1. SQL Server

|  |  |
| --- | --- |
| Computers used in  this Lab | SRV0001  SRV0002 |
| More information | Support for SQL Server versions for System Center Configuration Manager  <https://docs.microsoft.com/en-us/sccm/core/plan-design/configs/support-for-sql-server-versions>  International support in System Center Configuration Manager  <https://docs.microsoft.com/en-us/sccm/core/plan-design/hierarchy/international-support>  How to determine the version and edition of SQL Server and its components  <https://support.microsoft.com/en-gb/kb/321185> |
| Description | In this chapter, we will install and configure the SQL Server and SQL Reporting Services to be used with SCCM. Basic configuration like max memory and recovery model will also be performed |

* 1. Creating Firewall Rules for SQL

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Open Windows Firewall with Advanced Security and click Inbound Rules |
| 02. Click New Rule |
| 03. On New Inbound Rule Wizard, select Port and click Next |
| 04. On Protocol and Ports select TCP and type 1433 under specify local ports and click Next |
| 05. On Action, click Next |
| 06. On Profile, click Next |
| 07. On Name, type SQL Server (TCP 1433) Inbound and click Finish |
| 08. Click New Rule |
| 09. On New Inbound Rule Wizard, select Port and click Next |
| 10. On Protocol and Ports select TCP and type 4022 under specify local ports and click Next |
| 11. On Action, click Next |
| 12. On Profile, click Next |
| 13. On Name, type SQL Server SSB (TCP 4022) Inbound and click Finish |

This can also be achieved via PowerShell using the commands below:

New-NetFirewallRule -DisplayName "SQL Server (TCP 1433) Inbound " -Action Allow -Direction Inbound -LocalPort 1433 -Protocol TCP

New-NetFirewallRule -DisplayName "SQL Server (TCP 4022) Inbound " -Action Allow -Direction Inbound -LocalPort 4022 -Protocol TCP

* 1. Installing SQL Server 2017

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01.Execute setup.exe from \\srv0001\TrainingFiles\Source\SQLServer\Extract |
| 02. On the SQL Server Installation Center, Click Installation |
| 03. Under the Installation, click New SQL Server stand-alone installation or add features to an existing installation |
| 04. Under Product Key, select Specify a free version, select Evaluation, and click Next |
| 05. Under License Terms, click I accept the license terms and click Next |
| 06. Under Microsoft Update, make sure “Use Microsoft Update to check for updates (recommended)” is not checked and click Next |
| 07. Under Product Updates uncheck Include SQL Server product updates and Click Next  Note: if there is an Error screen, it can be safety ignored as this is because the machine does not have internet access or was unable to connect to the Microsoft Servers |
| 08. Under Install Rules, Click Next |
| 09. Under Feature Selection select Database Engine Services and Change the Instance root directory to C:\SQLServer\, Shared feature directory to C:\SQLServer\ and Shared feature directory (x86) to C:\SQLServer (x86)\ and click Next |
| 10. Instance Configuration select Default Instance. Click Next |
| 11. Under Server Configuration, Account Name for SQL Server Agent, click Browser and type SYSTEM. Click Check Names and click ok. Repeat same steps for SQL Server Database Engine.  Note: Using LocalSystem as Service Account for SQL is not a best practice, however, it is easy for us. For more information about why this is not best practices, refer to <https://www.mssqltips.com/sqlservertip/2384/why-system-account-is-a-bad-idea-for-sql-server-service-account/> |
| 12. Under Server Configuration, click Collation |
| 13. Click Customize and select SQL Collation |
| 14. Select SQL\_Lating1\_General\_CP1\_CI\_AS. Click Ok and then Next |
| 15. Under Database Engine Configuration click Add and type SCCM Admins. Click Check Names and OK. |
| 16. Under Database Engine, click TempDB and configure the Initial Size as 1024. Click next  Note: The TempDB initial size should be approximately 25% of the size of the estimated SCCM database. |
| 17. Under Ready to Install, click Install |
| 18. Once the setup is completed, click Close |

This can also be achieved via PowerShell using the commands below:

$inifile = @"

[OPTIONS]

IACCEPTPYTHONLICENSETERMS="False"

ACTION="Install"

SUPPRESSPRIVACYSTATEMENTNOTICE="False"

IACCEPTROPENLICENSETERMS="False"

ENU="True"

QUIET="False"

QUIETSIMPLE="True"

;UIMODE="Normal"

UpdateEnabled="False"

USEMICROSOFTUPDATE="False"

UpdateSource="MU"

FEATURES=SQLENGINE

HELP="False"

INDICATEPROGRESS="False"

X86="False"

INSTANCENAME="MSSQLSERVER"

INSTALLSHAREDDIR="C:\SQLServer"

INSTALLSHAREDWOWDIR="C:\SQLServer (x86)\ "

INSTANCEID="MSSQLSERVER"

SQLTELSVCACCT="NT Service\SQLTELEMETRY"

SQLTELSVCSTARTUPTYPE="Automatic"

INSTANCEDIR="C:\SQLServer"

AGTSVCACCOUNT="NT AUTHORITY\SYSTEM"

AGTSVCSTARTUPTYPE="Manual"

COMMFABRICPORT="0"

COMMFABRICNETWORKLEVEL="0"

COMMFABRICENCRYPTION="0"

MATRIXCMBRICKCOMMPORT="0"

SQLSVCSTARTUPTYPE="Automatic"

FILESTREAMLEVEL="0"

ENABLERANU="False"

SQLCOLLATION="SQL\_Latin1\_General\_CP1\_CI\_AS"

SQLSVCACCOUNT="NT AUTHORITY\SYSTEM"

SQLSVCINSTANTFILEINIT="False"

SQLSYSADMINACCOUNTS="CLASSROOM\SCCM Admins"

SQLTEMPDBFILECOUNT="2"

SQLTEMPDBFILESIZE="1024"

SQLTEMPDBFILEGROWTH="64"

SQLTEMPDBLOGFILESIZE="8"

SQLTEMPDBLOGFILEGROWTH="64"

ADDCURRENTUSERASSQLADMIN="False"

TCPENABLED="1"

NPENABLED="0"

BROWSERSVCSTARTUPTYPE="Disabled"

"@

$inifile -replace "`n", "`r`n" | Out-File -FilePath "\\srv0001\TempFiles\installsql.ini"

Start-Process -Filepath ("\\srv0001\TrainingFiles\Source\SQLServer\Extract\setup.exe") -ArgumentList ('/ConfigurationFile="\\srv0001\TempFiles\\installsql.ini" /IAcceptSQLServerLicenseTerms') -wait

Start-sleep 30

* 1. Installing SQL Server 2017 CU7

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Execute SQLServer2017-KB4229789-x64.exe from \\srv0001\TrainingFiles\Source\SQLServer |
| 02. Under License Terms, click I accept the license terms and Privacy Statement and click Next |
| 03. Under Select Features, click Next |
| 04. Under Check Files In Use, once the check has been completed, click Next |
| 05. Under Ready to update, click Update |
| 06. Once the update is completed, click Close |

This can also be achieved via PowerShell using the commands below:

Start-Process -Filepath ("\\srv0001\TrainingFiles\Source\SQLServer\SQLServer2017-KB4229789-x64.exe") -ArgumentList ('/quiet /IAcceptSQLServerLicenseTerms /Action=Patch /AllInstances') -wait

* 1. Installing SQL Server Reporting Services

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Execute SQLServerReportingServices.exe from \\srv0001\TrainingFiles\Source\SQLServer |
| 02. On Microsoft SQL Server 2017 Reporting Services, under Welcome, click Install Reporting Services |
| 03. Under Choose and edition to install select Choose a free edition, select Evaluation (expires in 180 days) and click Next |
| 04. Under Review the license terms, click I accept the license terms and click Next |
| 05. Under Install Database Engine, select Install Reporting Services only and click Next |
| 06. Under Specify an installation location, click Install |
| 07. Under Setup completed, select Configure manually and customize settings and click Close |

This can also be achieved via PowerShell using the commands below:

Start-Process -Filepath ("\\srv0001\TrainingFiles\Source\SQLServer\SQLServer2017-KB4229789-x64.exe") -ArgumentList ('/quiet /IAcceptSQLServerLicenseTerms /Action=Patch /AllInstances') -wait

* 1. Configuring SQL Server Reporting Services

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Open Report Server Configuration Manager |
| 02. On The Report Server Configuration Connection connect to SRV0002 server and SSRS Report Server Instance |
| 03. On Report Server Configuration Manager, click Database |
| 04. Under Database, click Change Database |
| 05. Under change Database, Action, select Create a new report server database and click Next |
| 06. On database Server, make sure SRV0002 is already set as Server Name and click Next |
| 07. On Database, make sure Database Name is already set as ReportServer and click Next |
| 08. On Credentials, click Next |
| 09. On Summary, click Next |
| 10. On Progress and Finish, click Finish |
| 11. Back on Report Server Configuration Manager, select Web Service URL and then click Apply |
| 12. Select Web Portal URL and click Apply |

This can also be achieved via PowerShell using the commands below:

$wmiName = (Get-WmiObject –namespace root\Microsoft\SqlServer\ReportServer –class \_\_Namespace).Name

$rsConfig = Get-WmiObject –namespace "root\Microsoft\SqlServer\ReportServer\$wmiName\v14\Admin" -class MSReportServer\_ConfigurationSetting -filter "InstanceName='SSRS'"

##create database

$SQLScript = ($rsConfig.GenerateDatabaseCreationScript('ReportServer', 1033, $false)).Script

Invoke-Sqlcmd -ServerInstance 'srv0002' -Query $SQLScript

##add rights

$SQLScript = ($rsConfig.GenerateDatabaseRightsScript('classroom\sccmadmin', 'ReportServer', $false, $true)).Script

Invoke-Sqlcmd -ServerInstance 'srv0002' -Query $SQLScript

$rsConfig.SetDatabaseConnection('SRV0002', 'ReportServer', 0, 'classroom\sccmadmin','Pa$$w0rd')

$rsConfig.RemoveURL('ReportServerWebService', 'http://+:80', 1033)

$rsconfig.SetVirtualDirectory('ReportServerWebService', 'ReportServer', 1033)

$rsConfig.ReserveURL('ReportServerWebService', 'http://+:80', 1033)

$rsConfig.RemoveURL('ReportServerWebApp', 'http://+:80', 1033)

$rsconfig.SetVirtualDirectory('ReportServerWebApp','Reports',1033)

$rsConfig.ReserveURL('ReportServerWebApp', 'http://+:80', 1033)

$rsConfig.SetServiceState($true, $true, $true)

$rsConfig.InitializeReportServer($rsConfig.InstallationID)

Get-Service -Name SQLServerReportingServices | Restart-Service

start-sleep 30

* 1. Installing SQL Server Management Studio

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Execute SSMS-Setup-ENU.exe from \\srv0001\Trainingfiles\Source\SQLMgmt |
| 02. On the Welcome, click Install |
| 03. On Setup Completed, click Close |

This can also be achieved via PowerShell using the commands below:

Start-Process -Filepath ("\\srv0001\Trainingfiles\Source\SQLMgmt\SSMS-Setup-ENU.exe") -ArgumentList ('/install /quiet /norestart') -wait

* 1. Validating Installation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Open SQL Server Management Studio |
| 02. Connect to the SRV0002 server |
| 03. Click New query and type select @@version and click execute  Note: The result should be something similar to Microsoft SQL Server 2017 (RTM-CU7) (KB4229789) - 14.0.3026.27 (X64) May 10 2018 12:38:11 Copyright (C) 2017 Microsoft Corporation Enterprise Evaluation Edition (64-bit) on Windows Server 2016 Standard 10.0 <X64> (Build 14393: ) (Hypervisor) |
| 04. Now, type SELECT SERVERPROPERTY('productversion'), SERVERPROPERTY('productlevel'), SERVERPROPERTY('edition') and click Execute  Note: the result should be something like 14.0.3026.27 RTM Enterprise Evaluation Edition (64-bit) |
| 05. Open Internet Explorer and navigate to http://SRV0002/ReportServer |
| 06. Navigate to http://SRV0002/Reports |

This can also be achieved via PowerShell using the commands below:

$conn = New-Object System.Data.SqlClient.SqlConnection

$conn.ConnectionString = "Data Source=SRV0002;Initial Catalog=Master;trusted\_connection = true;"

$conn.Open()

$SqlCommand = $Conn.CreateCommand()

$SqlCommand.CommandTimeOut = 0

$SqlCommand.CommandText = "select @@version"

$DataAdapter = new-object System.Data.SqlClient.SqlDataAdapter $SqlCommand

$dataset = new-object System.Data.Dataset

$DataAdapter.Fill($dataset)

$SqlCommand2 = $Conn.CreateCommand()

$SqlCommand2.CommandTimeOut = 0

$SqlCommand2.CommandText = "SELECT SERVERPROPERTY ('productversion'),SERVERPROPERTY ('productlevel'), SERVERPROPERTY ('edition')"

$DataAdapter2 = new-object System.Data.SqlClient.SqlDataAdapter $SqlCommand2

$dataset2 = new-object System.Data.Dataset

$DataAdapter2.Fill($dataset2)

$dataset.Tables[0] | select Column1

$dataset2.Tables[0] | select Column1,Column2,Column3

$conn.close()

$web = New-Object -ComObject msxml2.xmlhttp

$url = @("http://localhost:80/reports", "http://localhost:80/reportserver")

$url | foreach {

$item = $\_

Write-host "Checking $item"

try {

$web.open('GET', $item, $false)

$web.send()

Write-host "HTTP Return $($web.status)"

} catch {

Write-host "ERROR: $($\_)"

}

}

* 1. SQL Server Max Memory

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Open SQL Server Management Studio |
| 02. Connect to the SRV0002 server |
| 03. On Object Explorer, right SRV0002 (SQL Server) and click Properties |
| 04. Under Server Properties – SRV0002, click Memory and set the minimum and maximum memory for the server to 4096. Click Ok  Note: It is recommended to allow 4GB for the Windows OS and set the memory to a minimum of 8GB. If the server has more memory free, the values should be 80% for the SQL (if it is running on its own server) and 50% if the server is co-hosted on the SCCM Server |

This can also be achieved via PowerShell using the commands below:

$maxMem = 4096

$minMem = 4096

[reflection.assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo") | Out-Null

$srv = new-object Microsoft.SQLServer.Management.Smo.Server($SQLInstanceName)

$srv.ConnectionContext.LoginSecure = $true

$srv.Configuration.MaxServerMemory.ConfigValue = $maxMem

$srv.Configuration.MinServerMemory.ConfigValue = $minMem

$srv.Configuration.Alter()

* 1. SQL Server Recovery Model for SQL Server Reporting Services Database

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Open SQL Server Management Studio |
| 02. Connect to the SRV0002 server |
| 03. Expand Databases, select ReportServer and click Properties |
| 04. Under Database Properties – ReportServer, click Options and change Recovery Model from Full to Single. Click Ok |

This can also be achieved via PowerShell using the commands below:

$Server="SRV0002"

$db = "ReportServer"

[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SMO") | out-null

$SMOserver = New-Object ('Microsoft.SqlServer.Management.Smo.Server') -argumentlist $Server

$SMOserver.Databases["$db"] | select Name, RecoveryModel | Format-Table

$SMOserver.databases["$db"].recoverymodel = "Simple"

$SMOserver.databases["$db"].alter()

$SMOserver.Databases["$db"] | select Name, RecoveryModel | Format-Table

* 1. Validating Static Port and Services

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Open SQL Server 2016 Configuration Manager |
| 02. Expand SQL Server Network Configuration and click Protocols for MSSQLSERVER |
| 03. Select with right click TCP/IP and click Properties |
| 04. Click IP Addresses tab and confirm that a static TCP Port is used.  Note: SCCM does not support TCP Dynamic Ports |

This can also be achieved via PowerShell using the commands below:

foreach ($item in (Get-Item -Path "Registry::HKEY\_LOCAL\_MACHINE\Software\Microsoft\Microsoft SQL Server\Instance Names\SQL" | select-object -ExpandProperty Property)) {

$instance = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\Software\Microsoft\Microsoft SQL Server\Instance Names\SQL\").$item

$info = Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\Software\Microsoft\Microsoft SQL Server\$instance\$item\SuperSocketNetLib\Tcp\IpAll" | select TcpDynamicPorts, TcpPort

"{0} - {1} - {2}" -f $item, $info.TcpDynamicPorts, $info.TcpPort }