

Institution des Chartreux

Documentation déploiement WDS-MDT

Documentation PPE

FAUBLADIER--ANETTE Alexandre, Matéo
Monteiro, Thomas Anthoard
24/05/2025

Table des matières

Introduction.....	1
Installation de la solution de déploiement.....	1
Configuration du serveur DHCP.....	1
Configuration des services de déploiements WDS et MDT.....	4
Ajout de fonctionnalités supplémentaires.....	12
Implémentation d'application lors du déploiement.....	12
Intégration et export de l'image MDT en ISO.....	13
Mise en place d'une base de données SQL.....	13
Sources.....	13

Introduction

Microsoft Deployment Tools (MDT) et Windows Deployment Services (WDS) sont deux services que nous pouvons installer sur un contrôleur Active Directory. Ils permettent de déployer des systèmes d'exploitation Windows directement depuis le réseau. Cette solution puissante offre également la possibilité de personnaliser l'infrastructure en y intégrant des scripts et des applications. Dans notre environnement de production, nous avons effectivement intégré des agents de supervision afin de remonter automatiquement les informations des nouvelles machines créées sur le réseau et de les superviser dès leur mise en service.

Installation de la solution de déploiement

Pour pouvoir installer notre solution, nous allons avoir besoin du matériel suivant :

- Un contrôleur Active Directory (machine Windows Server 1)
- Un Deployment Share connecté à l'Active Directory (machine Windows Server 2)
- Un réseau dédié aux déploiements
- Une machine virtuelle vierge permettant de tester les déploiements via le réseau.

Configuration du serveur DHCP

Pour pouvoir utiliser les services de déploiement via le réseau, il est nécessaire de configurer les options appropriées sur le serveur DHCP de l'Active Directory. Cela permet d'autoriser le démarrage PXE, un outil qui permet de délivrer des images système via le réseau.

Pour nous permettre de pouvoirs installer de manière propre notre serveur de déploiement WDS-MDT, nous avons suivi [la liste des tutoriels suivants](#). Nous avons appliqué les commandes et les actions sur nos machines. Nous permettant ainsi d'obtenir le même résultat.

Dans un premier temps, nous allons pouvoirs ajouter les options nécessaires en effectuant ces commandes PowerShell. Pour l'option 60 (PXE) :

A. DHCP : ajouter la prise en charge de l'option 60

Dans la liste des options DHCP, l'option 60 n'est pas disponible dans la liste. À l'aide de PowerShell (ou de netsh), nous allons pouvoir remédier à cela.

Sur le serveur DHCP, en l'occurrence SRV-ADDS-01 dans mon exemple, ouvrez une console PowerShell en tant qu'administrateur et exécutez cette commande :

```
Add-DhcpServerv4OptionDefinition -ComputerName SRV-ADDS-01 -Name PXEClient -Description "PXE Support" -OptionId 060 -Type
```

Après avoir actualisé la console DHCP, on peut voir que cette option est accessible :

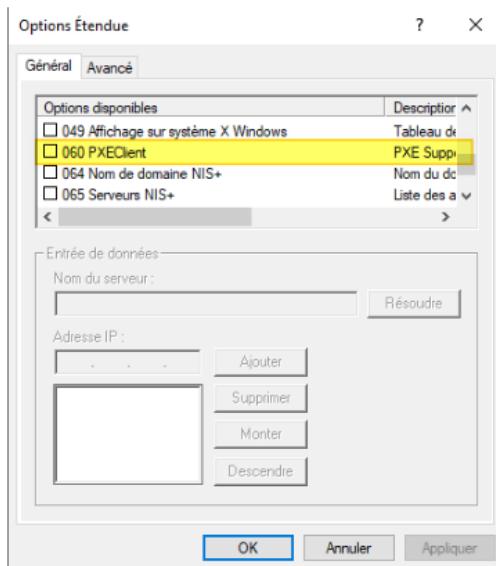


Figure X – Configuration de l'option 60 du DHCP sur notre AD

Une fois cela exécuter, nous allons par la suite définir des variables nous permettant de plus facilement exécuter les commandes qui suivent. Ces dernières nous permettrons de pouvoirs configurer le DHCP de plusieurs manières nous permettant de déployer des images à travers le réseau notamment avec les caractéristiques suivantes :

- Définition de plusieurs classes de fournisseurs DHCP pour plusieurs architecture différentes
- Configuration d'une stratégie pour le mode BIOS x86 et x64.
- Configuration d'une stratégie pour le mode UEFI x86
- Configuration d'une stratégie pour le mode UEFI x64

En premier nous allons déclarer nos variables de nos serveurs avec nos ip que voici :

```

# Nom d'hôte du serveur DHCP
$DhcpServerName = "SRVAD2025"
# Adresse IP du serveur WDS (PXE)
$PxeServerIp = "172.16.10.85"
# Adresse réseau de l'étendue DHCP ciblée
$Scope = "172.16.10.0"

```

Figure X – Déclaration des variables

Puis, grâce aux commandes ci-dessous, nous allons définir trois classes de fournisseurs DHCP correspondantes à des architectures différentes. Par exemple, la valeur "PXEClient: Arch:00007" correspond à du boot PXE pour l'UEFI en x64

```

Add-DhcpServerv4Class -ComputerName $DhcpServerName -Name "PXEClient - UEFI x64" -Type Vendor -Data "PXEClient:Arch:00007" -Description
"PXEClient:Arch:00007"
Add-DhcpServerv4Class -ComputerName $DhcpServerName -Name "PXEClient - UEFI x86" -Type Vendor -Data "PXEClient:Arch:00006" -Description
"PXEClient:Arch:00006"
Add-DhcpServerv4Class -ComputerName $DhcpServerName -Name "PXEClient - BIOS x86 et x64" -Type Vendor -Data "PXEClient:Arch:00000" -Description
"PXEClient:Arch:00000"

```

Figure X – Déclaration de plusieurs fournisseur DHCP

Par la suite nous allons taper cette suite de commande nous permettant ainsi de configurer le DHCP pour réagir aux différents type de BIOS ou UEFI selon leurs architectures :

```

$PolicyNameBIOS = "PXEClient - BIOS x86 et x64"
Add-DhcpServerv4Policy -ComputerName $DhcpServerName -ScopeId $Scope -Name $PolicyNameBIOS -Description "Options DHCP pour boot BIOS x86 et x64" -
Condition Or -VendorClass EQ, "PXEClient - BIOS x86 et x64"
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 066 -Value $PxeServerIp -PolicyName $PolicyNameBIOS
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 067 -Value boot\x64\wdsnbp.com -PolicyName $PolicyNameBIOS

$PolicyNameUEFIx86 = "PXEClient - UEFI x86"
Add-DhcpServerv4Policy -ComputerName $DhcpServerName -ScopeId $Scope -Name $PolicyNameUEFIx86 -Description "Options DHCP pour boot UEFI x86" -
Condition Or -VendorClass EQ, "PXEClient - UEFI x86"
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 060 -Value PXEClient -PolicyName $PolicyNameUEFIx86
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 066 -Value $PxeServerIp -PolicyName $PolicyNameUEFIx86
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 067 -Value boot\x86\wdsmefw.efi -PolicyName $PolicyNameUEFIx86

$PolicyNameUEFIx64 = "PXEClient - UEFI x64"
Add-DhcpServerv4Policy -ComputerName $DhcpServerName -ScopeId $Scope -Name $PolicyNameUEFIx64 -Description "Options DHCP pour boot UEFI x64" -
Condition Or -VendorClass EQ, "PXEClient - UEFI x64"
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 060 -Value PXEClient -PolicyName $PolicyNameUEFIx64
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 066 -Value $PxeServerIp -PolicyName $PolicyNameUEFIx64
Set-DhcpServerv4OptionValue -ComputerName $DhcpServerName -ScopeId $Scope -OptionId 067 -Value boot\x64\wdsmgfw.efi -PolicyName $PolicyNameUEFIx64

```

Figure X – Mise en place des différentes stratégies pour le mode BIOS et UEFI

Une fois ces éléments mis en place, notre DHCP devrait pouvoirs être visible depuis la machine virtuelle vierge.

Configuration des services de déploiements WDS et MDT

Toujours en suivant les informations décrites dans les documentations d'IT-connect, nous allons mettre en place sur notre seconde machine Windows serveur les éléments suivants :

- Service de Déploiement pour Windows (Windows Deployment Services)
- MDT (Microsoft Deployment Toolkit)

Le service WDS nous permettra de pouvoir déployer une image via le réseau tandis que MDT quant à lui nous permet de pouvoir créer une image LTI (Lite Touch Interface) permettant l'installation automatise de Windows.

Dans un premier temps, nous allons devoir installer les dépendances suivantes :

- Windows ADK pour Windows 11 23H2
- Windows PE version Windows 11 23H2
- Installation du serveur MDT 2022

Une fois ces prérequis installés sur la machine Windows, nous allons pouvoir effectuer l'installation du service de déploiement. Dans un premier temps nous allons ouvrir la console de MDT puis cliquer sur **Créer un nouveau DeploymentShare** :

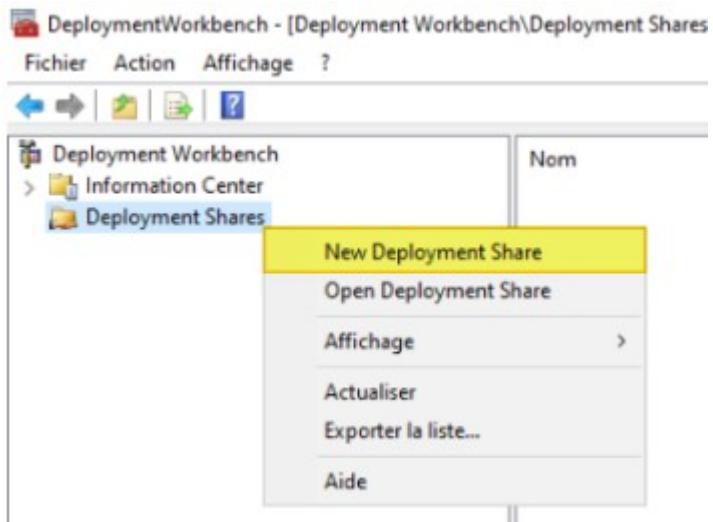


Figure X – Crédit d'un nouveau DeploymentShare

Une fois créer, notre DeploymentShare vas nous permettre de pouvoirs ajouter les OS à déployer, les applications à installer et les drivers (si présent) et intégrer. Pour le moment nous devons par la suite mettre en place un compte de service nous permettant de pouvoirs faire en sorte de lui données les droits nécessaires pour accéder et lire le DeploymentShare.

```
# Spécifier le nom et le mot de passe du compte de service
$ServiceAccountName = "Service_MDT"
$ServiceAccountPassword = ConvertTo-SecureString "123456789+aze@0" -AsPlainText -Force

# Créer le compte local
New-LocalUser $ServiceAccountName -Password $ServiceAccountPassword -FullName "MDT" -Description "Compte de service pour MDT"

# Ajouter les droits en lecture sur le partage
Grant-SmbShareAccess -Name "DeploymentShare" -AccountName "Service_MDT" -AccessRight Read -Force

# Attribuer au compte de service les permissions nécessaires pour accéder aux fichiers de déploiement MDT
$MDTSharePath = "\$env:COMPUTERNAME\DeploymentShare"
$Acl = Get-Acl $MDTSharePath
$Rule = New-Object System.Security.AccessControl.FileSystemAccessRule("Service_MDT", "ReadAndExecute", "ContainerInherit, ObjectInherit", "None", "Allow")
$Acl.SetAccessRule($Rule)
Set-Acl $MDTSharePath $Acl
```

Figure X – Ajout d'un compte utilisateur permettant d'accéder au DeploymentShare

Une fois ajouter, nous allons pouvoirs mettre en place une image Windows 11 dans MDT. Pour ce faire, nous allons nous rendre dans notre deploymentShare récemment créer et créer un nouveau dossier dans lequel nous allons importer un OS.

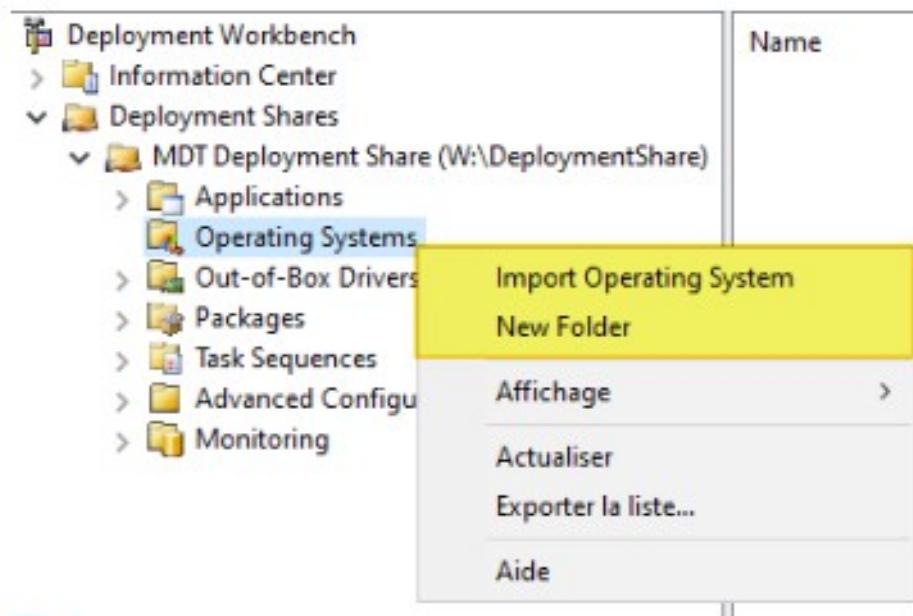


Figure X – Création et import d'un OS dans MDT

Une fois importer, nous allons pouvoirs créer une nouvelle « task sequence » qui vas nous permettre de pouvoirs configurer notre OS, quel OS doit être inclus dans la task séquence, les mots de passes, les compte etc... Cette dernière contient l'ensemble des tâches qui seront exécutées pendant le déploiement de la machine.

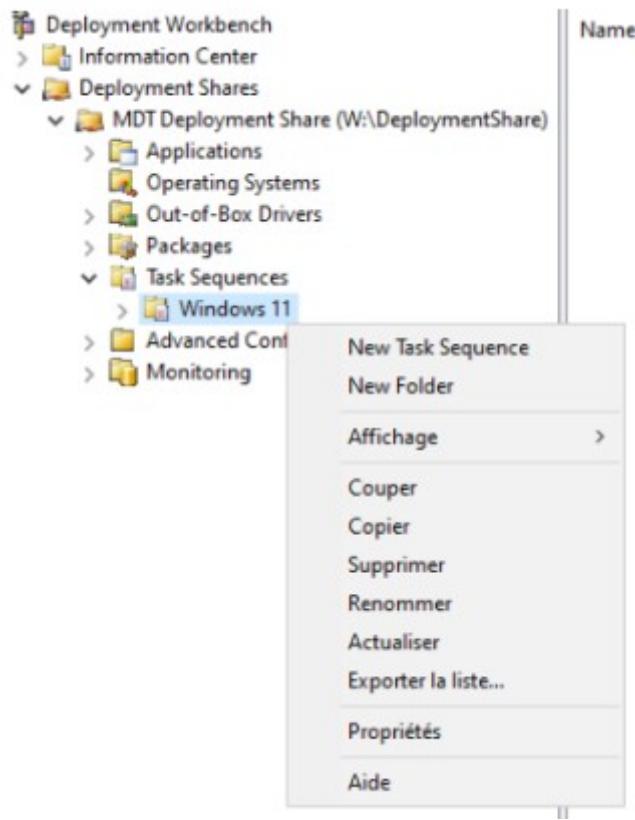


Figure X – Création d'une nouvelle task sequence

Une fois créé, nous allons nous rendre dans les propriété de la séquences de tâche qui vas nous donner la possibilité d'activer le paramètres suivants : « Windows Update » :

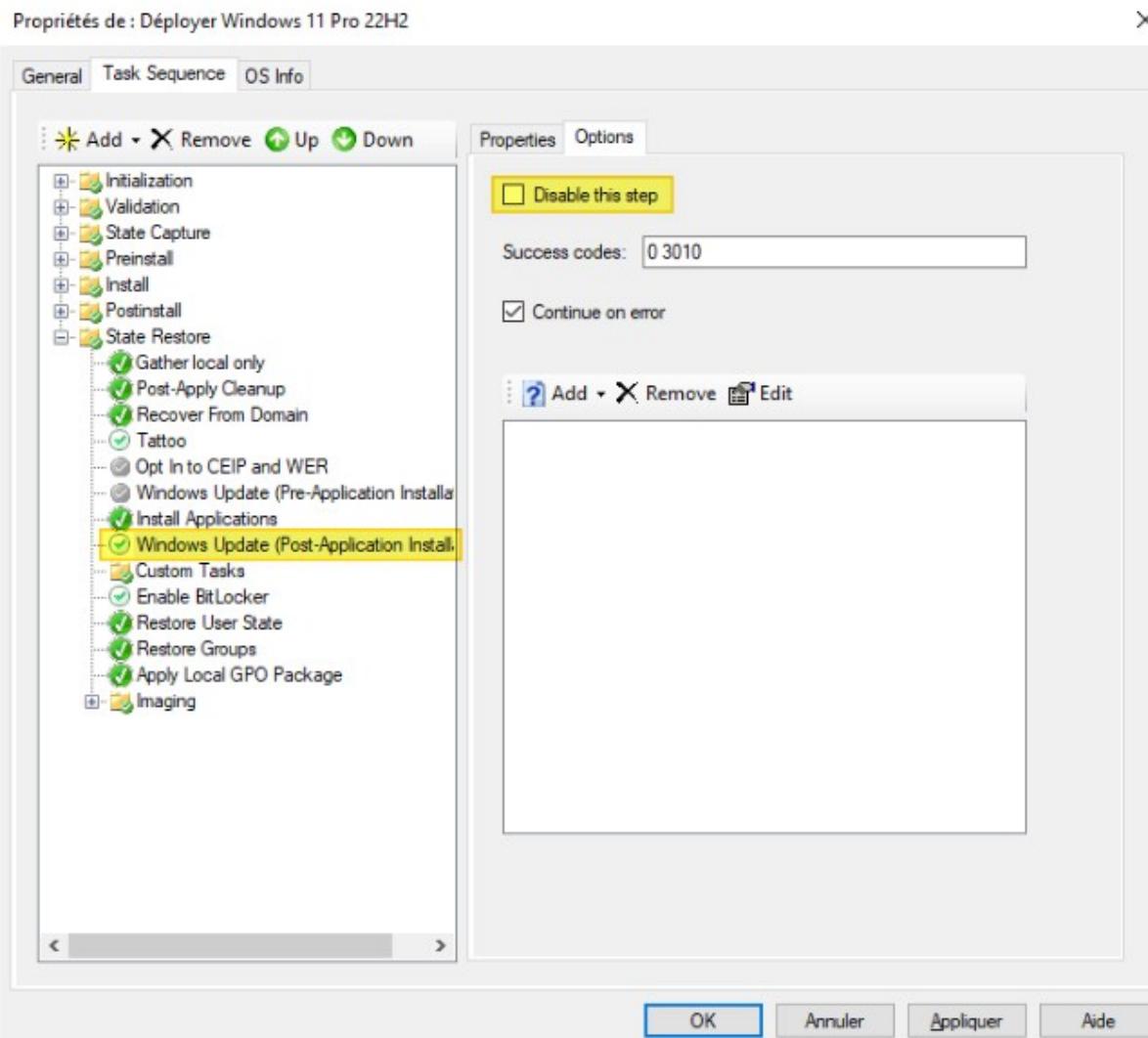


Figure X – Activation de Windows Update dans la séquence de tâche

Une fois cela fais, nous allons devoir créer un dossier vierge (comme sur la figure suivante) permettant de régler un bug dans MDT. En plus de cela, nous allons aussi dans les propriété du déploiement share devoirs activer uniquement X64 car Windows 11 ne support pas le 32 bits.

```
mkdir "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs"
```

Figure X – Crédit de la création du dossier vierge

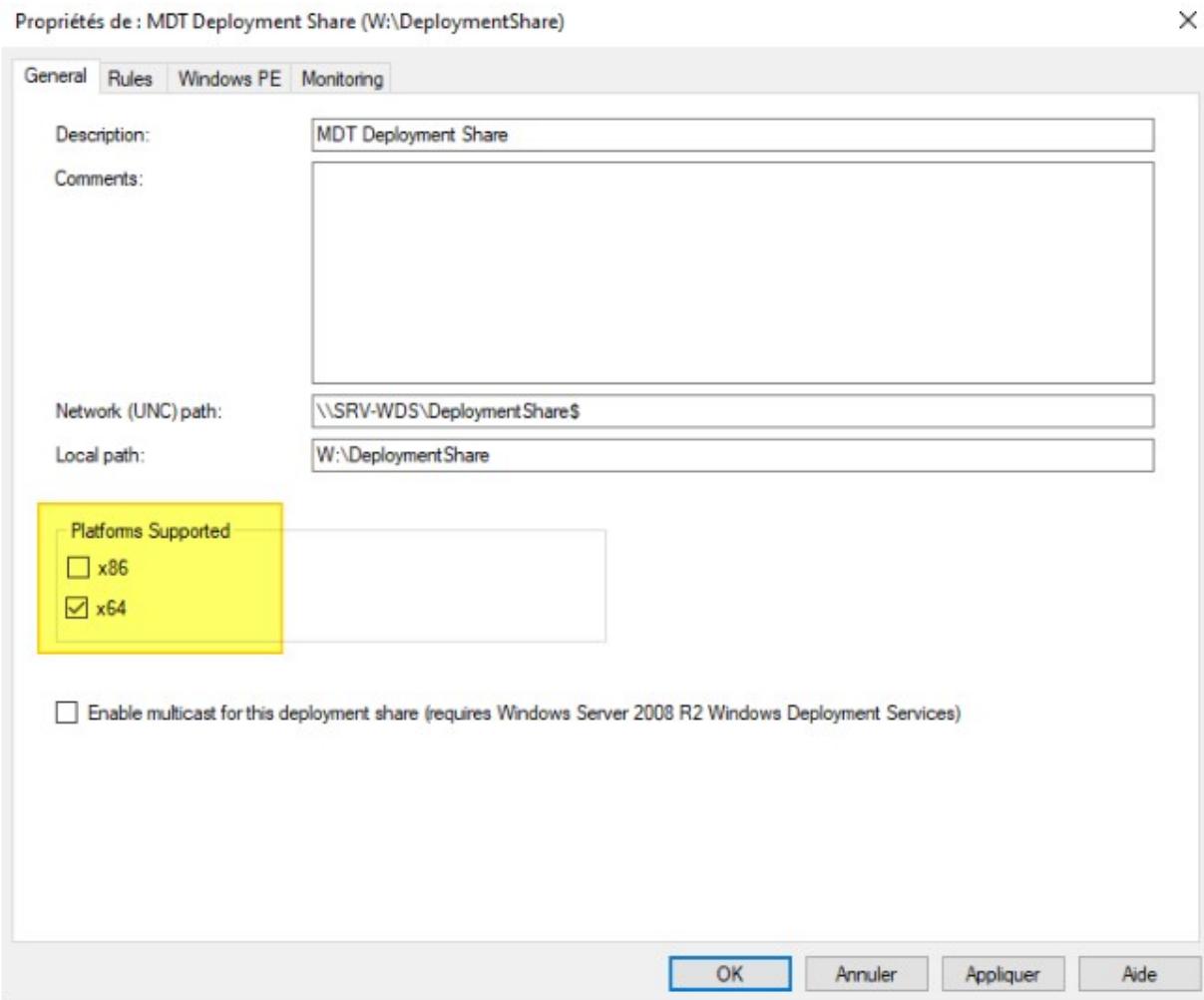


Figure X – Désactivation du support 32 bits du à Windows 11

Par la suite nous allons devoirs modifier le fichier se trouvant à l'adresse suivantes : « C:\Program Files\Microsoft Deployment Toolkit\Templates » portant le nom « Unattend_PE_x64 » et remplacer sont contenu par ceci :

*Unattend_PE_x64 - Bloc-notes

Fichier Edition Format Affichage Aide

```

<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="WindowsPE">
    <component name="Microsoft-Windows-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35">
      <Display>
        <ColorDepth>32</ColorDepth>
        <HorizontalResolution>1024</HorizontalResolution>
        <RefreshRate>60</RefreshRate>
        <VerticalResolution>768</VerticalResolution>
      </Display>
      <RunSynchronous>
        <RunSynchronousCommand wcm:action="add">
          <Description>Lite Touch PE</Description>
          <Order>1</Order>
          <Path>reg.exe add "HKLM\Software\Microsoft\Internet Explorer\Main" /t REG_DWORD</Path>
        </RunSynchronousCommand>
        <RunSynchronousCommand wcm:action="add">
          <Description>Lite Touch PE</Description>
          <Order>2</Order>
          <Path>wscript.exe X:\Deploy\Scripts\LiteTouch.wsf</Path>
        </RunSynchronousCommand>
      </RunSynchronous>
    </component>
  </settings>
</unattend>

```

Figure X – Patch Microsoft pour MDT Windows 11

Après avoir modifier le fichier, nous allons pouvoirs nous rendre dans les customsettings.ini de MDT et ajouter les éléments suivants en nous rendant dans les propriété du DeploymentShare :

```

[Settings]
Priority=Default
Properties=MyCustomProperty

[Default]
OSTInstall=Y
SkipCapture=NO
SkipAdminPassword=YES
SkipProductKey=YES
SkipComputerBackup=NO
SkipBitLocker=NO

_SMTSORGNAME=Networks.local

TimeZone=105
TimeZoneName=Romance Standard Time

```

Figure X – Customsettings.ini de l'entreprise Networks

La modification du Boostrap.ini est aussi une étape importante car c'est ce fichier qui se charge en premier lors du démarrage de l'environnement

WinPE, on va retrouver le chemin réseau vers le Deployment Share, ainsi que le nom de l'utilisateur et le mot de passe à utiliser pour se connecter à ce partage. Nous devrons ici mettre les informations du compte utilisateur créer plus tôt qui possède les droits de lecture sur le DeploymentShare (Service_MDT avec son mot de passe 123456789+aze@@).

```
[Settings]
Priority=Default

[Default]
DeployRoot=\\SRV-WDS\DeploymentShare$ 
UserID=Service_MDT
UserPassword=123456789+aze@@
UserDomain=SRVAD2025
SkipBDDWelcome=YES
KeyboardLocalePE=040c:0000040c
```

Figure X – Modification du Bootstrap.ini du DeploymentShare de Networks

Une fois toutes ces étapes de terminer, nous allons pouvoirs mettre à jour le DeploymentShare pour lui demander d'intégrer l'intégralité des changement que nous venons de réaliser sur le MDT. Dans l'interface de mise à jour veuillez sélectionner « Completely regenerate the boot image » permettant de recréer une image de zéro car il peut arriver que certains paramètres ne soit pas mis à jour et cause des erreurs.

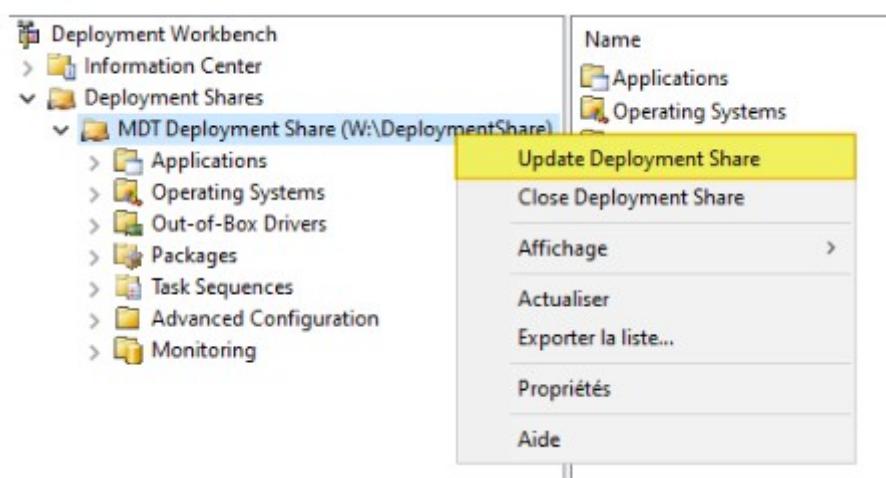


Figure X – Mise à jour du DeploymentShare pour intégrer les modifications

Nous allons pouvoir passer à la dernière étape qui est de renseigner l'image .wim que MDT nous a générée dans la console WDS.

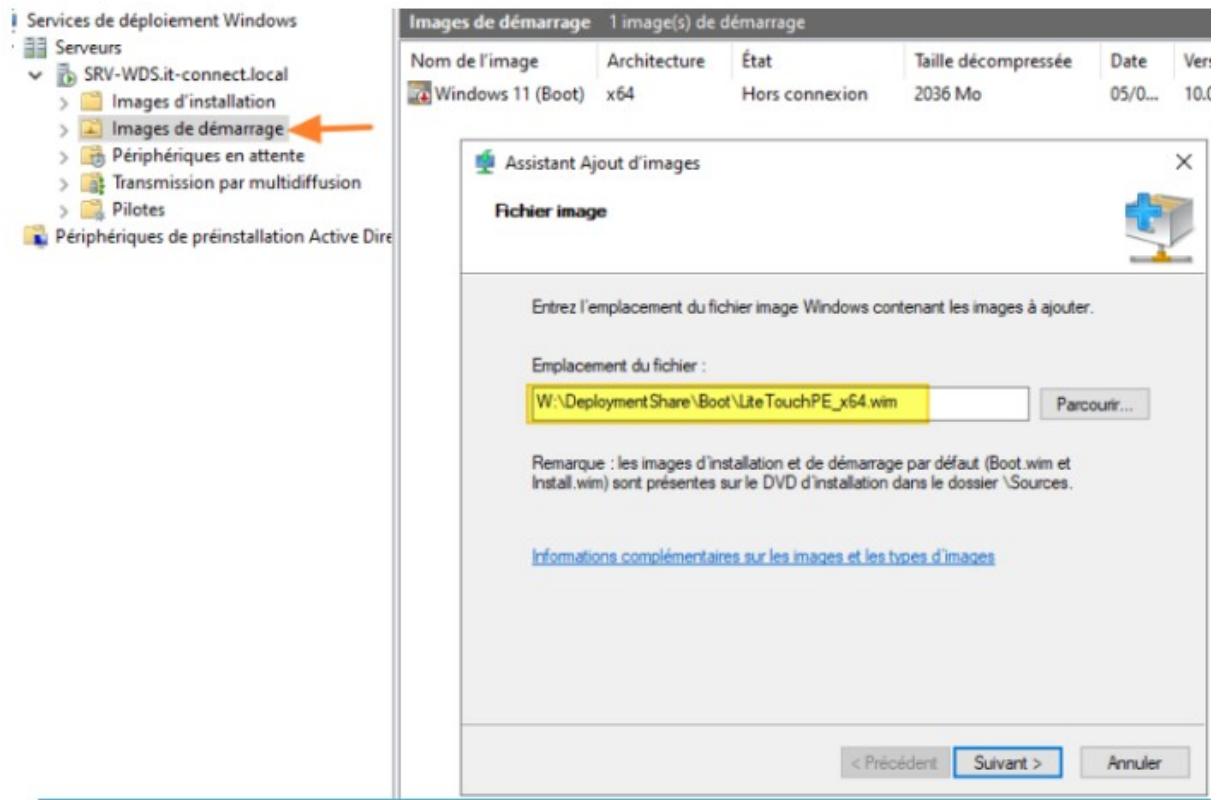


Figure X – Ajout de l'image aux images de démarrage de WDS

Une fois cela fait, nous pouvons tester le déploiement sur notre machine virtuelle et nous rendre compte que la machine prend bien une ip sur le DHCP, reçoit et communique avec le serveur WDS qui peut alors lui transmettre l'image MDT précédemment intégrer.

Une fois charger l'image nous transporte sur l'interface WinPE pour nous demander d'installer les éléments suivants :

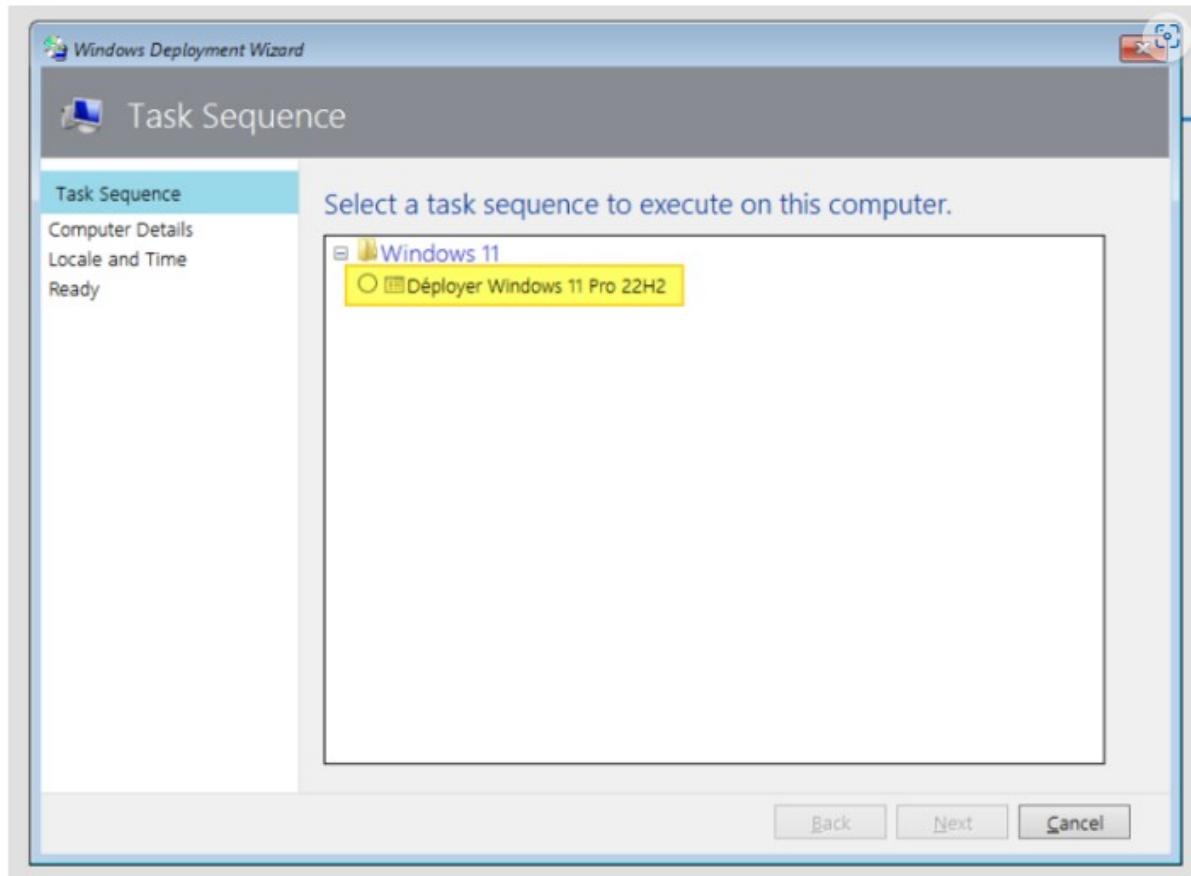


Figure X – Bon fonctionnement de WinPE et installation via LTI de l'image MDT

Ajout de fonctionnalités supplémentaires

Après avoirs installer WDS MDT, nous avons pu passer à l'étape suivante qui est de personnaliser notre DeploymentShare, pour cela nous avons intégrer un déploiement automatique d'agents Zabbix nous permettant de pouvoirs superviser l'ensemble de notre parc. Nous avons aussi intégrer un système de déploiement hors ligne via la console MDT nous permettant de créer des images ISO.

Implémentation d'application lors du déploiement

Intégration et export de l'image MDT en ISO

Mise en place d'une base de données SQL

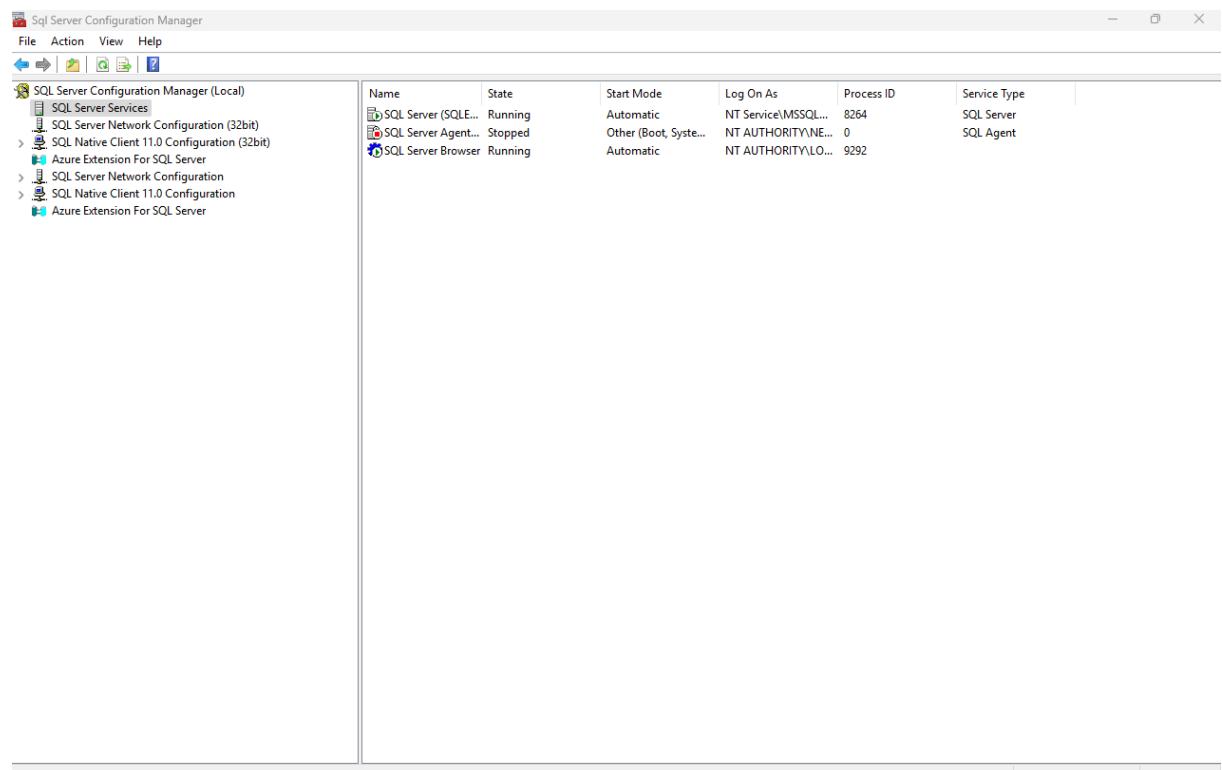


Figure X – Installation du serveur SQL Express et activation des services pour mdt

Microsoft Deployment Toolkit (MDT) permet de déployer des systèmes d'exploitation de manière automatisée et personnalisée. Pour aller plus loin dans la personnalisation, MDT peut utiliser une base de données SQL Server afin de stocker et centraliser des informations spécifiques sur les machines à déployer (nom d'ordinateur, rôle, emplacement, etc.).

Pour que cette fonctionnalité fonctionne, il est indispensable d'activer certains services SQL Server.

SQL Server (SQLEXPRESS)

Ce service correspond au moteur de base de données SQL Server. Il est indispensable au fonctionnement de la base de données utilisée par MDT. Lorsque l'on crée la base MDT via l'assistant du Deployment Workbench, ce service doit être démarré. Par la suite, lors des déploiements, MDT interrogera cette base pour récupérer les paramètres liés aux machines.

SQL Server Browser

Ce service permet aux applications clientes de découvrir automatiquement les instances SQL Server sur le réseau, en particulier lorsque l'instance utilisée n'est pas l'instance par défaut (comme c'est le cas pour "SQLEXPRESS"). Dans le cadre de MDT, ce service facilite la connexion entre le serveur MDT et l'instance SQL.

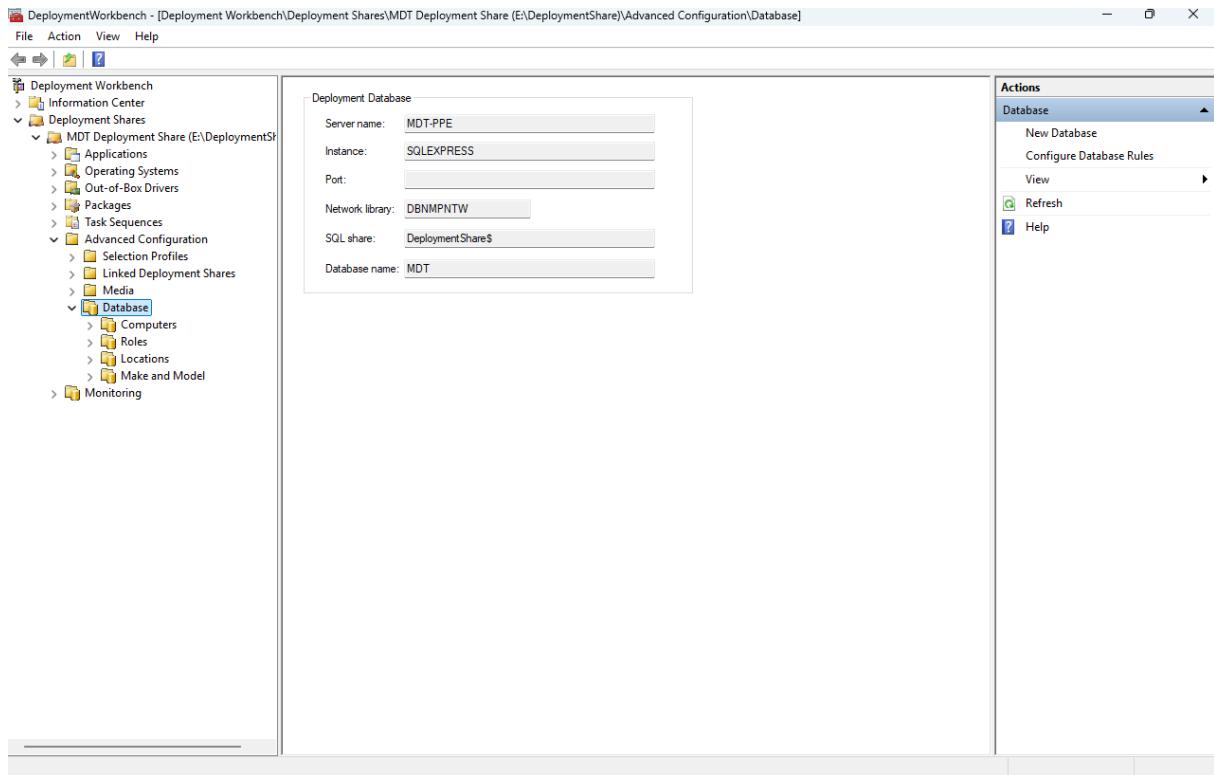


Figure X – Configuration de la BDD sur MDT

Après avoir procédé avec succès à l'installation du serveur **SQL Server Express**, nous avons configuré la base de données MDT directement depuis la console d'administration **Deployment Workbench**, fournie avec **Microsoft Deployment Toolkit (MDT)**.

Cette étape permet de créer une base SQL nommée par défaut MDT, qui servira à centraliser et personnaliser les déploiements. La configuration s'effectue via un assistant accessible dans le menu "**Advanced Configuration**" > "**Database**" de l'interface MDT. Celui-ci permet de spécifier :

- L'instance SQL à utiliser (par exemple : .\SQLEXPRESS),
- Le nom de la base à créer,
- Les options d'authentification.

Une fois la configuration terminée, MDT est capable d'utiliser cette base pour attribuer automatiquement des paramètres spécifiques aux machines lors des déploiements (nom d'hôte, rôle, paramètres réseau, applications, etc.).

Cette intégration permet d'**automatiser et personnaliser en profondeur le processus de déploiement**, en fonction de critères comme l'adresse MAC ou l'emplacement physique des postes.

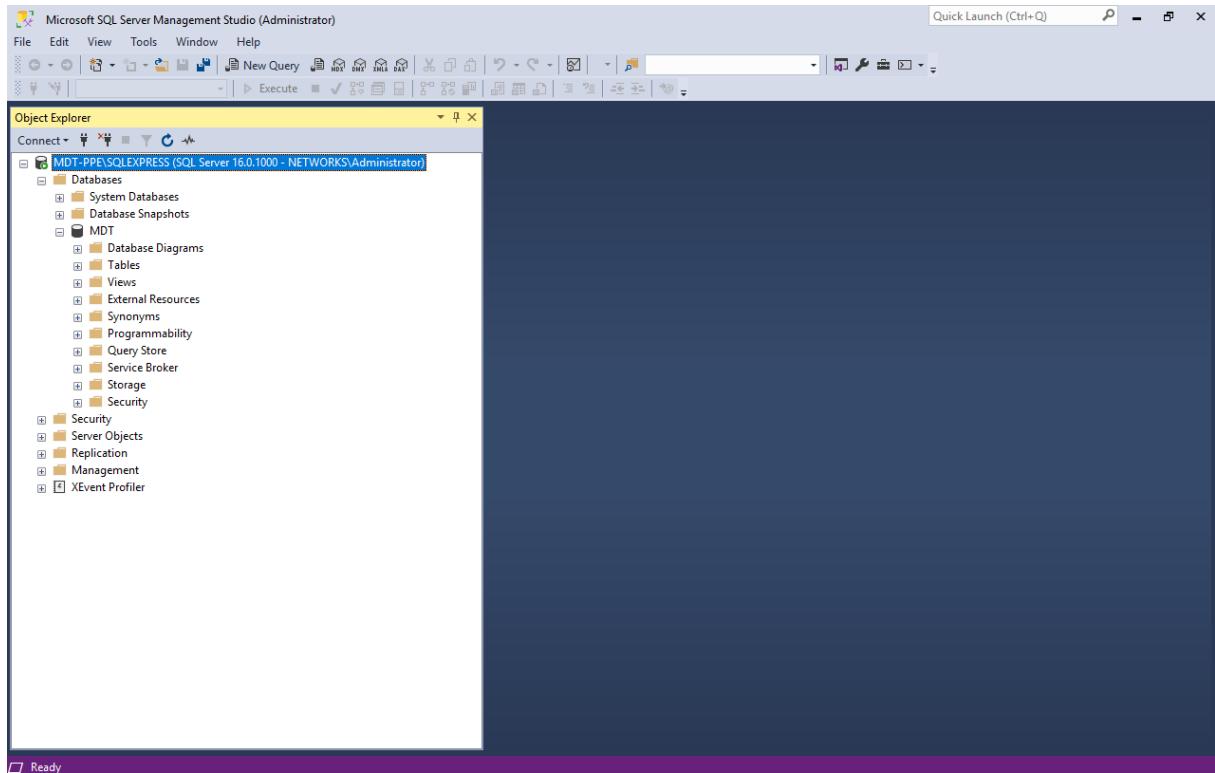


Figure X – Visualisation de la bdd sur le serveur SQL Express

La base de données **MDT**, visible ici dans SQL Server Management Studio (SSMS), est utilisée par **Microsoft Deployment Toolkit** pour **personnaliser les déploiements d'images système**. Elle permet, par exemple, d'attribuer automatiquement un nom d'ordinateur, une configuration réseau, un rôle ou un emplacement à une machine en fonction de son adresse MAC, son UUID ou d'autres critères.

Sources

Pour la création et l'utilisation d'image, nous avons utiliser le site d'IT-connect nous permettant de réaliser une installation propre des services. Leurs documentations est retrouvable ici :

[Partie 1 – DHCP](#)

[Partie 2 – WDS-MDT](#)