

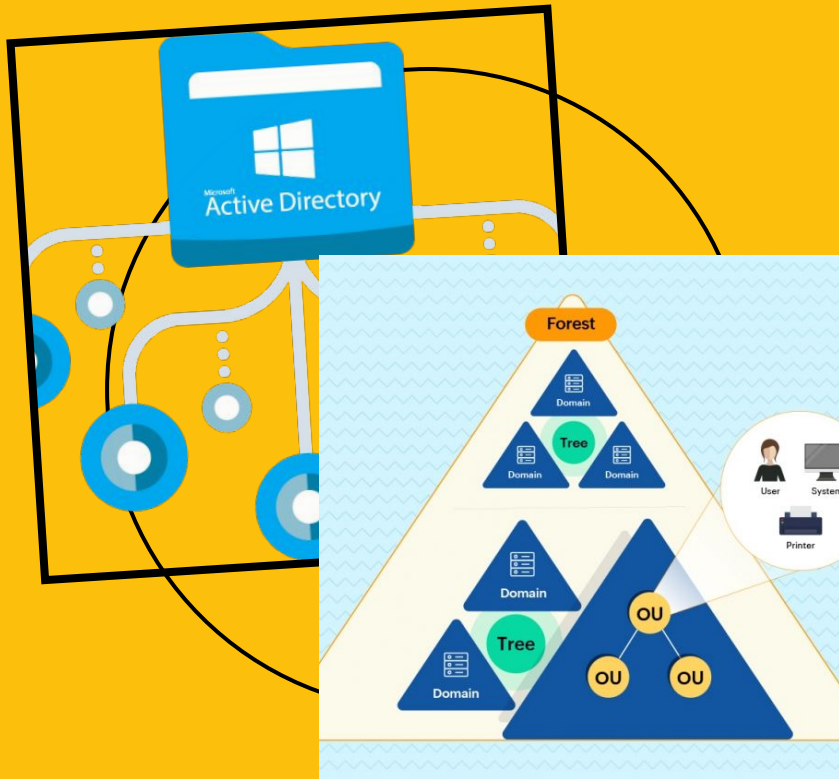
Company profile



MOLL NANS



Alla Kaid Ahmed



SOMMAIRE

- ✦ ACTIVE DIRECTORY ?
- ✦ DIAGRAMME ACTIVE DIRECTORY
- ✦ INSTALLATION ET CONFIGURATION
- ✦ CONNECTION DEPUIS UNE AUTRE VM

An illustration of a person with dark skin and short dark hair, wearing a yellow t-shirt with a white wavy pattern. They have their arms raised in a questioning gesture. Above their head is a grey speech bubble containing a white question mark, and another faint question mark is visible in the background. The entire illustration is set within a white circular frame on a yellow background.

ACTIVE DIRECTORY ?

ACTIVE DIRECTORY (AD) EST UN SERVICE D'ANNUAIRE DÉVELOPPÉ PAR MICROSOFT, INTÉGRÉ AUX SYSTÈMES D'EXPLOITATION WINDOWS SERVER. IL FOURNIT UN ENSEMBLE DE SERVICES DE RÉPERTOIRE POUR LES RÉSEAUX D'ENTREPRISE, PERMETTANT LA GESTION CENTRALISÉE DES RESSOURCES, DES UTILISATEURS, DES ORDINATEURS ET D'AUTRES OBJETS RÉSEAU.



1

contrôleur de domaine

SERVEURS EXÉCUTANT ACTIVE DIRECTORY, STOCKANT LA BASE DE DONNÉES.

2

forêt et domaines

FORÊTS REGROUPANT DES DOMAINES PARTAGEANT UNE STRUCTURE D'ANNUAIRE COMMUNE.

3

Unités d'organisation

SUBDIVISIONS D'UN DOMAINE POUR UNE GESTION PLUS GRANULAIRE.

4

Objets

Utilisateurs, groupes, ordinateurs, imprimantes, ressources partagées, etc.

Fonctionnalités et Avantages

fonctionnalité

- Gestion des ressources
- Politiques de sécurité
- Intégration avec Microsoft
- Réplication des données

Avantages

- Centralisation
- Sécurité
- Interopérabilité
- Évolutivité

Inconvénients et Cas Pratiques



Inconvénients

Complexité de mise en place et configuration.
Coûts de licence et maintenance. Dépendance
à Microsoft.



Cas Pratiques

Gestion des utilisateurs et groupes.
Authentification unique. Politiques de groupe.
Intégration avec services Microsoft.

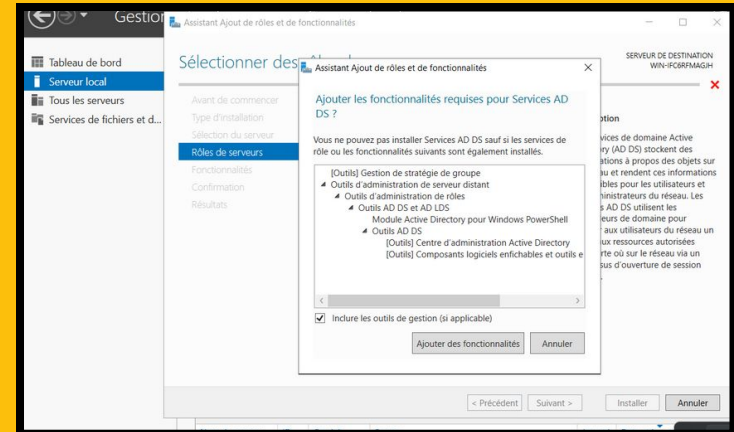
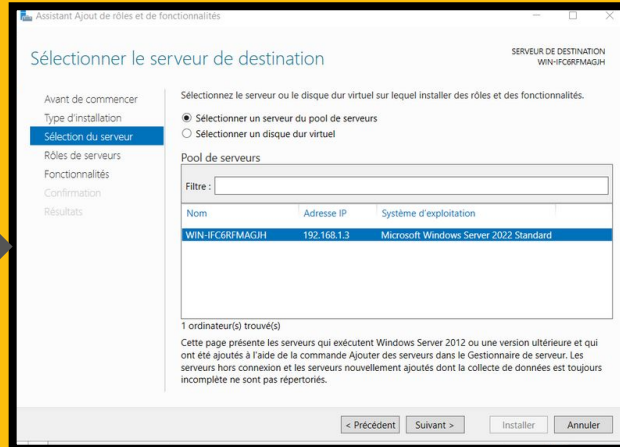
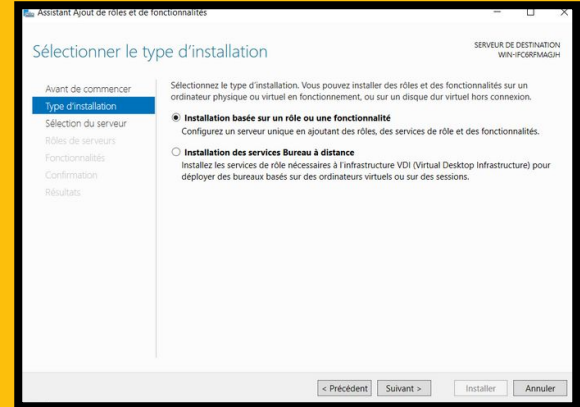
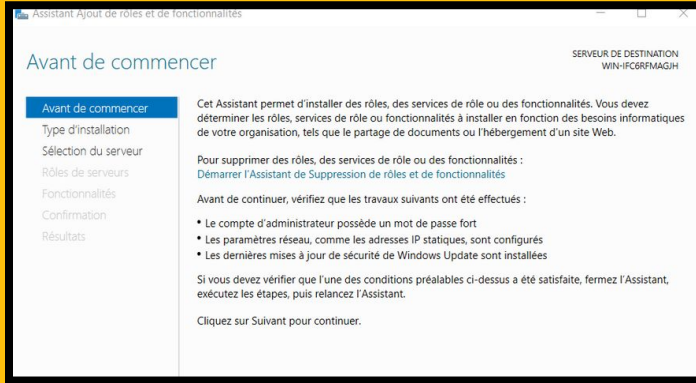
tableau representatif

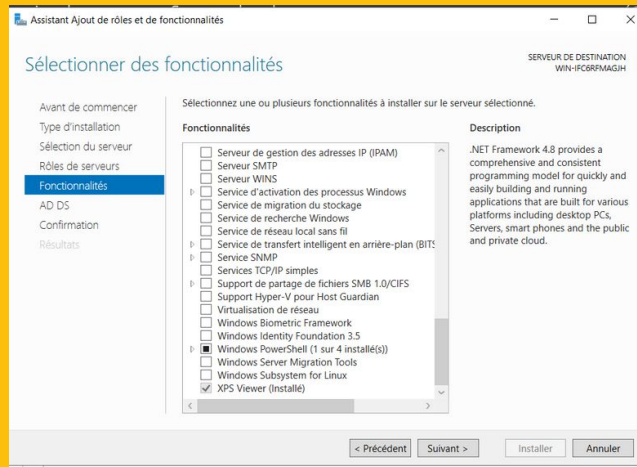
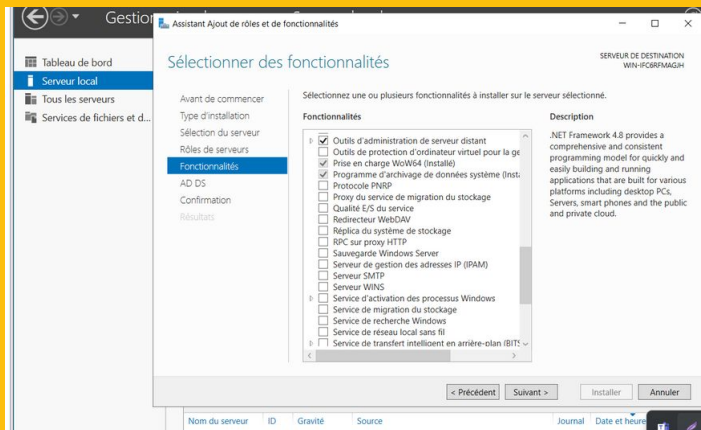
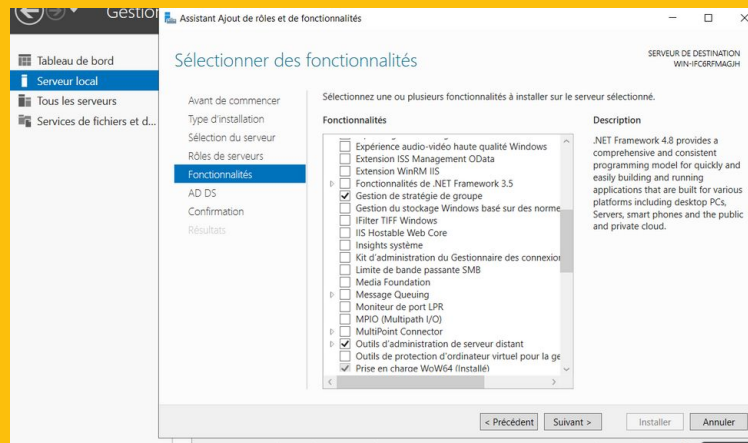
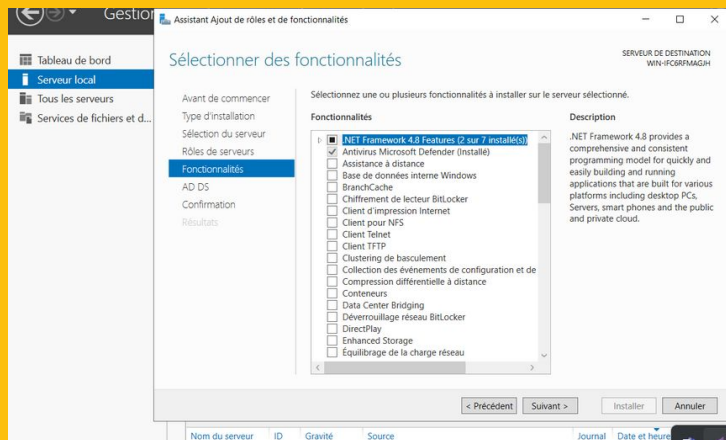
Dossier et File				
le Goupe	file-transmission	file_petit_versement	file_grand_versement (der application	
Catégorie A	controle total	non	non	non
Catégorie B	non	oui control total	non	non
Catégorie C	non	oui	oui	
Catégorie D + Expert-comptable	oui control total	oui control total	oui control total	oui control total
Chefs de projet	non	non	non	oui
Clients	non	non	non	non
Prestataires + informaticiens	lecture et execution	lecture et execution	lecture et execution	lecture et execution

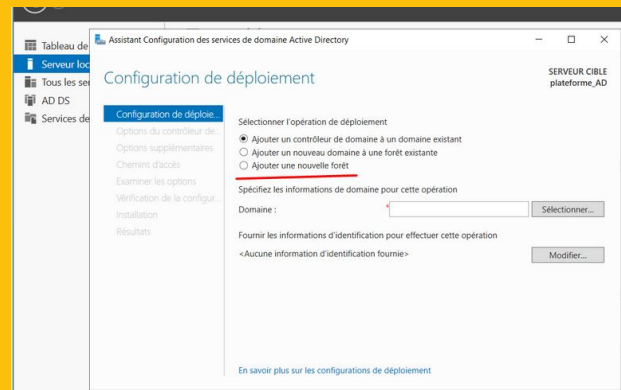
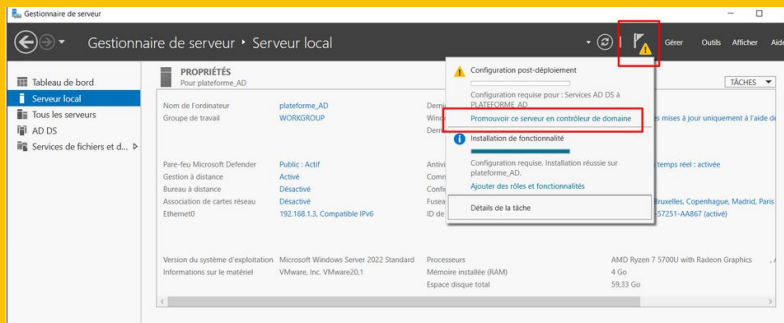
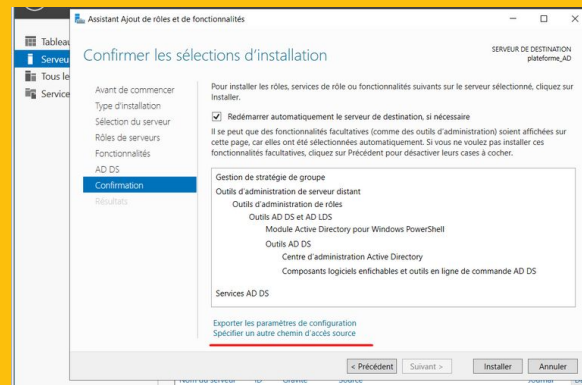
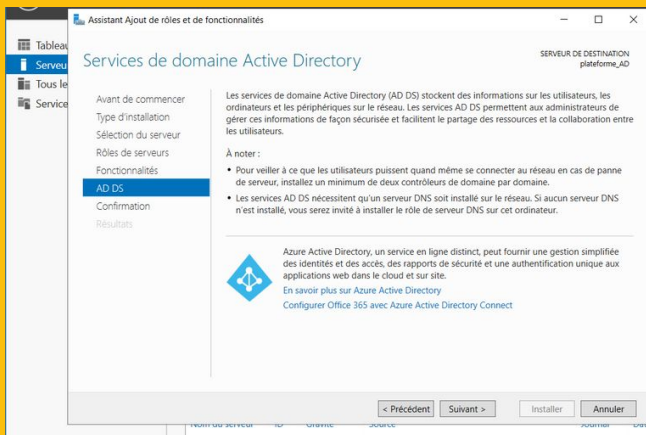
Groupe	Utilisateur	
Catégorie A	Amélie Lotte	
Catégorie A	Arthur Neutron	
Catégorie B	Bruno Desange	
Catégorie B	Bob Voulet	
Catégorie C	Céline Stoner	
Catégorie C	Cyril Potet	
Catégorie D	Dorian Matias	
Catégorie D	Bernard Taperio	
clients	Karim	
clients	Césarine Cordonnier	
fonction	Groupe	
Les guichetiers	Catégorie A	
Les conseillers	Catégorie B	
Les gestionnaires et conseillers	Catégorie C	
Le personnel dirigeant	Catégorie D	
Les clients	clients	
Les informaticiens et les prestataires	Prestataires + informaticiens	
Les chefs de projet	Les chefs de projet	

INSTALLATION ET CONFIGURATION ACTIVE

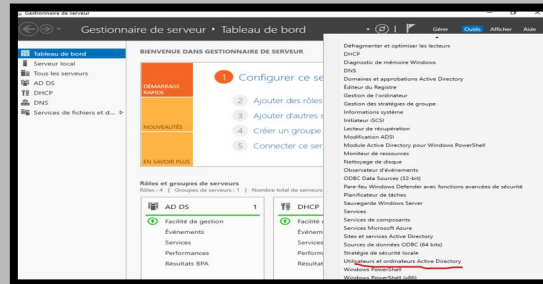
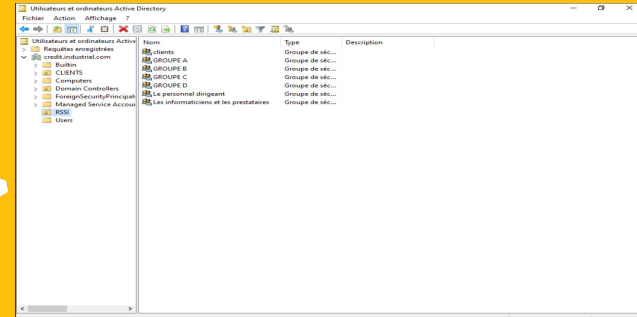
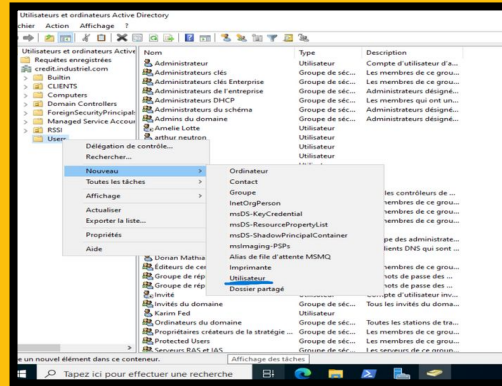
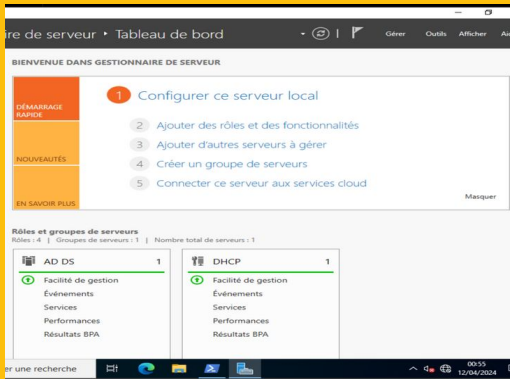
DIRECTORY



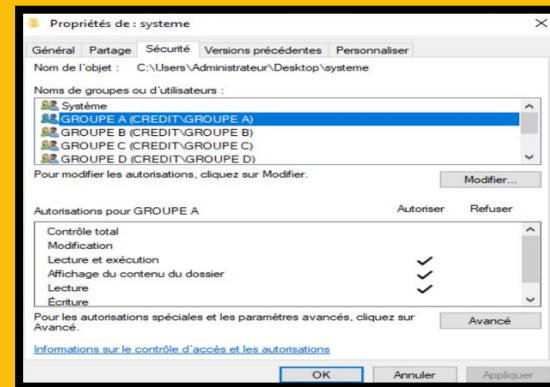
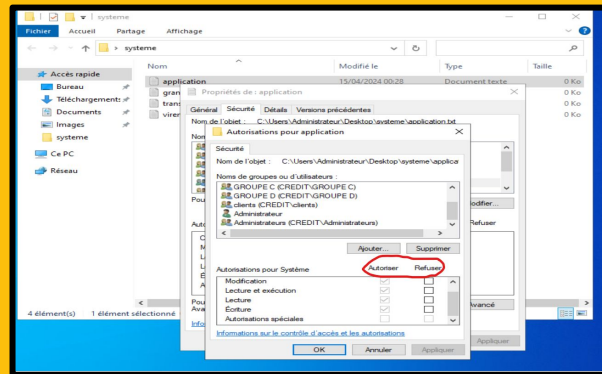
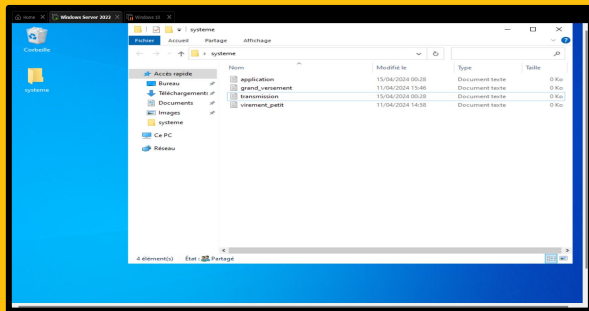




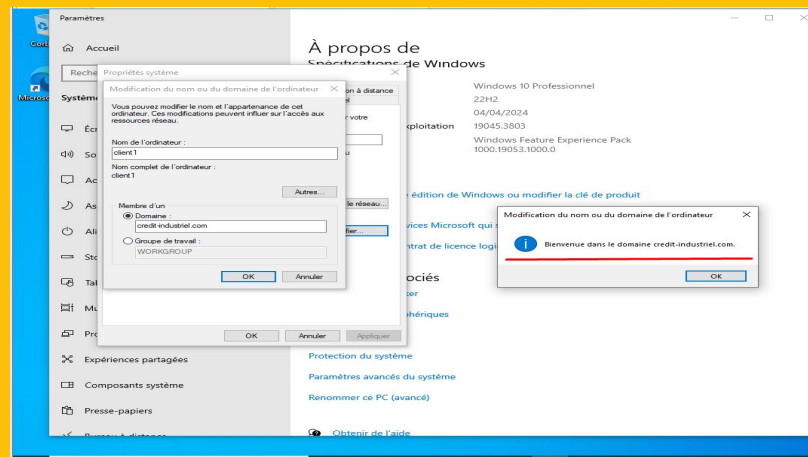
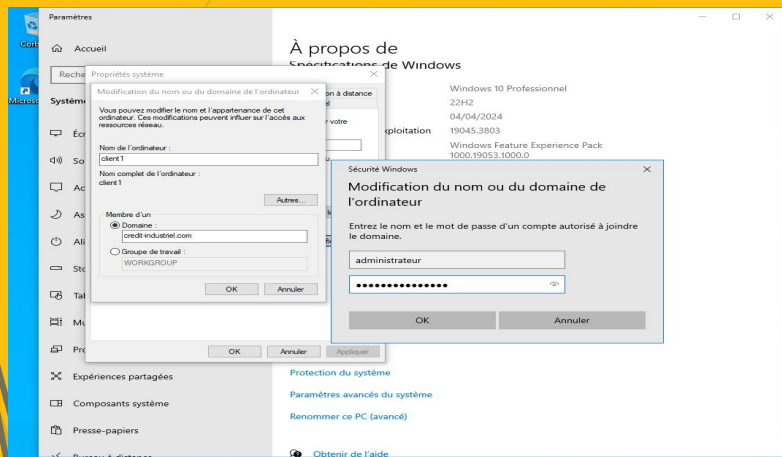
creation des groupes et des utilisateurs



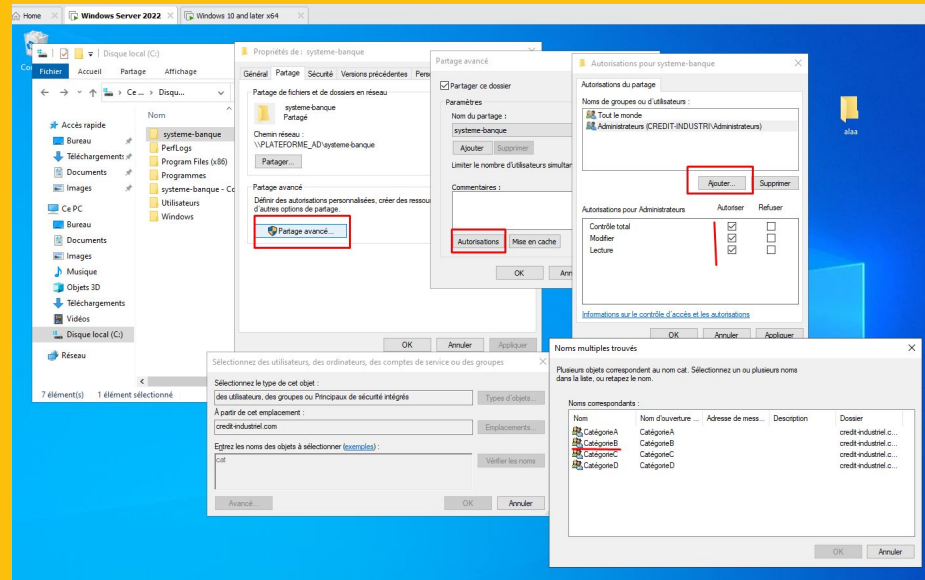
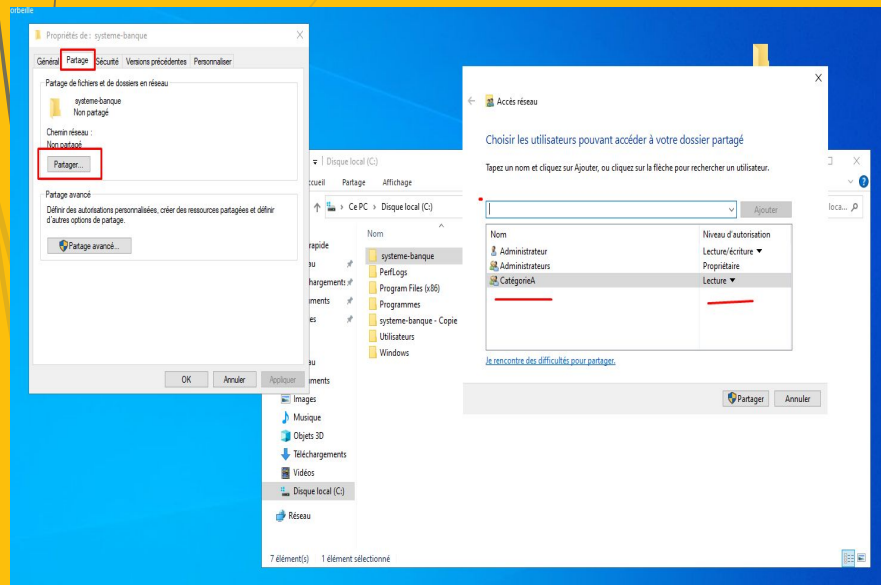
CRÉATIONS DU SYSTÈME ET SÉCURITÉ



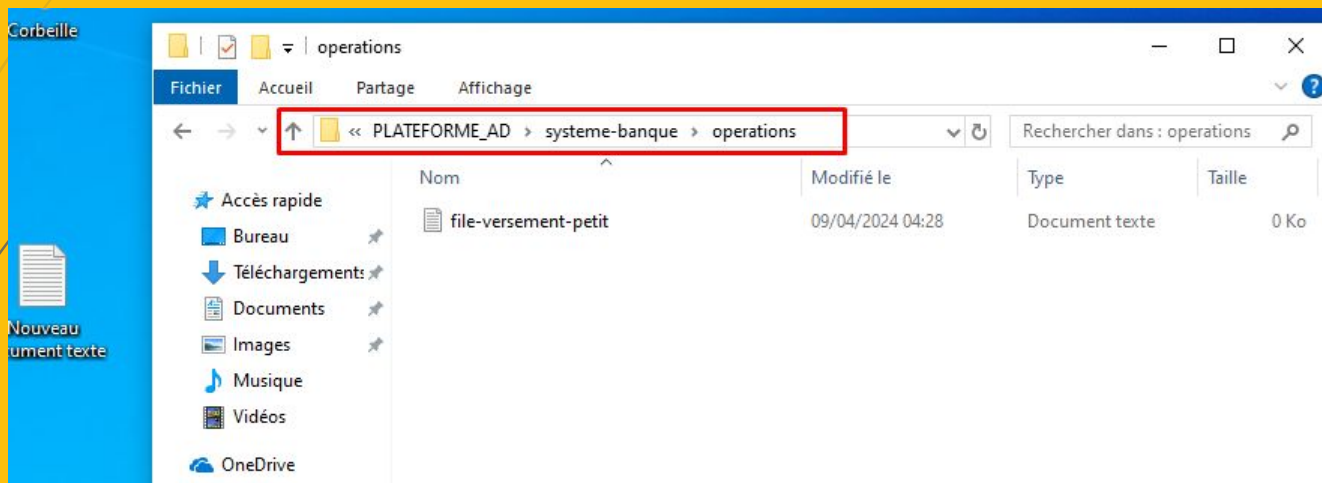
Faire joindre une machine au Domain



Partage des dossier



Accéder au dossier partagé depuis la machine du client



Permission NTFS et de partage

si une permission de partage est plus restrictive que celle de NTFS, la permission de partage prévaudra. Cependant, si la permission de partage est plus permissive que celle de NTFS, la permission de NTFS sera appliquée.

Configuration du partage	Configuration NTFS	Permission résultante
Lecture seule	Lecture et écriture	Lecture seule
Lecture	Lecture et écriture	Lecture seule
Écriture	Lecture seule	Lecture seule
Contrôle total	Lecture seule	Lecture seule
Contrôle total	Lecture et écriture	Lecture seule
Aucun accès	Lecture seule	Aucun accès

Récupération des comptes inactifs

Spécifiez le nombre de jours d'inactivité que vous souhaitez
rechercher
\$nombreJoursInactivité = 20

Obtenez la date actuelle
\$dateActuelle = Get-Date

Calculez la date limite d'inactivité
\$dateLimite = \$dateActuelle.AddDays(-\$nombreJoursInactivité)

Récupérez les comptes d'utilisateurs inactifs dans Active Directory
\$comptesInactifs = Get-ADUser -Filter {LastLogonTimeStamp -lt
\$dateLimite} -Properties LastLogonTimeStamp | Where-Object {
\$_.Enabled -eq \$true }

Affichez les comptes d'utilisateurs inactifs
\$comptesInactifs | Select-Object Name, SamAccountName, Enabled,
LastLogonTimeStamp

```

PS C:\> # Spécifiez le nombre de jours d'inactivité que vous souhaitez rechercher
PS C:\> $nombreJoursInactivité = 0
PS C:\>
PS C:\> # Obtenez la date actuelle
PS C:\> $dateActuelle = Get-Date
PS C:\>
PS C:\> # Calculez la date limite d'inactivité
PS C:\> $dateLimite = $dateActuelle.AddDays(-$nombreJoursInactivité)
PS C:\>
PS C:\> # Récupérez les comptes d'utilisateurs inactifs dans Active Directory
PS C:\> $comptesInactifs = Get-ADUser -Filter {LastLogonTimeStamp -lt $dateLimite} -Properties LastLogonTimeStamp | Where-Obj
ect { $_.Enabled -eq $true }
PS C:\>
PS C:\> # Créez le chemin du fichier de sortie
PS C:\> $fichierSortie = "C:\resultats_utilisateurs_inactifs.csv"
PS C:\>
PS C:\> # Exportez les comptes d'utilisateurs inactifs vers un fichier CSV
PS C:\> $comptesInactifs | Select-Object Name, SamAccountName, Enabled, LastLogonTimeStamp | Export-Csv -Path $fichierSortie
-NotypeInformation
PS C:\>
PS C:\> # Affichez un message pour confirmer l'exportation
PS C:\> Write-Host "Les résultats ont été exportés vers : $fichierSortie"
Les résultats ont été exportés vers : C:\resultats_utilisateurs_inactifs.csv
PS C:\>

```

resultats_utilisateurs_inactifs - Bloc-notes

Fichier Edition Format Affichage Aide

```

"Name","SamAccountName","Enabled","LastLogonTimeStamp"
"Am?lie Lotte","ALotte","True","133570067406859013"
"Bob Voulet","BVoulet","True","133570099966625860"
"Administrateur","Administrateur","True","133575602067115186"

```

Identifications des doublons

```
# Récupérer tous les utilisateurs d'Active Directory
```

```
$utilisateurs = Get-ADUser -Filter *
```

```
# Créer un tableau pour stocker les noms de famille déjà rencontrés
```


```
$nomsFamille = @()
```

```
# Créer un tableau pour stocker les doublons
```

```
$doublons = @()
```

```
# Parcourir tous les utilisateurs
```

```
foreach ($utilisateur in $utilisateurs) {
```



```
# Vérifier si le nom de famille est déjà dans le
tableau
if ($nomsFamille -contains $utilisateur.Surname) {
    # Ajouter le nom de famille à la liste des
    doublons
    $doublons += $utilisateur.Surname
} else {
    # Ajouter le nom de famille au tableau des
    noms de famille
    $nomsFamille += $utilisateur.Surname
}
}
```

```
# Afficher les doublons
if ($doublons.Count -gt 0) {
    Write-Host "Doublons de noms de famille trouvés
:"
    $doublons
} else {
    Write-Host "Aucun doublon de nom de famille
trouvé."
}
```

```
×
▼
Home x Windows Server 2022 x Windows 10 and later x64 x
Administrateur : Windows PowerShell
PS C:\> # Récupérer tous les utilisateurs d'Active Directory
PS C:\> $utilisateurs = Get-ADUser -Filter *
PS C:\> # Créer un tableau pour stocker les noms de famille déjà rencontrés
PS C:\> $nomsFamille = @()
PS C:\> # Créer un tableau pour stocker les doublons
PS C:\> $doublons = @()
PS C:\> # Parcourir tous les utilisateurs
PS C:\> foreach ($utilisateur in $utilisateurs) {
>>     # Vérifier si le nom de famille est déjà dans le tableau
>>     if ($nomsFamille -contains $utilisateur.Surname) {
>>         # Ajouter le nom de famille à la liste des doublons
>>         $doublons += $utilisateur.Surname
>>     } else {
>>         # Ajouter le nom de famille au tableau des noms de famille
>>         $nomsFamille += $utilisateur.Surname
>>     }
>> }
PS C:\>
PS C:\> # Afficher les doublons
PS C:\> if ($doublons.Count -gt 0) {
>>     Write-Host "Doublons de noms de famille trouvés :"
>>     $doublons
>> } else {
>>     Write-Host "Aucun doublon de nom de famille trouvé."
>> }
Doublons de noms de famille trouvés :
kaid
PS C:\>
PS C:\>
```



Alerte quand il y a des connexions après certaines heures

```
# Spécifiez les heures après lesquelles on souhaite générer des alertes
$heures_cibles = 17..23 # Heures de 17h à 23h (5PM à 11PM)


# Répertoire où enregistrer les alertes
$chemin_dossier_alertes = "C:\Alertes"

# Créer le dossier s'il n'existe pas
If (-not (Test-Path $chemin_dossier_alertes)) {
    New-Item -ItemType Directory -Path $chemin_dossier_alertes | Out-Null
}

# Nom du fichier d'alerte
$фichier_alerte = Join-Path -Path $chemin_dossier_alertes -ChildPath
"Alerte_Connexion_APDC_$(Get-Date -Format 'yyyyMMdd_HHmms').txt"
```



```
# Rechercher les connexions récentes depuis la dernière
heure spécifiée
$heure_actuelle = Get-Date
$heure_cible_precedente = $heure_actuelle.AddHours(-1)
# Rechercher les événements de connexion dans les
journaux de sécurité de tous les contrôleurs de domaine
$evenements_connexion = Get-WinEvent -FilterHashtable @{
    LogName = 'Security'
    ID = 4624 # ID de l'événement de connexion
    StartTime = $heure_cible_precedente
} -ComputerName (Get-ADDomainController -Filter *).Name
# Filtrer les événements de connexion qui ont eu lieu après
les heures cibles
$evenements_connexion_apres_heures =
$evenements_connexion | Where-Object {
    $heure_connexion = $_.TimeCreated
    $heure_connexion.Hour -in $heures_cibles
}
```



```
# Si des connexions ont été trouvées après les
heures cibles, générer une alerte
if ($evenements_connexion_apres_heures) {
    # Construire le contenu de l'alerte
    $contenu_alerte = "Des connexions ont été
détectées après les heures cibles dans Active
Directory.`n"
    $contenu_alerte += "Nombre de connexions
détectées :
$($evenements_connexion_apres_heures.Count)`n
"

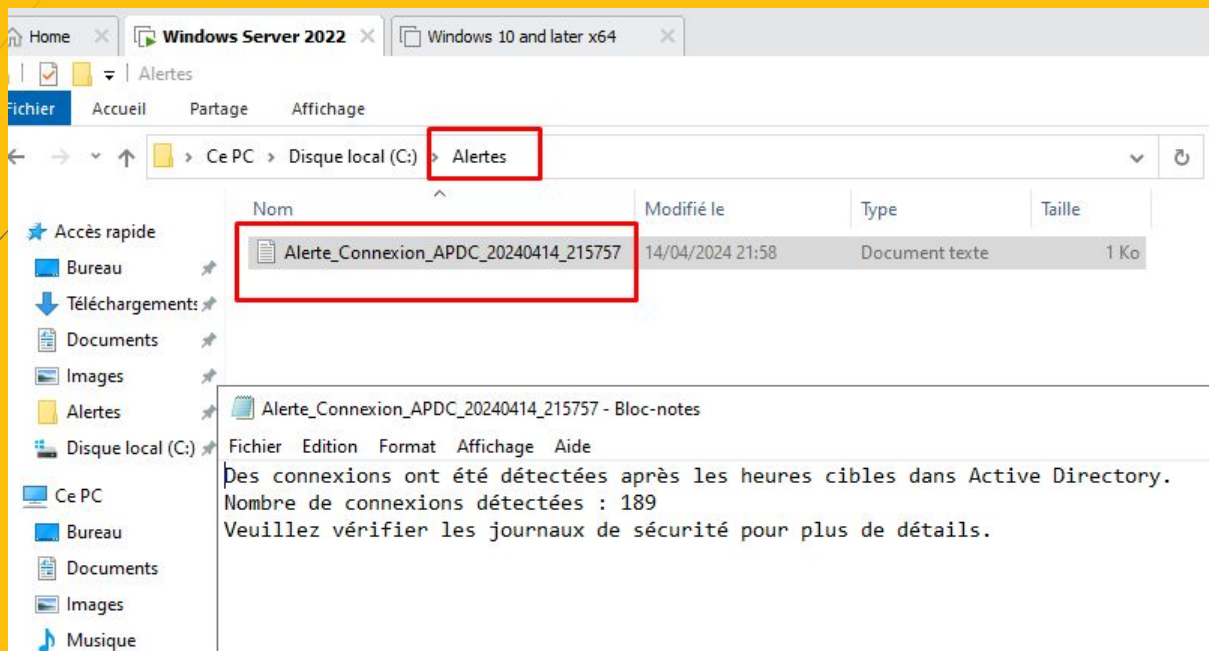
    $contenu_alerte += "Veuillez vérifier les journaux
de sécurité pour plus de détails.`n"

    # Enregistrer l'alerte dans un fichier
    $contenu_alerte | Out-File -FilePath
$fichier_alerte -Encoding UTF8

    Write-Host "Alerte enregistrée dans
$fichier_alerte"
}
```




```
Administrateur : Windows PowerShell
>> }
PS C:\>
PS C:\> # Nom du fichier d'alerte
PS C:\> $fichier_alerte = Join-Path -Path $chemin_dossier_alertes -ChildPath "Alerte_Connexion_APDC_$(Get-Date -Format 'yyyyM
Mdd_HH:mm:ss').txt"
PS C:\>
PS C:\> # Recherchez les connexions récentes depuis la dernière heure spécifiée
PS C:\> $heure_actuelle = Get-Date
PS C:\> $heure_cible_precedente = $heure_actuelle.AddHours(-1)
PS C:\>
PS C:\> # Recherchez les événements de connexion dans les journaux de sécurité de tous les contrôleurs de domaine
PS C:\> $evenements_connexion = Get-WinEvent -FilterHashtable @{
>>     LogName = 'Security'
>>     ID = 4624 # ID de l'événement de connexion
>>     StartTime = $heure_cible_precedente
>> } -ComputerName (Get-ADDomainController -Filter *).Name
PS C:\>
PS C:\> # Filtrer les événements de connexion qui ont eu lieu après les heures cibles
PS C:\> $evenements_connexion_apres_heures = $evenements_connexion | Where-Object {
>>     $heure_connexion = $_.TimeCreated
>>     $heure_connexion.Hour -in $heures_cibles
>> }
PS C:\>
PS C:\> # Si des connexions ont été trouvées après les heures cibles, générer une alerte
PS C:\> if ($evenements_connexion_apres_heures) {
>>     # Construire le contenu de l'alerte
>>     $contenu_alerte = "Des connexions ont été détectées après les heures cibles dans Active Directory.`n"
>>     $contenu_alerte += "Nombre de connexions détectées : $($evenements_connexion_apres_heures.Count)`n"
>>     $contenu_alerte += "Veuillez vérifier les journaux de sécurité pour plus de détails.`n"
>>
>>     # Enregistrer l'alerte dans un fichier
>>     $contenu_alerte | Out-File -FilePath $fichier_alerte -Encoding UTF8
>>
>>     Write-Host "Alerte enregistrée dans $fichier_alerte"
>> }
```




Alerte lorsque plus de 3 modifications sont faites dans une journée sur un fichier

```
# Chemin du fichier à surveiller
$fileToMonitor = "C:\
systeme-banque\operations\file-versement-grand.txt"
# Nombre maximal de modifications autorisées par jour
$maxModificationsPerDay = 3
# Déterminer la date d'aujourd'hui
$currentDate = Get-Date -Format "yyyy-MM-dd"
# Vérifier si le fichier existe
if (Test-Path $fileToMonitor) {
    # Obtenir les détails de modification du fichier
    $fileDetails = Get-Item $fileToMonitor
    # Obtenir le nombre de modifications faites aujourd'hui
    $modificationsToday = $fileDetails.LastWriteTime |
Where-Object { $_.ToString("yyyy-MM-dd") -eq
$currentDate } | Measure-Object
```




```
# Vérifier si le nombre de modifications aujourd'hui
dépasse la limite
if ($modificationsToday.Count -gt
$maxModificationsPerDay) {
    # Envoyer une alerte
    Write-Host "Alerte : Plus de
    $maxModificationsPerDay modifications ont été
    faites dans le fichier aujourd'hui."
    # Créer un dossier et un fichier nommé
    "fichier_a_surveiller" dans le répertoire C:\
    $folderPath = "C:\ fichier_a_surveiller"
    $filePath = "C:\
    fichier_a_surveiller\fichier_a_surveiller.txt"
    # Vérifier si le dossier existe, sinon le créer
    if (-not (Test-Path $folderPath)) {
        New-Item -ItemType Directory -Path
        $folderPath | Out-Null
        Write-Host "Dossier 'fichier_a_surveiller' créé
        dans C:\."
    }
}
```



```
# Créer le fichier "fichier_a_surveiller.txt"
    New-Item -ItemType File -Path $filePath |
Out-Null
    Write-Host "Fichier 'fichier_a_surveiller.txt' créé
dans C:\fichier_a_surveiller."
} else {
    Write-Host "Aucune modification détectée
dans le fichier aujourd'hui."
}
} else {
    Write-Host "Le fichier spécifié n'existe pas."
}
```


```
Sélection Administrateur : Windows PowerShell


>>> Write-Host "Alerte : Plus de $maxModificationsPerDay modifications ont été faites dans le fichier aujourd'hui."
>>>
>>> # Créer un dossier et un fichier nommé "fichier_a_surveiller" dans le répertoire C:\
>>> $folderPath = "C:\ fichier_a_surveiller"
>>> $filePath = "C:\ fichier_a_surveiller\fichier_a_surveiller.txt"
>>>
>>> # Vérifier si le dossier existe, sinon le créer
>>> if (-not (Test-Path $folderPath)) {
>>>     New-Item -ItemType Directory -Path $folderPath | Out-Null
>>>     Write-Host "Dossier 'fichier_a_surveiller' créé dans C:\."
>>> }
>>>
>>> # Créer le fichier "fichier_a_surveiller.txt"
>>> New-Item -ItemType File -Path $filePath | Out-Null
>>> Write-Host "Fichier 'fichier_a_surveiller.txt' créé dans C:\fichier_a_surveiller."
>>> } else {
>>>     Write-Host "Aucune modification détectée dans le fichier aujourd'hui."
>>> }
>>> } else {
>>>     Write-Host "Le fichier spécifié n'existe pas."
>>> }
>>>
Aucune modification détectée dans le fichier aujourd'hui.
PS C:\Users\Administrateur> _
```



quelques idées qui permettent de sécuriser le système.

- Nous devrions mettre en place des **politiques de mot** de passe robustes, y compris l'utilisation de mots de passe complexes et leur changement périodique, ainsi que l'exploration des options de mots de passe à usage unique ou d'authentification multifactorielle.
- Nous devrions **configurer une surveillance des journaux** d'audit pour détecter les activités suspectes telles que les tentatives de connexion infructueuses ou les changements de privilèges.
- Il est crucial de maintenir à jour tous les systèmes AD avec **les derniers correctifs** de sécurité pour remédier aux vulnérabilités connues.

- 
- L'application du principe du **moindre privilège est essentielle**, en limitant les privilèges aux seuls utilisateurs nécessaires et en utilisant des comptes d'administration distincts.
 - Nous devrions mettre en œuvre des **contrôles d'accès basés sur les rôles** pour limiter l'accès aux ressources en fonction des besoins des utilisateurs.
 - La sécurité des données sensibles stockées dans AD devrait être renforcée par **le chiffrement et la sécurisation des sauvegardes**.
 - Une formation régulière des utilisateurs sur les meilleures pratiques de sécurité est nécessaire pour les sensibiliser aux risques tels que le phishing.
 - **Des tests de vulnérabilité et des analyses régulières** sont essentiels pour identifier et corriger les failles de sécurité dans l'infrastructure AD.

- 
- Il est primordial de **sauvegarder régulièrement les données AD** et de disposer d'un plan de reprise après sinistre en cas d'incident.
 - La sécurité physique des serveurs AD doit être garantie en limitant l'accès physique et réseau aux contrôleurs de domaine.
 - **La restriction des ports** et services exposés sur les contrôleurs de domaine et la gestion des groupes privilégiés sont également essentielles pour renforcer la sécurité.
 - Enfin, la formation continue du personnel chargé de la gestion et de la sécurité d'Active Directory est cruciale pour rester à jour sur les meilleures pratiques et les menaces émergentes.