



BREAKING NEWS

PODCAST ASF 025: Rob Farley interview 0

NEWS Last Week Reading (2019-11-10) 0

NEWS Last V

 December 12, 2018 Comments 1 Michal Sadowski 3428 Views

Preparation for SQL Server installation

It is a first post on series related to installation and configuration of SQL Server installations.

This series is for professionals who starts their journey with SQL Server administration and also for those who want to extend and structure their knowledge on SQL Server administration.

Preparations

We are starting with following assumptions:

1. It is brand new installation on standalone server
2. We are using Windows Server as an operating system
3. Server is up to date with all security updates from Windows Update

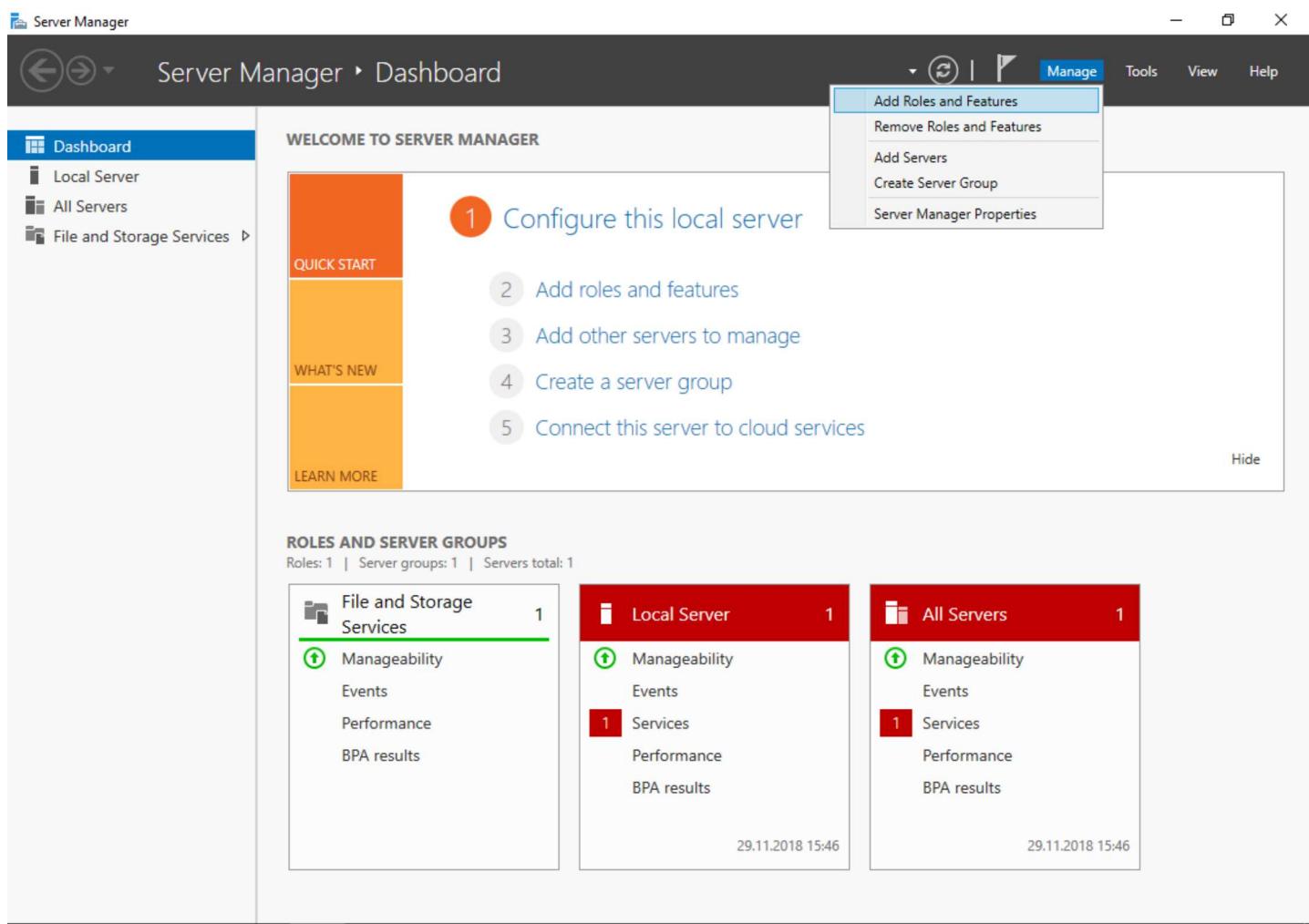
4. Database server is a part of Active Directory domain
5. Active Directory schema is at least the Windows Server 2012 domain functional level
6. All drives used for SQL Server are configured as independent LUNs
7. Latest publicly available version of SQL Server will be used.

As a first step we need to check if all prerequisites for the installation are met. Most important is to have .NET framework installed and server is restarted before installation.

For installation of .NET framework, we need to have ISO file of the Windows Server.

How to install .NET Framework in Windows Server environment?

As a first step start Server Manager and go to **Manage -> Add Roles and Features**



On **Before you begin screen** click **Next**

 Add Roles and Features Wizard

Before you begin

DESTINATION SERVER
WIN-T3USEAPOE06

Before You Begin

- Installation Type
- Server Selection
- Server Roles
- Features
- Confirmation
- Results

This wizard helps you [install](#) roles, role services, or features. You determine which roles, role services, or features to [install](#) based on the computing needs of your organization, such as sharing documents, or hosting a website.

To remove roles, role services, or features:
[Start the Remove Roles and Features Wizard](#)

Before you continue, verify that the following tasks have been completed:

- The Administrator account has a strong password
- Network settings, such as static IP addresses, are configured
- The most current security updates from Windows Update are installed

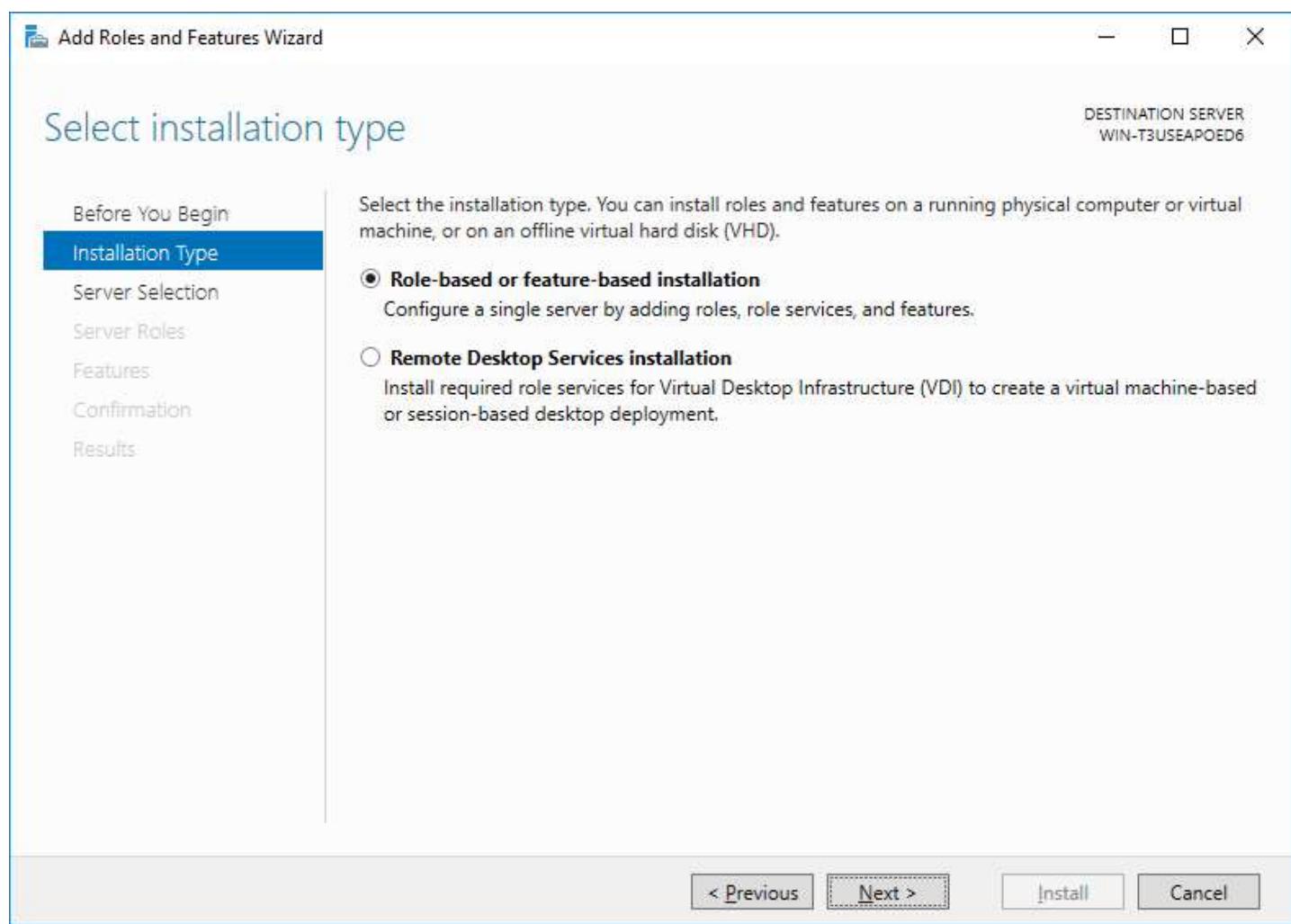
If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.

To continue, click Next.

[Skip this page by default](#)

[< Previous](#) Next > Install Cancel

On **Select installation type** screen select **Role-based or feature-based installation** and click **Next**



On **Select destination server** choose **Select a server from server pool** and select the server and click **Next**

Add Roles and Features Wizard

Select destination server

DESTINATION SERVER
SQL.datacommunity.local

Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results

Select a server or a virtual hard disk on which to install roles and features.

Select a server from the server pool
 Select a virtual hard disk

Server Pool

Filter:

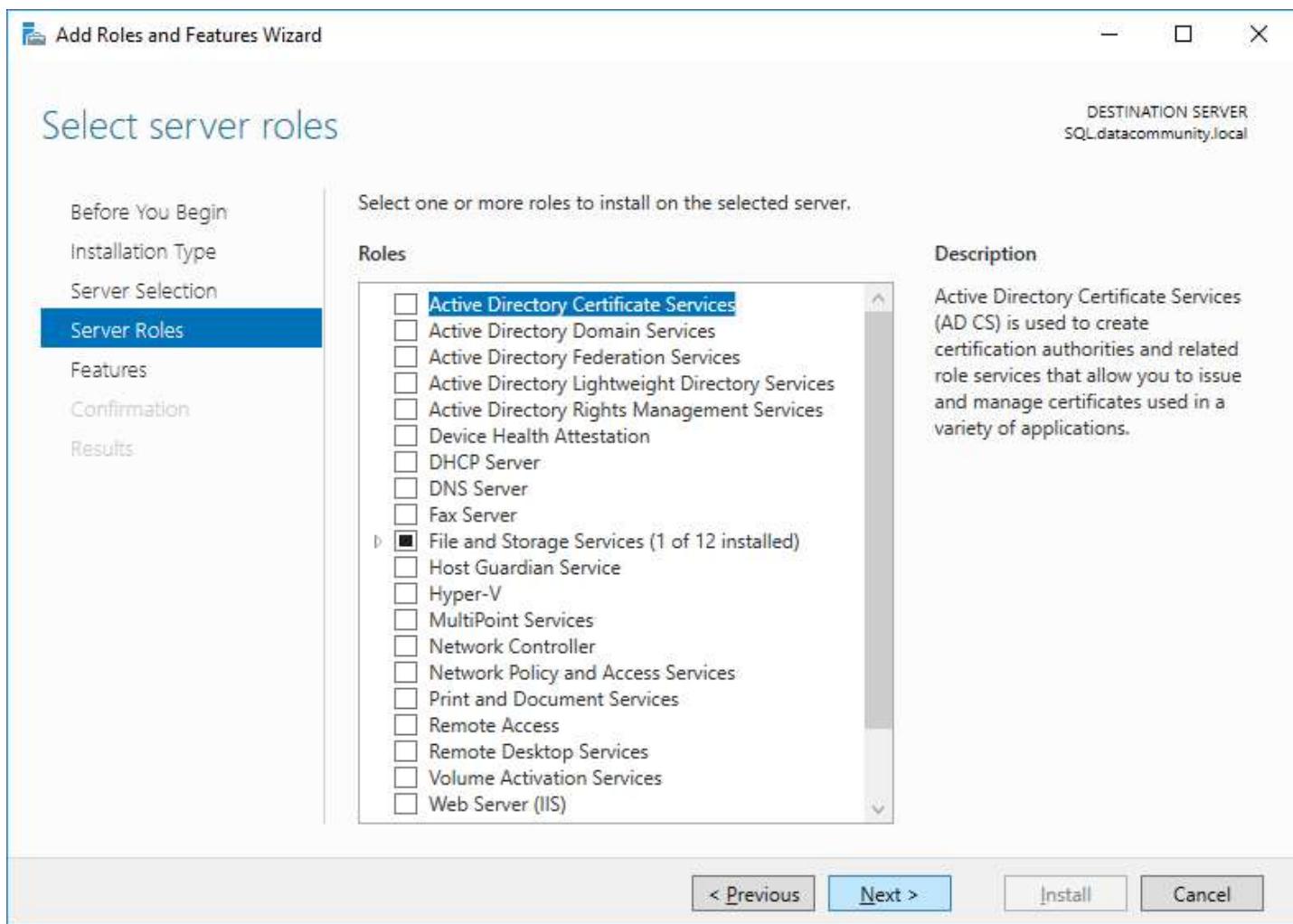
Name	IP Address	Operating System
SQL.datacommunity.local	10.20.30.23	Microsoft Windows Server 2016 Datacenter

1 Computer(s) found

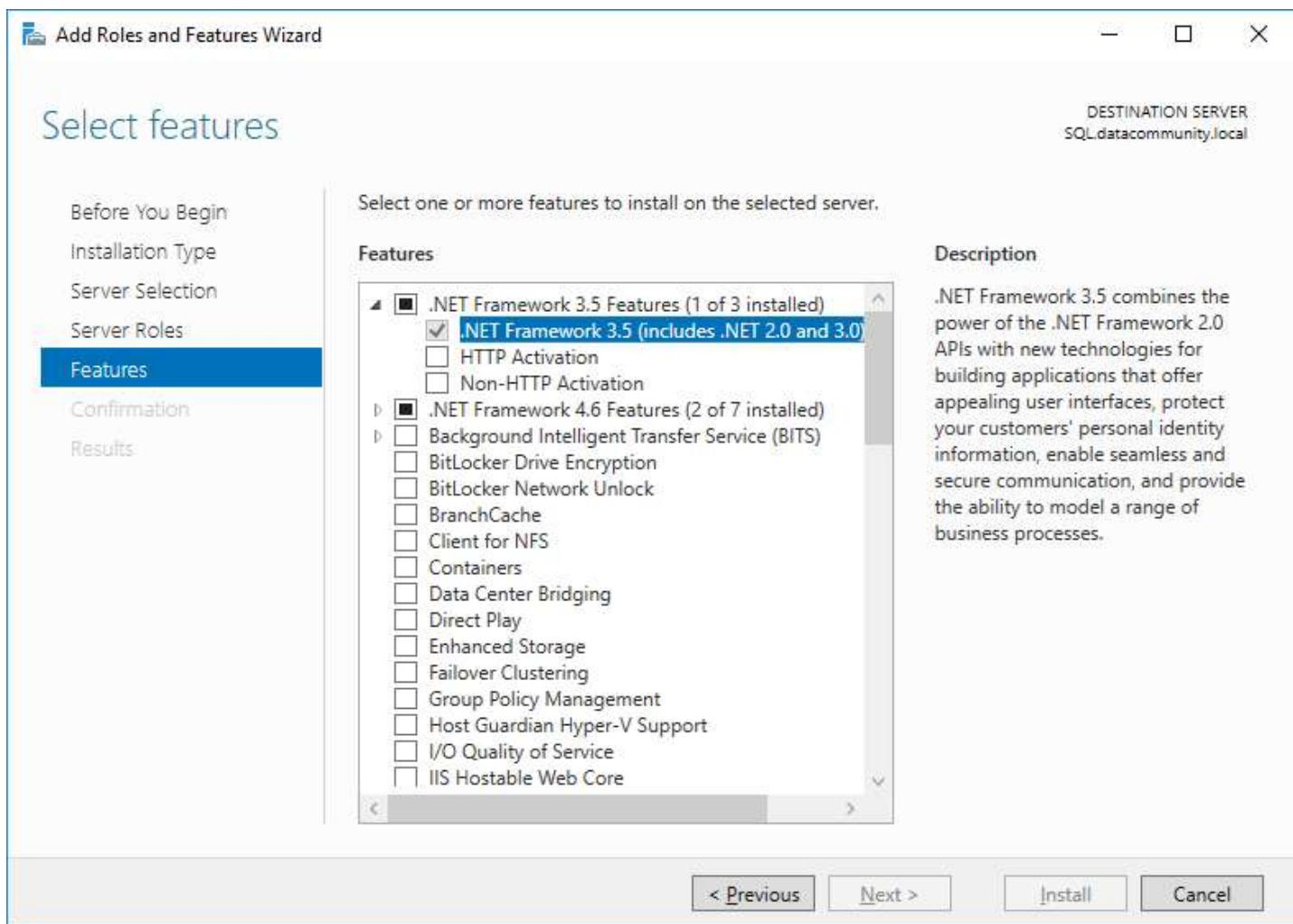
This page shows servers that are running Windows Server 2012 or a newer release of Windows Server, and that have been added by using the Add Servers command in Server Manager. Offline servers and newly-added servers from which data collection is still incomplete are not shown.

< Previous [Next >](#) [Install](#) Cancel

On **Select server roles** click **Next**



On **Select features** screen expand **.NET Framework 3.5 Features** and select **.NET Framework 3.5 (includes .NET 2.0 and 3.0)** and click **Next**



Click on **Install** to proceed with installation.

You can also check if .NET is installed using PowerShell command:

```
1
2 Get-WindowsFeature -Name Net-Framework-Core
3
```

Display Name	Name	Install State
[X] .NET Framework 3.5 (includes .NET 2.0 and 3.0)	NET-Framework-Core	Installed

Configuration of the disks for SQL Server

Once .NET is installed we can proceed to configuration of separate drives for usage by SQL Server – it is common practice due to mainly performance advantages of this setup.

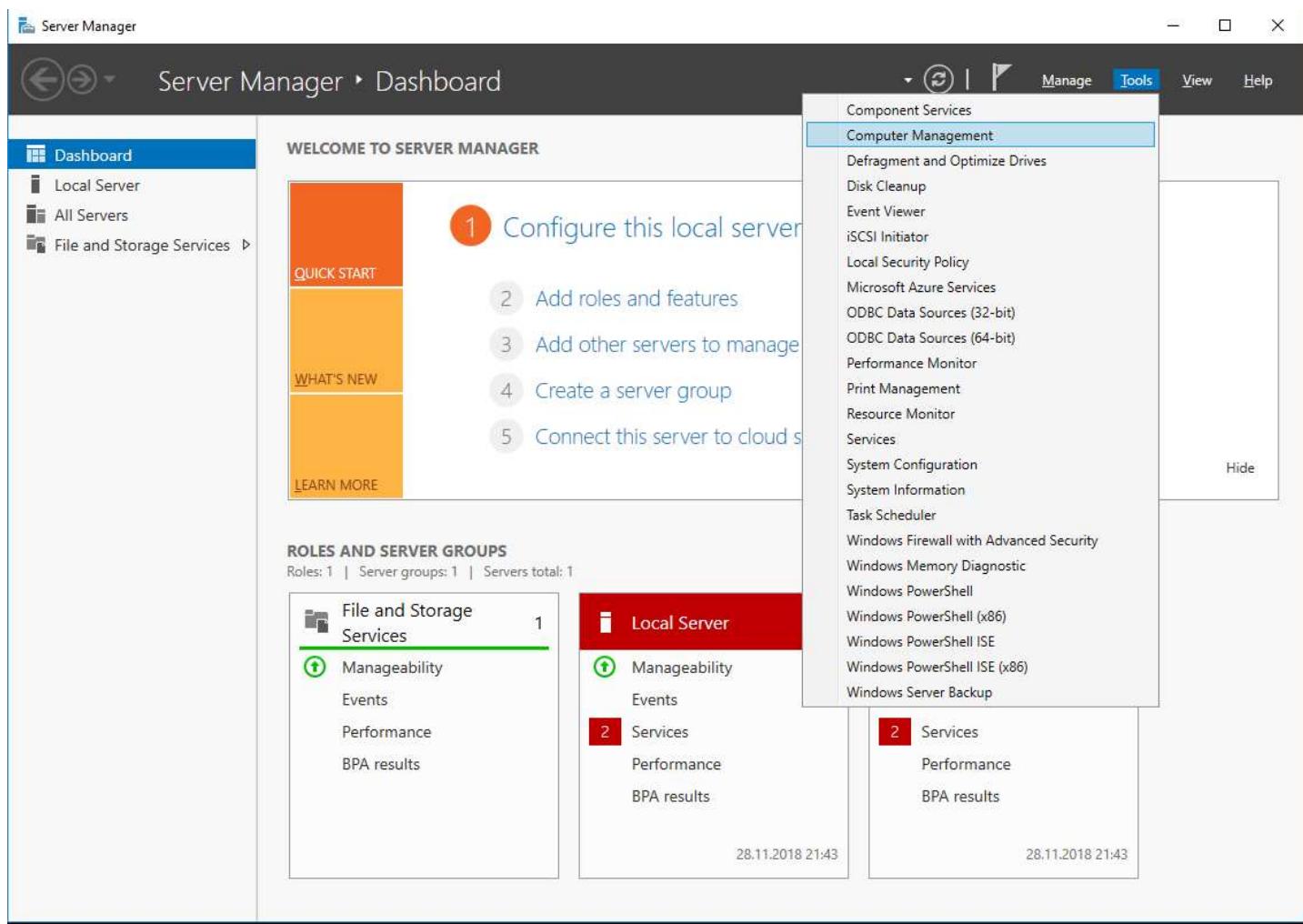
In my lab environment, I will be using following drives:

- D:\ for binaries of SQL Server and system databases
- 3 drives for Data files of user databases configured as mount points on D:\ drive
- 1 drive for Transaction Log of user databases configured as mount points on D:\ drive
- 4 drives for tempdb configured as mount points on D:\ drive – in my lab I am using virtual machine with 4 cores thus 4 separate disks should be used. As a starting point, you should start with 1 tempdb data file per CPU core up to 8 files
- 1 drive for backups configured as mount points on D:\ drive

Before we start with installation, it is required to format drives with desired settings. You may wonder what can be configured with configuration? Yes, there are some settings that should be adjusted for drives used by SQL Server:

- File system – SQL Server support NTFS and ReFS
- Allocation unit – should be changed from default (512 bytes) set to 64k for drives used for data
- Deselect “*Perform quick format*” option – this option will force to format physical storage, avoid dynamic allocation. This will minimize random performance hiccups during normal operation.

As a first drive will be used, I will configure it as a normal drive with D:\ drive letter. Below step-by-step screen shots how to configure the drive.



Browse to Computer Management (Local) -> Storage -> Disk Management and right-click on first disk

Volume	Layout	Type	File System	Status	Capacity	Free Space	% Free
---	Simple	Basic		Healthy (Recovery Partition)	450 MB	450 MB	100 %
---	Simple	Basic		Healthy (EFI System Partition)	99 MB	99 MB	100 %
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	127,45 GB	112,69 GB	88 %

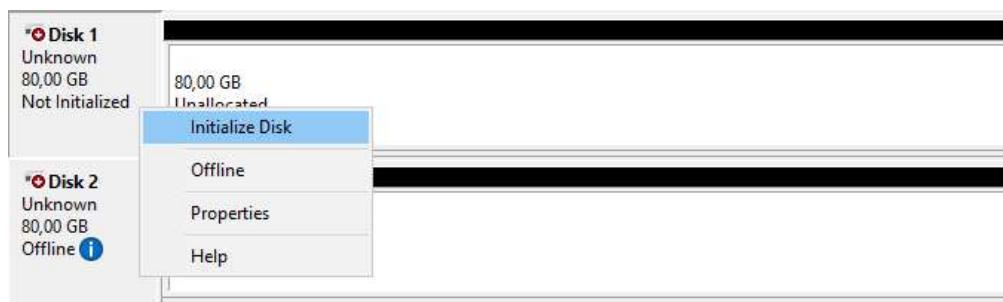
Disk	Capacity	Status	File System
Disk 0	127,98 GB	Online	450 MB Healthy (Recovery Partition) 99 MB Healthy (EFI System) (C) 127,45 GB NTFS Healthy (Boot, Page File, Crash Dump, Primary Partition)
Disk 1	80,00 GB	Offline	80,00 GB Unallocated
Disk 2	80,00 GB	Offline	80,00 GB Unallocated
Disk 3	80,00 GB	Offline	80,00 GB Unallocated
Disk 4	60,00 GB	Offline	60,00 GB Unallocated
Disk 5	60,00 GB	Offline	60,00 GB Unallocated
Disk 6	40,00 GB	Offline	40,00 GB Unallocated
Disk 7	...		

Actions

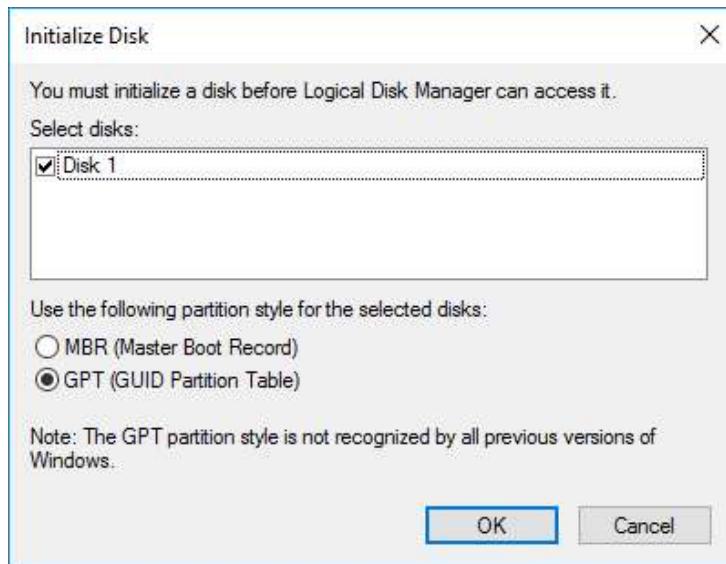
- Disk Management
- More Actions

Select **Online** option

Once disk is switched from *Offline* state to *Not Initialized* right-click and select **Initialize Disk**



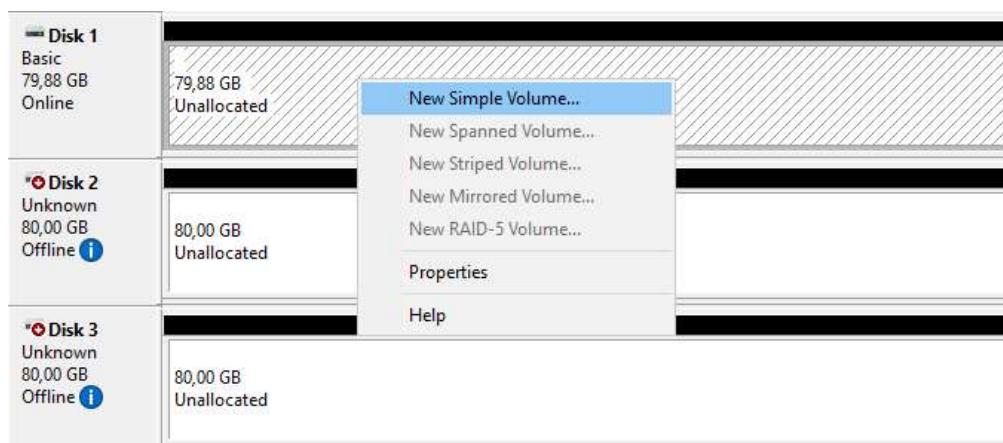
On **Initialize Disk** pop-up select all disks and use **GPT (GUID Partition Table)** option



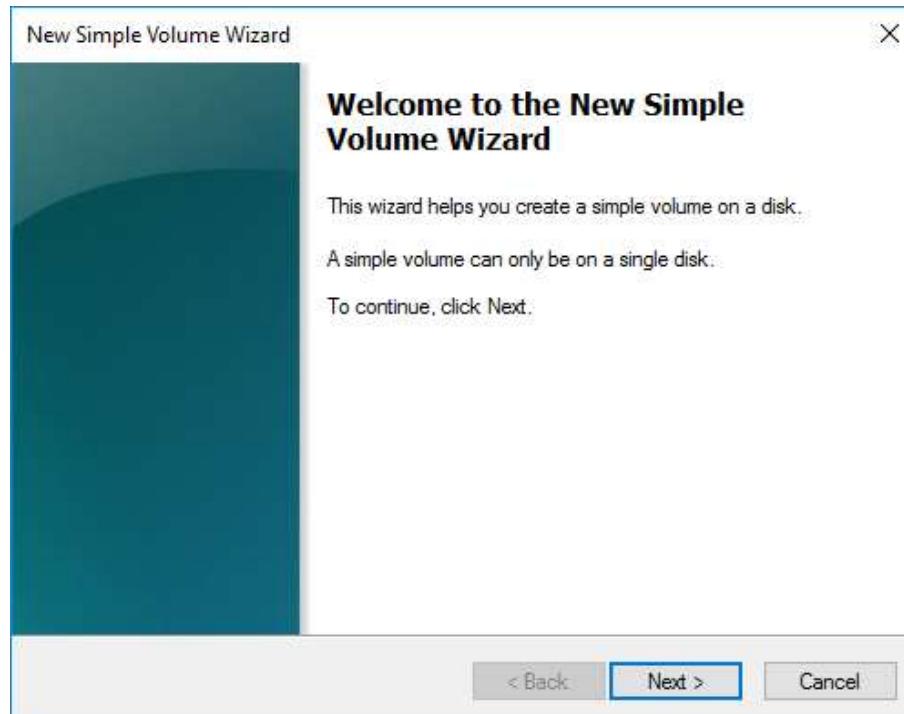
Disk should change its state to *Online*



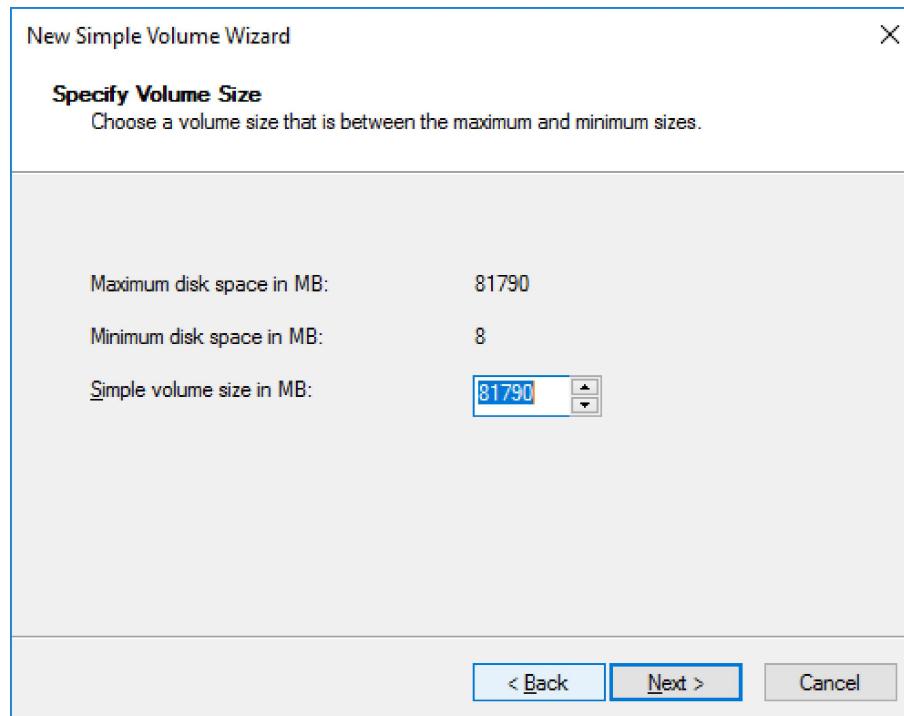
Right-click Unallocated space and select **New Simple Volume...**



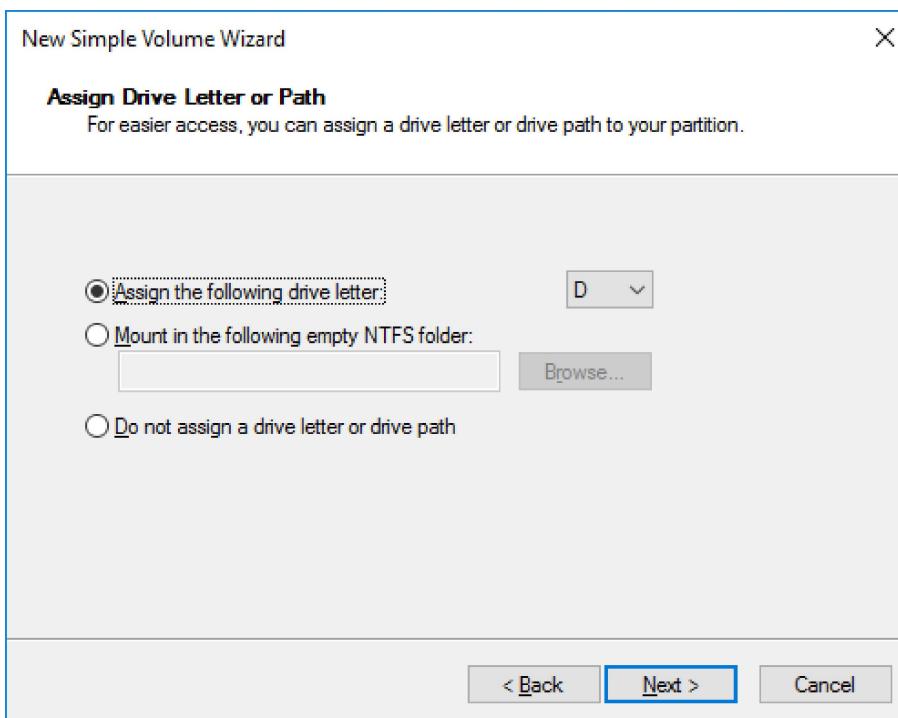
On the **New Simple Volume Wizard** click **Next**



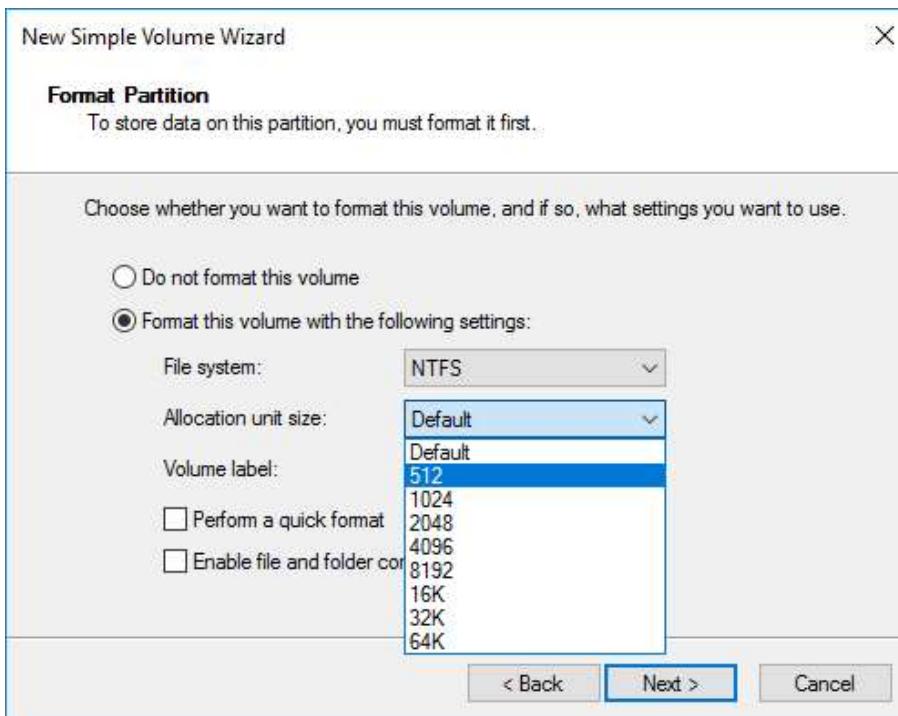
On the **Specify Volume Size** provide size of the volume and click **Next**



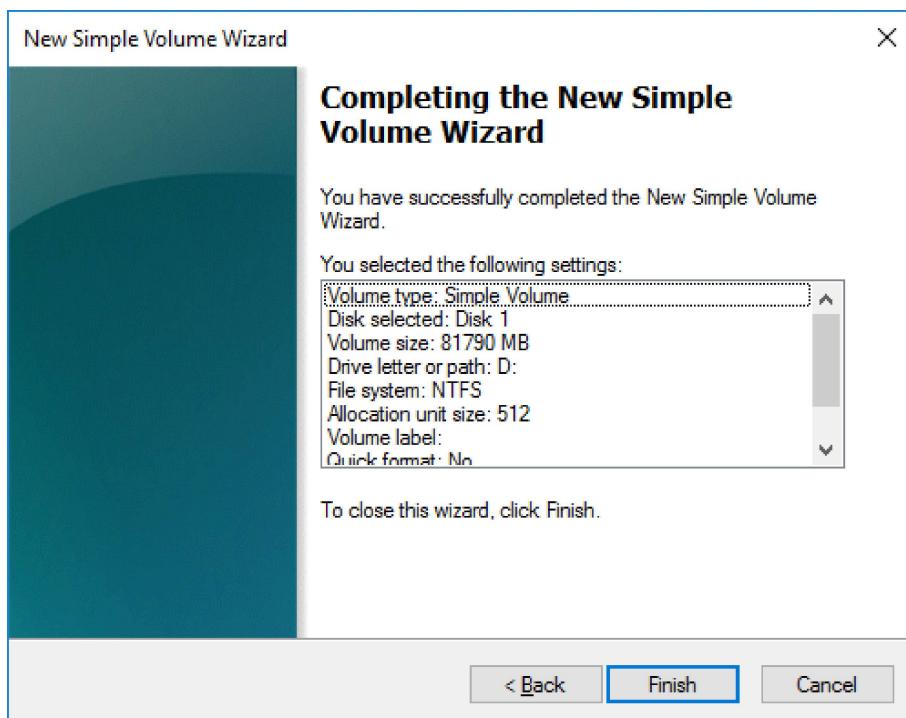
On the **Assign Drive Letter or Path** select D drive and click **Next**



On **Format Partition** select **NTFS** file system and correct allocation unit – for all drives used for storing users data files, allocation unit should be set to 64K. Here we are formatting drive for binaries of SQL Server thus using default 512. Deselect *Perform a quick format* option.



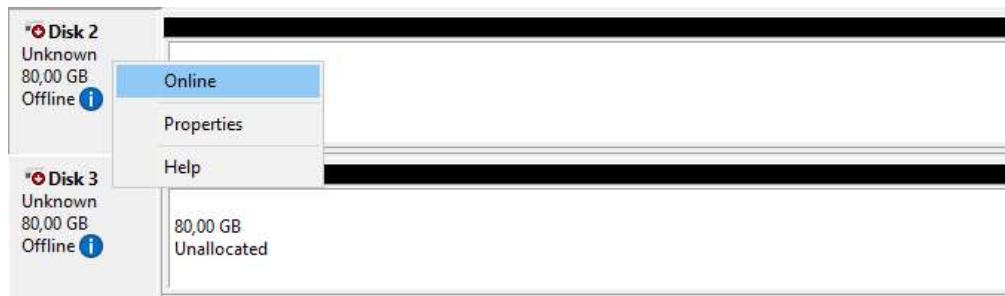
On the summary screen click on **Finish** to start formatting of the drive.



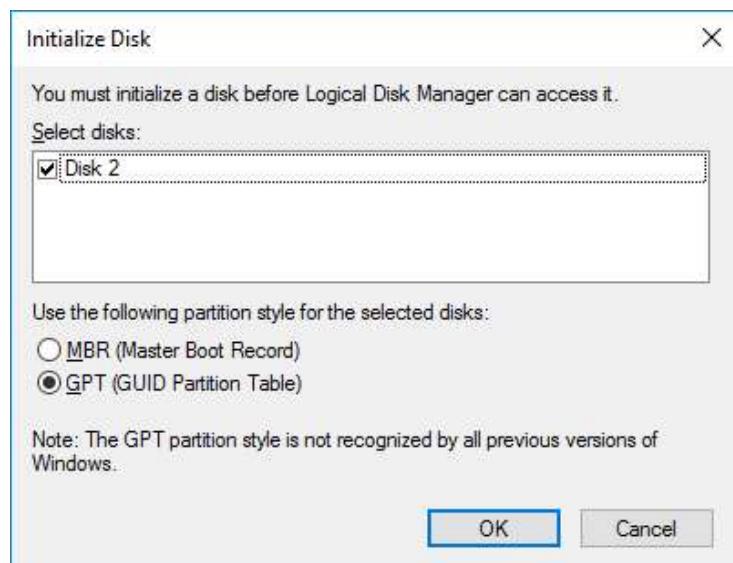
Next drives will be configured as a mount points on the D:\ drive. I am using mount points due to fact that it is not commonly used and can help us with having too many drives in the system (situation when all drives letters are used).

Below step-by-step screen shots how to configure the drive as a mount point.

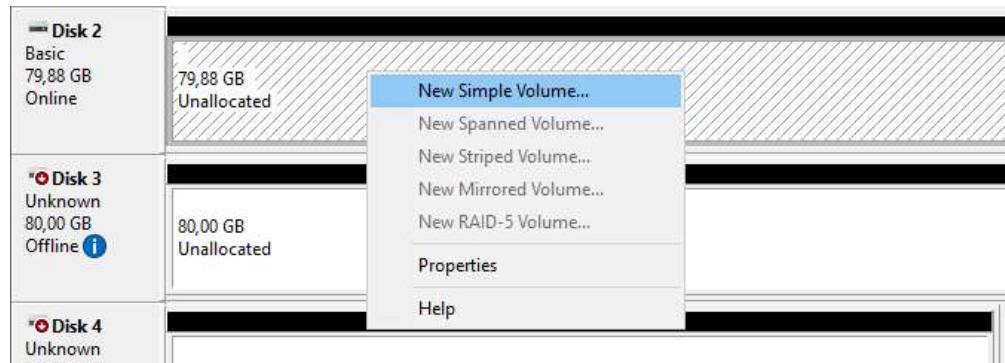
Right click on Disk 2 and select **Online**



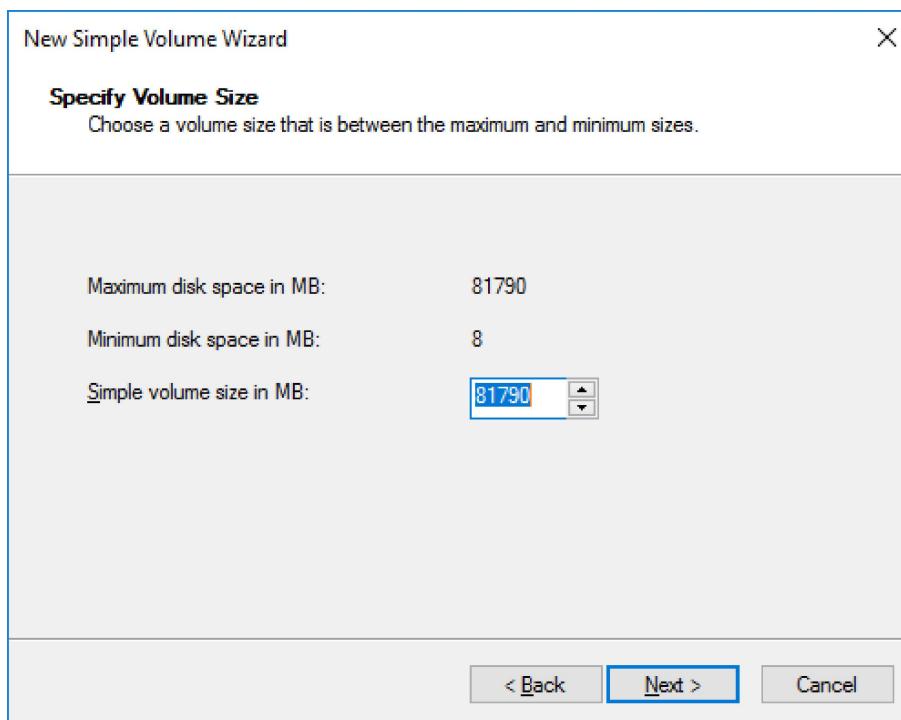
On **Initialize Disk** pop-up select all disks and use **GPT (GUID Partition Table)** option



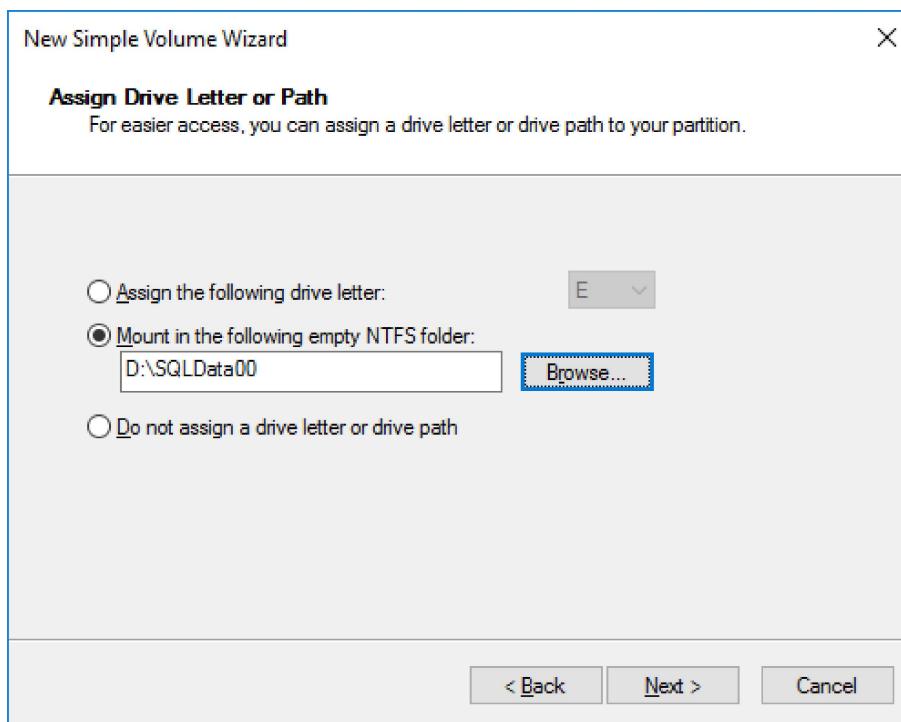
Right-click Unallocated space and select **New Simple Volume...**



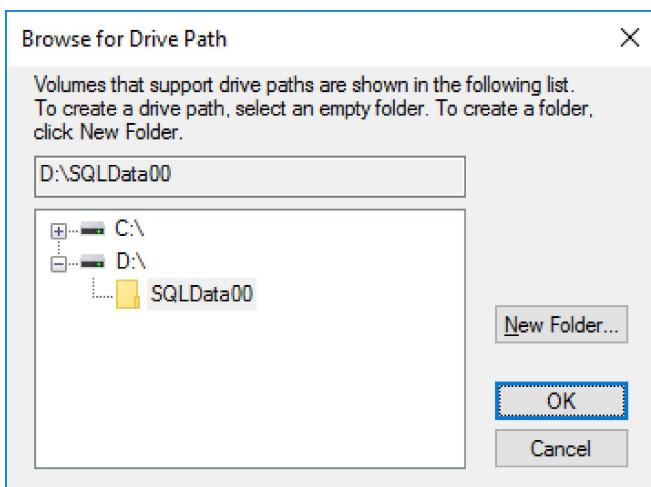
On the **Specify Volume Size** provide size of the volume and click **Next**



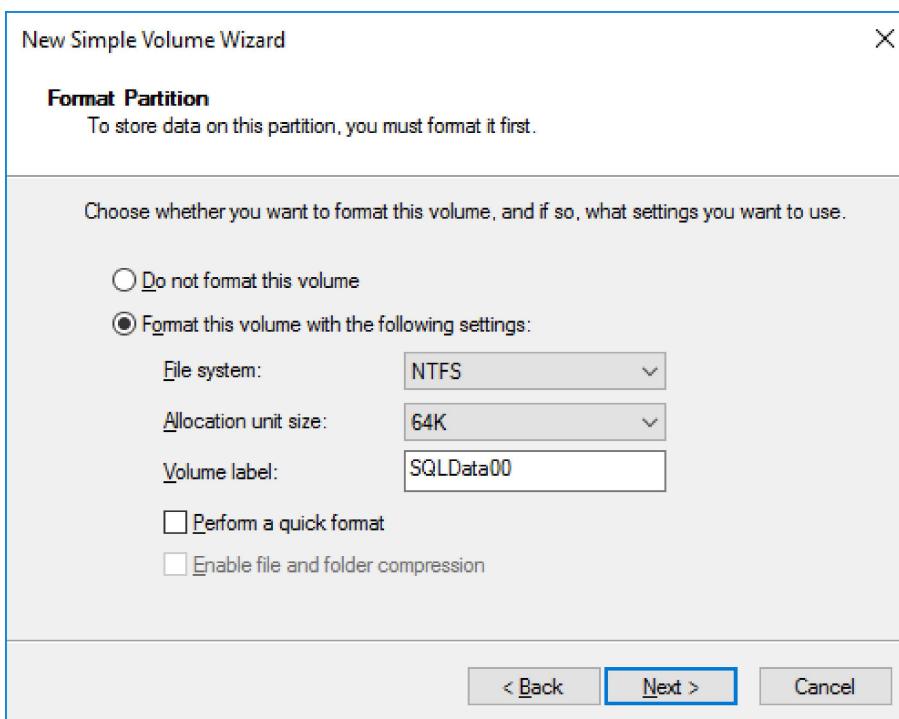
On the **Assign Drive Letter or Path** select **Mount in the following empty NTFS folder** and click **Browse...**



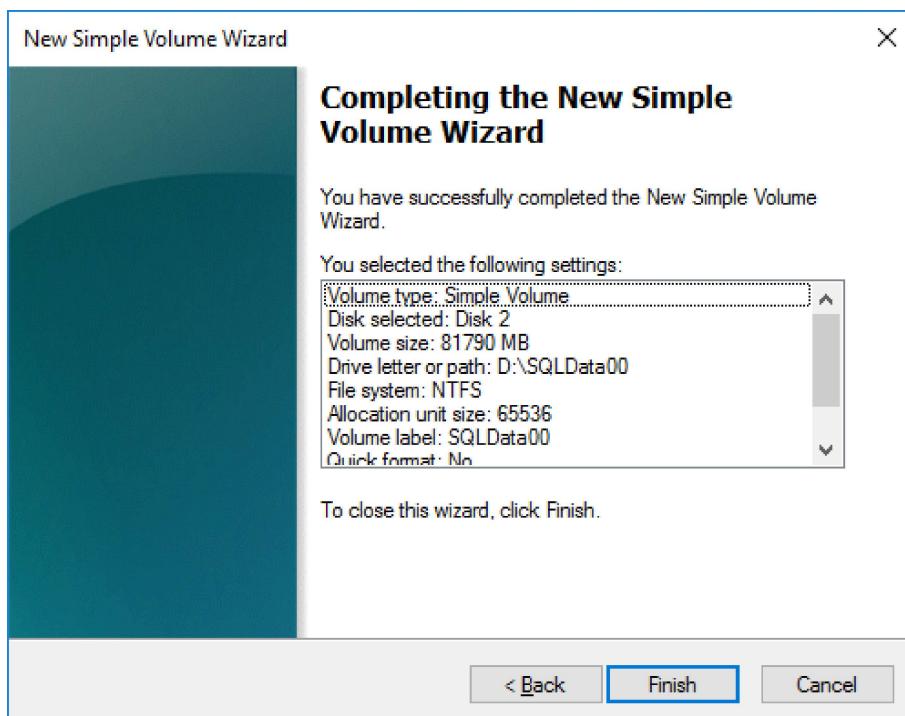
Select D:\ drive and click on **New Folder...** to create folder called SQLData00. Select this folder and click **OK**



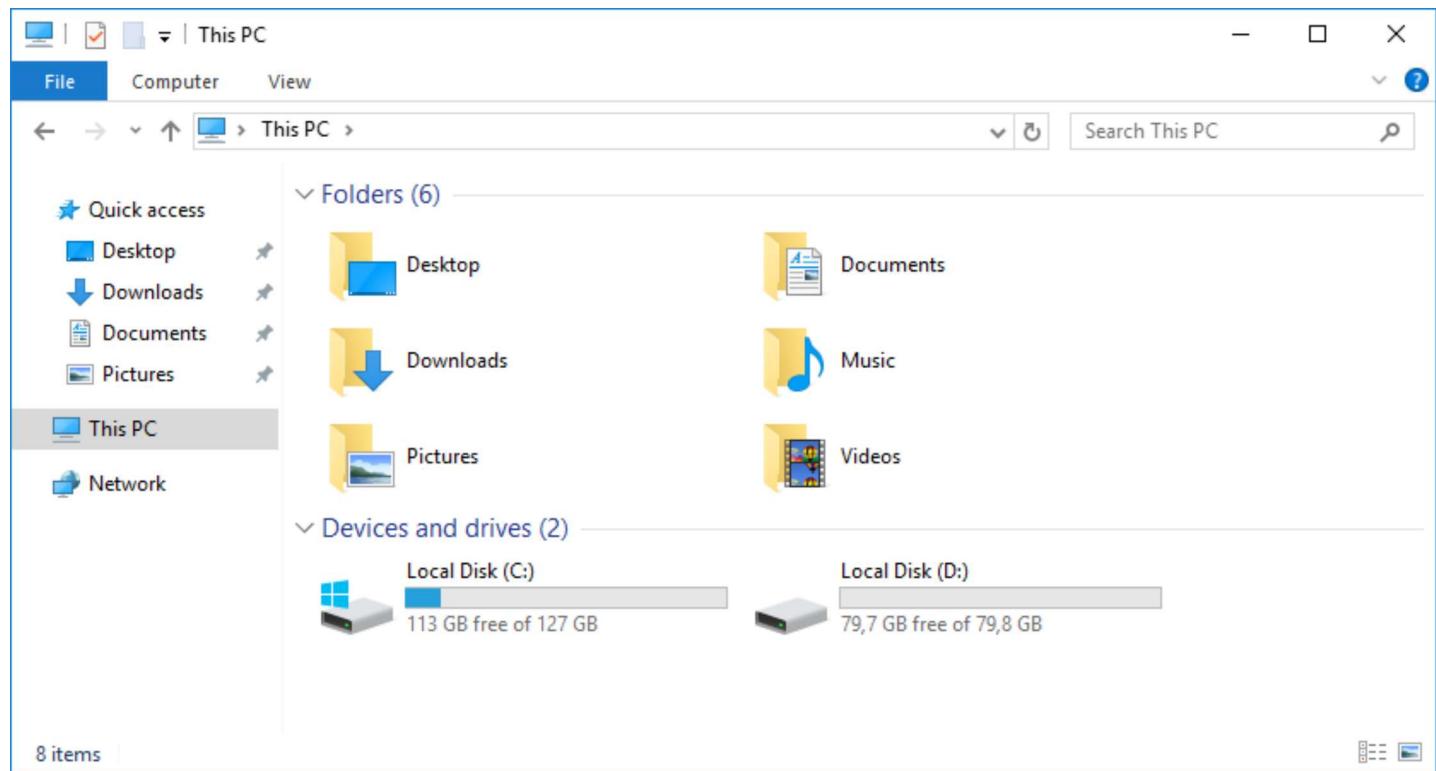
On **Format Partition** select **NTFS file system** and **64K allocation unit**, provide Volume Label (SQLData00) and Deselect *Perform a quick format* option.



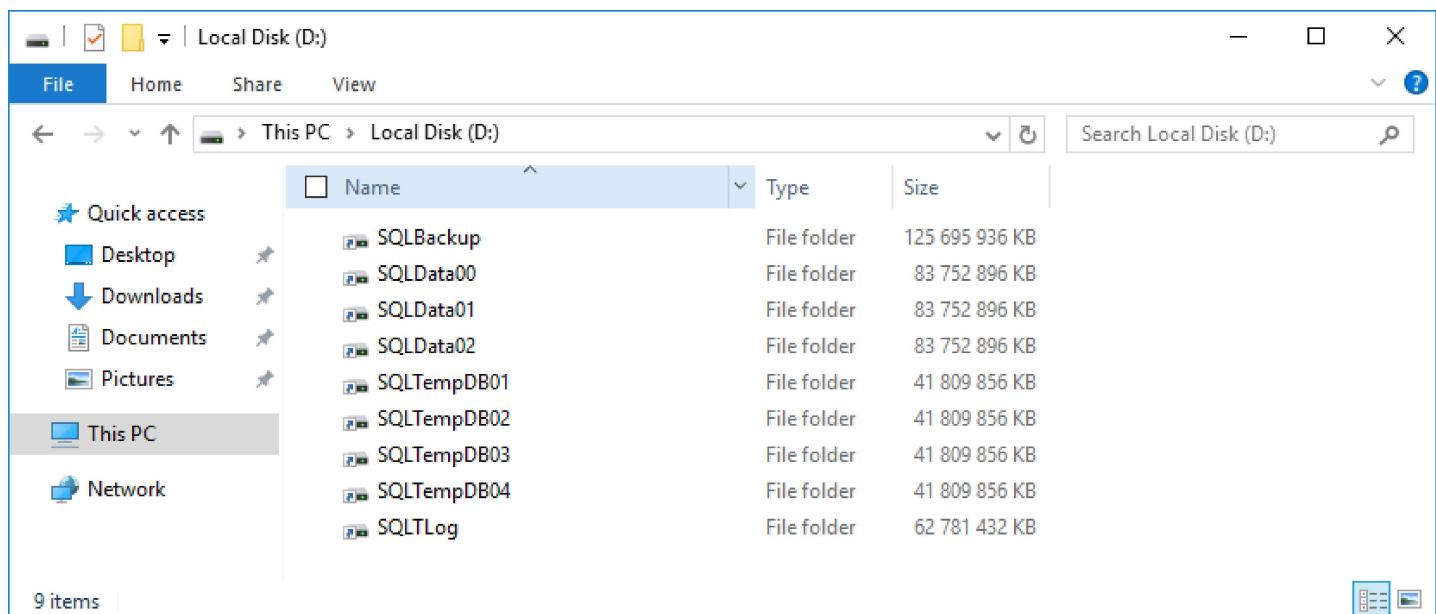
On the summary screen click on **Finish** to start formatting of the drive.



Repeat the same steps for the other drives (SQLData01-02, SQLLog, SQLTempDB01-04 and SQLBackup). Once all steps are completed you should see following drives in **This PC**:



When browsing to D:\ drive all mount points should be visible and in online state.



Group Managed Service Accounts for SQL Server

One of the new feature of the Windows Server 2008 R2 was **Managed Service Account** (MSA). This feature allows you to create an **Active Directory** account that is tied to specific computer object with complex password and automatic process of password renewal (like for the computer object). Such object significantly increase security as the Database Administrator (DBA) doesn't need to change password for SQL Server accounts. One of the restriction of the MSA was that it can't be used for more than one computer. This limitation was solved new feature in Windows Server 2012 called **Group Managed Service Account** (gMSA). Below there are steps describing how enable gMSA on Active Directory and how to configure SQL Server to take advantage of it.

As a first step **Key Distribution Service** (KDC) Root Key needs to be created. For that we are going to execute PowerShell command on domain controller by user with Domain Administrator or Enterprise Administrator privileges.

As we don't want to wait default 10 hours, additional parameter `-EffectiveTime` has been used:

```

1
2 Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10))
3

```

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10))
Guid
f2c72d68-555e-e56c-333c-05c0628a95c0

PS C:\Users\Administrator>

```

Second step is to created **Global Security Group** for all Computers where SQL Server will be installed

```

1
2 New-ADGroup -Name "SQLServers" ` 
3 -GroupCategory Security ` 
4 -GroupScope Global ` 
5 -DisplayName "Production SQL Servers" ` 
6 -Path "CN=Users,DC=datacommunity,DC=local" ` 
7 -Description "Group for all SQL Servers using gMSA"
8

```

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> New-ADGroup -Name "SQLServers" -GroupCategory Security -GroupScope G
lobal -DisplayName "Production SQL Servers" -Path "CN=Users,DC=datacommunity,DC=local" -Descrip
tion "Group for all SQL Servers using gMSA"
PS C:\Users\Administrator>

```

Third step is to add all SQL Server computers to the group created in previous step:

```

1
2 Add-ADGroupMember "SQLServers" -Members "SQL$"
3

```

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Add-ADGroupMember "SQLServers" -Members "SQL$"
PS C:\Users\Administrator>

```

To validate members of the AD Group, following command can be used:

```

1
2 Get-ADGroupMember "SQLServers"
3

```

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Get-ADGroupMember "SQLServers"

distinguishedName : CN=SQL,CN=Computers,DC=datacommunity,DC=local
name             : SQL
objectClass      : computer
objectGUID       : bb9c6da0-4cc2-47d5-b4c3-87abf1c9117f
SamAccountName   : SQL$
SID              : S-1-5-21-326639741-2192422865-3886895453-4103

PS C:\Users\Administrator>

```

Fourth step is to create gMSA accounts for all services that we are going to use – in our case two accounts:

- SQL Server Database Engine
- SQL Server Agent

To add accounts following PowerShell command can be executed:

```

1 New-ADServiceAccount -name SQLServerEngine ` 
2 -DNSHostName SQLServerEngine.contoso.com ` 
3 -PrincipalsAllowedToRetrieveManagedPassword SQLServers
4
5
6 New-ADServiceAccount -name SQLServerAgent ` 
7 -DNSHostName SQLServerAgent.contoso.com ` 
8 -PrincipalsAllowedToRetrieveManagedPassword SQLServers
9

```

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> New-ADServiceAccount -name SQLServerEngine -DNSHostName SQLServerEng
ine.contoso.com -PrincipalsAllowedToRetrieveManagedPassword SQLServers
PS C:\Users\Administrator> New-ADServiceAccount -name SQLServerAgent -DNSHostName SQLServerAgen
t.contoso.com -PrincipalsAllowedToRetrieveManagedPassword SQLServers
PS C:\Users\Administrator>

```

On the member server (SQL computer in our case) there are additional steps that needs to be executed:

Import AD module by running:

```

1 Add-WindowsFeature RSAT-AD-PowerShell
2
3

```

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator.DATACOMMUNITY> Add-WindowsFeature RSAT-AD-PowerShell
Success Restart Needed Exit Code      Feature Result
----- ----- ----- ----- {Remote Server Administration Tools, Activ...
True   No        Success          {Remote Server Administration Tools, Activ...
PS C:\Users\Administrator.DATACOMMUNITY>

```

Both gMSA accounts needs to be installed on the member server by running following command:

```

1 Install-ADServiceAccount SQLServerEngine
2 Install-ADServiceAccount SQLServerAgent
3
4

```

```
Administrator: Windows PowerShell
PS C:\Users\Administrator.DATACOMMUNITY> Install-ADServiceAccount SQLServerEngine
PS C:\Users\Administrator.DATACOMMUNITY> Install-ADServiceAccount SQLServerAgent
PS C:\Users\Administrator.DATACOMMUNITY>
```

To check if gMSA are installed correctly you can run `Test-ADServiceAccount` cmdlet:

```
1 Test-ADServiceAccount SQLServerEngine
2 Test-ADServiceAccount SQLServerAgent
3
4
```

```
Administrator: Windows PowerShell
PS C:\Users\Administrator.DATACOMMUNITY> Test-ADServiceAccount SQLServerEngine
True
PS C:\Users\Administrator.DATACOMMUNITY> Test-ADServiceAccount SQLServerAgent
True
PS C:\Users\Administrator.DATACOMMUNITY>
```

After this step – Active Directory domain and member server are ready for the installation of SQL Server.

Summary

Proper configuration of the Active Directory (AD) domain and operating system can give significant performance, security and manageability advantages. Every time when you build new server for SQL Server, check if all steps described in this blog post are configured.

Links

More information about described above steps can be found on following sites:

<https://docs.microsoft.com/en-us/dotnet/framework/install/on-windows-10>

<https://docs.microsoft.com/en-us/windows-server/storage/disk-management/overview-of-disk-management>

<https://docs.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/group-managed-service-accounts-overview>

[https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/dd758814\(v=sql.100\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/dd758814(v=sql.100))

« Previous

Last week reading (2018-12-09)

Next »

ASF 017: Alex Whittles interview

Tags ◀ Active Directory

◀ installation

◀ PowerShell

Category

◀ Starting as DBA from zero

Share this post:



ABOUT AUTHOR

**Michał Sadowski** 6 posts

Senior Database Platform Engineer, leader of Krakow chapter of Data Community Krakow - PASS Chapter. With more than 10 years of experience in Microsoft technology, helps in advisory for performance optimization on database and application levels. Holds many Microsoft certificates including MCSA: Windows Server 2008/2012/2016, MCSA: SQL Server 2012/2014/2016, MCSE: Data Platform and MCT.

[Twitter](#) [LinkedIn](#)[View all posts by this author →](#)

YOU MIGHT ALSO LIKE



Installation of SQL Server using GUI

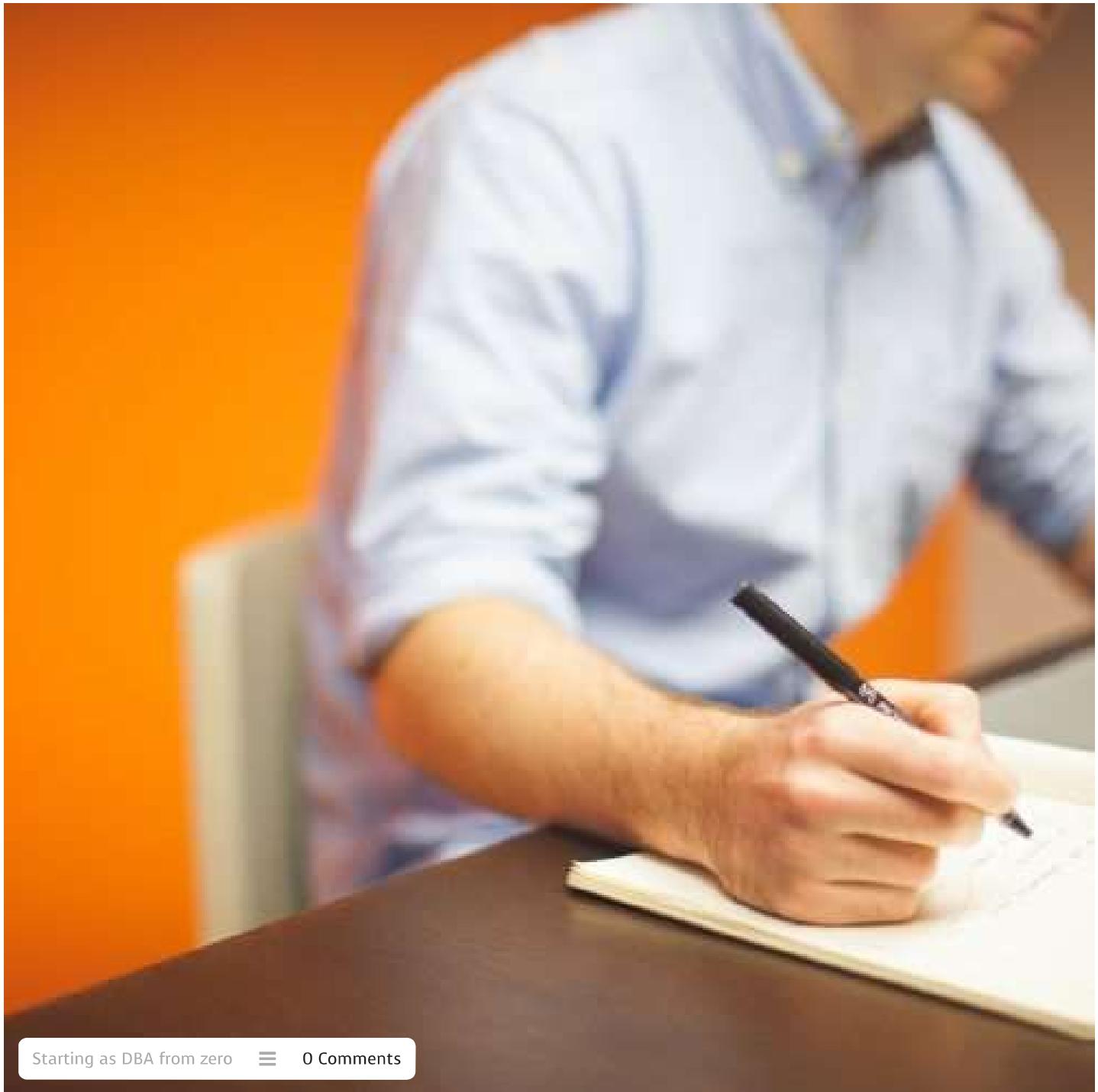
It is a second post on series related to installation and configuration of SQL Server installations. This series is for professionals who starts their journey with SQL Server administration and



General 0 Comments

Configuration of SQL Server instance

It is the fifth post on series related to installation and configuration of SQL Server client tools required. This series is for professionals who start their journey with SQL Server



Starting as DBA from zero ≡ 0 Comments

Installation SQL Server client tools

It is the fourth post on series related to installation and configuration of SQL Server client tools required. This series is for professionals who start their journey with SQL Server

1 COMMENT

Configuration of SQL Server instance | SQLPlayer

JUNE 18, 08:01 REPLY

[...] steps described in this first blog post has been [...]

Leave a Reply

Comment *

Your comment..

Nickname *

Nickname

E-mail *

E-mail

Website

Website

Post a Comment

ARCHIVES

Archives

Select Month



CATEGORIES

Categories

Select Category ▾

CONFERENCES



META

[Log in](#)

[Entries feed](#)

[Comments feed](#)

[WordPress.org](#)

BREAKING NEWS

- 14/11/2019 07:32 [Installation and setting up Power BI Gateway](#)
- 10/11/2019 17:55 [Last Week Reading \(2019-11-10\)](#)
- 03/11/2019 23:59 [Last Week Reading \(2019-11-03\)](#)
- 30/10/2019 07:30 [Script and deploy the data for database from SSDT project](#)
- 29/10/2019 08:03 [Last Week Reading \(2019-10-27\)](#)

© Kamil Nowinski 2017-2019 • All Rights Reserved

