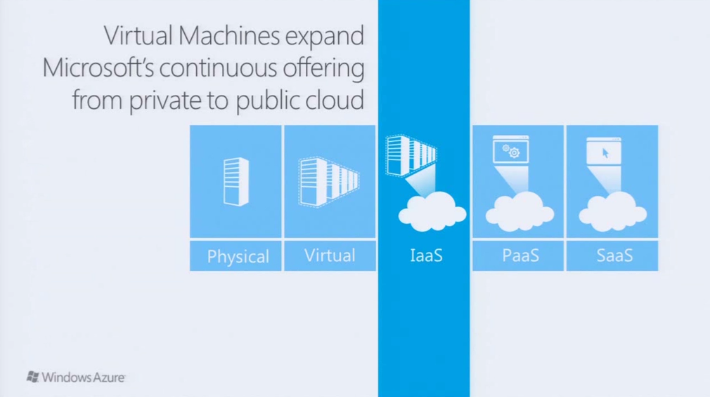


Figure 4: Positioning of Microsoft Azure Virtual Machine Services

With Azure Virtual Machine Services, Microsoft is enabling you to deploy custom server images to Azure as IaaS instances (see Figure 4). The Virtual Machines in Azure are based on Hyper-V virtual hard drives (VHD) and are able to run different operating systems as Guest OS.

From an operational perspective, the Azure Virtual Machine Service offers similar experiences as virtual machines deployed on premises. However, it has the significant advantage that you don’t need to procure, administer and manage the infrastructure. Developers and Administrators have full control of the operating system image within these virtual machines. Administrators can logon remotely into those virtual machines to perform maintenance and troubleshooting tasks as well as software deployment tasks. In regard to deployment, the only restrictions are the sizes and capabilities of Azure VMs. These may not be as fine granular in configuration as this could be done on premise. There is a choice of VM types that represent a combination of:

* Number of vCPUs,
* Memory,
* Number of VHDs that can be attached,
* Network and Storage bandwidths.

The size and limitations of various different virtual machines sizes offered can be seen in a table in this article: <http://msdn.microsoft.com/en-us/library/windowsazure/dn197896.aspx>

As you will realize there are different families or series of virtual machines. As of July 2015, you can distinguish the following families of VMs:

* A0-A7 VM types: Not all of those are certified for SAP. First VM series that Azure IaaS got introduced with.
* A8-A11 VM types: High Performance computing instances. Running on different better performing compute hosts than other A-series VMs.
* D-Series VM types: Better performing than A0-A7. Not all of the VM types are certified with SAP.
* DS-Series VM types: use same hosts as D-series, but are able to connect to Azure Premium Storage (see [chapter 3.3.2](#_Azure_Premium_Storage) of this document). Again not all VM types are certified with SAP.
* G-series VM types: High memory VM types.

You may find the same CPU and memory configurations in different VM series. Nevertheless, when you look up the throughput performance of these VMs out of the different series they might differ significantly. Despite having the same CPU and memory configuration. Reason is that the underlying host server hardware at the introduction of the different VM types had different throughput characteristics. Usually the difference shown in throughput performance also is reflected in the price of the different VMs.

Please note that not all different VM series might be offered in each one of the Azure Regions (for Azure Regions see next chapter). Also be aware that not all VMs or VM-Series are certified for SAP.

**IMPORTANT: For the use of SAP NetWeaver based applications, only the subset of VM types and configurations listed in SAP Note** [1928533 – SAP Applications on Azure: Supported Products and Sizing](http://service.sap.com/sap/support/notes/1928533) **are supported.**

## Azure Regions

Microsoft allows to deploy Virtual Machines into so called ‘Azure Regions’. An Azure Region may be one or multiple datacenters that are located in close proximity. For most of the geopolitical regions in the world Microsoft has at least two Azure Regions. E.g. in Europe there is an Azure Region of ‘North Europe’ and one of ‘West Europe’. Such two Azure Regions within a geopolitical region are separated by significant enough distance so that natural or technical disasters do not affect both Azure Regions in the same geopolitical region. Since Microsoft is steadily building out new Azure Regions in different geopolitical regions globally, the number of these regions is steadily growing and as of July 2015 reached the number of 19 Azure Regions with additional Regions announced already. You as a customer can deploy SAP systems into all these regions, including the two Azure Regions in China.

## The Microsoft Azure Virtual Machine Concept

Microsoft Azure offers an Infrastructure as a Service (IaaS) solution to host Virtual Machines with similar functionalities as an on-premises virtualization solution. You are able to create Virtual Machines from within the Azure Portal or PowerShell, which both also offer deployment and management capabilities.

Another interesting feature is the ability to create images from Virtual Machines, which allows you to prepare certain repositories from which you are able to quickly deploy Virtual Machine instances which meet your requirements.

More documentation on Azure VMs can be found here:

<http://azure.microsoft.com/en-us/documentation/articles/fundamentals-application-models/#VMachine>

<http://www.windowsazure.com/en-us/manage/windows/>

### Fault Domains

Fault Domains represent a physical unit of failure, very closely related to the physical infrastructure contained in data centers, and while a physical blade or rack can be considered a Fault Domain, there is no direct one-to-one mapping between the two.

When you deploy multiple Virtual Machines as part of one SAP system in Microsoft Azure Virtual Machine Services, you can influence the Azure Fabric Controller to deploy your application into different Fault Domains, thereby meeting the requirements of the Microsoft Azure SLA. However, the distribution of Fault Domains over an Azure Scale Unit (collection of hundreds of Compute nodes or Storage nodes and networking) or the assignment of VMs to a specific Fault Domain is something over which you do not have direct control. In order to direct the Azure fabric controller to deploy a set of VMs over different Fault Domains, you need to assign an Azure Availability Set to the VMs at deployment time. For more information on Azure Availability Sets, see [chapter 3.2.3](#_Azure_Availability_Sets) in this document.

### Upgrade Domains

Upgrade Domains represent a logical unit that help to determine how a VM within an SAP system, that consists of SAP instances running in multiple VMs, will be updated. When an upgrade occurs, Microsoft Azure goes through the process of updating these Upgrade Domains one by one. By spreading VMs at deployment time over different Upgrade Domains you can protect your SAP system partly from potential downtime. In order to force Azure to deploy the VMs of an SAP system spread over different Upgrade Domains, you need to set a specific attribute at deployment time of each VM. Similar to Fault Domains, an Azure Scale Unit is divided into multiple Upgrade Domains. In order to direct the Azure fabric controller to deploy a set of VMs over different Upgrade Domains, you need to assign an Azure Availability Set to the VMs at deployment time. For more information on Azure Availability Sets, see the next section[\_Azure\_Availability\_Sets](#_Azure_Availability_Sets) in this document.

### Azure Availability Sets

Azure Virtual Machines within one Azure Availability Set will be distributed by the Azure Fabric Controller over different Fault and Upgrade Domains. The purpose of the distribution over different Fault and Upgrade Domains is to prevent all VMs of an SAP system from being shut down in the case of infrastructure maintenance or a failure within one Fault Domain. By default, VMs are not part of an Availability Set. The participation of a VM in an Availability Set is defined at deployment time or later on by a reconfiguration and re-deployment of a VM.

To understand the concept of Azure Availability Sets and the way Availability Sets relate to Fault and Upgrade Domains, please read this documentation:

* <http://azure.microsoft.com/en-us/documentation/articles/manage-availability-virtual-machines/>
* <http://michaelwasham.com/windows-azure-powershell-reference-guide/understanding_configuring_availability_sets_powershell/>

Be aware that it may not be possible to include all VM types out of different VM series within one Availability Set. Known restrictions are:

* An Availability Set that should include VM types A8, A9, A10 or A11 only can include these four VM types.
* An Availability Set that includes A0-A7 VMs or D-series VMs can’t include G-Series VMs at this point in time.

### Azure Affinity Groups – Changed recommendations!!!

The concept of Azure Affinity Groups is inherited out of earlier days of Azure topology and does not apply today anymore. There is no need to leverage the construct of Azure Affinity groups anymore! Please be aware that this is a change compared to earlier releases of the paper!

## Storage: Microsoft Azure Storage and Data Disks

Microsoft Azure Virtual Machines utilize different storage types. When implementing SAP on Azure Virtual Machine Services it is important to understand the differences between these two main types of storage:

* Non-Persistent, volatile storage.
* Persistent storage.

The non-persistent storage is directly attached to the running Virtual Machines and resides on the compute nodes themselves – the local instance storage (temporary storage). The size depends on the size of the Virtual Machine chosen when the deployment started. This storage type is volatile and therefore the disk is initialized when a Virtual Machine instance is restarted. Typically, the pagefile for the operating system is located on this temporary disk. The drive of the temporary storage is named D:\ in a deployed VM. The actual drive is volatile because it is getting stored on the host server itself. If the VM moved in a redeployment (e.g. due to maintenance on the host or shutdown and restart) the content of the drive is lost. Therefore, it is not an option to store any important data on this drive. The type of media used for this type of storage differs between different VM series with very different performance characteristics which as of June 2015 look like:

* A0-A7: Very limited performance. Not recommended for anything beyond Windows page file
* A8-A11: Very good performance characteristics with some ten thousand IOPS and >1GB/sec throughput.
* D-Series: Very good performance characteristics with some then thousand IOPS and >1GB/sec throughput.
* DS-Series: Very good performance characteristics with some ten thousand IOPS and >1GB/sec throughput.
* G-Series: Very good performance characteristics with some ten thousand IOPS and >1GB/sec throughput.

Statements above are applying to the VM types that are certified with SAP. The VM-series with excellent IOPS and throughput qualify for leverage by some DBMS features. Please see the document ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’ for more details.

Microsoft Azure Storage provides persisted storage and the typical levels of protection and redundancy seen on SAN storage. Disks based on Azure Storage are virtual hard disk (VHDs) located in the Azure Storage Services. The local OS-Disk (C:\) is stored on the Azure Storage, and additional Volumes/Disks mounted to the VM get stored there, too.

It is possible to upload an existing VHD from on-premises or create empty ones from within Azure and attach those to deployed VMs. Those VHDs are referenced as *Azure* *Disks*.

After creating or uploading a VHD into Azure Storage, it is possible to mount and attach those to an existing Virtual Machine and to copy existing (unmounted) VHD.

As those VHDs are persisted, data and changes within those are safe when rebooting and recreating a Virtual Machine instance. Even if an instance or a whole hosted service is deleted, these VHDs stay safe and can be redeployed or in case of non-OS disks can be mounted to other VMs.

Within the network of Azure Storage different redundancy levels can be configured:

* Minimum level that can be selected is ‘local redundancy’, which is equivalent to three-replica of the data within the same datacenter of an Azure Region (see [chapter 3.1](#_Azure_Regions)).
* Zone redundant storage which will spread the three images over different datacenters within the same Azure Region.
* Default redundancy level is geographic redundancy which asynchronously replicates the content into another 3 images of the data into another Azure Region which is hosted in the same geopolitical region.

Also see the table on top of this article in regards to the different redundancy options: <http://azure.microsoft.com/en-us/pricing/details/storage/>

More information in regards to Azure Storage can be found here:

<http://www.windowsazure.com/en-us/manage/services/storage/>

<http://www.windowsazure.com/en-us/solutions/storage-backup-recovery/>

<http://msdn.microsoft.com/en-us/library/windowsazure/ee691964.aspx>

### Azure Standard Storage

Azure Standard BLOB storage was the type of storage available when Azure IaaS was released. There were IOPS quotas enforced per single VHD. Latency experienced was not in the same class as SAN/NAS devices typically deployed for high-end SAP systems hosted on-premises. Nevertheless, the Azure Standard Storage proved sufficient for many hundreds SAP systems meanwhile deployed in Azure.

Azure Standard Storage is charged based on the actual data that is stored, the volume of storage transactions, outbound data transfers and redundancy option chosen. Many VHDs can be created at the maximum 1TB in size, but as long as those remain empty there is no charge. If you then fill one VHD with 100GB each, you will be charged for storing 100GB and not for is the nominal size the VHD got created with.

### Azure Premium Storage

In April 2015 Microsoft introduced Azure Premium Storage. Premium Storage got introduced with the goal to provide:

* Better I/O latency.
* Better throughput.
* Less variability in I/O latencies.

For that purpose, a lot of changes were introduced of which the two most significant are:

* Usage of SSD disks in the Azure Storage nodes
* A new read cache that is backed by the local SSD of an Azure compute node

In opposite to Standard storage where capabilities did not change dependent on the size of the disk (or VHD), Premium Storage currently has 3 different disk categories which are shown in the lower part of this article: <http://azure.microsoft.com/en-us/pricing/details/storage/>

You see that IOPS/VHD and disk throughput/VHD are dependent on the size category of the disks

Cost basis in the case of Premium Storage is not the actual data volume stored in such VHDs, but the size category of such a VHD, independent of the amount of the data that is stored within the VHD.

You also can create VHDs on Premium Storage that are not directly mapping into the size categories shown. This may be the case, especially when copying VHDs from Standard Storage into Premium Storage. In such cases a mapping to the next largest Premium Storage disk option is performed.

Please be aware that only certain VM series can benefit from the Azure Premium Storage. As of July 2015, this is only the DS-series. The DS-series is basically the same as D-series with the exception that DS-series has the ability to mount Premium Storage based VMs additionally to VHDs that are hosted on Azure Standard Storage.

If you are checking out the part of the DS-series VMs in this article: <https://msdn.microsoft.com/en-us/library/azure/dn197896.aspx> you also will realize that there a data volume limitations to Premium Storage VHDs on the granularity of the VM level. Different DS-series VMs also have different limitations in regards to the number of VHDs that can be mounted. These limits are documented in the article mentioned above as well. But in essence it means that if you mount 32 x P30 disks/VHDs to a single DS14 VM that you cannot get 32 x the maximum throughput of a P30 disk. Instead the maximum throughput on VM level as documented in the article will limit data throughput.

More information on Premium Storage can be found here: <http://azure.microsoft.com/blog/2015/04/16/azure-premium-storage-now-generally-available-2>

### Azure Storage Accounts

When deploying services or VMs in Azure, deployment of VHDs and VM Images must be organized in units called Azure Storage Accounts. When planning an Azure deployment, you need to carefully consider the restrictions of Azure. On the one side, there is a limited number of Storage Accounts per Azure subscription. Although each Azure Storage Account can hold a large number of VHD files, there is a fixed limit on the total IOPS per Storage Account. When deploying hundreds of SAP VMs with DBMS systems creating significant IO calls, it is recommended to distribute high IOPS DBMS VMs between multiple Azure Storage Accounts. Care must be taken not to exceed the current limit of Azure Storage Accounts per subscription. Because storage is a vital part of the database deployment for an SAP system, this concept is discussed in more detail in the already referenced document ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’.

More information about Azure Storage Accounts can be found here: <http://azure.microsoft.com/en-us/documentation/articles/storage-scalability-targets> . Reading this article, you will realize that there are differences in the limitations between Azure Standard Storage Accounts and Premium Storage Accounts. Major differences are the volume of data that can be stored within such a Storage Account. In Standard Storage the volume is a magnitude larger than with Premium Storage. On the other side the Standard Storage Account is severely limited in IOPS (see column ‘Total Request Rate’), whereas the Azure Premium Storage Account has no such limitation. We will discuss details and results of these differences when discussing the deployments of SAP systems, especially the DBMS servers.

Within a Storage Account, you have the possibilities to create different containers for the purpose of organizing and categorizing different VHDs. These containers are usually used to e.g. separate VHDs of different VMs. There are no performance implications in using just one container or multiple containers underneath a single Azure Storage Account.

Within Azure a VHD name follows the following naming connection that needs to provide a unique name for the VHD within Azure:

http(s)://<storage account name>.blob.core.windows.net/<container name>/<vhd name>

As mentioned the string above needs to uniquely identify the VHD that is stored on Azure Storage.

## Microsoft Azure Networking

Microsoft Azure will provide a network infrastructure which allows the mapping of all scenarios which we want to realize with SAP software. The capabilities are:

* Access from the outside, directly to the VMs via Windows Terminal Services; or
* Access to services and specific ports used by applications within the VMs; or
* Internal Communication and Name Resolution between a group of VMs deployed as Azure VMs; or
* Cross-Premises Connectivity between a customer’s on-premises network and the Azure network
* Cross Azure Region or Datacenter Connectivity between Azure sites

More information can be found here:

<http://www.windowsazure.com/en-us/manage/services/networking/>

There are a lot of different possibilities to configure name and IP resolution in Azure. In this document, Cloud-Only scenarios rely on the default of using Azure DNS (in contrast to defining an own DNS service). This enforces some restrictions as explained here:<http://msdn.microsoft.com/en-us/library/windowsazure/jj156088.aspx#bkmk_IDNSfeatures>

For Cross-Premises scenarios we are relying on the fact that the on-premises AD/DNS has been extended via VPN or private connection to Azure. For certain scenarios as documented here, it might be necessary to have an AD replica installed in Azure.

Because networking and name resolution is a vital part of the database deployment for an SAP system, this concept is discussed in more detail in the already referenced document ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’.

### Cloud Services and Virtual Networks

Following are some explanatory notes to avoid confusions regarding the different types of network available in Azure.

#### Azure Cloud Services

The Azure Cloud Services were designed to provide a quick and powerful way to deploy and manage applications and services. Azure handles the deployment details from provisioning and load balancing to health monitoring for continuous availability.

The Azure Cloud Service is a wrapper for the deployment of the Virtual Machine and essentially the same construct as you get for role deployment in the PaaS model. It can be seen as an automatically generated private network with a DHCP, default Gateway and Azure DNS. The Cloud Service provides a Gateway which exposes the virtual machines ports to the internet to make the virtual machines accessible. The built in Azure name resolution does not work across different Cloud Services and NetBIOS is not supported. The given IP addresses (public and private) could potentially change after the VM has been stopped and restarted. You do not have control over the TCP/IP configuration of that network, neither of the private IP-address range, nor of the gateway public IP-address.

There are two ways to create an Azure Cloud Service:

* The most intuitive way to create a Cloud Service is to deploy a new virtual machine with the selected value “Create a new Cloud Service” from the drop down list of the field “Cloud Service” in the “Create a Virtual Machine” dialog of the Azure Management Portal as shown in Figure 5: Create a new cloud service. The Azure Cloud Service will be generated hidden in the background while deploying the virtual machine.
* Or you may create a Cloud Service explicitly on the Cloud Services tab of the Azure Management Platform and add virtual machines later.

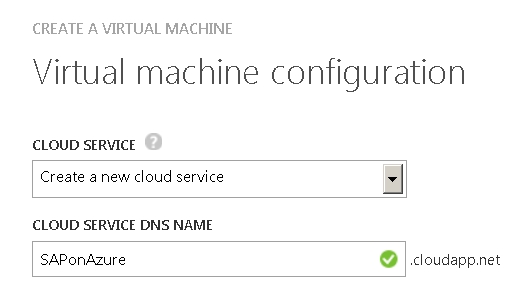


Figure 5: Create a new cloud service.

More information can be found here: <http://www.windowsazure.com/en-us/documentation/services/cloud-services/>

In order to expose the http(s) endpoints of SAP Java instances/systems that are hosted in Azure Virtual Machine Services to the internet, Azure provides a subdomain on the cloudapp.net domain so your users can access your application through URL. However, you can also expose your Azure hosted service on your own domain name. The process of configuring your own domain name for an Azure Cloud Service is documented here: <http://www.windowsazure.com/en-us/develop/net/common-tasks/custom-dns/>.

#### Azure Virtual Networks

An Azure Virtual Network is less restricted and offers several advantages over Cloud Services. The confusing point may be that it does not replace the Cloud Services but coexists with them. The Virtual Machines are still wrapped in Cloud Services and you may have as many Cloud Services in the Azure Virtual Network as virtual machines deployed. The assignment of IP addresses after a restart of VMs or for VMs that were shut down for a longer time is not fixed by default in Azure Virtual Networks. In this regards the behavior is similar to the handling of IP addresses with Azure Cloud Services.

By building up an Azure Virtual Network you can define the address range of the private IP addresses allocated by the DHCP functionality of the Azure DNS server. The Azure Virtual Network settings will override the Cloud Services, e.g. internal private IP addresses and the DNS server address which are preset by the network configuration. In Cross-Premises scenarios, the IP address range defined will still be allocated using DHCP by Azure. However, Domain Name resolution will be done on-premises (assuming that the VMs are a part of an on-premises domain) and hence can resolve addresses beyond different Azure Cloud Services.

Originally an Azure Virtual Network was bound to an Affinity Group. With that a Virtual Network in Azure got restricted to the Azure Scale Unit that the Affinity Group got assigned to. In the end, this meant the Virtual Network was restricted to the resources available in the Azure Scale Unit. This has since changed and now Azure Virtual Networks can stretch across more than one Azure Scale Unit. However, that requires that Azure Virtual Networks are **NOT** associated with Affinity Groups anymore at creation time. We already mentioned earlier that in opposite to recommendations a year ago, you should **NOT leverage Azure Affinity Groups anymore**. For details, please see: <http://azure.microsoft.com/blog/2014/05/14/regional-virtual-networks/>

You need to set up an Azure Virtual Network if

* You do not want to expose the virtual machines endpoints directly to the internet; or
* You need on premises connectivity (Cross-Premises); or
* You need to leverage Static IP assignment for VMs; or
* You want to expand on premises AD and DNS into Azure

More details and a decision tree can be found here: <http://msdn.microsoft.com/library/azure/jj156007.aspx#BKMK_MoreInformation>

There is more detail here: <http://www.windowsazure.com/en-us/documentation/services/virtual-network/>

Achtung Be Aware:

By default, once a VM is deployed you cannot change the Virtual Network configuration or Cloud Service assignment. The TCP/IP settings must be left to the Azure DHCP server. By default, the following rules apply in regards with TCP/IP address assignment:

* Default behavior is Dynamic IP assignment.
* IP addresses will remain assigned even through shutdown/reboots within a Cloud Service as long as at least one of the VMs within the Cloud Service is up and running.
* As soon as all VMs of a Cloud Service are shutdown, IP addresses may be different after the restart of the VMs within the Cloud Service unless the Cloud service is deployed within an Azure Virtual Network AND the VMs were assigned fixed IP addresses.

The MAC address of the virtual network card may change and Windows Server will pick up the new network card and will automatically use DHCP to assign the IP and DNS addresses in this case.

#### Static IP Assignment

It is possible to assign IP addresses to VMs within an Azure Virtual Network. Running the VMs in an Azure Virtual Network opens a great possibility to leverage this functionality if needed or required for some scenarios. The IP assignment remains valid throughout the existence of the VM, independent of whether the VM is running or shutdown. As a result, you need to take the overall number of VMs (running and stopped VMS) into account when defining the range of IP addresses for the Virtual Network. The IP address remains assigned either until the VM is deleted or until the IP address gets de-assigned again. Please see detailed information here: <http://msdn.microsoft.com/en-us/library/windowsazure/dn630228.aspx>

#### Multiple NICs per VM

For VMs that are created within Azure Virtual Networks, you can define multiple virtual network interface cards (vNIC). With the ability to have multiple vNICs you can start to set up network traffic separation where e.g. client traffic is routed through one vNIC and backend traffic is routed through a second vNIC. Dependent on the type of VM there are different limitations in regards to the number of vNICs. Exact details, functionality and restrictions can be read here: <https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-multiple-nics/>.

### Site-to-Site Connectivity

Cross-Premises is Azure VMs and On-Premises linked with a transparent and permanent VPN connection. It is expected to become the most common SAP deployment pattern in Azure. The assumption is that operational procedures and processes with SAP instances in Azure should work transparently. This means you should be able to print out of these systems as well as use the SAP Transport Management System (TMS) to transport changes from a development system in Azure to a test system which is deployed on premise. More documentation around site-to-site can be found here: <http://www.windowsazure.com/en-us/manage/services/networking/cross-premises-connectivity/>

#### VPN Tunnel Device

In order to create a site-to-site connection (on-premises Datacenter to Azure Datacenter), you will need to either obtain and configure a VPN device, or use Routing and Remote Access Service (RRAS) which was introduced as a software component with Windows Server 2012.

<http://msdn.microsoft.com/en-us/library/windowsazure/jj156075.aspx>

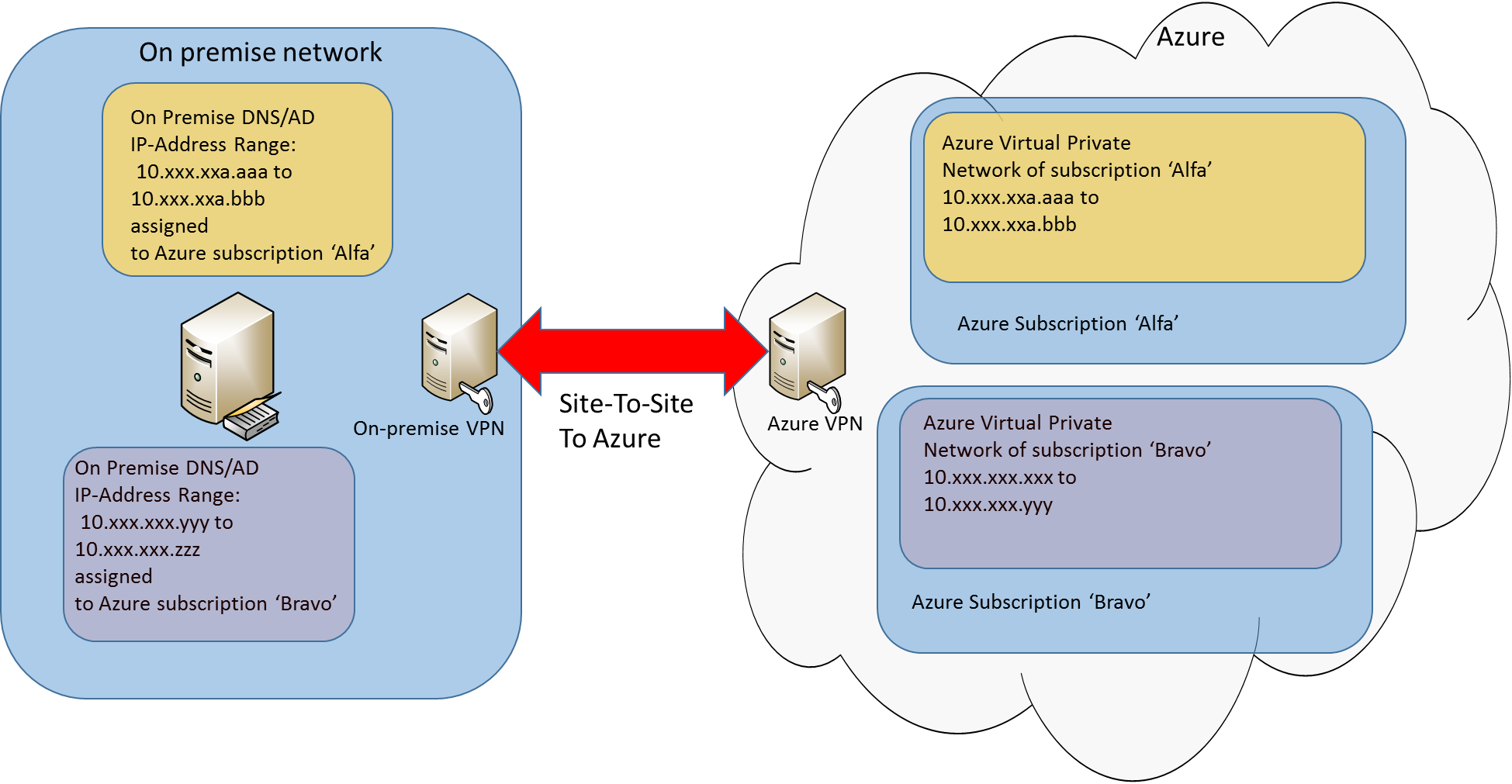


Figure Site-to-site connection between on-premises and Azure

Figure 6 shows two Azure subscriptions have IP address subranges reserved for usage in Azure Virtual Networks in Azure. The connectivity from the on-premises network to Azure is established via VPN.

### Point-to-Site VPN

Point-to-site VPN requires every client machine to connect with its own VPN into Azure. For the SAP scenarios we are looking at, point-to-site connectivity is not practical. Therefore, no further references will be given to point-to-site VPN connectivity. More information can be found here: <http://msdn.microsoft.com/en-us/library/azure/dn133798.aspx> .

### Multi-Site VPN

Azure also nowadays offers the possibility to create Multi-Site VPN connectivity for one Azure subscription. Previously a single subscription was limited to one site-to-site VPN connection. This limitation went away with Multi-Site VPN connections for a single subscription. This makes it possible to leverage more than one Azure Region for a specific subscription through Cross-Premises configurations.

For more documentation please see: <http://msdn.microsoft.com/en-us/library/azure/dn690124.aspx>

### VNet to VNet Connection

Using Multi-Site VPN, you need to configure a separate Azure Virtual Network in each of regions. However very often you have the requirement that the software components in the different regions should communicate with each other. Ideally this communication should not be routed from one Azure Region to on-premises and from there to the other Azure Region. To shortcut, Azure offers the possibility to configure a connection from one Azure Virtual Network in one region to another Azure Virtual Network hosted in another region. This functionality is called VNet-to-VNet connection. More details on this functionality can be found here: <http://msdn.microsoft.com/en-us/library/azure/dn690122.aspx>.

### Private Connection to Azure – ExpressRoute

Microsoft Azure ExpressRoute allows the creation of private connections between Azure datacenters and either the customer's on-premises infrastructure or in a co-location environment. ExpressRoute is offered by various MPLS (packet switched) VPN providers or other Network Service Providers. ExpressRoute connections do not go over the public Internet. ExpressRoute connections offer higher security, more reliability through multiple parallel circuits, faster speeds and lower latencies than typical connections over the Internet.

Find more details on Azure ExpressRoute and offerings here:

<http://msdn.microsoft.com/library/azure/dn606309.aspx>

<http://msdn.microsoft.com/en-us/library/azure/dn606292.aspx>

Express Route enables multiple Azure subscriptions through one ExpressRoute circuit as documented here: <http://msdn.microsoft.com/en-us/library/azure/dn835110.aspx>

### Forced tunneling in case of Cross-Premise

For VMs joining on-premises domains through site-to-site, point-of-site or ExpressRoute, you need to make sure that the Internet proxy settings are getting deployed for all the users in those VMs as well. By default, software running in those VMs or users using a browser to access the internet would not go through the company proxy, but would connect straight through Azure to the internet. But even the proxy setting is not a 100% solution to direct the traffic through the company proxy since it is responsibility of software and services to check for the proxy. If software running in the VM is not doing that or an administrator manipulates the settings, traffic to the Internet can be detoured again directly through Azure to the Internet.

In order to avoid this, you can configure Forced Tunneling with site-to-site connectivity between on-premises and Azure. The detailed description of the Forced Tunneling feature is published here: <http://msdn.microsoft.com/en-us/library/azure/dn835140.aspx>

Forced Tunneling with ExpressRoute is enabled by customers advertising a default route via the ExpressRoute BGP peering sessions.

### Summary of Azure Networking

This chapter contained many important points about Azure Networking. Here is a summary of the main points:

* The Azure Cloud Service is a wrapper for the deployment of virtual machines, applications and services. It provides a shared Gateway which maps private service ports to public Endpoints.
* DNS services are provided within an Azure Cloud Service.
* A Cloud Service can host multiple different VM’s, but a VM cannot be deployed in more than one Cloud Service.
* You can deploy multiple Azure Cloud Services within one Azure subscription.
* Azure Virtual Networks allows to set up the network according to your own needs.
* Creating an Azure Virtual Network is optional for Cloud-Only deployments and mandatory for Cross-Premises deployments.
* Azure Virtual Networks can be leveraged to assign IP address ranges to VMs or assign fixed IP addresses to VMs.
* Deployment of VMs within Azure Virtual Networks still requires the use of Cloud Services.
* You can deploy multiple Cloud Services within one Azure Virtual Network.
* Software components in VMs that run in different Cloud Services within one Azure Virtual Network cannot communicate with each other by using the VM name, but only by using the IP address.
* To set up a Site-To-Site or Point-To-Site connection you need to create an Azure Virtual Network first.
* Even with site-to-site, point-to-site or ExpressRoute connectivity, you still deploy VMs into Azure Cloud Services which again are assigned to Azure Virtual Networks.
* Once a virtual machine has been deployed it is no longer possible to change Cloud Service and Virtual Network settings assigned to the VM, except DNS.

## Quotas in Azure Virtual Machine Services

We need to be clear about the fact that the storage and network infrastructure is shared between VMs running a variety of services in the Azure infrastructure. And just as in the customer’s own datacenters, overprovisioning of some of the infrastructure resources does take place to a degree. The Microsoft Azure Platform uses disk, CPU, network and other quotas to limit the resource consumption and to preserve consistent and deterministic performance. The different VM types (A5, A6, etc) have different quotas for the number of disks, CPU, RAM and Network.

**Note:** CPU and memory resources of the VM types supported by SAP are pre-allocated on the host nodes. This means that once the VM is deployed, the resources on the host will be available as defined by the VM type.

When planning and sizing SAP on Azure solutions the quotas for each virtual machine size must be considered. The VM quotas are described here:

<http://msdn.microsoft.com/en-us/library/windowsazure/dn197896.aspx>

The quotas described represent the theoretical maximum values. The limit of IOPS per VHD may be achieved with small IOs (8kb) but possibly may not be achieved with large IOs (1Mb). The IOPS limit is enforced on the granularity of single VHDs.

As a rough decision tree to decide whether an SAP system fits into Azure Virtual Machine Services and its capabilities or whether an existing system needs to be configured differently in order to deploy the system on Azure, the decision tree below can be used:



Figure Decision tree to decide ability to deploy SAP on Azure

**Step 1:** The most important information to start with is the SAPS requirement for a given SAP system. The SAPS requirements need to be separated out into the DBMS part and the SAP application part, even if the SAP system is already deployed on-premises in a 2-tier configuration. For existing systems, the SAPS related to the hardware in use often can be determined or estimated based on existing SAP benchmarks. The results can be found here: <http://global.sap.com/campaigns/benchmark/index.epx>. For newly deployed SAP systems, you should have gone through a sizing exercise which should determine the SAPS requirements of the system.

**Step 2**: For existing systems, the I/O volume and I/O operations per second on the DBMS server should be measured. For newly planned systems, the sizing exercise for the new system also should give rough ideas of the I/O requirements on the DBMS side. If unsure, you eventually need to conduct a Proof of Concept.

**Step 3:** Compare the SAPS requirement for the DBMS server with the SAPS the different VM types of Azure can provide. The information on SAPS of the different Azure VM types is documented in SAP Note [1928533 – SAP Applications on Azure: Supported products and Sizing](http://service.sap.com/sap/support/notes/1928533) . The focus should be on the DBMS VM first since the database layer is the layer in an SAP NetWeaver system that does not scale out in the majority of deployments. In contrast, the SAP application layer can be scaled out. If none of the SAP supported Azure VM types can deliver the required SAPS, the workload of the planned SAP system can’t be run on Azure. You either need to deploy the system on-premises or you need to change the workload volume for the system.

**Step 4:** As documented here: <http://msdn.microsoft.com/en-us/library/windowsazure/dn197896.aspx>, Azure enforces an IOPS quota per VHD independent whether you use Standard Storage or premium Storage. Dependent on the VM type, the number of VHDs which can be mounted varies. As a result, you can calculate a maximum IOPS number that can be achieved with each of the different VM types. Dependent on the database file layout, you can stripe VHDs to become one volume in the guest OS. However, if the current IOPS volume of a deployed SAP system exceeds the calculated limits of the largest VM type of Azure and if there is no chance to compensate with more memory, the workload of the SAP system can be impacted severely. In such cases, you can hit a point where you should not deploy the system on Azure.

**Step 5:** Especially in SAP systems which are deployed on-premises in 2-Tier configurations, the chances are that the system might need to be configured on Azure in a 3-Tier configuration. In this step, you need to check whether there is a component in the SAP application layer which can’t be scaled out and which would not fit into the CPU and memory resources the different Azure VM types offer. If there indeed is such a component, the SAP system and its workload can’t be deployed into Azure. But if you can scale-out the SAP application components into multiple Azure VMs, the system can be deployed into Azure.

**Step 6:** If the DBMS and SAP application layer components can be run in Azure VMs, the configuration needs to be defined with regard to:

* Number of Azure VMs,
* VM types for the individual components,
* Number of VHDs in DBMS VM to provide enough IOPS.

# Managing Azure Assets

## Microsoft Azure Portal

The Microsoft Azure Management Portal is one of the two interfaces to manage Azure VM deployments. The basic management tasks, like deploying VMs from images of VHDs, can be done through the portal. In addition, the creation of Storage Accounts, Virtual Networks and other Azure components are also tasks the portal can handle very well. However, functionality like uploading VHDs from on-premises to Azure or copying a VHD within Azure are tasks which require either third party tools or administration through PowerShell.

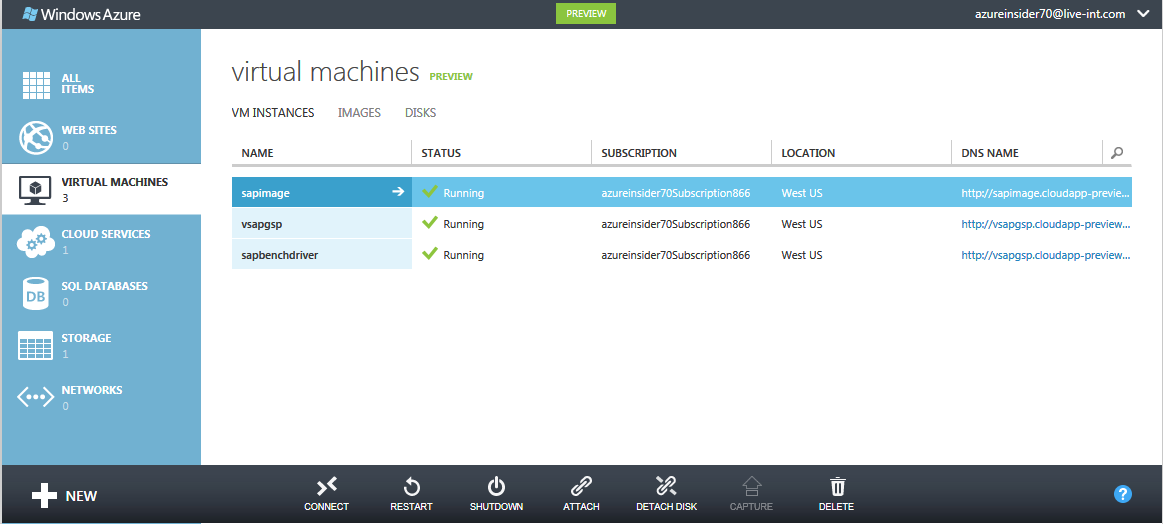


Figure Microsoft Azure Portal - Virtual Machine overview

In the section pane *Virtual Machines* you are able to see a list of currently deployed Azure Virtual Machine instances (in here it is e.g *sapimage, vsapgsp, sapbenchdriver*).

When diving into the Virtual Machine instance details (by clicking on the name), there is a lot of information about the state of the Virtual Machine instance available. For example, you can see the current workload of the Virtual Machine instance; furthermore, there is information about the hostname, the currently mounted disk drives and several identifiers available as shown below.

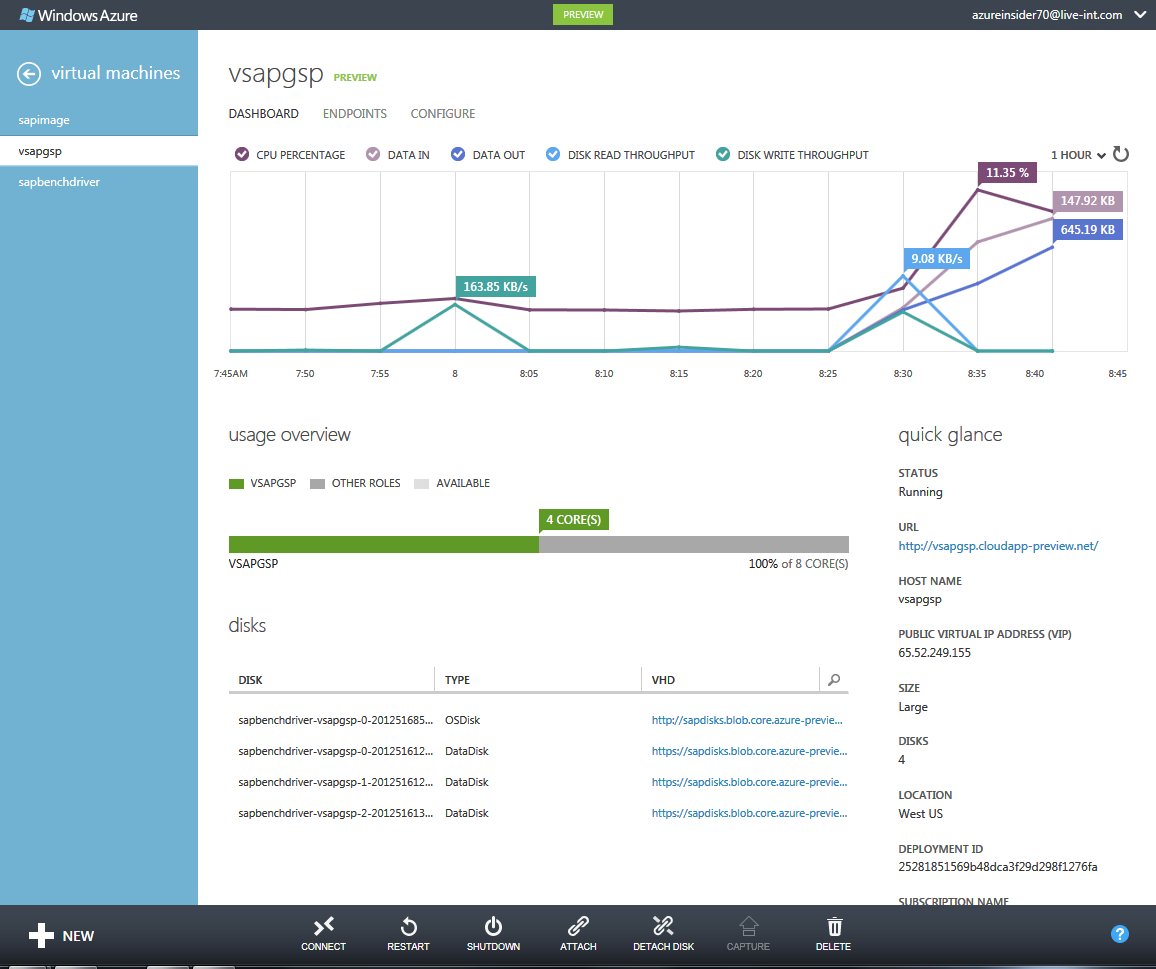


Figure Microsoft Azure Portal - Virtual Machine details and monitoring

Administration and configuration tasks for the Virtual Machine instance are possible from within the Azure Portal.

Besides restarting and shutting down a Virtual Machine you can also attach, detach and create data disks for the Virtual Machine instance, to capture the instance for image preparation, change endpoint configuration and configure the size of the Virtual Machine instance.

The Azure Portal provides basic functionality to deploy and configure VMs and many other Azure services. However not all available functionality is covered by the Azure Portal. In the Azure Portal, it’s not possible to perform tasks like:

* Uploading and downloading VHDs to or from Azure,
* Copying VMs,
* Assigning fixed IP addresses to VMs.

Also any type of automation regarding deployment is not possible with the Azure portal. Tasks such as scripted deployment of multiple VMs is not possible via the Azure Portal.

The Azure Portal may require adding an Azure Management certificate as described here: <http://msdn.microsoft.com/en-us/library/windowsazure/gg981929.aspx>

## Management via Microsoft Azure PowerShell cmdlets

Windows PowerShell is a powerful and extensible framework that has been widely adopted by customers deploying larger numbers of systems in Azure. After the installation of PowerShell cmdlets on a desktop, laptop or dedicated management station with certificates, the PowerShell cmdlets can be run remotely.

The process to enable a local desktop/laptop for the usage of Azure PowerShell cmdlets and how to configure those for the usage with the Azure subscription(s) is described here:

<http://www.windowsazure.com/en-us/documentation/articles/install-configure-powershell/?fwLinkID=320552>

More detailed steps on how to install, update and configure the Azure PowerShell cmdlets can also be found in chapter 4.1 of the document: ‘*SAP NetWeaver on Microsoft Azure Virtual Machine Services – Deployment Guide*’.

Customer experience so far has been that PowerShell (PS) is certainly the more powerful tool to deploy VMs and to create custom steps in the deployment of VMs. All of the customers running SAP instances in Azure are using PS cmdlets to supplement management tasks they do in the Azure Portal or are even using PS cmdlets exclusively to manage their deployments in Azure. Since the Azure specific cmdlets share the same naming convention as the more than 2000 Windows related cmdlets, it is an easy task for Windows administrators to leverage those cmdlets.

Deployment of the Azure Monitoring Extension for SAP (see [chapter 9.1](#_Azure_Monitoring_Solution) in this document) is only possible via PowerShell, therefore it is mandatory to setup and configure PowerShell when deploying or administering an SAP NetWeaver system in Azure.

As Azure provides more functionality, new PS cmdlets are going to be added that requires an update of the cmdlets. Therefore it makes sense to check the Azure Download site at least once the month <http://www.windowsazure.com/en-us/downloads/> for a new version of the cmdlets. The new version will just be installed on top of the older version.

For a general list of Azure related PowerShell commands check here: <http://msdn.microsoft.com/en-us/library/azure/jj554330.aspx> .

# Different ways to deploy VMs for SAP in Azure

In this chapter you will learn the different ways to deploy a VM in Azure. Additional preparation procedures, as well as handling of VHDs and VMs in azure are covered in this chapter.

## Deployment of VMs for SAP

Microsoft Azure offers multiple ways to deploy VMs and associated disks. Thus it is very important to understand the differences since preparations of the VMs might differ depending on the method of deployment. In general, we will take a look at the following scenarios:

1. **Moving a VM from on-premises to Azure with a non-generalized disk**You plan to move a specific SAP system from on-premises to Azure. This can be done by uploading the VHD which contains the OS, the SAP Binaries and DBMS binaries plus the VHDs with the data and log files of the DBMS to Azure. In contrast to scenario #2 below, you keep the hostname, SAP SID and SAP user accounts in the Azure VM as they were configured in the on-premises environment. Therefore, generalizing the image is not necessary. Please see [chapters 5.2.1](#_Uploading_a_VHD) and [5.3.1](#_Uploading_a_VHD) of this document for on-premises preparation steps and upload of non-generalized VMs or VHDs to Azure. Please read chapter 3.4 in the guide: ‘*SAP NetWeaver on Microsoft Azure Virtual Machine Services – Deployment Guide*’ for detailed steps of deploying such an image in Azure.
2. **Deploying a VM with a customer specific image**Due to specific patch requirements of your OS or DBMS version, the provided images in the Azure Gallery might not fit your needs. Therefore, you might need to create a VM using your own ‘private’ OS/DBMS VM image which can be deployed several times afterwards. To prepare such a ‘private’ image for duplication, the Windows settings (like Windows SID and hostname) must be abstracted/generalized on the on-premises VM. If you have already installed SAP content in your on-premises VM (especially for 2-Tier systems), you can adapt the SAP system settings after the deployment of the Azure VM through the instance rename procedure supported by the SAP Software Provisioning Manager (SAP Note [1619720 - System Rename for SAP Systems based on SAP NetWeaver](http://service.sap.com/sap/support/notes/1619720)). See [chapters 5.2.2](#_Preparing_an_own_2) and [5.3.2](#_Deployment_of_a) of this document for on-premises preparation steps and upload of a generalized VM to Azure. Please read chapter 3.3 in the guide: ‘*SAP NetWeaver on Microsoft Azure Virtual Machine Services – Deployment Guide*’ for detailed steps of deploying such an image in Azure.
3. **Deploying a VM out of the Azure Gallery**

You would like to use a Microsoft or 3rd party provided VM image from the Azure Gallery to deploy your VM. After you deployed your VM in Azure, you follow the same guidelines and tools to install the SAP software and/or DBMS inside your VM as you would do in an on-premises environment. For more detailed deployment description, please see chapter 3.2 in the guide: ‘*SAP NetWeaver on Microsoft Azure Virtual Machine Services – Deployment Guide*’.

## Preparing VMs with SAP for Azure

Before uploading VMs into Azure you need to make sure the VMs and VHDs fulfill certain requirements. There are small differences depending on the deployment method that is used.

### Preparation for moving a VM from on-premises to Azure with a non-generalized disk

A common deployment method is to move an existing VM which runs an SAP system from on-premises to Azure. That VM and the SAP system in the VM just should run in Azure using the same hostname and very likely the same SAP SID. In this case the VM should not be an ‘Azure image’ for multiple deployments, but an ‘Azure Disk’ for a one-time deployment. If the on-premises network got extended into Azure (see [chapter 2.2](#_Hybrid-IT_-_Deployment) in this document), then even the same domain accounts can be used within the VM as those were used before on-premises.

Requirements when preparing your own Azure VM Disk are:

* Originally the VHD containing the operating system could have a maximum size of 127GB only. This limitation got eliminated at the end of March 2015. Now the VHD containing the operating system can be up to 1TB in size as any other Azure Storage hosted VHD as well.
* It needs to be in the fixed VHD format. Dynamic VHDs or VHDs in VHDx format are not yet supported on Azure. Dynamic VHDs will be converted to static VHDs when you upload the VHD with the PowerShell commandlets.
* VHDs which are mounted to the VM and should be mounted again in Azure to the VM need to be in a fixed VHD format as well. The same size limit of the OS disk applies to data disks as well. VHDs can have a maximum size of 1TB. Dynamic VHDs will be converted to static VHDs when you upload the VHD with the PowerShell commandlets.
* Add another local account with administrator privileges which can be used by Microsoft support or which can be assigned as context for services and applications to run in until the VM is deployed and more appropriate users can be used.
* For the case of using a Cloud-Only deployment scenario (see [chapter 2.1](#_Azure-Only_-_Virtual) of this document) in combination with this deployment method, domain accounts might not work once the Azure Disk is deployed in Azure. This is especially true for accounts which are used to run services like the DBMS or SAP applications. Therefore you need to replace such domain accounts with VM local accounts and delete the on-premises domain accounts in the VM. Keeping on-premises domain users in the VM image is not an issue when the VM is deployed in the Cross-Premises scenario as described in [chapter 2.2](#_Hybrid-IT_-_Deployment) in this document.
* If domain accounts were used as DBMS logins or users when running the system on-premises and those VMs are supposed to be deployed in Cloud-Only scenarios, the domain users need to be deleted. You need to make sure that the local administrator plus another VM local user is added as a login/user into the DBMS as administrators.
* Add other local accounts as those might be needed for the specific deployment scenario.
* Make sure that drive D:\ is not used and is available as a free letter to name drives.
* Set disk automount for attached disks as described in [chapter 5.5.3](#_Setting_automount_for) in this document.

In this scenario no generalization (sysprep) of the VM is required to upload and deploy the VM on Azure.

### Preparation for deploying a VM with a customer specific image for SAP

Azure offers the possibility to create your own custom images. These will show up in the subscription’s private gallery of the Azure Portal and can be deployed either through the Azure Portal or with PowerShell cmdlets. In contrast to Azure Disks, you can deploy many VMs from an Azure Image.

Requirements when preparing your own Azure VM Image are:

* Originally the VHD containing the operating system could have a maximum size of 127GB only. This limitation got eliminated at the end of March 2015. Now the VHD containing the operating system can be up to 1TB in size as any other Azure Storage hosted VHD as well.
* It needs to be in the fixed VHD format. Dynamic VHDs or VHDs in VHDx format are not yet supported on Azure. Dynamic VHDs will be converted to static VHDs when you upload the VHD with the PowerShell commandlets.
* VHDs which are mounted to the VM and should be mounted again in Azure to the VM need to be in a fixed VHD format as well. The same size limit of the OS disk applies to data disks as well. VHDs can have a maximum size of 1TB. Dynamic VHDs will be converted to static VHDs when you upload the VHD with the PowerShell commandlets.
* Since all the Domain users registered as users in the VM will not exist in a Cloud-Only scenario (see [chapter 2.1](#_Azure-Only_-_Virtual) of this document), services using such domain accounts might not work once the Image is deployed in Azure. This is especially true for accounts which are used to run services like DBMS or SAP applications. Therefore you need to replace such domain accounts with VM local accounts and delete the on-premises domain accounts in the VM. Keeping on-premises domain users in the VM image might not be an issue when the VM is deployed in the Cross-Premise scenario as described in [chapter 2.2](#_Hybrid-IT_-_Deployment) in this document.
* Add another local account with administrator privileges which can be used by Microsoft support in problem investigations or which can be assigned as context for services and applications to run in until the VM is deployed and more appropriate users can be used.
* In Cloud-Only deployments and where domain accounts were used as DBMS logins or users when running the system on-premises, the domain users should be deleted. You need to make sure that the local administrator plus another VM local user is added as a login/user of the DBMS as administrators.
* Add other local accounts as those might be needed for the specific deployment scenario.
* If we have SAP ABAP stacks in the template and renaming of the host name from the original name at the point of the Azure deployment is likely, it is recommended to copy the latest versions of the SAP Software Provisioning Manager DVD into the template. This will enable you to easily use the SAP provided rename functionality to adapt the changed hostname and/or change the SID of the SAP system within the deployed VM image as soon as a new copy is started.
* Make sure that drive D:\ is not used and is available as a free letter to name drives.
* Set disk automount for attached disks as described in [chapter 5.5.3](#_Setting_automount_for) in this document.
* SAP GUI (for administrative and setup purposes) can be pre-installed in such a template.
* Other software necessary to run the VMs successfully in Cross-Premises scenarios can be installed as long as this software can work with the rename of the VM.

If the VM is prepared sufficiently to be generic and eventually independent of accounts/users not available in the targeted Azure deployment scenario, the last preparation step of generalizing such an image is conducted.

#### Generalizing a VM using Sysprep

The last step is to log in to a VM with an Administrator account. Open a Windows command window as ‘administrator’. Go to …\windows\system32\sysprep and execute sysprep.exe.

A small window will appear. Use these settings:

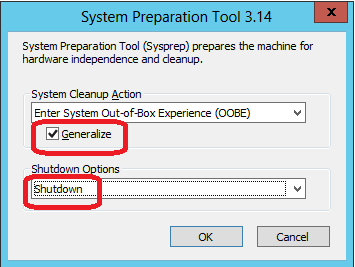


Figure Run sysprep on the virtual machine

It is important to check the ‘Generalize’ option (the default is un-checked) and change the Shutdown Option from its default of ‘Reboot’ to ‘Shutdown’. This procedure assumes that the sysprep process is executed on-premises in the Guest OS of a VM.

If you want to perform the procedure with a VM already running in Azure, the sequence as described here is a better one:

<http://social.msdn.microsoft.com/Forums/windowsazure/en-US/fafb9ee6-1e57-46ba-8440-27467ad986cf/image-capture-issue-vm-unexpectedly-started-after-guestinitiated-shutdown>.

## Transferring VMs and VHDs between on-premises to Azure

Since uploading VM images and disks to Azure is not possible via the Azure Portal, you need to use Azure PowerShell cmdlets. A second possibility is the use of the tool ‘AzCopy’. The tool can copy VHDs between on-premises and Azure (in both directions). It also can copy VHDs between Azure Regions. Please consult this documentation for download and usage of AzCopy:

<http://azure.microsoft.com/en-us/documentation/articles/storage-use-azcopy/>

A third alternative would be to use various third party GUI oriented tools. However, please make sure that these tools are supporting Azure Page Blobs. For our purposes **we need to use Azure Page Blob store** (the differences are described here: <http://msdn.microsoft.com/en-us/library/windowsazure/ee691964.aspx>). Also the tools provided by Azure are very efficient in compressing the VMs and VHDs which need to be uploaded. This is important because this efficiency in compression reduces the upload time (which varies anyway depending on the upload link to the internet from the on-premises facility and the Azure deployment region targeted). It is a fair assumption that uploading a VM or VHD from European location to the U.S. based Azure datacenters will take longer than uploading the same VMs/VHDs to the European Azure Datacenters.

The process of uploading VMs and VHDs to Azure in order to deploy or mount those VHDs is slightly different depending on whether an image should be uploaded or a VHD with an Operating System or a VHD which should be mounted to a deployed VM. The flow diagram of these three different deployment methods looks like this:



Figure Three different paths of uploading and deploying VMs and VHDsIn the following two sections, we will describe more details on these different methods using PowerShell commands.

### Difference Between an Azure Disk and Azure Image

Azure Disks and Azure Images are logical constructs that are built up on top of a physical VHD that is stored in the Azure Blob Store. A VHD that contains a bootable operating system can be declared as either an Azure Disk or an Azure Image. The big difference is that an Azure Disk that contains an operating system can be deployed only one time. Whereas you can deploy many VMs from an Azure Image. We assume that the operating system contained in the VHD that was made an Azure Image was generalized/sys-prep’ed.

Looking at the different layers, the difference between Azure Disk and Azure Image might be best shown with the graphics below

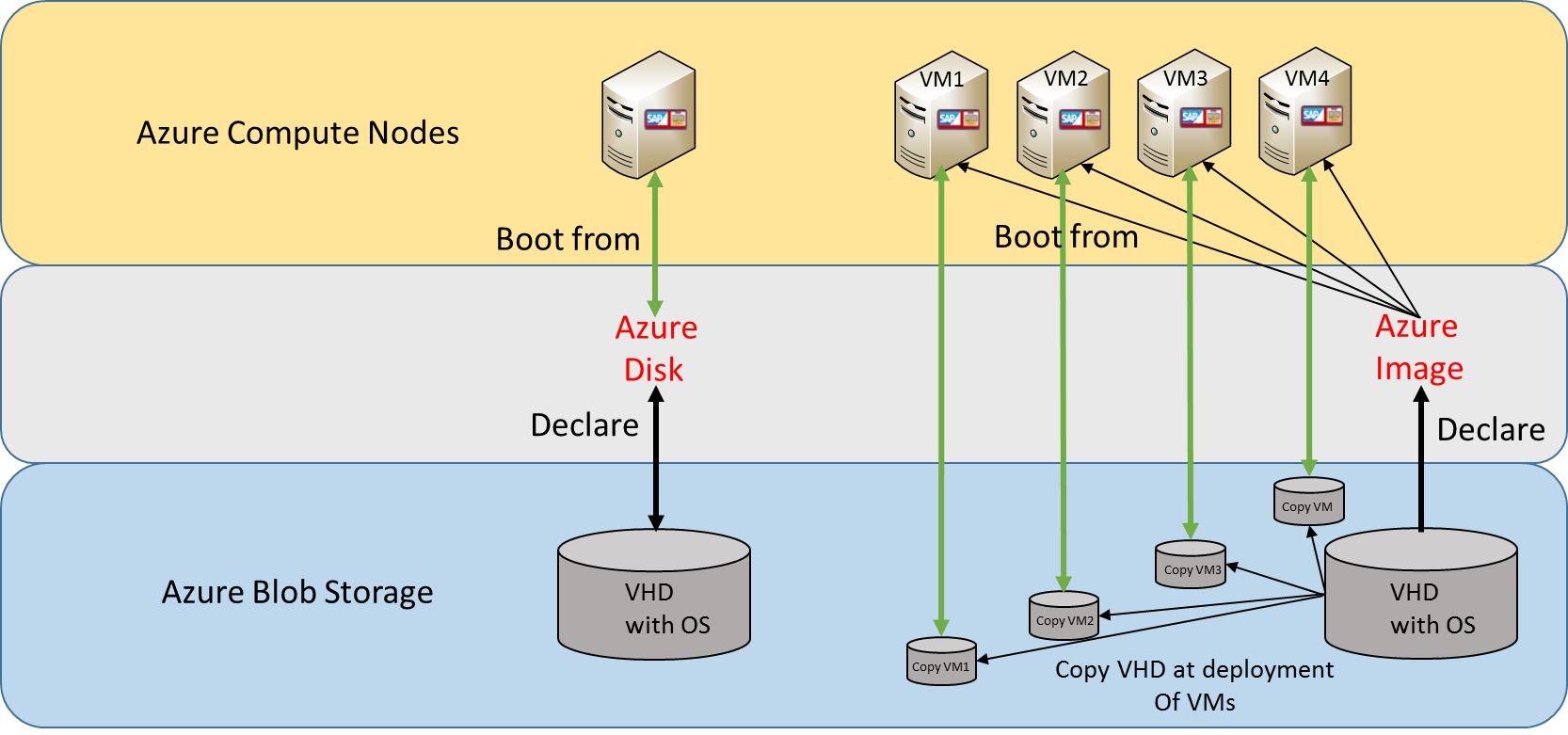


Figure Relationship between VHDs, Azure Disks and Azure Images

The figure shows graphically the relationship between the physical VHD with a bootable operating system and the different possibilities of using such a VHD.

In the left case the VHD is declared as Azure Disk. In this case, a VM deployment boots directly from the VHD. All the changes made during using the deployed VM are persisted in this VHD. Therefore, the VHD can be used by one deployed VM only at one point in time. Between the entities of VHD, Azure Disk and VM we have a 1:1:1 relationship.

In the right case, a VHD is declared as Azure Image. At deployment of a VM out of the Azure Image, Azure Storage will copy the VHD and the VM is booted of the copy of the original VHD. Changes made within the VM are persisted in the individual copy of the VHD and NOT in the original VHD. However, you can only create one single Azure Image per VHD. Between the entities of VHD, Azure Image and VM we have a 1:1:n relationship.

### Uploading a VHD from on-premises to Azure

To upload an existing VM or VHD from the on-premises network such a VM or VHD needs to meet the requirements as listed in [chapter 5.2.1](#_Preparing_an_own_1) of this document.

Such a VM does NOT need to be generalized with sysprep.exe and can be uploaded in the state and shape it has after shutdown on the on-premises side. The same is true for additional VHDs which don’t contain any operating system.

#### Uploading a VHD and making it an Azure Disk

In this case we want to upload a VHD, either with or without an OS in it, and make it a disk in Azure which then can be mounted to a VM or which can be deployed as a VM if the VHD contains a bootable operating system. This is a two-step process using the PowerShell cmdlets:

* **Add-AzureVHD**; Used to upload the VHD into an Azure Storage Account: <http://msdn.microsoft.com/en-us/library/windowsazure/dn205185.aspx>
* **Add-AzureDisk**; Will add an uploaded VHD to the disk repository so that it can be used to be mounted against a VM (<http://msdn.microsoft.com/en-us/library/windowsazure/jj152877.aspx> ) or it can be deployed as VM if the Azure disk contains a bootable OS.

An example of how PowerShell cmdlets can be used to upload a VM built out of various VHDs can be found here:

<http://michaelwasham.com/2013/01/04/migrate-a-virtual-machine-to-windows-azure-with-powershell/>

The upload procedure itself will not differ. The uploaded disk(s) with the base VMs or VHDs will show up in the Azure Portal in the private disk gallery.

### Deployment of a VM Image

To upload an existing VM or VHD from the on-premises network in order to use it as an Azure VM image such a VM or VHD need to meet the requirements listed in [chapter 5.2.2](#_Preparing_an_own_2) of this document.

Such a VM needs to be prepared with sysprep.exe.

#### Uploading a VM Image with PowerShell

We look into a two-step process using two PowerShell cmdlets to achieve this:

* **Add-AzureVHD**; Used to upload the VHD into an Azure Storage Account: <http://msdn.microsoft.com/en-us/library/windowsazure/dn205185.aspx>
* **Add-AzureVMImage**; is used to create an image out of an uploaded VHD which has an OS in it: <http://msdn.microsoft.com/en-us/library/windowsazure/jj152881.aspx>

This command will add a VHD with an OS that has been prepared with sysprep.exe to the Azure subscription’s private gallery of Images.

### Downloading VHDs to on-premises

Azure Infrastructure as a Service is not a one-way street of only being able to upload VHDs and SAP systems. You can move SAP systems from Azure back into the on-premises world as well.

Once the SAP system is stopped and the VM is shutdown, you can use the PowerShell cmdlet **Save-AzureVHD** on the on-premises target to download the VHD disks back to the on-premises world. In order to do that, you need the URL of the VHD which you can find in the ‘Storage Section’ of the Azure Portal (need to navigate to the Storage Account and the storage container where the VHD was created) and you need to know where the VHD should be copied to.

Then you can leverage the command by simply defining the parameter –Source as the URL of the VHD to download and the LocalFilePath as the physical location of the VHD (including its name). The command could look like:

PSC:\Windows\system32> **Save-AzureVhd -Source** http://s...........blob.core.windows.net/v...../sapidedata.vhd **-LocalFilePath** E:\\Azure\_downloads\sapidesdata.vhd

For more details of the **Save-AzureVHD** cmdlet, please check here: <http://msdn.microsoft.com/en-us/library/azure/dn495297.aspx>.

During the time of the download the VHDs can’t be active. Even when downloading VHDs which are mounted to VMs, the VM needs to be shutdown. If you only want to download the database content which then should be used to set up a new system on-premises and if it is acceptable that during the time of the download and the setup of the new system that the system in Azure can still be operational, you could avoid a long downtime by performing a compressed database backup into a VHD and just download that VHD instead of also downloading the OS base VM.

## Transferring VMs and VHDs within Azure

### ****Copying SAP systems within Azure****

An SAP system or even a dedicated DBMS server supporting an SAP application layer will likely consist of several VHDs which contain either the OS with the binaries or the data and log file(s) of the SAP database. Neither the Azure functionality of copying VHDs nor the Azure functionality of saving VHDs to disk has a synchronization mechanism which would snapshot multiple VHDs synchronously. Therefore, the state of the copied or saved VHDs even if those are mounted against the same VM would be different. This means that in the concrete case of having different data and logfile(s) contained in the different VHDs, the database in the end would be inconsistent.

**Conclusion: In order to copy or save VHDs which are part of an SAP system configuration you need to stop the SAP system and also need to shut down the deployed VM. Only then can you copy or download the set of VHDs to either create a copy of the SAP system in Azure or on-premises.**

Data disks are stored as VHD files in an Azure Storage Account. Having uploaded a VHD file you can create an Azure Disk from the VHD file. You can only create one Azure Disk from each VHD-file, i.e. if you want to create a set of identical data disks you first have to replicate the VHD-file by copying. The name of the VHD file in Azure must be unique within Azure. As mentioned earlier already, the name is kind of a three-part name that looks like:

http(s)://<storage account name>.blob.core.windows.net/<container name>/<vhd name>

You can use Azure PowerShell cmdlets to copy a VHD. A second possibility is to use a third party storage browser to make a copy of the VHD:

<http://azurestorageexplorer.codeplex.com/downloads/get/391105>

There also are professional editions of Azure Storage Explorers which can be found here:

<http://www.cerebrata.com/>

<http://clumsyleaf.com/products/cloudxplorer>

The usage of Azure PS cmdlets is described in an example here: <http://michaelwasham.com/2013/03/27/windows-azure-powershell-cmdlets-now-supports-storage/> .

The copy of a VHD itself is a process which takes only a few seconds (similar to SAN hardware creating snapshots with lazy copy and copy on write).

After you have a copy of the VHD file you can start to create a disk. A simple PS cmdlet to do this is:

* **Add-AzureDisk** <http://msdn.microsoft.com/en-us/library/dn495252.aspx>

### Copying disks between Azure Storage Accounts

This task cannot be performed on the Azure Management Portal. As alternatives you have Azure PowerShell cmdlets you could use or a third party storage browser. The PowerShell cmdlets can create and manage blobs, which include the ability to asynchronously copy blobs across Storage Accounts and across regions within the Azure subscription. A second possibility is to use one of the third party Azure storage browser which proved to be very capable and supporting Azure Standard as well as Premium Storage.

Copying VHDs between subscriptions is also possible. An example of a script doing so can be downloaded or reviewed here: <http://gallery.technet.microsoft.com/scriptcenter/Copy-all-VHDs-in-Blog-829f316e>.

The basic flow of the PS cmdlet logic looks like this:

* Define the source blob file.
* Define the target blob file.
* Connect to the source Azure Storage Account and get the source blob URI.
* Connect to the target Storage Account and create the target container if necessary.
* Set the source Storage Account.
* Copy the blob to the target container.
* Check the status of the copy in a loop.
* Create a disk from the blob file.

For examples see: <http://michaelwasham.com/windows-azure-powershell-reference-guide/copying-vhds-blobs-between-storage-accounts/>.

## Disk Handling

### VM/VHD structure for SAP deployments

Ideally the handling of the structure of a VM and the associated VHDs should be very simple. In on-premises installations, customers developed many ways of structuring a server installation. With many customers we saw configurations where, for example, SAP and DBMS binaries were not installed on the c:\ drive where the OS was installed. There were various reasons for this, but when we went back to the root, it usually was that the drives were small and OS upgrades needed additional space 10-15 years ago. Both conditions do not apply these days too often anymore. Today the c:\ drive can be mapped on large volume disks or VMs. In order to keep deployments simple in their structure, it is recommended to follow the following deployment pattern for SAP NetWeaver systems in Azure:

* One base VHD which contains the OS and all the binaries of the DBMS and/or SAP. Since March 2015, this VHD can be up to 1TB in size instead of earlier restrictions that limited it to 127GB.
* One or multiple VHDs which contains the DBMS log file of the SAP database and the log file of the DBMS temp storage area (if the DBMS supports this). If the database log IOPS requirements are high, you need to stripe multiple VHDs in order to reach the IOPS volume required.
* A number of VHDs containing one or two database files of the SAP database and the DBMS temp data files as well (if the DBMS supports this).
* The Operating System pagefile should be on the D: drive (non-persistent disk).



Figure 13 Reference Configuration of Azure IaaS VM for SAP

The number of VHDs used for the DBMS data files and the type of Azure Storage these VHDs are hosted on should be determined by the IOPS requirements and the latency required. Exact quotas are described here: <http://msdn.microsoft.com/en-us/library/windowsazure/dn197896.aspx>

Experience of SAP deployments over the last 2 years taught us some lessons which can be summarized as:

* IOPS traffic to different data files is not always the same since existing customer systems might have differently sized data files representing their SAP database(s). As a result it turned out to be better using Windows Storage Spaces or Striping over multiple VHDs to place the data files LUNs carved out of those.
* There were situations, especially with Azure Standard Storage where an IOPS rate hit the quota of a single VHD against the DBMS transaction log. In such scenarios the use of Premium Storage is recommended or alternatively aggregating multiple Standard Storage VHDs with Windows Storage Spaces
* Premium Storage is showing significant better performance, especially for critical transaction log writes. For SAP scenarios that are expected to deliver production like performance, it is highly recommended to use VM-Series that can leverage Azure Premium Storage.

Keep in mind that the VHD which contains the OS, and as we recommend, the binaries of SAP and SQL Server (base VM) as well, **is not anymore limited to 127GB**. **It now can have up to 1TB in size**. This should be enough space to keep all the necessary file including e.g. SAP batch job logs.

For more suggestions and more details, specifically for DBMS VMs, please consult the guide: ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’

### Disk Handling

In most scenarios you need to create additional disks in order to deploy the SAP database into the VM. We talked about the considerations on number of VHDs in [chapter 5.5.1](#_VM/VHD_structure_for) of this document. The Azure Portal allows to attach and detach disks once a base VM is deployed. The disks can be attached/detached when the VM is up and running as well as when it is stopped. When attaching a disk, the Azure Portal offers to attach an empty disk or an existing disk which at this point in time is not attached to another VM.

**Note:** Azure Disks can be attached to only one VM at any given time.



Figure Attach / detach disks with Azure Standard Storage

You need to decide whether you want to create a new and empty VHD (which would be created in the same Storage Account as the base VM is in) or whether you want to select an existing VHD that was uploaded earlier and should be attached to the VM now.

**IMPORTANT**: You **DO NOT** want to use Host Caching with Azure Standard Storage. You should leave the Host Cache preference at the default of NONE. With Azure Premium Storage you should enable Read Caching if the I/O characteristic is mostly read like typical I/O traffic against database data files. In case of database transaction log file no caching is recommended.

If disks are attached, you need to log in into the VM to open the Windows Disk Manager. If automount is not enabled as recommended in [chapter 5.5.3](#_Setting_automount_for), the newly attached volume needs to be taken online and initialized. If the new disk is an empty disk, you need to format the disk as well. For formatting, especially for DBMS data and log files the same recommendations as for bare-metal deployments of the DBMS apply.

As already mentioned in [chapter 3.2](#_Storage:_Microsoft_Azure), an Azure Storage account does not provide infinite resources in terms of I/O volume, IOPS and data volume. Usually DBMS VMs are most affected by this. It might be best to use a separate Storage Account for each VM if you have few high I/O volume VMs to deploy in order to stay within the limit of the Azure Storage Account volume. Otherwise, you need to see how you can balance these VMs between different Storage accounts without hitting the limit of each single Storage Account. More details are discussed in the ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’. You should also keep these limitations in mind for pure SAP application server VMs or other VMs which eventually might require additional VHDs.

Another topic which is relevant for Storage Accounts is whether the VHDs in a Storage Account are getting Geo-replicated. Geo-replication is enabled or disabled on the Storage Account level and not on the VM level. If geo-replication is enabled, the VHDs within the Storage Account would be replicated into another Azure Datacenter within the same region. Before deciding on this, you should think about the following restriction:

Azure Geo-replication works locally on each VHD in a VM and does not replicate the IOs in chronological order across multiple VHDs in a VM. Therefore, the VHD that represents the base VM as well as any additional VHDs attached to the VM are replicated independent of each other. This means there is no synchronization between the changes in the different VHDs. The fact that the IOs are replicated independently of the order in which they are written means that geo-replication is not of value for database servers that have their databases distributed over multiple VHDs. In addition to the DBMS, there also might be other applications where processes write or manipulate data in different VHDs and where it is important to keep the order of changes. If that is a requirement, geo-replication in Azure should not be enabled. Dependent on whether you need or want geo-replication for a set of VMs, but not for another set, you can already categorize VMs and their related VHDs into different Storage Accounts that have geo-replication enabled or disabled.

### ****Setting automount for attached disks****

For VMs which are created from own Images or Disks, it is necessary to check and possibly set the automount parameter. Setting this parameter will allow the VM after a restart or redeployment in Azure to mount the attached/mounted drives again automatically. The parameter is set for the images provided by Microsoft in the Azure Gallery.

In order to set the automount, please check the documentation of the command line executable diskpart.exe here: <http://technet.microsoft.com/en-us/library/cc766465(v=WS.10).aspx>

And here:

<http://technet.microsoft.com/en-us/library/cc753703.aspx#BKMK_examples>

The Windows command line window should be opened as administrator.

## ****Final Deployment****

For the final Deployment and exact steps, especially with regards to the deployment of SAP Extended Monitoring, please refer to the Guide: ‘*SAP NetWeaver on Azure Virtual Machine Services - Deployment Guide*’.

# Accessing SAP systems running within Azure VMs

For Cloud-Only scenarios, you might want to connect to those SAP systems across the public internet using SAP GUI. For these cases, the following procedures need to be applied.

Later in the document we will discuss the other major scenario, connecting to SAP systems in Cross-Premises deployments which have a site-to-site connection (VPN tunnel) or Azure ExpressRoute connection between the on-premises systems and Azure systems.

## Remote Access to SAP systems

To enable Remote Access into an Azure Virtual Machine, you have to enable so called *Endpoints* for the appropriate Virtual Machine instance. *Endpoints* can be described as publically opened ports to communicate with Virtual Machine instances from outside of Microsoft Azure.

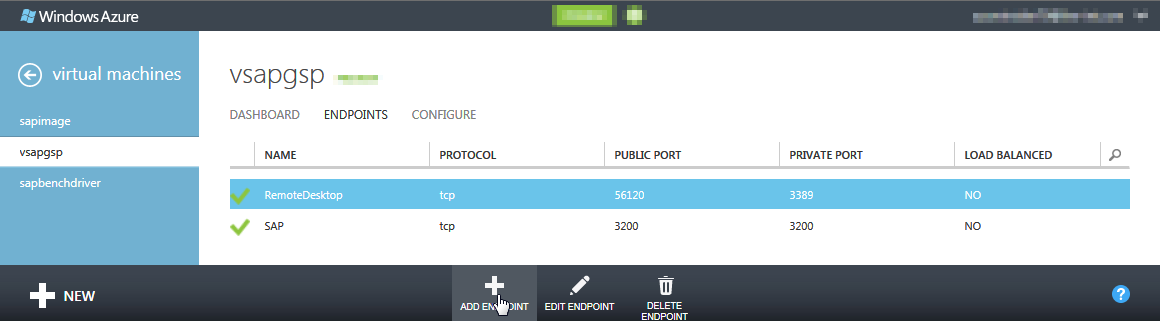


Figure Adding endpoints for external access

This needs to be done due to a security concept in the Azure Virtual Machine Services, which does not allow external access to the Virtual Machine without the appropriate endpoint configuration.

First, open the virtual machine detail page and change to the Endpoints tab (see ). By clicking on *Add endpoint*, the dialogue for adding endpoints will appear.

In the process of creating an endpoint, the public and private port numbers have to be defined. Azure will map the public port to the private/internal port which is not published. To reach the SAP system and particularly the dispatcher with the system number 00, the private port 3200 has to be exposed by an endpoint. Either you select identical port numbers, or use different ones. If your SAP instance number is different from 00, the endpoint which needs to be exposed needs to be 3200 + Instance-Number. E.g. if your instance number is 15, you would need to expose port 3215. If you have multiple VMs running SAP instances within the same Azure Cloud Service and the SAP instances have the same instance ID, you need to map the private port 32nn to a different public port for each SAP system and configure your SAP GUI accordingly.

Endpoints can be secured by ACLs as well. More information can be found here: <http://azure.microsoft.com/en-us/documentation/articles/virtual-machines-set-up-endpoints/>.

### Configuration of the SAP System and SAP GUI connectivity for Cloud-Only scenario

Please see this article which describes details to this topic: <http://blogs.msdn.com/b/saponsqlserver/archive/2014/06/24/sap-gui-connection-closed-when-connecting-to-sap-system-in-azure.aspx>

### Changing Firewall Settings within VM

By default, the Windows Firewall within an Azure deployed VM is turned on. You now need to allow the SAP Port to be opened, otherwise the SAP GUI will not be able to connect.

To do this:

* Open Control Panel\System and Security\Windows Firewall to ‘Advanced Settings’.
* Now right-click on Inbound Rules and chose ‘New Rule’.
* In the following Wizard chose to create a new ‘Port’ rule.

In the next step of the wizard, leave the setting at TCP and type in the port number you want to open. Since our SAP instance ID is 00, we took 3200. If your instance has a different instance number, the port you defined earlier based on the instance number should be opened.

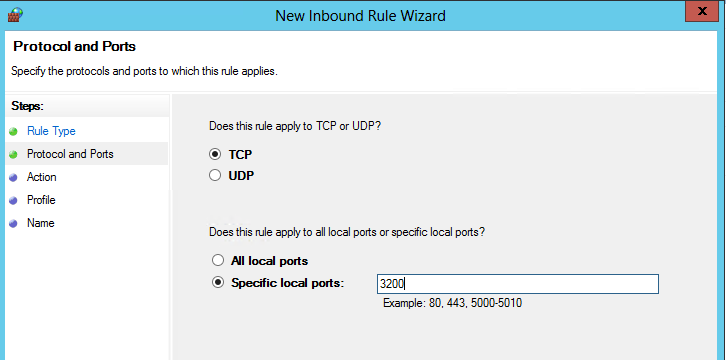


Figure Port rule definition

In the next part of the wizard, you need to leave the item ‘Allow Connection’ checked.

In the next step of the wizard you need to define whether the rule applies for Domain, Private and Public network. Please adjust it if necessary to your needs. However, connecting with SAP GUI from the outside through the public network, you need to have the rule applied to the public network.

In the last step of the wizard, you need to give the rule a name and then save the rule by pressing ‘Finish’

The rule becomes effective immediately.

### Security recommendations

The SAP GUI does not connect immediately to any of the SAP instances (port 32xx) which are running, but first connects via the port opened to the SAP message server process (port 36xx). In the past the very same port was used by the message server for the internal communication to the application instances. To prevent on-premises application servers from inadvertently communicating with a message server in Azure the internal communication ports can be changed. It is highly recommended to change the internal communication between the SAP message server and its application instances to a different port number on systems that have been cloned from on-premises systems, such as a clone of development for project testing etc. This can be done with the default profile parameter:

rdisp/msserv\_internal

as documented in: <http://help.sap.com/saphelp_spm21_bw/helpdata/en/4e/cffdb69d10424e97eb1d993b1e2cfd/frameset.htm>

More recent SAP installations are supposed to have this parameter activated and should have a different port defined as communication port.

## Connecting SQL Server Graphical User Interface Tools to SQL Server in Azure VMs

Again this description assumes a Cloud-Only scenario where we don’t have a site-to-site, multi-site or ExpressRoute connectivity connecting our assets in Azure into the corporate network. Even without a direct network connecting, there might be a need to connect SQL Server Management Tools installed on-premises on laptops or desktops to the SQL Server instances running in Azure VMs. The process to enable this is more or less a duplicate of what was described above for the SAP GUI.

A detailed description with screenshot sequences can be found here: <http://www.windowsazure.com/en-us/manage/windows/common-tasks/install-sql-server/>

# Concepts of Cloud-Only deployment of SAP instances

## Single VM with SAP NetWeaver demo/training scenario

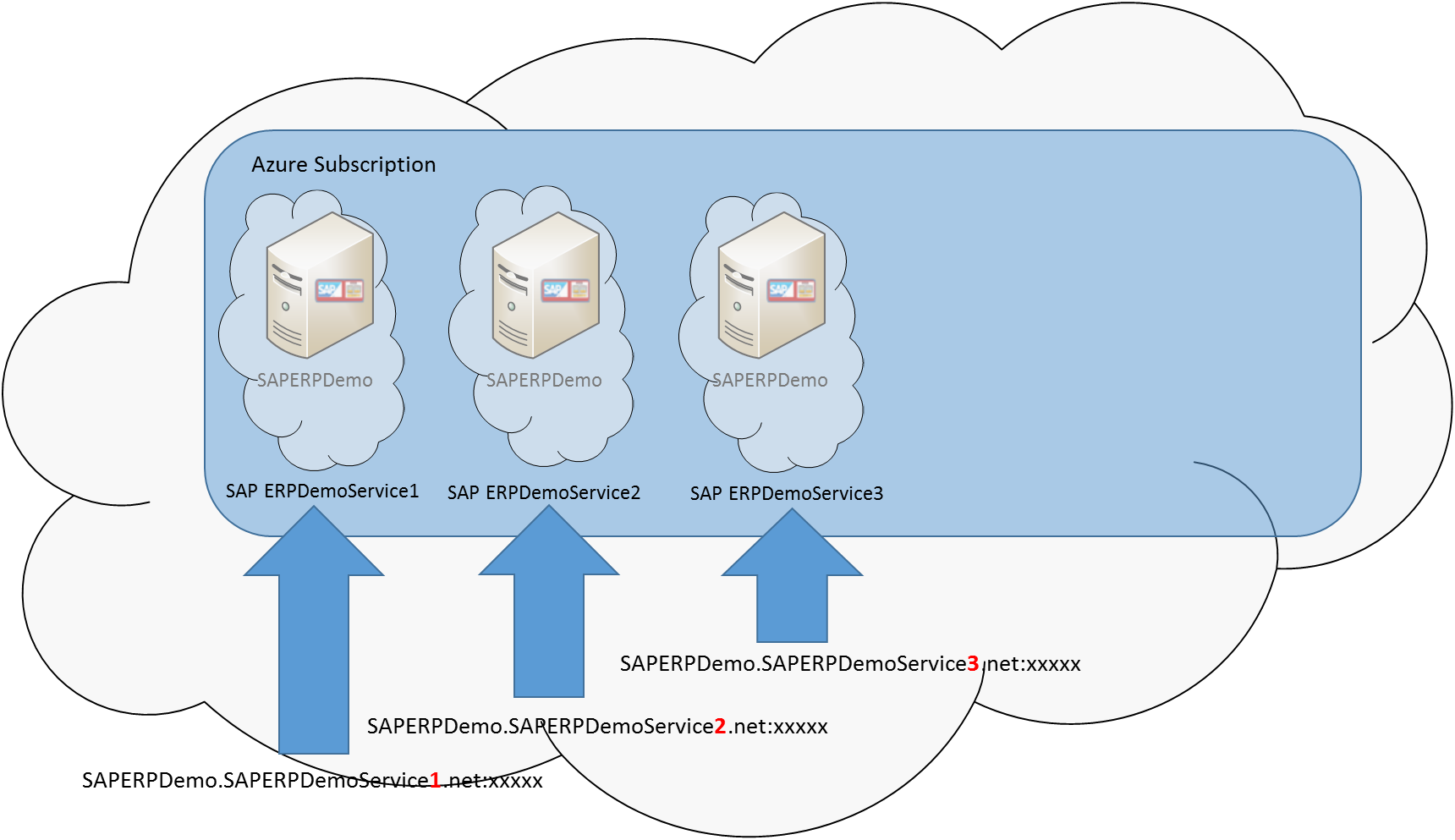


Figure Running single VM SAP Demo systems with the same VM names, isolated in Azure Cloud Services

In this scenario (see [chapter 2.1](#_Single_VM_deployments) of this document) we are implementing a typical training/demo system scenario where the complete training/demo scenario is contained within a single VM. We assume that the deployment is done through VM image templates. We also assume that multiple of these demo/trainings VMs need to be deployed with the VMs having the same name.

The assumption is that you created a VM Image as described in some sections of [chapter 5](#_Different_ways_to).2 in this document.

The sequence of events to implement the scenario looks like this:

* Select the image in the Azure Portal or find it in the private Azure image gallery with a PowerShell cmdlet.
* Define the name of the VM. For the Cloud-Only scenario every VM will have the same name. The SAP SID of the SAP NetWeaver instances in those VMs will be the same as well. Within the Azure Cloud Service, the name of the VM needs to be unique, but in different Azure Cloud Services you can run VMs with the same name since Domain Name Resolution is limited to the Cloud Service Boundaries.
* The Default ‘Administrator’ account of Windows is not valid anymore in Azure deployed Windows VMs. Therefore, a new administrator user name needs to be defined together with a password.
* The size of the VM also needs to be defined.
* In in this scenario, you need to create a new Cloud Service Name for every new VM. A public TCP/IPv4 address will be assigned to the cloud service and <DNS NAME>.cloudapp.net will be used as DNS name. The cloud service name (DNS NAME) must be unique within Microsoft Azure. The Cloud Service will provide access to the virtual machine inside via port forwarding (endpoint configuration as described in [chapter 6](#_Accessing_SAP_systems) of this document). Since you want to name every demo or trainings VM the same, each of the VMs needs to be deployed in a separate Cloud Service.
* The Region/Network needs to be decided on. There are several options to define for different scenarios:
  + We only specify a Region. This means Azure will deploy the VM without trying to optimize the location of the VM relative to the Azure storage location. In an extreme case, the VM could get deployed in a different datacenter in that region than it is stored on in Azure Storage. This is not an optimal setting.
  + Azure Virtual Networks offer a different type of sectioning. Azure Virtual Networks allow to define IP address ranges for VMs and allow many more capabilities for Cross-Premises scenarios where Azure Virtual Networks are essential. Detailed information is provided at <http://msdn.microsoft.com/en-us/library/windowsazure/jj156007.aspx>. A dedicated Virtual Network in such a configuration would only make sense if you have another trainings/demo landscape that you want to deploy besides this one trainings/demo landscape. You also would then have one Virtual Network per landscape.
* Defining an Azure Availability Set: In this scenario where it is all about single VMs that operate independently of each other, there is no need to define an Azure Availability Set. For details on Azure Availability Sets, please read [chapter 3.1.3](#_Azure_Availability_Sets) of this document.
* Endpoints: If you deploy through the Azure Portal, two public endpoints are suggested. You can add additional endpoints or delete suggested endpoints. However, you should not delete the RDP endpoint.
* Adding additional disks and restore necessary content.
* For access to the VMs, through RDP using the Azure Portal, the addresses then will look like displayed in Figure 17 where the VM name is the same. However, the Cloud Service name as second component does change. ‘xxxxx’ symbolizes the public port for the RDP endpoint that Azure assigned to the RDP connection to the VM.

Achtung Be aware:

* + All data disk names must be unique within your Azure subscription.
  + All blob names (URLs to the blobs) must be unique within Azure.
  + If you clone a virtual machine from one image and have attached the data disks in the exact same order, then all drive letters are set the same way in all Windows systems (even though the drive signature differs).
  + The cloud service gateway is reachable via its DNS Name.
  + All public port numbers of the default RDP endpoint (forwarding to port 3389) are set randomly by Azure. You may set the public port number explicitly to ease access. Otherwise you have to identify the public port number of each virtual machine either using the Azure Management Portal or the PowerShell Get-AzureEndpoint cmdlet.

## Implement a set of VMs which need to communicate within Azure

This Cloud-Only scenario is a typical scenario for training and demo purposes where the software representing the demo/training scenario is spread over multiple VMs. The different components installed in the different VMs need to communicate with each other. Again, in this scenario no on-premises network communication or Cross-Premises scenario is needed.

This scenario is an extension of the installation described in [chapter 7.1](#_Single_VM_with) of this document. In this case more virtual machines will be added to an existing cloud service. In the following example the training landscape consists of an SAP ASCS/SCS VM, a VM running a DBMS and an SAP Application Server instance VM.

Before you build this scenario you need to think about basic settings as already exercised in the scenario before.

### Cloud Service and Virtual Machine naming

All cloud service names must be unique. Develop your own naming scheme of your services, such as <ServiceName>-Suffix. The DNS-Name will set the cloud service name, which will usually not be displayed in the Azure Management Portal until two or more virtual machines have been added to the service.

The virtual machine name has to be unique within the cloud service.

### Setup Network for communication between the different VMs

There are three ways to setup the network for this scenario:

#### ****Azure** **network****

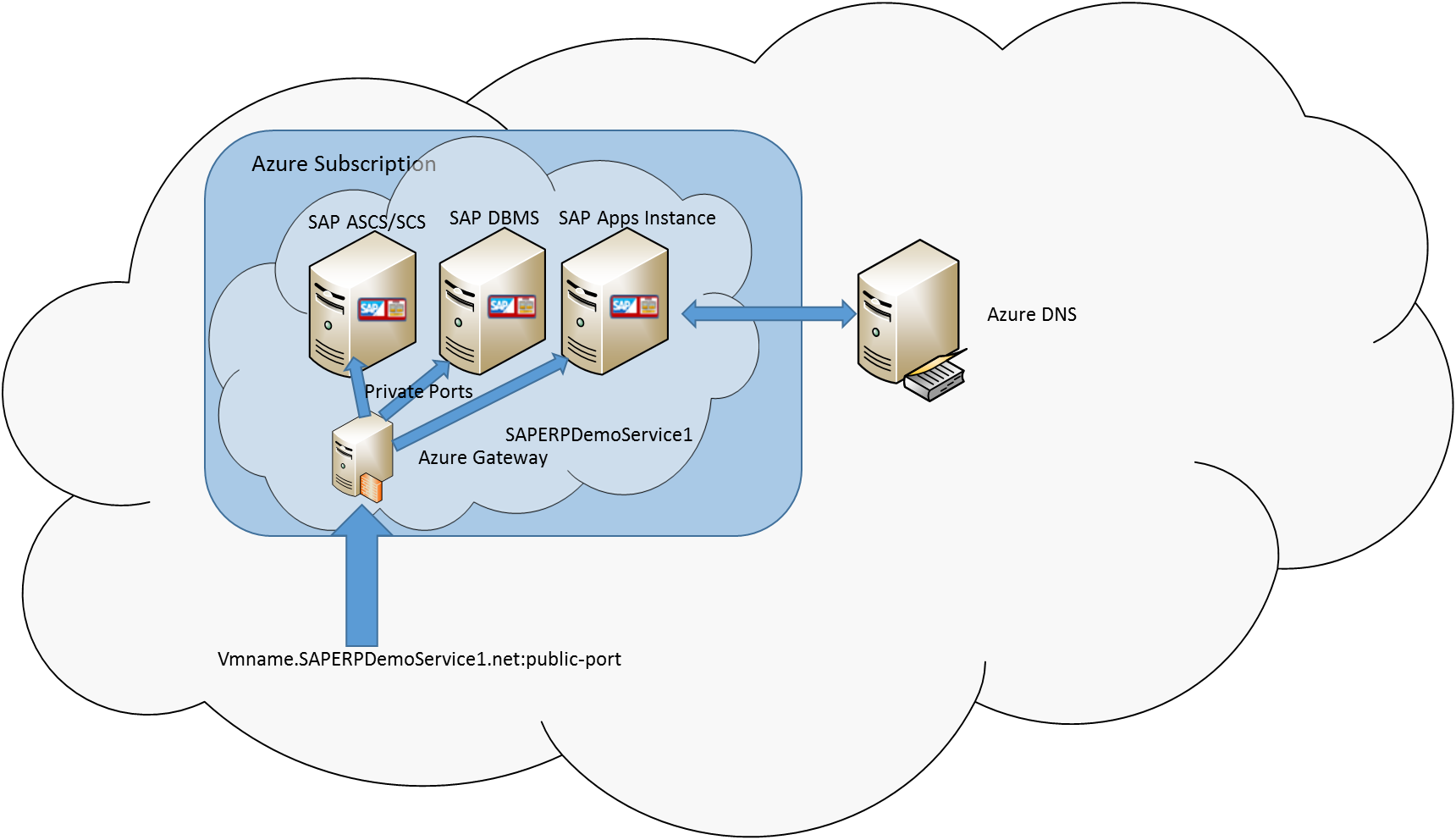


Figure Set of VMs within one Azure Cloud Service using Azure DNS

The easiest way to get the virtual machines to communicate with each other in a private network is just to define an Azure Cloud Service and have the VMs added to that Cloud Service. If you neither specify a virtual network, nor give a DNS server name at setup, Azure will configure the cloud service to use the built-in Azure naming service. As shown in Figure 18, the IP-configuration will be set up by Azure DHCP and the addresses may change after restart of the VM. All virtual machines in the same cloud service will be in the same private network and the Azure naming service enables the communications between the VMs by hostnames within the Cloud Service. All communication to and from outside the network will be routed over the gateway, i.e. the public IP and ports of the cloud service. The virtual machines inside the cloud service can be reached by port forwarding. Therefore, you have to define an endpoint for each virtual machine service you want to connect to specifying a unique public port number.

Each cloud service will have its own ‘private network’ established.

#### Virtual Network

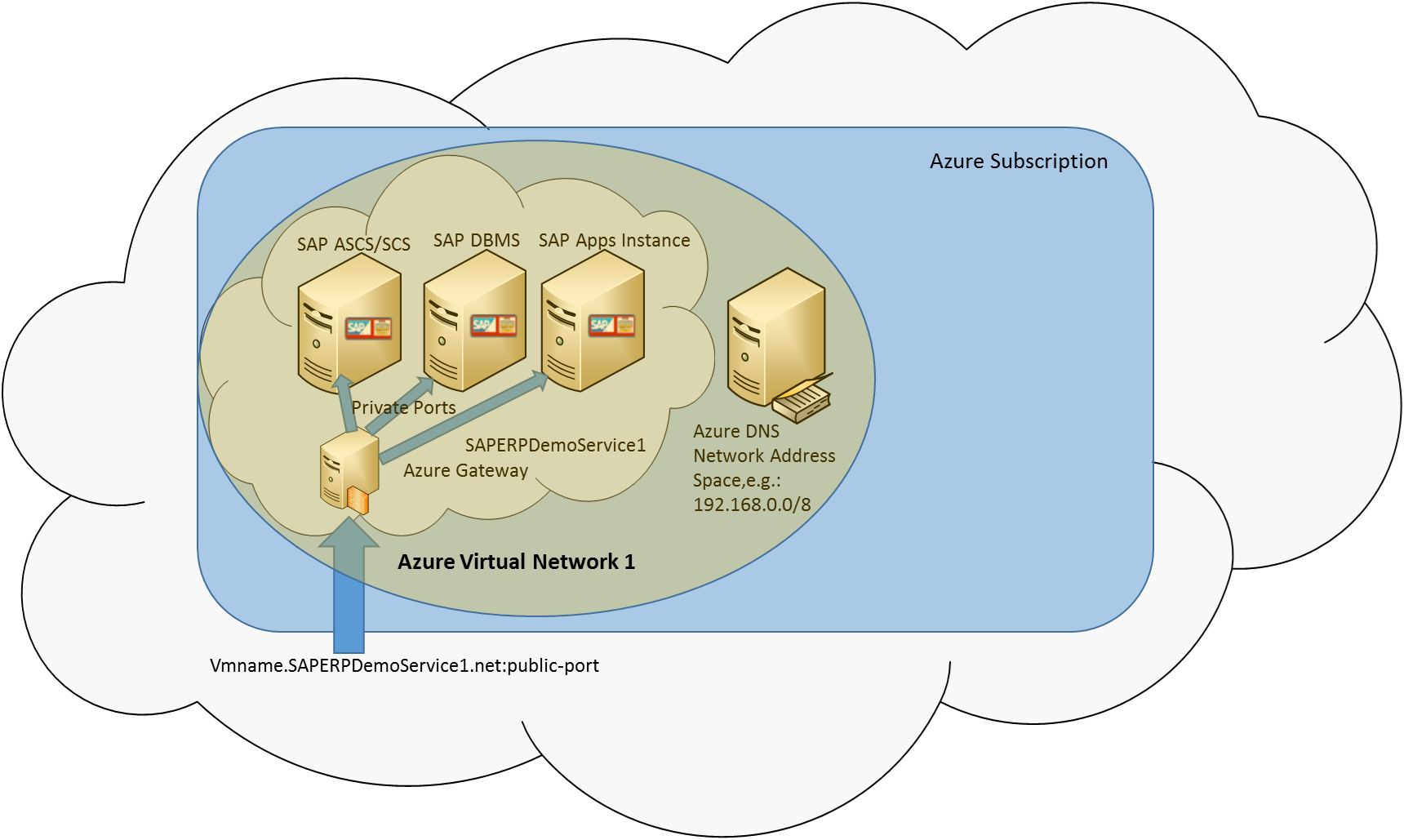


Figure Set of VMs within an Azure Virtual Network

If you want to have more control over the network settings, you need to create an Azure Virtual Network. DNS name resolution will be provided by Azure or you can configure your own DNS server outside Azure (not to be further discussed here). In this scenario we do not configure our own DNS. For all virtual machines inside one Azure Virtual Network communication via hostnames will be enabled.

The reasons to separate training or demo landscapes by Virtual networks and not only Cloud services could be:

* The SAP landscape as set up needs its own AD/Domain and a Windows Domain Server needs to be part of each of the landscapes.
* The SAP landscape as set up has components that need to work with fixed IP addresses (see [chapter 3.3.1](#_Azure_Virtual_Networks) in this document).

More details about Azure Virtual Networks and how to define them can be found here: <http://msdn.microsoft.com/en-us/library/windowsazure/jj156007.aspx>

**Note:** The fact that the VMs run within an Azure Virtual Network, does not eliminate the requirement for the associated VMs to run within the same cloud service to be able to communicate with each other using hostnames. Assuming a Cloud-Only deployment, within an Azure Virtual Network, VMs can only communicate with a VM in different Azure Cloud Services (within the same Azure Virtual Network) using the IP address. The hostnames will not be resolved. Please note this is different in Cross-Premises deployments.

#### Sub-networks

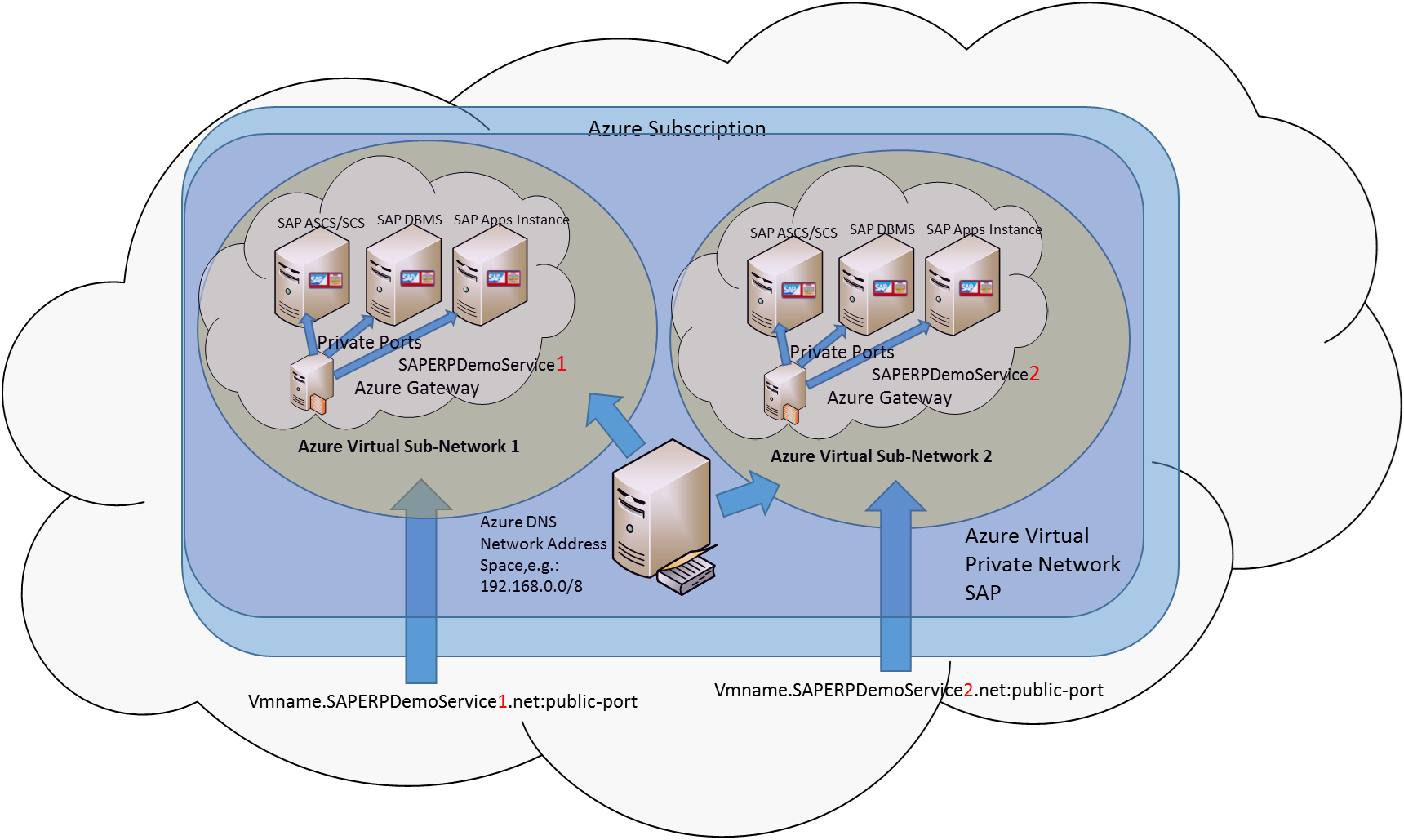


Figure Two SAP landscapes deployed within Subnetworks within the same Azure Virtual Network

If you do not want to have a virtual network for each server group, you can use sub-networks. Split the virtual network in as many segments as the number of SAP landscape you want to create. Then attach one cloud service to each sub-network.

In such a network the Azure domain name service will serve for each sub-network segment separately. Hence the hostnames of the virtual machines must be unique only within the sub-network. However cross-border inter-machine communication between the Sub-Networks can only be initiated by IP addresses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Virtual Network Name: VirtualNetwork | | | Address Space: 192.168.0.0/8 | |
| Sub-Network Name: Subnetwork-0 | | | Address Space: 192.168.1.00/29 | |
| Cloud Service DNS-Name: CloudService-0 | | | IP-Address: 168.63.30.64 | |
| *Hostname* | *IP-Address* | *Port No.* | *Gateway* | *DNS* |
| VirtualMachine-0 | 192.168.1.04 | 63000 | 192.168.1.01 | 168.63.129.16 |
| VirtualMachine-1 | 192.168.1.05 | 63001 | 192.168.1.01 | 168.63.129.16 |
| VirtualMachine-2 | 192.168.1.06 | 63002 | 192.168.1.01 | 168.63.129.16 |
| Sub-Network Name: Subnetwork-1 | | | Address Space: 192.168.1.08/29 | |
| Cloud Service DNS-Name: CloudService-1 | | | IP-Address: 168.63.30.41 | |
| *Hostname* | *IP-Address* | *Port No.* | *Gateway* | *DNS* |
| VirtualMachine-0 | 192.168.1.12 | 63000 | 192.168.1.09 | 168.63.129.16 |
| VirtualMachine-1 | 192.168.1.13 | 63001 | 192.168.1.09 | 168.63.129.16 |
| VirtualMachine-2 | 192.168.1.14 | 63002 | 192.168.1.09 | 168.63.129.16 |
| Sub-Network Name: Subnetwork-2 | | | Address Space: 192.168.1.16/29 | |
| Cloud Service DNS-Name: CloudService-2 | | | IP-Address: 168.63.30.158 | |
| *Hostname* | *IP-Address* | *Port No.* | *Gateway* | *DNS* |
| VirtualMachine-0 | 192.168.1.20 | 63000 | 192.168.1.17 | 168.63.129.16 |
| VirtualMachine-1 | 192.168.1.21 | 63001 | 192.168.1.17 | 168.63.129.16 |
| VirtualMachine-2 | 192.168.1.22 | 63002 | 192.168.1.17 | 168.63.129.16 |

### Gateway/Endpoint configuration

Inside the virtual network sub-segments, the virtual machines can communicate with the usual restrictions on networking. Communication to and from outside the virtual network have to be routed over the cloud services public IP address. For each service you want to connect to through the Internet, you have to configure an endpoint as described in [chapter 6.1](#_Remote_Access_SAP) of this document.

Achtung Be aware:

* Hostnames must be unique within a cloud service.
* All data disk names must be unique within your Azure subscription.
* All blob (VHD) names (URLs of the blobs) must be unique within your Azure.

# Deploying SAP VMs with Corporate Network Connectivity (Cross-Premises)

You run an SAP landscape and want to divide the deployment between bare-metal for high-end DBMS servers, on-premises virtualized environments for application layers and smaller 2-Tier configured SAP systems and Azure IaaS. The base assumption is that SAP systems within one SAP landscape need to communicate with each other and with many other software components deployed in the company, independent of their deployment form. There also should be no differences introduced by the deployment form for the end user connecting with SAP GUI or other interfaces. These conditions can only be met when we have the on-premises Active Directory and DNS services extended to the Azure systems through site-to-site/multi-site connectivity or private connections like Azure ExpressRoute.

In order to get more background on the implementation details of SAP on Azure, we encourage you to read [chapter 7](#_Concepts_of_Azure-Only) of this document which explains some of the basics constructs of Azure and how these should be used with SAP applications in Azure.

## Scenario of an SAP landscape

The Cross-Premises scenario can be roughly described like in the graphics below:

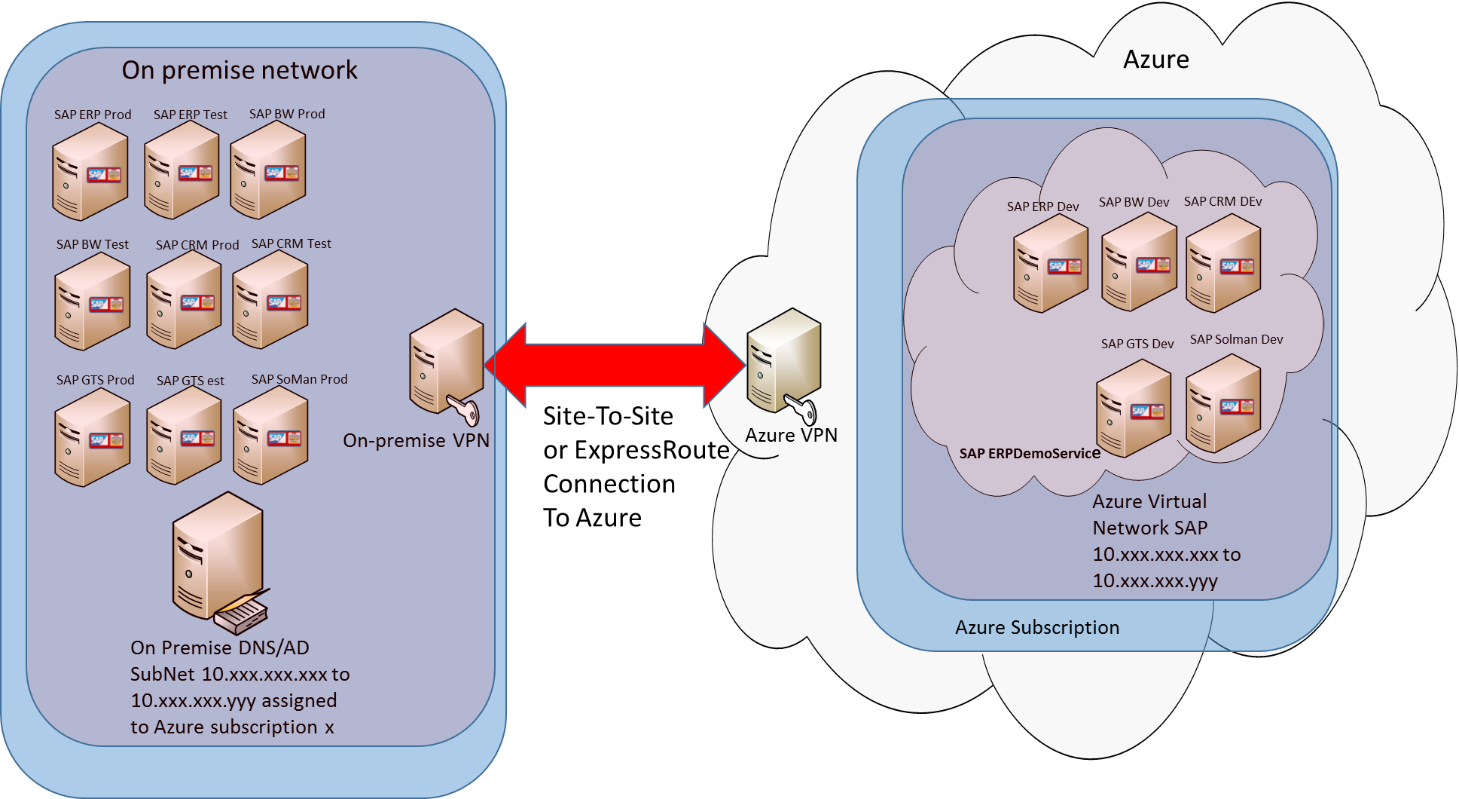


Figure Site-to-Site connectivity between on-premises and Azure assets

The scenario shown in Figure 21 describes a scenario where the on-premises AD and DNS is extended to Azure. On the on-premises side, a certain IP address range is reserved per Azure subscription. The IP address range will be assigned to an Azure Virtual Network on the Azure side. This means that an Azure Virtual Network is a requirement for this scenario.

### Security considerations

The minimum requirement is the use of secure communication protocols such as SSL/TLS for browser access or VPN-based connections for system access to the Azure services. The assumption is that companies handle the VPN connection between their corporate network and Azure very differently. Some companies might blankly open all the ports. Some other companies might want to be very precise in which ports they need to open, etc.

In the table below typical SAP communication ports are listed. Basically it is sufficient to open the SAP gateway port.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Port Name | Example <nn> = 01 | Default  Range  (min-max) | Comment |
| Dispatcher | sapdp<nn>1) | 3201 | 3200 – 3299 | SAP Dispatcher, used by SAP GUI for Windows and Java |
| Message server | sapms<sid>2) | 3600 | free sapms<anySID> | sid = SAP-System-ID |
| Gateway | Sapgw<nn>1) | 3301 | free | SAP gateway, used for CPIC and RFC communication |
| SAP router | Sapdp99 | 3299 | free | Only CI (central instance)  Service names can be reassigned in /etc/services to an arbitrary value after installation. |

1) nn = SAP Instance Number

2) sid = SAP-System-ID

More detailed information on ports required for different SAP products or services by SAP products can be found here <http://scn.sap.com/docs/DOC-17124>. With this document you should be able to open dedicated ports in the VPN device necessary for specific SAP products and scenarios.

Other security measures when deploying VMs in such a scenario could be to deploy VMs without a public endpoint for RDP or PowerShell. The default VM deployment will enable these two endpoints. Removing these endpoints can occur in the process of deploying in Azure Portal as well as with the PowerShell cmdlets which allow deployment of VMs without public endpoints. If necessary endpoints can be added/deleted to/from the VMs on demand. As mentioned in this [chapter 6.1](#_Remote_Access_SAP), endpoints also can be secured by ACLs.

## Dealing with different Virtual Machine Series

In the course of last 12 months Microsoft added many more VM types that differ either in number of vCPUs, memory or more important on hardware it is running on. Not all those VMs are supported with SAP (see supported VM types in SAP Note[1928533 – SAP Applications on Azure: Supported Products and Sizing](http://service.sap.com/sap/support/notes/1928533) ). Some of those VMs run on different host hardware generations. These host hardware generations are getting deployed in the granularity of an Azure Scale-Unit. Means cases may arise where the different VM sizes you chose can’t be run on the same Scale-Unit. A Virtual Machine Cloud Service is limited in the ability to span Scale-Units based of different hardware. E.g. if you want to run the DBMS on A8-A11 VMs and the SAP application layer on D-Series VMs, you would be forced to deploy a single SAP system or different SAP systems within different Azure Cloud Services. In our examples one cloud Service for the DBMS server and one Cloud Service for the SAP application layer.

### Printing on a local network printer from SAP instance in Azure

#### Printing over TCP/IP in Cross-Premises scenario

Setting up your on-premises TCP/IP based network printers in an Azure VM is overall the same as in your corporate network, assuming you do have a VPN Site-To-Site tunnel or ExpressRoute connection established.

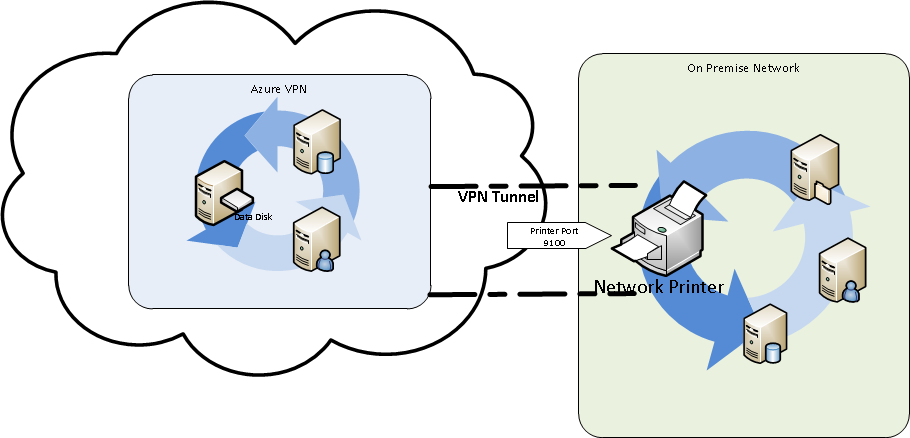


Figure Network printing

How to:

* Some network printers come with a configuration wizard which makes it easy to set up your printer in an Azure VM. If no wizard software has been distributed with the printer the “manual” way to set up the printer is to create a new TCP/IP printer port.
* Open Control Panel -> Devices and Printers -> Add a printer
* Choose Add a printer using a TCP/IP address or hostname
* Type in the IP address of the printer
* Printer Port standard 9100
* If necessary install the appropriate printer driver manually.

#### Host-based printer over SMB (shared printer) in Cross-Premises scenario

Host-based printers are not network-compatible by design. But a host-based printer can be shared among computers on a network as long as the printer is connected to a powered-on computer. Connect your corporate network either Site-To-Site or ExpressRoute and share your local printer. The SMB protocol uses NetBIOS instead of DNS as name service. The NetBIOS host name can be different from the DNS host name. The standard case is that the NetBIOS host name and the DNS host name are identical. The DNS domain does not make sense in the NetBIOS name space. Accordingly, the fully qualified DNS host name consisting of the DNS host name and DNS domain must not be used in the NetBIOS name space.

The printer share is identified by a unique name in the network:

* Host name of the SMB host (always needed).
* Name of the share (always needed).
* Name of the domain if printer share is not in the same domain as SAP system.
* Additionally, a user name and a password may be required to access the printer share.

How to:

* Share your local printer.
* In the Azure VM open the Windows Explorer and type in the share name of the printer.
* A printer installation wizard will guide you through the installation process.

#### USB Printer (printer forwarding)

In Azure the ability of the Remote Desktop Services to provide users the access to their local printer devices in a remote session is not available.

More details on printing with Windows can be found here: <http://technet.microsoft.com/en-us/library/jj590748.aspx> .

### Integration of SAP Azure Systems into Correction and Transport System (TMS) in Cross-Premises

The SAP Change and Transport System (TMS) needs to be configured to export and import transport request across systems in the landscape. We assume that the development instances of an SAP system (*DEV*) are located in Azure whereas the quality assurance *(QA)* and productive systems *(PRD)* are on-premises. Furthermore, we assume that there is a central transport directory.

#### Configuring the Transport Domain

Configure your Transport Domain on the system you designated as the Transport Domain Controller as described in [Configuring the Transport Domain Controller](http://help.sap.com/erp2005_ehp_04/helpdata/en/44/b4a0b47acc11d1899e0000e829fbbd/content.htm). A system user *TMSADM* will be created and the required RFC destination will be generated. You may check these RFC connections using the transaction *SM59*. Hostname resolution must be enabled across your transport domain.

How to:

* In our scenario we decided the on-premises *QAS* system will be the CTS domain controller. Call transaction *STMS*. The *TMS* dialog box appears. A *Configure Transport Domain* dialog box is displayed. (This dialog box only appears if you have **not yet** configured a transport domain.)
* Make sure that the automatically created user *TMSADM* is authorized (*SM59* -> *ABAP Connection* -> *TMSADM@E61.DOMAIN\_E61* -> *Details* -> *Utilities(M)* -> *Authorization Test*). The initial screen of transaction *STMS* should show that this SAP System is now functioning as the controller of the transport domain as shown here:

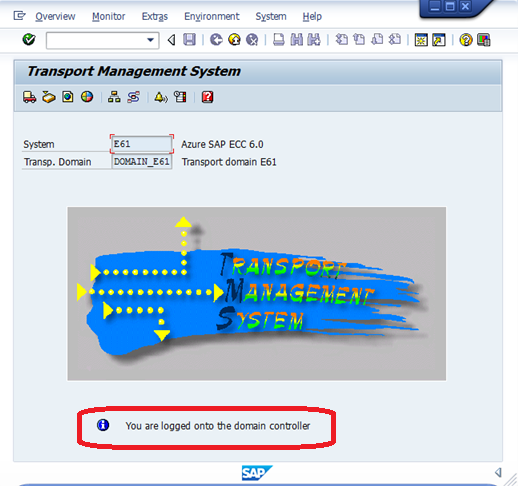


Figure Initial screen of transaction STMS on the domain controller

### [Including SAP Systems in the Transport Domain](http://help.sap.com/saphelp_nw70ehp3/helpdata/en/44/b4a0c17acc11d1899e0000e829fbbd/content.htm?frameset=/en/44/b4a0b47acc11d1899e0000e829fbbd/frameset.htm)

The sequence of including an SAP system in a transport domain looks as follows:

* On the *DEV* system in Azure go to the transport system (Client 000) and call transaction *STMS*. Choose *Other Configuration* from the dialog box and continue with *Include System in Domain*. Specify the Domain Controller as target host ([Including SAP Systems in the Transport Domain](http://help.sap.com/erp2005_ehp_04/helpdata/en/44/b4a0c17acc11d1899e0000e829fbbd/content.htm?frameset=/en/44/b4a0b47acc11d1899e0000e829fbbd/frameset.htm)). The system is now waiting to be included in the transport domain.
* For security reasons, you then have to go back to the domain controller to confirm your request. Choose *System Overview* and *Approve* of the waiting system. Then confirm the prompt and the configuration will be distributed.

This SAP system now contains the necessary information about all the other SAP systems in the transport domain. At the same time, the address data of the new SAP system is sent to all the other SAP systems, and the SAP system is entered in the transport profile of the transport control program. Check whether RFCs and access to the transport directory of the domain work.

Continue with the configuration of your transport system as usual as described in the documentation [Change and Transport System](http://help.sap.com/saphelp_nw70ehp3/helpdata/en/48/c4300fca5d581ce10000000a42189c/content.htm?frameset=/en/44/b4a0b47acc11d1899e0000e829fbbd/frameset.htm).

How to:

* Make sure your S*TMS* on premises is configured correctly.
* Make sure the hostname of the Transport Domain Controller can be resolved by your virtual machine on Azure and vice visa.
* Call transaction *STMS* -> *Other Configuration* -> *Include System in Domain.*
* Confirm the connection in the on premises *TMS* system.
* Configure transport routes, groups and layers as usual.

In site-to-site connected Cross-Premises scenarios, the latency between on-premises and Azure still can be substantial. If we follow the sequence of transporting objects through development and test systems to production or think about applying transports or support packages to the different systems, you realize that, dependent on the location of the central transport directory, some of the systems will encounter high latency reading or writing data in the central transport directory. The situation is similar to SAP landscape configurations where the different systems are spread through different data centers with substantial distance between the data centers.

In order to work around such latency and have the systems work fast in reading or writing to or from the transport directory, you can setup two STMS transport domains (one for on-premises and one with the systems in Azure and link the transport domains. Please check this documentation which explains the principles behind this concept in the SAP TMS:  
<http://help.sap.com/saphelp_me60/helpdata/en/c4/6045377b52253de10000009b38f889/content.htm?frameset=/en/57/38dd924eb711d182bf0000e829fbfe/frameset.htm>.

How to:

* Set up a transport domain on each location (on-premises and Azure) using transaction STMS <http://help.sap.com/saphelp_nw70ehp3/helpdata/en/44/b4a0b47acc11d1899e0000e829fbbd/content.htm>
* Link the domains with a domain link and confirm the link between the two domains. <http://help.sap.com/saphelp_em70/helpdata/en/14/c795388d62e450e10000009b38f889/content.htm>
* Distribute the configuration to the linked system.

### RFC traffic between SAP instances located in Azure and on-premises (Cross-Premises)

RFC traffic between systems which are on-premises and in Azure needs to work. To setup a connection call transaction *SM59* in a source system where you need to define an RFC connection towards the target system. The configuration is similar to the standard setup of an RFC Connection.

We assume that in the Cross-Premises scenario, the VMs which run SAP systems that need to communicate with each other are in the same Windows domain. Therefore the setup of an RFC connection between SAP systems does not differ from the setup steps and inputs in on-premises scenarios.

### Accessing ‘local’ fileshares from SAP instances located in Azure or vice versa

SAP instances located in Azure need to access file shares which are within the corporate premises. In addition, on-premises SAP instances need to access file shares which are located in Azure. To enable the file shares you must configure the permissions and sharing options on the local system. Moreover, you have to open port 445 (TCP and UPD) to enable direct-hosted SMB traffic.

# Supportability

## Azure Monitoring Solution for SAP

In order to enable the monitoring of mission critical SAP systems on Azure the SAP monitoring tools SAPOSCOL or SAP Host Agent get data off the Azure Virtual Machine Service host via an Azure Monitoring Extension for SAP. Since the demands by SAP were very specific to SAP applications, Microsoft decided not to generically implement the required functionality into Azure, but leave it for customers to deploy the necessary monitoring components and configurations to their Virtual Machines running in Azure. However, deployment and lifecycle management of the monitoring components will be mostly automated by Azure.

### Solution design

The solution developed to enable SAP Monitoring is based on the architecture of Azure VM Agent and Extension framework. The idea of the Azure VM Agent and Extension framework is to allow installation of software application(s) available in the Azure VM Extension gallery within a VM. The principle idea behind this concept is to allow (in cases like the Azure Monitoring Extension for SAP), the deployment of special functionality into a VM and the configuration of such software at deployment time.

Since February 2014, the ‘Azure VM Agent’ that enables handling of specific Azure VM Extensions within the VM is injected into VMs by default on VM creation in the Azure Portal (<http://blogs.msdn.com/b/wats/archive/2014/02/17/bginfo-guest-agent-extension-for-azure-vms.aspx> )

If VMs are deployed through the Azure Portal, REST API or PowerShell, there is an option available to inject the Azure VM Agent. For already deployed VMs, Microsoft provides a download location for this Azure VM Agent, so that it can be installed and enabled for VMs that are deployed and running already.

The basic building blocks of the Monitoring solution in Azure for SAP looks like this:

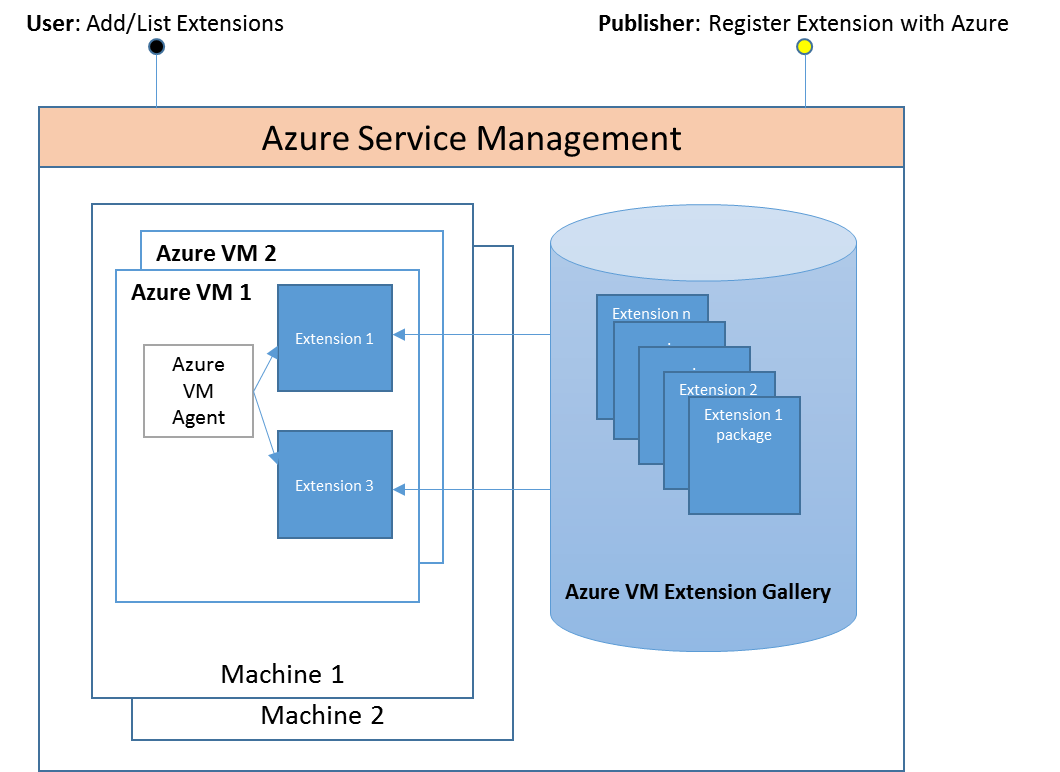


Figure Microsoft Azure Extension components

As shown in the block diagram above, one part of the monitoring solution for SAP is hosted in the Azure VM Image and Azure Extension Gallery which is a globally replicated repository that is managed by Azure Operations. It is the responsibility of the joint SAP/MS team working on the Azure implementation of SAP to work with Azure Operations to publish new versions of the Azure Monitoring Extension for SAP. This Azure Monitoring Extension for SAP will use the Windows Azure Diagnostics (WAD) Extension to get the necessary information.

When you deploy a new VM, the ‘Azure VM Agent’ is automatically added into the VM. The function of this agent is to coordinate the loading and configuration of the Azure Extensions for monitoring of SAP NetWeaver Systems.

However, there is a step that still needs to be executed by the customer. This is the enablement and configuration of the performance collection. The process related to the ‘configuration’ is automated by a PowerShell script. The PowerShell script can be downloaded in the Microsoft Azure Script Center.

The overall Architecture of the Azure monitoring solution for SAP looks like:

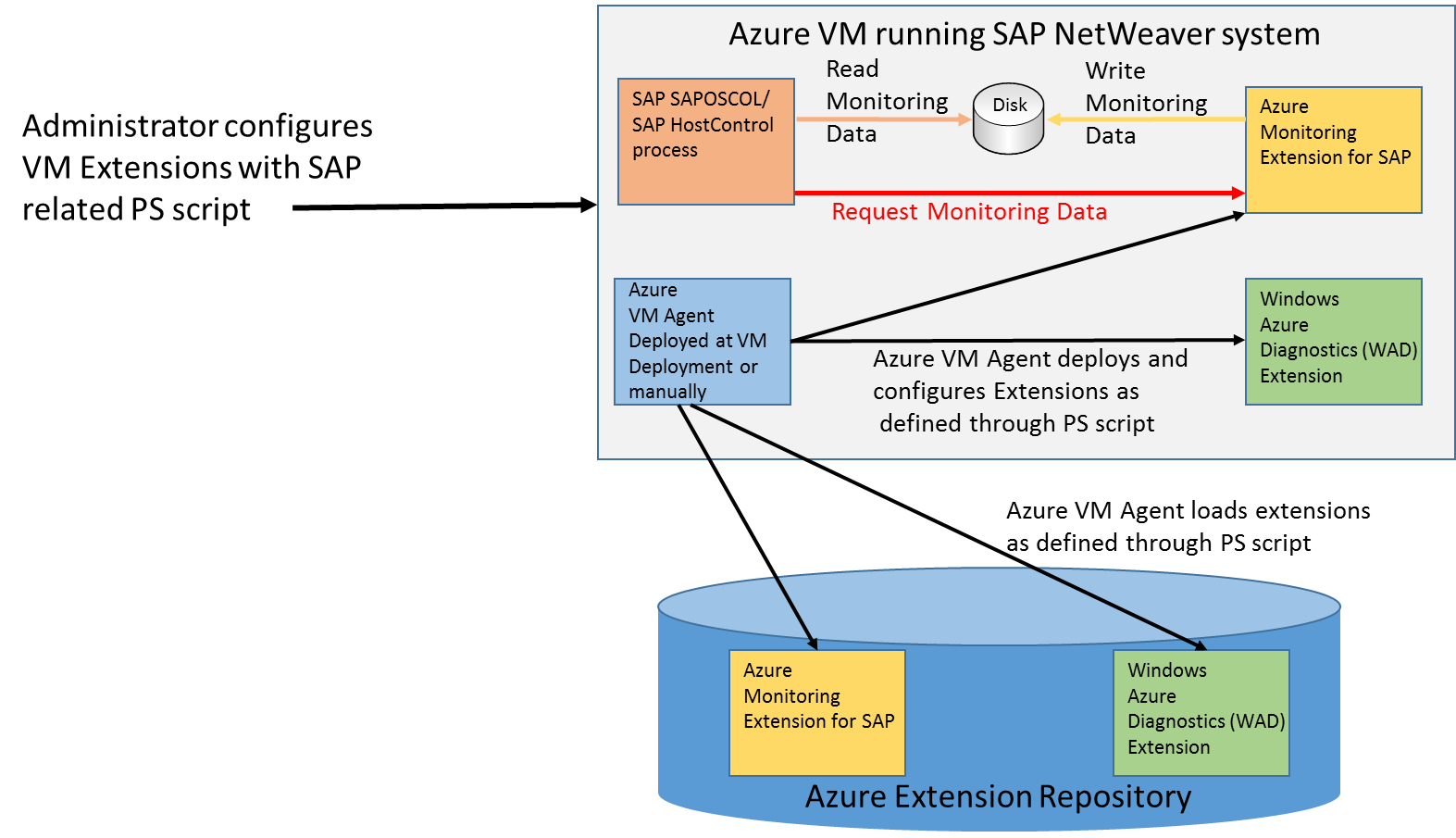


Figure Azure monitoring solution for SAP NetWeaver

**For the exact how-to and for detailed steps of using these PowerShell cmdlets during deployments, follow the instructions given in the ‘*SAP NetWeaver on Microsoft Azure Virtual Machine Services - Deployment Guide*’.**

## Integration of Azure located SAP instance into SAProuter

SAP instances running in Azure need to be accessible from SAProuter as well.

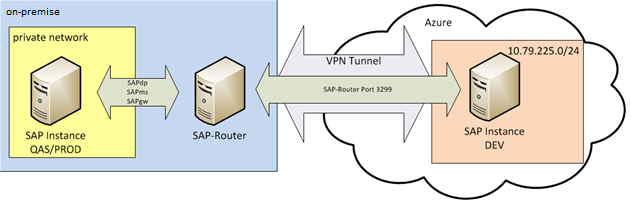


Figure SAP-Router Network Connection

A SAProuter enables the TCP/IP communication between participating systems if there is no direct IP connection. This provides the advantage that no end-to-end connection between the communication partners is necessary on network level. The SAProuter is listening on port 3299 by default.

To connect SAP instances through a SAProuter you need to give the SAProuter string and host name with any attempt to connect.

# SAP NetWeaver AS Java

So far the focus of the document has been SAP NetWeaver in general or the SAP NetWeaver ABAP stack. In this small section, specific considerations for the SAP Java stack are listed. One of the most important SAP NetWeaver Java exclusively based applications is the SAP Enterprise Portal. Other SAP NetWeaver based applications like SAP PI and SAP Solution Manager use both the SAP NetWeaver ABAP and Java stacks. Therefore, there certainly is a need to consider specific aspects related to the SAP NetWeaver Java stack as well.

## SAP Enterprise Portal

The setup of an SAP Portal in an Azure Virtual Machine does not differ from an on premises installation if you are deploying in Cross-Premises scenarios. Since the DNS is done by on-premises, the port settings of the individual instances can be done as configured on-premises. The recommendations and restrictions described in this document so far apply for an application like SAP Enterprise Portal or the SAP NetWeaver Java stack in general.

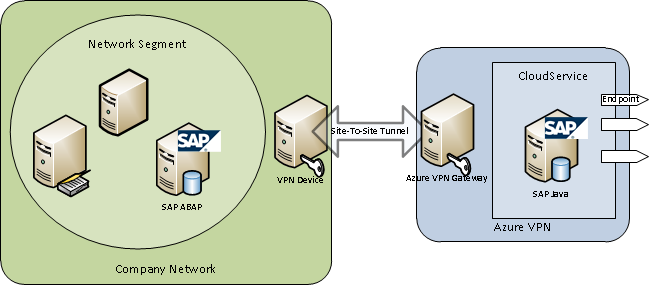
A special deployment scenario by some customers is the direct exposure of the SAP Enterprise Portal to the Internet while the virtual machine host is connected to the company network via site-to-site VPN tunnel or ExpressRoute. For such a scenario, you need to work again with Public Endpoints assigned to the Virtual Machine host. The same mechanics would need to be applied when you want to connect to an SAP Java instance from on-premises in a Cloud-Only scenario.

Figure 27:

Figure Exposed SAP Portal

The initial portal URI is http(s):**<Portalserver>:5XX00/irj** where the port is formed by **50000 plus (Systemnumber × 100)**. The default portal URI of SAP system 00 is **CloudService.Cloudapp.net:Endpoint-PublicPort/irj**. For more details, have a look a

<http://help.sap.com/saphelp_nw70ehp1/helpdata/de/a2/f9d7fed2adc340ab462ae159d19509/frameset.htm>.

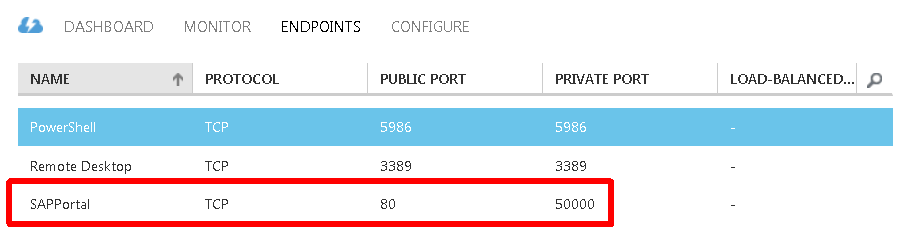


Figure Endpoint configuration

Make sure that the appropriate endpoints have been added to the Azure Virtual Machine. In Figure 28, the internal port of the endpoint was set to 50000 since the system number of the Java instance is 00. As public port the default port of 80 was set. Please note that the public ports of different VMs running SAP NetWeaver Java instances need to differ. As a result, endpoints of additional VMs within the same Azure Cloud Service would not be able to have the public port of 80 assigned as public port. Hence if running multiple SAP systems with Java instances within one Azure Cloud Service you would need to introduce some rules for the enumeration space of the public ports. The Azure cloud service DNS name and the public port of the assigned endpoint cover the portal URI.

If you want to customize the URL and/or ports of your SAP Enterprise Portal, please check this documentation:

* (<http://wiki.scn.sap.com/wiki/display/EP/Change+Portal+URL>)
* (<http://wiki.scn.sap.com/wiki/display/NWTech/Change+Default++port+numbers%2C+Portal+port+numbers>)

# High Availability (HA) and Disaster Recovery (DR) for SAP NetWeaver running on Azure Virtual Machines

## Overview

The Term ***high availability*** (***HA***) is generally related to a set of technologies that minimizes IT disruptions by providing business continuity of IT services through redundant, fault-tolerant or failover protected components inside of the **same** data center. In our case, inside of the same Azure Region.

***Disaster recovery*** (***DR***) is also targeting minimizing IT services disruption, and their recovery but across **different** data centers, that are usually located hundreds of kilometers away. In our case usually between different Azure Regions within the same geopolitical region or as established by you as a customer.

## Overview of High Availability

We can separate SAP high availability in Azure discussion into two parts:

* **Azure infrastructure high availability**, e.g. HA of compute (VMs), network, storage, cloud services etc. and its benefits for increasing SAP application availability, e.g. achieving SAP application “higher” availability.
* **SAP application high availability**, e.g. HA of SAP software components:
  + SAP application servers
  + SAP ASCS/SCS instance
  + DB server

and how it can be combined with Azure infrastructure HA.

SAP High Availability in Azure has some differences compared to SAP High Availability in on-premises physical or virtual environment. This paper from SAP describes standard SAP High Availability configurations in virtualized environments on Windows: <http://scn.sap.com/docs/DOC-44415>.

## Azure Infrastructure High Availability

There is no single-VM SLA available on Azure Virtual Machines right now. To get an idea how the availability of a single VM might look like you can simply build the product of the different available Azure SLAs: <http://www.windowsazure.com/en-us/support/legal/sla/>.

The basis for the calculation is 30 days per month, or 43200 minutes. Therefore, 0.05% downtime corresponds to 21.6 minutes. As usual, the availability of the different services will multiply in the following way:

(Availability Service #1/100) \* (Availability Service #2/100) \* (Availability Service #3/100) \*…

Like:

(99.95/100) \* (99.9/100) \* (99.9/100) = 0.9975 or an overall availability of 99.75%.

### Virtual Machine (VM) High Availability

There are two types of Azure platform events that can affect the availability of your virtual machines: planned maintenance and unplanned maintenance.

* **Planned maintenance** events are periodic updates made by Microsoft to the underlying Azure platform to improve overall reliability, performance, and security of the platform infrastructure that your virtual machines run on.
* **Unplanned maintenance** events occur when the hardware or physical infrastructure underlying your virtual machine has faulted in some way. This may include local network failures, local disk failures, or other rack level failures. When such a failure is detected, the Azure platform will automatically migrate your virtual machine from the unhealthy physical machine hosting your virtual machine to a healthy physical machine. Such events are rare, but may also cause your virtual machine to reboot.

More details can be found in this documentation:

<http://azure.microsoft.com/en-us/documentation/articles/virtual-machines-manage-availability>

### Azure Storage Redundancy

The data in your Microsoft Azure Storage Account is always replicated to ensure durability and high availability, meeting the Azure Storage SLA even in the face of transient hardware failures

Since Azure Storage is keeping 3 images of the data, RAID5 or RAID1 are not necessary.

More details can be found in this article:   
<http://azure.microsoft.com/en-us/documentation/articles/storage-redundancy/>

### Utilizing Azure Infrastructure VM Restart to Achieve “Higher Availability” of SAP Applications

If you decide not to use functionalities like Windows Server Failover Clustering (WSFC), you can utilize **Azure VM Restart** to protect an SAP System against planned and unplanned Azure physical server infrastructure and overall underlying Azure platform.

Achtung IMPORTANT

It is important to mention that Azure VM Restart primarily **protects VMs** and **NOT** **applications**. VM Restart does not offer high availability for SAP applications, but it does offer a certain level of **infrastructure availability** and therefore indirectly “**higher availability**” of SAP systems. There is also no SLA for the time it will take to restart a VM after a planned or unplanned host outage. Therefore, this method of ‘high availability’ is not suitable for critical components of a SAP system like (A)SCS or DBMS.

Another important infrastructure element for high availability is storage. E.g. Azure Storage SLA is 99,9 % availability. If we deploy all VMs with its disks into a single Azure Storage Account, potential Azure Storage unavailability will cause unavailability of all VMs that are placed in that Azure Storage Account, and also all SAP components running in side of those VMs.

Instead of putting all VMs into one single Azure Storage Account, you can also use dedicated storage accounts for each VM, and in this way increase overall VM and SAP application availability by using multiple independent Azure Storage Accounts.

An example of architecture of an SAP NetWeaver system that uses Azure infrastructure HA could look like this:



Figure Utilizing Azure infrastructure HA to achieve SAP application “higher” availability

For critical SAP components we achieved the following:

* **High Availability of SAP Application Servers (AS)**

SAP application server instances are redundant components. Each SAP AS instance is deployed on its own VM, that is running in a different Azure Fault and Upgrade Domain (see chapters [3.2.1](#_Fault_Domains) and [3.2.2](#_Upgrade_Domains)). This is ensured by using Azure Availability Sets (see [chapter 3.2.3](#_Azure_Availability_Sets)). Potential planned or unplanned unavailability of an Azure Fault or Upgrade Domain will cause unavailability of a restricted number of VMs with their SAP AS instances.

Each SAP AS instance is placed in its own Azure Storage account – potential unavailability of one Azure Storage Account will cause unavailability of only one VM with its SAP AS instance. However, be aware that there is a limit of Azure Storage Accounts within on Azure subscription. To ensure automatic start of (A)SCS instance after the VM reboot, make sure to set Autostart parameter in (A)SCS instance start profile described in [chapter 11.5](#_Using_g_Aautostart).

Please also read [chapter 11.4.1 High Availability for SAP Application Server Instances](#_High_Availability_for_1) for more details.

* **“Higher” Availability of SAP (A)SCS instance**

Here we utilize Azure VM Restart to protect the VM with installed SAP (A)SCS instance. In the case of planned or unplanned downtime of Azure severs, VMs will be restarted on another available server. As mentioned earlier, Azure VM Restart primarily **protects VMs** and **NOT** **applications**, in this case the (A)SCS instance. Through the VM Restart we’ll reach indirectly “higher availability” of SAP (A)SCS instance. To insure automatic start of (A)SCS instance after the VM reboot, make sure to set Autostart parameter in (A)SCS instance start profile described in [chapter 11.5](#_Using_g_Aautostart). This means the (A)SCS instance as a Single Point of Failure (SPOF) running in a single VM will be the determinative factor for the availability of the whole SAP landscape.

* **“Higher” Availability of DBMS Server**

Here, similar to SAP (A)SCS instance use case, we utilize Azure VM Restart to protect the VM with installed DBMS software, and we achieve “higher availability” of DBMS software through VM Restart.

DBMS running in a single VM is also SPOF, and it is the determinative factor for the availability of the whole SAP landscape.

## SAP Application High Availability on Azure IaaS

To achieve full SAP system high availability, we need to protect all critical SAP system components, e.g. redundant SAP application servers, and unique components (e.g. Single Point of Failure) like SAP (A)SCS instance and DBMS.

### High Availability for SAP Application Servers

For the SAP application servers/dialog instances it’s not necessary to think about a specific high availability solution. High availability is simply achieved by redundancy and thereby having enough of them in different virtual machines. They should all be placed in the same **Azure Availability Set** to avoid that the VMs might be updated at the same time during planned maintenance downtime. The basic functionality which builds on different Upgrade and Fault Domains within an Azure Scale Unit was already introduced in [chapter 3.1.2](#_Upgrade_Domains). Azure Availability Sets were presented in [chapter 3.1.3](#_Azure_Availability_Sets) of this document.

There is no infinite number of Fault and Upgrade Domains that can be used by an Azure Availability Set within an Azure Scale Unit. This means that putting a number of VMs into one Availability Set sooner or later in the fact that more than one VM ends up in the same Fault or Upgrade Domain

Deploying a number of SAP application server instances in their dedicated VMs and assuming that we got 5 Upgrade Domains, the following picture emerges at the end:

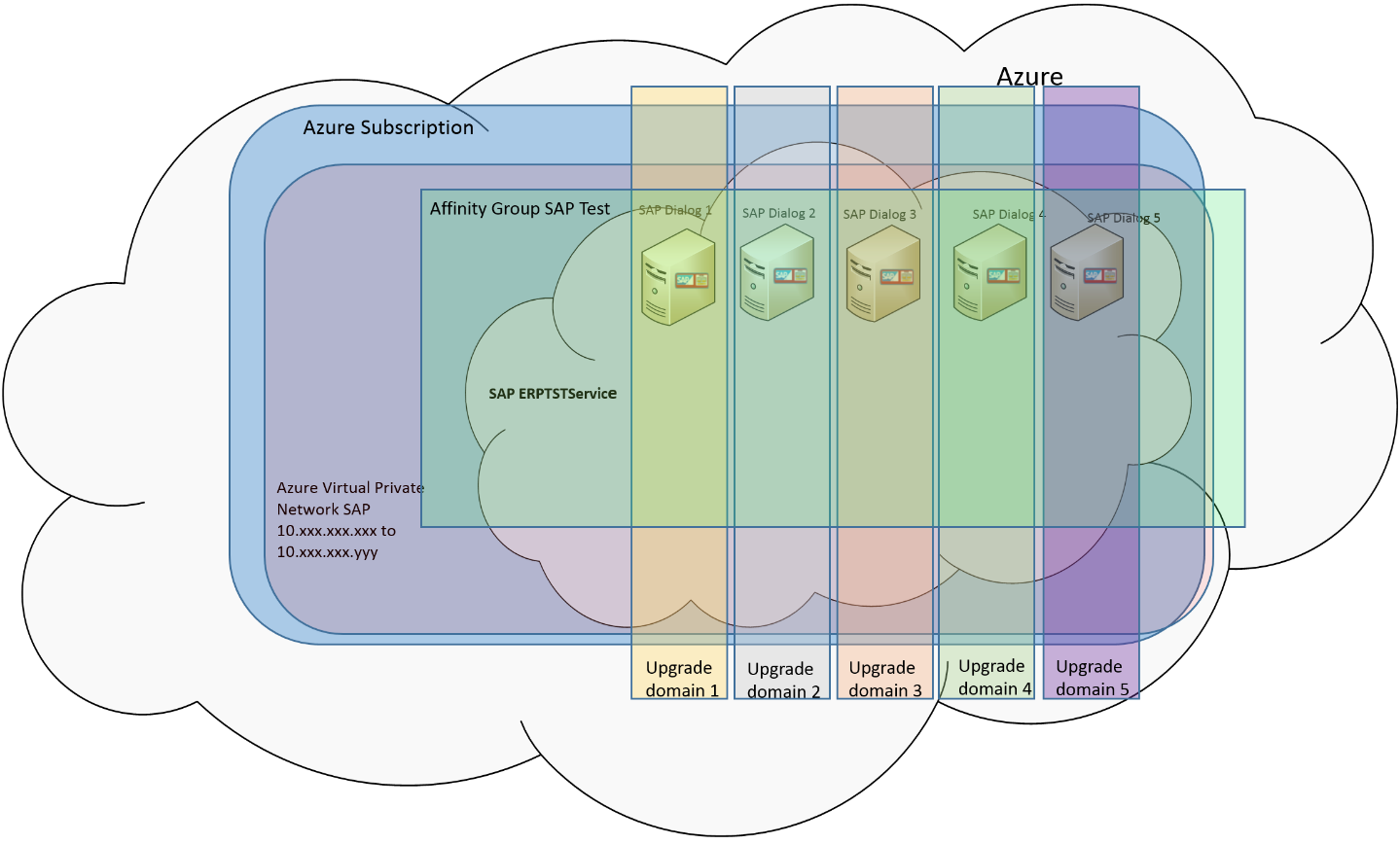


Figure HA of SAP Application Servers in Azure

In this case, the SAP system deployed has five VMs in the application layer and as in our example got deployed over 5 Upgrade Domains each VM end in its own Upgrade Domain. In the case where a sixth VM gets deployed in the same Availability Set, there will be one Upgrade Domain that contains a second VM of the SAP application layer.

### High Availability for the SAP (A)SCS instance

Windows Server Failover Cluster (WSFC) is a frequently used solution to protect the SAP (A)SCS instance.

At this point in time Azure on itself would not be able to provide the functionality to set up the required Windows Server Failover Cluster in a native manner.

As of July 2015 the Azure cloud platform running the Windows operating system does not provide the possibility of using:

* A cluster shared volume
* Multiple virtual IP addresses

**WSFC cluster shared volume in Windows Azure**

However, you can create a Windows Server Failover Cluster shared volume with a 3rd party software SIOS DataKeeper Cluster Edition, which can be used to install and cluster SAP ASCS instance on the Azure IaaS platform.

The SIOS DataKeeper solution provides a shared disk cluster resource to Windows Failover Clusters by having:

* An additional Azure VHD attached to each of the virtual machines (VMs) that are in a Windows Cluster configuration
* SIOS DataKeeper Cluster Edition running on both VM nodes
* Having SIOS DataKeeper Cluster Edition configured in a way that it synchronously mirrors the content of the additional VHD attached volume from source VMs to additional VHD attached volume of target VM.
* SIOS DataKeeper is abstracting the source and target local volumes and presenting them to Windows Failover Cluster as a single shared disk.

**WSFC virtual IP addresses in Windows Azure**

In an on-premises solution the DNS-Server routes the traffic with a virtual IP-address to the active Cluster node. On Azure we use the Azure Internal Load Balancer (ILB) to route the traffic to the cluster node where SAP ASCS instance is running.



Figure Schema of a Windows Server Failover Cluster configuration in Azure using SIOS DataKeeper

You can find more details on how to install a Windows Failover Cluster with SIOS Datakeeper and SAP in the *‘SAP NetWeaver on Azure - Clustering SAP ASCS Instance using Windows Server Failover Cluster on Azure with SIOS DataKeeper’* white paper.

### High Availability for the SAP database instance

The SAP system also is supported by two DBMS VMs where we use DBMS high-availability functionality to replicate data from the active DBMS instance to the second VM into a passive DBMS instance.  
  
High Availability and Disaster recovery functionality for DBMS in general as well as specific DBMS are described in the ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’.

### End-to-End High Availability for the Complete SAP System

Here is an example of a complete SAP NetWeaver HA architecture in Azure.

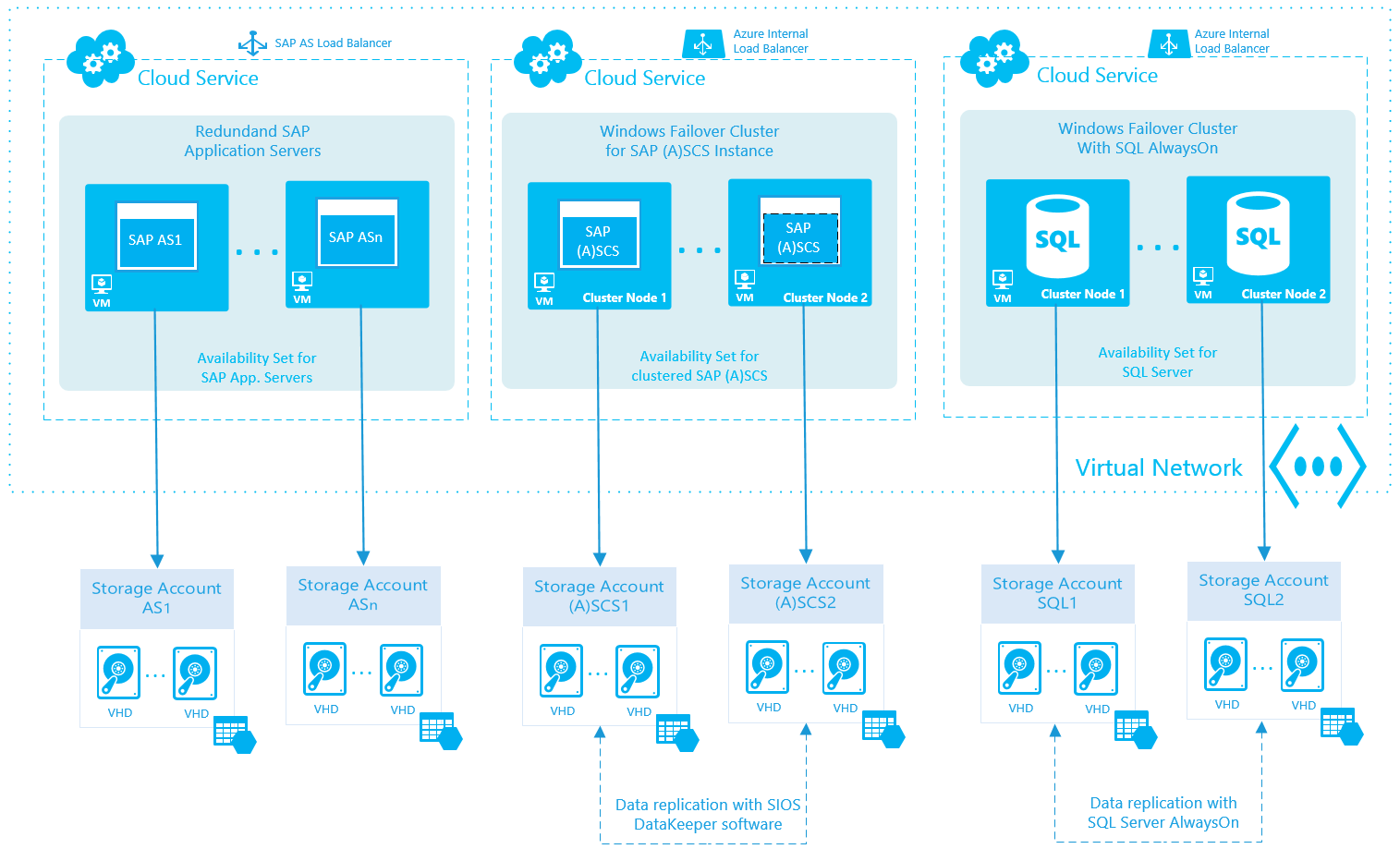


Figure SAP NetWeaver Application HA Architecture with SQL Server in Azure IaaS

The following Azure constructs are used for the SAP NetWeaver system, to minimize impact by infrastructure issues and host patching:

* The complete system is deployed on Azure (required - DBMS layer, (A)SCS instance and complete application layer need to run in the same location).
* The complete system runs within one Azure subscription (required).
* The complete system runs within one Azure Virtual Network (required).
* The separation of the VMs of one SAP system into three Availability Sets is possible even with all the VMs belonging to the same Virtual Network.
* All VMs running DBMS instances of one SAP system are in one Availability Set and one Cloud Service. We assume that there is more than one VM running DBMS instances per system since native DBMS high availability features are used, like SQL Server AlwaysOn or Oracle Data Guard.
* All VMs running DBMS instances use own storage account. DBMS data and log files are replicated from one storage account to another storage account using DBMS high availability functions that synchronize the data. Unavailability of one storage account will cause unavailability of one SQL Windows cluster node, but not the whole SQL Server service.
* All VMs running (A)SCS instance of one SAP system are in one Availability Set and own Cloud Service. Inside of those VMs is configure Windows Sever Failover Cluster (WSFC) to protect (A)SCS instance.
* All VMs running (A)SCS instances use own storage account. (A)SCS instance files and SAP global folder are replicated from one storage account to another storage account using SIOS DataKeeper replication. Unavailability of one storage account will cause unavailability of one (A)SCS Windows cluster node, but not the whole (A)SCS service.
* ALL the VMs representing the SAP application server layer are in third Availability Set and own Cloud Service.
* ALL the VMs running SAP application servers use own storage account. Unavailability of one storage account will cause unavailability of one SAP application server, where other SAP AS continue to run.

The concepts as explained above may need to be compromised a bit when you deploy many SAP systems and the number of VMs deployed are exceeding the maximum limit of Storage Accounts per subscription. In such cases, VHDs of VMs need to be combined within one Storage Account. Usually you would do so by combining VHDs of SAP application layer VMs of different SAP systems. We also combined different VHDs of different DBMS VMs of different SAP systems in one Azure Storage Account. Thereby keeping the IOPS limits of Azure Storage Accounts in mind (<https://azure.microsoft.com/en-us/documentation/articles/storage-scalability-targets> )

## Using Autostart for SAP instances

SAP offered the functionality to start SAP instances immediately after the start of the OS within the VM. The exact steps were documented in [SAP Knowledge Base Article 1909114 - How to start SAP instances automatically using parameter Autostart](http://service.sap.com/sap/support/notes/1909114). However, SAP is not recommending to use the setting anymore because there is no control in the order of instance restarts, assuming more than one VM got affected or multiple instances ran per VM. Assuming a typical Azure scenario of one SAP application server instance in a VM and the case of a single VM eventually getting restarted, the Autostart is not really critical and can be enabled by adding this parameter:

* Autostart = 1

Into the start profile of the SAP ABAP and/or Java instance.

**Note:** The Autostart parameter can have some downfalls as well. In more detail, the parameter triggers the start of a SAP ABAP or Java instance when the related Windows service of the instance is started. That certainly is the case when the operating systems boots up. However, restarts of SAP services are also a common thing for SAP Software Lifecycle Management functionality like SUM or other updates or upgrades. These functionalities are not expecting an instance to be restarted automatically at all. Therefore, the Autostart parameter should be disabled before running such tasks. The Autostart parameter also should not be used for SAP instances that are clustered, like ASCS/SCS/CI.

## Larger 3-Tier SAP systems

High-Availability aspects of 3-Tier SAP configurations got discussed in earlier sections already. But what about systems where the DBMS server requirements are too large to have it located in Azure, but the SAP application layer could be deployed into Azure?

### Location of 3-Tier SAP configurations

It is not supported to split the application tier itself or the application and DBMS tier between on-premises and Azure. An SAP system is either completely deployed on-premises or in Azure. It is also not supported to have some of the application servers run on-premises and some others in Azure. That is the starting point of the discussion. We also are not supporting to have the DBMS components of a SAP system and the SAP application server layer deployed in two different Azure Regions. E.g. DBMS in West US and SAP application layer in Central US. Reason for not supporting such configurations is the latency sensitivity of the SAP NetWeaver architecture.

However, over the course of last year datacenter partners developed co-locations to Azure Regions. These co-locations often are in very close proximity to the physical Azure datacenters within an Azure Region. The short distance and connection of assets in the co-location through ExpressRoute into Azure can result in a latency that is less than 2ms. In such cases, to locate the DBMS layer (including storage SAN/NAS) in such a co-location and the SAP application layer in Azure can be possible. As of July 2015, we don’t have any deployments like that. But different customers with non-SAP application deployments are using such approaches already.

## Offline Backup of SAP systems

Dependent on the SAP configuration chosen (2-Tier or 3-Tier) there could be a need to backup. The content of the VM itself plus to have a backup of the database. The DBMS related backups are expected to be done with database methods. A detailed description for the different databases, can be found in the document: ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’. On the other hand, the SAP data can be backed up in an offline manner (including the database content as well) as described in this section or online as described in the next section.

The offline backup would basically require a shutdown of the VM through the Azure Portal and a copy of the base VM disk plus all attached VHDs to the VM. This would preserve a point in time image of the VM and its associated disk. It is recommended to copy the ‘backups’ into a different Azure Storage Account. Hence the procedure described in [chapter 5.4.2](#_Copying_disks_between) of this document would apply.

A restore of that state would consist of deleting the base VM as well as the original disks of the base VM and mounted VHDs, copying back the saved VHDs to the original Storage Account and then redeploying the system.

Another possibility would be to leverage the so called BLOB snapshot API through PowerShell as it is described here: <http://blog.greatrexpectations.com/2013/04/24/using-blob-snapshots-to-backup-azure-virtual-machines/> Please note that the snapshot which is described on this page for the base VM blob/VHD only would need to be executed for each VHD as those are mounted to the VM while the VM is shutdown.

## Online backup of an SAP system

Backup of the DBMS is performed with DBMS specific methods as described in the ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’.

Other VMs within the SAP system can be backed up using Azure Virtual Machine Backup functionality. Azure Virtual Machine Backup got introduced early in 2015 and meanwhile is a standard method to backup a complete VM in Azure. Azure Backup stores the backups in Azure and allows a restore of a VM again.

Theoretically VMs that run databases can be backed up in a consistent manner as well if the DBMS systems supports the Windows VSS (Volume Shadow Copy Service - <https://msdn.microsoft.com/en-us/library/windows/desktop/bb968832(v=vs.85).aspx> ) as e.g. SQL Server does. However, be aware that based on Azure VM backups point-in-time restores of databases is not possible. Therefore, the recommendation is to perform backups of databases with DBMS functionality instead of relying on Azure VM Backup

To get familiar with Azure Virtual Machine Backup please start here: <https://azure.microsoft.com/en-us/documentation/articles/backup-azure-vms/>.

Other possibilities are to use a combination of Microsoft Data Protection Manager installed in an Azure VM and Azure Backup to backup/restore databases. More information can be found here: <https://azure.microsoft.com/en-us/documentation/articles/backup-azure-dpm-introduction/>.

## Azure as DR site for production SAP landscapes

Since Mid 2014, extensions to various components around Hyper-V. System Center and Azure enable the usage of Azure as DR site for VMs running on-premise based on Hyper-V.

A blog detailing how to deploy this solution is documented here:

[**Protecting SAP Solutions with Azure Site Recovery**](http://blogs.msdn.com/b/saponsqlserver/archive/2014/11/19/protecting-sap-solutions-with-azure-site-recovery.aspx)

## Summary

The key points of High Availability for SAP systems in Azure are:

* At this point in time, the SAP single point of failure cannot be secured as it can in on-premises deployments. The reason is that Shared Disk clusters can’t yet be built in Azure without the use of 3rd party software.
* For the DBMS layer you need to use DBMS functionality that does not rely on shared disk cluster technology. Details are documented in the guide: ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’
* To minimize the impact of problems within Fault Domains in the Azure infrastructure or host maintenance, you should use Azure Availability Sets:
  + It is recommended to have one Availability Set for the SAP application layer.
  + It is recommended to have a separate Availability Set for the SAP DBMS layer.
  + It is NOT recommended to apply the same Availability set for VMs of different SAP systems.
* For Backup purposes of the SAP DBMS layer, please check the guide: ‘*DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services*’.
* Backing up SAP Dialog instances makes little sense since it is usually faster to redeploy simple dialog instances.
* Backing up the VM which contains the global directory of the SAP system and with it all the profiles of the different instances, does make sense and should be performed with Windows Backup. Since there are differences between Windows Server 2008 (R2) and Windows Server 2012 (R2), which make it easier to backup using the more recent Windows Server releases, we recommend to run Windows Server 2012 (R2) as guest operating system.