BY RAPHAEL PEREZ, MICROSOFT MVP IN ENTERPRISE MOBILITY

MVP Profile: <https://mvp.microsoft.com/en-us/PublicProfile/4027143>

Twitter: @dotraphael

LinkedIn: <https://uk.linkedin.com/in/dotraphael>

Blog: <http://thedesktopteam.com/raphael>

Microsoft Endpoint Configuration Manager Administration E-BOOK

Version: Draft 4.00 | Date: February 2020

2020

Table of Contents

[1. Document Change Control Sheet 9](#_Toc32830058)

[1.1. Document History 9](#_Toc32830059)

[2. About 10](#_Toc32830060)

[2.1. Raphael Perez (Author) 10](#_Toc32830061)

[2.2. David Nudelman (Reviewer) 10](#_Toc32830062)

[3. Introduction 11](#_Toc32830063)

[4. Lab Information 12](#_Toc32830064)

[4.1. PowerShell 14](#_Toc32830065)

[4.2. Installing a Hyper-V Server 15](#_Toc32830066)

[4.3. Installing Hyper-V Role 15](#_Toc32830067)

[4.4. Downloading Software 16](#_Toc32830068)

[4.5. Creating Windows Virtual Machines 17](#_Toc32830069)

[4.6. CLASSROOM-ROUTER01 17](#_Toc32830070)

[4.7. CLASSROOM-SRV0001 18](#_Toc32830071)

[4.8. CLASSROOM-SRV0002 18](#_Toc32830072)

[4.9. CLASSROOM-WKS0001 18](#_Toc32830073)

[4.10. CLASSROOM-WKS0002 19](#_Toc32830074)

[4.11. CLASSROOM-WKS0004 19](#_Toc32830075)

[5. Active Directory 20](#_Toc32830076)

[5.1. Extending Active Directory Schema 20](#_Toc32830077)

[5.2. Creating the System Management Container 20](#_Toc32830078)

[6. Pre-Requirements 22](#_Toc32830079)

[6.1. Installing .NET Framework 3.5 22](#_Toc32830080)

[6.2. Validating .NET Framework 4.6 Installation 22](#_Toc32830081)

[6.3. Installing Microsoft Remote Differential Compression 23](#_Toc32830082)

[6.4. Installing Assessment and Deployment Kit (ADK) for Windows 10 1903 23](#_Toc32830083)

[6.5. Installing Assessment and Deployment Kit (ADK) WinPE AddOn 24](#_Toc32830084)

[6.6. Installing Assessment and Deployment Kit (ADK) FIX 24](#_Toc32830085)

[6.7. Logs Reading 25](#_Toc32830086)

[7. SQL Server 27](#_Toc32830087)

[7.1. Creating Firewall Rules for SQL 27](#_Toc32830088)

[7.2. Installing SQL Server 2017 27](#_Toc32830089)

[7.3. Installing SQL Server 2017 latest CU 29](#_Toc32830090)

[7.4. Installing SQL Server Reporting Services 30](#_Toc32830091)

[7.5. Configuring SQL Server Reporting Services 30](#_Toc32830092)

[7.6. Installing SQL Server Management Studio 31](#_Toc32830093)

[7.7. Validating Installation 32](#_Toc32830094)

[7.8. SQL Server Max Memory 33](#_Toc32830095)

[7.9. SQL Server Recovery Model for SQL Server Reporting Services Database 33](#_Toc32830096)

[7.10. Validating Static Port and Services 34](#_Toc32830097)

[8. Installation of MECM Primary Site 35](#_Toc32830098)

[8.1. Downloading MECM Setup Files 35](#_Toc32830099)

[8.2. Pre-requirements Check 35](#_Toc32830100)

[8.3. Creating Firewall Rules for MECM Site Server Console Communication 36](#_Toc32830101)

[8.4. Site Server Installation 36](#_Toc32830102)

[8.5. Installation Status 38](#_Toc32830103)

[8.6. Console Overview 40](#_Toc32830104)

[8.7. Validating Installation 43](#_Toc32830105)

[9. Upgrade to Current Branch 1910 50](#_Toc32830106)

[9.1. Before you begin 50](#_Toc32830107)

[9.2. Run the prerequisite check 51](#_Toc32830108)

[9.3. Installing an Update 52](#_Toc32830109)

[9.4. Turning on Features 55](#_Toc32830110)

[10. Basic Site Configuration 56](#_Toc32830111)

[10.1. Installation of basic Roles (DP, MP, FSP, SRS) 56](#_Toc32830112)

[10.1.1. Distribution Point 56](#_Toc32830113)

[10.1.1.1. Creating Firewall Rules 57](#_Toc32830114)

[10.1.1.2. Install Requirements 57](#_Toc32830115)

[10.1.1.3. Installing Site System Role 58](#_Toc32830116)

[10.1.2. Management Point 60](#_Toc32830117)

[10.1.2.1. Creating Firewall Rules 60](#_Toc32830118)

[10.1.2.2. Install Requirements 61](#_Toc32830119)

[10.1.2.3. Installing Site System Role 61](#_Toc32830120)

[10.1.3. Fallback Status Point 64](#_Toc32830121)

[10.1.3.1. Creating Firewall Rules 64](#_Toc32830122)

[10.1.3.2. Install Fallback Requirements 64](#_Toc32830123)

[10.1.3.3. Installing Site System Role 65](#_Toc32830124)

[10.1.4. Reporting Services Point 66](#_Toc32830125)

[10.1.4.1. Creating Firewall Rules 66](#_Toc32830126)

[10.1.4.2. Installing Site System Role 67](#_Toc32830127)

[10.2. Boundaries 69](#_Toc32830128)

[10.3. Boundary Group 70](#_Toc32830129)

[10.4. Distribution Point Group 70](#_Toc32830130)

[10.5. Network Access Account 71](#_Toc32830131)

[10.6. Discovery 71](#_Toc32830132)

[10.6.1. Active Directory Forest Discovery 71](#_Toc32830133)

[10.6.2. Active Directory System Discovery 72](#_Toc32830134)

[10.6.3. Active Directory User Discovery 73](#_Toc32830135)

[10.6.4. Active Directory Group Discovery 74](#_Toc32830136)

[11. Basic Client Settings 76](#_Toc32830137)

[11.1. Changing Default Client Settings 76](#_Toc32830138)

[12. Client Installation 78](#_Toc32830139)

[12.1. Windows Client Installation 78](#_Toc32830140)

[12.1.1. Push Configuration 78](#_Toc32830141)

[12.1.2. Manual Push 79](#_Toc32830142)

[12.1.3. Validating the Installation and Installation Process on the Client 80](#_Toc32830143)

[12.1.4. Validating the Installation and Installation Process on the MECM Console 81](#_Toc32830144)

[12.2. Client Properties 81](#_Toc32830145)

[12.3. Resource Explorer 81](#_Toc32830146)

[13. Collections 83](#_Toc32830147)

[13.1. Windows 10 Workstation Collection 83](#_Toc32830148)

[13.2. Windows 8 Workstation Collection 84](#_Toc32830149)

[13.3. Active Directory OU Workstations Collection 85](#_Toc32830150)

[13.4. Active Directory OU Users Collection 86](#_Toc32830151)

[13.5. Collection Membership Incremental Evaluation 87](#_Toc32830152)

[14. Remote Control 89](#_Toc32830153)

[14.1. Creating Device Settings for Remote Tools 89](#_Toc32830154)

[14.2. Checking the Policies that should be applied 90](#_Toc32830155)

[14.3. Validating Remote Control 90](#_Toc32830156)

[14.4. Starting Remote Control 91](#_Toc32830157)

[14.5. Monitoring Remote Access from Client 92](#_Toc32830158)

[14.6. Monitoring Remote Access from Server 92](#_Toc32830159)

[15. Hardware Inventory 94](#_Toc32830160)

[15.1. Changing Default Client Settings 94](#_Toc32830161)

[15.2. Updating Client Policies 95](#_Toc32830162)

[15.3. Starting Hardware Inventory 95](#_Toc32830163)

[15.4. Resource Explorer 96](#_Toc32830164)

[16. Client Health 97](#_Toc32830165)

[16.1. Client Status Settings 97](#_Toc32830166)

[16.2. Client Health Configuration 97](#_Toc32830167)

[16.3. Executing CCMEVAL manually 98](#_Toc32830168)

[16.4. Forcing CCMEVAL failure 98](#_Toc32830169)

[16.5. Monitoring Client Health 99](#_Toc32830170)

[17. Software Metering 103](#_Toc32830171)

[17.1. Changing Default Client Settings 103](#_Toc32830172)

[17.2. Updating Default Software Metering Settings and Clearing existing rules 103](#_Toc32830173)

[17.3. Creating Rule 104](#_Toc32830174)

[17.4. Starting Validation Software Metering 104](#_Toc32830175)

[17.5. Summarization Software Metering Data Manually 105](#_Toc32830176)

[17.6. Monitoring Software Metering Data via Reports 105](#_Toc32830177)

[17.7. Monitoring Software Metering Data via Collections 106](#_Toc32830178)

[18. Primary Users 108](#_Toc32830179)

[18.1. Changing Default Client Settings 108](#_Toc32830180)

[18.2. Manual association by the Administrator 108](#_Toc32830181)

[18.3. Manual association by the User 109](#_Toc32830182)

[19. Application Management - Basic 111](#_Toc32830183)

[19.1. Creating Application 111](#_Toc32830184)

[19.2. Adding Requirements to Applications 111](#_Toc32830185)

[19.3. Return Codes 112](#_Toc32830186)

[19.4. Distributing Application Content to Distribution Point Group 113](#_Toc32830187)

[19.5. Monitoring Application Content Distribution via Console 113](#_Toc32830188)

[19.6. Monitoring Application Content Distribution via Reports 114](#_Toc32830189)

[20. Deploying and monitoring Applications 115](#_Toc32830190)

[20.1. Deploying Application 115](#_Toc32830191)

[20.2. Installing available application 116](#_Toc32830192)

[20.3. Monitoring Application Deployment via Console 117](#_Toc32830193)

[20.4. Monitoring Application Deployment via Client Logs 117](#_Toc32830194)

[20.5. Monitoring Application Deployment via Reports 118](#_Toc32830195)

[21. Application Management – App-V 5 Applications 119](#_Toc32830196)

[21.1. Creating an Application App-V Client 5.0 119](#_Toc32830197)

[21.2. Creating a Virtual Application Robocopy 121](#_Toc32830198)

[21.3. Visualizing Application Relationship 122](#_Toc32830199)

[22. Deploying and monitoring available Virtual Applications to a User 123](#_Toc32830200)

[22.1. Enable App-V in-box 123](#_Toc32830201)

[22.2. Deploying Application 124](#_Toc32830202)

[22.3. Installing available application 124](#_Toc32830203)

[23. Application Management – Advanced 125](#_Toc32830204)

[23.1. Preparing the Environment 125](#_Toc32830205)

[23.2. Creating an Application Supersedence 128](#_Toc32830206)

[23.3. Creating an Application Supersedence Deployment 129](#_Toc32830207)

[23.4. Installing an Application Supersedence 130](#_Toc32830208)

[23.5. Creating a Global Condition 131](#_Toc32830209)

[23.6. Using a Custom Global Condition 131](#_Toc32830210)

[23.7. Deploying an Approval Required Application 132](#_Toc32830211)

[23.8. Requesting application via Software Center 132](#_Toc32830212)

[23.9. Approving/Denying application requests 133](#_Toc32830213)

[23.10. Installing an approved application 133](#_Toc32830214)

[23.11. Simulate Application Deployment 134](#_Toc32830215)

[24. Software Update 135](#_Toc32830216)

[24.1. Installation of Windows Server Update Services 135](#_Toc32830217)

[24.2. Installation of the Software Update Point 136](#_Toc32830218)

[24.3. Manual Synchronization 141](#_Toc32830219)

[24.4. Changing the List of Products 141](#_Toc32830220)

[24.5. Changing Default Client Settings 142](#_Toc32830221)

[24.6. MECM Client Scan 143](#_Toc32830222)

[24.7. Reporting Compliance via Console for All Updates 144](#_Toc32830223)

[24.8. Deploying a Patch 144](#_Toc32830224)

[24.9. Installing a Patch 146](#_Toc32830225)

[24.10. Reporting Compliance via Console for an Update Group 147](#_Toc32830226)

[24.11. Reporting Compliance via Reports 147](#_Toc32830227)

[25. Windows 10 Express Updates 149](#_Toc32830228)

[25.1. Preparing the Environment 149](#_Toc32830229)

[25.2. Enabling Express Updates support in MECM 150](#_Toc32830230)

[25.3. Monitoring Windows 10 Express Updates via Client Logs 151](#_Toc32830231)

[26. 3rd Party Updates with PatchMyPc 152](#_Toc32830232)

[26.1. Enable Configuration Manager-generated certificates for HTTP site systems 152](#_Toc32830233)

[26.2. Enabling 3rd Paty Updates 153](#_Toc32830234)

[26.3. Installing Patch My PC Publishing Service 154](#_Toc32830235)

[26.4. Configuring Patch My PC Publishing Service 154](#_Toc32830236)

[26.5. Enable Patch My PC Products 155](#_Toc32830237)

[27. Endpoint Protection 157](#_Toc32830238)

[27.1. Configuring Software Update Point 157](#_Toc32830239)

[27.2. Installing Endpoint Protection Point 158](#_Toc32830240)

[27.3. Automating the definition updates delivery 160](#_Toc32830241)

[27.4. Changing Default Client Settings 161](#_Toc32830242)

[27.5. Testing Malware activity 162](#_Toc32830243)

[27.6. Monitoring Malware Activity via Console 162](#_Toc32830244)

[27.7. Monitoring Malware Activity via Reports 163](#_Toc32830245)

[28. Compliance Settings 165](#_Toc32830246)

[28.1. Changing Default Client Settings 165](#_Toc32830247)

[28.2. Registry Configuration Items with Auto-remediation 165](#_Toc32830248)

[28.3. Application Settings Configuration Items without Auto-remediation 166](#_Toc32830249)

[28.4. Creating Baselines 168](#_Toc32830250)

[28.5. Deploying Baselines 168](#_Toc32830251)

[28.6. Starting Validation Compliance Settings 168](#_Toc32830252)

[28.7. Monitoring Baselines from Client 169](#_Toc32830253)

[28.8. Monitoring Baselines via Console 171](#_Toc32830254)

[28.9. Monitoring Baselines via Reports 171](#_Toc32830255)

[28.10. Monitoring Baselines via Collections 172](#_Toc32830256)

[29. Data Warehouse 173](#_Toc32830257)

[29.1. Data Warehouse service point 173](#_Toc32830258)

[29.2. Data Warehouse retention period 175](#_Toc32830259)

[29.3. SQL Server Permission 176](#_Toc32830260)

[29.4. Force Manual Synchronisation 176](#_Toc32830261)

[29.5. Monitoring Historical data via Reports 177](#_Toc32830262)

[30. Role Based Access Control 178](#_Toc32830263)

[30.1. Creating Security Scope 178](#_Toc32830264)

[30.2. Using Security Scope 178](#_Toc32830265)

[30.3. Creating an Application Administrator 179](#_Toc32830266)

[30.4. Testing new Security Rights 179](#_Toc32830267)

[31. Backup via MECM 181](#_Toc32830268)

[31.1. Configuring Backup 181](#_Toc32830269)

[31.2. Starting Backup Manually 181](#_Toc32830270)

[31.3. Monitoring Backup 182](#_Toc32830271)

[32. Restore 183](#_Toc32830272)

[32.1. Stop MECM Site 183](#_Toc32830273)

[32.2. Delete MECM Database 183](#_Toc32830274)

[32.3. Restore MECM Site 184](#_Toc32830275)

[32.4. Post-Restore Tasks 185](#_Toc32830276)

[32.4.1. Accounts Password Reset 185](#_Toc32830277)

[32.4.2. Distribution Point Self-Signed Certificate Reset 186](#_Toc32830278)

[33. Appendix B – Unmissable Sites 187](#_Toc32830279)

[34. Appendix A – Tools 188](#_Toc32830280)

[34.1. DataExplorer 188](#_Toc32830281)

[34.2. HealthCheck Toolkit 188](#_Toc32830282)

[34.3. PatchMaster 188](#_Toc32830283)

[34.4. SCCM Client Center 188](#_Toc32830284)

[34.5. Mark Cochrane RegkeytoMof 3.3a 188](#_Toc32830285)

[34.6. OSD WebPortal 188](#_Toc32830286)

[34.7. RCT Community 189](#_Toc32830287)

[34.8. RuckZuck 189](#_Toc32830288)

[34.9. Clean Software Update Groups console extension for ConfigMgr 189](#_Toc32830289)

[34.10. Cireson Remote Manage app 189](#_Toc32830290)

[34.11. Reg2CI 189](#_Toc32830291)

[34.12. Collection Commander 189](#_Toc32830292)

[34.13. SQL Server Index and Statistics Maintenance 190](#_Toc32830293)

[34.14. PowerShell – SQL Audit Script 190](#_Toc32830294)

[34.15. 1E’s Free Tools 190](#_Toc32830295)

[34.16. ConfigMgr Task Sequence Monitor 190](#_Toc32830296)

[34.17. ConfigMgr OSD FrontEnd 190](#_Toc32830297)

[34.18. Windows-Nood OSD FrontEnd 190](#_Toc32830298)

1. Document Change Control Sheet
   1. Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change/Reference |
| April/2016 | Raphael Perez | 1.00 | Initial Release |
| February/2017 | Raphael Perez | 2.00 | Updated to Windows Server 2016, SQL 2016 and SCCM 1610 |
| February/2018 | Raphael Perez | 3.00 | Removed Linux related topics, Updated Windows 10 to 1709, added sections for Cloud Management Gateway (CMG), Windows 10 Express updates, SCCM Data warehouse, Office 365 Installer, Windows 10 Update Readiness, Pending Reboot, updated some other section for SCCM 1710 and updated/added some powershell scripts |
| February/2020 | Raphael Perez | Draft 4.0 | Updated Scripts, MECM to 1910, Windows Server 2019, Windows 10 1909, ADK to 1903, |

1. About
   1. Raphael Perez (Author)

Raphael is a Microsoft MVP (<https://mvp.microsoft.com/en-us/PublicProfile/4027143>) with over 20 years of experience in IT, of which 14 years have been dedicated to Device Management, System Center and Automation.

One of four MVPs in Enterprise Client Management in the UK, Raphael holds more than 25 Microsoft certifications and is an MCT (Microsoft Certified Trainer). Since 2008, Raphael has been providing Microsoft training from basic to advanced levels in several categories.

Throughout his career, Raphael has participated as a speaker in well-known events such as TechEd and Gartner Security Risk Management. He also organised community events and lectured around the world, sharing best practices and knowledge within the industry.

Bilingual in English and Portuguese, Raphael has authored diverse articles published in Microsoft's TechEd, served as the editor-in-chief of a magazine focused on System Center in Brazil and wrote two books: "Understanding System Center 2012 SP1 Configuration Manager: The walkthrough book" (<https://wp.me/p3ttD0-am> and <https://wp.me/p3ttD0-8S>) and "System Center 2012 R2 Configuration Manager: Automation from Zero to Hero" (<https://wp.me/p3ttD0-pd>).

He is a Community leader, attending physical and virtual meetings and engaging with the community across several forums, twitter (<http://twitter.com/dotraphael>), LinkedIn (<http://www.linkedin.com/in/dotraphael>) and his blog (<http://www.thedesktopteam.com/>).

* 1. David Nudelman (Reviewer)

David has over 15 years of experience in IT Infrastructure strategy, deployment, migration and management. He is a very experienced technical leader that focus on enabling and training his team to achieve more. He holds certifications from Microsoft, Citrix, HP and VMware, and was awarded seven times as Microsoft Most Valuable Professional, due to his outstanding contributions to the Technical Community.

As a conference speaker David has a very informal style of delivering presentations and speeches. Mr. Nudelman presented at key conferences such as TechEd Europe and US, IP Expo, Global Azure Bootcamp, Computer Weekly CW500 and many more. He is a Cloud Activist, encouraging and helping companies to embrace and adopt cloud technologies.

David is a blogger and writer, contributing to communities such as The Desktop Team ([www.thedesktopteam.com](http://www.thedesktopteam.com)) and IT Pro Spain ([www.itpro.es](http://www.itpro.es)). He is one of the top 5% contributors to the Microsoft TechNet forums, earning multiple times the “Microsoft Community Contributor” award.

Find out more about him on Twitter (<https://twitter.com/nudelmanuk>) or on his personal blog at <http://thedesktopteam.com/david>

1. Introduction

Microsoft Endpoint Manager is an integrated solution for managing devices that includes Configuration Manager, Intune, Desktop Analytics and Autopilot.

Microsoft Endpoint Configuration Manager (MECM) is a Microsoft product that allows management of Windows devices and includes features like remote control, patch management operating system deployment among others.

MECM, formerly known as SCCM (System Center Configuration Manager) and SMS (System Management Server) was originally released in 1994. Renamed in November 2007 to SCCM when it became part of the System Center family and again in November 2019 when it became member of the Microsoft Endpoint Manager family.

With proper planning, MECM can increase IT productivity and efficiency by reducing manual tasks, increase device security and empower users by allowing them to have the right software at the right time.

MECM can manage the whole lifecycle of a device, whenever it is connected to the corporate network or outside corporate boundaries (i.e. Internet) and provide with IT with comprehensive information that can be used to proper manage device, either by built-in features or integration with other solutions (i.e. Desktop Analysis, 3rd Party Patching).

This e-book has been created to provide you with a step by step information, so you can start understanding a Microsoft Endpoint Configuration Manager (MECM) world. The intended audience are technical people who want to learn or improve their understanding of MECM. Minimum knowledge of the following software and technologies is assumed, including but not limited to Active Directory, SQL Server, Windows Server, Microsoft Deployment Toolkit, BitLocker, Hyper-V, PowerShell, and Windows Client (i.e. Windows 7, Windows 8, Windows 10).

It is recommended to use this e-book as it has been written because there are dependencies between the chapters.

Please note that the terms Microsoft Endpoint Configuration Manager, ConfigMgr, Configuration Manager, CM, MECM and SCCM all refer to the same Microsoft product, and the terms are used interchangeably.

1. Lab Information

The Microsoft Endpoint Configuration Manager lab environment was created using Hyper-V 2019 Virtual Machines connected to its own virtual network. The lab has seven (7) virtual machines installed on one (1) Hyper-V host, installed with default configuration, as per following configuration:

|  |  |  |  |
| --- | --- | --- | --- |
| Virtual Machine | Hardware | Description | Base OS |
| HYPER-V | RAM: 24GB  Drive 01 (C): 500GB  Drive 02 (D): DVD  Processor/Core: 4  Network Adapter | Hyper-V Server | Windows Server 2019  IP Address: DHCP |
| ROUTER01 | RAM: 512MB  Drive 01: 2GB  Processor/Core: 1  Network Adapter  Network Adapter | Linux router used to connect VMs to the internet | VyOS  External IP: DHCP  Internal IP: 192.168.3.254  Internal Subnet 255.255.255.0  Internal DNS 192.168.3.1 |
| SRV0001 | RAM: 2048MB  Drive 01 (C): 127GB  Drive 02 (D): DVD  Processor/Core: 1  Network Adapter | Domain Controller for domain called classroom.intranet (netbios name classroom), DNS, DHCP and Enterprise CA | Windows Server 2019  IP Address: 192.168.3.1  Subnet 255.255.255.0  Default Gateway: 192.168.3.254  DNS 192.168.3.1 |
| SRV0002 | RAM: 8192MB  Drive 01 (C): 127GB  Drive 02 (D): DVD  Processor/Core: 2  Network Adapter | Site Server for ConfigMgr | Windows Server 2019  IP Address: 192.168.3.2  Subnet 255.255.255.0  Default Gateway: 192.168.3.254  DNS 192.168.3.1 |
| WKS0001 | RAM: 2048MB  Drive 01 (C): 127GB  Processor/Core: 1  Network Adapter | Windows 10 Enterprise Edition x64 – Workstation | Windows 10 x64  IP Address: DHCP |
| WKS0002 | RAM: 2048MB  Drive 01 (C): 127GB  Processor/Core: 1  Network Adapter | Windows 10 Enterprise Edition x64 – Workstation | Windows 10 x64  IP Address: DHCP |
| WKS0004 | RAM: 2048MB  Drive 01 (C): 127GB  Processor/Core: 1  Network Adapter | Windows 8.1 Enterprise Edition x64 – Workstation | Windows 8.1 x64  IP Address: DHCP |

All user accounts have the password set to Pa$$w0rd and the below list explains its utilization:

|  |  |
| --- | --- |
| Account | Objective |
| CLASSROOM\administrator | Domain admin account |
| CLASSROOM\admworkstation | Domain user account used to demonstrate RBA settings. |
| CLASSROOM\mecmadmin | Account with full rights on the MECM Servers |
| CLASSROOM\mecmpush | Account used for client push. This account has admin rights on all workstations |
| CLASSROOM\svc\_mecmna | Account used as network account |
| CLASSROOM\svc\_ssrsea | Account used as SSRS execution account |
| CLASSROOM\svc\_mecmjoin | Account used to join computers to the domain |
| CLASSROOM\User01 | Account used to deploy software to |
| CLASSROOM\User02 | Account used to deploy software to |

The following table shows the groups created to be used on this training and its objective:

|  |  |
| --- | --- |
| Group | Objective |
| CLASSROOM\MECM Admins | Contain all users with Full Access to the MECM Infrastructure and it is a member of the Remote Tools group |
| CLASSROOM\Workstation Admins | Contain the Admworkstation user |
| CLASSROOM\MECM Remote Tools | Contain users with rights to remote access client machines |
| CLASSROOM\MECM Servers | Contain all MECM Servers |

The following table shows the group policies used on this training and its objective:

|  |  |  |  |
| --- | --- | --- | --- |
| Group Policy | Objective | Link | Enabled |
| Disable Windows Service | Set the BITS Window Service as disabled | Workstations Disabled OU | YES |
| MECM Local Administrators | Set the Local Administrators membership group for the MECM Servers | MECM Servers OU | YES |
| Workstation Local Administrators | Set the Local Administrators group membership for the Desktops | Workstations OU | YES |
| Workstation Local Firewall | Set the Workstations Firewall Exclusion for the Client Push | Workstations OU | YES |

* 1. PowerShell

Automation is a key skill for IT Professionals in today’s world and everything can be automated. Within Windows and Microsoft Endpoint Configuration Manager this is also true, so I have created over 200 scripts that can help you when using this e-book. The collection of scripts can be downloaded from <https://github.com/dotraphael/MECMAdminEbookv4>.

Some of the scripts are used to create the entire lab environment using Hyper-V. It is recommended to use PowerShell ISE instead of a normal PowerShell console as it is richer environment. While many PowerShell scripts are expected to run without any user intervention, they have not been created to log or show results easily. Some scripts require you to run few lines at a time as a reboot of the machine may be necessary.

**Note:** To be able to run the PowerShell scripts, you need to change the PowerShell Execution Policy accordingly. This is necessary because the scripts are not signed and may be run from a remote location. Perform this change in a production environment is not recommended.

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

Set-ExecutionPolicy Unrestricted -Force

Finally, whenever possible, I have created scripts to automate tasks, they are located under the Course Scripts folder, however, some tasks may not have an associated script (i.e. Browse Internet, Microsoft Azure, etc.).

Most of the time, these scripts run from the Configuration Manager Drive. To achieve this, you can start the PowerShell via the MECM Console or use the following PowerShell commands to enable the PowerShell MECM environment from a normal PowerShell:

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ModulePath = $env:SMS\_ADMIN\_UI\_PATH

if ($ModulePath -eq $null) {

$ModulePath = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment").SMS\_ADMIN\_UI\_PATH

}

$ModulePath = $ModulePath.Replace("bin\i386","bin\ConfigurationManager.psd1")

$Certificate = Get-AuthenticodeSignature -FilePath "$ModulePath" -ErrorAction SilentlyContinue

$CertStore = New-Object System.Security.Cryptography.X509Certificates.X509Store("TrustedPublisher")

$CertStore.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::MaxAllowed)

$Certexist = ($CertStore.Certificates | where {$\_.thumbprint -eq $Certificate.SignerCertificate.Thumbprint}) -ne $null

if ($Certexist -eq $false) {

$CertStore.Add($Certificate.SignerCertificate)

}

$CertStore.Close()

import-module $ModulePath -force

if ((get-psdrive $SiteCode -erroraction SilentlyContinue | measure).Count -ne 1) {

new-psdrive -Name $SiteCode -PSProvider "AdminUI.PS.Provider\CMSite" -Root $servername

}

cd "$($SiteCode):"

**Note:** To be able to run some of the PowerShell scripts on the workstations, Domain Users Group will be added to the Local Administrators Group on the WKS0001, WKS0002 and WKS0004.

* 1. Installing a Hyper-V Server

Before we start, we need to build a Hyper-V Server that will host our Virtual Environment.

**Note:** Hyper-V server requires Virtualization capability in the host hardware, for details see <https://technet.microsoft.com/en-us/windows-server-docs/compute/hyper-v/system-requirements-for-hyper-v-on-windows>.

To create a Hyper-V Server, perform the following actions:

|  |
| --- |
| 01. Download Windows Server 2019 Evaluation from Microsoft website <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019> and burn a DVD |
| 02. Insert the Windows Server 2019 DVD-ROM and turn on your computer. After a few minutes, you receive the Windows Server 2019 screen shown. Select the correct Language, Time and Currency Format and Keyboard or input method and Click Next. |
| 03. On the next Install Windows screen, click Install now. |
| 04. On the Select the Operating System you want to install, select Windows Server 2019 Standard Evaluation (Desktop Experience) and click Next. |
| 05. Under License terms, select I accept the license terms and click Next |
| 06. Under Which type of installation do you want? Click Custom: Install Windows only (advanced) |
| 07. Under Where do you want to install Windows? Click Next |
| 08. The Installation will start and it will take some time to complete (15-30 minutes depending on your hardware). |
| 09. Once the installation is completed, On Customize Settings, you must change the password before logging on for the first time. Type the new password and once completed, click Finish. |
| 10. Perform windows update until there is no other update to be applied |
| 11. Download the collection of Scripts from <https://github.com/dotraphael/MECMAdminEbookv4/raw/master/TrainingFilesv4ADM.zip> and extract to C:\ |

* 1. Installing Hyper-V Role

|  |
| --- |
| Perform this task on the Hyper-V server logged on as administrator |
| 01. In Server Manager, on the Manage menu, click Add Roles and Features. |
| 02. On the Before you begin page, verify that your destination server and network environment are prepared for the role and feature you want to install. Click Next. |
| 03. On the Select installation type page, select Role-based or feature-based installation and then click Next. |
| 04. On the Select destination server page, select a server from the server pool and then click Next. |
| 05. On the Select server roles page, select Hyper-V. |
| 06. To add the tools that you use to create and manage virtual machines, click Add Features, and click Next. |
| 07. On the Features page, click Next. |
| 08. On the Hyper-V page, click Next |
| 09. On the Create Virtual Switches page, click Next |
| 10. On the Virtual Machine Migration page, click Next |
| 11. On the Default Stores page, click Next |
| 12. On the Confirm installation selections page, select Restart the destination server automatically if required. |
| 13. On the Add Roles and Features Wizard message, click Yes and them Install |
| 14. When the server reboots, open the Server Manager so the installation can finish. Once done, click close |

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

##Install the Hyper-V Role

Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart

#Machine will reboot, to validate if the hyper-v role was installed, use

Get-WindowsFeature -name Hyper-V

* 1. Downloading Software

Once we have our Hyper-V host configured, it is time to download the required software and create the virtual machines, so Perform this task on the the Hyper-V server logged on as administrator:

|  |
| --- |
| 01. Open PowerShell (run as administrator) and navigate to C:\Trainingfiles\Scripts |
| 02. Execute .\DownloadSoftware.ps1  Note: If anti-virus software has been installed on the Hyper-V host, it is recommended to add an exclusion for C:\TrainingFiles otherwise it will identify the C:\TrainingFiles\Source\Eicar\eicar test file.txt as a Virus. This file is not a virus, it is an industry standard test file for antivirus engines. More information can be found at <http://www.eicar.org/> |
| 03. Copy the App-v5.0 SP3 Client (appv\_client\_setup.exe)[[1]](#footnote-2) to C:\Trainingfiles\Source\App-v5 Client |

* 1. Creating Windows Virtual Machines

|  |
| --- |
| Perform this task on the Hyper-V server logged on as administrator |
| 01. Open PowerShell (run as administrator) and navigate to C:\Trainingfiles\Scripts |
| 02. Execute .\CreateVMs.ps1 |

* 1. CLASSROOM-ROUTER01

|  |
| --- |
| Perform this task on the router01 virtual machine |
| 01. Boot Virtual Machine CLASSROOM-ROUTER01 |
| 02. On Vyos – Boot Menu select Live (amd64-vyos) and press Enter |
| 02. Log in using vyos as login and password |
| 03. Type install image and press enter |
| 04. On Would you like to continue, press enter |
| 05. On Partition, press enter |
| 06. On Install the image on, press enter |
| 07. On Continue, type Yes and press enter |
| 08. On How big of a root partition should I create, press enter |
| 09. On What would you like to name this image, press enter |
| 10. On Which one should I copy to sda, press enter |
| 11. On Enter password for user ‘vyos’, type Pa$$w0rd and press enter |
| 12. On Retype password for user ‘vyos’ type Pa$$w0rd and press enter |
| 13. On Which drive should grub modify the boot partition on, press enter |
| 14. Type poweroff and press enter |
| 15. On Proceed with poweroff, type Yes and press enter |
| 16. Select Media -> DVD Drive -> Eject vyos-rolling-latest.iso and power on the virtual machine |
| 17. Log on with login vyos and password Pa$$w0rd |
| 18. Type configure and press enter |
| 19. Type set interface ethernet eth0 address dhcp and press enter |
| 20. Type set interface ethernet eth0 description 'External' and press enter |
| 21. Type set interface ethernet eth1 address 192.168.3.254/24 and press enter |
| 22. Type set interface ethernet eth1 description 'Internal' and press enter |
| 23. Type set system name-server 8.8.8.8 and press enter |
| 24. Type set system name-server 8.8.4.4 and press enter |
| 25. Type set system host-name router01 and press enter |
| 26. Type set nat source rule 100 outbound-interface 'eth0' and press enter |
| 27. Type set nat source rule 100 source address '192.168.3.0/24' and press enter |
| 28. Type set nat source rule 100 translation address masquerade and press enter |
| 29. Type commit and press enter |
| 30. Type save and press enter |
| 31. Type exit and press enter |
| 32. Type show interfaces and press enter |
| 33. Type ping www.google.com and press enter |

* 1. CLASSROOM-SRV0001

|  |
| --- |
| Perform this task on the srv0001 virtual machine |
| 01. Confirm the Virtual Machine CLASSROOM-ROUTER01 is up and is providing internet connectivity |
| 02. Boot Virtual Machine CLASSROOM-SRV0001 |
| 03. Log on as administrator |
| 04. Open PowerShell (run as administrator) and navigate to C:\Trainingfiles\Scripts |
| 05. Type.\SRV0001.ps1 and press Enter |
| 06. Type .\SRV0001-01-InstallDC.ps1 and press Enter  Note: The computer will restart automatically |
| 07. Log on as administrator, Open PowerShell (run as administrator) and navigate to C:\Trainingfiles\Scripts |
| 08. Type .\SRV0001-02-ConfigureDC.ps1 and press Enter |

* 1. CLASSROOM-SRV0002

|  |
| --- |
| Perform this task on the srv0002 virtual machine as administrator |
| 01. Confirm the Virtual Machine CLASSROOM-ROUTER01 is up and is providing internet connectivity |
| 02. Confirm the Virtual Machine CLASSROOM-SRV0001 is up and has been configured as Domain Controller |
| 03. Boot Virtual Machine CLASSROOM-SRV0002 |
| 04. Log on as classroom\administrator |
| 05. Open PowerShell (run as administrator) and navigate to C:\Trainingfiles\Scripts |
| 06. Type .\SRV0002.ps1 and press Enter  Note: Computer will shutdown |

* 1. CLASSROOM-WKS0001

|  |
| --- |
| Perform this task on the wks0001 virtual machine as administrator |
| 01. Confirm the Virtual Machine CLASSROOM-ROUTER01 is up and is providing internet connectivity |
| 02. Confirm the Virtual Machine CLASSROOM-SRV0001 is up and has been configured as Domain Controller |
| 03. Boot Virtual Machine CLASSROOM-WKS0001 |
| 04. Log on as classroom\administrator |
| 05. Open PowerShell (run as administrator) |
| 06. Type Set-ExecutionPolicy Unrestricted -force and press Enter |
| 07. Type \\srv0001\Trainingfiles\Scripts\WKS0001.ps1 and press Enter  Note: Computer will shutdown |

* 1. CLASSROOM-WKS0002

|  |
| --- |
| Perform this task on the wks0002 virtual machine as administrator |
| 01. Confirm the Virtual Machine CLASSROOM-ROUTER01 is up and is providing internet connectivity |
| 02. Confirm the Virtual Machine CLASSROOM-SRV0001 is up and has been configured as Domain Controller |
| 03. Boot Virtual Machine CLASSROOM-WKS0002 |
| 04. Log on as classroom\administrator |
| 05. Open PowerShell (run as administrator) |
| 06. Type Set-ExecutionPolicy Unrestricted -force and press Enter |
| 07. Type \\srv0001\Trainingfiles\Scripts\WKS0002.ps1 and press Enter  Note: Computer will shutdown |

* 1. CLASSROOM-WKS0004

|  |
| --- |
| Perform this task on the wks0004 virtual machine as administrator |
| 01. Confirm the Virtual Machine CLASSROOM-ROUTER01 is up and is providing internet connectivity |
| 02. Confirm the Virtual Machine CLASSROOM-SRV0001 is up and has been configured as Domain Controller |
| 03. Boot Virtual Machine CLASSROOM-WKS0004 |
| 04. Log on as classroom\administrator |
| 05. Open PowerShell (run as administrator) |
| 06. Type Set-ExecutionPolicy Unrestricted -force and press Enter |
| 07. Type \\srv0001\Trainingfiles\Scripts\WKS0004.ps1 and press Enter  Note: Computer will shutdown |

1. Active Directory

|  |  |
| --- | --- |
| Computers used in  this Lab | SRV0001 |
| More information | Schema extensions for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/network/schema-extensions> |
| Description | In this chapter, we will extend the active directory schema for MECM. The Extension adds a new container and new attributes to the AD schema, allowing the MECM environment to publish information to the AD and domain-joined machines to query these information. Extending the schema is not required task, however, it can simplify the deployment, configuration and management of a MECM environment. |

* 1. Extending Active Directory Schema

|  |
| --- |
| Perform this task on the srv0001 virtual machine logged on as administrator |
| 01. Open Windows Explorer |
| 02. Navigate to C:\trainingfiles\Source\MECMCB\Extract\SMSSETUP\BIN\X64 |
| 03. Execute extadsch.exe (run as administrator) |
| 03. Review the extending of the schema by examining the content of the C:\extadsch.log file  Note: If the schema has already been extended, the C:\extadsch.log will print lines with already exists information. |

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

#Extend Schema

Start-Process -Filepath ('C:\trainingfiles\Source\MECMCB\Extract\SMSSETUP\BIN\X64\extadsch.exe') -wait -NoNewWindow

Start-Sleep 30

#Confirm Schema Extension

$schema = [DirectoryServices.ActiveDirectory.ActiveDirectorySchema]::GetCurrentSchema()

start-sleep 15

$schema.RefreshSchema()

$schema.FindClass("mSSMSSite")

* 1. Creating the System Management Container

|  |
| --- |
| Perform this task on the srv0001 virtual machine logged on as administrator |
| 01. Open Administrative Tools and then ADSI Edit. |
| 02. Right Click ADSI Edit and click Connect to… |
| 03. On Connection Settings, click Ok |
| 04. Expand Default naming context [SRV0001.classroom.intranet] -> DC=classroom,DC=intranet and then select CN=System |
| 05. right click CN=System and choose New -> Object |
| 06. Choose object class as Container, click Next |
| 07. In Value, type System Management and click Next |
| 08. Click Finish |
| 09. Back on the ADSI Edit screen, right click CN=System Management and choose Properties  Note: A refresh of the list of objects may be necessary |
| 10. Select the Security tab |
| 11.Click on the Advanced button |
| 12. On Advanced Security Settings for System Management, Click Add |
| 13. On Permission Entry for System Management, click Select a Principal |
| 14. On Select Users, Computers, Service Account, or Group, type MECM Servers and click Check Names. Click OK |
| 15. On Permission Entry for System Management, select Allow Full Control under permissions, and This object and all descendant objects under Apply to.  Click Ok three (3) times. |

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

Import-Module ActiveDirectory

$root = (Get-ADRootDSE).defaultNamingContext

if (!([adsi]::Exists("LDAP://CN=System Management,CN=System,$root"))) {

$smcontainer = New-ADObject -Type Container -name "System Management" -Path "CN=System,$root" -Passthru

}

$acl = get-acl "ad:CN=System Management,CN=System,$root"

$objGroup = Get-ADGroup -filter {Name -eq "MECM Servers"}

$All = [System.DirectoryServices.ActiveDirectorySecurityInheritance]::SelfAndChildren

$ace = new-object System.DirectoryServices.ActiveDirectoryAccessRule $objGroup.SID, "GenericAll", "Allow", $All

$acl.AddAccessRule($ace)

Set-acl -aclobject $acl "ad:CN=System Management,CN=System,$root"

1. Pre-Requirements

|  |  |
| --- | --- |
| Computers used in  this Lab | SRV0001  SRV0002 |
| More information | Site and site system prerequisites for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/configs/site-and-site-system-prerequisites>  Prepare Windows Servers to support Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/network/prepare-windows-servers>  Log files in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/hierarchy/log-files> |
| Description | In this chapter, we will prepare the environment for the installation of the MECM as well as register the default log viewer for CMTrace |

* 1. Installing .NET Framework 3.5

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open Server Manager |
| 02. Click Manage and Add Roles and Features |
| 03. Before you begin, click Next |
| 04. Select Role-based or feature-based installation and click Next |
| 05. Select select a server from the server pool and select the server “SRV0002.classroom.intranet” and click Next |
| 06. Under select server roles, keep the default, and click Next |
| 07. Under features, expand .NET Framework 3.5 Features and select .NET Framework 3.5 (includes .NET 2.0 and 3.0). Click Next |
| 08. Under Confirm installation selections, click Specify an alternate source path |
| 09. Under Specify alternate source path, type \\srv0001\Trainingfiles\Source\WS2016\sources\sxs and click Ok. One back, Click Install |
| 10. Once the installation is succeeded. Click Close |

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

Get-WindowsFeature -Name Net-Framework-Core | Install-WindowsFeature -source '\\srv0001\Trainingfiles\Source\WS2019\sources\sxs'

* 1. Validating .NET Framework 4.6 Installation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open Server Manager |
| 02. Click Manage and Add Roles and Features |
| 03. Before you begin, click Next |
| 04. Select Role-based or feature-based installation and click Next |
| 05. Select select a server from the server pool and select the server “SRV0002.classroom.intranet” and click Next |
| 06. Under select server roles, keep the default, and click Next |
| 07. Under features, expand .NET Framework 4.6 Features and Confirm that .NET Framework 4.6 is installed. Click Cancel |

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

Get-WindowsFeature -Name NET-Framework-45-Features

* 1. Installing Microsoft Remote Differential Compression

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open Server Manager |
| 02. Click Manage and Add Roles and Features |
| 03. Before you begin, click Next |
| 04. Select Role-based or feature-based installation and click Next |
| 05. Select select a server from the server pool and select the server “SRV0002.classroom.intranet” and click Next |
| 06. Under select server roles, keep the default, and click Next |
| 07. Under features, Select Remote Differential Compression. Click Next |
| 08. Under Confirm installation selections click Install |
| 09. Once the installation is succeeded. Click Close |

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

Get-WindowsFeature -Name RDC | Install-WindowsFeature

* 1. Installing Assessment and Deployment Kit (ADK) for Windows 10 1903

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Execute adksetup.exe (run as administrator) from \\srv0001\Trainingfiles\Source\AdkW10 |
| 02. Under Specify location, click Next |
| 03. Under Windows Kits Privacy, click Next |
| 04. Under License Agreement, click Accept |
| 05. Under Select the features you want to install only the following:   * Deployment Tools * Imaging and Configuration Designer (ICD) * Configuration Designer * User State Migration Tool (USMT)   Click Install |
| 06. Once the installation is completed, click Close |

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

Start-Process -Filepath ("\\srv0001\TrainingFiles\Source\AdkW10\adksetup.exe") -ArgumentList ("/Features OptionId.DeploymentTools OptionId.ImagingAndConfigurationDesigner OptionId.UserStateMigrationTool /norestart /quiet /ceip off") -wait -NoNewWindow

start-sleep 5

* 1. Installing Assessment and Deployment Kit (ADK) WinPE AddOn

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Execute adkwinpesetup.exe (run as administrator) from \\srv0001\Trainingfiles\Source\AdkW10WinPe |
| 02. Under Specify location, click Next |
| 03. Under Windows Kits Privacy, click Next |
| 04. Under License Agreement, click Accept |
| 05. Under Select the features you want to install only the following:   * Windows Preinstallation Environment (Windows PE)   Click Install |
| 06. Once the installation is completed, click Close |

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

Start-Process -Filepath ("\\srv0001\TrainingFiles\Source\AdkW10WinPe\adkwinpesetup.exe") -ArgumentList ("/Features OptionId.WindowsPreinstallationEnvironment /norestart /quiet /ceip off") -wait -NoNewWindow

start-sleep 5

* 1. Installing Assessment and Deployment Kit (ADK) FIX

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Copy the file ImageCat.exe and ImgMgr.exe from \\srv0001\TrainingFiles\Source\AdkW10Fix to C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools\WSIM\ |

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

Copy-Item -Path '\\srv0001\TrainingFiles\Source\AdkW10Fix\ImageCat.exe' -Destination 'C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools\WSIM\' -Force

Copy-Item -Path '\\srv0001\TrainingFiles\Source\AdkW10Fix\ImgMgr.exe' -Destination 'C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools\WSIM\' -Force

* 1. Logs Reading

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Copy CMTrace.exe from \\srv0001\Trainingfiles\Source\MECMCB\Extract\SMSSETUP\TOOLS to C:\Windows |
| 02. Execute C:\Windows\CMTrace.exe |
| 03. When asked Do you want to make this program the default viewer for log files?, click Yes. |
| 04. Close Configuration Manager Trace Log Tool  Note: Repeat the process on every machine that you want to read MECM (or other) logfiles using CMTrace  Note: With MECM 1806, CMTrace is now copied by default during the client installation to c:\Windows\CCM\CMTrace.exe |

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

Copy-Item '\\srv0001\Trainingfiles\Source\MECMCB\Extract\SMSSETUP\TOOLS\CMTrace.exe' 'C:\windows\cmtrace.exe'

$executecmtrace = "`"$($env:windir)\CMTrace.exe`" `"%1`""

New-PSDrive -Name HKCR -PSProvider Registry -Root HKEY\_CLASSES\_ROOT | Out-Null

New-Item -Path "hkcr:Local Settings\Software\Microsoft\Windows\Shell\MuiCache" -Force | Out-Null

Set-ItemProperty -Path "hkcr:Local Settings\Software\Microsoft\Windows\Shell\MuiCache" -Name ($env:windir + '\CMTrace.exe') -Value 'Configuration Manager Trace Log Tool' | Out-Null

New-Item -Path "hkcr:.lo\_" -Force | Out-Null

Set-ItemProperty -Path "hkcr:.lo\_" -Name '(Default)' -Value 'Log.File' | Out-Null

New-Item -Path "hkcr:.log" -Force | Out-Null

Set-ItemProperty -Path "hkcr:.log" -Name '(Default)' -Value 'Log.File' | Out-Null

New-Item -Path "hkcr:Log.File" -Force | Out-Null

New-Item -Path "hkcr:Log.File\Shell" -Force | Out-Null

New-Item -Path "hkcr:Log.File\Shell\Open" -Force | Out-Null

New-Item -Path "hkcr:Log.File\Shell\Open\Command" -Force | Out-Null

Set-ItemProperty -Path "hkcr:\Log.File\shell\open\command" -Name '(Default)' -Value $executecmtrace | Out-Null

New-Item -Path "hkcu:Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache" -Force | Out-Null

Set-ItemProperty -Path "hkcu:Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache" -Name ($env:windir + '\CMTrace.exe') -Value 'Configuration Manager Trace Log Tool' | Out-Null

New-Item -Path "hkcu:Software\Classes\.lo\_" -Force | Out-Null

Set-ItemProperty -Path "hkcu:Software\Classes\.lo\_" -Name '(Default)' -Value 'Log.File' | Out-Null

New-Item -Path "hkcu:Software\Classes\.log" -Force | Out-Null

Set-ItemProperty -Path "hkcu:Software\Classes\.log" -Name '(Default)' -Value 'Log.File' | Out-Null

New-Item -Path "hkcu:Software\Classes\Log.File" -Force | Out-Null

New-Item -Path "hkcu:Software\Classes\Log.File\Shell" -Force | Out-Null

New-Item -Path "hkcu:Software\Classes\Log.File\Shell\Open" -Force | Out-Null

New-Item -Path "hkcu:Software\Classes\Log.File\Shell\Open\Command" -Force | Out-Null

Set-ItemProperty -Path "hkcu:Software\Classes\Log.File\shell\open\command" -Name '(Default)' -Value $executecmtrace | Out-Null

Start-Process -Filepath ('C:\windows\cmtrace.exe')

1. SQL Server

|  |  |
| --- | --- |
| Computers used in  this Lab | SRV0001  SRV0002 |
| More information | Support for SQL Server versions for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/configs/support-for-sql-server-versions>  International support in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/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 MECM. 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 mecmadmin |
| 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 mecmadmin |
| 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 MECM 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 MECM 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\MECM 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 -NoNewWindow

Start-sleep 30

* 1. Installing SQL Server 2017 latest CU

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Execute SQLServer2017-KB4515579-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-KB4515579-x64.exe") -ArgumentList ('/quiet /IAcceptSQLServerLicenseTerms /Action=Patch /AllInstances') -wait -NoNewWindow

* 1. Installing SQL Server Reporting Services

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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\SQLServerReportingServices.exe") -ArgumentList ('/IAcceptLicenseTerms /Quiet /Norestart') -wait -NoNewWindow

* 1. Configuring SQL Server Reporting Services

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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'"

import-module "C:\SQLServer (x86)\140\Tools\PowerShell\Modules\SQLPS\SQLPS.psd1" -Force

start-sleep 5

##create database

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

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

##add rights

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

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

$rsConfig.SetDatabaseConnection('SRV0002', 'ReportServer', 0, 'classroom\mecmadmin','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 mecmadmin |
| 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 -NoNewWindow

* 1. Validating Installation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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-CU17) (KB4515579) - 14.0.3238.1 (X64) Sep 13 2019 15:49:57 Copyright (C) 2017 Microsoft Corporation Enterprise Evaluation Edition (64-bit) on Windows Server 2019 Standard Evaluation 10.0 <X64> (Build 17763: ) (Hypervisor) |
| 04. Now, type SELECT SERVERPROPERTY('productversion'), SERVERPROPERTY('productlevel'), SERVERPROPERTY('edition') and click Execute  Note: the result should be something like 14.0.3238.1 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 mecmadmin |
| 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 MECM 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 mecmadmin |
| 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 mecmadmin |
| 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: MECM 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 }

1. Installation of MECM Primary Site

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | Supported configurations for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/configs/supported-configurations>  List of Prerequisite Checks for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/install/list-of-prerequisite-checks>  Install Configuration Manager sites  <https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/install/installing-sites> |
| Description | In this chapter, we will install MECM as well as validate if the installation has occurred correctly. We will also be installing the Configuration Manager Toolkit, a set of extra tools to help administration of the MECM environment. |

* 1. Downloading MECM Setup Files

|  |
| --- |
| Perform this task on the SRV0001 virtual machine logged on as administrator |
| 01. Execute setupdl.exe from C:\Trainingfiles\Source\MECMCB\Extract\SMSSETUP\BIN\X64  Note: This step was already executed on section 4.4 and can be safely ignored |
| 02. Once the Configuration Manager Setup download is loaded, type C:\Trainingfiles\Source\MECMCB\Redist and click download |
| 03. Once the download is completed, examine the \\srv0001\Trainingfiles\Source\MECMCB\Redist folder |

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

Start-Process -Filepath ("\\srv0001\Trainingfiles\Source\MECMCB\Extract\SMSSETUP\BIN\X64\SETUPDL.exe") -ArgumentList ("\\srv0001\Trainingfiles\Source\MECMCB\Redist") -wait -NoNewWindow

* 1. Pre-requirements Check

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open Command Prompt as administrator |
| 02. type \\srv0001\Trainingfiles\Source\MECMCB\Extract\SMSSETUP\BIN\X64\prereqchk.exe /pri /sql SRV0002.classroom.intranet /sdk SRV0002.classroom.intranet and press enter |
| 03. Once the Installation prerequisite check completed, confirm that there are no errors. Click OK  Note: You should see the following warnings:   * WSUS on site Server * Verify site server permission to publish to active directory * SQL Server process memory allocation |
| 04. You can also review the errors and warnings by examining the content of the C:\ConfigMgrPrereq.log file |

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

Start-Process -Filepath ("\\srv0001\Trainingfiles\Source\MECMCB\Extract\SMSSETUP\BIN\X64\prereqchk.exe") -ArgumentList ("/pri /sql SRV0002.classroom.intranet /sdk SRV0002.classroom.intranet") -wait -NoNewWindow

* 1. Creating Firewall Rules for MECM Site Server Console Communication

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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 135 under specify local ports and click Next |
| 05. On Action, click Next |
| 06. On Profile, click Next |
| 07. On Name, type MECM Console (TCP 135) Inbound and click Finish |

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

New-NetFirewallRule -DisplayName "MECM Console (TCP 135) Inbound" -Action Allow -Direction Inbound -LocalPort 135 -Protocol TCP

* 1. Site Server Installation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Execute splash.hta from \\srv0001\Trainingfiles\Source\MECMCB\Extract |
| 02. Under Microsoft Endpoint Configuration Manager, click Install |
| 03. On Before You Begin, click Next |
| 04. On Available Setup Options, click Install a Configuration Manager Primary Site and click Next |
| 05. Select Install the evaluation edition of this product and click Next |
| 06. Under Product License Terms, select I accept for all products and click Next |
| 07. Under Prerequisite Downloads, select Use previously downloaded files and in path type \\srv0001\Trainingfiles\Source\MECMCB\Redist and click Next |
| 08. Under server language selection, leave the default selection and click Next |
| 09. Under Client Language selection leave the default selection and click Next |
| 10. Under Site and Installation Settings, use the following information:   * Site Code: 001 * Site Name: Training Lab * Installation Folder: C:\ConfigMgr * Install the Configuration Manager Console: Checked   Click Next  Note: In production, MECM should not be installed on the System Drive. It is only being installed here on the System Drive because it is a lab environment.  Note: A file called NO\_SMS\_ON\_DRIVE.SMS should be used to exclude MECM from using a specific drive. Fore more information, refer to <https://technet.microsoft.com/en-us/library/bb632890.aspx>  Note: Site Code, Site Name and Installation Folder cannot be changed after the installation. |
| 12. Under Primary Site Installation select Install the primary site as a stand-alone site and click Next |
| 13. On the Configuration Manager question window, click Yes |
| 14. Under Database Information leave the default and click Next |
| 15. Under Database Information (Path to the SQL files) leave the default and click Next |
| 16. Under SMS Provider Settings, leave the default and click Next |
| 17. Under Client Computer Communication Settings select Configure the communication method on each site system role and click Next |
| 18. Under Site System Roles uncheck Install a management point and Install a distribution Point and click Next  Note: These site system roles are going to be installed later |
| 19. Under Diagnostic and Usage Data, click Next |
| 20. Under Service Connection Point Setup, leave the default and click Next |
| 21. Under Settings summary, review the settings and click Next |
| 22. The prerequisite check will validate the system. Once it is done, click Begin Install |
| 23. Once the installation is completed, click Close.  Note: Installation takes about half an hour. During the installation process, you can press the View Log button multiple times and examine the progress of the installation by reviewing the log file status. |
| 24. At the root of C-partition, multiple log files are created that tell the status of the installation:   * ConfigMgrAdminUISetup.log: MECM console installation log * ConfigMgrPrereq.log: Prerequisites review log * ConfigMgrSetup.log: site server installation log * ConfigMgrSetupWizard.log: installation wizard log * smstsvc.log: installation program log (the errors in it can be ignored) |

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

$inifile = @"

[Identification]

Action=InstallPrimarySite

[Options]

ProductID=EVAL

SiteCode=001

SiteName=Training Lab

SMSInstallDir=c:\ConfigMgr

SDKServer=SRV0002.classroom.intranet

RoleCommunicationProtocol=HTTPorHTTPS

ClientsUsePKICertificate=0

PrerequisiteComp=1

PrerequisitePath=\\srv0001\Trainingfiles\Source\MECMCB\Redist

MobileDeviceLanguage=0

AdminConsole=1

JoinCEIP=0

[SQLConfigOptions]

SQLServerName=SRV0002.classroom.intranet

SQLServerPort=1433

DatabaseName=CM\_001

SQLSSBPort=4022

SQLDataFilePath=C:\SQLServer\MSSQL14.MSSQLSERVER\MSSQL\DATA\

SQLLogFilePath=C:\SQLServer\MSSQL14.MSSQLSERVER\MSSQL\DATA\

[CloudConnectorOptions]

CloudConnector=1

CloudConnectorServer=SRV0002.classroom.intranet

UseProxy=0

ProxyName=

ProxyPort=

[SystemCenterOptions]

SysCenterId=Lzyga7QBe84u7mZvvIcFmoh9fWeQymoIYs0Cvqz4yhU=

[HierarchyExpansionOption]

"@

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

Start-Process -Filepath ("\\srv0001\TrainingFiles\Source\MECMCB\Extract\SMSSETUP\BIN\X64\setup.exe") -ArgumentList ('/script "\\srv0001\TempFiles\installcmcb.ini"') -wait -NoNewWindow

* 1. Installation Status

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Services console |
| 02. Services for MECM are still named with SMS-prefix.  Note the existence of:   * SMS\_EXECUTIE * SMS\_SITE\_BACKUP * SMS\_SITE\_COMPONENT\_MANAGER * SMS\_SITE\_SQL\_BACKUP * SMS\_SITE\_VSS\_WRITER   Note: All services should be with Status running and the startup type as Automatic. The only exception is the SMS\_SITE\_BACKUP, that will set to Manual and will be started only when needed by MECM |
| 03. Open Windows Explorer and navigate to C:\ConfigMgr. |
| 04. Open Local Users and groups and navigate to groups |
| 05. Multiple groups exist for MECM, and they have a prefix of ConfigMgr or SMS.  Note: The following groups should be created:   * ConfigMgr\_CollectedFilesAccess * ConfigMgr\_DViewAccess * SMS Admins * SMS\_SiteSystemToSiteServerConnection\_MP\_<SITECODE> * SMS\_SiteSystemToSiteServerConnection\_SMSProv\_<SITECODE> * SMS\_SiteSystemToSiteServerConnection\_Stat\_<SITECODE> * SMS\_SiteToSiteConnection\_<SITECODE> |
| 06. On SRV0001 (Domain Controller), open ADSI Edit |
| 07. Expand Default naming context, DC=CLASSROOM,DC=intranet, CN=System, CN=System Management  Note: Existence of CN=SMS-Site-001 record from mSSMSSite class |

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

Get-CimInstance win32\_service | where-object {$\_.Name -in ("SMS\_EXECUTIVE","SMS\_SITE\_BACKUP","SMS\_SITE\_COMPONENT\_MANAGER","SMS\_SITE\_SQL\_BACKUP","SMS\_SITE\_VSS\_WRITER")} | select Name,StartMode,State,Status

$dn = New-Object System.DirectoryServices.DirectoryEntry

$dsLookFor = new-object System.DirectoryServices.DirectorySearcher($dn)

$dsLookFor.Filter = ("CN=SMS-SITE-001")

$dsLookFor.SearchScope = "subtree";

$dsLookFor.findOne()

Get-ChildItem -Path C:\ConfigMgr

$Groups = Gwmi win32\_group | where { $\_.Name -in ("ConfigMgr\_CollectedFilesAccess", "ConfigMgr\_DViewAccess", "SMS Admins", "SMS\_SiteSystemToSiteServerConnection\_MP\_001", "SMS\_SiteSystemToSiteServerConnection\_SMSProv\_001", "SMS\_SiteSystemToSiteServerConnection\_Stat\_001", "SMS\_SiteToSiteConnection\_001") } | select Name

if ($Groups.Count -ne 7) { Write-Host "Should be 7 Groups" } else { "Groups OK" }

* 1. Console Overview

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console. As the Management Console starts, it is good to notice that the new console is not MMC-based. |
| 02. Let’s look at few main points:   * Console uses new Ribbon-user interface (Originally introduced in Office 2007). * Bottom left corner has a list of main views:   + Assets and Compliance: here is information on machines and users as well as their settings   + Software Library: includes application management, software updates and operating systems deployment related tasks   + Monitoring: information on the status of MECM and reporting related settings   + Administration: MECM environment and configuration related settings   + Community: Up to date documentation and support articles. |
| 03. Click Assets and Compliance. Here are the settings for:   * Users: Manage users and user groups for the hierarchy * Devices: Manage devices for the hierarchy * User Collections: Manage user collection for the hierarchy * Device Collections: Manage device collection for the hierarchy * User State Migration: Manage user state migration for when you deploy operating system * Asset Intelligence: Manage the Asset Intelligence catalog, import license files, and synchronize with System Center Online to reconcile software licenses. * Software Metering: Configure rules to monitor software application usage. * Compliance Settings: Manage configuration items and configuration baselines to assess and remediate the compliance of settings on devices. * Endpoint Protection: Manage Antimalware and Firewall policies. * All Corporate-owned Devices: Manage Corporate-owned Devices and Device Enrollment Profiles |
| 04. Click Devices. Note that ribbon bar functions update to relevant tasks. This view shows all devices know to MECM. Right-click Devices and look at the available options. From the right side of the windowpane, right-click SRV0002 and notice the options. Ribbon displays options from both left side selection e.g. Devices and right-side selection SRV0002. |
| 05. Click Software Library. There are 6 levels:   * Application Management: Manage application deployments for users and devices, and configure global conditions for all applications in the hierarchy. * Software Updates: Manage software updates, software update groups, deployment packages for software updates, and automatic deployment rules * Operating Systems: Manage drivers, operating system images, upgrade packages, boot images, and task sequences to deploy operating systems and virtual hard disks. * Windows 10 Servicing: Manage Servicing for Windows 10 * Office 365 Client Management: Monitor amd Manage Office 365 clients * Scripts: Create and run PowerShell scripts on devices and collections |
| 06. Expand Application Management. There are 9 sublevels:   * Applications: Manage and deploy applications to users and devices, and configure rules to install and uninstall applications. * License Information for Store Apps: Manage Licensed Store Applications from Windows Store for Business and Apple’s VPP. * Packages: Manage packages that contain the files and instructions to deploy programs to users and devices. * Approval Requests: Manage application requests from users for Software Center applications that require approval. * Global Conditions: Manage global conditions for all applications in the site hierarchy. * App-V Virtual Environments: Virtual Environment * Windows Sideloading Key: Windows Sideloading Keys * Application Management Policies: Configure application management policies for the hierarchy. * App Configuration Policies: Manage app configuration policies. |
| 07. Expand Software Update. There are 5 sublevels:   * All Software Updates: Synchronize, configure, download, and deploy software updates. * Software Update Groups: Manage software updates as a group * Deployment Packages: Manage software update deployment packages * Automatic Deployment Rules: Manage rules that automatically identify, download, add to a software update group, and optionally deploy software updates that meet specific criteria. * Third-Party Software Update Catalogs: Third-Party Software Update Catalogs |
| 08. Expand Operating Systems. There are 6 sublevels:   * Drivers: Manage device drivers and device driver catalogs to deploy operating systems * Driver Packages: Manage device driver packages. * Operating System Images: Manage Windows image files for operating system deployment * Operating System Upgrade Packages: Manage operating system upgrade packages * Boot Images: Manage boot images for operating system deployment. * Task Sequences: Manage task sequences that automate steps or tasks on client computers. |
| 09. Expand Windows 10 Servicing. There are 3 sublevels:   * All Windows 10 Updates: Manage Updates for Windows 10. * Servicing Plans: Manage servicing plans for Windows 10. * Windows Update for Business Policies: Manage Windows Update for Business policies for devices connected directly to the Windows Update service |
| 10. Click Monitoring. There are 22 levels here:   * Alerts: View and manage alerts. * Queries: View and manage Configuration Manager queries. * Reporting: View and manage reports and report subscriptions, and configure report options. * Site Hierarchy: View and manage the status of all sites in the hierarchy by using a hierarchy diagram or a geographical view. The geographical view of the hierarchy requires a web browser and access to the Internet. * System Status: View and manage site status, component status, conflicting records, and status message queries. * Deployments: View information about the status of deployed software. * Phased Deployments: View information about the status of phased deployments. * Client Operations: View client operation details. * Script Status: Script status * Client Status: View and configure options for client status. * Database Replication: View site-to-site link status. * Distribution Status: View content status, distribution point status, and distribution point configuration status. * Software Update Point Synchronization Status: View software update point synchronization status across the hierarchy. * Site Server Status: Site Server Status * Updates and Servicing Status: View the status of Configuration Manager updates you’ve installed in your hierarchy. * Security: View Endpoint Protection and Health Attestation details. * Upgrade Readiness: Analyze device compatibility with Windows 10 to facilitate upgrades. * Compliance Settings: Compliance Settings Page. * Surface Devices: View information about Surface devices in your environment * Co-Management: View co-management information in your environment * Cloud Management: View information about Cloud Managed devices in your environment * Package Conversion Status: Package Conversion Status |
| 11. Click Administration. There are 10 main levels:   * Updates and Servicing: Updates and Servicing will let you install recent updates as they become available. * Hierarchy Configuration: Manage boundaries, site-to-site communication, discovery methods, Active Directory forest and Exchange Server connection settings. * Cloud Services: Manage subscriptions to cloud services in your hierarchy. * Site Configuration: Manage servers and site system roles, components, site maintenance, and status configuration * Client Settings: Configure default and custom client settings. * Security: Manage administrative users, security roles, security scopes, certificates, and accounts that you configure in the Configuration Manager console. * Distribution Points: Manage individual distribution points and configuration properties, and view disk space capacity. * Distribution Point Groups: Manage distribution points as a group * Migration: Manage migration of data from sites in a Configuration Manager hierarchy to sites in this Configuration Manager hierarchy. * Management Insights: Better understand your environment and take action, based on analysis of data from the site database   Note: Only the user that performed the installation will have permissions in MECM by default. You need to remember to give other users permissions. |
| 12. Click on File->Connect via PowerShell |
| 13. On Do you want to run Software from this untrusted publisher? Type A and enter |
| 14. Type Get-Module -Name ConfigurationManager | select Version and confirm the version is 5.1902.1085.1700 |

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

#Step 12, 13 and 14 only

$ModulePath = $env:SMS\_ADMIN\_UI\_PATH

if ($ModulePath -eq $null) {

$ModulePath = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment").SMS\_ADMIN\_UI\_PATH

}

$ModulePath = $ModulePath.Replace("bin\i386","bin\ConfigurationManager.psd1")

$Certificate = Get-AuthenticodeSignature -FilePath "$ModulePath" -ErrorAction SilentlyContinue

$CertStore = New-Object System.Security.Cryptography.X509Certificates.X509Store("TrustedPublisher")

$CertStore.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::MaxAllowed)

$Certexist = ($CertStore.Certificates | where {$\_.thumbprint -eq $Certificate.SignerCertificate.Thumbprint}) -ne $null

if ($Certexist -eq $false) {

$CertStore.Add($Certificate.SignerCertificate)

}

$CertStore.Close()

import-module $ModulePath -force

Get-Module -Name ConfigurationManager | select Version

Remove-Module ConfigurationManager -Force

* 1. Validating Installation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Under monitoring, expand System Status and click Component Status |
| 03. Search for SMS\_DATABASE\_NOTIFICATION\_MONITOR |
| 04. Right Click SMS\_DATABASE\_NOTIFICATION\_MONITOR, Show Messages and click All |
| 05. Under Status Messages: Set Viewing Period, click OK |
| 06. Verify the existence of Message ID 2420  Note: During the installation, these messages are normal, however, it should not occur after the installation |
| 07. Double click any 2420 messages to see its details. Once done, click Ok |
| 08. Search for SMS\_SITE\_SQL\_BACKUP |
| 09. Right Click SMS\_SITE\_SQL\_BACKUP, Show Messages and click All |
| 10. Under Status Messages: Set Viewing Period, click OK |
| 11. Verify the existence of Message ID 4959  Note: If this Message ID exist, there is a problem with the SQL Server Account SPN. Refer to <https://docs.microsoft.com/en-us/sccm/core/servers/manage/modify-your-infrastructure#bkmk_SPN> for more information |
| 12. Search for SMS\_SITE\_COMPONENT\_MANAGER |
| 13. Right Click SMS\_SITE\_COMPONENT\_MANAGER, Show Messages and click All |
| 14. Under Status Messages: Set Viewing Period, click OK |
| 15. Verify the existence of Message ID 1027 |
| 16. Double click the message to see its details. Once done, click Ok |
| 17. Search for SMS\_HIERARCHY\_MANAGER |
| 18. Right Click SMS\_HIERARCHY\_MANAGER, Show Messages and click All |
| 19. Under Status Messages: Set Viewing Period, click OK |
| 20. Verify the existence of Message ID 3306 |
| 21. Double click any of the messages to see its details. Once done, click Ok |
| 22. Verify the existence of Message ID 3323  Note: If this Message ID exist, the MECM Server will accept HTTP or HTTPS connections. |
| 23. Double click any of the messages to see its details. Once done, click Ok |
| 24. Verify the existence of Message ID 3351  Note: If this Message ID exist, MECM found there is an enabled firewall rule for TCP ports for SQL Server (TCP 1433) & SQL Server Service Broker (TCP 4022). |
| 25. Double click any of the messages to see its details. Once done, click Ok |
| 26. Verify the existence of Message ID 3353  Note: If this Message ID exist, confirm there is an enabled firewall rule for TCP ports for SQL Server (TCP 1433) & SQL Server Service Broker (TCP 4022). |
| 27. Double click any of the messages to see its details. Once done, click Ok |
| 28. Verify the existence of Message ID 4909  Note: If this Message ID exist, the container System Management does not exist. |
| 29. Double click any of the messages to see its details. Once done, click Ok |
| 30. Verify the existence of Message ID 4911  Note: If this Message ID exist, the container System Management have the correct permissions. |
| 31. Double click any of the messages to see its details. Once done, click Ok |
| 32. Verify the existence of Message ID 4912  Note: If this Message ID exist, there is already an object inside the container System Management the MECM cannot update. |
| 33. Double click any of the messages to see its details. Once done, click Ok |
| 34. Verify the existence of Message ID 4913  Note: If this Message ID exist, there is a problem with the security of the System Management container. |
| 35. Double click any of the messages to see its details. Once done, click Ok |
| 36. Search for SMS\_REPLICATION\_CONFIGURATION\_MONITOR |
| 37. Right Click SMS\_REPLICATION\_CONFIGURATION\_MONITOR, Show Messages and click All |
| 38. Under Status Messages: Set Viewing Period, click OK |
| 39. Verify the existence of Message ID 4629 |
| 40. Double click any of the messages to see its details. Once done, click Ok |
| 41. Verify the existence of Message ID 620  Note: If found, check the performance of your SQL Server. |
| 42. Search for SMS\_DMP\_DOWNLOADER |
| 43. Right Click SMS\_DMP\_DOWNLOADER, Show Messages and click All |
| 44. Under Status Messages: Set Viewing Period, click OK |
| 45. Verify the existence of Message ID 4629 |
| 46. Double click any of the messages to see its details. Once done, click Ok |
| 47. Verify the existence of Message ID 9700  Note: If found check the network and/or internet. Also, this message is also normal to be seen during the installation. |
| 48. Verify the existence of Message ID 1104  Note: During the installation, this message is normal, however, it should not occur after the installation |
| 49. Search for SMS\_WINNT\_SERVER\_DISCOVERY\_AGENT |
| 50. Right Click SMS\_WINNT\_SERVER\_DISCOVERY\_AGENT, Show Messages and click All |
| 51. Under Status Messages: Set Viewing Period, click OK |
| 52. Verify the existence of Message ID 4202 |
| 53. Double click any of the messages to see its details. Note the number of system roles found, it should be 5. Once done, click Ok |
| 54. Click Administration. |
| 55. Expand Site Configuration and click Servers and Site System Roles  Note: Confirm that the Count of Roles (ignoring the SMS Provider) match the number at message ID 4202 |

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

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_DATABASE\_NOTIFICATION\_MONITOR' and stmsg.MessageID = 2420 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Warning: Found SMS\_DATABASE\_NOTIFICATION\_MONITOR 2420 id's" -ForegroundColor Yellow

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_SITE\_SQL\_BACKUP' and stmsg.MessageID = 4959 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Error: Missing SQL SPN Information - https://technet.microsoft.com/en-us/library/hh427336.aspx#BKMK\_ManageSPNforDBSrv" -ForegroundColor Red

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_SITE\_COMPONENT\_MANAGER' and stmsg.MessageID = 1027 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_SITE\_COMPONENT\_MANAGER 2017 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 3306 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_HIERARCHY\_MANAGER 3306 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsg.RecordID from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 3323 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_HIERARCHY\_MANAGER 3323 id's"

break

} else { Start-Sleep 10 }

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 3351 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_HIERARCHY\_MANAGER 3351 id's"

break

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 3353 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Error: Found SMS\_HIERARCHY\_MANAGER 3353 id's" -ForegroundColor Red

break

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 4909 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Error: Found SMS\_HIERARCHY\_MANAGER 4909 id's" -ForegroundColor Red

break

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 4911 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_HIERARCHY\_MANAGER 4911 id's"

} else {

Write-Host "ERROR: Not Found SMS\_HIERARCHY\_MANAGER 4911 id's" -ForegroundColor Red

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 4012 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Error: Found SMS\_HIERARCHY\_MANAGER 4012 id's" -ForegroundColor Red

break

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_HIERARCHY\_MANAGER' and stmsg.MessageID = 4913 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "ERROR: Found SMS\_HIERARCHY\_MANAGER 4913 id's" -ForegroundColor Red

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_REPLICATION\_CONFIGURATION\_MONITOR' and stmsg.MessageID = 4629 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_REPLICATION\_CONFIGURATION\_MONITOR 4629 id's"

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_REPLICATION\_CONFIGURATION\_MONITOR' and stmsg.MessageID = 620 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Error: Found SMS\_REPLICATION\_CONFIGURATION\_MONITOR 620 id's" -ForegroundColor Red

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_DMP\_DOWNLOADER' and stmsg.MessageID = 4629 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_DMP\_DOWNLOADER 4629 id's"

}

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsg.RecordID from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_DMP\_DOWNLOADER' and stmsg.MessageID = 9700 and stmsg.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Error: Found SMS\_DMP\_DOWNLOADER 9700 id's" -ForegroundColor Red

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_001") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_WINNT\_SERVER\_DISCOVERY\_AGENT' and stmsg.MessageID = 4202 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '001'"

if ($component -ne $null) {

Write-Host "Found SMS\_WINNT\_SERVER\_DISCOVERY\_AGENT 4202 id's"

break

} else { Start-Sleep 10 }

}

if ($component -is [Array]) {

$Total = $component[0].InsStrValue

} else {

$Total = $component.InsStrValue

}

$roles = gwmi -Namespace ("root\sms\site\_001") -query "select \* from SMS\_SCI\_SysResUse where FileType=2 and RoleName != 'SMS Provider'"

if ($roles.count -ne $total) {

Write-Host "ERROR: Found $($roles.count). expected $Total" -ForegroundColor Red

} else {

Write-Host "All $Total roles have been created"

}

1. Upgrade to Current Branch 1910

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | What's new in version 1910 of Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/changes/whats-new-in-version-1910>  Install in-console updates for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/manage/install-in-console-updates>  Upgrade to Configuration Manager current branc  <https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/install/upgrade-to-configuration-manager>  Checklist for installing update 1910 for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/manage/checklist-for-installing-update-1910> |
| Description | In this chapter, we will be upgrading the existing installation (version 1902) to the latest (version 1910) and turning on some of the extra available features. |

* 1. Before you begin

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console |
| 02. if a new version is available, a message will appear with the text: A new update is available for Configuration Manager. You can view and enable available updates in the Administration workspace from the Cloud Services > Updates and Servicing node. |
| 03. Click Administration. |
| 04. Expand Cloud Services and then click Updates and Servicing  Note: All updates will be listed and in the State column, it will show as Available  Note: All downloaded files are going to be saved to C:\ConfigMgr\EasySetupPayload |
| 05. You can also review the following logs:   * ConfigMgr\Logs\Dmpdownloader.log: Records details on downloads from Microsoft and Intune. |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$MECMversion = '1910'

$Update = Get-CMSiteUpdate -Name "Configuration Manager $($MECMversion)" -Fast | where {$\_.UpdateType -eq 0}

if ($update -eq $null) {

Write-Host "No update found for MECM version $MECMversion, forcing check it now"

#$Component = gwmi -Namespace "root\SMS\site\_$($SiteCode)" -query "select \* from SMS\_SCI\_Component where FileType = 2 and ItemName = 'SMS\_DMP\_DOWNLOADER|SMS Dmp Connector' and ItemType='Component' and SiteCode='$($SiteCode)'"

#$props = $component.Props

#$prop = $props | where {$\_.PropertyName -eq 'EasySetupDownloadInterval'}

#$prop.Value = 1

#$component.Props = $props

#$component.Put() | Out-Null

Invoke-CMSiteUpdateCheck

start-sleep 120

}

$Update = Get-CMSiteUpdate -Name "Configuration Manager $($MECMversion)" -Fast | where {$\_.UpdateType -eq 0}

if ($update -eq $null) {

Write-Host "ERROR: No update found for MECM version $MECMversion" -ForegroundColor Red

} else {

if ($Update.State -eq 262146) {

Write-Host "Update found and ready to install"

} elseif ($Update.State -ne 262145) {

Write-Host "Update found, forcing download to start as soon as possbile"

Invoke-CMSiteUpdateDownload -Name "Configuration Manager $($MECMversion)" -Force

Write-host "Package in the queue to be downloaded"

} else {

Write-Host "Update being downloaded"

}

}

* 1. Run the prerequisite check

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Cloud Services and then click Updates and Servicing |
| 03. Select Configuration Manager 1910 and click Run prerequisite check |
| 04. Click Monitoring |
| 05. Click Site Servicing Status |
| 06. Click Show Status to monitor the prerequisite check |
| 07. You can also review the errors and warnings by examining the content of the C:\ConfigMgrPrereq.log file |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$MECMversion = '1910'

while ($true) {

$SiteUpdate = Get-CMSiteUpdate -Name "Configuration Manager $($MECMversion)" -Fast | where {$\_.UpdateType -eq 0}

if ($SiteUpdate -ne $null) {

if ($SiteUpdate.State -ne 262146) {

Write-Host "Update is in Downloading state..."

Start-Sleep 30

} else {

Write-Host "Update is ready, executing pre-req"

Invoke-CMSiteUpdatePrerequisiteCheck -Name "Configuration Manager $($MECMversion)"

break

}

}

}

* 1. Installing an Update

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Cloud Services and then click Updates and Servicing |
| 03. Select Configuration Manager 1910 and click Install Update Pack |
| 04. On Configuration Manager Updates Wizard, click Next |
| 05. On Features, unselect all and click next  Note: We will enable see how to enable new features later |
| 06. On Options for Client Update, click Next |
| 07. On Review and accept the terms for this update pack, select I accept the license terms and Privacy statement and click Next |
| 08. On Summary, click Next |
| 09. On Completion, click Close  Note: The Installation will start, and it will take some time to complete (60-90 minutes depending on your hardware). The State of the update is now show as Installed |
| 10. Once the installation is completed, restart the console and the message “A new version of the console is available (5.1910.1067.1400). Click ok to close the console and install the new version now. Click cancel to continue working with the old console (5.1902.1085.1700). Working in the old console might corrupt data.”. Click Ok |
| 11. Once completed, the console will restart |
| 12. Click Administration. |
| 13. Expand Cloud Services and then click Updates and Servicing  Note: The State of the update is now show as Installed |
| 14. You can also review the following logs:   * C:\ConfigMgr\Logs\CMUpdate.log: Records details of the upgrade process. * C:\ConfigMgr\Logs\hman.log: Records information about site configuration changes, and the publishing of site information in Active Directory Domain Services. * C:\ConfigMgrAdminUISetup.log: MECM console installation log |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$MECMversion = '1910'

while ($true) {

$SiteUpdate = Get-CMSiteUpdate -Name "Configuration Manager $($MECMversion)" -Fast | where {$\_.UpdateType -eq 0}

if ($SiteUpdate -ne $null) {

if ($SiteUpdate.State -ne 131074) {

Write-Host "Pre-Check is still happening..."

Start-Sleep 30

} else {

Write-Host "Pre-Req done, starting update"

Install-CMSiteUpdate -Name $SiteUpdate.Name

Get-Process -Name Microsoft.ConfigurationManagement | Stop-Process

break

}

}

}

while ($true) {

$SiteUpdate = Get-CMSiteUpdate -Name "Configuration Manager $($MECMversion)" -Fast | where {$\_.UpdateType -eq 0}

if ($SiteUpdate -ne $null) {

if ($SiteUpdate.State -ne 196612) {

Write-Host "Installation is still happening..."

Start-Sleep 30

} else {

Write-Host "Installation done, upgrading MECM Console"

$InstallationFolder = (Get-ItemProperty -Path "hklm:Software\Wow6432Node\Microsoft\ConfigMgr10\Setup" -ErrorAction SilentlyContinue)."UI Installation Directory"

$Connection = (Get-ItemProperty -Path "hklm:Software\Wow6432Node\Microsoft\ConfigMgr10\AdminUI\Connection" -ErrorAction SilentlyContinue)."Server"

if ($InstallationFolder -eq $null) {

$InstallationFolder = "C:\ConfigMgr\AdminConsole"

}

if ($InstallationFolder.Substring($InstallationFolder.Length-1) -eq '\') {

$InstallationFolder = $InstallationFolder.Substring(0, $InstallationFolder.Length-1)

}

if ($Connection -eq $null) {

$Connection = $servername

}

cd c:

Remove-Module ConfigurationManager -Force

Start-Process -Filepath ("C:\ConfigMgr\EasySetupPayload\$(($SiteUpdate | select PackageGuid).PackageGuid)\SMSSETUP\BIN\I386\consolesetup.exe") -ArgumentList ("/q TargetDir=`"$($InstallationFolder)`" DefaultSiteServerName=$($Connection)") -Wait -NoNewWindow

Start-Sleep 5

Start-Process -Filepath ("C:\ConfigMgr\AdminConsole\bin\Microsoft.ConfigurationManagement.exe")

$ModulePath = $env:SMS\_ADMIN\_UI\_PATH

if ($ModulePath -eq $null) {

$ModulePath = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment").SMS\_ADMIN\_UI\_PATH

}

$ModulePath = $ModulePath.Replace("bin\i386","bin\ConfigurationManager.psd1")

$Certificate = Get-AuthenticodeSignature -FilePath "$ModulePath" -ErrorAction SilentlyContinue

$CertStore = New-Object System.Security.Cryptography.X509Certificates.X509Store("TrustedPublisher")

$CertStore.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::MaxAllowed)

$Certexist = ($CertStore.Certificates | where {$\_.thumbprint -eq $Certificate.SignerCertificate.Thumbprint}) -ne $null

if ($Certexist -eq $false) {

$CertStore.Add($Certificate.SignerCertificate)

}

$CertStore.Close()

import-module $ModulePath -force

if ((get-psdrive $SiteCode -erroraction SilentlyContinue | measure).Count -ne 1) {

new-psdrive -Name $SiteCode -PSProvider "AdminUI.PS.Provider\CMSite" -Root $servername

}

cd "$($SiteCode):"

break

}

} else {

Write-Host "Installation is still happening..."

Start-Sleep 30

}

}

* 1. Turning on Features

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Cloud Services, Updates and Servicing and then click Features |
| 03. Select Approve application requests for users per device and click Turn On |
| 04. On the Confirmation Manager question, click Yes. |
| 05. Restart the Configuration Manager Console |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$FeatureName = "Approve application requests for users per device"

Get-CMSiteFeature -Name $FeatureName | Enable-CMSiteFeature -force

Get-Process -Name Microsoft.ConfigurationManagement | Stop-Process

start-sleep 10

Start-Process -Filepath ("C:\ConfigMgr\AdminConsole\bin\Microsoft.ConfigurationManagement.exe")

1. Basic Site Configuration

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | Prepare Windows Servers to support Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/network/prepare-windows-servers>  Deploy and manage content management infrastructure for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/configure/deploy-and-manage-content>  Define site boundaries and boundary groups for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/configure/define-site-boundaries-and-boundary-groups>  Run discovery for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/deploy/configure/run-discovery>  Manage accounts to access content in Configuration Manager  <https://docs.microsoft.com/en-gb/configmgr/core/plan-design/hierarchy/accounts> |
| Firewall Rules | Many firewall rules are being created in this lab for the same set of TCP Ports (80 and 443). They are only being created multiple times to show what ports are required per Site Role. In a production environment, when co-hosting site roles, there is no need to create multiple rules for the same set of TCP Ports |
| Description | In this chapter, we will be performing the basic configuration of the MECM. Installation and Configuration of the basic infrastructure for the MECM to work, including Distribution Point, Management Point, FallBack Status Point, SQL Reporting Services Point, Boundaries, Boundary Group, Distribution Point Group, Network Access Account and Discovery.  **Note:** Distribution Point and Management Point can be installed during the MECM Installation. It has not been done in this lab because we wanted to show you how to perform the installation at later stage, allowing you to be prepared to perform the same steps when installing in a remove server.  **Note:** from MECM 1806 and usage of Enhanced HTTP site system, Network Access Account for OS Deployment is not a requirement anymore. For more information refer to <https://docs.microsoft.com/en-us/configmgr/core/plan-design/hierarchy/enhanced-http> |

* 1. Installation of basic Roles (DP, MP, FSP, SRS)
     1. Distribution Point
        1. Creating Firewall Rules

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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 80 under specify local ports and click Next** |
| 05. **On Action, click Next** |
| 06. **On Profile, click Next** |
| 07. **On Name, type IIS Distribution Point (TCP 80) 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 443 under specify local ports and click Next** |
| 11. **On Action, click Next** |
| 12. **On Profile, click Next** |
| 13. **On Name, type IIS Distribution Point (TCP 443) Inbound and click Finish** |

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

New-NetFirewallRule -DisplayName "IIS Distribution Point (TCP 80) Inbound" -Action Allow -Direction Inbound -LocalPort 80 -Protocol TCP

New-NetFirewallRule -DisplayName "IIS Distribution Point (TCP 443) Inbound" -Action Allow -Direction Inbound -LocalPort 443 -Protocol TCP

* + - 1. Install Requirements

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Open Server Manager** |
| 02. **Click Manage and Add Roles and Features** |
| 03. **Before you begin, click Next** |
| 04. **Select Role-based or feature-based installation and click Next** |
| 05. **Select select a server from the server pool and select the server “SRV0002.classroom.intranet” and click Next** |
| 06. **Under select server roles, select Web Server (IIS)** |
| 07. **On Add Roles and Features Wizard, click Add Features and click Next** |
| 08. **Under Select Features, select Remote Differential Compression and click Next** |
| 09. **Under Web Server Role (IIS), click Next** |
| 10. **On Select role service, select:**   * **Security->Windows Authentication** * **Application Deployment->ISAP Extensions** * **Management Tools->IIS 6 Management Compatibility->IIS 6 Metabase Compatibility** * **Management Tools->IIS 6 Management Compatibility->IIS 6 WMI Compatibility** * **Management Tools->IIS Management Scripts and Tools**   **Click Next** |
| 11. **Under Confirm installation selections, click Install** |
| 12. **Once the installation is succeeded. Click Close** |

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

@("RDC", "Web-Server", "Web-ISAPI-Ext", "Web-Metabase", "Web-Windows-Auth") | Get-WindowsFeature | Install-WindowsFeature

* + - 1. Installing Site System Role

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Site Configuration and click Servers and Site System Roles** |
| 03. **Right click \\SRV0002.classroom.intranet and click Add Site System Roles** |
| 04. **On Add Site System Roles Wizard, General, click Next** |
| 05. **Under proxy, click Next** |
| 06. **Under Specify roles for this server, select Distribution Point and click Next** |
| 07. **Under Specify distribution point settings select “Install and configure IIS if required by Configuration Manager”. Leave the other default options and click Next** |
| 08. **Under Specify drive settings for this distribution point click Next** |
| 09. **Under Specify settings to install operating systems by using PXE boot click Next** |
| 10. **Under specify multicast settings for operating system deployment click Next** |
| 11. **Under specify the content validation settings, click Next** |
| 12. **Under specify the boundary groups associate with this site system click Next** |
| 13. **Under confirm the settings, click Next** |
| 14. **Under You have successfully completed the Add Site System Roles wizard with the following settings click close** |
| 15. **Click Monitoring** |
| 16. **Expand System Status and click Component Status** |
| 17. **Search for SMS\_DISTRIBUTION\_MANAGER** |
| 18. **Right Click SMS\_DISTRIBUTION\_MANAGER, Show Messages and click All** |
| 19. **Under Status Messages: Set Viewing Period, click OK** |
| 20. **Verify the existence of Message ID 2302**  Note: **When installing a new Distribution Point, it is normal see this message for the default MECM Client packages (<SITECODE>00002, <SITECODE>00003 and <SITECODE>00007) and MECM will retry the package again every 30 minutes. Once MECM successfully distribute the package, you will see the Message ID 2301** |
| 21. **Double click any 2302 messages to see its details. Once done, click Ok** |
| 22. **Verify the existence of Message ID 2399** |
| 23. **Double click this message to see its details. Once done, click Ok** |
| 24. **Verify the existence of Message ID 2362**  Note: **This message will appear if the Install and Configure IIS checkbox was selected** |
| 25. **Double click this message to see its details. Once done, click Ok** |
| 26. **Verify the existence of Message ID 9501**  Note: **This message will appear if the PXE options was not selected** |
| 27. **Double click this message to see its details. Once done, click Ok** |
| 28. **Verify the existence of Message ID 9503**  Note: **This message will appear if the Multicast option was not selected** |
| 29. **Double click this message to see its details. Once done, click Ok** |
| 30. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\DistMgr.log**   Note: **As the Install and Configure IIS checkbox was selected, the DISM command line will run** |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

if ((Get-CMSiteSystemServer -SiteSystemServerName "$servername") -eq $null) { New-CMSiteSystemServer -SiteCode $SiteCode -UseSiteServerAccount -ServerName $servername }

Add-CMDistributionPoint -CertificateExpirationTimeUtc "$((Get-Date).AddYears(20).ToString())" -SiteSystemServerName $servername -SiteCode $siteCode -ClientConnectionType Intranet -InstallInternetServer

start-sleep 90

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_DISTRIBUTION\_MANAGER' and stmsg.MessageID = 2302 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "ERROR: Found SMS\_DISTRIBUTION\_MANAGER 2302 id's" -ForegroundColor Red

}

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_DISTRIBUTION\_MANAGER' and stmsg.MessageID = 2391 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "ERROR: Found SMS\_DISTRIBUTION\_MANAGER 2391 id's" -ForegroundColor Red

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_DISTRIBUTION\_MANAGER' and stmsg.MessageID = 2362 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_DISTRIBUTION\_MANAGER 2362 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_DISTRIBUTION\_MANAGER' and stmsg.MessageID = 2399 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_DISTRIBUTION\_MANAGER 2399 id's"

break

} else { Start-Sleep 10 }

}

* + 1. Management Point
       1. Creating Firewall Rules

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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 80 under specify local ports and click Next** |
| 05. **On Action, click Next** |
| 06. **On Profile, click Next** |
| 07. **On Name, type IIS Management Point (TCP 80) 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 443 under specify local ports and click Next** |
| 11. **On Action, click Next** |
| 12. **On Profile, click Next** |
| 13. **On Name, type IIS Management Point (TCP 443) Inbound and click Finish** |
| 14. **Click New Rule** |
| 15. **On New Inbound Rule Wizard, select Port and click Next** |
| 16. **On Protocol and Ports select TCP and type 10123 under specify local ports and click Next** |
| 17. **On Action, click Next** |
| 18. **On Profile, click Next** |
| 19. **On Name, type IIS Client Notification (TCP 10123) Inbound and click Finish** |

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

New-NetFirewallRule -DisplayName "IIS Management Point (TCP 80) Inbound" -Action Allow -Direction Inbound -LocalPort 80 -Protocol TCP

New-NetFirewallRule -DisplayName "IIS Management Point (TCP 443) Inbound" -Action Allow -Direction Inbound -LocalPort 443 -Protocol TCP

New-NetFirewallRule -DisplayName "IIS Client Notification (TCP 10123) Inbound" -Action Allow -Direction Inbound -LocalPort 10123 -Protocol TCP

* + - 1. Install Requirements

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Open Server Manager** |
| 02. **Click Manage and Add Roles and Features** |
| 03. **Before you begin, click Next** |
| 04. **Select Role-based or feature-based installation and click Next** |
| 05. **Select select a server from the server pool and select the server “SRV0002.classroom.intranet” and click Next** |
| 06. **Under select server roles, select Web Server (IIS)** |
| 07. **On Add Roles and Features Wizard, click Add Features and click Next** |
| 08. **Under Select Features, select Background Intelligent Transfer Service (BITS)->IIS Server Extension** |
| 09. **On Add Roles and Features Wizard, click Add Features and click Next** |
| 10. **Under Web Server Role (IIS), click Next** |
| 11. **On Select role service, select:**   * **Security->Windows Authentication** * **Application Deployment->ISAP Extensions** * **Management Tools->IIS 6 Management Compatibility->IIS 6 Metabase Compatibility** * **Management Tools->IIS 6 Management Compatibility->IIS 6 WMI Compatibility** * **Management Tools->IIS Management Scripts and Tools.**   **Click Next** |
| 12. **Under Confirm installation selections, click Install** |
| 13. **Once the installation is succeeded. Click Close** |

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

@("Web-Server", "Web-ISAPI-Ext", "Web-Metabase", "Web-Windows-Auth", "BITS-IIS-Ext") | Get-WindowsFeature | Install-WindowsFeature

* + - 1. Installing Site System Role

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Site Configuration and click Servers and Site System Roles** |
| 03. **Right click \\SRV0002.classroom.intranet and click Add Site System Roles** |
| 04. **On Add Site System Roles Wizard, General, click Next** |
| 05. **Under proxy, click Next** |
| 06. **Under Specify roles for this server, select Management Point and click Next** |
| 07. **Under Specify management point settings, click Next** |
| 08. **Under specify management point database settings click Next** |
| 09. **Under confirm the settings, click Next** |
| 10. **Under You have successfully completed the Add Site System Roles Wizard with the following settings click Close** |
| 11. **Click Monitoring** |
| 12. **Expand System Status and click Component Status** |
| 13. **Search for SMS\_MP\_CONTROL\_MANAGER** |
| 14. **Right Click SMS\_MP\_CONTROL\_MANAGER, Show Messages and click All** |
| 15. **Under Status Messages: Set Viewing Period, click OK** |
| 16. **Verify the existence of Message ID 1013, 1014 and 1015** |
| 17. **Double click any of the 1013, 1014 and 1015 messages to see its details. Once done, click Ok** |
| 18. **Verify the existence of Message ID 500** |
| 19. **Double click on the messages to see its details. Once done, click Ok** |
| 20. **Verify the existence of Message ID 5460.** |
| 21. **Double click on the message to see its details. Once done, click Ok** |
| 22. **Open Internet Explorer and navigate to http://SRV0002.classroom.intranet/sms\_mp/.sms\_aut?mplist** |
| 23. **Navigate to http://SRV0002.classroom.intranet/sms\_mp/.sms\_aut?mpcert** |
| 24. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\MPSetup.log: Records the installation wrapper process.** * **C:\ConfigMgr\Logs\mpMSI.log: Records details of installation.** * **C:\ConfigMgr\Logs\mpcontrol.log: Records the registration of the management point with WINS. Records the availability of the management point every 10 minutes.** * **C:\ConfigMgr\Logs\mpfdm.log: Records the actions of the management point component that moves client files to the corresponding INBOXES folder on the site server.** |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

if ((Get-CMSiteSystemServer -SiteSystemServerName "$servername") -eq $null) { New-CMSiteSystemServer -SiteCode $SiteCode -UseSiteServerAccount -ServerName $servername }

Add-CMManagementPoint -SiteSystemServerName $servername -SiteCode $siteCode

start-sleep 90

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_MP\_CONTROL\_MANAGER' and stmsg.MessageID = 1013 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_MP\_CONTROL\_MANAGER 1013 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_MP\_CONTROL\_MANAGER' and stmsg.MessageID = 1014 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_MP\_CONTROL\_MANAGER 1014 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_MP\_CONTROL\_MANAGER' and stmsg.MessageID = 1015 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_MP\_CONTROL\_MANAGER 1015 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_MP\_CONTROL\_MANAGER' and stmsg.MessageID = 500 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_MP\_CONTROL\_MANAGER 500 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_MP\_CONTROL\_MANAGER' and stmsg.MessageID = 5460 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_MP\_CONTROL\_MANAGER 5460 id's"

break

} else { Start-Sleep 10 }

}

$web = New-Object -ComObject msxml2.xmlhttp

$url = "http://$($servername):80/sms\_mp/.sms\_aut?mplist"

try {

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

$web.send()

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

} catch {

Write-host "MPList ERROR: $($\_)" -ForegroundColor Red

}

$web = New-Object -ComObject msxml2.xmlhttp

$url = "http://$($servername):80/sms\_mp/.sms\_aut?mpcert"

try {

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

$web.send()

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

} catch {

Write-host "MPCert ERROR: $($\_)" -ForegroundColor Red

}

* + 1. Fallback Status Point
       1. Creating Firewall Rules

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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 80 under specify local ports and click Next** |
| 05. **On Action, click Next** |
| 06. **On Profile, click Next** |
| 07. **On Name, type IIS Fallback Status Point (TCP 80) Inbound and click Finish** |

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

New-NetFirewallRule -DisplayName "IIS Fallback Status Point (TCP 80) Inbound" -Action Allow -Direction Inbound -LocalPort 80 -Protocol TCP

* + - 1. Install Fallback Requirements

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Open Server Manager** |
| 02. **Click Manage and Add Roles and Features** |
| 03. **Before you begin, click Next** |
| 04. **Select Role-based or feature-based installation and click Next** |
| 05. **Select select a server from the server pool and select the server “SRV0002.classroom.intranet” and click Next** |
| 06. **Under select server roles, select Web Server (IIS)** |
| 07. **On Add Roles and Features Wizard, click Add Features** |
| 08. **Under Select Features, click Next** |
| 09. **On Add Roles and Features Wizard, click Add Features and click Next** |
| 10. **Under Web Server Role (IIS), click Next** |
| 11. **On Select role service, select:**   * **Management Tools->IIS 6 Management Compatibility->IIS 6 Metabase Compatibility** * **Management Tools->IIS 6 Management Compatibility->IIS 6 WMI Compatibility** * **Management Tools->IIS Management Scripts and Tools.**   **Click Next** |
| 12. **Under Confirm installation selections, click Install** |
| 13. **Once the installation is succeeded. Click Close** |

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

@("Web-Server", "Web-Metabase") | Get-WindowsFeature | Install-WindowsFeature

* + - 1. Installing Site System Role

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Site Configuration and click Servers and Site System Roles** |
| 03. **Right click \\SRV0002.classroom.intranet and click Add Site System Roles** |
| 04. **On Add Site System Roles Wizard, General, click Next** |
| 05. **Under proxy, click Next** |
| 06. **Under Specify roles for this server, select Fallback Status Point and click Next** |
| 07. **Under Specify Fallback Status Point settings leave the default settings and click Next** |
| 08. **Under confirm the settings, click Next** |
| 09. **Under You have successfully completed the Add Site System Roles wizard with the following settings click close** |
| 10. **Click Monitoring** |
| 11. **Expand System Status and click Component Status** |
| 12. **Search for SMS\_FALLBACK\_STATUS\_POINT** |
| 13. **Right Click SMS\_FALLBACK\_STATUS\_POINT, Show Messages and click All** |
| 14. **Under Status Messages: Set Viewing Period, click OK** |
| 15. **Verify the existence of Message ID 1013, 1014 and 1015** |
| 16. **Double click any of the 1013, 1014 and 1015 messages to see its details. Once done, click Ok** |
| 17. **Verify the existence of Message ID 500** |
| 18. **Double click on the messages to see its details. Once done, click Ok** |
| 19. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\fspMSI.log: Records messages generated by the installation.** * **C:\ConfigMgr\Logs\SMSFSPSetup.log: Records messages generated by the installation.** * **C:\ConfigMgr\Logs\FspIsapi.log: Records details about communications to the fallback status point from mobile device legacy clients and client computers.** |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

if ((Get-CMSiteSystemServer -SiteSystemServerName "$servername") -eq $null) { New-CMSiteSystemServer -SiteCode $SiteCode -UseSiteServerAccount -ServerName $servername }

Add-CMFallbackStatusPoint -SiteSystemServerName $servername -SiteCode $siteCode

start-sleep 90

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_FALLBACK\_STATUS\_POINT' and stmsg.MessageID = 1013 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_FALLBACK\_STATUS\_POINT 1013 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_FALLBACK\_STATUS\_POINT' and stmsg.MessageID = 1014 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_FALLBACK\_STATUS\_POINT 1014 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_FALLBACK\_STATUS\_POINT' and stmsg.MessageID = 1015 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_FALLBACK\_STATUS\_POINT 1015 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_FALLBACK\_STATUS\_POINT' and stmsg.MessageID = 500 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_FALLBACK\_STATUS\_POINT 500 id's"

break

} else { Start-Sleep 10 }

}

* + 1. Reporting Services Point
       1. Creating Firewall Rules

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 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 80 under specify local ports and click Next** |
| 05. **On Action, click Next** |
| 06. **On Profile, click Next** |
| 07. **On Name, type SQL Server Reporting Services (TCP 80) Inbound and click Finish** |

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

New-NetFirewallRule -DisplayName "SQL Server Reporting Services (TCP 80) Inbound" -Action Allow -Direction Inbound -LocalPort 80 -Protocol TCP

* + - 1. Installing Site System Role

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Site Configuration and click Servers and Site System Roles** |
| 03. **Right click \\SRV0002.classroom.intranet and click Add Site System Roles** |
| 04. **On Add Site System Roles Wizard, General, click Next** |
| 05. **Under proxy, click Next** |
| 06. **Under Specify roles for this server, select Reporting Services Point and click Next** |
| 07. **Under Specify Reporting Services Point settings, click Verify and under User name click Set New Account** |
| 08. **Under Windows User Account, type the following;**   * **User name: CLASSROOM\svc\_ssrsea** * **Password: Pa$$w0rd** * **Confirm password: Pa$$w0rd**   **Click Ok** |
| 09. **Once back to the Add Site System Roles Wizard, click Next** |
| 10. **Under confirm the settings, click Next** |
| 11. **Under You have successfully completed the Add Site System Roles wizard with the following settings click close** |
| 12. **Click Monitoring** |
| 13. **Expand System Status and click Component Status** |
| 14. **Search for SMS\_SRS\_REPORTING\_POINT** |
| 15. **Right Click SMS\_SRS\_REPORTING\_POINT, Show Messages and click All** |
| 16. **Under Status Messages: Set Viewing Period, click OK** |
| 17. **Verify the existence of Message ID 1013, 1014 and 1015** |
| 18. **Double click any of the 1013, 1014 and 1015 messages to see its details. Once done, click Ok** |
| 19. **Verify the existence of Message ID 500** |
| 20. **Double click on the messages to see its details. Once done, click Ok** |
| 21. **Monitor C:\ConfigMgr\Logs\srsrp.log for the status of the reports importation.**  Note: **Once the Reporting Services Point is installed, MECM will import reports into the SQL Server Reporting Service. As there are few hundred reports, this process can take some time.** |
| 22. **Return to Configuration Manager console, monitoring workspace, expand Reporting and click Reports**  Note: **You should see all reports being populated. Once the process of importing the reports are done, there should be over 450 reports.** |
| 23. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\srsrpsetup.log: Records messages generated by the installation.** * **C:\ConfigMgr\Logs\srsrpMSI.log: Records messages generated by the installation.** * **C:\ConfigMgr\Logs\srsrp.log: Records information about the activity and status of the reporting services point.** |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$sqlServerInstance = 'SSRS'

$Secure = 'Pa$$w0rd'| ConvertTo-SecureString -AsPlainText -Force

$account = "CLASSROOM\svc\_ssrsea"

New-CMAccount -Name "$account" -Password $Secure -SiteCode "$SiteCode"

if ((Get-CMSiteSystemServer -SiteSystemServerName "$servername") -eq $null) { New-CMSiteSystemServer -SiteCode $SiteCode -UseSiteServerAccount -ServerName $servername }

Add-CMReportingServicePoint -ReportServerInstance $sqlServerInstance -SiteSystemServerName "$servername" -UserName "$account" -DatabaseName "CM\_$SiteCode" -DatabaseServerName "$servername" -FolderName "ConfigMgr\_$SiteCode" -SiteCode "$SiteCode"

start-sleep 90

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_SRS\_REPORTING\_POINT' and stmsg.MessageID = 1013 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_SRS\_REPORTING\_POINT 1013 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_SRS\_REPORTING\_POINT' and stmsg.MessageID = 1014 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_SRS\_REPORTING\_POINT 1014 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_SRS\_REPORTING\_POINT' and stmsg.MessageID = 1015 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_SRS\_REPORTING\_POINT 1015 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_SRS\_REPORTING\_POINT' and stmsg.MessageID = 500 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_SRS\_REPORTING\_POINT 500 id's"

break

} else { Start-Sleep 10 }

}

$web = New-Object -ComObject msxml2.xmlhttp

$url = "http://localhost:80/reportserver/ConfigMgr\_$SiteCode"

while ($true) {

try {

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

$web.send()

if ($web.status -eq "404") { start-sleep 10 }

if ($web.status -eq "200") {

Write-Host "Found ConfigMgr\_$SiteCode reporting site"

break

}

} catch {

#

}

}

* 1. Boundaries

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Hierarchy Configuration and click Boundaries** |
| 03. **Click Create Boundary** |
| 04. **On General select:**   * **Type: IP Address Range** * **Starting IP Address: 192.168.3.1** * **Ending IP Address: 192.168.3.254**   **Click Ok** |

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

New-CMBoundary -DisplayName "Training Lab Boundary" -BoundaryType IPRange -Value "192.168.3.1-192.168.3.254"

* 1. Boundary Group

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Hierarchy Configuration and click Boundary Group** |
| 03. **Click Create Boundary Group** |
| 04. **On General:**   * **Name: Training Lab** * **Boundaries: 192.168.3.1-192.168.3.254**   **Change to the References Tab** |
| 05. **On References:**   * **Select Use this boundary group for site assignment** * **Site Systems Servers: \\SRV0002.rflsytems.intranet**   **Click OK** |
| 06. **Right Click Training Lab and click Show Members** |
| 07. **The 192.168.3.1-192.168.3.254 boundary was added to the boundary group Training Lab** |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

New-CMBoundaryGroup -Name "Training Lab" -AddSiteSystemServerName @($servername) -DefaultSiteCode $SiteCode

Add-CMBoundaryToGroup -BoundaryGroupName "Training Lab" -BoundaryName "Training Lab Boundary"

* 1. Distribution Point Group

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration** |
| 02. **Click Distribution Point Group** |
| 03. **Click Create Group** |
| 04. **On General, type Training Lab under Name and then Click Add** |
| 05. **On Add Distribution Points, select SRV0002.CLASSROOM.INTRANET and click OK twice** |
| 06. **Right Click Training Lab and click Show Members** |
| 07. **The SRV0002.CLASSROOM.INTRANET boundary was added to the distribution point group Training Lab** |

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

$servername = "SRV0002.classroom.intranet"

New-CMDistributionPointGroup -Name "Training Lab"

Add-CMDistributionPointToGroup -DistributionPointGroupName "Training Lab" -DistributionPointName "$servername"

* 1. Network Access Account

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Site Configuration and click Sites** |
| 03. **Select 001 – Training Lab site and click Configuration Site Components -> Software Distribution** |
| 04. **On Software Distribution Component Properties, change to the Network Access Account tab** |
| 05. **Under Network Access Account tab select Specify the account that accesses network locations and click Set -> New Account** |
| 06. **Under Windows User account type:**   * **User Name: CLASSROOM\svc\_mecmna** * **Password: Pa$$w0rd** * **Confirm Password: Pa$$w0rd**   **Click Verify** |
| 07. **Under verify type \\SRV0002\sms\_site for Network Share and click Test Connection** |
| 08. **Once the connection was successfully verified, click Ok three times** |

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

$SiteCode = "001"

$Secure = 'Pa$$w0rd'| ConvertTo-SecureString -AsPlainText -Force

$account = "CLASSROOM\svc\_mecmna"

New-CMAccount -Name "$account" -Password $Secure -SiteCode $SiteCode

Set-CMSoftwareDistributionComponent -SiteCode $SiteCode -NetworkAccessAccountNames $account

* 1. Discovery
     1. Active Directory Forest Discovery

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Hierarchy Configuration and click Discovery Methods** |
| 03. **Select Active Directory Forest Discovery and click Properties** |
| 04. **Under General, select Enabled Active Directory Forest Discovery and select Automatically create IP address range boundaries for IP subnets when they are discovered and Automatically create Active Directory site boundaries when they are discovered. Leave the schedule to run every 1 weeks and click OK** |
| 05. **When asked if you want to run full discovery as soon as possible, click Yes** |
| 06. **Select Boundaries and confirm that 3 new boundaries have been populated there** |
| 07. **Select Active Directory Forests and confirm that the Classroom.intranet is populated there.** |
| 08. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\ADForestDisc.Log: Records Active Directory Forest Discovery actions.** |

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

$SiteCode = "001"

Set-CMDiscoveryMethod -ActiveDirectoryForestDiscovery -EnableActiveDirectorySiteBoundaryCreation $True -Enabled $True -EnableSubnetBoundaryCreation $True -SiteCode $SiteCode

Invoke-CMForestDiscovery -SiteCode $SiteCode

* + 1. Active Directory System Discovery

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Hierarchy Configuration and click Discovery Methods** |
| 03. **Select Active Directory System Discovery and click Properties** |
| 04. **Under General, select Enabled Active Directory System Discovery and add the active directory containers you want to search (using the yellow button)**  Note: **In this exercise we will use the classroom.intranet domain** |
| 05. **Click Polling Schedule to change the default schedule** |
| 06. **Click Active Directory Attributes to add more attributes to be discovered**  Note: **In this exercise, we will add pwdLastSet attribute** |
| 07. **Click options to add a filter to the discovered machines**  Note: **This option requires an active directory functional level of Windows Server 2003 or later**  Note: **In this exercise, we will select both options with default value of 90 days** |
| 08. **Click Ok and When asked if you want to run full discovery as soon as possible, click Yes** |
| 09. **Click Assets and Compliance and select Devices.**  Note: **There are a few more machines discovered now.**  Note: **Disabled Computers Accounts in Active Directory and Computers that do not have an IP Address are not going to be discovered** |
| 10. **Select WKS0001 and click Properties** |
| 11. **Once the WKS0001 Properties open, note the discovery method used under Agent Name.** |
| 12. **note the pwdLastSet property was also added to the list. Click Ok** |
| 13. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\adsysdis.log: Records Active Directory System Discovery actions.**   Note: **Look for a line that starts: INFO: Search filter. This line shows the actual search query that was send to AD. There are two types of queries: Full synchronization and Incremental synchronization.** |

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

$SiteCode = "001"

$domainName = "LDAP://DC=$($env:USERDNSDOMAIN.Split(".") -join ",DC=")"

Set-CMDiscoveryMethod -ActiveDirectorySystemDiscovery -AddActiveDirectoryContainer "$($domainName)" -Enabled $True -EnableDeltaDiscovery $True -EnableFilteringExpiredLogon $True -EnableFilteringExpiredPassword $True -SiteCode $SiteCode -TimeSinceLastLogonDays 90 -TimeSinceLastPasswordUpdateDays 90 -AddAdditionalAttribute @("pwdLastSet") -Recursive

Invoke-CMSystemDiscovery -SiteCode $SiteCode

* + 1. Active Directory User Discovery

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Hierarchy Configuration and click Discovery Methods** |
| 03. **Select Active Directory User Discovery and click Properties** |
| 04. **Under General, select Enabled Active Directory User Discovery and add the active directory containers you want to search (using the yellow button)**  Note: **In this exercise we will use the classroom.intranet domain** |
| 05. **Click Polling Schedule to change the default schedule** |
| 06. **Click Active Directory Attributes to add more attributes to be discovered**  Note: **In this exercise, we will add physicalDeliveryOfficeName and department attribute** |
| 07. **Click Ok and When asked if you want to run full discovery as soon as possible, click Yes** |
| 08. **Click Assets and Compliance and select Users.** |
| 09. **Select CLASSROOM\User01 (User01) and click Properties** |
| 10. **Once the CLASSROOM\User01 (User01) Properties open, note the discovery method used under Agent Name.** |
| 11. **Note the physicalDeliveryOfficeName and department property was also added to the list. Click Ok** |
| 09. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\adusrdis.log: Records Active Directory User Discovery actions.**   Note: **Look for a line that starts: INFO: Search filter. This line shows the actual search query that was send to AD. There are two types of queries: Full synchronization and Incremental synchronization.** |

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

$SiteCode = "001"

$domainName = "LDAP://DC=$($env:USERDNSDOMAIN.Split(".") -join ",DC=")"

Set-CMDiscoveryMethod -ActiveDirectoryUserDiscovery -AddActiveDirectoryContainer "$($domainname)" -DeltaDiscoveryIntervalMinutes 30 -Enabled $True -EnableDeltaDiscovery $True -SiteCode $SiteCode -AddAdditionalAttribute @("physicalDeliveryOfficeName") -recursive

Invoke-CMUserDiscovery -SiteCode $SiteCode

* + 1. Active Directory Group Discovery

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. **Start Configuration Manager Console and Click Administration.** |
| 02. **Expand Hierarchy Configuration and click Discovery Methods** |
| 03. **Select Active Directory Group Discovery and click Properties** |
| 04. **Under General, select Enabled Active Directory Group Discovery and click Add -> Location** |
| 05. **Under Add Active Directory Location add active directory containers you want to search and click Ok**  Note: **In this exercise, we will use the classroom.intranet domain** |
| 06. **One Back to the Active Directory Group Discovery Properties, click Next** |
| 07. **Click Polling Schedule to change the default schedule** |
| 08. **Click options to add a filter to the discovered machines**  Note: **This option required active directory functional level of Windows Server 2003 or later**  Note: **In this exercise, we will leave the default unchecked option** |
| 09. **Click Ok and When asked if you want to run full discovery as soon as possible, click Yes** |
| 10. **Click Assets and Compliance and select Users.** |
| 11. **Select CLASSROOM\Allowed RODC Password Replication Group and click Properties** |
| 12. **Once the CLASSROOM\Allowed RODC Password Replication Group Properties open, note the discovery method used under Agent Name.** |
| 13. **You can also review the following logs:**   * **C:\ConfigMgr\Logs\adsgdis.log: Records Active Directory Group Discovery actions.**   Note: **Look for a line that starts: INFO: Search filter. This line shows the actual search query that was send to AD. There are two types of queries: Full synchronization and Incremental synchronization.** |

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

$SiteCode = "001"

$domainName = "LDAP://DC=$($env:USERDNSDOMAIN.Split(".") -join ",DC=")"

$discovery = New-CMADGroupDiscoveryScope -LdapLocation "$($domainName)" -Name "$($domainName)" -SiteCode $SiteCode -RecursiveSearch $true

Set-CMDiscoveryMethod -ActiveDirectoryGroupDiscovery -AddGroupDiscoveryScope ($discovery) -Enabled $True -SiteCode $SiteCode -EnableDeltaDiscovery $true

Invoke-CMGroupDiscovery -SiteCode $SiteCode

1. Basic Client Settings

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | About client settings in Configuration Manager  <https://docs.microsoft.com/en-gb/configmgr/core/clients/deploy/about-client-settings> |
| Comments | Configuring default client settings apply to every single MECM client. Depending on the settings it is recommended to create a new client setting and apply only to computers that need the new settings |
| Description | In this chapter, we will be changing the Default Client Settings with the basic configuration that will be required during this lab.  **Note:** Changing the default client settings apply to every single MECM client. Depending on the settings it is recommended to create a new client setting and apply only to computers that need the new settings |

* 1. Changing Default Client Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select the default client settings and click Properties  Note: The Changing the Default Client Settings should be avoided as it applies to all clients and there is no way to revert it back. My recommendation is to create a new client setting with the changes and deploying to a collection with all clients if needed |
| 04. Under Default Settings, click Computer Agent and change Organization name displayed in Software Center to Training Lab |
| 05. Click Hardware Inventory and change the schedule from 7 days to 1 day |
| 06. Click Software Center and change:   * Select these new settings to specify company information: Yes * Company Name: Training Lab * Select a logo for Software Center, click browse and select \\srv0001\TrainingFiles\Scripts\traininglab.jpg * Unselect Hide installed applications in Software Center |
| 07. Click Software Inventory and change the Schedule from 7 days to 1 day |
| 08. Under software Inventory click Set Types |
| 09. Under Configure Client Settings, add file type \*.exe and click Ok  Note: Adding \*.exe for All client hard disk will take long and will fills the database with probably useless information. Validate your environment and use a better path location such as \*.exe from %programfiles% or \*.sys from %systemroot%\system32\drivers. |
| 10. Back to Default settings  Note: The authors’ personal recommendation is disabling software inventory, as it does not bring any real benefit. As it relies on WMI to search for the files, and WMI is not indexed, it may take longer than expected and thus may affect the client performance |
| 11. Click State Messaging and set the reporting cycle to 2 minutes. Click Ok.  Note: Setting a low value here will allow workstations to send information quicker to the server, however, setting it to a low value is not recommended in a production environment as it will affect Server Performance as well as may increasing network traffic. |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ClientSettingsName = "Default Client Agent Settings"

Set-CMClientSetting -ComputerAgentSettings -Name "$ClientSettingsName" -BrandingTitle "Training Lab"

$schedule = New-CMSchedule -RecurCount 1 -RecurInterval Days

Set-CMClientSetting -HardwareInventorySettings -Name "$ClientSettingsName" -EnableHardwareInventory $True -InventorySchedule $Schedule

Set-CMClientSettingSoftwareCenter -DefaultSetting -EnableCustomize $true -CompanyName "Training Lab" -LogoFilePath "\\srv0001\TrainingFiles\Scripts\traininglab.jpg" -HideInstalledApplication $false

$dict = @{"FileName"="\*.exe"; Exclude=$true; ExcludeWindirAndSubfolders=$true; Subdirectories=$true; Path="\*"}

Set-CMClientSettingSoftwareInventory -DefaultSetting -Enable $True -Schedule $schedule -AddInventoryFileType $dict

Set-CMClientSetting -Name "$ClientSettingsName" -StateMessageSettings -StateMessagingReportingCycleMinutes 2

1. Client Installation

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  SRV0003  WKS0001  WKS0002  WKS0004 |
| More information | About client installation properties in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/deploy/about-client-installation-properties>  How to deploy clients to Windows computers in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/deploy/deploy-clients-to-windows-computers> |
| Description | In this chapter, we will be configuring SCCM for the client installation (Windows only), will be installing Windows client as well as looking at the initial information that is returned from the client once it has been installed |

* 1. Windows Client Installation
     1. Push Configuration

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Sites |
| 03. Select 001 – Training Lab, Client Installation Settings and click Client Push Installation |
| 04. On Client Push Installation Properties, general tab, unselect “Allow connection fallback to NTLM” and then change to the Accounts Tab |
| 05. On the Accounts tab, click New (yellow button) |
| 06. Under Windows User Account, type CLASSROOM\svc\_mecmpush as UserName and Pa$$word for password and confirm password. Click Verify >> |
| 07. Under verify, select data source network share and network share type \\WKS0001\c$ and click Test Connection |
| 08. Click Ok twice when the connection is successfully verified |
| 09. Once back to the Client Push Installation Properties, change to the Installation Properties tab |
| 10. On the Installation Properties tab, type FSP=SRV0002 at the end of the Installation Properties and click Ok |

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

$SiteCode = "001"

$Secure = 'Pa$$w0rd'| ConvertTo-SecureString -AsPlainText -Force

$account = "CLASSROOM\svc\_mecmpush"

New-CMAccount -Name "$account" -Password $Secure -SiteCode "$SiteCode"

Set-CMClientPushInstallation -AddAccount "$account" -EnableAutomaticClientPushInstallation $False -EnableSystemTypeConfigurationManager $False -EnableSystemTypeServer $False -EnableSystemTypeWorkstation $False -InstallationProperty "SMSSITECODE=$($SiteCode) FSP=$($servername)" -InstallClientToDomainController $False -SiteCode "$($SiteCode)"

#Enable Kerberos only

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select \* From SMS\_SCI\_Component where FileType=2 and ItemName = 'SMS\_DISCOVERY\_DATA\_MANAGER|SMS Site Server' and SiteCode='$SiteCode'"

$component.get()

$props = $component.Props

$prop = $props | where {$\_.PropertyName -eq 'ENABLEKERBEROSCHECK'}

$prop.Value = 2 #change to 3 if NTLM needs to be enabled

$component.Props = $props

$component.Put() | Out-Null

* + 1. Manual Push

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Device Collections |
| 03. Right click All Systems and click Update Membership |
| 04. On “This action will re-evaluate the membership rules for the selected collection and might take some time to finish”, click Yes  Note: It will take about 10 seconds for the process to complete  Note: Updating All Systems collection is only required if the device has been discovered and not yet added to the All Systems collection. In a default installation, it will take about 5 minutes after the discovery |
| 05. Right Click All Systems and click Show Members |
| 04. Select WKS0001 and click Install Client |
| 05. On Before You Begin, click Next |
| 06. Under Installation Options, select Install the client software from a specific site, confirm the site 001 is selected and click Next |
| 07. Under Summary, click Next |
| 08. Under Completion, click Close |
| 09. You can also review the following logs:   * C:\ConfigMgr\Logs\ccm.log: Records client push installation activities.   Note: Repeat the process for the WKS0002 and WKS0004 machines |

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

$SiteCode = "001"

Invoke-CMCollectionUpdate -Name "All Systems"

Start-sleep 10

@("WKS0001", "WKS0002", "WKS0004") | foreach { Get-CMDevice -Name $\_ | Install-CMClient -SiteCode "$SiteCode" }

* + 1. Validating the Installation and Installation Process on the Client

|  |
| --- |
| Perform this task on the WKS0001 virtual machine logged on as user01 |
| 01. On the client WKS0001, open Task manager and confirm that ccmsetup.exe is running |
| 02. Examine the content of C:\windows\ccmsetup folder |
| 03. Once the Installation is completed, the ccmexec.exe process will appear instead of ccmsetup.exe |
| 04. Open Control Panel and confirm that Configuration Manager exist. Open it. |
| 05. On the Configuration Manager Properties, General Tab, confirm the Site Code is SMS:001 and Assigned management point is SRV0002.CLASSROOM.INTRANET. Change to the Components tab |
| 06. Under Components tab, confirm the Components that are installed and enabled. Change to the Actions tab |
| 07. Under Actions tab, confirm that there are 2 actions. Change to the Site tab  Note: After the installation, it is normal to have only 2 actions until the post-installation process finishes. After this, there will be 10 actions to be used, depending on the options enabled in the client settings |
| 08. Under site, confirm the Configuration manager service location is set to 001. Click Ok |
| 09. Open Software Center under Start -> Microsoft System Center -> Configuration Manager |
| 10. Confirm that the Software Center opens without any problem. |
| 11. You can also review the following client logs:   * C:\Windows\CCMSetup\Logs\ccmsetup.log: Records ccmsetup tasks for client setup, client upgrade, and client removal. Can be used to troubleshoot client installation problems. * C:\Windows\CCMSetup\Logs\client.msi.log: Records setup tasks performed by client.msi. Can be used to troubleshoot client installation or removal problems. |

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

while ($true) {

$Process = Get-Process -Name ccmsetup -ErrorAction SilentlyContinue

if ($Process -ne $null) { Start-Sleep 10 }

else { Write-host "Process ccmsetup.exe does not exist or already finished"; break }

}

while ($true) {

$Process = Get-Process -Name ccmexec -ErrorAction SilentlyContinue

if ($Process -eq $null) { Start-Sleep 10 }

else { Write-host "Process ccmexec exist"; break }

}

start-sleep 60

"Client is assigned to $((Invoke-WMIMethod -Namespace root\ccm -Class SMS\_Client -Name GetAssignedSite).sSiteCode)"

* + 1. Validating the Installation and Installation Process on the MECM Console

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Devices and confirm that WKS0001 says:   * Installed: Yes * Site Code: 001 * Client Activity: Active   Note: The Client Activity will only show Active after the client post-installations have been completed and the device has been online for at least 15 minutes |

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

$SiteCode = "001"

Invoke-CMCollectionUpdate -Name "All Systems"

Start-sleep 10

Get-CMDevice -Name "WKS000?" | select Name, IsClient, SiteCode, ClientActiveStatus

* 1. Client Properties

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Devices |
| 03. Select WKS0001 and click Properties |
| 04. Once the Properties Open note the Agent Edition: Windows desktop or server |
| 05. Operating System Name and version shows Microsoft Windows NT Workstation 10.0. Click Ok  Note: Repeat the process for the WKS0002 and WKS0004 machines  Note: The Operating System may be shown as Microsoft Windows NT Workstation 10.0 (Tablet Edition) |

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

Get-CMDevice -Name "WKS000?" | select Name, ClientEdition, DeviceOS

* 1. Resource Explorer

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Devices |
| 03. Right click a Windows computer and click Start -> Resource Explorer |
| 04. Once the Resource Explorer open, expand hardware  Note: If the inventory information is empty, you will need to wait a bit longer, so the device has time to collect and send the information to the server and the server has time to process. The wait time may be close to 30 minutes |
| 05. Select Disk Drives |
| 06. Select Logical Disk |
| 07. Select Memory |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ModulePath = $env:SMS\_ADMIN\_UI\_PATH

if ($ModulePath -eq $null) {

$ModulePath = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment").SMS\_ADMIN\_UI\_PATH

}

$ModulePath = $ModulePath.Replace("bin\i386","bin\resourceexplorer.exe")

#windows machine

$Device = get-CMDevice -name "WKS0001"

Start-Process -Filepath ("$ModulePath") -ArgumentList ("-s -sms:ResourceID=$($Device.ResourceID) -sms:connection=\\$($servername)\root\sms\site\_$($siteCode)")

1. Collections

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | Introduction to collections in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/collections/introduction-to-collections>  How to create collections in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/collections/create-collections>  Best practices for collections in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/collections/best-practices-for-collections> |
| Collection Rules | Creating collection rules with like are not recommended as this could cause a performance issue on the MECM and will slow down the collection membership update. In this lab, we are using like only for demonstration purposes. More information about this, refer to <http://www.enhansoft.com/blog/configuration-manager-collections-and-collection-evaluation-viewer> and <http://www.enhansoft.com/blog/how-to-fix-a-poorly-written-wql-query> |
| Description | In this chapter, we will be Creating several collections as well as chaning the default interval of collection membership incremental evaluation.  **Note:** In this lab we are changing the collection membership incremental evaluation to speed up the process of updating collections and decreasing the default value is not recommended in a production environment |

* 1. Windows 10 Workstation Collection

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Device Collections |
| 03. Select Device Collections and click Create Device Collection |
| 04. Under Specify details for this collection type Windows 10 Workstations under name and select All Systems under Limiting Collection. Click Next |
| 05. Under Define membership rules for this collection, select Add query and click Query Rule |
| 06. Under Query Rule Properties, type Windows 10 for Name and click Edit Query Statement |
| 07. Under Query Statement Properties, change to criteria tab |
| 08. Under criteria tab click New button |
| 09. Under Criterion Properties, click Select |
| 10. Under Select Attribute select:   * Attribute Class: System Resource * Attribute: Operating System Name and Version   Click Ok |
| 11. Under Criterion Properties, select Operator is Like and type %Windows NT Workstation 10% under value. Click Ok 3 times |
| 12. Under Create Device Collection Wizard, select Use incremental updates for this collection and click Next  Note: It is not recommended to have over 250 collections with the Incremental updates enabled |
| 13. Under confirm the settings, click Next |
| 14. Under The Create Device Collection Wizard completed successfully, click Close |
| 15. Under device collections, the new Collection is still under update status.  Note: Once the collection is created, there is a process to populate it and it may take a while. In this lab, wait 30 seconds or refresh it couple of times until you see the Member Count increment to 2 |
| 16. Select the collection and click Show Members |
| 17. The collection will be expanded under Devices and all devices that match the query filter will be displayed. |

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

$CollUpdate = New-CMSchedule -Start "01/01/2015 9:00 PM" -DayOfWeek Saturday -RecurCount 1

$Collection = New-CMDeviceCollection -Name "Windows 10 Workstations" -LimitingCollectionName "All Systems" -RefreshSchedule $CollUpdate -RefreshType Both

Add-CMDeviceCollectionQueryMembershipRule -CollectionId $Collection.CollectionID -RuleName "Windows 10" -QueryExpression "select \* from SMS\_R\_System where OperatingSystemNameandVersion like 'Microsoft Windows NT Workstation 10.%'"

start-sleep 20

Get-CMCollectionMember -CollectionName "Windows 10 Workstations" | select Name

* 1. Windows 8 Workstation Collection

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Device Collections |
| 03. Select Device Collections and click Create Device Collection |
| 04. Under Specify details for this collection type Windows 8 Workstations under name and select All Systems under Limiting Collection. Click Next |
| 05. Under Define membership rules for this collection, select Add query and click Query Rule |
| 06. Under Query Rule Properties, type Windows 8 for Name and click Edit Query Statement |
| 07. Under Query Statement Properties, change to criteria tab |
| 08. Under criteria tab click New button |
| 09. Under Criterion Properties, click Select |
| 10. Under Select Attribute select:   * Attribute Class: System Resource * Attribute: Operating System Name and Version   Click Ok |
| 11. Under Criterion Properties, select Operator is Like and type %Windows NT Workstation 6.3% under value. Click Ok 3 times |
| 12. Under Create Device Collection Wizard, select Use incremental updates for this collection and click Next  Note: It is not recommended to have over 250 collections with the Incremental updates enabled |
| 13. Under confirm the settings, click Next |
| 14. Under The Create Device Collection Wizard completed successfully, click Close |
| 15. Under device collections, the new Collection is still under update status.  Note: Once the collection is created, there is a process to populate it and it may take a while. In this lab, wait 30 seconds or refresh it couple of times until you see Member Count change to 1 |
| 16. Select the collection and click Show Members |
| 17. The collection will be expanded under Devices and all devices that match the query filter will be displayed. |

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

$CollUpdate = New-CMSchedule -Start "01/01/2015 9:00 PM" -DayOfWeek Saturday -RecurCount 1

$Collection = New-CMDeviceCollection -Name "Windows 8 Workstations" -LimitingCollectionName "All Systems" -RefreshSchedule $CollUpdate -RefreshType Both

Add-CMDeviceCollectionQueryMembershipRule -CollectionId $Collection.CollectionID -RuleName "Windows 8" -QueryExpression "select \* from SMS\_R\_System where OperatingSystemNameandVersion like 'Microsoft Windows NT Workstation 6.3%'"

start-sleep 20

Get-CMCollectionMember -CollectionName "Windows 8 Workstations" | select Name

* 1. Active Directory OU Workstations Collection

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Device Collections |
| 03. Select Device Collections and click Create Device Collection |
| 04. Under Specify details for this collection type Workstation OU under name and select All Systems under Limiting Collection. Click Next |
| 05. Under Define membership rules for this collection, select Add query and click Query Rule |
| 06. Under Query Rule Properties, type Workstation OU for Name and click Edit Query Statement |
| 07. Under Query Statement Properties, change to criteria tab |
| 08. Under criteria tab click New button |
| 09. Under Criterion Properties, click Select |
| 10. Under Select Attribute select:   * Attribute Class: System Resource * Attribute: System OU Name   Click Ok |
| 11. Under Criterion Properties, select Operator is Like and type CLASSROOM.INTRANET/CLASSROOM/WORKSTATIONS% under value. Click Ok 3 times |
| 12. Under Create Device Collection Wizard, select Use incremental updates for this collection and click Next  Note: It is not recommended to have over 250 collections with the Incremental updates enabled |
| 13. Under confirm the settings, click Next |
| 14. Under The Create Device Collection Wizard completed successfully, click Close |
| 15. Under device collections, the new Collection is still under update status.  Note: Once the collection is created, there is a process to populate it and it may take a while. In this lab, wait 30 seconds or refresh it couple of times until you see Member Count change to 3 |
| 16. Select the collection and click Show Members |
| 17. The collection will be expanded under Devices and all devices that match the query filter will be displayed. |

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

$CollUpdate = New-CMSchedule -Start "01/01/2015 9:00 PM" -DayOfWeek Saturday -RecurCount 1

$Collection = New-CMDeviceCollection -Name "Workstation OU" -LimitingCollectionName "All Systems" -RefreshSchedule $CollUpdate -RefreshType Both

Add-CMDeviceCollectionQueryMembershipRule -CollectionId $Collection.CollectionID -RuleName "Workstation OU" -QueryExpression "select SMS\_R\_SYSTEM.ResourceID,SMS\_R\_SYSTEM.ResourceType,SMS\_R\_SYSTEM.Name,SMS\_R\_SYSTEM.SMSUniqueIdentifier,SMS\_R\_SYSTEM.ResourceDomainORWorkgroup,SMS\_R\_SYSTEM.Client from SMS\_R\_System where SMS\_R\_System.SystemOUName like 'CLASSROOM.INTRANET/CLASSROOM/WORKSTATIONS/%'"

start-sleep 20

Get-CMCollectionMember -CollectionName "Workstation OU" | select Name

* 1. Active Directory OU Users Collection

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Users Collections |
| 03. Select User Collections and click Create User Collection |
| 04. Under Specify details for this collection type Users OU under name and select All Users and User Groups under Limiting Collection. Click Next |
| 05. Under Define membership rules for this collection, select Add query and click Query Rule |
| 06. Under Query Rule Properties, type Users OU for Name and click Edit Query Statement |
| 07. Under Query Statement Properties, change to criteria tab |
| 08. Under criteria tab click New button |
| 09. Under Criterion Properties, click Select |
| 10. Under Select Attribute select:   * Attribute Class: UserResource * Attribute: User OU Name   Click Ok |
| 11. Under Criterion Properties, select Operator is equal to and type CLASSROOM.INTRANET/CLASSROOM/USERS under value. Click Ok 3 times |
| 12. Under Create User Collection Wizard, select Use incremental updates for this collection and click Next  Note: It is not recommended to have over 250 collections with the Incremental updates enabled |
| 13. Under confirm the settings, click Next |
| 14. Under The Create User Collection Wizard completed successfully, click Close |
| 15. Under User collections, the new Collection is still under update status.  Note: Once the collection is created, there is a process to populate it and it may take a while. In this lab, wait 30 seconds or refresh it couple of times until you see Member Count change to 2 |
| 16. Select the collection and click Show Members |
| 17. The collection will be expanded under Users and all users that match the query filter will be displayed. |

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

$CollUpdate = New-CMSchedule -Start "01/01/2015 9:00 PM" -DayOfWeek Saturday -RecurCount 1

$Collection = New-CMUserCollection -Name "Users OU" -LimitingCollectionName "All Users and User Groups" -RefreshSchedule $CollUpdate -RefreshType Both

Add-CMUserCollectionQueryMembershipRule -CollectionId $Collection.CollectionID -RuleName "Users OU" -QueryExpression "select \* from SMS\_R\_User where SMS\_R\_User.UserOUName = 'CLASSROOM.INTRANET/CLASSROOM/USERS'"

start-sleep 20

Get-CMCollectionMember -CollectionName "Users OU" | select Name

* 1. Collection Membership Incremental Evaluation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Sites |
| 03. Select 001 – Training Lab site and click Configuration Site Components -> Collection Membership Evaluation |
| 04. On Collection Membership Evaluation Component Properties change the default interval (minutes) to 3  Note: It is not recommended to decrease the default interval, however, there are situations (normally when there are too many collections – over thousands, or too many with incremental updates – near the recommended 200 limit) where you would need to increase the interval to decrease the CPU utilisation. |

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

$SiteCode = "001"

Set-CMCollectionMembershipEvaluationComponent -SiteCode $SiteCode -EvaluationMins 3

1. Remote Control

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001 |
| More information | Introduction to remote control in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/remote-control/introduction-to-remote-control>  Configuring remote control in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/remote-control/configuring-remote-control>  How to audit remote control usage in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/remote-control/audit-remote-control-usage> |
| Description | In this chapter, we will be creating a Device Client Setting to be used for Remote Tools as well as configuring, validating and monitoring access via MECM Remote Control |

* 1. Creating Device Settings for Remote Tools

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select Client Settings and click Create Custom Client Device Settings |
| 04. Type Remote Control for Windows 10 on Name and select Remote Tools under Select the custom settings to be enforced on client devices |
| 05. Click Remote Tools |
| 06. Click Configure and select Enable Remote Control on client computers and Domain  Note: Enabling Private and Public should only be used for non-domain machines. Using this option for domain machines may decrease its security depending on the settings you have selected. As example, if you select the Remote Desktop, the connection can be initiated from any network. |
| 07. Under Remote Tools click Set Viewers and add CLASSROOM\MECM Remote Tools. Click Ok  Note: This option is only required if the operator does not have administrative rights on target machines or if the Grant Remote Control permission to local Administrators group is set to No |
| 08. Under remote tools perform the following changes:   * Prompt user for Remote Control permission: yes * Play a sound on client: None * Manage Remote Desktop settings: Yes * Allow Permitted viewers to connect by using Remote Desktop Connection: Yes * Require network level authentication on computers that run Windows Vista operating system and later versions: No   Note: These settings are being used in a lab environment where security should not be an issue.  Click Ok |
| 09. Select the Remote Control for Windows 10 and click Deploy |
| 10. Under select collection, click Windows 10 Workstations and click Ok |
| 11. Select Deployments and confirm that the Client Settings has been deployed to the collection |

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

$ClientSettingsName = "Remote Control for Windows 10"

New-CMClientSetting -Name "$ClientSettingsName" -Type Device

Set-CMClientSetting -Name "$ClientSettingsName" -RemoteToolsSettings -AccessLevel FullControl -AllowPermittedViewersToRemoteDesktop $True -AllowRemoteControlOfUnattendedComputer $True -AudibleSignal PlayNoSound -FirewallExceptionProfile Domain -ManageRemoteDesktopSetting $True -ManageSolicitedRemoteAssistance $True -ManageUnsolicitedRemoteAssistance $True -PermittedViewer "CLASSROOM\MECM Remote Tools" -RemoteAssistanceAccessLevel FullControl -RequireAuthentication $False -PromptUserForPermission $True

Start-CMClientSettingDeployment -ClientSettingName "$ClientSettingsName" -CollectionName "Windows 10 Workstations"

* 1. Checking the Policies that should be applied

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Devices |
| 03. Select a Windows 10 Client and click Client Settings->Resultant Client Settings |
| 04. Once the Resultant Client Settings open, check the Remote Tools and confirm the changes have been made. Click Ok |

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

$DeviceList = get-cmdevice -Fast | where-object {$\_.DeviceOS -like 'Microsoft Windows NT Workstation 10.0\*'}

Get-CMResultantSettings -Name ($DeviceList[0].Name) -SettingsType Device -Setting 'RemoteTools'

* 1. Validating Remote Control

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Under Machine Policy Retrieval & Evaluation Cycle click Ok  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 04. Open Computer Management |
| 05. Expand Local users and Groups and select Groups |
| 06. Open properties of the ConfigMgr Remote Control Users and confirm that the Permitted viewers have been added to this group |

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

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

$groupName = "ConfigMgr Remote Control Users"

$LocalGroup = [ADSI]("WinNT://./$groupName,group")

$GMembers = $LocalGroup.psbase.invoke("Members")

$gmembers | foreach { $\_.GetType().InvokeMember("Name",'GetProperty', $null, $\_, $null) }

* 1. Starting Remote Control

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Assets and Compliance. |
| 02. Click Devices |
| 03. Select the Machine you want to remote access and click Start -> Remote Control |
| 04. The Remote Control will connect to the client |
| 05. On the client, it will need approve or Deny the connection if the client settings -> remote tools -> Prompt user for remote control permissions is set to yes |
| 06. Once the connection is established, the remote user will be able to access the machine |
| 07. The connection will remain active even when the user logs off.  Note: When Enhanced session is being used on the virtual machine, once the users logs off, the session will be closed |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ModulePath = $env:SMS\_ADMIN\_UI\_PATH

if ($ModulePath -eq $null) {

$ModulePath = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment").SMS\_ADMIN\_UI\_PATH

}

$ModulePath += "\CmRcViewer.exe"

$device = Get-CMDevice -Name "WKS0001"

if ($Device.IsClient -eq $true) { Start-Process -Filepath ("$ModulePath") -ArgumentList ("$($device.Name) \\$($servername)") } else { "Computer is not a MECM Client" }

* 1. Monitoring Remote Access from Client

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. On the client, you can also review the following client logs:   * C:\Windows\ccm\Logs\cmRcService.log: Records information for the remote control service. |

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

Start-Process -Filepath ("c:\windows\cmtrace.exe") -ArgumentList ("c:\Windows\ccm\Logs\cmRcService.log")

* 1. Monitoring Remote Access from Server

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Reporting and Click Reports |
| 03. Search for remote, select Remote Control – All remote control information and click Run |
| 04. Once the report opens, all information about successfully connections will be presented |
| 05. On the monitoring workspace, expand System Status and click Status Message Queries |
| 06. Search for Remote |
| 07. Select Remote Control Activity Targeted at a Specific System and click Show Messages |
| 08. Under Remote Control Activity Targeted at a Specific System select:   * Machine Name: WKS0001 * Time: Select date and time: 6 hours ago   Click Ok |
| 09. Verify the existence of Message ID 30076 |
| 10. Double click any 30076 messages to see its details. Once done, click Ok |
| 11. Verify the existence of Message ID 30077 |
| 12. Double click any 30077 messages to see its details. Once done, click Ok |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

#Open Report

Invoke-CMReport -ReportPath "Status Messages - Audit/Remote Control - All remote control information" -SiteCode "$SiteCode" -SrsServerName "$servername"

#Query:

$Date = (Get-Date).AddHours(-6)

gwmi -Namespace "root\sms\site\_$SiteCode" -ComputerName "$servername" -query "select stat.\*, ins.\*, att1.\*, stat.Time from SMS\_StatusMessage as stat left join SMS\_StatMsgInsStrings as ins on stat.RecordID = ins.RecordID left join SMS\_StatMsgAttributes as att1 on stat.RecordID = att1.RecordID inner join SMS\_StatMsgInsStrings as ins2 on stat.RecordID = ins2.RecordID where stat.MessageType = 768 and stat.MessageID >= 30069 and stat.MessageID <= 30087 and ins2.InsStrIndex = 2 and ins2.InsStrValue = 'WKS0001' and stat.Time >= '$($Date.ToString('yyyy/MM/dd HH:mm:ss.fff'))' order by stat.Time desc"

$remoteaccesslist = gwmi -Namespace "root\sms\site\_$SiteCode" -ComputerName "$servername" -query "select stat.Time, stat.MessageID, ins.InsStrIndex, ins.InsStrValue, att1.AttributeID, att1.AttributeTime, att1.AttributeValue from SMS\_StatusMessage as stat left join SMS\_StatMsgInsStrings as ins on stat.RecordID = ins.RecordID left join SMS\_StatMsgAttributes as att1 on stat.RecordID = att1.RecordID inner join SMS\_StatMsgInsStrings as ins2 on stat.RecordID = ins2.RecordID where stat.MessageType = 768 and stat.MessageID >= 30069 and stat.MessageID <= 30087 and ins2.InsStrIndex = 2 and ins2.InsStrValue = 'WKS0001' and stat.Time >= '$($Date.ToString('yyyy/MM/dd HH:mm:ss.fff'))' order by stat.Time desc"

foreach ($remoteaccess in $remoteaccesslist) {

$props = @{ 'Time'=$remoteaccess.stat.Time;

'MessageID'=$remoteaccess.stat.MessageID

'InsStrIndex'=$remoteaccess.ins.InsStrIndex

'InsStrValue'=$remoteaccess.ins.InsStrValue

'AttributeID'=$remoteaccess.att1.AttributeID

'AttributeTime'=$remoteaccess.att1.AttributeTime

}

$obj = new-object -TypeName psobject -Property $props

write-output $obj

}

1. Hardware Inventory

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001 |
| More information | Introduction to hardware inventory in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/inventory/introduction-to-hardware-inventory>  How to configure hardware inventory in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/manage/inventory/configure-hardware-inventory> |
| Description | In this chapter, we will look at Hardware Inventory, extend collection of data, capturing data from the client as well as looking at the new data from the MECM Server |

* 1. Changing Default Client Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select the default client settings and click Properties |
| 04. Under Default Settings, click Hardware Inventory and then, Set Classes… |
| 05. Make the following changes:   * Under Services (Win32\_Service) choose:   + State * Under Environment (Win32\_Environment) choose:   + Name   + User Name   + Caption   + Description   + Install Date   + Status   + System Variable   + Variable Value   Click Ok twice |

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

$Classes = @()

$Classes += New-CMInventoryReportClass -id 'MICROSOFT|SERVICE|1.0' -ReportProperty @('DisplayName','Name','PathName','ServiceType','StartMode','StartName','Status','State')

$Classes += New-CMInventoryReportClass -id 'MICROSOFT|ENVIRONMENT|1.0' -ReportProperty @('Name','UserName','Caption','Description','InstallDate','Status','SystemVariable','VariableValue')

Set-CMClientSettingHardwareInventory -DefaultSetting -AddInventoryReportClass @($classes)

* 1. Updating Client Policies

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Under Machine Policy Retrieval & Evaluation Cycle click Ok  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |

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

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

* 1. Starting Hardware Inventory

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Hardware Inventory Cycle  Note: Using this option will force the client to perform the hardware inventory. By default, this happen every once every 7 days and can be changed under Client Settings -> Hardware Inventory -> Hardware Inventory schedule |
| 03. Under Hardware Inventory Cycle click Ok |
| 04. On the client, you can also the following client logs:   * C:\Windows\ccm\Logs\InventoryAgent.log: Records activities of hardware inventory, software inventory, and heartbeat discovery actions on the client. |

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

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000001}")

Start-Sleep 60

* 1. Resource Explorer

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Devices |
| 03. Right click a Windows computer and click Start -> Resource Explorer |
| 04. Once the resource explorer open, expand hardware |
| 05. Select Environment |
| 06. Select Services and notice the new column |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ModulePath = $env:SMS\_ADMIN\_UI\_PATH

if ($ModulePath -eq $null) {

$ModulePath = (Get-ItemProperty -Path "Registry::HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment").SMS\_ADMIN\_UI\_PATH

}

$ModulePath = $ModulePath.Replace("bin\i386","bin\resourceexplorer.exe")

#windows machine

$Device = get-CMDevice -name "WKS0001"

Start-Process -Filepath ("$ModulePath") -ArgumentList ("-s -sms:ResourceID=$($Device.ResourceID) -sms:connection=\\$($servername)\root\sms\site\_$($siteCode)")

1. Client Health

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002 |
| More information | How to configure client status in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/clients/deploy/configure-client-status> |
| Description | In this chapter, we will look at the Client Health monitoring solution build in the MECM, we will understand how it works, reporting as well as break some client for reporting |

* 1. Client Status Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Click Client Status and then Client Status Settings |
| 03. On Client Status Settings Properties, change the default values to:   * Client policy requests during the following days: 1 * Heartbeat discovering during the following days: 1 * Hardware inventory during the following days: 1 * Software inventory during the following days: 1 * Status messages during the following days: 1   Note: These settings are being used in a lab environment. It is not recommended to decrease these numbers, however, sometimes you will need to increase these number to reflect individual company requirements.  Click OK |

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

Set-CMClientStatusSetting -ClientPolicyDays 1 -HardwareInventoryDays 1 -HeartbeatDiscoveryDays 1 -SoftwareInventoryDays 1 -StatusMessageDays 1

* 1. Client Health Configuration

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. On the client, open the Task Scheduler |
| 02. On the task scheduler, expand Task Scheduler (Local) -> Task Scheduler Library -> Microsoft and click Configuration Manager  Note: This Task is created during the client installation  Note: If you want to control when the client is executed and control the remediation procedures, use the CCMEVALINTERVAL, CCMEVALHOUR and NOTIFYONLY Client Installation Properties |
| 03. Select Configuration Manager Health Evaluation and click Properties |
| 04. On the General tab confirm that the task runs as a System |
| 05. Change to the Actions tab and confirm that it executes a program called ccmeval.exe |
| 06. Change to the Settings tab and confirm that it runs on demand and start if a scheduled start is missed.  Click Ok |
| 07. Open C:\windows\ccm\CcmEval.xml on Notepad and examine the validations  Note: MECM Client executes up to 30 validations (depending on OS version).  Note: Editing the ccmeval.xml is not supported. |

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

$Task = Get-ScheduledTask -TaskName "Configuration Manager Health Evaluation"

$task.Principal.UserId # should be System

$task.Actions.Execute # should be c:\windows\ccm\ccmeval.exe

$task.Settings.AllowDemandStart # should be true

$task.Settings.StartWhenAvailable # should be true

* 1. Executing CCMEVAL manually

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Command Prompt (run as administrator) |
| 02. Navigate to C:\Windows\CCM |
| 03. Type ccmeval.exe and press Enter |
| 04. You can also review the following logs:   * C:\Windows\ccm\logs\ccmeval.log: Records Configuration Manager Client status evaluation activities and details for components that are required by the Configuration Manager client. * C:\Windows\ccm\logs\CcmEvalTask.log: Records the Configuration Manager Client status evaluation activities that are initiated by the evaluation scheduled task. |

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

Start-Process -Filepath ("c:\windows\ccm\ccmeval.exe") -wait

* 1. Forcing CCMEVAL failure

|  |
| --- |
| Perform this task on the srv0001 virtual machine logged on as administrator |
| 01. Open Active Directory User and Computers |
| 02. Navigate to classroom.intranet->Classroom->Workstations->Enabled |
| 03. Move WKS0002 to classroom.intranet->Classroom->Workstations->Disabled |

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

#On SRV0001

Get-ADComputer WKS0002 | Move-ADObject -TargetPath 'OU=Disabled,OU=Workstations,OU=Classroom,DC=classroom,DC=intranet'

|  |
| --- |
| Perform this task on the wks0002 virtual machine logged on as administrator |
| 01. Start services console |
| 02. Search for Background Intelligent Transfer Service, it will exist. |
| 03. Open Command Prompt (run as administrator) |
| 04. Type gpupdate /force and press Enter and execute it again |
| 05. Start services console |
| 06. Search for Background Intelligent Transfer Service, it will not exist. |
| 07. Navigate to C:\Windows\CCM |
| 08. Type ccmeval.exe and press Enter |
| 09. You can also review the following logs:   * C:\Windows\ccm\logs\ccmeval.log: Records Configuration Manager Client status evaluation activities and details for components that are required by the Configuration Manager client. * C:\Windows\ccm\logs\CcmEvalTask.log: Records the Configuration Manager Client status evaluation activities that are initiated by the evaluation scheduled task. |

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

#On WKS0002

#get service information

Get-Service -Name BITS

#force group policy update

Start-Process -Filepath ("gpupdate") -ArgumentList ("/force") -wait -NoNewWindow

Start-sleep 10

#force group policy update

Start-Process -Filepath ("gpupdate") -ArgumentList ("/force") -wait -NoNewWindow

Start-sleep 10

#get service information, it will generate error if bits does not exist/access denied

Get-Service -Name BITS

#execute evaluation

Start-Process -Filepath ("c:\windows\ccm\ccmeval.exe") -wait

Start-sleep 60

* 1. Monitoring Client Health

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Client Status and click Client Check. |
| 03. Click Refresh Client Status |
| 04. Click client check passed.  Note: All links are clickable as well as the graphic |
| 05. Under the Client that passed client check from All Desktop and Server clients, select a WKS0001 and click Client Check details. |
| 06. Confirm that all rules passed evaluation and/or remediation (green button)  Note: if there is any remediation to be taken and passed, the rule name will be displayed. |
| 07. Click Monitoring |
| 08. Expand Client Status and click Client Check. Click client check failed. |
| 08. Under the Client that failed client check from All Desktop and Server clients, select a WKS0002 and click Client Check details. |
| 09. A Remediation Failed result is shown for the Verify BITS service exist with error Access Denied and another for Verfity/Remediate BITS Startup type with error Dependency Failed. |

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

#On SRV0002

$SiteCode = "001"

$task = Get-CMSiteSummaryTask -TaskName "Client Health Scheduled Task"

$Task.RunNow = $true

$Task.Put()

Start-Sleep 60

$Device = Get-CMDevice -Name "WKS0001" -Fast

gwmi -namespace "root\sms\site\_$SiteCode" -query "select \* from SMS\_CH\_EvalResult where ResourceID = $($Device.ResourceID)" | select HealthCheckDescription

$Device = Get-CMDevice -Name "WKS0002" -Fast

gwmi -namespace "root\sms\site\_$SiteCode" -query "select \* from SMS\_CH\_EvalResult where ResourceID = $($Device.ResourceID)" | select HealthCheckDescription

|  |
| --- |
| Perform this task on the srv0001 virtual machine logged on as administrator |
| 01. Open Active Directory User and Computers |
| 02. Navigate to classroom.intranet->Classroom->Workstations->Disabled |
| 03. Move WKS0002 to classroom.intranet->Classroom->Workstations-> Enabled |

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

#On SRV0001

Get-ADComputer "WKS0002" | Move-ADObject -TargetPath 'OU=Enabled,OU=Workstations,OU=Classroom,DC=classroom,DC=intranet'

|  |
| --- |
| Perform this task on the wks0002 virtual machine logged on as administrator |
| 01. Open Command Prompt (run as administrator) |
| 02. Type gpupdate /force and press Enter and execute it again |
| 03. Start services console |
| 04. Search for Background Intelligent Transfer Service. The service now exists and the Startup type is set to Manual and the Status is not running. |
| 05. Navigate to C:\Windows\CCM |
| 06. Type ccmeval.exe and press Enter |

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

#On WKS0002

#get service information

Get-Service -Name BITS

#force group policy update

Start-Process -Filepath ("gpupdate") -ArgumentList ("/force") -wait -NoNewWindow

Start-sleep 10

#force group policy update

Start-Process -Filepath ("gpupdate") -ArgumentList ("/force") -wait -NoNewWindow

Start-sleep 10

#get service information

Get-Service -Name BITS

#execute evaluation

Start-Process -Filepath ("c:\windows\ccm\ccmeval.exe") -wait

Start-sleep 60

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Client Status and click Client Check. |
| 03. Click Refresh Client Status |
| 04. Click client check passed.  Note: All links are clickable as well as the graphic |
| 05. Under the Client that passed client check from All Desktop and Server clients, select a WKS0002 and click Client Check details. |
| 06. Confirm that all rules passed evaluation and/or remediation (green button)  Note: if there is any remediation to be taken and passed, the rule name will be displayed. |

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

#On SRV0002

$SiteCode = "001"

$task = Get-CMSiteSummaryTask -TaskName "Client Health Scheduled Task"

$Task.RunNow = $true

$Task.Put()

Start-Sleep 60

$Device = Get-CMDevice -Name "WKS0002"

gwmi -namespace "root\sms\site\_$SiteCode" -query "select \* from SMS\_CH\_EvalResult where ResourceID = $($Device.ResourceID)" | select HealthCheckDescription

1. Software Metering

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002 |
| More information | Software metering in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/monitor-app-usage-with-software-metering> |
| Description | In this chapter, we will look at how to configure MECM to collect information about software when they run as well as we will look how we can report based on collected data |

* 1. Changing Default Client Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select the default client settings and click Properties |
| 04. Under Default Settings, click Software Metering and change the schedule from 7 days to 1 day |

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

$SiteCode = "001"

$ClientSettingsName = "Default Client Agent Settings"

$schedule = New-CMSchedule -RecurCount 1 -RecurInterval Days

Set-CMClientSetting -Name "$ClientSettingsName" -SoftwareMetering -Enable $True -Schedule $schedule

* 1. Updating Default Software Metering Settings and Clearing existing rules

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Software Metering |
| 03. Click Software Metering Properties |
| 04. Under Software Metering Properties, uncheck the Automatically create disabled metering rules from recent usage inventory data and click Ok |
| 05. Select all already create software metering rules and click delete. |
| 06. On the question Are you sure you want to delete these YY items?, click Yes |

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

#Disable auto-create sw metering rules

Set-CMSoftwareMeteringSetting -AutoCreateDisabledRule $False

#delete all already create sw metering rules

Get-CMSoftwareMeteringRule | Remove-CMSoftwareMeteringRule -Force

* 1. Creating Rule

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Software Metering |
| 03. Click Create Software Metering Rule |
| 04. Under Create Software Metering Rule Wizard, fill up with the following:   * Name: Notepad * File Name: notepad.exe * Original File Name: NOTEPAD.EXE.MUI * Version: \* * Language: - Any –   Click Next  Note: In version, we used \* instead of the version to capture any version  Note: In Language, we used – Any – to capture any language  Note: You can use the Browse button to have all settings filled up automatically  Note: All information can be view when you right click the application and select properties, and then go to the details tab |
| 05. Under Summary, click Next |
| 06. Under Completion, click Close |

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

New-CMSoftwareMeteringRule -ProductName Notepad -FileName notepad.exe -FileVersion \* -OriginalFileName NOTEPAD.EXE.MUI -LanguageId 65535

* 1. Starting Validation Software Metering

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Under Machine Policy Retrieval & Evaluation Cycle click Ok  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take a few minutes to complete |
| 04. Open notepad.exe and leave it open for couple of minutes, then close |
| 05. Open Control Panel and then click Configuration Manager |
| 06. Change to the Actions Tab, select Software Metering Usage Report Cycle and click Run Now  Note: Using this option will force the client to send report usage data to the server. By default, this happen every 7 days and can be changed under Client Settings -> Software Metering -> Schedule |
| 07. Under Software Metering Usage Report Cycle click Ok |

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

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

for($i=1; $i -le 3; $i++){

Start-Process -Filepath ("notepad.exe")

start-sleep 60

Stop-Process -Name notepad

start-sleep 10

}

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000031}")

Start-Sleep 60

* 1. Summarization Software Metering Data Manually

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open Command Prompt as administrator |
| 02. Navigate to "C:\ConfigMgr\tools\ServerTools” |
| 03. type runmetersumm.exe CM\_001 enter  Note: This task is managed by the Site Maintenance Tasks Summarize Software Metering File Usage Data and Summarize Software Metering Monthly Usage Data that, by default, runs every day between 00:00 and 05:00 |

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

$SiteCode = "001"

Start-Process -Filepath ("C:\ConfigMgr\tools\ServerTools\runmetersumm.exe") -ArgumentList ("CM\_$SiteCode") -wait -NoNewWindow

* 1. Monitoring Software Metering Data via Reports

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Under monitoring, expand Reporting and click Reports |
| 03. Search for Software Metering. Select Users that have run a specific metered software program and click Run |
| 04. Once the report is open, fill up the filters and click view report |
| 05. Select Time of day usage summary for a specific metered software program and click run |
| 06. Once the report is open, fill up the filters and click view report |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$swrule = Get-CMSoftwareMeteringRule -ProductName Notepad

$Date = Get-Date

#Open Report

$dict = @{"Rule Name"="$($swrule.ProductName)"; "Month (1-12)"="$($Date.Month)"; "Year"="$($Date.Year)" }

Invoke-CMReport -ReportPath "Software Metering/Users that have run a specific metered software program" -SiteCode "$SiteCode" -SrsServerName "$servername" -ReportParameter $dict

* 1. Monitoring Software Metering Data via Collections

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Device Collections |
| 03. Select Device Collections and click Create Device Collection |
| 04. Under Specify details for this collection type Computers that Run Notepad.exe Last 30 days under name and select All Systems under Limiting Collection. Click Next |
| 05. Under Define membership rules for this collection, select Add query and click Query Rule |
| 06. Under Query Rule Properties, type Software Metering Rule Notepad and click Edit Query Statement |
| 07. Under Query Statement Properties, click Show Query Language and type  select  \* from  SMS\_R\_System  inner join SMS\_MonthlyUsageSummary on SMS\_MonthlyUsageSummary.ResourceID = SMS\_R\_System.ResourceID  inner join SMS\_MeteredFiles on SMS\_MeteredFiles.FileID = SMS\_MonthlyUsageSummary.FileID and SMS\_MeteredFiles.SecurityKey = "00100037"  where DateDiff(dd, SMS\_MonthlyUsageSummary.LastUsage, GetDate()) < 30  Click Ok  Note: Change the 00100037 by the Rule ID you want to use |
| 08. Under Create Device Collection Wizard, select Use incremental updates for this collection and click Next  Note: It is not recommended to have over 250 collections with the Incremental updates option enabled because it might cause evaluation delays when enabled it for many collections. |
| 09. Under confirm the settings, click Next Twice |
| 10. Under The Create Device Collection Wizard completed successfully, click Close |
| 11. Under device collections, the new Collection is still under update status.  Note: Once the collection is created, there is a process to populate it and it may take a while. In this lab, wait 30 seconds or refresh it couple of times until you see Member Count change to 1 |
| 16. Select the collection and click Show Members |
| 17. The collection will be expanded under Devices and all devices that match the query filter will be displayed. |

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

$swrule = Get-CMSoftwareMeteringRule -ProductName Notepad

$CollUpdate = New-CMSchedule -Start "01/01/2015 9:00 PM" -DayOfWeek Saturday -RecurCount 1

$NewCol = New-CMDeviceCollection -Name "Computers that Run Notepad.exe Last 30 days" -LimitingCollectionName "All Systems" -RefreshSchedule $CollUpdate -RefreshType Both

Add-CMDeviceCollectionQueryMembershipRule -CollectionId $NewCol.CollectionID -RuleName "Software Metering Rule Notepad" -QueryExpression "select \* from SMS\_R\_System inner join SMS\_MonthlyUsageSummary on SMS\_MonthlyUsageSummary.ResourceID = SMS\_R\_System.ResourceID inner join SMS\_MeteredFiles on SMS\_MeteredFiles.FileID = SMS\_MonthlyUsageSummary.FileID and SMS\_MeteredFiles.SecurityKey = '$($swrule.SecurityKey)' where DateDiff(dd, SMS\_MonthlyUsageSummary.LastUsage, GetDate()) < 30"

start-sleep 20

Get-CMCollectionMember -CollectionName "Computers that Run Notepad.exe Last 30 days" | select Name

1. Primary Users

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0002 |
| More information | Associate users with a destination computer in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/osd/get-started/associate-users-with-a-destination-computer>  Link users and devices with user device affinity in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/link-users-and-devices-with-user-device-affinity> |
| Description | In this chapter, we will assign a primary user to a device. This can be useful when you want deploy applications only when users connect to their primary machines |

* 1. Changing Default Client Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select the default client settings and click Properties |
| 04. Click User and Device Affinity. Select Yes under the following:   * Automatically configure user device affinity from usage data * Allow user to define their primary devices   Click Ok |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ClientSettingsName = "Default Client Agent Settings"

Set-CMClientSetting -Name "$ClientSettingsName" -UserDeviceAffinitySettings -AllowUserAffinity $True -AutoApproveAffinity $True

* 1. Manual association by the Administrator

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Devices |
| 03. Select WKS0001 and click Edit Primary Users |
| 04. Search for User01 and click Add. Once done, click Ok |
| 05. Once back to the console, select the WKS0001 machine and click Primary Users under Related Objects |
| 06. A new node will appear under Users called Primary Users for the WKS0001 |

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

Get-CMUserDeviceAffinity -DeviceName "WKS0001"

Add-CMUserAffinityToDevice -DeviceName "WKS0001" -UserName "CLASSROOM\User01"

Get-CMUserDeviceAffinity -DeviceName "WKS0001"

* 1. Manual association by the User

|  |
| --- |
| Perform this task on the wks0002 virtual machine logged on as user02 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Under Machine Policy Retrieval & Evaluation Cycle click Ok  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 07. Click Start, Microsoft Endpoint Manager and click Software Center |
| 08. Click Options |
| 09. Under Work information, select “I regularly use this computer to do my work”. |

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

#On the Client

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

Start-Process -Filepath ("C:\Windows\CCM\ClientUX\SCClient.exe")

#continue steps 8 and 9

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Users |
| 03. Search for User02. Once done, select the CLASSROOM\User02 user and click Primary Device under Related Objects |
| 04. A new node will appear under Devices called Primary Devices for the CLASSROOM\User02 |

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

#On the Server

Get-CMUserDeviceAffinity -UserName "CLASSROOM\User02"

1. Application Management - Basic

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | Introduction to application management in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/understand/introduction-to-application-management>  Create applications with Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/create-applications> |
| Description | In this chapter, we will look at all steps required to create an application |

* 1. Creating Application

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Click Create Application |
| 04. Under Specify settings for this application, use the following:   * Type: Windows Installer (\*.msi file) * Location: \\srv0001\TrainingFiles\Source\Chrome for Windows\googlechromestandaloneenterprise64.msi   Click Next |
| 05. Under Import Information, click Next |
| 06. Under General Information confirm that the Installation program has been populated with a silent command line and click Next |
| 07. Under Summary, click Next |
| 08. Under The Create Application Wizard completed successfully click Close |

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

$AppName = "Google Chrome"

New-CMApplication -Name "$AppName"

Add-CMMsiDeploymentType -ApplicationName "$AppName" -ContentLocation "\\srv0001\TrainingFiles\Source\Chrome for Windows\googlechromestandaloneenterprise64.msi" -DeploymentTypeName "Google Chrome - Windows Installer (\*.msi file)"

* 1. Adding Requirements to Applications

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Applications Management and click Applications |
| 03. Select the Google Chrome application and click Deployment Types tab at the bottom of the console |
| 04. Select the Deployment Type and click Properties |
| 05. Under Google Chrome - Windows Installer (\*.msi file) Properties, General tab, change to Requirements tab |
| 06. Under the requirement tab, click Add |
| 07. Under Create Requirement, select the following information:   * Category: Device * Condition: Operating System * Rule Type: Value * Operator: One of * Values: All Windows 10 (64-bit)   Click Ok twice |

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

$AppName = "Google Chrome"

$DTName = "Google Chrome - Windows Installer (\*.msi file)"

$OS\_GC = Get-CMGlobalCondition -Name 'Operating System' | Where-Object { $\_.ModelName -eq 'GLOBAL/OperatingSystem' }

$OSx64\_GC = $OS\_GC | New-CMRequirementRuleOperatingSystemValue -PlatformString Windows/All\_x64\_Windows\_10\_and\_higher\_Clients -RuleOperator OneOf

$App = Get-CMApplication | where-object {$\_.LocalizedDisplayName -eq $AppName}

$DT = Get-CMDeploymentType -InputObject $App | Where-Object {$\_.LocalizedDisplayName -eq $DTName}

$DT | Set-CMDeploymentType -MsiOrScriptInstaller -ClearRequirements -AddRequirement $OSx64\_GC

* 1. Return Codes

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Applications Management and click Applications |
| 03. Select the Google Chrome application and click Deployment Types tab at the bottom of the console |
| 04. Select the Deployment Type and click Properties |
| 05. Under Google Chrome - Windows Installer (\*.msi file) Properties, General tab, change to Return Codes tab |
| 06. Under return codes, note that the list is already populated with the default return codes for the most common situations. Click Ok  Note: If the MSI app fails with an exit code in  {4, 5, 8, 13, 14, 39, 51, 53, 54, 55, 59, 64, 65, 67, 70, 71, 85, 86, 87, 112, 128, 170, 267, 999, 1003, 1203, 1219, 1220, 1222, 1231, 1232, 1238, 1265, 1311, 1323, 1326, 1330, 1618, 1622, 2250}, a retry will happen every 2 hours for up to 10 times. |

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

$AppName = "Google Chrome"

$DTName = "Google Chrome - Windows Installer (\*.msi file)"

([xml](Get-CMDeploymentType -ApplicationName "$AppName" -DeploymentTypeName "$DTName").SDMPackageXML).AppMgmtDigest.deploymenttype.installer.customdata.exitcodes.exitcode

* 1. Distributing Application Content to Distribution Point Group

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management, Applications |
| 03. Select Google Chrome and click Distribute Content |
| 04. Under Review selected content click Next |
| 05. Under Review the content to distribute, click Next |
| 06. Under Specify the content destination, click Add Distribution Point Group |
| 07. Under Add-Distribution Point Groups, select Training Lab and click Ok. Once back to the Specify the content destination, click Next |
| 08. Under Confirm the settings, click Next |
| 09. Under The distribute content wizard completed successfully, click Close |

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

Start-CMContentDistribution -ApplicationName "Google Chrome" -DistributionPointGroupName "Training Lab"

* 1. Monitoring Application Content Distribution via Console

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Distribution Status and click Content Status |
| 03. Select Google Chrome and verify the Completion Statistics. You can also view a break down when clicking View Status |
| 04. You can also review the following logs:   * C:\ConfigMgr\Logs\DistMgr.log: Records details about package creation, compression, delta replication, and information updates. * C:\ConfigMgr\Logs\PkgXferMgr.log: Records the actions of the SMS\_Executive component that is responsible for sending content from a primary site to a remote distribution point. |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$AppName = "Google Chrome"

Get-CMDistributionStatus | Where-Object {$\_.SoftwareName -eq $AppName} | select Targeted, NumberErrors, NumberInProgress, NumberSuccess, NumberUnknown

* 1. Monitoring Application Content Distribution via Reports

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Reporting, Reports and click Software Distribution – Content |
| 03. Select Application content distribution status and click Run |
| 04. Once the report opens, click Values, select Google Chrome application, click okand click View Report |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$AppName = "Google Chrome"

#Open Report

Invoke-CMReport -ReportPath "Software Distribution - Content/Application content distribution status" -SiteCode "$SiteCode" -SrsServerName "$servername" -ReportParameter @{"Application Name"="$($AppName)"}

1. Deploying and monitoring Applications

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002  WKS0004 |
| More information | Deploy applications with Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/deploy-applications>  Monitor applications from the Configuration Manager console  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/monitor-applications-from-the-console> |
| All Systems & All Users collection | Deploying anything to All Systems or All Users is not recommended, and it goes against best practices. We are using these 2 collections for demonstration purposes only. |
| Description | In this chapter, we will look at all steps required to deploy an existing application as well as monitor its deployment status |

* 1. Deploying Application

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Select Google Chrome and click Deploy |
| 04. Under Specify general information for this deployment, click Browse (Collection) and select the Collection you want to deploy. Click Next  Note: You can deploy to Devices as well as Users. In this example, we are using a Device Collection – Workstation OU |
| 05. Under Specify the content destination, click Next |
| 06. Under specify settings to control how this software is deployed, click Next  Note: Action can be Install or Uninstall and Purpose can be Available or Required. |
| 07. Under Specify the schedule for this deployment, click Next |
| 08. Under Specify the user experience for the installation of this software on the selected devices, click Next |
| 09. Under specify Configuration Manager and Operations Manager alert options, click Next |
| 10. Under Confirm the settings for this new deployment click Next |
| 11. Under Completion, click Close |

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

$AppName = "Google Chrome"

$ColName = "Workstation OU"

New-CMApplicationDeployment -CollectionName "$ColName" -Name "$AppName" -DeployAction Install -DeployPurpose Available

* 1. Installing available application

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then Configuration Manager. |
| 02. Change to the actions tab. Select Machine Policy Retrieval & Evaluation cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Once the Machine Policy Retrieval & Evaluation cycle message appears, click Ok.  Note: Depending on the MECM environment, the machine policy retrieval & evaluation cycle can take few minutes. |
| 04. Click Start, Microsoft Endpoint Manager and click Software Center |
| 05. Under Available Software, select Google Chrome and click Install |
| 06. If needed, click Installation Status tab to follow the installation process |
| 07. Once the installation is finished, click Installed Software to see all installed software  Note: Repeat the process on the WKS0002.  Note: You will not be able to install the application on WKS0004 as it does not meet the requirements |

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

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

$AppName = "Google Chrome"

$app = gwmi -Namespace 'root\CCM\ClientSDK' -Class 'CCM\_Application' | Where-Object { ($\_.Name -eq "$($AppName)") -and ($\_.InstallState -eq "NotInstalled") -and ($\_.AllowedActions -contains "Install")}

[int]$code = Invoke-WmiMethod -Namespace 'root\CCM\ClientSDK' -Class 'CCM\_Application' -Name Install -ArgumentList @(0, $app.Id, $app.IsMachineTarget, $false, 'High', $app.Revision) | select -ExpandProperty ReturnValue

if ($code -ne 0) {

throw "Error invoking Installation of '$($app.Name)' ($code)."

} else {

"Successfully invoked Installation of '$($app.Name)'."

}

* 1. Monitoring Application Deployment via Console

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Click Deployments |
| 03. Select Google Chrome and click Summarization  Note: The Application Deployment summarization runs once every 60 minutes by default, this can be changed on Administration -> Site Configuration -> Sites -> <Site> -> Status Summarizers -> Application Deployment Summarizer |
| 04. Click Ok once the Configuration Manager information screen appears |
| 05. After the summarization, under Completion Statistics, view Status.  Click View Status for more information |
| 06. Under View Status, Asset Details, you can see which device received the software and Under Requirements Not Met you can see which device did not have the software installed. |

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

$AppName = "Google Chrome"

$ColName = "Workstation OU"

Get-CMDeployment -CollectionName "$ColName" -SoftwareName "$AppName" | Invoke-CMDeploymentSummarization

Start-Sleep 10

Get-CMDeployment -CollectionName "$ColName" -SoftwareName "$AppName" | select ApplicationName, CollectionName, NumberErrors, NumberInProgress, NumberOther, NumberSuccess, NumberTargeted, NumberUnknown

Get-CMApplication -Name $AppName | Get-CMApplicationDeploymentStatus | Get-CMDeploymentStatusDetails | select MachineName, ComplianceState, InstalledState, StatusType

* 1. Monitoring Application Deployment via Client Logs

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. On the client, you can also review the following client logs:   * C:\Windows\ccm\Logs\AppDiscovery.log: Records details about the discovery or detection of applications on client computers. * C:\Windows\ccm\Logs\AppEnforce.log: Records details about enforcement actions (install and uninstall) taken for applications on the client. * C:\Windows\ccm\Logs\ContentTransferManager.log: Schedules the Background Intelligent Transfer Service (BITS) or the Server Message Block (SMB) to download or to access packages. * C:\Windows\ccm\Logs\DataTransferService.Log: Records all BITS communication for policy or package access. * C:\Windows\ccm\Logs\LocationServices.log: Records the client activity for locating management points, software update points, and distribution points. * C:\Windows\ccm\Logs\SCCClient\_<Domain>@<User>\_1.log: Records the activity in Software Center for the specified user on the client computer. * C:\Windows\ccm\Logs\SCCClient\_<Domain>@<User>\_2.log: Records the historical activity in Software Center for the specified user on the client computer. * C:\Windows\ccm\Logs\SCNotify\_<Domain>@<User>\_1.log: Records the activity for notifying users about software for the specified user. * C:\Windows\ccm\Logs\SCNotify\_<Domain>@<User>\_2.log: Records the historical information for notifying users about software for the specified user. |

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

Start-Process -Filepath ("c:\windows\cmtrace.exe") -ArgumentList ("c:\Windows\ccm\Logs\AppDiscovery.log c:\Windows\ccm\Logs\AppEnforce.log c:\Windows\ccm\Logs\ContentTransferManager.log c:\Windows\ccm\Logs\DataTransferService.Log")

* 1. Monitoring Application Deployment via Reports

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Reporting, Reports and click Software Distribution – Application Monitoring |
| 03. Select Application Compliance and click Run |
| 04. Under Application Compliance reports, use Google Chrome for Application and Workstation OU for collection and click View report  Note: You can drill down to a more specific report using the links inside the reports |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$AppName = "Google Chrome"

$ColName = "Workstation OU"

#Open Report

Invoke-CMReport -ReportPath "Software Distribution - Application Monitoring/Application compliance" -SiteCode "$SiteCode" -SrsServerName "$servername" -ReportParameter @{"Application"="$($AppName)"; "Collection"="$($ColName)"}

1. Application Management – App-V 5 Applications

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001 |
| More information | Create App-V virtual environments in System Center Configuration Manager  <https://docs.microsoft.com/en-us/sccm/apps/deploy-use/create-app-v-virtual-environments> |
| Description | In this chapter, we will look at all steps required to create an application that depends on some requirements |

* 1. Creating an Application App-V Client 5.0

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Select Application and Click Create Application |
| 04. Under Specify Settings for this application, select Manually specify the application information and click Next |
| 05. Under Specify information about this application type App-V Client 5.0 as Name and click Next |
| 06. Under Specify the Configuration Manager Application Catalog entry click Next |
| 07. Under Configure deployment types and the priority in which they will be applied for this application click Add |
| 08. Under Specify settings for this deployment type, select Script installer as type and click Next |
| 09. Under specify general information for this deployment type App-V client for Windows 8 x64 as Name and click Next |
| 10. Under Specify information about the content to be delivered to target devices, fill up the following information:   * Content location: \\srv0001\trainingfiles\source\App-V5 Client * Installation program: "appv\_client\_setup.exe" /q /ACCEPTEULA * Uninstall program: "appv\_client\_setup.exe" /UNINSTALL /q   Click Next |
| 11. Under Specify how this deployment type is detected, click Add Clause |
| 12. Under detection rule, fill up the following:   * Settings Type: Windows Installer * Product Code: {6313DBA3-0CA9-4CD8-93B3-373534146B7B}   Click Ok |
| 11. Under Specify how this deployment type is detected, click Add Clause |
| 13. Under detection rule, fill up the following:   * Settings Type: Registry * Hive: HKEY\_LOCAL\_MACHINE * Key: SOFTWARE\Microsoft\AppV\Client * Value: Version * Data Type: String * This registry setting must satisfy the following rule to indicate the presence of this application * Operator: Begins with * Value: 5.2   Click Ok and Once back to the specify how this deployment type is detected, change the connector from And to Or and click Next |
| 14. Under specify user experience settings for the application, fill up the following:   * Installation behaviour: Install for system * Logon requirement: Whether or not a user logged on   Click Next |
| 15. Under Specify installation requirements for this deployment type, click Add |
| 16. Under Create Requirement, select the following information:   * Category: Device * Condition: Operating System * Rule Type: Value * Operator: One of * Values: All Windows 8.1 (64-bit) and All Windows 10 (64-bit)   Click Ok and once back to the Specify installation requirements for this deployment type click Next |
| 17. Under specify software dependencies for this deployment type, click Next |
| 18. Under Confirm the settings for this deployment type, click Next |
| 19. Under The create Deployment Type Wizard completed successfully, click Close |
| 20. Once back to Create Application Wizard, click Next |
| 21. Under confirm the settings for this application, click Next |
| 22. Under The Create Application Wizard completed successfully, click Close |

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

$AppName = "App-V Client 5.0"

$DTName = "App-V client for Windows 8 x64"

$MSIProductCode = "{6313DBA3-0CA9-4CD8-93B3-373534146B7B}"

$OS\_GC = Get-CMGlobalCondition -Name 'Operating System' | Where-Object { $\_.ModelName -eq 'GLOBAL/OperatingSystem' }

$OSx64\_GC = $OS\_GC | New-CMRequirementRuleOperatingSystemValue -PlatformString Windows/All\_x64\_Windows\_8.1\_Client, Windows/All\_x64\_Windows\_10\_and\_higher\_Clients -RuleOperator OneOf

$App = New-CMApplication -Name "$AppName"

$DT = Add-CMScriptDeploymentType -ApplicationName "$AppName" -DeploymentTypeName "$DTName" -InstallCommand '"appv\_client\_setup.exe" /q /ACCEPTEULA' -ProductCode "$MSIProductCode" -ContentLocation "\\srv0001\trainingfiles\source\App-V5 Client" -LogonRequirementType WhereOrNotUserLoggedOn -UninstallCommand '"appv\_client\_setup.exe" /UNINSTALL /q'

$DT | Set-CMDeploymentType -MsiOrScriptInstaller -InstallationBehaviorType InstallForSystem -ClearRequirements -AddRequirement $OSx64\_GC

$DT | Set-CMScriptDeploymentType -AddDetectionClause $rule -DetectionClauseConnector @{"LogicalName"=$rule.Setting.LogicalName;"Connector"="OR"}

* 1. Creating a Virtual Application Robocopy

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Select Application and Click Create Application |
| 04. Under Specify Settings for this application, select Microsoft Application Virtualization 5 as Type and \\srv0001\trainingfiles\Source\Robocopy App-V5\Robocopy.appv as Location and click Next |
| 05. Under View imported information, click Next |
| 06. Under Specify information about this application, click Next |
| 07. Under Confirm the settings for this application, click Next |
| 08. Under The create application wizard complete successfully, click Close |
| 09. Select the Robocopy application and click Deployment Types tab at the bottom of the console |
| 11. Select the Deployment Type and click Properties |
| 12. Under Robocopy – Microsoft Application Virtualization 5, change to Content tab |
| 13. On Content tab, under Select the deployment option to use when a client is connected with a fast (LAN) network boundary, change deployment options to Stream content from distribution point.  Change to the Requirements tab |
| 14. Under the requirement tab, click Add |
| 15. Under Create Requirement, select the following information:   * Category: Device * Condition: Operating System * Rule Type: Value * Operator: One of * Values: All Windows 8.1 (64-bit) and All Windows 10 (64-bit)   Click Ok and change to the Dependencies tab |
| 16. Under Dependencies, click Add |
| 17. Under Add Dependency, type App-V Client as Dependency group name and click Add |
| 18. Under Specify required application Select App-V Client 5.0 and check App-V Client for Windows 8 x64. Click Ok three times  Note: Once done, distribute the application content to DP Group and note that all requirements are also going to be distributed |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$ParentAppName = "App-V Client 5.0"

$AppName = "Robocopy"

$DTName = "Robocopy - Microsoft Application Virtualization 5"

$OS\_GC = Get-CMGlobalCondition -Name 'Operating System' | Where-Object { $\_.ModelName -eq 'GLOBAL/OperatingSystem' }

$OSx64\_GC = $OS\_GC | New-CMRequirementRuleOperatingSystemValue -PlatformString Windows/All\_x64\_Windows\_8.1\_Client, Windows/All\_x64\_Windows\_10\_and\_higher\_Clients -RuleOperator OneOf

$App = New-CMApplication -Name "$AppName"

$DT = Add-CMAppv5XDeploymentType -ApplicationName "$AppName" -ContentLocation "\\srv0001\trainingfiles\Source\Robocopy App-V5\Robocopy.appv" -DeploymentTypeName "$DTName" -FastNetworkDeploymentMode DownloadContentForStreaming

$DT | New-CMDeploymentTypeDependencyGroup -GroupName "App-V Client" | Add-CMDeploymentTypeDependency -DeploymentTypeDependency (Get-CMDeploymentType -ApplicationName "$ParentAppName") -IsAutoInstall $true

$DT | Set-CMDeploymentType -ClearRequirements -AddRequirement $OSx64\_GC

Start-CMContentDistribution -ApplicationName "$AppName" -DistributionPointGroupName "Training Lab"

* 1. Visualizing Application Relationship

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as sccmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Select Robocopy and click View Relationships -> Dependency |
| 04. The Robocopy Dependencies will open and you will be able to see all software dependencies |

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

. c:\TrainingFiles\Scripts\ShowDependentApplications\_v0\_9.ps1

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$AppName = "App-V Client 5.0"

ShowDependentApplications -ApplicationName "$AppName" -SiteCode "$SiteCode" -SiteServer "$servername"

1. Deploying and monitoring available Virtual Applications to a User

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002  WKS0004 |
| More information | Deploy applications with Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/deploy-applications>  Monitor applications from the Configuration Manager console  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/monitor-applications-from-the-console>  Enable the App-V in-box client  <https://docs.microsoft.com/en-gb/windows/application-management/app-v/appv-enable-the-app-v-desktop-client> |
| Description | In this chapter, we will look at all steps required to deploy an App-V application to a Windows 10 (with Update Anniversary) as well as Windows 8.1 where an App-V client needs to be installed before |

* 1. Enable App-V in-box

|  |
| --- |
| Perform this task on the WKS0001 virtual machine logged on as user01 |
| 01. Open Windows PowerShell (as administrator) |
| 02. Type Enable-Appv and press ENTER  Note: If you get an error “Enable-Appv : App-V is not available on this edition of Windows.”, make sure you are running an Enterprise version of Windows 10. App-V is not available on Windows Professional. For a workaround, refer to <http://blog.eastern.nl/2016/08/run-or-enable-microsoft-app-v-client-on-windows-10-anniversary-update/> |
| 03. Restart the device.  Note: Repeat the process on WKS0002 |

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

Enable-Appv

start-sleep 5

Start-Process -Filepath ("shutdown") -ArgumentList ("/r /t 0")

* 1. Deploying Application

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Select Robocopy and click Deploy |
| 04. Under Specify general information for this deployment, click Browse (Collection) and select the Collection you want to deploy. Click Next  Note: You can deploy to a Device as well as Users. In this example, we are using the user collection - Users OU |
| 05. Under Specify the content destination, click Next |
| 06. Under specify settings to control how this software is deployed, click Next  Note: Action can be Install or Uninstall and Purpose can be Available or Required. |
| 07. Under Specify the schedule for this deployment, click Next |
| 08. Under Specify the user experience for the installation of this software on the selected devices, click Next |
| 09. Under specify Configuration Manager and Operations Manager alert options, click Next |
| 10. Under Confirm the settings for this new deployment click Next |
| 11. Under completion, click Close |

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

New-CMApplicationDeployment -CollectionName "Users OU" -Name "Robocopy"

* 1. Installing available application

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Software Center under Start -> Microsoft Endpoint Manager |
| 02. Under Available Software, select Robocopy and click Install |
| 03. If needed, click Installation Status tab to follow the installation process |
| 04. Once the installation is finished, click Installed Software to see all installed software  Note: Repeat the process on the WKS0002 and WKS0004  Note: The App-V 5.0 Client will only be installed on WKS0004 |

1. Application Management – Advanced

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002  WKS0004 |
| More information | How to revise and supersede applications in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/revise-and-supersede-applications>  How to create global conditions in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/create-global-conditions>  Simulate application deployments with Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/apps/deploy-use/simulate-application-deployments> |
| Description | In this chapter, we will look at all steps required to replace an application with a new version; steps required to deploy an application that requires approval from the MECM administrator; creating custom deployment requirements as well as perform simulated deployment, so we can test the detection method, requirements and dependencies of an application deployment without installing or uninstalling the application. |

* 1. Preparing the Environment

Before Starting this Module, you will need to:

1. Create a Firefox 40 Application with the Following Information, Deploy and Install

* Specific Settings for this application select Manually specify
* Name: Firefox 40
* Deployment Type: Script
* Name: Firefox 40 Installation for Windows 8.1 and 10
* Content Location: \\srv0001\TrainingFiles\Source\Firefox 40
* Install program: "Firefox Setup 40.0.exe" -ms
* Uninstall program: "C:\Program Files (x86)\Mozilla Firefox\uninstall\helper.exe" -ms
* Detection Rule:
  + Setting Type: File System
  + Type: File
  + File: %ProgramFiles%\Mozilla Firefox
  + File or folder name: firefox.exe
  + This file or folder is associated with a 32-bit application on 64-bit systems checked
  + Property: Version
  + Operator: Greater than or equal to
  + Value: 40.0
* Installation behaviour: Install for a system
* Logon requirement: Whether or not a user is logged on
* Requirements: Operating System, one of, All Windows 10 (64-bit), All Windows 8.1 (64-bit)
* Distribute content to a Distribution Point Group
* Make it available to Users OU collection (Available deployment)
* Log on to WKS0001, WKS0004 as user01 and install the software
* Log on to WKS0002 as user02 and install the software

1. Create a Firefox 49 Application with the Following Information

* Specific Settings for this application select Manually specify
* Name: Firefox 49
* Deployment Type: Script
* Name: Firefox 49 Installation for Windows 8.1 and 10
* Content Location: \\srv0001\TrainingFiles\Source\Firefox 49
* Install program: "Firefox Setup 49.0.1.exe" -ms
* Uninstall program: "C:\Program Files (x86)\Mozilla Firefox\uninstall\helper.exe" -ms
* Detection Rule:
  + Setting Type: File System
  + Type: File
  + File: %ProgramFiles%\Mozilla Firefox
  + File or folder name: firefox.exe
  + This file or folder is associated with a 32-bit application on 64-bit systems checked
  + Property: Version
  + Operator: Greater than or equal to
  + Value: 49.0
* Installation behaviour: Install for a system
* Logon requirement: Whether or not a user is logged on
* Requirements: Operating System, one of, All Windows 10 (64-bit), All Windows 8.1 (64-bit)
* Distribute content to a Distribution Point Group

1. Create a Java 8 Application with the Following Information:

* Specific Settings for this application select Manually specify
* Name: Java8
* Deployment Type: Script
* Name: Java8 for Windows 10
* Content Location: \\srv0001\trainingfiles\Source\Java8
* Install program: "Java8.exe" /s
* Detection Rule:
  + Setting Type: File System
  + Type: fileA
  + File: %ProgramFiles%\Java\jre1.8.0\_101\bin
  + File or folder name: java.exe
  + This file or folder is associated with a 32-bit application on 64-bit systems checked
  + Property: Version
  + Operator: Greater than or equal to
  + Value: 8.0.1010
* Installation behaviour: Install for a system
* Logon requirement: Whether or not a user is logged on
* Requirements: Operating System, one of, All Windows 10 (64-bit)
* Distribute content to a Distribution Point Group

1. Create an Acrobat Reader XI Application with the Following Information:

* Specific Settings for this application select Manually specify
* Name: Acrobat Reader XI
* Deployment Type: Script
* Name: Acrobat Reader XI for Windows 10
* Content Location: \\srv0001\trainingfiles\Source\AdobeXI
* Install program: "AdbeRdr11010\_en\_US.exe" /msi EULA\_ACCEPT=YES REMOVE\_PREVIOUS=YES /qn
* Detection Rule:
  + Setting Type: Windows Installer
  + Product Code: {AC76BA86-7AD7-1033-7B44-AB0000000001}
* Installation behaviour: Install for a system
* Logon requirement: Whether or not a user is logged on
* Requirements: Operating System, one of, All Windows 10 (64-it)
* Distribute content to a Distribution Point Group

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

$OS\_GC = Get-CMGlobalCondition -Name 'Operating System' | Where-Object { $\_.ModelName -eq 'GLOBAL/OperatingSystem' }

$OSx64\_GC = $OS\_GC | New-CMRequirementRuleOperatingSystemValue -PlatformString Windows/All\_x64\_Windows\_8.1\_Client, Windows/All\_x64\_Windows\_10\_and\_higher\_Clients -RuleOperator OneOf

$AppName = "Firefox 40"

$DTName = "Firefox 40 Installation for Windows 8.1 and 10"

$App = New-CMApplication -Name "$AppName"

$DetectionRule = New-CMDetectionClauseFile -ExpectedValue "40.0" -ExpressionOperator GreaterEquals -FileName "firefox.exe" -Path "%ProgramFiles%\Mozilla Firefox" -PropertyType Version -Value

$DT = Add-CMScriptDeploymentType -AddDetectionClause $DetectionRule -ApplicationName $AppName -DeploymentTypeName $DTName -InstallCommand '"Firefox Setup 40.0.exe" -ms' -ContentLocation '\\srv0001\TrainingFiles\Source\Firefox 40' -InstallationBehaviorType InstallForSystem -LogonRequirementType WhereOrNotUserLoggedOn -UninstallCommand '"C:\Program Files (x86)\Mozilla Firefox\uninstall\helper.exe" -ms'

$DT | Set-CMDeploymentType -ClearRequirements -AddRequirement $OSx64\_GC

Start-CMContentDistribution -ApplicationName "$AppName" -DistributionPointGroupName "Training Lab"

New-CMApplicationDeployment -CollectionName "Users OU" -Name "$AppName"

$AppName = "Firefox 49"

$DTName = "Firefox 49 Installation for Windows 8.1 and 10"

$App = New-CMApplication -Name "$AppName"

$DetectionRule = New-CMDetectionClauseFile -ExpectedValue "49.0" -ExpressionOperator GreaterEquals -FileName "firefox.exe" -Path "%ProgramFiles%\Mozilla Firefox" -PropertyType Version -Value

$DT = Add-CMScriptDeploymentType -AddDetectionClause $DetectionRule -ApplicationName $AppName -DeploymentTypeName $DTName -InstallCommand '"Firefox Setup 49.0.1.exe" -ms' -ContentLocation '\\srv0001\TrainingFiles\Source\Firefox 49' -InstallationBehaviorType InstallForSystem -LogonRequirementType WhereOrNotUserLoggedOn -UninstallCommand '"C:\Program Files (x86)\Mozilla Firefox\uninstall\helper.exe" -ms'

$DT | Set-CMDeploymentType -MsiOrScriptInstaller -InstallationBehaviorType InstallForSystem -ClearRequirements -AddRequirement $OSx64\_GC

Start-CMContentDistribution -ApplicationName "$AppName" -DistributionPointGroupName "Training Lab"

$AppName = "Java8"

$DTName = "Java8 for Windows 10"

$App = New-CMApplication -Name "$AppName"

$DetectionRule = New-CMDetectionClauseFile -ExpectedValue "8.0.1010" -ExpressionOperator GreaterEquals -FileName "java.exe" -Path "%ProgramFiles%\Java\jre1.8.0\_101\bin" -PropertyType Version -Value

$DT = Add-CMScriptDeploymentType -AddDetectionClause $DetectionRule -ApplicationName $AppName -DeploymentTypeName $DTName -InstallCommand '"Java8.exe" /s' -ContentLocation '\\srv0001\TrainingFiles\Source\Java8' -InstallationBehaviorType InstallForSystem -LogonRequirementType WhereOrNotUserLoggedOn

$DT | Set-CMDeploymentType -ClearRequirements -AddRequirement $OSx64\_GC

Start-CMContentDistribution -ApplicationName "$AppName" -DistributionPointGroupName "Training Lab"

$AppName = "Acrobat Reader XI"

$DTName = "Acrobat Reader XI for Windows 10"

$App = New-CMApplication -Name "$AppName"

$DT = Add-CMScriptDeploymentType -ApplicationName "$AppName" -DeploymentTypeName "$DTName" -InstallCommand '"AdbeRdr11010\_en\_US.exe" /msi EULA\_ACCEPT=YES REMOVE\_PREVIOUS=YES /qn' -ProductCode "{AC76BA86-7AD7-1033-7B44-AB0000000001}" -ContentLocation "\\srv0001\trainingfiles\Source\AdobeXI" -InstallationBehaviorType InstallForSystem -LogonRequirementType WhereOrNotUserLoggedOn

$DT | Set-CMDeploymentType -ClearRequirements -AddRequirement $OSx64\_GC

Start-CMContentDistribution -ApplicationName "$AppName" -DistributionPointGroupName "Training Lab"

* 1. Creating an Application Supersedence

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Select Firefox 49 and click Properties |
| 04. Under Firefox 49 Properties, change to the Supersedence tab |
| 05. Under supersedence tab click Add |
| 06. Under Specify supersedence relationship click Browse |
| 07. Under Choose Application, click Firefox 40 and click Ok |
| 08. Once back to the Specify Supersedence Relationship, change the New Deployment Type to Firefox 49 Install for Windows 8.1 and 10, check Uninstall and click Ok twice |

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

$OldAppName = "Firefox 40"

$OldDTName = "Firefox 40 Installation for Windows 8.1 and 10"

$NewAppName = "Firefox 49"

$NewDTName = "Firefox 49 Installation for Windows 8.1 and 10"

$OldDeploymentType = Get-CMDeploymentType -ApplicationName "$OldAppName" -DeploymentTypeName "$OldDTName"

$NewDeploymentType = Get-CMDeploymentType -ApplicationName "$NewAppName" -DeploymentTypeName "$NewDTName"

Add-CMDeploymentTypeSupersedence -SupersededDeploymentType $OldDeploymentType -SupersedingDeploymentType $NewDeploymentType -IsUninstall $true

* 1. Creating an Application Supersedence Deployment

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Once back to the select Firefox 49 and click Deploy |
| 02. Under Specify general information for this deployment, select Users OU Collection and click Next |
| 03. Under Specify the content destination, click Next |
| 04. Under Specify settings to control how this software is deployed, select the following:   * Action: Install * Purpose: Available * Automatically upgrade any superseded version of this application: Checked   Click Next |
| 05. Under Specify the schedule for this deployment, click Next |
| 06. Under specify the user experience for the installation of this software on the selected devices, click Next |
| 07. Under Specify Configuration Manager and Operations manager alert options, click Next |
| 08. Under Confirm the settings for this new deployment, click Next |
| 09. Under The Deploy Software Wizard completed successfully, click Close |

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

$SiteCode = "001"

$AppName = "Firefox 49"

$CollName = "Users OU"

New-CMApplicationDeployment -Name "$AppName" -CollectionName "$CollName" -UpdateSupersedence $True

* 1. Installing an Application Supersedence

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then Configuration Manager. |
| 02. Change to the actions tab. Select User Policy Retrieval & Evaluation cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Once the User Policy Retrieval & Evaluation cycle message appears, click Ok.  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 04. Click Start, Microsoft Endpoint Manager and click Software Center |
| 05. Under Available Software, for a short period you should see both Firefox 40 and Firefox 49 available. Short after you should see the Installation of Firefox 49 start |
| 06. If needed, click Installation Status tab to follow the installation process. Repeat the process on WKS0002 and WKS0004 |

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

$UserSID = ((gwmi -query "select \* from win32\_useraccount where name = '$($env:USERNAME)' and domain='$($env:USERDOMAIN)'").SID).replace('-','\_')

$sched = ([wmi]"root\ccm\Policy\$UserSID\ActualConfig:CCM\_Scheduler\_ScheduledMessage.ScheduledMessageID='{00000000-0000-0000-0000-000000000026}'");

$sched.Triggers = @('SimpleInterval;Minutes=1;MaxRandomDelayMinutes=0');

$sched.Put()

start-sleep 10

$sched = ([wmi]"root\ccm\Policy\$UserSID\ActualConfig:CCM\_Scheduler\_ScheduledMessage.ScheduledMessageID='{00000000-0000-0000-0000-000000000027}'");

$sched.Triggers = @('SimpleInterval;Minutes=1;MaxRandomDelayMinutes=0');

$sched.Put()

start-sleep 60

$AppName = "Firefox 49"

$app = gwmi -Namespace 'root\CCM\ClientSDK' -Class 'CCM\_Application' | Where-Object { ($\_.Name -eq "$($AppName)") -and ($\_.InstallState -eq "NotInstalled") -and ($\_.AllowedActions -contains "Install")}

[int]$code = Invoke-WmiMethod -Namespace 'root\CCM\ClientSDK' -Class 'CCM\_Application' -Name Install -ArgumentList @(0, $app.Id, $app.IsMachineTarget, $false, 'High', $app.Revision) | select -ExpandProperty ReturnValue

if ($code -ne 0) {

throw "Error invoking Installation of '$($app.Name)' ($code)."

} else {

"Successfully invoked Installation of '$($app.Name)'."

}

* 1. Creating a Global Condition

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Global Conditions |
| 03. Click Create Global Condition |
| 04. Under Specify details about this global condition, use the following:   * Name: Computer Model * Device Type: Windows * Condition type: Setting * Setting type: WQL Query * Data Type: String * Namespace: root\cimv2 * Class: Win32\_ComputerSystem * Property: Model   Click Ok  Note: Be sure that you typed the information correctly as, once it is used, it cannot be changed |

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

New-CMGlobalConditionWqlQuery -Class "Win32\_ComputerSystem" -DataType String -Name "Computer Model" -Property "Model" -Namespace "root\cimv2"

* 1. Using a Custom Global Condition

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Applications Management and click Applications |
| 03. Select the Java8 and click Deployment Types tab at the bottom of the console |
| 04. Select the Deployment Type and click Properties |
| 05. Under Java8 for Windows 10 Properties, change to Requirements tab and click Add |
| 06. Under Create Requirement, create a new requirement with the following information:   * Category: Custom * Condition: Computer Model * Rule Type: Value * Operator: Equals * Value: SVD11225CYB   Click Ok Twice |

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

$GC = Get-CMGlobalCondition -Name "Computer Model"

$GC\_Model = $GC | New-CMRequirementRuleCommonValue -RuleOperator IsEquals -Value1 'SVD11225CYB'

$AppName = "Java8"

$DTName = "Java8 for Windows 10"

$DT = Get-CMDeploymentType -ApplicationName $AppName -DeploymentTypeName $DTName

$DT | Set-CMDeploymentType -AddRequirement $GC\_Model

* 1. Deploying an Approval Required Application

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Applications Management and click Applications |
| 03. Select Java8 and click Deploy |
| 04. Under Specify general information for this deployment, click Browse (Collection) and select the Collection you want to deploy. Click Next  Note: You can deploy to a Device as well as Users. In this example, we are using Users OU collection |
| 05. Under Specify the content destination, click Next |
| 06. Under specify settings to control how this software is deployed, select Require administrator approval if users request this application and click Next |
| 07. Under Specify the schedule for this deployment, click Next |
| 08. Under Specify the user experience for the installation of this software on the selected devices, click Next |
| 09. Under specify Configuration Manager and Operations Manager alert options, click Next |
| 10. Under Confirm the settings for this new deployment click Next |
| 11. Under Completion, click Close |

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

New-CMApplicationDeployment -CollectionName "Users OU" -Name "Java8" -ApprovalRequired $true

* 1. Requesting application via Software Center

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Software Center under Start -> Microsoft Endpoint Manager |
| 02. Under Available Software, select Java8 |
| 03. On Java8, type some information on the text box and click Request |
| 04. Once the request is submitted, a message will be displayed that the request was submitted. |

* 1. Approving/Denying application requests

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Approval Requests |
| 03. Right click the visible columns and add Comments |
| 04. Once done, select the application, and click Approve |
| 05. Under Application Request(s) add an Approver’s Comments and click OK |

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

#approve request

Get-CMApprovalRequest -ApplicationName "Java8" -User "CLASSROOM\User01" | Approve-CMApprovalRequest

#deny request

#Get-CMApprovalRequest -ApplicationName "Java8" -User "CLASSROOM\User01" | Deny-CMApprovalRequest

* 1. Installing an approved application

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then Configuration Manager. |
| 02. Change to the actions tab. Select User Policy Retrieval & Evaluation cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Once the User Policy Retrieval & Evaluation cycle message appears, click Ok.  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 04. Click Start, Microsoft Endpoint Manager and click Software Center |
| 05. Under Available Software, select Java 8 and click Install  Note: The installation will not be completed and the Status will be changed to “This software is not applicable to your device” |

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

$UserSID = ((gwmi -query "select \* from win32\_useraccount where name = '$($env:USERNAME)' and domain='$($env:USERDOMAIN)'").SID).replace('-','\_')

$sched = ([wmi]"root\ccm\Policy\$UserSID\ActualConfig:CCM\_Scheduler\_ScheduledMessage.ScheduledMessageID='{00000000-0000-0000-0000-000000000026}'");

$sched.Triggers = @('SimpleInterval;Minutes=1;MaxRandomDelayMinutes=0');

$sched.Put()

start-sleep 10

$sched = ([wmi]"root\ccm\Policy\$UserSID\ActualConfig:CCM\_Scheduler\_ScheduledMessage.ScheduledMessageID='{00000000-0000-0000-0000-000000000027}'");

$sched.Triggers = @('SimpleInterval;Minutes=1;MaxRandomDelayMinutes=0');

$sched.Put()

start-sleep 60

$AppName = "Java8"

$app = gwmi -Namespace 'root\CCM\ClientSDK' -Class 'CCM\_Application' | Where-Object { ($\_.Name -eq "$($AppName)") -and ($\_.InstallState -eq "NotInstalled")}

if ($App.ApplicabilityState -eq "NotApplicable") {

Write-host "'$($app.Name)' not applicable." -ForegroundColor Yellow

} else {

Write-host "'$($app.Name)' is applicable."

}

* 1. Simulate Application Deployment

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Applications Management and click Applications |
| 03. Select Acrobat Reader XI and click Simulate Deployment |
| 04. Under Specify general information for this deployment, click Browse (Collection) and select the Collection you want to deploy and select Install for Action. Click Next  Note: You can deploy to a Device as well as Users. In this example, we are using Workstation OU collection |
| 05. Under Confirm the settings for this new Deployment, click Next |
| 06. Under Completion, click Close  Note: Once done, refresh the machine policy on WKS0001, WKS0002 and WKS0004 and monitor the application deployment using the same steps as we have seen under the Monitoring Application Deployment module. |

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

Start-CMApplicationDeploymentSimulation -CollectionName "Workstation OU" -Name "Acrobat Reader XI"

1. Software Update

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002  WKS0004 |
| More information | Prerequisites for Software Updates in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/sum/plan-design/prerequisites-for-software-updates>  Best Practices for Software Updates in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/sum/plan-design/software-updates-best-practices>  Introduction to Software Updates in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/sum/understand/software-updates-introduction>  Prepare for software updates management  <https://docs.microsoft.com/en-us/configmgr/sum/get-started/prepare-for-software-updates-management>  Example scenario to deploy and monitor monthly software updates  <https://docs.microsoft.com/en-us/configmgr/sum/deploy-use/example-scenario-deploy-monitor-monthly-security-updates> |
| Description | In this chapter, we will look at all steps required to manage Microsoft updates in a MECM environment, including deployment, monitoring and reporting compliance. |

* 1. Installation of Windows Server Update Services

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open Server Manager |
| 02. Click Manage and Add Roles and Features |
| 03. Before you begin, click Next |
| 04. Select Role-based or feature-based installation and click Next |
| 05. Select select a server from the server pool and select the server “SRV0002.classroom.intranet” and click Next |
| 06. Under select server roles, select Windows Server Update Services |
| 07. The Add Roles and Features wizard will open to add required requirements, click Add Features and once back to the Select server roles page, click Next |
| 08. Under Select features, click Next |
| 09. Under Windows Server Update Services, click Next |
| 10. Under Select role services, select WSUS Services and SQL Server Connectivity and uncheck WID Connectivity. Click Next  Note: Unselecting WID Connectivity will make the WSUS to be used with a full installation of a SQL Server |
| 11. Under Content location select, type C:\WSUS under Store updates in the following location and click Next |
| 12. Under DB Instance, type localhost and click Check Connection. Click Next |
| 13. Under Confirm installation selections, click Install |
| 14. Once the installation is succeeded, click Close |
| 15. Click the yellow triangle and then Launch Post-Installation tasks |
| 16. Once done, the yellow triangle will disappear |

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

#using SQL Server Installed on the same box

Get-WindowsFeature -Name UpdateServices-Services,UpdateServices-DB | Install-WindowsFeature -IncludeManagementTools

Start-Process -Filepath ('C:\Program Files\Update Services\Tools\WsusUtil.exe') -ArgumentList ('PostInstall CONTENT\_DIR="C:\WSUS" SQL\_INSTANCE\_NAME="srv0002"') -wait

#using Windows Internal Database (WID)

#Get-WindowsFeature -Name UpdateServices | Install-WindowsFeature -IncludeManagementTools

#Start-Process -Filepath ('C:\Program Files\Update Services\Tools\WsusUtil.exe') -ArgumentList ('PostInstall CONTENT\_DIR="C:\WSUS"') -wait

* 1. Installation of the Software Update Point

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Servers and Site System Roles |
| 03. Right click \\SRV0002.classroom.intranet and click Add Site System Roles |
| 04. On Add Site System Roles Wizard, General, click Next |
| 05. Under proxy, click Next |
| 06. Under Specify roles for this server, select Software Update point and click Next |
| 07. Under Specify software update point settings, select WSUS is configured to use ports 8530 and 8531 for client communications (default settings for WSUS on Windows Server 2012) and click Next |
| 08. Under specify proxy and account settings for the software update point, click Next |
| 09. Under Specify synchronization source settings, click Next |
| 10. Under specify on a schedule and run every 1 days. Click Next the synchronization schedule, select Enable synchronization |
| 11. Under select behaviour for software updates that are superseded, select Immediately expire a superseded software update and Run WSUS cleanup wizard. Click Next |
| 12. Under select the software update classifications that you want to synchronize, leave selected Critical updates and Security Updates selected. Click Next |
| 13. Under Select the products that you want to synchronize, unselect all and Click Next  Note: The updated Products and classifications are going to be added after the initial synchronization |
| 14. Under select the language settings that you want to synchronize, leave select only English (Software Update File and Summary Details). Click Next |
| 15. Under confirm the settings, click Next |
| 16. Under You have successfully completed the Add Site System Roles wizard with the following settings click close |
| 17. Click Monitoring |
| 18. Expand System Status and click Component Status |
| 19. Search for WSUS |
| 20. Right Click SMS\_WSUS\_CONTROL\_MANAGER, Show Messages and click All |
| 21. Under Status Messages: Set Viewing Period, click OK |
| 22. Verify the existence of Message ID 1013, 1014, 1015 and 500 |
| 23. Right Click SMS\_WSUS\_CONFIGURATION\_MANAGER, Show Messages and click All |
| 24. Under Status Messages: Set Viewing Period, click OK |
| 25. Verify the existence of Message ID 501, 500 and 4629 |
| 26. Verify the existence of Message ID 6600  Note: If you see the message id 6600, confirm you have selected the correct port during the installation of the Software Update Point |
| 27. Double click any 6600 messages to see its details. Once done, click Ok |
| 28. Right Click SMS\_WSUS\_WSYNC\_MANAGER, Show Messages and click All |
| 24. Under Status Messages: Set Viewing Period, click OK |
| 25. Verify the existence of Message ID 501 and 500 |
| 26. Verify the existence of Message ID 6703  Note: If the logs reveal that there was a problem during the synchronization, it is normally network issues.  Note: During the installation of the Software Update Point, this message is normal, however, it should not occur after the installation |
| 27. You can also review the following logs:   * C:\ConfigMgr\Logs\SUPSetup.log: Records details about the SUP installation. When the SUP installation completes, Installation was successful is written to this log file. * C:\ConfigMgr\Logs\WCM.log: Records details about the SUP configuration and connections to the WSUS server for subscribed update categories, classifications, and languages. * C:\ConfigMgr\Logs\WSUSCtrl.log: Records details about the configuration, database connectivity, and health of the WSUS server for the site. * C:\ConfigMgr\Logs\wsyncmgr.log: Records details about the software updates sync process. * C:\ConfigMgr\Logs\WUSSyncXML.log: Records details about the Inventory Tool for the Microsoft Updates sync process. |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

if ((Get-CMSiteSystemServer -SiteSystemServerName "$servername") -eq $null) { New-CMSiteSystemServer -SiteCode $SiteCode -UseSiteServerAccount -ServerName $servername }

Add-CMSoftwareUpdatePoint -SiteSystemServerName "$ServerName" -ClientConnectionType "Intranet" -SiteCode $SiteCode -WsusiisPort 8530 -WsusiissslPort 8531 -WsusSsl $false

start-sleep 90

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'SMS\_WSUS\_CONTROL\_MANAGER' and stmsg.MessageID = 1013 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_CONTROL\_MANAGER 1013 id's"

break

} else { Start-Sleep 10 }

}

while ($true)

{

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_CONTROL\_MANAGER' and stmsg.MessageID = 1014 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_CONTROL\_MANAGER 1014 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_CONTROL\_MANAGER' and stmsg.MessageID = 1015 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_CONTROL\_MANAGER 1015 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_CONTROL\_MANAGER' and stmsg.MessageID = 500 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_CONTROL\_MANAGER 500 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_CONFIGURATION\_MANAGER' and stmsg.MessageID = 500 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_CONFIGURATION\_MANAGER 500 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_CONFIGURATION\_MANAGER' and stmsg.MessageID = 501 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_CONFIGURATION\_MANAGER 501 id's"

break

} else { Start-Sleep 10 }

}

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_CONFIGURATION\_MANAGER' and stmsg.MessageID = 6600 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "ERROR: Found SMS\_WSUS\_CONFIGURATION\_MANAGER 6600 id's" -ForegroundColor Red

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_CONFIGURATION\_MANAGER' and stmsg.MessageID = 4629 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_CONFIGURATION\_MANAGER 4629 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_SYNC\_MANAGER' and stmsg.MessageID = 500 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_SYNC\_MANAGER 500 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_SYNC\_MANAGER' and stmsg.MessageID = 501 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_SYNC\_MANAGER 501 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_SYNC\_MANAGER' and stmsg.MessageID = 4629 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_WSUS\_SYNC\_MANAGER 4629 id's"

break

} else { Start-Sleep 10 }

}

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_WSUS\_SYNC\_MANAGER' and stmsg.MessageID = 6703 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "ERROR: Found SMS\_WSUS\_SYNC\_MANAGER 6703 id's" -ForegroundColor Red

}

start-sleep 60

$Languages = @("Chinese (Simplified, China)", "French", "German", "Japanese", "Russian")

$Classifications = @()

$Products = @()

#get list of all products

Get-WmiObject -Namespace "Root\SMS\Site\_$SiteCode" SMS\_UpdateCategoryInstance -Filter "(IsSubscribed = 'True') and (CategoryInstance\_UniqueID like 'Product:%')" | foreach {$Products += $\_.LocalizedCategoryInstanceName }

#get list of all classifications

Get-WmiObject -Namespace "Root\SMS\Site\_$SiteCode" SMS\_UpdateCategoryInstance -Filter "(IsSubscribed = 'True') and (CategoryInstance\_UniqueID like 'UpdateClassification:%')" | foreach {$Classifications += $\_.LocalizedCategoryInstanceName }

$schedule = New-CMSchedule -RecurCount 1 -RecurInterval Days

#remove all classifications, languages & products from sync

Set-CMSoftwareUpdatePointComponent -SiteCode $SiteCode -RemoveUpdateClassification $Classifications -RemoveLanguageSummaryDetail $Languages -RemoveLanguageUpdateFile $Languages -RemoveProduct $Products

start-sleep 20

#add Critical Updates only

Set-CMSoftwareUpdatePointComponent -SiteCode $SiteCode -AddUpdateClassification ('Critical Updates','Security Updates')

Set-CMSoftwareUpdatePointComponent -SiteCode $SiteCode -Schedule $Schedule -SynchronizeAction SynchronizeFromMicrosoftUpdate

* 1. Manual Synchronization

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Software Updates, select All Software Updates and click Synchronize Software Updates |
| 03. Once the Run Synchronization question windows appear, click Yes |
| 04. Click Monitoring |
| 05. Click Software Update Point Synchronization Status and wait until the synchronization is completed |
| 06. You can also review the following logs:   * C:\ConfigMgr\Logs\wsyncmgr.log: Records details about the software updates synchronization process. |

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

$SiteCode = "001"

Sync-CMSoftwareUpdate -FullSync $True

start-sleep 10

#6705 = Database

#6704 = Inprogress (WSUS Server)

#6702 = Success

#6702 = Starting

while ($true) {

$return = Get-CMSoftwareUpdateSyncStatus | select LastSyncErrorCode, LastSyncState

Write-Output $return

if ($return.LastSyncState -ne 6702) {start-sleep 10 } else { break }

}

* 1. Changing the List of Products

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Sites |
| 03. Right click 001 – Training Lab and click Configure Site Components -> Software Update Point |
| 04. On Software Update Point Component Properties, change to the Products Tab |
| 05. Under Products, select Windows 8.1, Windows 10 and Windows 10, version 1903 and later and click Ok. |
| 06. Force a manual Synchronization of the Software Updates |
| 07. Once the synchronization is completed, Click Software Library |
| 08. Expand Software Updates and click All Software Updates to see all updates that have been synchronized |

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

$SiteCode = "001"

Set-CMSoftwareUpdatePointComponent -SiteCode $SiteCode -AddProduct @('Windows 8.1', 'Windows 10')

Start-Sleep 20

Sync-CMSoftwareUpdate -FullSync $True

start-sleep 60

#6705 = Database

#6704 = Inprogress (WSUS Server)

#6702 = Success

#6702 = Starting

while ($true) {

$return = Get-CMSoftwareUpdateSyncStatus | select LastSyncErrorCode, LastSyncState

Write-Output $return

if ($return.LastSyncState -ne 6702) {start-sleep 10 } else { break }

}

* 1. Changing Default Client Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select the default client settings and click Properties |
| 04. Under Default Settings, click Software Updates and change the following:   * Software update scan schedule: Occurs every 1 day * Software deployment re-evaluation: Occurs every 3 days * When any software update deployment is reached, install all other software update deployment with deadline coming within a specific period of time: Yes * Period of time for which all pending deployments with deadlines in this tie will also be installed: 7 days   Once done, click Ok. |

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

$ScanSchedule = New-CMSchedule -RecurCount 1 -RecurInterval Days

$DeploymentReEvaluation = New-CMSchedule -RecurCount 3 -RecurInterval Days

$ClientSettingsName = "Default Client Agent Settings"

Set-CMClientSetting -Name "$ClientSettingsName" -SoftwareUpdate -BatchingTimeout 7 -DeploymentEvaluationSchedule $DeploymentReEvaluation -Enable $True -EnforceMandatory $True -ScanSchedule $ScanSchedule -TimeUnit Days

* 1. MECM Client Scan

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Once the User Policy Retrieval & Evaluation cycle message appears, click Ok twice.  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 04. Open Control Panel and then click Configuration Manager |
| 05. Change to the Actions Tab, select Software Updates Scan Cycle and click Run now  Note: Software Update Scan Cycle action did not exist before as the computer had not received the latest policy |
| 06. Under Software Updates Scan Cycle click Ok |
| 07. On the client, you can also review the following client logs:   * C:\Windows\ccm\Logs\ScanAgent.log: Records details about scan requests for software updates, the WSUS location, and related actions. * C:\Windows\ccm\Logs\UpdatesDeployment.log: Records details about deployments on the client, including software update activation, evaluation, and enforcement. Verbose logging shows additional information about the interaction with the client user interface. * C:\Windows\ccm\Logs\UpdatesHandler.log: Records details about software update compliance scanning and about the download and installation of software updates on the client. * C:\Windows\ccm\Logs\UpdatesStore.log: Records details about compliance status for the software updates that were assessed during the compliance scan cycle. * C:\Windows\ccm\Logs\WUAHandler.log: Records details about the Windows Update Agent on the client when it searches for software updates. * C:\Windows\WindowsUpdate.log: Records details about when the Windows Update Agent connects to the WSUS server and retrieves the software updates for compliance assessment and whether there are updates to the agent components.   Note: Repeat the process for the WKS0002 and WKS0004 machines  Note: WindowsUpdate.log is no longer used in Windows 10 as the logs are now generated using ETW (Event Tracing for Windows). For more information, refer to <https://blogs.technet.microsoft.com/charlesa_us/2015/08/06/windows-10-windowsupdate-log-and-how-to-view-it-with-powershell-or-tracefmt-exe/> |

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

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000113}")

Start-Sleep 60

* 1. Reporting Compliance via Console for All Updates

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Software Updates, select All Software Updates and click Run Summarization |
| 03. Click Ok once the Configuration Manager information screen appears |
| 04. Search for Update for Update for Windows 8.1 for x64-based Systems (KB3173424) and see the compliance via Statistics |

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

Invoke-CMSoftwareUpdateSummarization

Start-Sleep 20

Get-CMSoftwareUpdate -Name "\*KB3173424\*" -fast | select LocalizedDisplayName, NumMissing, NumNotApplicable, NumPresent, NumTotal, NumUnknown, PercentCompliant

* 1. Deploying a Patch

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Software Updates and click All Software Updates |
| 03. Search for Update for Windows 8.1 for x64-based Systems (KB3173424) |
| 04. Select the Update and click Create Software Update Group |
| 05. Under Create Software Update Group type Windows 8x Critical Updates as Name and click Ok |
| 06. Select Software Update Groups |
| 07. Select Windows 8x Critical Updates and click Show Members |
| 08. Confirm that all updates have been added to the list. |
| 09. Select Software Update Group |
| 10. Select Windows 8x Critical Updates and Click Deploy |
| 11. Under Specify general information for this deployment, select Windows 8 Workstations collection and click Next |
| 12. Under Specify deployment settings for this deployment, select:   * Type of Deployment: Available * Detail level: All messages   click Next |
| 13. Under Configure schedule details for this deployment, click Next |
| 14. Under Specify the user experience for this deployment, click Next |
| 15. Under Specify Software update alert options for this deployment, click Next |
| 16. Under specify download settings for this deployment, select if software updates are not available on preferred distribution point or remote distribution point, download content from Microsoft Updates and click Next  Note: Selecting if software updates are not available on preferred distribution point or remote distribution point download content from Microsoft Updates allows the download of Microsoft updatesfrom Microsoft in case a machine cannot download from a Distribution Point, is useful for mobile users or in the case of a distribution point failure. Use carefully as this will generate extra internet network traffic and this will need to be allowed by your company firewall and may affect 3g/4g network data usage and encur costs |
| 17. Under Specify the package to use the following:   * Create a new deployment package: checked * Name: Windows 8x Critical Updates * Package source: \\SRV0001\WSUSDownloadContent\W8xCriticalUpdates   Once done, click Next |
| 18. Under Specify the distribution points or distribution points groups to host the content, click Add Distribution Point Group, under Add-Distribution Point Groups, select Training Lab, and click Ok. Once back, click Next |
| 19. Under specify the source location for software updates that you will download, select Download software updates from the internet and click Next |
| 20. Under Specify the language of the updates, click Next |
| 21. Under Confirm the settings, click Next |
| 22. Under The Deploy Software Updates wizard completed successfully, click Close |
| 23. Navigate to \\SRV0001\WSUSDownloadContent\W8xCriticalUpdates and confirm that the windows8.1-kb3173424-x64.cab has been downloaded  Note: The downloaded file will be under a subfolder of the W8xCriticalUpdates  Note: Once done, confirm that the content has been distributed to the distribution point using the Content Status under Monitoring->Distribution Status |

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

$KB = "KB3173424"

$SupGroup = "Windows 8x Critical Updates"

$DepGroup = "Windows 8x Critical Updates"

$DepPath = "\\SRV0001\WSUSDownloadContent\W8xCriticalUpdates"

$ColName = "Windows 8 Workstations"

New-CMSoftwareUpdateGroup -Name "$SupGroup"

Start-Sleep 5

$UpdateList = Get-CMSoftwareUpdate -Name "\*$($KB)\*" -fast | Where-Object {$\_.LocalizedDisplayName -like "\*x64\*"}

Set-CMSoftwareUpdateGroup -Name "$SupGroup" -AddSoftwareUpdate $UpdateList

Start-Sleep 5

(Get-CMSoftwareUpdate -UpdateGroupName "$SupGroup" -fast).LocalizedDisplayName

Start-Sleep 5

New-CMSoftwareUpdateDeploymentPackage -Name "$DepGroup" -Path "$DepPath"

start-sleep 10

Save-CMSoftwareUpdate -DeploymentPackageName "$DepGroup" -SoftwareUpdateGroupName "$SupGroup"

Start-Sleep 60

Start-CMContentDistribution -DeploymentPackageName "$DepGroup" -DistributionPointGroupName "Training Lab"

Start-Sleep 10

New-CMSoftwareUpdateDeployment -CollectionName "$ColName" -SoftwareUpdateGroupName "$SupGroup" -AcceptEula -DeploymentType Available -DownloadFromMicrosoftUpdate $True -VerbosityLevel AllMessages

* 1. Installing a Patch

|  |
| --- |
| Perform this task on the wks0004 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Under Machine Policy Retrieval & Evaluation Cycle click Ok  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 04. Open Software Center and click Updates |
| 05. Select the Update and click Install |
| 06. Once done, the button will be disabled and the label will change to Uninstall |

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

#update policies

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

$Updates = Get-WmiObject -Namespace "root\ccm\clientSDK" -Class CCM\_SoftwareUpdate | Where-Object { $\_.ComplianceState -eq 0 -and $\_.EvaluationState -eq 0}

Invoke-WmiMethod -Class CCM\_SoftwareUpdatesManager -Name InstallUpdates -ArgumentList (,$Updates) -Namespace root\ccm\clientsdk

* 1. Reporting Compliance via Console for an Update Group

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Software Updates and click Software Update Groups. |
| 03. Select the Update Group ad click Run Summarization  Note: The Summarization, by default, occurs once per hour. |
| 04. Click Ok once the Configuration Manager information screen appears |
| 05. After the summarization, under Statistics, review the statistics. |

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

Invoke-CMSoftwareUpdateSummarization

Start-Sleep 20

Get-CMSoftwareUpdate -IsDeployed $true | select LocalizedDisplayName, NumMissing, NumNotApplicable, NumPresent, NumTotal, NumUnknown, PercentCompliant

* 1. Reporting Compliance via Reports

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Reporting, Reports and click Software Updates – A Compliance |
| 03. Select Compliance 5 - Specific computer |
| 04. Under Compliance 5 - Specific computer report, fill up the parameters (you may use the Values link) as following:   * Device Name: WKS0004.CLASSROOM * Vendor: Microsoft * Update Class: Critical Updates   click View report  Note: You can drill down to a more specific report using the links inside the reports  Note: You can order by Approved to easily find approved updates information |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

#Open Report

$dict = @{"Device Name"="WKS0004.CLASSROOM"; "Vendor"="Microsoft"; "Update Class"="Critical Updates"; }

Invoke-CMReport -ReportPath "Software Updates - A Compliance/Compliance 5 - Specific computer" -SiteCode "$SiteCode" -SrsServerName "$servername" -ReportParameter $dict

1. Windows 10 Express Updates

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002 |
| More information | Manage Express installation files for Windows 10 updates  <https://docs.microsoft.com/en-us/configmgr/sum/deploy-use/manage-express-installation-files-for-windows-10-updates> |
| Description | In this chapter, we will look at all steps required to enable Express Updates support in a MECM environment, including some client Troubleshooting and logs. |

* 1. Preparing the Environment

Before Starting this Module, you will need to:

1. Deploy the latest Software Update patch available for Windows 10 to the Windows 10 Workstations collection
   * Search for KB4530684 (December/2019 update, change for the most up to date update available)
   * Create a Software Update Group named Windows 10x Security Updates
   * Download the Windows 10x Security Updates Software Update Group to a new Deployment Package named Windows 10x Security Updates and save it to \\SRV0001\WSUSDownloadContent\W10xSecurityUpdates
   * Deploy as available to Windows 10 workstations collection

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

$KB = "KB4530684" #update from december/2019

$SupGroup = "Windows 10x Security Updates"

$DepGroup = "Windows 10x Security Updates"

$DepPath = "\\SRV0001\WSUSDownloadContent\W10xSecurityUpdates"

$ColName = "Windows 10 Workstations"

New-CMSoftwareUpdateGroup -Name "$SupGroup"

Start-Sleep 5

$UpdateList = Get-CMSoftwareUpdate -Name "\*$($KB)\*" -fast | Where-Object {$\_.LocalizedDisplayName -like "\*x64\*"}

Set-CMSoftwareUpdateGroup -Name "$SupGroup" -AddSoftwareUpdate $UpdateList

Start-Sleep 5

(Get-CMSoftwareUpdate -UpdateGroupName "$SupGroup" -fast).LocalizedDisplayName

Start-Sleep 5

New-CMSoftwareUpdateDeploymentPackage -Name "$DepGroup" -Path "$DepPath"

start-sleep 10

Save-CMSoftwareUpdate -DeploymentPackageName "$DepGroup" -SoftwareUpdateGroupName "$SupGroup"

Start-Sleep 60

Start-CMContentDistribution -DeploymentPackageName "$DepGroup" -DistributionPointGroupName "Training Lab"

Start-Sleep 10

New-CMSoftwareUpdateDeployment -CollectionName "$ColName" -SoftwareUpdateGroupName "$SupGroup" -AcceptEula -DeploymentType Available -DownloadFromMicrosoftUpdate $True -VerbosityLevel AllMessages

* 1. Enabling Express Updates support in MECM

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Sites |
| 03. Right click 001 – Training Lab and click Configure Site Components -> Software Update Point |
| 04. On Software Update Point Component Properties, change to the Update Files Tab |
| 05. Under Update FIles, select Download both full files for all approved updates and express installation files for Windows 10. Click OK. |
| 06. Click Client Settings |
| 07. Select Client Settings and click Create Custom Client Device Settings |
| 08. Type Windows 10 Express Updates on Name and select Software Updates under Select the custom settings to be enforced on client devices |
| 09. Click Software Updates |
| 10. Select Yes for Enable Installation of Express installation files on clients and configure the Port the MECM Client will use to communicate with the Distribution Point. Click Ok |
| 11. Select the Windows 10 Express Updates and click Deploy |
| 12. Under select collection, click Windows 10 Workstations and click Ok |
| 13. Select Deployments and confirm that the Client Settings has been deployed to the collection |

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

$SiteCode = "001"

set-CMSoftwareUpdatePointComponent -ContentFileOption ExpressForWindows10Only -SiteCode $SiteCode

$ClientSettingsName = "Windows 10 Express Updates"

New-CMClientSetting -Name "$ClientSettingsName" -Type Device

Set-CMClientSettingSoftwareUpdate -Name "$ClientSettingsName" -Enable $true

$CS = (gwmi -namespace root\sms\site\_001 -query "select \* from SMS\_ClientSettings where Name = '$ClientSettingsName'")

$CS.Get()

$ac = $CS.AgentConfigurations

$agent = $ac[0]

$agent.EnableExpressUpdates = $true

$agent.ExpressUpdatesPort = 8005

$CS.AgentConfigurations = $ac

$CS.Put()

Start-CMClientSettingDeployment -ClientSettingName "$ClientSettingsName" -CollectionName "Windows 10 Workstations"

* 1. Monitoring Windows 10 Express Updates via Client Logs

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 03. Once the User Policy Retrieval & Evaluation cycle message appears, click Ok twice.  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 04. Open Software Center and click Updates |
| 05. Select the Update and click Install |
| 06. Once done, the button will be disabled and the label will change to Uninstall |
| 07. On the client, you can also review the following client logs:   * C:\Windows\ccm\Logs\DeltaDownload.log: Records details about the Windows 10 express updates agent setting, including enabled/disabled state and firewall port. |

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

Start-Process -Filepath ("c:\windows\cmtrace.exe") -ArgumentList ("c:\Windows\ccm\Logs\DeltaDownload.log")

#Client

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000113}")

Start-Sleep 60

$Updates = Get-WmiObject -Namespace "root\ccm\clientSDK" -Class CCM\_SoftwareUpdate | Where-Object { $\_.ComplianceState -eq 0 -and $\_.EvaluationState -eq 0}

Invoke-WmiMethod -Class CCM\_SoftwareUpdatesManager -Name InstallUpdates -ArgumentList (,$Updates) -Namespace root\ccm\clientsdk

1. 3rd Party Updates with PatchMyPc

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002 |
| More information | Enhanced HTTP  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/hierarchy/enhanced-http>  Enable third-party updates  <https://docs.microsoft.com/en-us/configmgr/sum/deploy-use/third-party-software-updates>  WSUS no longer issues self-signed certificates  <https://blogs.technet.microsoft.com/wsus/2013/08/15/wsus-no-longer-issues-self-signed-certificates/> |
| Description | In this chapter, we will look at keeping 3rd party software up to date with PatchMyPc |

* 1. Enable Configuration Manager-generated certificates for HTTP site systems

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Sites |
| 03. Right click 001 – Training Lab and click Properties |
| 04. On Training Lab Properties, change to Communication Security Tab |
| 05. On Communication Security Tab, select Use Configuration Manager-generated certificates for HTTP site systems and click Ok |

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

Set-CMSite -ClientComputerCommunicationType HttpsOrHttp -SiteCode $SiteCode

$Component = gwmi -Namespace "root\SMS\site\_$($SiteCode)" -query "select \* from SMS\_SCI\_Component where FileType = 2 and ItemName = 'SMS\_SITE\_COMPONENT\_MANAGER|SMS Site Server' and ItemType='Component' and SiteCode='$($SiteCode)'"

$props = $component.Props

$prop = $props | where {$\_.PropertyName -eq 'IISSSLState'}

$prop.Value = 1248

$component.Props = $props

$component.Put() | Out-Null

* 1. Enabling 3rd Paty Updates

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Sites |
| 03. Right click 001 – Training Lab and click Configure Site Components -> Software Update Point |
| 04. On Software Update Point Component Properties, change to the Third Party Updates Tab |
| 05. Under Third Party Updates, select Enable third-party software updates, select Configuration Manager manages the certificate under WSUS signing certificate configuration and click Click OK. |
| 06. Click Client Settings |
| 07. Click Software Library. |
| 08. Expand Software Updates, select All Software Updates and click Synchronize Software Updates |
| 09. Once the Run Synchronization question windows appear, click Yes |
| 10. Click Monitoring |
| 11. Click Software Update Point Synchronization Status and wait until the synchronization is completed |
| 12. Click Administration. |
| 13. Expand Security and Certificates |
| 14. Check the existance of a Third-Party WSUS Signing certificate type |
| 15. Select the default client settings and click Properties |
| 16. Under Default Settings, click Software Updates and change the following:   * Enable third party software updates: Yes   Once done, click Ok. |
| 17. Select the Windows 10 Express Updates and click Properties |
| 18. Under Default Settings, click Software Updates and change the following:   * Enable third party software updates: Yes   Once done, click Ok. |

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

Set-CMSoftwareUpdatePointComponent -SiteCode $SiteCode -EnableThirdPartyUpdates $true

$Component = gwmi -Namespace "root\SMS\site\_$($SiteCode)" -query "select \* from SMS\_SCI\_Component where FileType = 2 and ItemName = 'SMS\_WSUS\_SYNC\_MANAGER|SMS Site Server' and ItemType='Component' and SiteCode='$($SiteCode)'"

$props = $component.Props

$prop = $props | where {$\_.PropertyName -eq 'EnableThirdPartyUpdates'}

$prop.Value = 3

$component.Props = $props

$component.Put() | Out-Null

Sync-CMSoftwareUpdate -FullSync $True

start-sleep 30

#6705 = Database

#6704 = Inprogress (WSUS Server)

#6702 = Success

#6702 = Starting

while ($true) {

$return = Get-CMSoftwareUpdateSyncStatus | select LastSyncErrorCode, LastSyncState

Write-Output $return

if ($return.LastSyncState -ne 6702) {start-sleep 10 } else { break }

}

if (-not (Get-CMCertificate | where {($\_.IssuedTo -eq 'CN=WSUS Publishers Self-signed') -and ($\_.Type -eq 6)})) {

Write-host 'Certificate not generated yet'

start-sleep 10

} else {

write-host 'WSUS Certificate cenerated'

}

Set-CMClientSettingSoftwareUpdate -DefaultSetting -EnableThirdPartyUpdates $true

Set-CMClientSettingSoftwareUpdate -Name 'Windows 10 Express Updates' -EnableThirdPartyUpdates $true

start-sleep 20

* 1. Installing Patch My PC Publishing Service

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Execute PatchMyPC-Publishing-Service.msi from \\srv0001\Trainingfiles\Source\PatchMyPc |
| 02. Under Welcome to the Patch My PC Publishing Service Setup Wizard, click Next |
| 03. Under End-User License Agreement, select I accept the terms in the License Agreement and click Next |
| 04. Under Select Installation Folder, click Next |
| 05. Under Ready to Install, click Install |
| 06. Once the installation is completed, click Finish |

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

Start-Process -Filepath ('msiexec') -ArgumentList ('/i "\\srv0001\TrainingFiles\Source\PatchMyPc\PatchMyPC-Publishing-Service.msi" /qb') -wait -NoNewWindow

* 1. Configuring Patch My PC Publishing Service

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Click Start, Patch My PC Publishing Service and click Path My PC Publishing Service |
| 02. On the General Tab, under Catalog Information, select Use Trial Mode and on the Switch To Trail Catalog Mode? Message, click Yes |
| 03. Change to Updates Tab |
| 04. Under Updates tab, expand Mozilla and select Mozilla Firefox (x64 en-US) (Full Content) and click Apply |
| 05. Change to Sync Schedule tab, click Run Publishing Service Sync and then click Ok on the Run Now Successful message. |
| 06. Back to the Patch My PC – Publishing Settings, Click Ok  Note: If the Please, review your settings… message appears, click OK |
| 07. You can also review the following logs:   * C:\Program Files\Patch My Pc\Patch My PC Publishing Service\PatchMyPc.Log: Records detailed information about the Patch My PC Publishing Service. |

* 1. Enable Patch My PC Products

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Software Updates, select All Software Updates and click Synchronize Software Updates |
| 03. Once the Run Synchronization question windows appear, click Yes |
| 04. Click Monitoring |
| 05. Click Software Update Point Synchronization Status and wait until the synchronization is completed |
| 06. Click Administration. |
| 07. Expand Site Configuration and click Sites |
| 09. Right click 001 – Training Lab and click Configure Site Components -> Software Update Point |
| 09. On Software Update Point Component Properties, change to the Products Tab |
| 10. Under Products, select Patch My PC and click Ok. |
| 11. Click Software Library. |
| 12. Expand Software Updates, select All Software Updates and click Synchronize Software Updates |
| 13. Once the Run Synchronization question windows appear, click Yes |
| 14. Click Monitoring |
| 15. Click Software Update Point Synchronization Status and wait until the synchronization is completed |
| 16. Click Software Library. |
| 17. Expand Software Updates and click All Software Updates |
| 18. Search for Firefox  Note: Once the update is available, it can be deployed using the same steps seen on module Software Updates |

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

$SiteCode = "001"

Sync-CMSoftwareUpdate -FullSync $True

start-sleep 30

#6705 = Database

#6704 = Inprogress (WSUS Server)

#6702 = Success

#6702 = Starting

while ($true) {

$return = Get-CMSoftwareUpdateSyncStatus | select LastSyncErrorCode, LastSyncState

Write-Output $return

if ($return.LastSyncState -ne 6702) {start-sleep 10 } else { break }

}

Set-CMSoftwareUpdatePointComponent -SiteCode $SiteCode -AddCompany @('Patch My PC')

Start-Sleep 20

Sync-CMSoftwareUpdate -FullSync $True

start-sleep 30

#6705 = Database

#6704 = Inprogress (WSUS Server)

#6702 = Success

#6702 = Starting

while ($true) {

$return = Get-CMSoftwareUpdateSyncStatus | select LastSyncErrorCode, LastSyncState

Write-Output $return

if ($return.LastSyncState -ne 6702) {start-sleep 10 } else { break }

}

Get-CMSoftwareUpdate -Name '\*Firefox\*' -Fast

1. Endpoint Protection

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002  WKS0004 |
| More information | Planning for Endpoint Protection in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/protect/plan-design/planning-for-endpoint-protection>  Configuring Endpoint Protection in Configuration Manager  <https://docs.microsoft.com/en-gb/configmgr/protect/deploy-use/endpoint-protection-configure>  Using Configuration Manager Software Updates to Deliver Definition Updates  <https://docs.microsoft.com/en-us/configmgr/protect/deploy-use/endpoint-definitions-configmgr> |
| Description | In this chapter, we will look at all steps required to manage the Microsoft anti-virus (Endpoint Protection) as well as Windows Defender (when running on Windows 10) |

* 1. Configuring Software Update Point

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Servers and Sites |
| 03. Select 001 – Training Lab, Configure Site Components and click Software Update Point |
| 04. Under Software Update Point Component Properties, click Classifications Tab |
| 05. Under Classifications Tab select Definition Updates. Change to the Products Tab |
| 06. Under Products Tab, select:   * System Center Endpoint Protection under Forefront products (used for Windows 8.1 and earlier) * Windows Defender under Windows (used for Windows 10 and later).   click Ok  Note: Once it is done, force a manual Synchronization of the Software Update. This is required so the new updates are going to be synchronized. |

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

Set-CMSoftwareUpdatePointComponent -SiteCode $SiteCode -ImmediatelyExpireSupersedence $True -AddUpdateClassification "Definition Updates" -AddProduct @('System Center Endpoint Protection', 'Windows Defender')

Start-Sleep 20

Sync-CMSoftwareUpdate -FullSync $True

Start-Sleep 60

#6705 = Database

#6704 = Inprogress (WSUS Server)

#6702 = Success

#6702 = Starting

while ($true) {

$return = Get-CMSoftwareUpdateSyncStatus | select LastSyncErrorCode, LastSyncState

Write-Output $return

if ($return.LastSyncState -ne 6702) {start-sleep 10 } else { break }

}

* 1. Installing Endpoint Protection Point

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Servers and Site System Roles |
| 03. Right click \\SRV0002.classroom.intranet and click Add Site System Roles |
| 04. On Add Site System Roles Wizard, General, click Next |
| 05. Under proxy, click Next |
| 06. Under Specify roles for this server, select Endpoint Protection point. |
| 07. Under Configuration Manager warning message, click Ok. Once back, click Next |
| 08. Under Endpoint Protection License Terms, click by checking this box, I acknowledge that I accept the License Terms and Privacy Statement and click Next |
| 09. Under Specify Microsoft Active Protection Service membership type, select the level of participation that you want and click Next  Note: More information about the Microsoft Active Protection membership type can be found <http://go.microsoft.com/fwlink/?LinkID=626987> |
| 10. Under confirm the settings, click Next |
| 11. Under You have successfully completed the Add Site System Roles wizard with the following settings click close |
| 12. Click Monitoring |
| 13. Expand System Status and click Component Status |
| 14. Search for ENDPOINT |
| 15. Right Click SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER, Show Messages and click All |
| 16. Under Status Messages: Set Viewing Period, click OK |
| 17. Verify the existence of Message ID 1013, 1014, 1015 and 500 |
| 18. Once installed, the Endpoint Protection client is also installed with Real-time protection disabled.  Note: Depending on the server performance, it may take a while for the Latest update definition be applied. |
| 19. You can also review the following logs:   * C:\ConfigMgr\Logs\EPSetup.log: Provides information about the installation of the Endpoint Protection site system role. * C:\ConfigMgr\Logs\EPMgr.log: Monitors the status of the Endpoint Protection site system role. * C:\ConfigMgr\Logs\EPCtrlMgr.log: Records details about the synchronization of malware threat information from the Endpoint Protection role server into the Configuration Manager database. |

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

if ((Get-CMSiteSystemServer -SiteSystemServerName "$servername") -eq $null) { New-CMSiteSystemServer -SiteCode $SiteCode -UseSiteServerAccount -ServerName $servername }

Add-CMEndpointProtectionPoint -ProtectionService BasicMembership -SiteSystemServerName "$ServerName" -SiteCode $SiteCode

start-sleep 90

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER' and stmsg.MessageID = 1013 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER 1013 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER' and stmsg.MessageID = 1014 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER 1014 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER' and stmsg.MessageID = 1015 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER 1015 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER' and stmsg.MessageID = 500 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found SMS\_ENDPOINT\_PROTECTION\_CONTROL\_MANAGER 500 id's"

break

} else { Start-Sleep 10 }

}

* 1. Automating the definition updates delivery

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Software Updates, and click Automatic Deployment Rules |
| 03. Select Deployment Rules and click Create Automatic Deployment Rule |
| 04. Under Specify the settings for this automatic deployment rule use the following information:   * Name: ADR-Definition Updates * Template: Definition Updates * Collection: All Desktop and Server Clients   Click Next |
| 05. Under Specify the settings for this Automatic Deployment Rule select Detail level as All Messages and click Next |
| 06. Under Select the property filters and search criteria, click Next |
| 07. Under Specify the recurring schedule for this rule, click Next |
| 08. Under Configure schedule details for this deployment select Time based on Client local time and click Next |
| 09. Under Specify the user experience for this deployment, click Next |
| 10. Under Specify software update alert options for this deployment, click Next |
| 11. Under Specify the software updates download behaviour for clients on slow boundaries, click Next |
| 12. Under Select Deployment Package for this automatic deployment rule, use the following information:   * Create a new deployment package: selected * Name: ADR-Definition Updates * Package Source: \\SRV0001\wsusDownloadContent\ADR-DefUpdates   Click Next |
| 13. Under Specify the distribution points or distribution points groups to host the content, click Add Distribution Point Group, under Add-Distribution Point Groups, select Training Lab, and click Ok. Once back, click Next |
| 14. Under Specify download location for this Automatic Deployment Rule, click Next |
| 15. Under Select Language for which software update files are downloaded, click Next |
| 16. Under Confirm the settings, click Next |
| 17. Under The Create Automatic Deployment Rule Wizard completed successfully, click Close |
| 18. Select the ADR-Definition Updates and click Run Now |
| 19. You can also review the following log:   * C:\ConfigMgr\Logs\ruleengine.log: Records details about automatic deployment rules for the identification, content download, and software update group and deployment creation. |

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

$ADRName = "ADR-Definition Updates"

$pkg = New-CMSoftwareUpdateDeploymentPackage -Name "$ADRName" -Path \\SRV0001\wsusDownloadContent\ADR-DefUpdates

start-sleep 5

Start-CMContentDistribution -DeploymentPackageId $pkg.PackageID -DistributionPointGroupName "Training Lab"

start-sleep 5

New-CMSoftwareUpdateAutoDeploymentRule -CollectionName "All Desktop and Server Clients" -Name "$ADRName" -AddToExistingSoftwareUpdateGroup $True -AllowSoftwareInstallationOutsideMaintenanceWindow $True -AvailableTime 1 -AvailableTimeUnit Hours -DeadlineImmediately $True -DeploymentPackageName "$ADRName" -DeployWithoutLicense $True -DownloadFromInternet $True -DownloadFromMicrosoftUpdate $True -EnabledAfterCreate $True -GenerateSuccessAlert $True -RunType RunTheRuleAfterAnySoftwareUpdatePointSynchronization -UserNotification HideAll -UseUtc $False -VerboseLevel AllMessages -UpdateClassification @('Critical Updates', 'Definition Updates') -Language @("English") -LanguageSelection @("English") -Product @('System Center Endpoint Protection', 'Windows Defender')

start-sleep 5

Invoke-CMSoftwareUpdateAutoDeploymentRule -Name "$ADRName"

* 1. Changing Default Client Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select the default client settings and click Properties |
| 04. Under Default Settings, click Endpoint Protection. Confirm that the following settings are configure:   * Manage Endpoint Protection client on client computers: Set to Yes * Install Endpoint Protection client on client computers: Set to Yes * Disable alternative sources (such as Microsoft Windows Update, Microsoft Windows Server Update Services, or UNC share) for the initial definition updates on client computers: Set to No   Click Ok  Note: Force the machine policy refresh on WKS0001, WKS0002 and WKS0004 for the Endpoint Protection client installation start and wait the definition updates before testing the malware activity |
| 05. You can also review the following client log:   * C:\Windows\CCM\Logs\EndpointProtectionAgent.log: Records details about the installation of the Endpoint Protection client and the application of antimalware policy to that client. |

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

#on the server

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

$ClientSettingsName = "Default Client Agent Settings"

Set-CMClientSetting -EndpointProtection -Name "$ClientSettingsName" -Enable $True -InstallEndpointProtectionClient $true -DisableFirstSignatureUpdate $False

#on client machines

#update policies

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

* 1. Testing Malware activity

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Navigate to \\srv0001\TrainingFiles\Source\Eicar and open eicar test file.txt file |
| 02. The Anti-virus will detect this a malware file and will deny access. Once it happens, click Ok to the warning message |
| 03. Open Windows Defender and click History |
| 04. Select All detected items and click View details |

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

#open notepad

Start-Process -Filepath ('notepad.exe') -ArgumentList ("\\srv0001\TrainingFiles\Source\Eicar\eicar test file.txt") -wait

#get info about malware

Get-MpThreat

* 1. Monitoring Malware Activity via Console

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Click Device Collections |
| 03. Select Windows 10 Workstation collection and click Properties |
| 04. Under Windows 10 Workstation Properties, change to the Alerts tab |
| 05. Under Alerts tab, click View this collection in the Endpoint Protection dashboard and click Ok |
| 06. Click Monitoring |
| 07. Expand Security, Endpoint Protection Status and click Endpoint Protection Status |
| 08. Confirm that there is a malware activity identified. Click Malware Detected  Note: Running Summarization may be needed |
| 09. The list of identified malwares appears. Select it on the list and click Files Modified  Note: Running Summarization may be needed |
| 10. Under Files Modified is possible to review list of files that the Endpoint Protection identified |

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

$SiteCode = "001"

$dkserver = "SRV0002.classroom.intranet"

$Collection = Get-CMDeviceCollection -Name "Windows 10 Workstations"

Invoke-WmiMethod -path "\\.\root\sms\site\_$($SiteCode):SMS\_Collection.CollectionID='$($Collection.CollectionID)'" -Name UpdateVisibilityInEPDashBoard -ArgumentList @(1)

start-sleep 5

Invoke-CMEndpointProtectionSummarization

start-sleep 10

Get-CMDetectedMalware -CollectionName "$($Collection.Name)"

* 1. Monitoring Malware Activity via Reports

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Expand Reporting, Reports and click Endpoint Protection |
| 03. Select Antimalware activity report and click Run |
| 04. Under Antimalware activity report, fill up the parameters (you may use the Values link) and click View report  Note: You can drill down to a more specific report using the links inside the reports |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

#Open Report

$dict = @{"Collection Name"="Windows 10 Workstations"}

Invoke-CMReport -ReportPath "Endpoint Protection/Antimalware activity report" -SiteCode "$SiteCode" -SrsServerName "$servername" -ReportParameter $dict

1. Compliance Settings

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002  WKS0001  WKS0002 |
| More information | Get started with compliance settings in Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/compliance/get-started/get-started-with-compliance-settings>  Common tasks for managing compliance on devices with the Configuration Manager client  <https://docs.microsoft.com/en-us/configmgr/compliance/plan-design/common-tasks-for-managing-compliance-on-devices-with-the-client> |
| Description | In this chapter, we will look on how to use Compliance Settings to monitor and auto-remediate non-compliance settings on managed devices |

* 1. Changing Default Client Settings

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Client Settings |
| 03. Select the default client settings and click Properties |
| 04. Under Default Settings, click Compliance Settings and then, set the schedule evaluation to 1 day. Click Ok |

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

$Schedule = New-CMSchedule -RecurCount 1 -RecurInterval Days

$ClientSettingsName = "Default Client Agent Settings"

Set-CMClientSetting -Compliance -Name "$ClientSettingsName" -EnableComplianceEvaluation $true -Schedule $Schedule

* 1. Registry Configuration Items with Auto-remediation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Asset and Compliance. |
| 02. Expand Compliance Settings and click Configuration Items |
| 03. Select Compliance Settings and click Create Configuration Item |
| 04. Under Specify general information about this configuration item type Internet Explorer Default Start Page under name and under Specify the type of configuration item that you want to create select Windows Desktops and Servers (custom) that is below Settings for devices managed with the Configuration Manager client. Click Next |
| 05. Under specify the version of Windows that will assess this configuration for compliance, click Next |
| 06. Under Specify settings for this operating system click New |
| 07. Use the following information:   * Name: IE Start Page * Settings type: Registry value * Data Type: String * Hive Name: HKEY\_CURRENT\_USER * Key Name: Software\Microsoft\Internet Explorer\Main * Value Name: Start Page * Select This registry is associated with a 64-bit application   Note: As this setting validates a 64-bit application and not a 32-bit application in a 64-bit machine (not under Wow6432 registry), selecting the “This registry is associated with a 64-bit application” is required)  Note: This setting has been configured as a User setting and will be evaluated only when user is logged on |
| 08. Change to Compliance Rules tab and click New |
| 09. On Specify rules to define compliance conditions for this setting, type Start Page Equals http://www.thedesktopteam.com in the Name, on the following values: type http://www.thedesktopteam.com, select Remediate noncompliant rules when supported and Report noncompliance if this setting instance is not found and select Critical with Event on Noncompliance severity for reports. click Ok twice |
| 10. Once back to the specify settings for this operating system, click Next |
| 11. Under specify compliance rules for this operating systems, confirm that 1 rule has been already created and click Next |
| 12. Under the wizard will create an operating system configuration item with the following settings, click Next |
| 13. Under The create Configuration Item Wizard completed successfully, click Close |

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

New-CMConfigurationItem -Name "Internet Explorer Default Start Page" -CreationType WindowsOS | Add-CMComplianceSettingRegistryKeyValue -ValueRule -DataType String -Name "IE Start Page" -Hive CurrentUser -KeyName "Software\Microsoft\Internet Explorer\Main" -ValueName "Start Page" -Is64Bit -RuleName "Start Page Equals http://www.thedesktopteam.com" -ExpressionOperator IsEquals -ExpectedValue "http://www.thedesktopteam.com" -NoncomplianceSeverity Critical -Remediate -ReportNoncompliance

* 1. Application Settings Configuration Items without Auto-remediation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Asset and Compliance. |
| 02. Expand Compliance Settings and click Configuration Items |
| 03. Select Compliance Settings and click Create Configuration Item |
| 04. Under Specify general information about this configuration item type Validate Mozilla Firefox Version under name and under Specify the type of configuration item that you want to create select Windows Desktops and Servers (custom) and This configuration item contains application settings that are below Settings for devices managed with the Configuration Manager client. Click Next |
| 05. Under specify how this application is detected on client devices select Always assume application is installed and click Next |
| 06. Under Specify settings for this operating system click New |
| 07. Use the following information:   * Name: Mozila Firefox CurrentVersion * Settings type: Registry value * Data Type: String * Hive Name: HKEY\_LOCAL\_MACHINE * Key Name: Software\Mozilla\Mozilla Firefox * Value Name: CurrentVersion * Select This registry is associated with a 64-bit application   Note: As this setting validates a 64-bit application and not a 32-bit application in a 64-bit machine (not under Wow6432 registry), selecting the “This registry is associated with a 64-bit application” is required)  Note: This setting has been configured as User setting and will be evaluated only when user is logged |
| 08. Change to Compliance Rules tab and click New |
| 09. On Specify rules to define compliance conditions for this setting, type Mozila Firefox CurrentVersion Equals 49.0.1 (x64 en-US) in the Name, on the following values: type 49.0.1 (x64 en-US) and select Report noncompliance if this setting instance is not found and select Critical with Event on Noncompliance severity for reports. click Ok twice |
| 10. Once back to the specify settings for this application, click Next |
| 11. Under specify compliance rules for this application, confirm that 1 rule has been already created and click Next |
| 12. Under specify the client operating system that will assess this configuration item for compliance, click Next |
| 13. Under the wizard will create an operating system configuration item with the following settings, click Next |
| 14. Under The create Configuration Item Wizard completed successfully, click Close |

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

New-CMConfigurationItem -Name "Validate Mozilla Firefox Version" -CreationType WindowsApplication | Add-CMComplianceSettingRegistryKeyValue -ValueRule -DataType String -Name "Mozila Firefox CurrentVersion" -Hive LocalMachine -KeyName "Software\Mozilla\Mozilla Firefox" -ValueName "CurrentVersion" -Is64Bit -RuleName "CurrentVersion Equals 49.0.1 (x64 en-US)" -ExpressionOperator IsEquals -ExpectedValue "49.0.1 (x64 en-US)" -NoncomplianceSeverity Critical -ReportNoncompliance

* 1. Creating Baselines

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Asset and Compliance. |
| 02. Expand Compliance Settings an$cid click Configuration Baselines |
| 03. Select Compliance Baselines and click Create Configuration Baseline |
| 04. Under Create Configuration Baseline type Workstation Baseline as Name and click Add -> Configuration Items |
| 05. Add All Available configuration items and click OK twice |

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

New-CMBaseline -Name "Workstation Baseline"

$ci1 = Get-CMConfigurationItem -name "Internet Explorer Default Start Page" -Fast

$ci2 = Get-CMConfigurationItem -name "Validate Mozilla Firefox Version" -Fast

Set-CMBaseline -Name "Workstation Baseline" -AddOSConfigurationItem $ci1.CI\_ID -AddRequiredConfigurationItem $ci2.CI\_ID

* 1. Deploying Baselines

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Asset and Compliance. |
| 02. Expand Compliance Settings and click Configuration Baselines |
| 03. Select Workstation Baseline and click Deploy |
| 04. Under Deploy Configuration Baselines, confirm Remediate noncompliant rules when supported is selected and click Browse |
| 05. Select Windows 10 Workstation and click Ok twice |

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

$Name = "Workstation Baseline"

$schedule = New-CMSchedule -RecurCount 1 -RecurInterval Days

New-CMBaselineDeployment -Name $Name -CollectionName "Windows 10 Workstations" -EnableEnforcement $true -Schedule $schedule

* 1. Starting Validation Compliance Settings

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open and Close Internet Explorer.  Note: Confirm the Start Page is not set as www.thedesktopteam.com.com |
| 02. Open Control Panel and then click Configuration Manager |
| 03. Change to the Actions Tab, select Machine Policy Retrieval & Evaluation Cycle and click Run now  Note: Using this option will force the client to connect to the server and update its settings. By default, this happen every 60 minutes and can be changed under Client Settings -> Client Policy -> Client policy polling interval (minutes) |
| 04. Under Machine Policy Retrieval & Evaluation Cycle click Ok  Note: Depending on the MECM environment, the user policy retrieval & evaluation cycle can take few minutes |
| 05. Change to Configurations tab.  Note: click refresh button may be needed |
| 06. Select the Workstation Baseline and click evaluate.  Note: Repeat the process for the WKS0002 machine |

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

(Get-ItemProperty -Path 'HKCU:\Software\Microsoft\Internet Explorer\Main').'Start Page'

$SMSCli = [wmiclass] "root\ccm:SMS\_Client"

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000021}")

start-sleep 10

$SMSCli.TriggerSchedule("{00000000-0000-0000-0000-000000000022}")

Start-Sleep 60

$Baselines = gwmi -Namespace root\ccm\dcm -Class SMS\_DesiredConfiguration -Filter "DisplayName = 'Workstation Baseline'"

$Baselines | % {

$MC = [WmiClass]"root\ccm\dcm:SMS\_DesiredConfiguration"

$InParams = $mc.psbase.GetMethodParameters("TriggerEvaluation")

$InParams.IsEnforced = $true

$InParams.IsMachineTarget = $false

$InParams.Name = "$($\_.Name)"

$InParams.Version = "$($\_.Version)"

$MC.InvokeMethod("TriggerEvaluation", $InParams, $null)

}

start-sleep 60

(Get-ItemProperty -Path 'HKCU:\Software\Microsoft\Internet Explorer\Main').'Start Page'

* 1. Monitoring Baselines from Client

|  |
| --- |
| Perform this task on the wks0001 virtual machine logged on as user01 |
| 01. Open Control Panel and then click Configuration Manager |
| 02. Change to Configurations tab |
| 03. Select the Baseline you want to view the report and click View Report  Note: You must be member of the local administrators group to visualize a Baseline report from the client |
| 04. You can also review the following logs:   * C:\Windows\ccm\logs\CIAgent.log: Records details about the process of remediation and compliance for compliance settings, software updates, and application management. * C:\Windows\ccm\logs\CIDownloader.log: Records details about configuration item definition downloads. * C:\Windows\ccm\logs\CITaskMgr.log: Records information about configuration item task scheduling. * C:\Windows\ccm\logs\DCMAgent.log: Records high-level information about the evaluation, conflict reporting, and remediation of configuration items and applications. * C:\Windows\ccm\logs\DCMReporting.log: Records information about reporting policy platform results into state messages for configuration items. * C:\Windows\ccm\logs\DcmWmiProvider.log: Records information about reading configuration item from Windows Management Instrumentation (WMI). |

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

$Baselines = gwmi -Namespace root\ccm\dcm -Class SMS\_DesiredConfiguration -Filter "DisplayName = 'Workstation Baseline'"

$ComplainceTable = @{

'0' = 'Non-Compliant'

'1' = 'Compliant'

'2' = 'Submitted'

'3' = 'Detecting'

'4' = 'Detecting'

'5' = 'Not Evaluated'

}

$StatusTable = @{

'0' = 'Evaluated'

'2' = 'Evaluating'

'5' = 'Not Evaluated'

}

#simple report

$Baselines | Select Version, @{Name='Status' ; Expression = { $StatusTable[$\_.Status.ToString()] } }, @{Name='Compliance' ; Expression = { $ComplainceTable[$\_.Lastcompliancestatus.ToString()] } }, @{Name='Last Evaluation' ; Expression = {$\_.ConvertToDateTime($\_.LastEvalTime) } } | format-table -AutoSize

#logs

Start-Process -Filepath ("c:\windows\cmtrace.exe") -ArgumentList ("C:\Windows\ccm\logs\CIAgent.log C:\Windows\ccm\logs\CIDownloader.log C:\Windows\ccm\logs\CITaskMgr.log C:\Windows\ccm\logs\DCMAgent.log C:\Windows\ccm\logs\DCMReporting.log C:\Windows\ccm\logs\DcmWmiProvider.log")

* 1. Monitoring Baselines via Console

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Click Deployments. |
| 03. Select the Workstation baseline deployment and click Run Summarization |
| 04. Click Ok when the Configuration Manager confirmation message appear. |
| 05. Select the Workstation baseline deployment and click View Status |
| 06. Under compliant see all assets that are compliant with the baseline |
| 07. Change to the Non-Compliant tab to see all assets in that state |

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

$SiteCode = "001"

$Deployments = Get-CMDeployment -CollectionName "Windows 10 Workstations" | where {$\_.SoftwareName -eq "Workstation Baseline"}

$ID = $Deployments.AssignmentID

$Deployments | Invoke-CMDeploymentSummarization

start-sleep 30

#Non-Compliant

gwmi -namespace root\sms\site\_$SiteCode -class "SMS\_DCMDeploymentNonCompliantAssetDetails" -Filter "AssignmentID = $($ID) and (RuleSubState = 0)" | select AssetName, RuleName, RuleStateDisplay

#Compliant

gwmi -namespace root\sms\site\_$SiteCode -class "SMS\_DCMDeploymentCompliantAssetDetails" -Filter "AssignmentID = $($ID)" | select AssetName

* 1. Monitoring Baselines via Reports

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Under monitoring, expand Reporting and click Reports |
| 03. Search for Compliance and Settings. Select Summary compliance by configuration baseline and click Run |
| 04. Once the report is open, you can navigate using the links.  Note: You can drill down to a more specific report using the links inside the reports |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

#Open Report

Invoke-CMReport -ReportPath "Compliance and Settings Management/Summary compliance by configuration baseline" -SiteCode "$SiteCode" -SrsServerName "$servername"

* 1. Monitoring Baselines via Collections

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Assets and Compliance. |
| 02. Expand Compliance Settings and click Configuration Baselines |
| 03. Select Workstation Baseline and click Deployments |
| 04. Select Windows 10 workstations and click Create New Collection -> Compliant |
| 05. On Create Device Collection Wizard, click Next |
| 06. On Membership Rules, select Use incremental updates for this collection and click Next |
| 07. On Summary, click Next |
| 08. On Completion, click Close |
| 09. Click Device Collections |
| 10. Select Workstation Baseline\_Windows 10 Workstations\_Compliant Collection and click Show Members  Note: Once the collection is created, there is a process to populate it and it may take a while. In this lab, wait 30 seconds or refresh it couple of times until you see Member Count increment to 2 |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$Deployments = Get-CMDeployment -CollectionName "Windows 10 Workstations" | where {$\_.SoftwareName -eq "Workstation Baseline"}

$CollUpdate = New-CMSchedule -Start "01/01/2015 9:00 PM" -DayOfWeek Saturday -RecurCount 1

$Collection = New-CMDeviceCollection -Name "Compliant Machines" -LimitingCollectionName "Windows 10 Workstations" -RefreshSchedule $CollUpdate -RefreshType Both

Add-CMDeviceCollectionQueryMembershipRule -CollectionId $Collection.CollectionID -RuleName "Compliant Machines" -QueryExpression "select \* from SMS\_R\_System inner join SMS\_G\_System\_DCMDeploymentState on SMS\_G\_System\_DCMDeploymentState.ResourceID = SMS\_R\_System.ResourceId WHERE BaselineID = '$($Deployments.ModelName)' AND CollectionID = '$($Deployments.CollectionID)' AND ComplianceState = 1"

start-sleep 20

Get-CMCollectionMember -CollectionName "Compliant Machines" | select Name

1. Data Warehouse

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | The Data Warehouse service point for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/manage/data-warehouse> |
| Description | In this chapter, we will look at the Data Warehouse, a long-term historical data repository for Configuration Manager |

* 1. Data Warehouse service point

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Servers and Site System Roles |
| 03. Right click \\SRV0002.classroom.intranet and click Add Site System Roles |
| 04. On Add Site System Roles Wizard, General, click Next |
| 05. Under proxy, click Next |
| 06. Under Specify roles for this server, select Data Warehouse service point and click Next |
| 07. Under Specify Data Warehouse service point settings, use:   * SQL Server fully qualified domain name: SRV0002.classroom.intranet * Database name: CM\_001\_DW * SQL Server port: 1433 * User name: CLASSROOM\svc\_ssrsea   click Next |
| 08. Under specify Data Warehouse service point schedule, leave the default and click Next |
| 09. Under confirm the settings, click Next |
| 10. Under You have successfully completed the Add Site System Roles wizard with the following settings click close |
| 11. Click Monitoring |
| 12. Expand System Status and click Component Status |
| 13. Search for DATA |
| 14. Right Click DATA\_WAREHOUSE\_SERVICE\_POINT, Show Messages and click All |
| 15. Under Status Messages: Set Viewing Period, click OK |
| 16. Verify the existence of Message ID 1013, 1014 and 1015 |
| 17. Verify the existence of Message ID 11201  Note: If this Message ID exist, MECM has initiated the synchronization with the Data Warehouse database. |
| 18. Double click message 11201 messages to see its details. Once done, click Ok |
| 19. Verify the existence of Message ID 11203  Note: If this Message ID exist, MECM has completed the synchronization with the Data Warehouse database. |
| 20. Double click message 11203 messages to see its details. Once done, click Ok |
| 21. You can also review the following logs:   * C:\ConfigMgr\Logs\dwsssetup.log: Records the installation wrapper process. * C:\ConfigMgr\Logs\dwssmsi.log: Records details of installation. * C:\ConfigMgr\Logs\srsrp.log: Records information about the reports being imported into the reporting services point. * C:\ConfigMgr\Logs\Microsoft.ConfigMgrDataWarehouse.log: Records information about data synchronization between the site database to the data warehouse database. |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$account = "CLASSROOM\svc\_ssrsea"

if ((Get-CMSiteSystemServer -SiteSystemServerName "$servername") -eq $null) { New-CMSiteSystemServer -SiteCode $SiteCode -UseSiteServerAccount -ServerName $servername }

Add-CMDataWarehouseServicePoint -SiteSystemServerName $ServerName -UserName $account -DataWarehouseDatabaseName "CM\_$($SiteCode)\_DW" -DataWarehouseDatabaseServerName $ServerName -DataWarehouseSqlPort 1433 -SiteCode $SiteCode -WeekFrequency 1

start-sleep 90

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsgin.InsStrValue from SMS\_StatMsg stmsg inner join SMS\_StatMsgInsStrings stmsgin on stmsg.RecordID = stmsgin.RecordID where stmsg.Component = 'DATA\_WAREHOUSE\_SERVICE\_POINT' and stmsg.MessageID = 1013 and stmsgin.InsStrIndex = 0 and stmsgin.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found DATA\_WAREHOUSE\_SERVICE\_POINT 1013 id's"

break

} else { Start-Sleep 10 }

}

while ($true)

{

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'DATA\_WAREHOUSE\_SERVICE\_POINT' and stmsg.MessageID = 1014 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found DATA\_WAREHOUSE\_SERVICE\_POINT 1014 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'DATA\_WAREHOUSE\_SERVICE\_POINT' and stmsg.MessageID = 1015 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found DATA\_WAREHOUSE\_SERVICE\_POINT 1015 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'DATA\_WAREHOUSE\_SERVICE\_POINT' and stmsg.MessageID = 11201 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found DATA\_WAREHOUSE\_SERVICE\_POINT 11201 id's"

break

} else { Start-Sleep 10 }

}

while ($true) {

$component = gwmi -Namespace ("root\sms\site\_$SiteCode") -query "select stmsg.\* from SMS\_StatMsg stmsg where stmsg.Component = 'DATA\_WAREHOUSE\_SERVICE\_POINT' and stmsg.MessageID = 11203 and stmsg.SiteCode = '$SiteCode'"

if ($component -ne $null) {

Write-Host "Found DATA\_WAREHOUSE\_SERVICE\_POINT 11203 id's"

break

} else { Start-Sleep 10 }

}

* 1. Data Warehouse retention period

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open SQL Server Management Studio |
| 02. Connect to the SRV0002 server |
| 03. Click New Query |
| 04. Change the Database to CM\_001 |
| 05. Type UPDATE sc\_sysresuse\_property SET Value3 = 1460 WHERE Name = 'DataRetentionDays' and click Execute  Note: Change the 1460 to the value required. If you want to change to 3 years, use 1095 (it is just 3 times 365 days) |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

Set-CMDataWarehouseServicePoint -SiteSystemServerName $servername -DataRetentionDays 1460 -SiteCode $SiteCode

* 1. SQL Server Permission

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open SQL Server Management Studio |
| 02. Connect to the SRV0002 server |
| 03. Expand Security -> Logins and select CLASSROOM\svc\_ssrsea |
| 04. Right click CLASSROOM\svc\_ssrsea and click Properties |
| 05. Under Login Properties - CLASSROOM\svc\_ssrsea, click User Mapping |
| 06. Select CM\_001\_DW and check Map. Under Database role membership for: CW\_001\_DW, select db\_datareader and click OK |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

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

$conn.ConnectionString = "Data Source=SRV0002;Initial Catalog=CM\_$($SiteCode)\_DW;trusted\_connection = true;"

$conn.Open()

$SqlCommand = $Conn.CreateCommand()

$SqlCommand.CommandTimeOut = 0

$SqlCommand.CommandText = "create user [CLASSROOM\svc\_ssrsea] from login [CLASSROOM\svc\_ssrsea]"

$SqlCommand.ExecuteNonQuery()

Start-Sleep 5

$SqlCommand.CommandText = "EXEC sp\_addrolemember 'db\_datareader', 'CLASSROOM\svc\_ssrsea'"

$SqlCommand.ExecuteNonQuery()

* 1. Force Manual Synchronisation

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Servers and Site System Roles |
| 03. Select \\SRV0002.classroom.intranet |
| 04. right click Data Warehouse service point and then Click Properties |
| 05. On Data Warehouse service point Properties, change to the Synchronization Settings tab |
| 06. On Synchronization Settings, click Synchronize now |
| 07. Click Ok twice |
| 08. You can also review the following logs:   * C:\ConfigMgr\Logs\Microsoft.ConfigMgrDataWarehouse.log: Records information about data synchronization between the site database and the data warehouse database. |

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

Set-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\SMS\DWSS' -Name 'LastSynchronizationTime' -Value ''

Get-Service -Name 'DATA\_WAREHOUSE\_SERVICE\_POINT' | Restart-Service

* 1. Monitoring Historical data via Reports

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Monitoring. |
| 02. Under monitoring, expand Reporting and click Reports |
| 03. Search for Historical. Select General hardware inventory – Historical and click Run |
| 04. Once the report is open, you can navigate using the links.  Note: You can drill down to a more specific report using the links inside the reports |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

#Open Report

$dict = @{"Device Name"="WKS0001" }

Invoke-CMReport -ReportPath "Data Warehouse/General hardware inventory - Historical" -SiteCode "$SiteCode" -SrsServerName "$servername" -ReportParameter $dict

1. Role Based Access Control

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | Configuration Manager Role-Based Administration  <https://docs.microsoft.com/en-us/configmgr/develop/core/servers/configure/role-based-administration>  Fundamentals of role-based administration for Configuration Manager  <https://docs.microsoft.com/en-gb/configmgr/core/understand/fundamentals-of-role-based-administration>  Configure role-based administration  <https://docs.microsoft.com/en-us/configmgr/core/plan-design/security/configure-security#BKMK_ConfigureRBA> |
| Description | In this chapter, we will look on how to configure MECM to secure access to the environment and allow certain group of people to perform only the tasks they are required without full access to the MECM |

* 1. Creating Security Scope

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Security and click Security Scopes |
| 03. Select Security Scopes and click Create Security Scope |
| 04. Under Create and assign security scope type Application Administrator for Windows 10 Machines as Security scope name and click Ok |

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

$SecurityScope = "Application Administrator for Windows 10 Machines"

New-CMSecurityScope -Name "$SecurityScope"

* 1. Using Security Scope

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Software Library. |
| 02. Expand Application Management and click Applications |
| 03. Select Firefox 49 and Java8 and click Set Security Scope |
| 04. Under Set Security scope for the selected securable objects, select Application Administrator for Windows 10 Machines and click Ok |

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

$SecurityScope = "Application Administrator for Windows 10 Machines"

get-cmApplication -name "Firefox 49" | Add-CMObjectSecurityScope -Name "$SecurityScope"

get-cmApplication -name "Java8" | Add-CMObjectSecurityScope -Name "$SecurityScope"

* 1. Creating an Application Administrator

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Security and click Administrative Users |
| 03. Select Administrative Users and click Add User or Group |
| 04. Under Specify a user or group to add as a Configuration Manager administrative user and use the following:   * User or group name: CLASSROOM\Workstation Admins * Assigned security roles: Application Administrator * Only the instances of objects that area assigned to the specified security scope or collections: Selected * Security scopes and collections: Application Administrator for Windows 10 Machines security scope and Windows 10 Workstations Collection.   Click Ok |

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

New-CMAdministrativeUser -Name "CLASSROOM\Workstation Admins" -RoleName @("Application Administrator") -CollectionName @("Windows 10 Workstations") -SecurityScopeName @("Application Administrator for Windows 10 Machines")

* 1. Testing new Security Rights

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Select the Configuration Manager Console and click Run as different user |
| 02. Log on as workstationadmin |
| 03. Once the console is open, notice that there are few missing nodes |
| 04. Click Devices and notice that only two devices appear |
| 05. Click Device Collections and notice that only Windows 10 Workstation and Workstation Baseline\_Windows 10 Workstations\_Noncompliant appears  Note: The Workstation Baseline\_Windows 10 Workstations\_Noncompliant collection appear because it is limited by the Windows 10 workstations |
| 06. Click Software Library |
| 07. Expand Application Management and click Applications.  Note: Only application associated with Security scope are visible. |

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

$username = "CLASSROOM\workstationadmin"

$password = 'Pa$$w0rd' | convertto-securestring -AsPlainText -Force

$cred = new-object -typename System.Management.Automation.PSCredential -argumentlist $username, $password

Start-Process "C:\ConfigMgr\AdminConsole\bin\Microsoft.ConfigurationManagement.exe" -Credential $Cred

1. Backup via MECM

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | Backup and recovery for Configuration Manager  <https://docs.microsoft.com/en-gb/configmgr/core/servers/manage/backup-and-recovery>  The CD.Latest folder for Configuration Manager  <https://docs.microsoft.com/en-us/configmgr/core/servers/manage/the-cd.latest-folder>  SQL Server Backup Recommendations for Configuration Manager  <https://stevethompsonmvp.wordpress.com/2013/06/07/sql-server-backup-recommendations-for-configuration-manager/> |
| Description | In this chapter, we will look at the MECM Backup and how to perform the tasks required to backup a MECM environment |

* 1. Configuring Backup

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Site Configuration and click Servers and Sites |
| 03. Select 001 – Training Lab site and click Site Maintenance |
| 04. Under Site Maintenance, select Backup Site Server and click Edit |
| 05. Under Backup Site Server Properties, check Enable this task and backup destination select \\srv0001\MECMBackup. Click Ok three times |

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

$SiteCode = "001"

Set-CMSiteMaintenanceTask -SiteCode $SiteCode -Name "Backup SMS Site Server" -DaysOfWeek Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday -Enabled $true -BeginTime "02:00" -LatestBeginTime "05:00" -devicename \\srv0001\MECMBackup

* 1. Starting Backup Manually

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Services console |
| 02. Select SMS\_SITE\_BACKUP and click Start |

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

Get-Service -Name "SMS\_SITE\_BACKUP" | Start-Service

* 1. Monitoring Backup

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Event Viewer |
| 02. Expand Windows Logs and click Application |
| 03. Verify the existence of the following Event IDs:   * 5055 (Site Backup task is starting) * 6829 (SMS Writer is about to stop the ConfigMgr Services as part of the preparation for the Site backup) * 3197 (I/O is frozen on Database) * 3198 (I/O is resumed on Database) * 5056 (Site Backup is starting to copy the files from the snapshot) * 5057 (Site Backup has successfully completed copying the files from the snapshot) * 6833 (Site Backup task has completed successfully) |
| 04. Open Windows Explorer and navigate to \\srv0001\MECMBackup\001Backup. This folder will contain all files needed to restore a MECM infrastructure.  Note: MECM will keep only the latest backup under this folder. To have a historical backup, a file called AfterBackup.bat can be created under C:\ConfigMgr\inboxes\smsbkup.box with the content like:  xcopy "\\srv0001\MECMBackup\\*.\*" "\\AnotherServer\Folder\%DATE%\<SITE CODE>Backup\\*.\*" /E /O /C  Note: MECM only backup the basic information to be able to recover a site. It will not backup the source files, Content Library folder, SUSDB and ReportServer database, etc. For more info about what is/is not backup by default, refer to Steve Thompson (MECM MVP) blog: MECM File Backup Considerations  (<https://stevethompsonmvp.wordpress.com/2017/01/17/sccm-file-backup-considerations/>) |
| 14. You can also review the following logs:   * C:\ConfigMgr\Logs\ smsbkup.log: Records details about the site backup activity. |

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

Get-Eventlog -Newest 100 -LogName Application -Source "SMS Server" -After (Get-Date).AddMinutes(-60) | where {$\_.eventID -in (5055, 6829,3197,3198,5056,5057,6833)} | select EventID, Message, TimeGenerated | sort-object TimeGenerated -Descending | format-list

Get-ChildItem -Path 'filesystem::\\srv0001\SCCMBackup'

Start-Process -Filepath ("c:\windows\cmtrace.exe") -ArgumentList ("C:\ConfigMgr\Logs\smsbkup.log")

1. Restore

|  |  |
| --- | --- |
| Computers used in  this Lab | ROUTER01  SRV0001  SRV0002 |
| More information | Backup and recovery for Configuration Manager  <https://docs.microsoft.com/en-gb/configmgr/core/servers/manage/backup-and-recovery> |
| Description | In this chapter, we will look at the steps required to recover a MECM environment from a Backup |

* 1. Stop MECM Site

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open Command Prompt (Run as Administrator) |
| 02. execute preinst.exe /stopsite from C:\ConfigMgr\bin\x64\00000409 |
| 03. Once the command is execute successfully, all site services will be stopped |

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

Start-Process -Filepath ("C:\ConfigMgr\bin\x64\00000409\preinst.exe") -ArgumentList ('/stopsite') -wait

Start-Sleep 5

* 1. Delete MECM Database

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Open SQL Server Management Studio |
| 02. Connect to the SRV0002 server |
| 03. Expand Databases |
| 04. Select CM\_001 and click Delete |
| 05. Under delete object, make sure delete backup and restore history information for database and close existing connections are checked. Click Ok |
| 06. Confirm the database has been deleted |

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

$Server="SRV0002.classroom.intranet"

$dbName="CM\_001"

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

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

$SMOserver.Databases | select Name, Size,DataSpaceUsage, IndexSpaceUsage, SpaceAva

if ($SMOserver.Databases[$dbName] -ne $null) {

$smoserver.KillAllProcesses($dbname)

$smoserver.databases[$dbname].drop()

}

* 1. Restore MECM Site

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Execute splash.hta from \\srv0001\MECMBackup\001Backup\CD.Latest |
| 02. Under Microsoft Endpoint Configuration Manager, click Install |
| 03. On Before You Begin, click Next |
| 04. On Available Setup Options, click Recover a site and click Next |
| 05. Under Site Server and Database Recovery Options select Recover the site database using the backup set of the following locations and type \\srv0001\MECMBackup\001Backup. Click Next |
| 06. Under Site Recover Information, click Next |
| 07. Under Product Key, Click Next |
| 08. Under Product License Terms, select all I accept the license terms and click Next |
| 09. Under Prerequisite Downloads, select Download required files and in path type C:\trainingfiles and click Next  Note: We cannot use the already downloaded files as they have been downloaded for 1902 version |
| 10. Under Site and Installation Settings, click Next |
| 11. Under Database Information (Server Information), click Next |
| 12. Under Database Information (File Location Information), click Next |
| 13. Under Diagnostic and Usage Data, click Next |
| 14. Under Settings summary, review the settings and click Next |
| 15. The prerequisite check will validate the system. Once it is done, click Begin Install |
| 16. Once the installation is completed, click Next.  Note: The Site Reset takes less than 10 minutes, however, post-recovery tasks can take extra minutes to complete, depending on the size of the infrastructure. |
| 17. Under finished, confirm some of the post-recovery actions that need be done by the administrator, click Close. |
| 18. At the root of C-partition, multiple log files are created that tell the status of the installation:   * ConfigMgrPrereq.log: Prerequisites review log * ConfigMgrSetup.log: site server installation log * ConfigMgrSetupWizard.log: installation wizard log |

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

$inifile = @"

[Identification]

Action=RecoverPrimarySite

CDLatest=1

[Options]

ProductID=EVAL

SiteCode=001

SiteName=Training Lab

SMSInstallDir=c:\ConfigMgr

SDKServer=SRV0002.classroom.intranet

PrerequisiteComp=0

PrerequisitePath=C:\trainingfiles

AdminConsole=0

JoinCEIP=0

[SQLConfigOptions]

SQLServerName=SRV0002.classroom.intranet

SQLServerPort=1433

DatabaseName=CM\_001

SQLSSBPort=4022

SQLDataFilePath=C:\SQLServer\MSSQL14.MSSQLSERVER\MSSQL\DATA\

SQLLogFilePath=C:\SQLServer\MSSQL14.MSSQLSERVER\MSSQL\DATA\

[CloudConnectorOptions]

CloudConnector=0

CloudConnectorServer=

UseProxy=0

ProxyName=

ProxyPort=

[SystemCenterOptions]

SysCenterId=nu8ZfCX5X5nhqqqlnJBaqIP6l50bxWjylWL/Q0Pl5vQ=

[HierarchyExpansionOption]

[RecoveryOptions]

ServerRecoveryOptions=4

DatabaseRecoveryOptions=10

BackupLocation=\\srv0001\MECMBackup\001Backup\

"@

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

Start-Process -Filepath ("\\srv0001\MECMBackup\001Backup\CD.Latest\SMSSETUP\BIN\X64\setup.exe") -ArgumentList ('/script "\\srv0001\TempFiles\restorecmcb.ini"') -wait -NoNewWindow

* 1. Post-Restore Tasks
     1. Accounts Password Reset

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Expand Security and click Accounts |
| 03. Select CLASSROOM\svc\_mecmna and click Properties |
| 04. On CLASSROOM\svc\_mecmna Properties, click Set |
| 05. On CLASSROOM\svc\_mecmna Windows User Account, type:   * Password: Pa$$w0rd * Confirm Password: Pa$$w0rd   Click Verify |
| 06. Under verify type \\SRV0002\sms\_site for Network Share and click Test Connection |
| 07. Once the connection was successfully verified, click Ok three times  Note: Repeat the password reset for the CLASSROOM\svc\_ssrsea and CLASSROOM\svc\_mecmpush |

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

$SiteCode = "001"

$Secure = 'Pa$$w0rd'| ConvertTo-SecureString -AsPlainText -Force

$account = "CLASSROOM\svc\_mecmna"

Set-CMAccount -UserName "$account" -Password $Secure -SiteCode $SiteCode

$account = "CLASSROOM\svc\_ssrsea"

Set-CMAccount -UserName "$account" -Password $Secure -SiteCode $SiteCode

$account = "CLASSROOM\svc\_mecmpush"

Set-CMAccount -UserName "$account" -Password $Secure -SiteCode $SiteCode

* + 1. Distribution Point Self-Signed Certificate Reset

|  |
| --- |
| Perform this task on the SRV0002 virtual machine logged on as mecmadmin |
| 01. Start Configuration Manager Console and Click Administration. |
| 02. Click Distribution Points |
| 03. Select SRV0002.CLASSROOM.INTRANET and click Properties |
| 04. On SRV0002.CLASSROOM.INTRANET Properties, change the create self-signed certificate date/time and click Ok  Note: Add a minute to the already existing time is enough to generate a new certificate |

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

$SiteCode = "001"

$servername = "SRV0002.classroom.intranet"

$CurrentExpirationDate = [DateTime]::FromFileTime(((Get-CMDistributionPoint -SiteSystemServerName "$servername").Props | where {$\_.PropertyName -eq "CertificateExpirationDate" }).Value1)

Set-CMDistributionPoint -SiteSystemServerName "$servername" -CertificateExpirationTimeUtc "$($CurrentExpirationDate.AddMinutes(1).ToString())"

1. Appendix B – Unmissable Sites

|  |  |
| --- | --- |
| Site Address | Comments |
| <http://www.thedesktopteam.com> | MVP Raphael Perez and MVP David Nudelman |
| <https://www.rflsystems.co.uk> | 3rd Party tools and Consultancy |
| <http://blog.colemberg.ch> | MVP Mirko Colemberg |
| <http://www.dekeukelaere.com>  <http://www.scug.be/tim> | MVP Tim De Keukelaere |
| <http://www.ronnipedersen.com> | MVP Ronni Pedersen |
| <http://sccm.biz> | MVP Nicolai Henriksen |
| <http://Stevethompsonmvp.wordpress.com> | MVP Steve Thompson |
| <https://rzander.azurewebsites.net/> | MVP Roger Zander |
| <http://sms-hints-tricks.blogspot.com/> | MVP Matthew Hudson |
| <http://faqshop.com/> | MVP Cliff Hobbs |

1. Appendix A – Tools
   1. DataExplorer

Data Explorer is a data visualization platform which helps enterprises to streamline the access control and management of strategic business information.

By displaying key metrics and indicators in one single screen, the software interface can be tailored and expandable to support objectives and needs.

More info at: <https://www.rflsystems.co.uk>

* 1. HealthCheck Toolkit

The Healthcheck tool supports you to analyse the health conditions of the Configuration Manager in an easy and practical manner.

Through the software, users can assess the status of the Configuration Manager’s performance, latest updates, disk space, client data and other key indicators.

More info at: <https://www.rflsystems.co.uk>

* 1. PatchMaster

PatchMaster is designed to automate the creation and delivery of Microsoft Patches to an SCCM Hierarchy and is available as freeware.

More info at: <https://gallery.technet.microsoft.com/PatchMaster-51716e13>

* 1. SCCM Client Center

The tool is designed for IT Professionals to troubleshoot SMS/SCCM Client related issues. The SCCM Client Center provides a quick and easy overview of client settings, including running services and SCCM settings in a good easy to use, user interface.

More info at: <https://sourceforge.net/projects/smsclictr/>

* 1. Mark Cochrane RegkeytoMof 3.3a

RegKeytoMof is used to quickly create custom Hardware Inventory entries formatted correctly for the sms\_def.mof and configuration.mof files, when the target is Registry keys.

More info at: <http://www.enhansoft.com/blog/how-to-use-regkeytomof>

Download at: <http://mnscug.org/images/Sherry/RegKeyToMOFv33a.zip>

* 1. OSD WebPortal

Complementary tool for SCCM that enables a couple of scenarios to simplify the OS staging process.

More info at: <https://gallery.technet.microsoft.com/OSD-Webportal-10139926-e13f2d78>

* 1. RCT Community

Recast’s Right Click Tools for SCCM help administrators reduce the complexity of systems management—simplifying everyday tasks and providing smart automation for more complex jobs. When you can do more; your environment is healthier, your end users are happier, and your organization is more secure.

More info at: <https://recastsoftware.com/#community>

* 1. RuckZuck

Software package manager, a quick way to install and update your Software.

More info at: <http://ruckzuck.tools/>

* 1. Clean Software Update Groups console extension for ConfigMgr

Console extension for ConfigMgr for cleaning up Software Update Groups automatically using PowerShell.

More info at: <https://gallery.technet.microsoft.com/Clean-Software-Update-5ae68ba2>

* 1. Cireson Remote Manage app

The Cireson Remote Manage app is the analyst’s dream. It is a toolbox of common remote functionalities needed by an analyst. Its integration with Configuration Manager allows an analyst to quickly deploy software, run diagnostic processes, view and manipulate services, processes, and view information – all without end-user interaction! The Cireson Remote Manage app is designed around simple remote administration. It’s an all-in-one toolbox, that doesn’t interrupt the end user.

More info at: <http://cireson.com/apps/remote-manage/>

* 1. Reg2CI

Reg2CI is a Command-Line Tool to convert .POL (Policy) or .REG Files (Registry) into System Center Configuration Manager CI's (Configuration Items).

More info at: <http://reg2ci.codeplex.com/>

* 1. Collection Commander

The tool is designed for IT Professionals to trigger PowerShell Scripts on a list of devices. Collection Commander provides a bunch of PowerShell Scripts to manage and troubleshoot Configuration Manager 2012 Agents. But ConfigMgr is not a requirement. The tool can run as standalone solution.

It's possible to integrate own PS scripts by placing the .ps1 files in the "PSScripts" Directory. That makes it super easy to access your own PowerShell Script repository with a few clicks. The selected PS Script will run against all marked devices (mark the full row !) with multiple Threads so you can run it against hundreds of Devices.

More info at: <http://cmcollctr.codeplex.com/>

* 1. SQL Server Index and Statistics Maintenance

IndexOptimize is the SQL Server Maintenance Solution’s stored procedure for rebuilding and reorganizing indexes and updating statistics. IndexOptimize is supported on SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, and SQL Server 2014.

More info at: https://ola.hallengren.com/sql-server-index-and-statistics-maintenance.html

* 1. PowerShell – SQL Audit Script

Used to baseline SQL Server instance and find common mis-configurations. Document current settings, database files, database properties and much more. Script output is converted to an XLS for analysis.

More info at: <https://stevethompsonmvp.wordpress.com/2014/05/19/powershell-sql-audit-script/>

* 1. 1E’s Free Tools

You can get cool tools, tips, white papers and scripts to simplify the daily life of the ConfigMgr Administrator. These free tools are focused on optimizing processes and resolving common issues to help you manage your Software Lifecycle Automation.

More info at: <http://www.1e.com/free-tools/>

* 1. ConfigMgr Task Sequence Monitor

ConfigMgr Task Sequence Monitor is an application that connects to your System Center Configuration Manager database to display data from task sequence deployments. It can be used to monitor running task sequences, such as OS deployments, or to review the results of historic task sequence deployments. If you are using MDT integration in ConfigMgr, you can view monitoring data both from MDT and ConfigMgr for your ZTI OS deployments.

More info at: <https://smsagent.wordpress.com/tools/configmgr-task-sequence-monitor/>

* 1. ConfigMgr OSD FrontEnd

ConfigMgr OSD FrontEnd has been developed with the goal to ultimately function in any environment easing some of the pains with operating system deployment that we have today. This version of the software has been created with the purpose of being initiated outside of the task sequence as a prestart command.

More info at: <http://www.scconfigmgr.com/2017/01/02/configmgr-osd-frontend-public-preview-available/>

* 1. Windows-Nood OSD FrontEnd

The Windows-Nood OSD FrontEnd is a frontend for Configuration Manager which allows Network administrators/techs to use three common OSD scrnarios for UEFI (and Legacy) based hardware, namely Backup, Reinstall and New Computer.

More info at: <https://www.windows-noob.com/forums/topic/11864-the-cm12-uefi-bitlocker-frontend-hta-part-1-the-features/>

**Recast RCT Free 2.4**

1. To have access to the App-V 5.0 SP3 client access to MDOP package is needed. Access the MSDN, TechNet or Microsoft volume license website to download MDOP 2014 R2 [↑](#footnote-ref-2)