

Clustering SAP ASCS/SCS Instance using Windows Server Failover Cluster on Azure with SIOS DataKeeper

Microsoft Corporation

Version: 2.00

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 way how SAP single point of failure components like SAP ASCS/SCS can be protected in Azure. Components of SAP ASCS/SCS are essential for the functionality of a SAP NetWeaver systems, e.g. SAP NetWeaver ABAP, SAP NetWeaver Java, SAP NetWeaver ABAP+Java systems. Therefore, high-availability functionality needs to be put in place to make sure that those components can sustain a failure of a server or a VM. For on premises deployments using the Windows Operating Systems, SAP integrates these components into Microsoft Windows Server Failover Cluster. 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. However, with the help of the product DataKeeper by SIOS, Windows Server Failover Cluster configurations as needed for SAP ASCS/SCS can be built on the Azure IaaS platform. This document describes how to use SIOS DataKeeper to set up a highly available SAP ASCS/SCS configuration 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.

© 2015 Microsoft Corporation. All rights reserved.

Contents

[1 Summary 6](#_Toc426658059)

[1.1 Resources 7](#_Toc426658060)

[1.2 Definitions upfront 8](#_Toc426658061)

[2 Windows Failover Clustering 10](#_Toc426658062)

[2.1 Quorum Modes 10](#_Toc426658063)

[3 Windows Failover Cluster on premises 11](#_Toc426658064)

[3.1 Storage 12](#_Toc426658065)

[3.2 Networking / Name Resolution 12](#_Toc426658066)

[4 Windows Failover Cluster with Microsoft Azure 13](#_Toc426658067)

[4.1 Name Resolution on Microsoft Azure 13](#_Toc426658068)

[4.2 Shared Disk on Microsoft Azure with SIOS DataKeeper 14](#_Toc426658069)

[4.3 SAP ASCS/SCS High Availability (HA) On Microsoft Azure 16](#_Toc426658070)

[5 Infrastructure Preparation 18](#_Toc426658071)

[5.1 Prerequisites 18](#_Toc426658072)

[5.2 Azure Virtual Network 18](#_Toc426658073)

[5.3 Dedicated Cloud Services 18](#_Toc426658074)

[5.4 Virtual Machines 19](#_Toc426658075)

[5.5 Data Disks 19](#_Toc426658076)

[5.6 Azure Internal Load Balancer 20](#_Toc426658077)

[5.7 Defined Availability Sets for the Cluster Nodes 20](#_Toc426658078)

[5.8 IP-Addresses / Host Names 20](#_Toc426658079)

[5.8.1 Setup Static IP-Address 21](#_Toc426658081)

[5.9 Add the Microsoft .NET Framework 3.5 Feature 21](#_Toc426658082)

[5.10 Add Machines to Domain 23](#_Toc426658083)

[5.10.1 Add Registry Entries on Both Cluster Nodes 23](#_Toc426658084)

[5.11 Windows Server Failover Cluster setup 25](#_Toc426658085)

[5.11.1 Collect Cluster nodes in Cluster configuration 25](#_Toc426658086)

[5.11.2 Configure Cluster File Share Witness 31](#_Toc426658087)

[6 Installing SIOS DataKeeper Cluster Edition 37](#_Toc426658088)

[6.1.1 Installing SIOS DataKeeper 37](#_Toc426658089)

[6.1.2 Setup SIOS DataKeeper 40](#_Toc426658090)

[7 Installation of SAP NetWeaver System 45](#_Toc426658091)

[7.1 SAP Installation with High Available ASCS/SCS Instance 45](#_Toc426658092)

[7.1.1 Create Virtual Hostname for clustered SAP ASCS/SCS 45](#_Toc426658093)

[7.1.2 ILB and Endpoints for NetWeaver System 46](#_Toc426658094)

[7.1.3 Create ILB and Endpoints for NW ABAP ASCS Instance 47](#_Toc426658095)

[7.1.4 Create ILB and Endpoints for NW Java SCS Instance 52](#_Toc426658096)

[7.1.5 Create ILB and Endpoints for NW ABAP+Java ASCS and SCS Instance 54](#_Toc426658097)

[7.1.6 Install SAP First Cluster Node 55](#_Toc426658098)

[7.1.7 Modify the SAP profile of the ASCS/SCS instance 55](#_Toc426658099)

[7.1.8 Add Probe Port 55](#_Toc426658100)

[7.1.9 Installing the Database Instance 57](#_Toc426658101)

[7.1.10 Installation Second Cluster Node 57](#_Toc426658102)

[7.1.11 Change Windows Service Startup Type of SAP ERS Instance 57](#_Toc426658103)

[7.1.12 Installation of SAP Primary Application Server (PAS) 58](#_Toc426658104)

[8 Testing SAP ASCS/SCS Instance Failover and SIOS Replication 59](#_Toc426658105)

[8.1 Starting point – SAP ASCS/SCS instance is running on Cluster Node A 59](#_Toc426658106)

[8.2 Failover process from Node A to Node B 60](#_Toc426658107)

[8.3 End point – SAP ASCS/SCS instance is running on Cluster Node B 60](#_Toc426658108)

[9 Links 62](#_Toc426658109)

# Summary

This document describes how to install and configure a high available (HA) SAP Central Services instance ASCS/SCS in a Windows Failover Cluster (WSFC) using Microsoft Azure as a platform.

As of August 2015 Azure does not provide a shared disk that could be used to create a Windows Failover Cluster configuration required by SAP. However Microsoft supports 3rd party shared disk clustering software on Azure to build Windows based highly available configurations required for these critical SAP components (see <https://support.microsoft.com/en-us/kb/2721672/>). Additional the usage of 3rd party software, there are several key differences between running highly available SAP Central Services on Azure compared to traditional hosted or on premises solutions.

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 to route the traffic.

This document explains how to overcome the lack of support for both cluster shared volumes and multiple IP addresses by using the 3rd-party software SIOS DataKeeper and the Azure Internal Load Balancer.

Normally, a complete HA architecture would not only include the ASCS/SCS instance, but other SAP components as well, like DBMS, SAP application servers, and on infrastructure side AD/DNS.

In this whitepaper we are focusing ONLY on ASCS/SCS HA, and by purpose NOT describing:

* DBMS High Availability
* SAP Application Server High Availability
* AD/DNS High Availability etc.

Therefore, we simplified the setup used throughout this paper with only one primary application server (PAS) and a non-clustered clustered DBMS running on one VM. AD/DNS is running on a separate Azure VM.

In a productive environment you need to separately deal with:

* HA of **SAP application servers** – here you need at least two SAP application servers running on at least two VMs within an Azure Availability Set.
* HA of **DBMS** - you would use different DBMS specific HA technologies like for example SQL Server AlwaysOn or Oracle Data Guard).
* HA of **AD/DNS** – you would need to distribute AD/DNS services across two (or more) Windows OS instances. As you would have a VPN connection from your on-premises site to Azure via Site-to-Site or ExpressRoute connection, your AD/DNS would run on premises, and may be extended to Azure VMs, to improve the response time from Azure VMs to AD/DNS, and to reduce the network traffic between Azure VMs and AD/DNS.

Therefore, this document only describes some basic cluster information that needs to be understood, the infrastructure preparation steps, installation of a clustered ASCS/SCS instance, installation of an SAP NetWeaver PAS with SQL Server as DBMS, Azure Internal Load Balancer (ILB) and SIOS DataKeeper software.

We assume you as the reader being familiar with Azure, Azure Cloud Services and how to implement and deploy SAP NetWeaver based software on Azure.

## Resources

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

* SAP NetWeaver on Azure Virtual Machines – Planning and Implementation Guide.
* SAP NetWeaver on Microsoft Azure Virtual Machine Services - Deployment Guide.
* DBMS Deployment Guide for SAP on Microsoft Azure Virtual Machine Services.
* SAP NetWeaver - Building an Azure based Disaster Recovery Solution.

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 (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 based 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>

## 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.

# Windows Failover Clustering

A Microsoft WSFC is the technical basis for a highly available SAP ASCS/SCS installation.

A failover cluster is a group of 1+n independent servers (nodes) that work together to increase the availability of applications and services. In the event node failure(s) occur, WSFC must determine the number of failures that can occur while still maintaining a healthy cluster in order to be able to provide the defined applications and/or services. Different quorum modes are available to achieve this.

## Quorum Modes

With WSFC four different quorum modes are available:

* **Node Majority**: Each node that is available and in communication can vote whenever they are available and in communication. The cluster functions only with a majority of the votes, that is, more than half. This option is recommended in case of an uneven number of nodes. For example: 3 nodes in a 7 node cluster can fail and the cluster will still achieve a majority and continue to run.
* **Node and Disk Majority**: Each node plus a designated disk in the cluster storage (the “disk witness”) can vote whenever they are available and in communication. The cluster functions only with a majority of the votes, that is, more than half. This mode makes sense in a cluster environment with an even number of nodes. As long as half of the nodes plus the disk are online the cluster remains in a healthy state.
* **Node and File Share Majority**: Each node plus a designated file share created by the administrator (the *file share witness*) can vote whether they are available and in communication. The cluster functions only with a majority of the votes, that is, more than half. This mode makes sense in a cluster environment with an even number of nodes and is similar to the *Node and Disk Majority* mode while it uses a witness file share instead of a witness disk. It is easy to implement but if the file share itself is not highly available it might become a single point of failure.
* **No Majority**: Disk Only: The cluster has quorum if one node is available and in communication with a specific disk in the cluster storage. Only the nodes that are also in communication with that disk can join the cluster. This mode is not recommended.

**Note**: Windows Server 10 provides another option with “Node + Cloud Witness”.

For more information on quorum modes, refer to the links at the end of this document.

# Windows Failover Cluster on premises

In this example, we have a cluster consisting of two nodes. If the network connection between the nodes fails while both nodes stay up and running, it is necessary to clarify which node is supposed to keep providing the applications and services of the cluster. A quorum disk or file share serves this purpose. The node that has access to the quorum disk or file share is the one to ensure accessibility of the services.

* In this example we are using a two node cluster for this reason we chose the *node and file share* quorum mode. The *node and disk majority* is also a valid option. In a productive environment it is recommended to use a quorum disk instead and use network and storage system technology to make it highly available.



Figure 1: Proposed Windows Server Failover Cluster configuration for SAP ASCS/SCS on Azure

## Storage

The figure above shows a shared storage cluster with two nodes. In a shared storage cluster on premises there is a shared storage that is visible for all nodes within the cluster. A locking mechanism protects the data against corruption. Additionally, all nodes can detect if another node fails. If one node fails, the remaining one takes ownership of the storage resources and ensures the availability of the services.

## Networking / Name Resolution

The cluster is reachable over a virtual IP-address and a virtual hostname provided by the DNS-server. The nodes on premises and the DNS Server can handle multiple IP addresses.

In a typical setup two or more network connections are used:

* A dedicated connection to the storage.
* A cluster-internal network connection for the heartbeat.
* A public network which is used by the clients to connect to the cluster.

# Windows Failover Cluster with Microsoft Azure

Compared to bare-metal or private cloud deployments, the Microsoft Azure Virtual Machine Services requires additional steps to configure a WSFC. In order to build a cluster a Shared Cluster Disk, several IP addresses and virtual hostnames are required for SAP ASCS/SCS instance.

Below we discuss the additional concepts and steps required when building an SAP HA Central Services cluster on Microsoft Azure. The steps show how to set up the 3rd party tool SIOS DataKeeper and configure the Azure Internal Load Balancer. These tools will give us the possibility to create a Windows Failover Cluster with a File Share Witness in Microsoft Azure

## Name Resolution on Microsoft Azure

The Microsoft Azure cloud platform doesn’t provide possibility to configure virtual IP addresses, e.g. floating IPs. For this reason, you need an alternative solution to set up a virtual IP address to reach the cluster resource in the cloud.

Azure provides the Internal Load Balancer (ILB). With the ILB the cluster can be reached over the cluster virtual IP address.

First, you need to deploy the ILB in the cloud service, which contains the cluster nodes. Then you need to configure all necessary port forwarding rules with the probe ports.

The clients can connect via the virtual hostname. The DNS Server resolves the Cluster-IP-Address and the ILB handles the forwarding to the active node of the cluster.



Figure 2: Schema of a Windows Server Failover Cluster configuration in Azure without Shared Disk

## Shared Disk on Microsoft Azure with SIOS DataKeeper

As of July 2015, Microsoft Azure does not provide shared storage to create a shared storage cluster. The 3rd party software SIOS DataKeeper Cluster Edition allows to create a mirrored storage which simulates a cluster shared storage. The SIOS solution provides real-time synchronous data replication. The way how a shared disk resource for a cluster is created is:

* Having an additional Azure VHD attached to each of the VMs that are in a Windows Cluster configuration.
* Having SIOS DataKeeper Cluster Edition running on both VM nodes.
* Having SIOS DataKeeper Cluster Edition configured in a way that it 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 one shared disk.

For more details on the SIOS DataKeeper product, please check this source: <http://us.sios.com/products/datakeeper-cluster/>



Figure 3: Schema of a Windows Server Failover Cluster configuration in Azure using SIOS DataKeeper

## SAP ASCS/SCS High Availability (HA) On Microsoft Azure

The following Azure environment is the base platform for the ASCS/SCS HA instance:



Figure 4: Schema of SAP ASCS/SCS Windows Server Failover Cluster configuration in Azure including the DBMS VM

In order to complete the picture as above, we added the DBMS VM. Important to understand is the following:

* The VMs running SAP ASCS/SCS and/or Enqueue Replication services with SAP ASCS/SCS need to be contained in an own Azure Cloud Service.
* The DBMS VM and/or other VMs running SAP dialog instances need to run in a separate Azure Cloud Service.
* If the DBMS needs to be protected with functionality that requires a Windows Server Failover Cluster configuration and multiple VMs as well, a second Internal Load Balancer (ILB) and a separate Azure Cloud Service are necessary.
* It is not necessary to run the domain controller in Azure. In Cross-Premises scenarios (see definition in [section 1.1](#_Definitions_upfront) ), the Domain Controller can run on premises as well.
* All VMs within an Azure Cloud Service should be added to the same Azure Availability Set, to ensure that they run in different Azure Fault and Upgrade domains (also see chapter 3 in *SAP NetWeaver on Microsoft Azure Virtual Machine Services – Planning and Implementation Guide*).

# Infrastructure Preparation

## Prerequisites

In this document we don’t want to discuss the setup of the Domain Controller and the DNS Service. As you will have network connectivity between your on-premises site and Azure (via Site-to-Site or Express Route), you will have your DC/DNS on premises, and you may extend them to Azure VMs inside of *ascsha-csd* cloud service. We describe the installation steps after the cluster nodes and the database server are joined into the domain. All Servers are running Windows Server 2012 R2 and have the Microsoft .NET Framework 3.5 deployed. The Domain Controller is configured as a Domain Controller AND DNS-Server. In this setting, all virtual machines get IP-addresses from an in Azure configured DHPC-Server. The DB- and Primary Application Server is configured in the same Cloud Service like the DC. The cluster nodes are configured in a dedicated Cloud Service. This simplifies the setup of the Internal Load Balancer.

## Azure Virtual Network

We configured a new Azure virtual network, which is logical extension of your on-premises network.

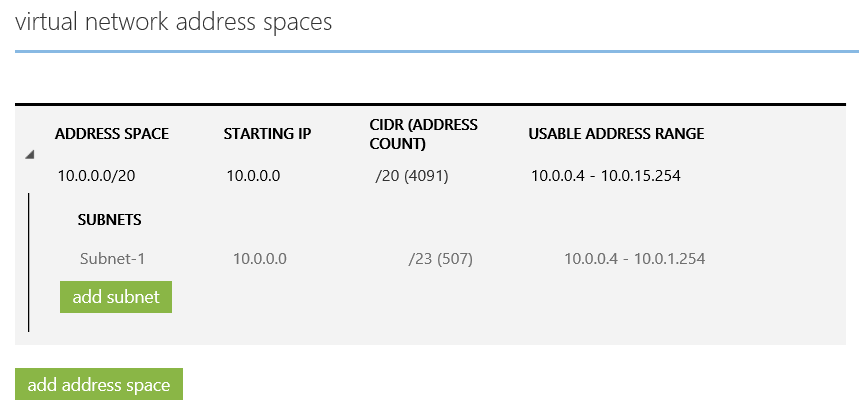


Figure 5: Configuration of Azure Virtual Network

## Dedicated Cloud Services

In this virtual network we need a dedicated Azure Cloud Service for the base Services like DC, DNS and DBMS (*ascsha-csd*). Additional we need a dedicated Cloud Service for the cluster nodes (*ascsha-clu*).

It is required to use two separate cloud services because it separates the nodes that build the cluster and are covered by the ILB from the nodes that are not covered by the ILB.

Two VMs are residing in the Cluster Service *ascsha-csd*:

* The first VM is the **Domain Controller** which also provides the **DNS-Service**. For the configuration example which guides us through this documentation we configured **cluster file share quorum** on this VM. As we all know the file share quorum cannot be placed on a node that is part of the Windows Cluster.
* On the second VM we deployed the DBMS together with the **SAP Primary Application Server (PAS)**.

The two nodes which will represent the cluster to protect the SAP ASCS/SCS instance are residing in the second cluster service *ascsha-clu*.

## Virtual Machines

With the configuration as described, you need four VMs:

|  |  |  |  |
| --- | --- | --- | --- |
| **VM Role** | **VM hostname** | **IP Address** | **Cloud Service** |
| Domain Controller, DNS, File Server for cluster file share quorum | ascsha-dc | 10.0.0.10 | ascsha-csd |
| SQL Server, SAP PAS | ascsha-dbas | 10.0.0.15 | ascsha-csd |
| Cluster Node A for ASCS/SCS-Instance | ascsha-clna | 10.0.0.41 | ascsha-clu |
| Cluster Node B for ASCS/SCS-Instance | ascsha-clnb | 10.0.0.42 | ascsha-clu |

Table 1: List of VMs in our example configuration

## Data Disks

With the configuration as described, you need the following disks/additional VHDs:

|  |  |  |
| --- | --- | --- |
| **VM hostname** | **Purpose** | **Size** |
| scsha-dbas | Data disks for SAP PAS and Data and log files | 1TB  (for standard storage) |
| ascsha-clna | Data Disk for SIOS-Mirror for clustered ASCS/SCS instance | 1TB (for standard storage) |
| ascsha-clnb | Data Disk for SIOS-Mirror for clustered ASCS/SCS instance | 1TB (for standard storage) |

Table 2: Additional Disks for the different VMs in our example

## Azure Internal Load Balancer

For more details on what the Azure Internal Load Balancer (ILB) is and how it works the following sources should be studied:

* <http://azure.microsoft.com/blog/2014/05/20/internal-load-balancing/>
* <https://azure.microsoft.com/en-us/documentation/articles/load-balancer-internal-getstarted>
* <http://michaelwasham.com/windows-azure-powershell-reference-guide/using-the-internal-load-balancer-with-azure-virtual-machines>

As described earlier an Azure Internal Load Balancer is required for creating the Windows Server Failover Cluster configuration. The Internal Load Balancer is suitable for the cluster configuration and does not expose the IP address of the cluster service externally or to the public. Hence an ILB configuration got created for *ascsha-clu* cloud service in order to forward requests to the specific cluster node where SAP ASCS/SCS instance is actually up and running.

## Defined Availability Sets for the Cluster Nodes

Furthermore, we have to define an Azure Availability Set for the Cluster nodes. It is strongly recommended to specify an Azure Availability Set if you use more than one VM for the same purpose. Setting up an Azure Availability Set makes sure that your VMs get deployed in different Fault and Upgrade Domains within an Azure Region. This avoids that maintenance activities or unplanned downtime will have impact to both VMs at the same time.

## IP-Addresses / Host Names

We used the following IP addresses for VMs and virtual host names of cluster and SAP ASCS/SCS instance:

|  |  |  |
| --- | --- | --- |
| **IP-Address** | **Hostname** | **Description** |
| 10.0.0.10 | ascsha-dc | Domain-Controller / DNS |
| 10.0.0.15 | ascsha-dbas | DB-Server / Prim. Appl. Server |
| 10.0.0.41 | ascsha-clna | Cluster-Node 1 |
| 10.0.0.42 | ascsha-clnb | Cluster-Node 2 |
| 10.0.0.40 | ascsha-clres | Virt. IP / Hostname 🡪 Cluster Resource |
| 10.0.0.44 | ascsha-clsap | Virt. IP / Hostname 🡪 SAP ASCS/SCS Cluster Service and Azure Load Balancer (ILB) |

Table 3: Names/Virtual names and IP addresses for VMs used in our configuration

### Setup Static IP-Address

After deploying the virtual machines for clustering we have to setup static IP-addresses. This can’t be done within the Guest-OS, but needs to be configured in the Azure Virtual Network configuration. This is best done using Windows PowerShell. More information on Azure Static IP addresses can be found here:

<http://azure.microsoft.com/blog/2014/04/22/static-internal-ip-address-for-virtual-machines/>   
<https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-reserved-private-ip/>

We have to setup the IP-address via PowerShell. You need the Azure module for PowerShell installed on the machine you use to administrate your Azure deployments. For more information on how to import Azure modules check the link section on the end of this document.

# Define Cloud Service Names

$DBASCloudService = "ascsha-csd"

$ClusterCloudService = "ascsha-clu"

# Define VM Names and IP Addresses

$AscsHaDcVMName = "ascsha-dc"

$AscsHaDcVMIP = "10.0.0.10"

$AscsHaDbAsVMName = "ascsha-dbas"

$AscsHaDbAsVMIP = "10.0.0.15"

$AscsHaCLNodeAVMName = "ascsha-clna"

$AscsHaCLNodeAVMIP = "10.0.0.41"

$AscsHaCLNodeBVMName = "ascsha-clnb"

$AscsHaCLNodeBVMIP = "10.0.0.42"

#set static IP for DC/DNS server

Get-AzureVM -ServiceName $DBASCloudService -Name $AscsHaDcVMName | Set-AzureStaticVNetIP -IPAddress $AscsHaDcVMIP | Update-AzureVM

#set static IP for the DBMS / PAS server

Get-AzureVM -ServiceName $DBASCloudService -Name $AscsHaDbAsVMName | Set-AzureStaticVNetIP -IPAddress $AscsHaDbAsVMIP | Update-AzureVM

#set static IP for the 1st cluster node

Get-AzureVM -ServiceName $ClusterCloudService -Name $AscsHaCLNodeAVMName | Set-AzureStaticVNetIP -IPAddress $AscsHaCLNodeAVMIP | Update-AzureVM

#set static IP for the 2nd cluster node

Get-AzureVM -ServiceName $ClusterCloudService -Name $AscsHaCLNodeBVMName | Set-AzureStaticVNetIP -IPAddress $AscsHaCLNodeBVMIP | Update-AzureVM

After executing these commands, you need to reboot the VMs to get the static IP addresses assigned. Please note that static IP addresses are set for the existing VMs only. No static IP was assigned to the virtual names as listed in **Table 3**.

## Add the Microsoft .NET Framework 3.5 Feature

The Microsoft .NET framework 3.5 is not automatically enabled or installed on most recent Windows Server releases. However, SIOS DataKeeper requires the .NET framework on all nodes the product is getting installed on. Therefore, it is required to install .NET 3.5 on all the Guest OS of the different VMs. Keep in mind that e.g. SQL Server, even with its most recent release still requires .NET 3.5 frameworks as well.

There are two ways to add .Net 3.5 Framework. The first possibility is to use ‘Add Roles and Features’ in Windows as shown below:

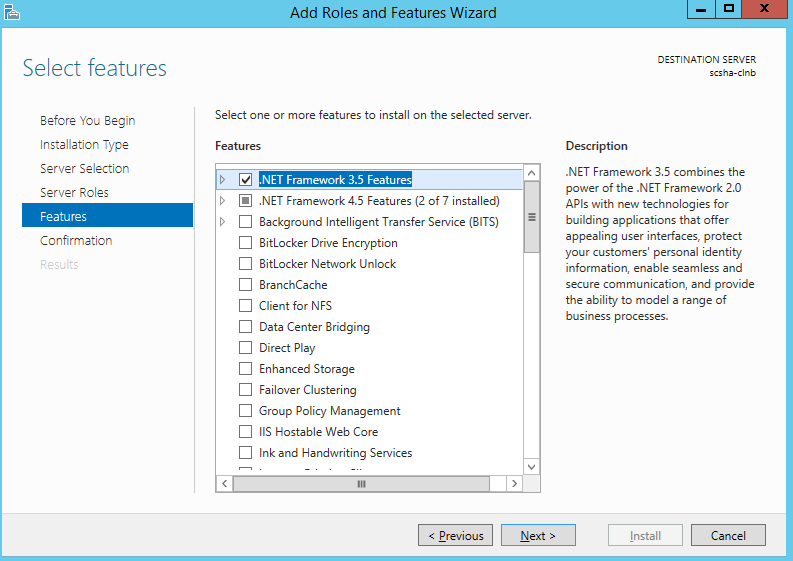


Fig.6: Install .Net framework 3.5 through ‘Add Role and Features Wizard’

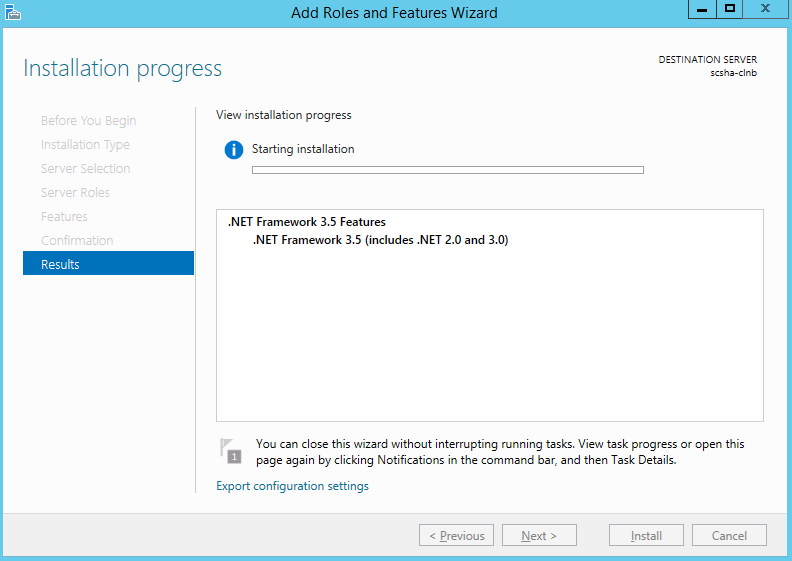


Fig.7: Progress bar installing .Net framework 3.5 through ‘Add Role and Features Wizard’

The second possibility to enable the .Net Framework 3.5 feature is using the command line tool *dism.exe*. For this type of installation, you need to have the ‘sxs’ directory of the Windows install media accessible. The following command needs to be executed in a command line window with administrator rights:

**Dism /online /enable-feature /featurename:NetFx3 /All /Source:installation\_media\_drive:\sources\sxs /LimitAccess**

## Add Machines to Domain

After assigning static IP addresses to the VMs, add the VMs to the domain.

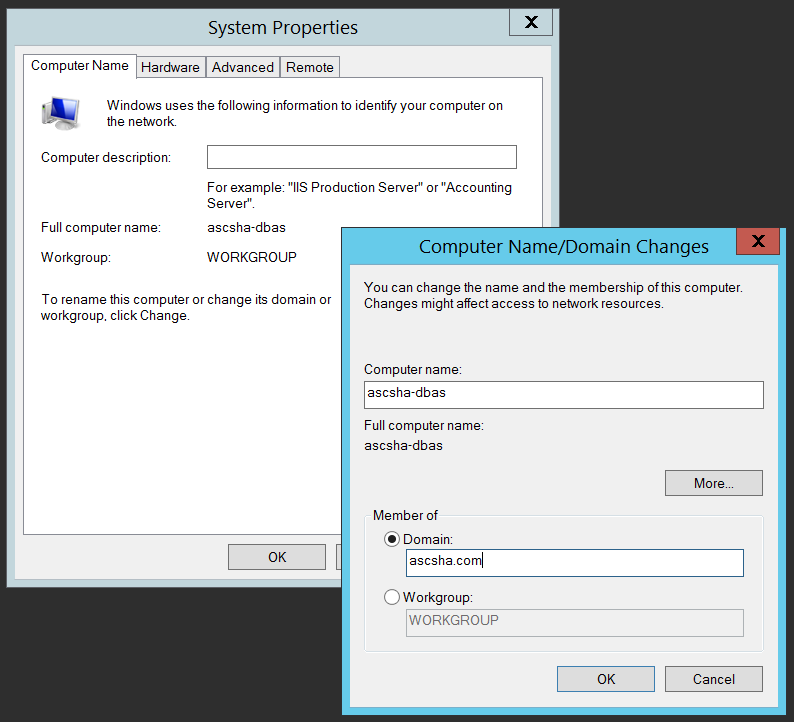


Fig.8: Adding a VM to a domain

Achtung **Important:**

You will have Cross-Premises scenarios with SAP, which means that your domain is on on-premises, and could be extended into Azure.

In our simplified example landscape, the Domain Controller is not installed as on on-premises Windows machine, but on Azure VM *ascsha-dc* [10.0.0.10].

### Add Registry Entries on Both Cluster Nodes

Azure Load Balancers, including the Azure Internal Load Balancer are closing connections when these connections are idle for a certain amount of time. On the other hand, SAP work processes in dialog instances are opening connections to the SAP Enqueue Process as soon as the first enqueue/dequeue request needs to be sent. These connections usually remain established until the work process or the enqueue process restart. However, if the connection is idle for some time, the Azure ILB will close it. Not really an issue since the SAP work process will re-establish the connection to the enqueue process if it does not exist anymore. These activities will be documented in the developer traces of SAP processes and will therefore create a lot of content into those traces for no really good reason. Therefore, we recommend to change the TCP/IP *KeepAliveTime* and *KeepAliveInterval* on both cluster nodes. The changes of the TCP/IP parameters need to be combined with SAP profile parameters which we will describe later in this document.

* Node A ascsha-clna.ascsha.com 10.0.0.41
* Node B ascsha-clnb.ascsha.com 10.0.0.42

Add the following windows registry entries:

|  |  |
| --- | --- |
| Path | HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters |
| Variable Name | **KeepAliveTime** |
| Variable Type | REG\_DWORD (Decimal) |
| Value | 900000 |
| Link to documentation | <https://technet.microsoft.com/en-us/library/cc957549.aspx> |

Table 4: First TCP/IP parameter to be changed

|  |  |
| --- | --- |
| Path | HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters |
| Variable Name | **KeepAliveInterval** |
| Variable Type | REG\_DWORD (Decimal) |
| Value | 300000 |
| Link to documentation | <https://technet.microsoft.com/en-us/library/cc957548.aspx> |

Table 5: Second TCP/IP parameter to be changed

Then reboot both cluster nodes in order to make these parameters valid.

## Windows Server Failover Cluster setup

### Collect Cluster nodes in Cluster configuration

First step is to add the Failover Clustering Feature to both cluster nodes. This is done with the ‘Add Role and Features Wizard’ and should not require any more descriptions.

Second step would be to setup the Failover Cluster by using the Windows Failover Cluster Manager.

In Failover Cluster Manager MMC, click on create Cluster and add the name of the first cluster node A, e.g. *ascsha-clna*.

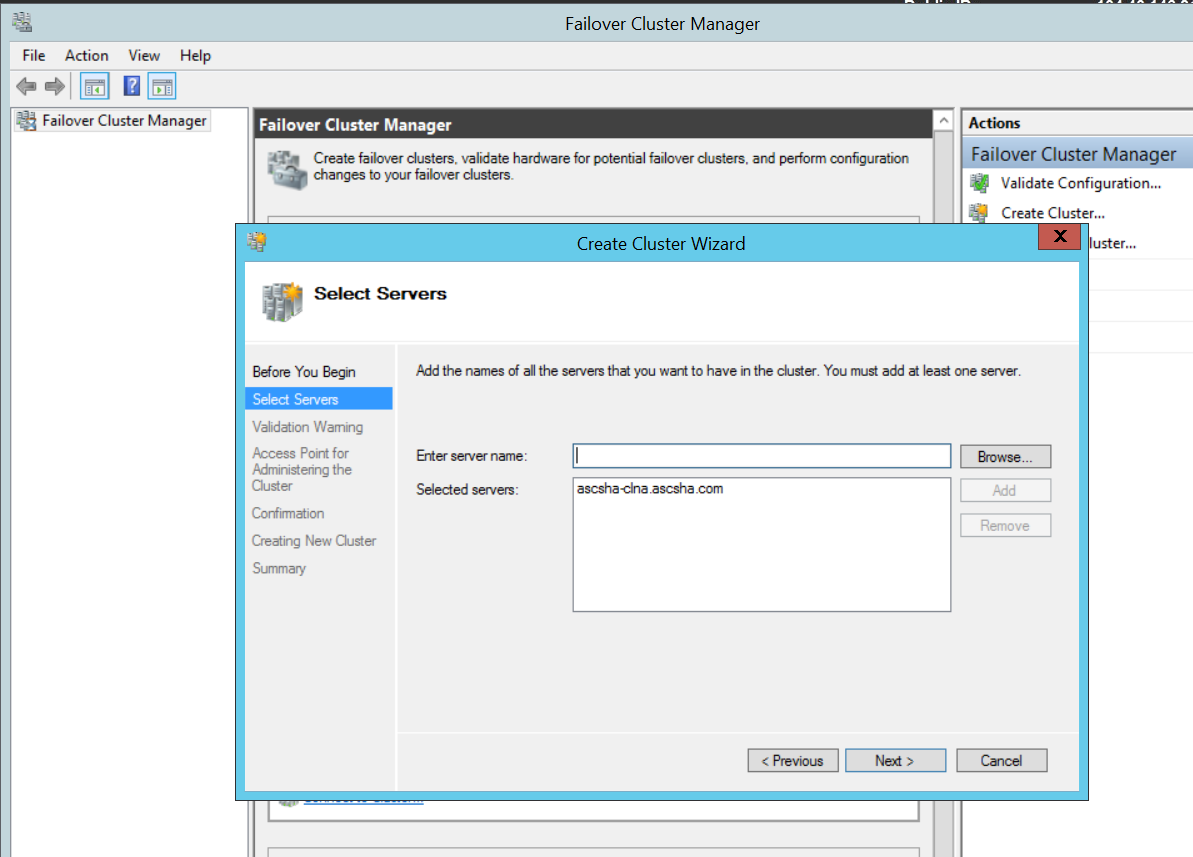


Fig.9: First Step for adding a failover cluster configuration – add server/VM name of first node that should be cluster nodes

In the next steps you are asked for network name (virtual hostname) of the cluster.

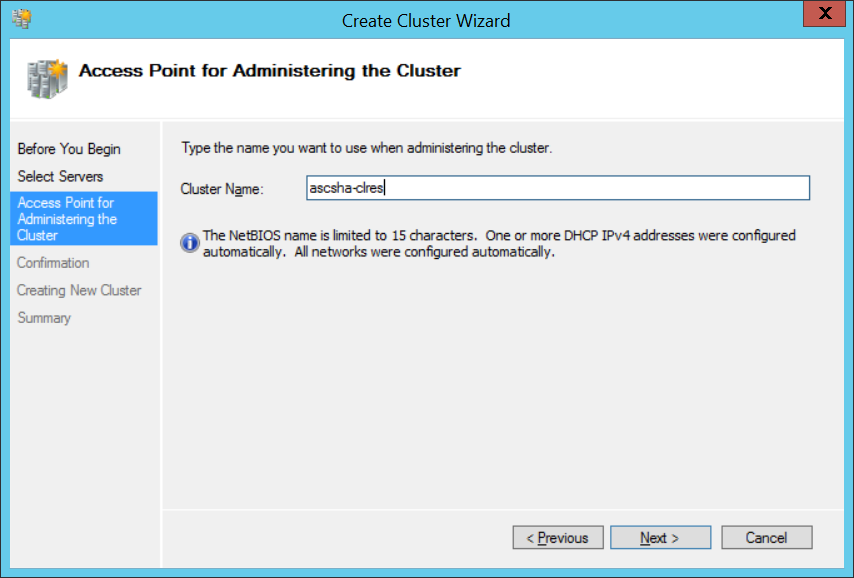


Fig.10: Second Step for adding a failover cluster configuration – define name of the cluster

Once the cluster is createde a Cluster Validation Test is run

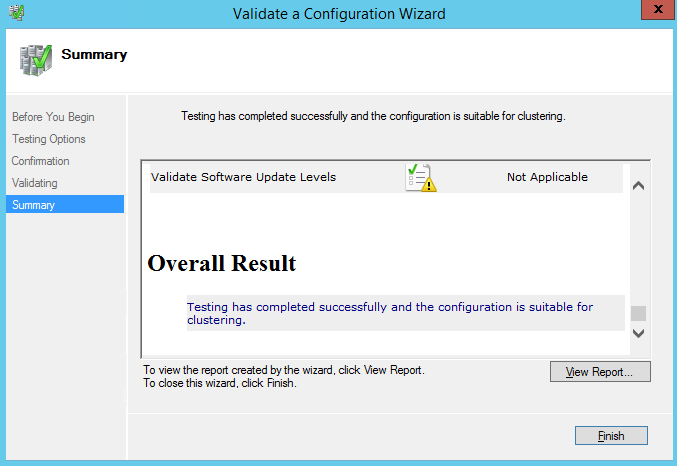


Fig.11: Last Step for adding a failover cluster configuration – Cluster Validation Check is run

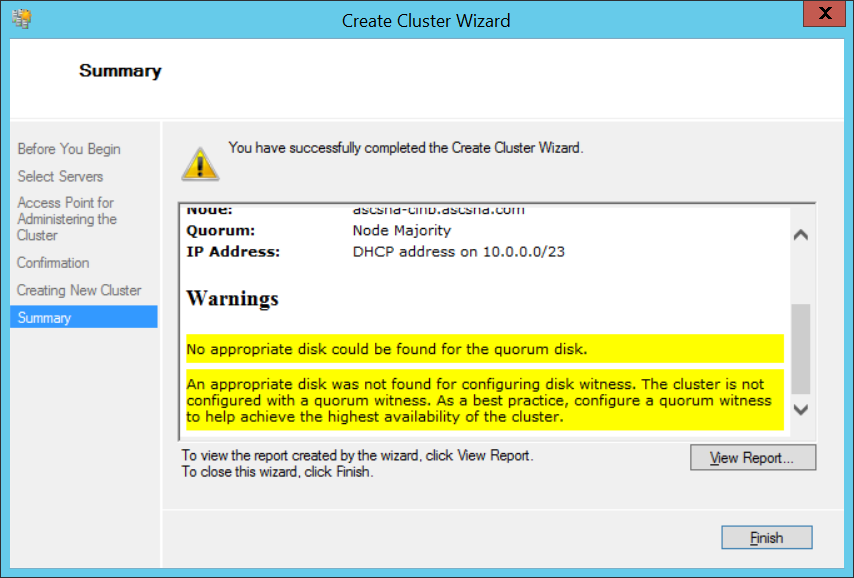


Fig.12: Last Step for adding a failover cluster configuration – Cluster Validation Check will show warnings about no quorum disk found

Any warnings about disks can be ignored at this stage, a file share witness will be added later along with the SIOS shared disks. At this stage we don’t care about a quorum.

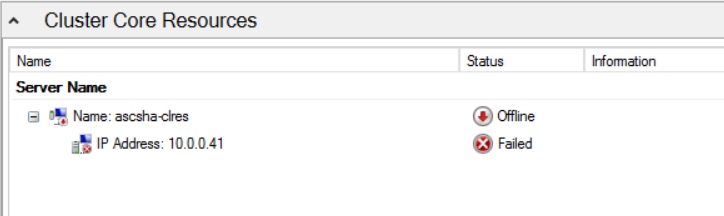


Fig.13: Cluster Core Resource with IP address is defined – new IP address needed however

Since the IP address of the server is pointing to one of the VM nodes, the cluster can’t start up. We now need to change the IP address of the core cluster service to the one we listed in table 3 in [section 5.8](#_IP-Addresses_/_Host).

According to the table we need to assign IP-Address [10.0.0.40] for the cluster virtual host name *ascsha-clres*. This is done through the property page of the IP resource of the core cluster service as shown below

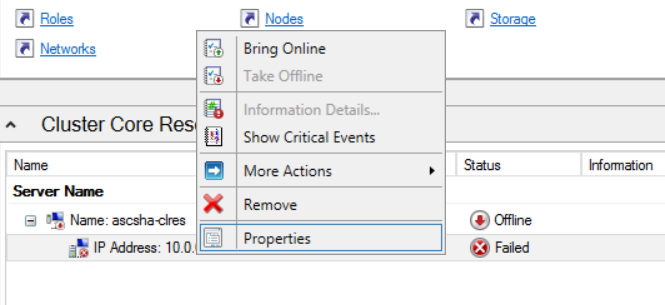


Fig.:14: Use ‘Properties’ to change to correct IP address

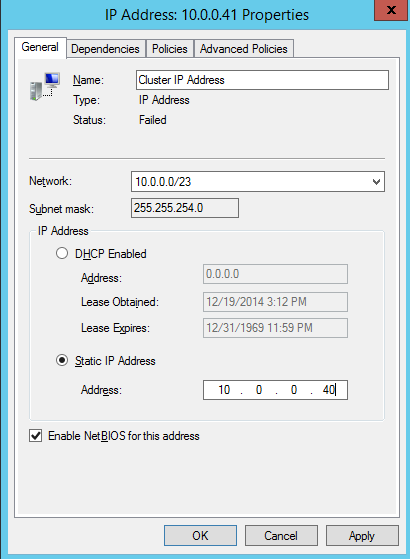


Fig.:15: Assign the IP address reserved for the cluster

After changing IP address, bring the cluster virtual hostname online.

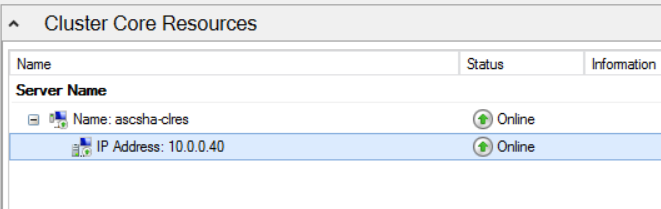


Fig.: 16: Cluster Core Service up and running with the correct IP address

Now that the Core Cluster Service is up and running you can add the second cluster node

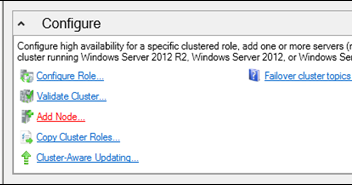


Fig.: 17: Add the second cluster node

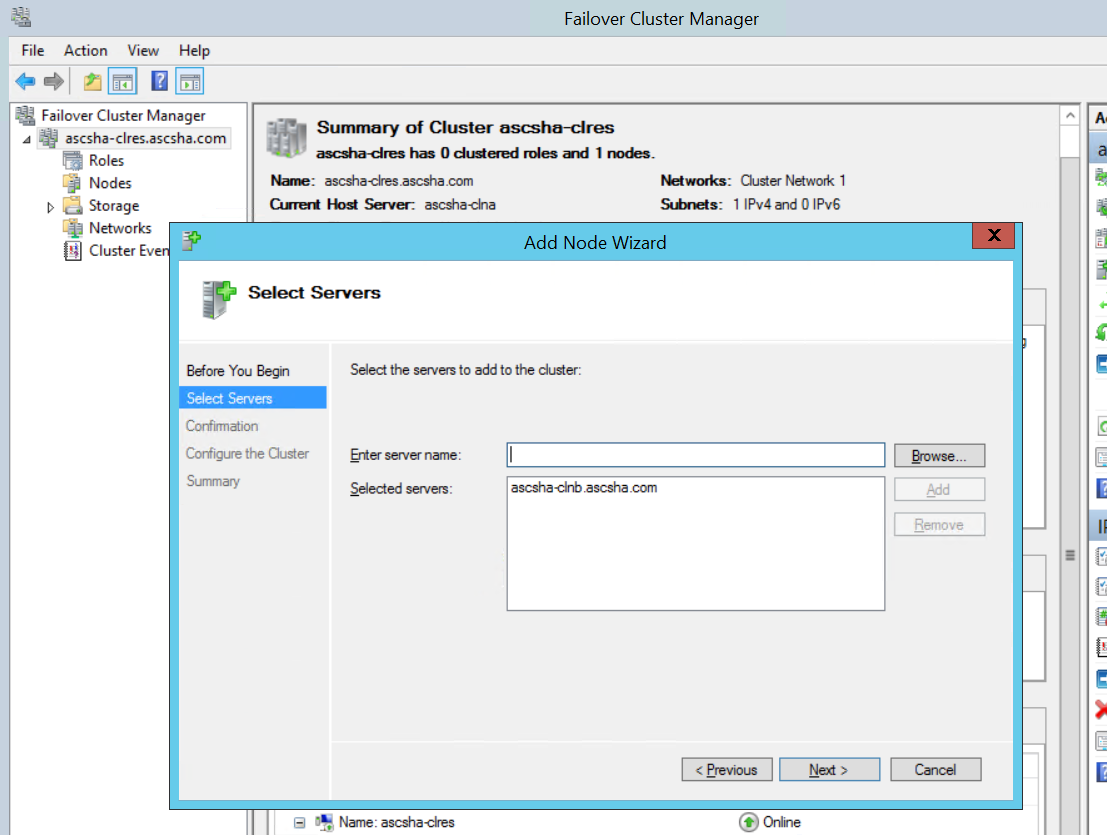


Fig. 18: Add the second cluster node hostname, e.g. *ascsha-clnb*.

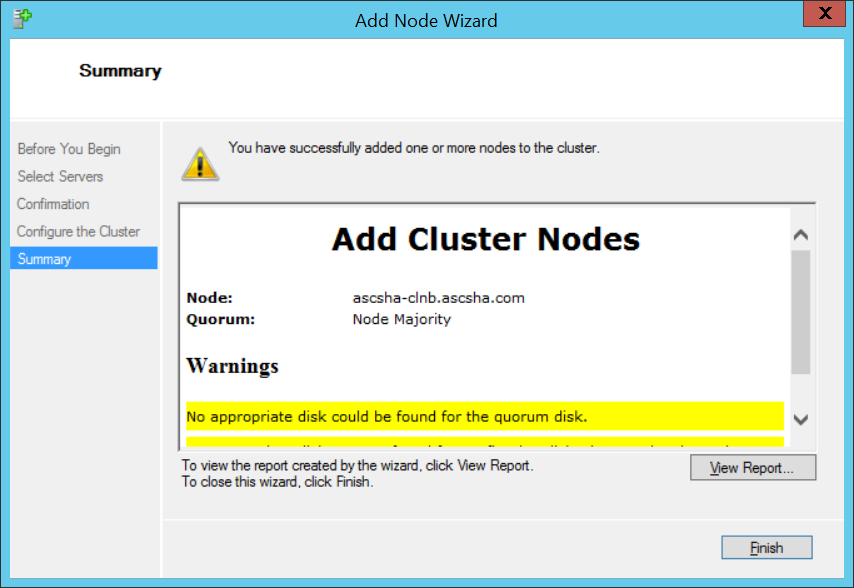


Fig.: 19: Again ignore the warning around the disk quorum

You can ignore warnings about Quorum and disks. Quorum and share disk setup will be solved later. The Cluster Shared Storage will be solved with SIOS-Installation.

### Configure Cluster File Share Witness

#### Create a File Share

We choose a File Share Witness instead of a Quorum Disk. This option is supported by SIOS DataKeeper.

In the configuration we use for illustrations in this paper, the File Share Witness is configured on the AD/DNS server *ascsha-dc*. Since you would have configured a VPN connection to Azure (via Site-to-Site or ExpressRoute), your AD/DNS resides on premises and as a result is not suitable to run a FileShare Witness.

Achtung **Important:** In the case that your AD/DNS is running only on premises, do not configure your File Share Witness on AD/DNS Windows OS running on premises, because network latency between cluster nodes running on Azure and AD/DNS on premises might be too large and cause connectivity issues. Be sure to configure the File Share Witness on an Azure Windows VM running close to cluster node, for example place it in *ascsha-csd* cloud service.

The Quorum Drive needs at least 1024 MB Free Space Recommended is 2048 MB

First step is to add the cluster name object

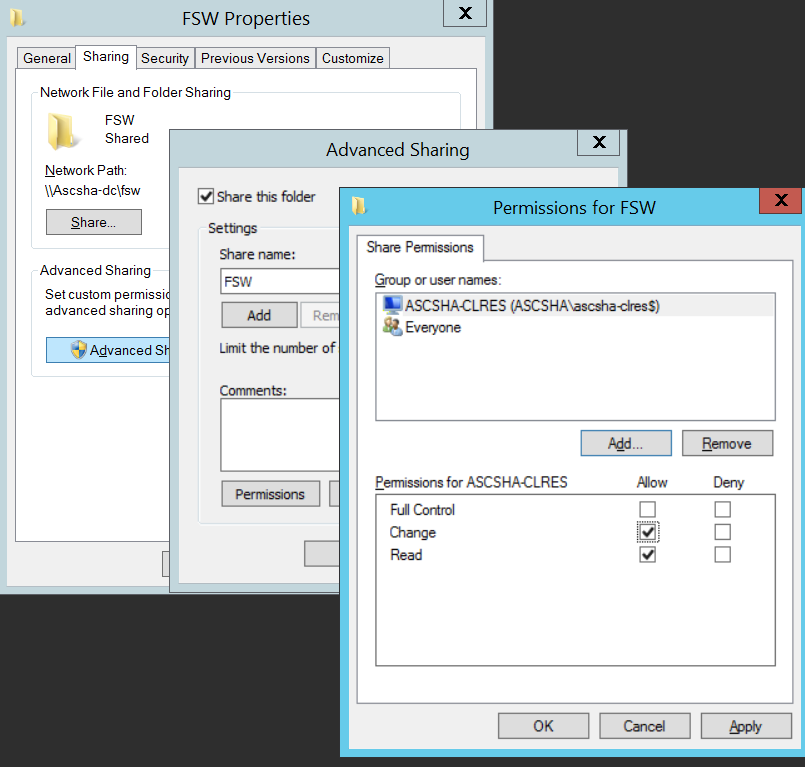


Fig.: 20: Assign the permissions on the share for the cluster name object

Make sure that the permissions include the ability to change data in the share for the cluster name object (in this case **ascsha-clre$**). In order to add the cluster name object into the list shown above you need to press ‘Add’ and then change the filter to allow checking for computer objects as well as shown below.

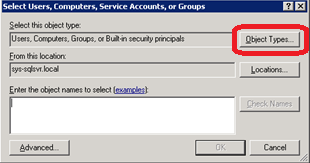


Fig. 21: Change Object type to include Computer Objects as well

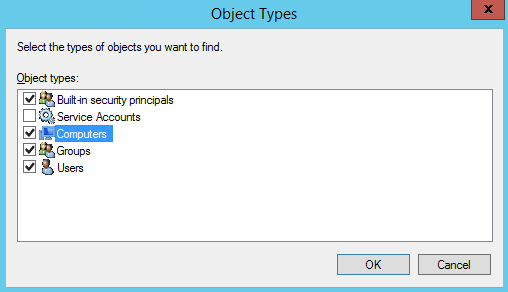


Fig. 22: Check the box for computer objects

After this enter the cluster name object in the screen shown in Figure 21. As the record shown in Fig.20 should be created now, you can change the permissions as shown in Figure 20.

Next step is to use the ‘Security’ Tab of the share and define the finer granular permissions for the cluster name object.

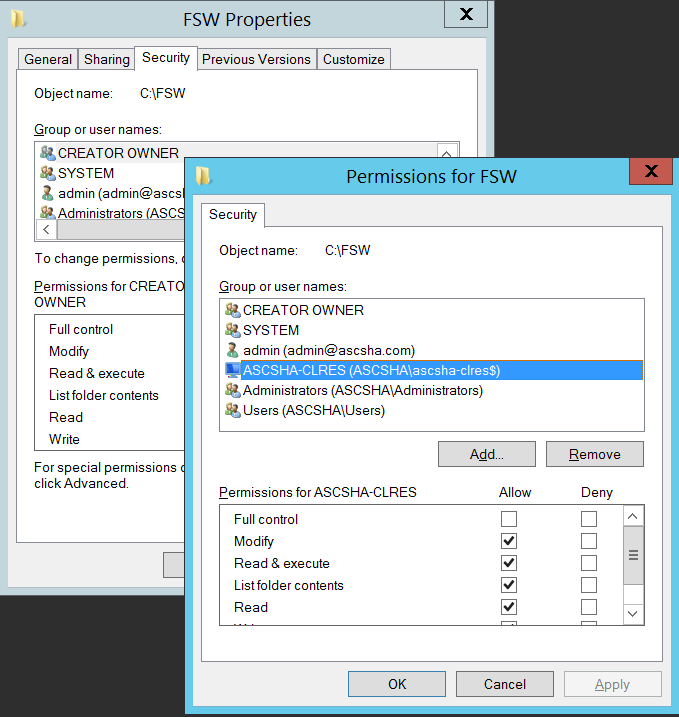


Fig. 23: Set security attributes for the cluster name object on the file share quorum

#### Configure File Share Witness Quorum in the Failover Cluster Manager

Next step is to change the cluster configuration to a file share witness using the Failover Cluster Manager.

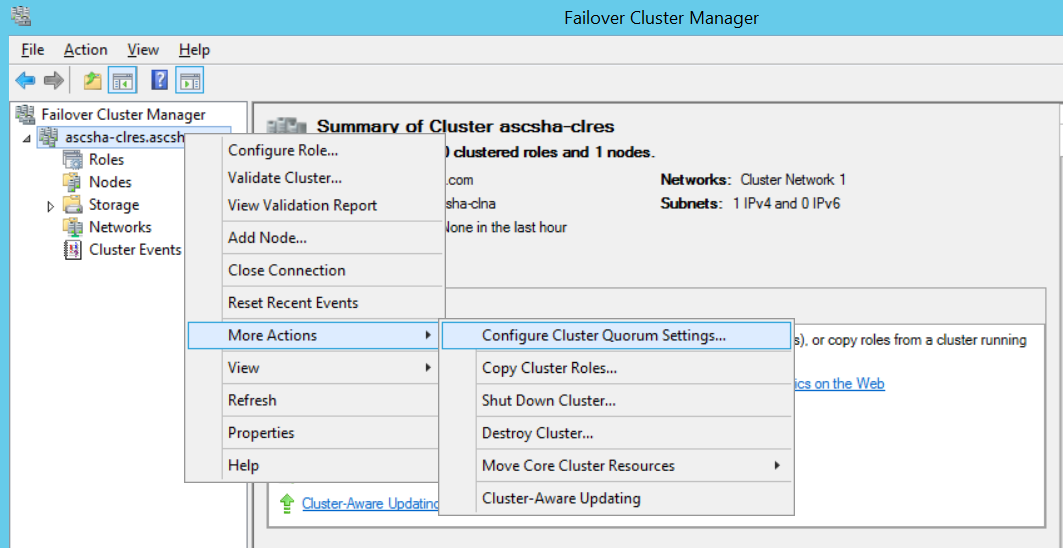


Fig. 24: Call ‘Configure Cluster Quorum Setting Wizard’ as shown here

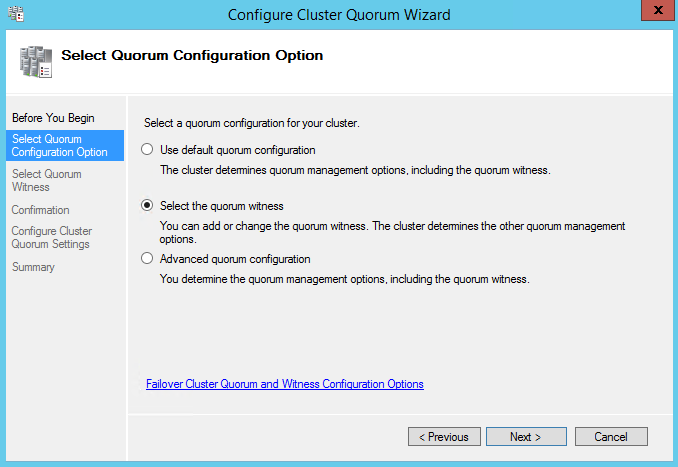


Fig. 25: Selection screen of different quorum configurations

In this selection, you need to choose ‘*Select the quorum witness’*.

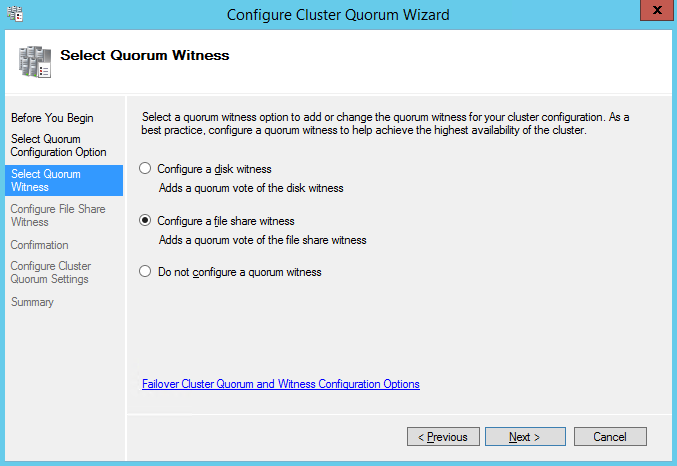


Fig. 26: Selection screen of different quorum methods

In our case you need to choose ‘*Configure a file share witness’*.

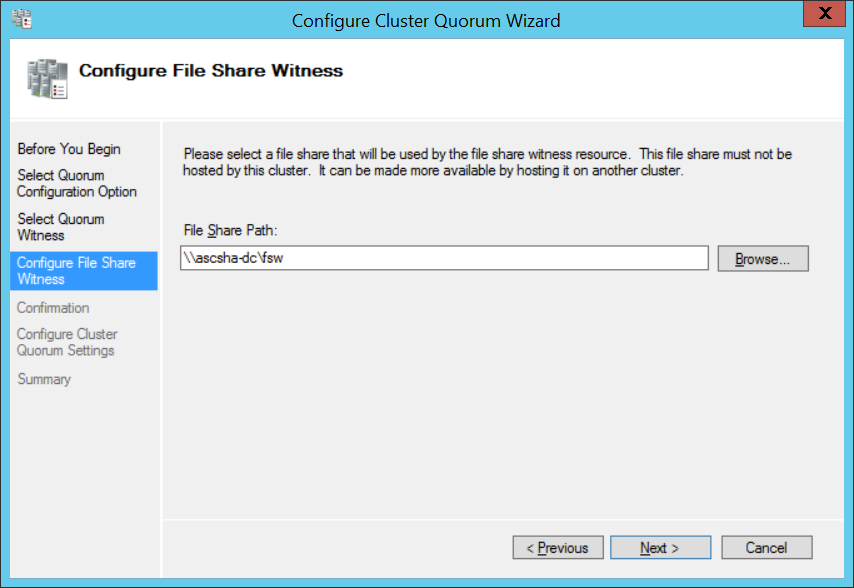


Fig. 27: Define file share location for witness share

In our example we need to add the UNC path to *FSW* our file share which is: [\\ascsha-dc\fsw](file://ascsha-dc/fsw)

Press ‘Next’ which will result in a list of the changes you want to do. Check them and press ‘Next again to change the Cluster configuration.

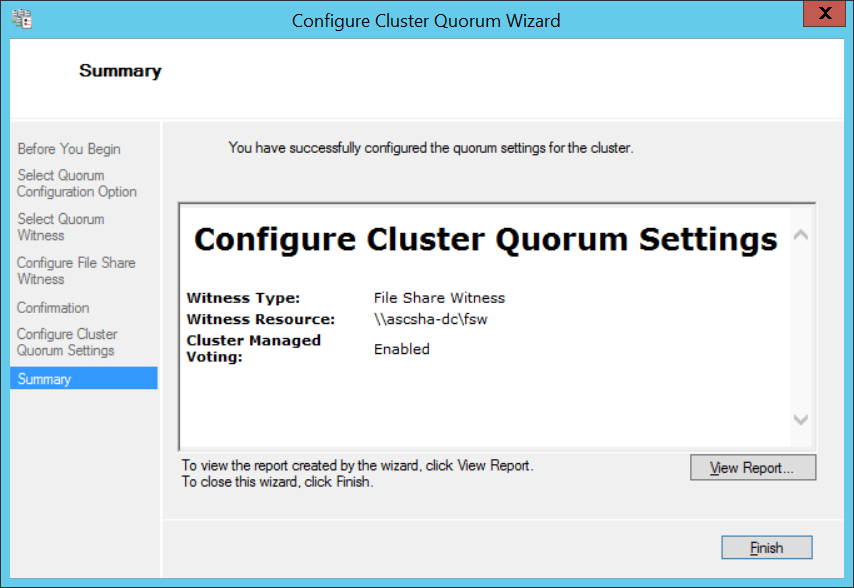


Fig. 28: Screen showing successful reconfiguration in the cluster

In this last step the cluster configuration should be reconfigured successfully.

# Installing SIOS DataKeeper Cluster Edition

The state we are in now is that we have a working Windows Server Failover Cluster configuration in Azure. However, this cluster configuration has no shared disk resource yet. In order to install a SAP ASCS/SCS, we need such shared disk resources. This is where SIOS DataKeeper Cluster Edition comes into play. Since Azure does not allow us to create such shared disk resources with the necessary functionality, we need to rely on e.g. SIOS DataKeeper to provide this functionality.

Let’s go through the installation of the SIOS DataKeeper Cluster Edition. It needs to be installed on each of our two nodes in the cluster. The SIOS DataKeeper enables the creation of virtual shared storage by creating a synced mirror and simulating Cluster Shared Storage.

Before installing the SIOS Solution you have to create a domain user *DataKeeperSvc*.   
Achtung **Important:** Add this user to the local Administrators group on both cluster nodes.

### Installing SIOS DataKeeper

Install the SIOS software on both cluster nodes



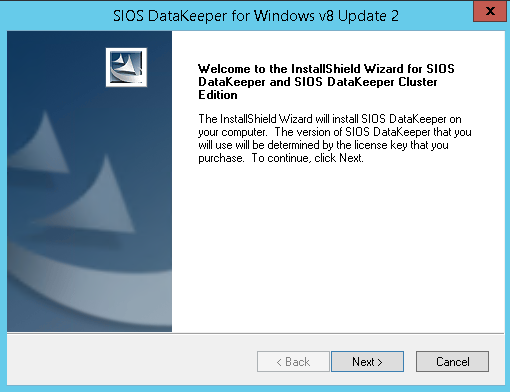


Fig. 29: First screen of SIOS DataKeeper installation

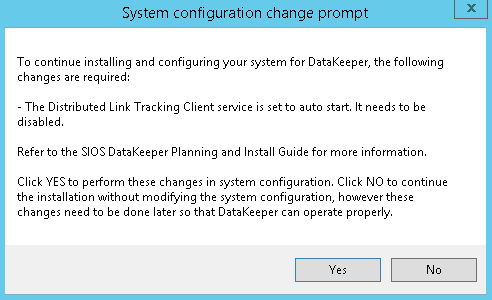


Fig. 30: DataKeeper informs of a Service to be disabled

As the screen shown in Figure 30 shows up, please choose ‘*Yes’*.

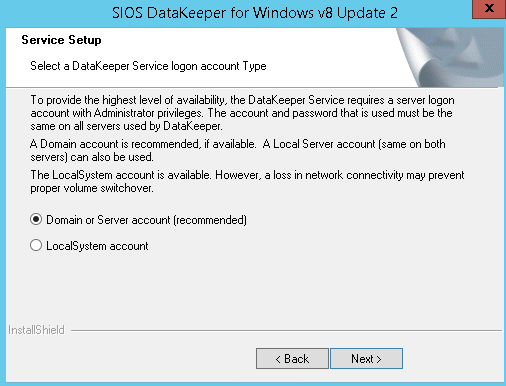


Fig. 31: User selection for SIOS DataKeeper

In the screen above we recommend to choose *Domain or Server account*.

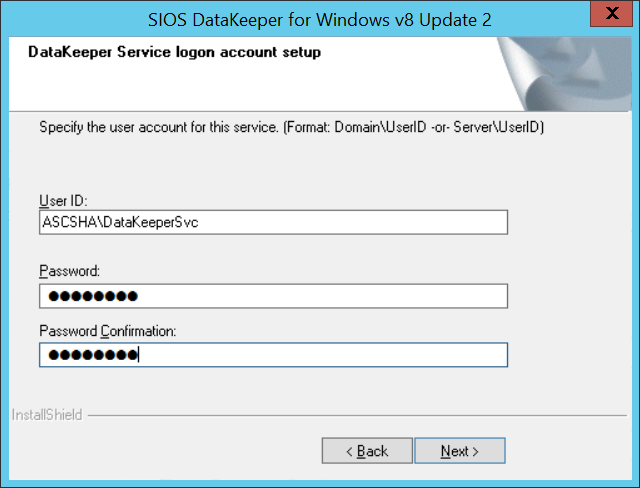


Fig. 32: Supply domain user and password to SIOS DataKeeper installation

Specify the domain account you created for SIOS DataKeeper and the passwords of that account.



Fig. 33: Provide your SIOS DataKeeper license

Install the license key for your SIOS DataKeeper in the screen shown in Figure 33. At the end of the installation you will be asked to **reboot the VM**.

### Setup SIOS DataKeeper

After installing the SIOS DataKeeper on both nodes we have to start the configuration. The goal of the configuration is to have synchronous data replication between the additional VHD attached to each of the VMs. The following steps show the configuration on both nodes.

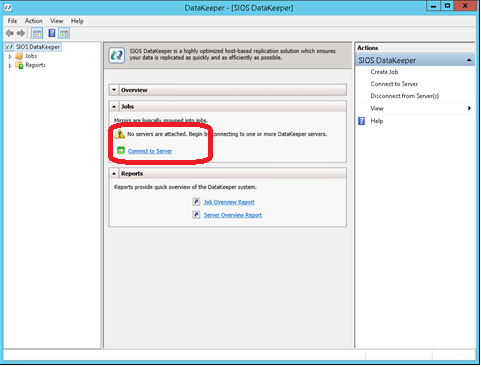


Fig. 34: DataKeeper Management and Configuration tool

Start the Management and Configuration Tool of DataKeeper and press the link ‘Connect Server’ (circled in red above)

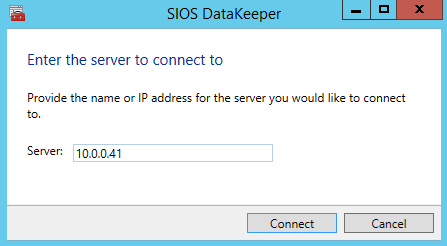


Fig. 35: Insert the name or TCP/IP address of the first node and in a second step the second node, the Management tool should connect to

The next step is to create the Replication Job between the two nodes



Fig. 36: Create Replication Job

A wizard will guide through the process

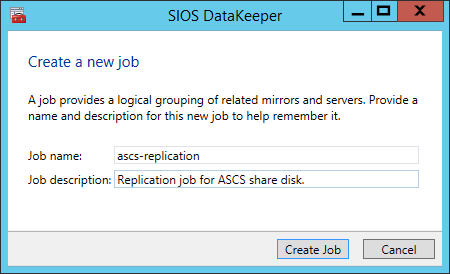


Fig. 37: Define the name of the Replication job

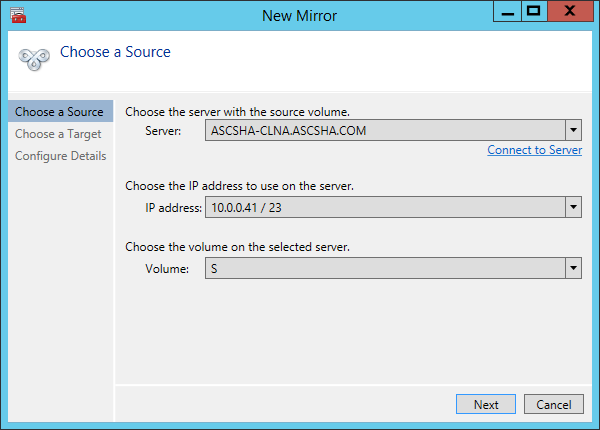


Fig. 38: Define the base data for the node which should be the current source node

In a first step the name, TCP/IP address and the disk volume of the source node needs to be defined. The second step is to define the target node. Again name, TCP/IP address and the disk volume of the target node need to be defined.

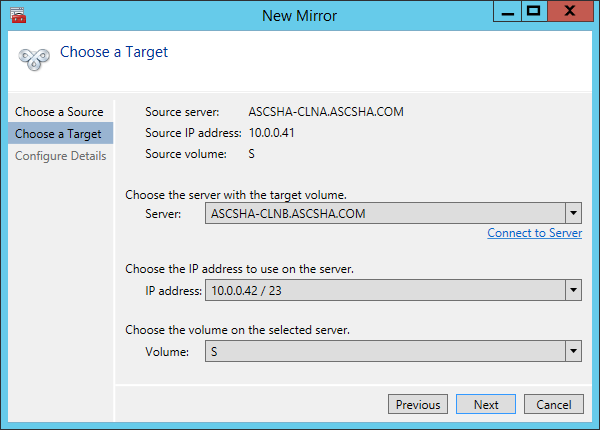


Fig. 39: Define the base data for the node which should be the current target node

The next step is to define the compression algorithms which should be applied. For our purposes we recommend to compress the replication stream. Especially in re-synchronization situations the compression of the replication stream reduced the re-synchronization time dramatically. Note that, compression is utilizing CPU and RAM resources of a VM. The higher the compression rate the more CPU is going to be utilized. You can adjust and change this setting afterwards. Another setting you need to check is whether the replication is executed asynchronously or synchronously. **For protecting SAP ASCS/SCS configurations the setting of Synchronous Replication is required**.

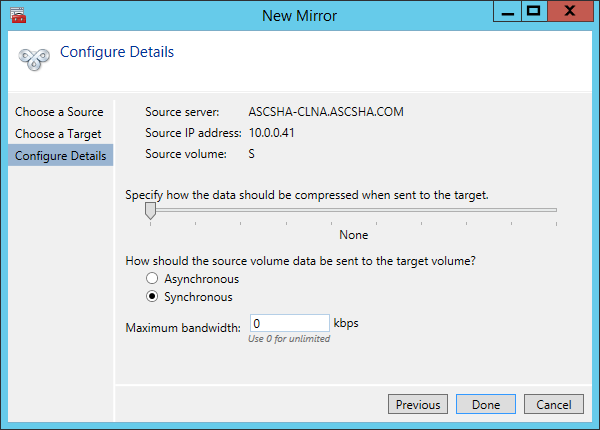


Fig. 40: Define details of the replication

The last step is to define whether the volume which is replicated by the replication job should be represented to a WSFC cluster configuration as a shared disk. For the SAP ASCS/SCS configuration we need to choose ‘YES’ so that the Windows cluster sees the replicated volume as shared disk that can be used as cluster volume.

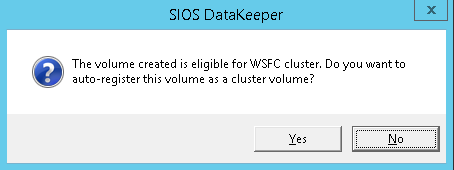


Fig. 41: Press ‘Yes’ to enable the replicated volume as cluster volume

After the creation is finished, the DataKeeper Management Tool lists the replication job as active.

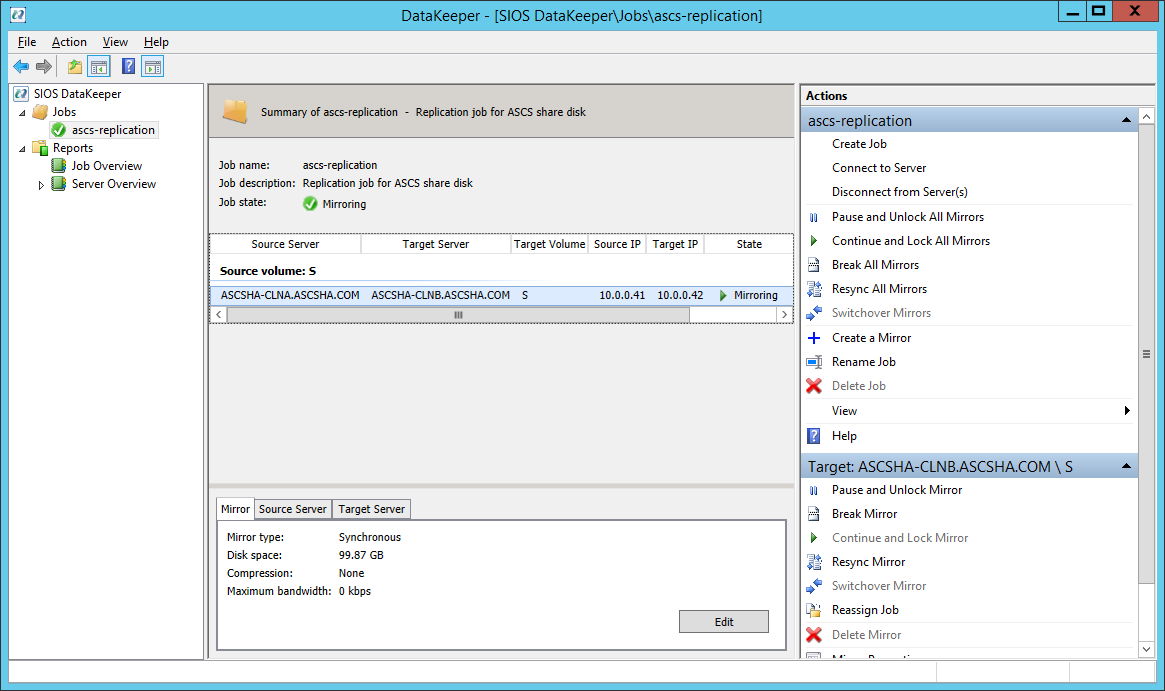
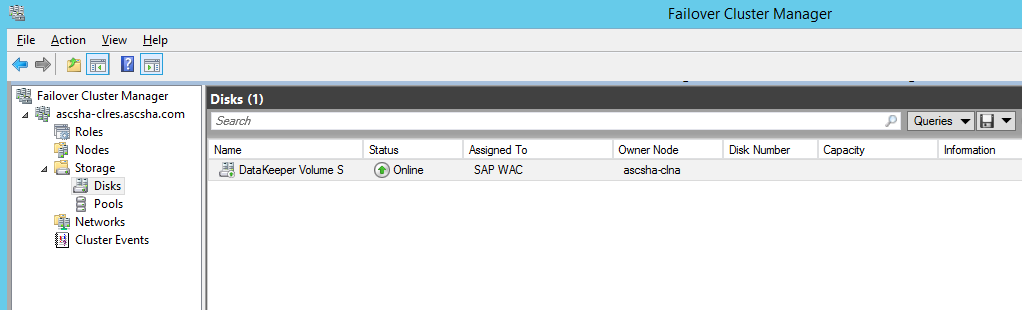


Fig. 42: DataKeeper synchronous mirroring for SAP ASCS/SCS share disk is active

As a result, the disk can now be seen in the Windows Failover Cluster Manager as a DataKeeper Disk as shown below.

  
  
Fig. 43: The replicated disk by DataKeeper is shown in Failover Cluster Manager

At this stage everything is prepared to start installing the SAP ASCS/SCS and the other components of the SAP system like DBMS and Primary Application Server. Some of those steps will be described in the next chapter.

# Installation of SAP NetWeaver System

We won’t describe the setup of the DBMS since setups vary dependent on the DBMS system used. However, we assume that High-Availability concerns with the DBMS are addressed with the functionalities the different DBMS vendors support for Azure. E.g. AlwaysOn or Database Mirroring for SQL Server and Oracle Data Guard for Oracle. In our example scenario we used for this documentation, we did not protect the DBMS additionally. There also are no special considerations with the different DBMS to interact with such a clustered SAP ASCS/SCS configuration on Azure.

Achtung **Important:**

The installation procedure of SAP NetWeaver ABAP systems, Java systems and ABAP+Java systems is almost identical. The biggest difference is that a SAP ABAP system has one ASCS instance. The SAP Java system has one SCS instance and SAP ABAP+Java system one ASCS and one SCS instance running in the same Microsoft failover cluster group. Any installation difference for each SAP NetWeaver installation stack will be explicitly mentioned. All other parts are assumed to be the same.

## SAP Installation with High Available ASCS/SCS Instance

Achtung **Important:** Do NOT place your page file on DataKeeper mirrored volumes, as it is not supported by DataKeeper. You can place your page file on the temporary D:\ drive of a VM or any other drive which is not replicated with SIOS DataKeeper.

### Create Virtual Hostname for clustered SAP ASCS/SCS

A first step is to create the necessary DNS entry for the Virtual Hostname of the ASCS/SCS Instance. The tool used is the Windows DNS Manager. Besides the Virtual hostname, the TCP/IP address assigned to the Virtual Hostname needs to be defined as well.

Achtung **Important: Keep in mind that the TCP/IP address we assign to the Virtual Hostname of the ASCS/SCS instance needs to be the same as the TCP/IP address we will assign to the Azure Load Balancer later on.**

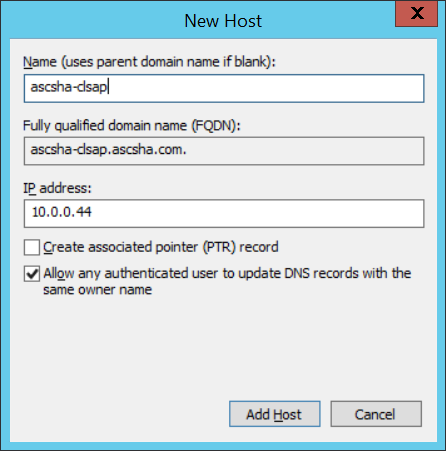


Fig. 44: Defining the DNS entry for the SAP ASCS/SCS cluster virtual name and TCP/IP address

The entry is shown in the DNS manager under the domain as shown in the next figure.

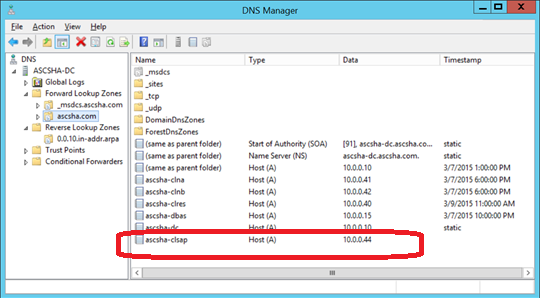


Fig. 45: New virtual name and TCP/IP address listed for SAP ASCS/SCS cluster configuration

### ILB and Endpoints for NetWeaver System

You need to create an Azure Internal Load Balancer, which is associated with *ascsha-clu* cloud service, and create the end points for the SAP ports. You will do this using the Azure PowerShell commands as listed below. A summary of the required SAP and non-SAP ports can be found within the next chapters and at the end of the document.

### Create ILB and Endpoints for NW ABAP ASCS Instance

Following Azure ILB endpoints are needed for SAP NetWeaver ABAP ASCS ports:

|  |  |  |
| --- | --- | --- |
| **Service** | **Default Ports Numbers** | **Concrete ports** for   * **ASCS** instance with instance number **00** * **ERS** with **10** |
| Enqueue Server | 32<InstanceNumber> | 3200 |
| ABAP Message Server | 36<InstanceNumber> | 3600 |
| Internal ABAP Message | 39<InstanceNumber> | 3900 |
| Message Server HTTP | 81<InstanceNumber> | 8100 |
| SAP Start Service ASCS HTTP | 5<InstanceNumber>13 | 50013 |
| SAP Start Service ASCS HTTPS | 5<InstanceNumber>14 | 50014 |
| Enqueue Replication | 5<InstanceNumber>16 | 50016 |
| SAP Start Service ERS HTTP | 5<InstanceNumber>13 | 51013 |
| SAP Start Service ERS HTTPS | 5<InstanceNumber>14 | 51014 |
| Win RM |  | 5985 |
| File Share |  | 445 |

Table 6: Port numbers of SAP NetWeaver ABAP instances

The PowerShell script below is assuming 00 as instance number for the ASCS/SCS instance. If you want to use a different instance number, you will need to change the PowerShell in lines where port numbers are set. **Important: As you can see in the fourth line of the PowerShell script below, that we are using the same TCP/IP address when creating the Load Balancer as we used when assigning the TCP/IP address to the Virtual Hostname.** Means for such a Windows Server Failover Cluster configuration we look at:

**TCP/IP address of Virtual SAP ASCS/SCS Hostname = TCP/IP address of Azure Load Balancer**

Achtung **Important:** This also means that in one Windows Server Failover Cluster on Azure can run only one *SAP <SID>* Failover Cluster role, e.g. for ABAP system one ASCS instance, for Java system one SCS instance , and for ABAP+Java one ASCS and one SCS instance in the same cluster role. Multi-SID clustering as described in SAP Installation Guide (see <http://service.sap.com/instguides> ) does not work in Azure.

$ILBName = "ILBASCSHA"

$SubNetworkName = "Subnet-1"

$SAPASCSCloudService = "ascsha-clu"

$StaticVNetIPAdd = "10.0.0.44"

#Define ASCS and ERS Instance Numbers

$ASCSInstanceNumber = "00"

$ERSInstanceNumber = "10"

#Define Clouster Node A and B, and Probe Port Number

$ClusterNodeA = "ascsha-clna"

$ClusterNodeB = "ascsha-clnb"

$ProbePortNumber ="59999"

$ILBName = "ILBASCSHA"

# Create Internal Load Balancer (ILB)

Add-AzureInternalLoadBalancer -InternalLoadBalancerName $ILBName -StaticVNetIPAddress $StaticVNetIPAdd -SubnetName $SubNetworkName -ServiceName $SAPASCSCloudService

#Check that ILB is created

Get-AzureInternalLoadBalancer -ServiceName $SAPASCSCloudService

#Define Clouster Node A and B, and Probe Port Number

$ClusterNodeA = "ascsha-clna"

$ClusterNodeB = "ascsha-clnb"

$ProbePortNumber ="59999"

#Define Port 32<ASCSInstanceNumber>

$PortNumber = "32" + $ASCSInstanceNumber

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 32<ASCSInstanceNumber> for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 32<ASCSInstanceNumber> for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 36<ASCSInstanceNumber>

$PortNumber = "36" + $ASCSInstanceNumber

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 36<ASCSInstanceNumber> for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 36<ASCSInstanceNumber> for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 39<ASCSInstanceNumber>

$PortNumber = "39" + $ASCSInstanceNumber

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 39<ASCSInstanceNumber> for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 39<ASCSInstanceNumber> for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 81<ASCSInstanceNumber>

$PortNumber = "81" + $ASCSInstanceNumber

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 81<ASCSInstanceNumber> for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 81<ASCSInstanceNumber> for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 5<ASCSInstanceNumber>13

$PortNumber = "5" + $ASCSInstanceNumber + "13"

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 5<ASCSInstanceNumber>13 for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 5<ASCSInstanceNumber>13 for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 5<ASCSInstanceNumber>14

$PortNumber = "5" + $ASCSInstanceNumber + "14"

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 5<ASCSInstanceNumber>14 for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 5<ASCSInstanceNumber>14 for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 5<ASCSInstanceNumber>16

$PortNumber = "5" + $ASCSInstanceNumber + "16"

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 5<ASCSInstanceNumber>16 for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 5<ASCSInstanceNumber>16 for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 5<ERSInstanceNumber>13

$PortNumber = "5" + $ERSInstanceNumber + "13"

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 5<ERSInstanceNumber>13 for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 5<ERSInstanceNumber>13 for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 5<ERSInstanceNumber>14

$PortNumber = "5" + $ERSInstanceNumber + "14"

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 5<ERSInstanceNumber>14 for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 5<ERSInstanceNumber>14 for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 5985

$PortNumber = "5985"

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 5985 for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 5985 for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

#Define Port 445

$PortNumber = "445"

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 445 for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 445 for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

### Create ILB and Endpoints for NW Java SCS Instance

The following Azure ILB endpoints are needed for SAP NetWeaver Java SCS ports:

|  |  |  |
| --- | --- | --- |
| **Service** | **Default Ports Numbers** | **Concrete ports** for   * **SCS** instance with instance number **00** * **ERS** with **10** |
| Enqueue Server | 32<InstanceNumber> | 3200 |
| Gateway Server | 33<InstanceNumber> | 3300 |
| Java Message Server | 39<InstanceNumber> | 3900 |
| Message Server HTTP | 81<InstanceNumber> | 8100 |
| SAP Start Service SCS HTTP | 5<InstanceNumber>13 | 50013 |
| SAP Start Service SCS HTTPS | 5<InstanceNumber>14 | 50014 |
| Enqueue Replication | 5<InstanceNumber>16 | 50016 |
| SAP Start Service ERS HTTP | 5<InstanceNumber>13 | 51013 |
| SAP Start Service ERS HTTPS | 5<InstanceNumber>14 | 51014 |
| Win RM |  | 5985 |
| File Share |  | 445 |

Table 7: Port numbers of SAP NetWeaver Java instances

As Java SCS instance ports are very similar to ABAP ASCS instance, to create ILB end point, execute above PowerShell code for ASCS instance (just adapt Java SCS instance number) as specified in [section 7.1.3](#_Create_ILB_and) and execute in addition below PowerShell code which creates SAP Java Gateway Server end point.

#Addition PowerShell code for NetWeaver Java only

#Define Port 33<ASCSInstanceNumber>

$PortNumber = "33" + $ASCSInstanceNumber

$PortName = "FSEP$PortNumber"

$LoadBalanceSetName = "ILBASCSHA$PortNumber"

# Add End Point for port 33<SCSInstanceNumber> for Cluster Node A

Write-Host "Adding end point to '$ClusterNodeA' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeA | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

# Add End Point for port 33<SCSInstanceNumber> for Cluster Node B

Write-Host "Adding end point to '$ClusterNodeB' cluster node end point with following parameters: Port Number:'$PortNumber', Port Name:'$PortName', Load Balance Set Name:'$LoadBalanceSetName' ..."

Get-AzureVM -ServiceName $SAPASCSCloudService -Name $ClusterNodeB | Add-AzureEndpoint -Name $PortName -LBSetName $LoadBalanceSetName -Protocol tcp -LocalPort $PortNumber -PublicPort $PortNumber -ProbePort $ProbePortNumber -ProbeProtocol tcp -ProbeIntervalInSeconds 10 -InternalLoadBalancerName $ILBName -IdleTimeoutInMinutes 30 | Update-AzureVM

### Create ILB and Endpoints for NW ABAP+Java ASCS and SCS Instance

Some SAP products are still using the dual stack configuration which requires you to run an SAP NetWeaver System that has one ASCS and one SCS instance which run in same *SAP <SID>* Windows Failover Cluster role.

The configuration procedure is following:

* Create ILB and ASCS endpoints using scripts (pass ASCS instance number) as described above in [section 7.1.3](#_Create_ILB_and)
* Create in addition SCS endpoints using scripts (pass SCS instance number) as described above in [section 7.1.4](#_Toc422819169)

As you run both instances on the same Windows Failover Cluster group, the ASCS and SCS instance will have different instance numbers and hence get different ports assigned.

### Install SAP First Cluster Node

The installation of the first ASCS/SCS cluster node does not differ in any way from the way it is documented in the SAP Installation documentation by:

* Execute the *First Cluster Node* option on cluster node A, e.g. on *ascsha-clna* host as in our example.

After this step finished, you need to perform a few steps which are not described in the usual SAP installation documentation.

### Modify the SAP profile of the ASCS/SCS instance

A new profile parameter needs to be added. This profile parameter avoids that connections between SAP work processes and the enqueue server are closed when they are idle for too long. We mentioned the problem scenario in [section 5.10.1](#_Add_Registry_Entries) of this document. In that section we introduced two changes to some basic TCP/IP connection parameters as well. In a second step we need to configure the enqueue server to send a keep\_alive signal so that the connections do not hit the idle threshold of the Azure ILB. For this purpose add this profile parameter:

enque/encni/set\_so\_keepalive = 1

to the SAP ASCS instance profile. In our example, the path is: <ShareDisk>:\usr\sap\WAC\SYS\profile\WAC\_ASCS00\_ascsha\_clsap

e.g. to the SAP SCS instance profile and corresponding path   
<ShareDisk>:\usr\sap\WAC\SYS\profile\WAC\_SCS00\_ascsha\_clsap

### Add Probe Port

In order to make the whole Cluster configuration work with an Azure Load Balancer, we need to leverage the probe functionality of the Internal Load Balancer. An Azure Internal Load Balancer usually balances and distributes the incoming workload between the different instances within the Azure Cloud Service it is part of. However, that would not work in such a cluster configuration since only one of the instances is active and the other is passive and can’t accept workload. In order to enable configurations where the Azure Internal Load Balancer will assign work to the active instance(s) only, a probe functionality got established. Through that functionality the Internal Load Balancer has the possibility to check which of the instances are active and subsequently target only that instance with the workload.

First let’s check the current *ProbePort* setting with this PowerShell command

Get-ClusterResource „SAP WAC IP" | Get-ClusterParameter

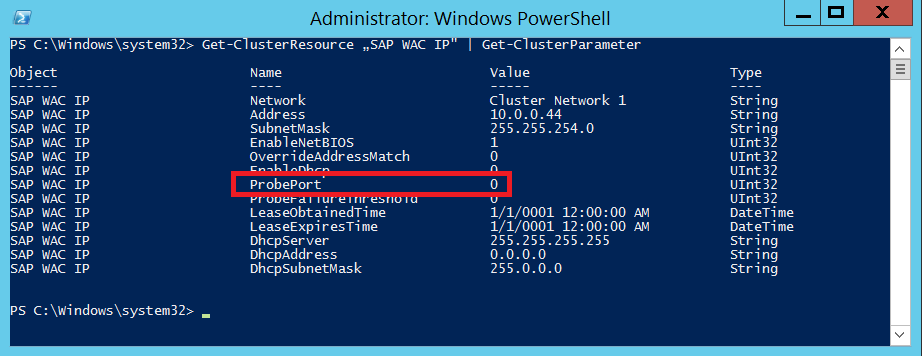


Fig. 46: Probe port of Cluster configuration is 0 by default

By default, the probe port number is set to 0. In order to make the configuration work, a port needs to be defined. In our case we chose port 59999. Assigning that port number can be done with the two commands below:

First get the SAP virtual host name cluster resource *SAP WAC IP*

$var = Get-ClusterResource | Where-Object { $\_.name -eq "SAP WAC IP" }

And then set the probe port to 59999

$var | Set-ClusterParameter -Multiple @{"Address"="10.0.0.44";"ProbePort"=59999;"Subnetmask"="255.255.254.0";"Network"="Cluster Network 1";"OverrideAddressMatch"=1;"EnableDhcp"=0}

You need to stop and start the *SAP WAC* cluster role in order to activate the changes.

After bringing the *SAP WAC* cluster role online, check that *ProbePort* is set to new value:

Get-ClusterResource | Where-Object { $\_.name -eq "SAP WAC IP" } | Get-ClusterParameter

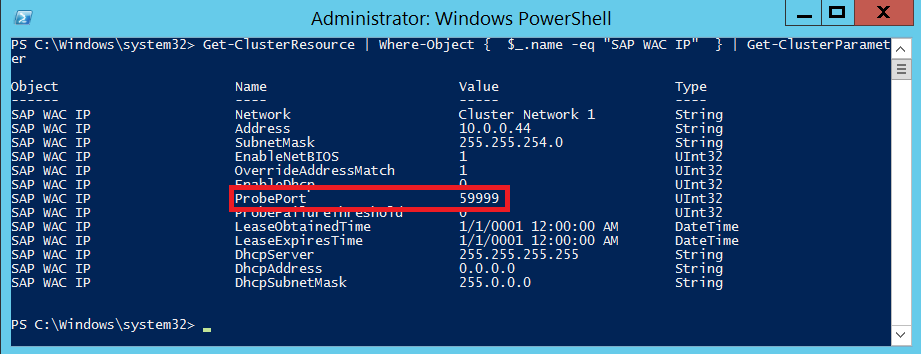


Fig. 47: Probe port of Cluster after change

You can see that the*ProbePort* is now set to 59999. Now you are able to access the file share [\\ascsha-clsap\sapmnt](file://ascsha-clsap/sapmnt) from other hosts like *ascsha-dbas*.

### Installing the Database Instance

Installing the database instance is in no way different from the process as descried in the SAP installation documentation. Therefore, it is not going to be documented in more detail here.

### Installation Second Cluster Node

Again the installation of the second cluster nodes does not differ from what is described in the SAP installation documentation. Therefore, we are not going into details in this section.

### Change Windows Service Startup Type of SAP ERS Instance

Change on both cluster node Windows service type of SAP ERS Instance to ***Automatic (Delayed Start)***

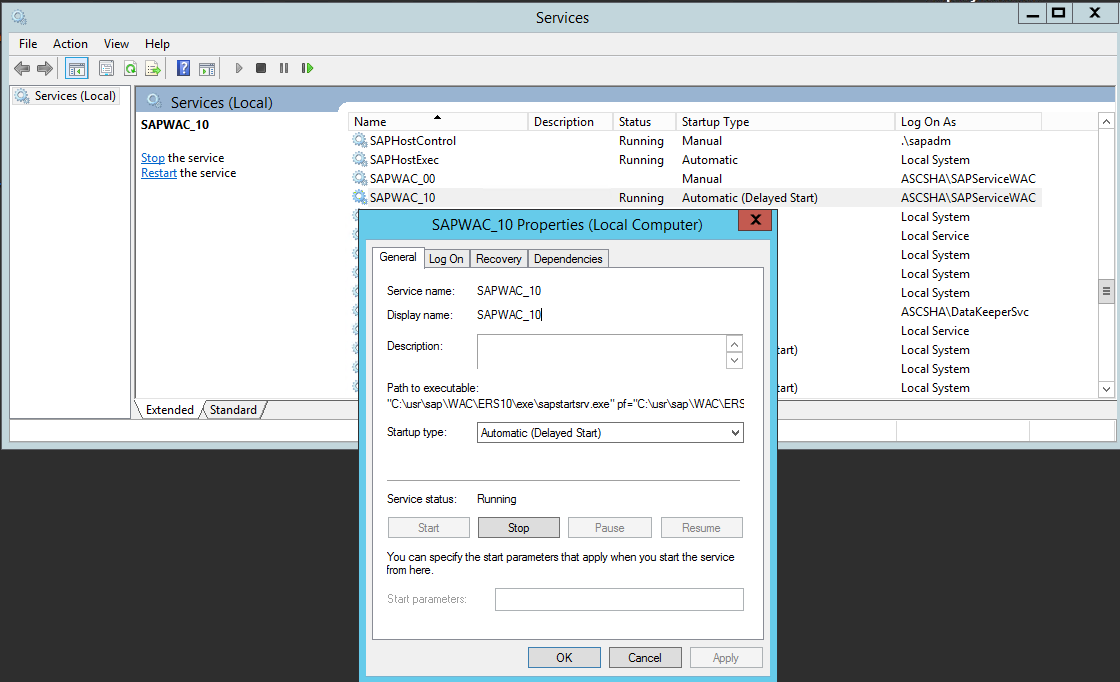


Fig. 48: Change Service type for SAP ERS instance to delayed automatic

### Installation of SAP Primary Application Server (PAS)

Execute the Primary Application Server Instance installation on the VM designated for hosting the PAS. In our case, this is the *ascsha-dbas* VM. Again the installation procedure is the same as described in the SAP installation documentation. There are no dependencies to Azure or DataKeeper specifics.

Following all these steps, you created a SAP ASCS/SCS setup in conjunction with SIOS DataKeeper in Azure.

# Testing SAP ASCS/SCS Instance Failover and SIOS Replication

You can easily test and monitor SAP ASCS/SCS instance failover and SIOS disk replication using *Failover Cluster Manager* and *SIOS DataKeeper* UI.

## Starting point – SAP ASCS/SCS instance is running on Cluster Node A

The *SAP WAC* cluster group (e.g. SAP ASCS/SCS instance) is running on cluster node A, e.g. on *ascsha-clna* node. The shared disk *S:* that is part of *SAP WAC* cluster group and used by the ASCS/SCS instance, is assigned to cluster node A.

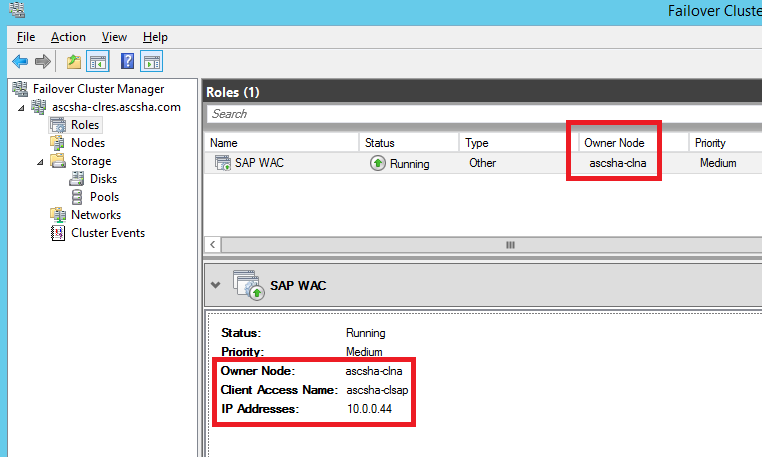


Fig. 49: Failover Cluster Manager: *SAP <SID>* cluster group is running on cluster Node A

Using the*SIOS DataKeeper* UI, you can see that the shared disk data is synchronously replicated from source volume *S:* on cluster node A (e.g. *ascsha-clna [10.0.0.41]*) to target volume *S:* on cluster node B (e.g. *ascsha-clnb [10.0.0.42]*)

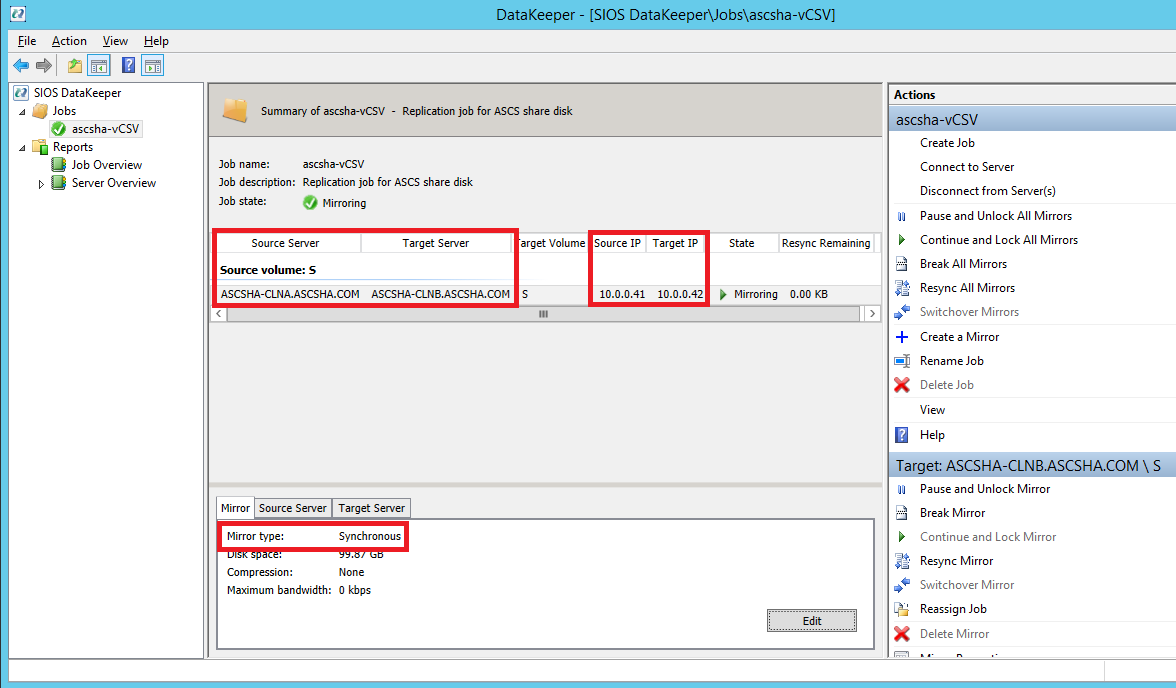


Fig. 50: SIOS DataKeeper: Replicating local volume from cluster node A to cluster node B

## Failover process from Node A to Node B

You can initiate a failover of the *SAP <SID>* cluster group from cluster node A to cluster node B:

* using Failover Cluster Manager
* using Failover Cluster PowerShell

Move-ClusterGroup -Name "SAP WAC"

* restarting cluster node A within the Windows guest OS

(This will initiate an automatic failover of *SAP <SID>* cluster group from Node A to Node B)

* restarting cluster node A from the Azure portal

(This will initiate an automatic failover of *SAP <SID>* cluster group from Node A to Node B)

* restarting cluster node A using Azure PowerShell

(This will initiate an automatic failover of *SAP <SID>* cluster group from Node A to Node B)

Restart-AzureVM -Name ascsha-clna -ServiceName ascsha-clu

## End point – SAP ASCS/SCS instance is running on Cluster Node B

After failing over, the SAP <SID> cluster group is running on cluster node B, e.g. on *ascsha-clnb*.

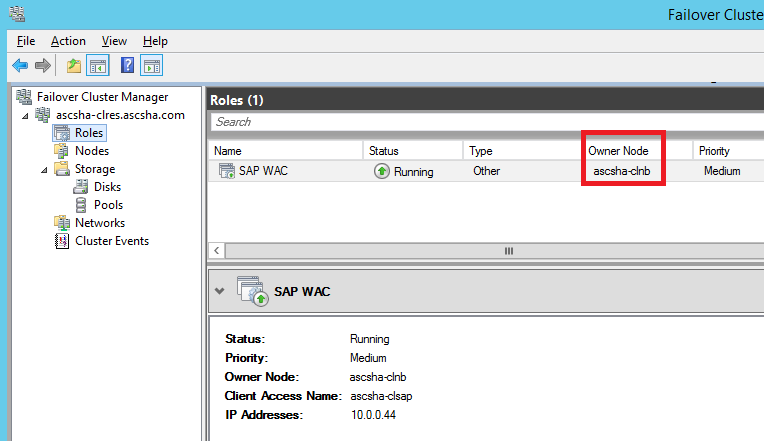


Fig. 51: Failover Cluster Manager: *SAP <SID>* cluster group is running on cluster Node B

The shared disk is now mounted to cluster node B. SIOS DataKeeper is replicating data from source volume *S:* on cluster node B (e.g. *ascsha-clnb [10.0.0.42]*) to target volume *S:* on cluster node A (e.g. *ascsha-clna [10.0.0.41]*).

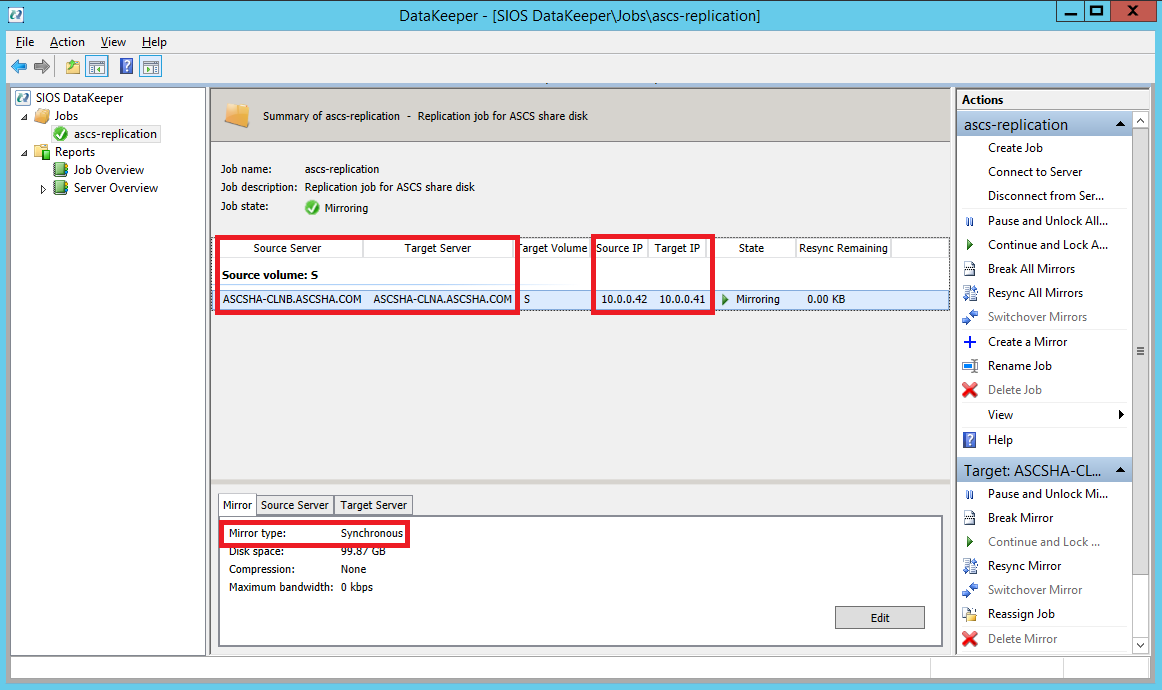


Fig. 52: SIOS DataKeeper: Replicating local volume from cluster node B to cluster node A

# Links

**Install Azure Modules on PowerShell**

<http://azure.microsoft.com/en-us/documentation/articles/install-configure-powershell/>

**Additional Information about Cluster Quorum Modes**

<http://technet.microsoft.com/de-de/library/cc770830(v=ws.10).aspx>

**Running SAP Applications on the Microsoft Platform Blog**

<http://blogs.msdn.com/b/saponsqlserver/>

**White Papers: Using SAP on Azure Virtual Machines (VMs)**

<https://msdn.microsoft.com/en-us/library/azure/dn745892.aspx>

**Azure Internal Load Balancer**

<http://azure.microsoft.com/blog/2014/05/20/internal-load-balancing/>

<http://blogs.msdn.com/b/brunoterkaly/archive/2014/07/23/fundamentals-of-azure-internal-load-balancers-ilbs.aspx>

<http://azure.microsoft.com/blog/2014/08/14/new-configurable-idle-timeout-for-azure-load-balancer/>

**DataKeeper Cluster Edition**

<http://us.sios.com/products/datakeeper-cluster/>

**Support of Windows Server Failover Cluster on Windows Azure Cloud**

<http://support.microsoft.com/en-us/kb/2721672>

**SIOS DataKeeper Cluster Edition Certification on Windows Azure Cloud**

<http://azure.microsoft.com/en-us/marketplace/partners/sios-datakeeper/sios-datakeeper-8-bring-your-own-license/>

**TCP/IP Ports Used by SAP Applications**

<http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/4e515a43-0e01-0010-2da1-9bcc452c280b?QuickLink=index&overridelayout=true&42472931642836>