

TP : Administration Avancée avec PowerShell

Objectifs du TP :

L'objectif de ce TP est de permettre aux étudiants de se familiariser avec l'utilisation de **PowerShell** pour effectuer des tâches avancées d'administration système, notamment l'automatisation de processus, l'administration à distance, et la planification de tâches. Ils utiliseront également **VS Code** pour écrire et tester des scripts PowerShell et **GitHub** pour versionner et partager leurs scripts.

Structure du Projet :

Le projet est divisé en trois sections principales, chacune ayant des objectifs spécifiques.

/admin-avancee-powershell

|

| — /scripts # Dossier contenant tous les scripts PowerShell

| | — gestion-processus.ps1 # Script pour la gestion des processus et des tâches en
arrière-plan

| | — admin-distance.ps1 # Script pour l'administration à distance avec
PowerShell Remoting

| | — taches-planifiees.ps1 # Script pour la planification et l'automatisation des
tâches

| | — utils.ps1 # Script utilitaire pour des fonctions réutilisables (ex : gestion
des erreurs, logging)

|

| — /docs # Dossier pour la documentation du projet

| | — README.md # Fichier principal de documentation du projet

| | — guide_utilisation.md # Guide détaillé pour l'utilisation de chaque script

|

| — .gitignore # Fichier pour ignorer les fichiers inutiles (logs, fichiers
temporaires, etc.)

1. Gestion des Processus et Tâches en Arrière-Plan

Objectif :

Automatiser et gérer des tâches en arrière-plan sur un système local.

Consignes :

1. **Créer un script PowerShell** qui lance une tâche en arrière-plan. Cette tâche peut impliquer des opérations longues (comme l'analyse de disque ou l'exécution d'un rapport).
2. **Gestion de l'état des tâches :**
 - Vérifier l'état d'exécution des tâches en arrière-plan.
 - Gérer les erreurs et afficher des informations sur l'état de la tâche (en cours, terminé, ou échoué).

2. Administration à Distance avec PowerShell

Objectif :

Utiliser PowerShell pour se connecter et administrer des systèmes distants via PowerShell Remoting.

Consignes :

1. **Configurer PowerShell Remoting** sur les machines distantes. Assurez-vous que la fonctionnalité WinRM est activée et que les règles de pare-feu nécessaires sont configurées.
2. **Créer un script PowerShell** qui permet d'exécuter une commande sur une machine distante. Cela pourrait être un rapport d'espace disque ou la gestion de services à distance.
3. **Créer une gestion à distance sécurisée :**
 - Utiliser `Get-Credential` pour sécuriser les informations de connexion.

- Utiliser `Enter-PSSession` ou `Invoke-Command` pour exécuter des commandes distantes.
-

3. Planification et Automatisation des Tâches

Objectif :

Automatiser l'exécution des tâches récurrentes en utilisant des tâches planifiées.

Consignes :

1. **Créer un script PowerShell** qui exécute une tâche récurrente (par exemple, la sauvegarde d'un répertoire ou la génération de rapports) à des intervalles définis (quotidiens, hebdomadaires, etc.).
 2. **Planifier des tâches à l'aide de `New-ScheduledTask` :**
 - Créer des tâches planifiées pour exécuter des scripts PowerShell automatiquement à des moments définis.
 - Configurer le déclencheur (par exemple, l'exécution chaque jour à une heure précise) et l'action à exécuter (par exemple, un script de sauvegarde).
 3. **Exécution de tâches à distance :**
 - Ajouter une fonctionnalité pour créer des tâches planifiées sur des machines distantes via PowerShell Remoting.
-

Livrables :

1. **Dépôt GitHub** avec tous les scripts PowerShell versionnés.
 - Chaque script doit être bien commenté pour expliquer son fonctionnement.
 - Chaque partie du projet doit être contenue dans un fichier script distinct.
2. **README.md** détaillant :
 - La mise en place de chaque section (gestion des processus, administration à distance, et planification des tâches).

- Comment exécuter chaque script, y compris les prérequis pour chaque section.
- Exemples de commandes ou de sorties attendues.