

Guide apprenant – Situation professionnelle TSSR

Automatisation de tâches d'exploitation par scripting PowerShell

Objectif pédagogique global

Cette situation professionnelle a pour objectif de te placer dans un **contexte réel de technicien systèmes et réseaux**, afin de :

- Analyser une situation existante,
- Proposer une solution d'automatisation,
- Développer un script PowerShell professionnel,
- Sécuriser, documenter et planifier cette solution.

Tu n'es pas évalué uniquement sur le résultat, mais aussi sur :

- Ta méthode,
- La clarté de ton script,
- Ta capacité à expliquer tes choix.

1 — Analyse de l'environnement et diagnostic

Objectif

Comprendre l'environnement de travail et produire un **état des lieux fiable** avant toute automatisation.

Tâches à accomplir

Tu disposes d'un dossier nommé **Depot** contenant des fichiers hétérogènes.

Tu dois :

1. Explorer le contenu du dossier à l'aide de PowerShell.
2. Identifier :
 - Le nombre total de fichiers,
 - Les types de fichiers présents (extensions),
 - Les fichiers les plus volumineux.
3. Produire un **rapport lisible** à destination du responsable informatique.

Indications (sans donner la solution)

- Utilise des cmdlets comme :
 - Get-ChildItem
 - Where-Object
 - Select-Object
 - Sort-Object
- Appuie-toi sur le **pipeline PowerShell**.
- Pense à stocker des résultats intermédiaires dans des **variables**.

Livrables attendus

- audit.ps1
- audit-report.txt

2 — Automatisation d'une tâche d'exploitation

Objectif du jour

Transformer l'audit en un **script d'automatisation sécurisé**.

Tâches à accomplir

1. Créer un script PowerShell paramétrable :
 - dossier source
 - mode de fonctionnement
2. Mettre en place deux modes :
 - **Audit** : aucune modification
 - **Action** : organisation réelle des fichiers
3. Structurer ton code avec des **fonctions réutilisables**.

Indications

- Utilise param() avec des paramètres obligatoires.
- Implémente un switch ou des if pour gérer les modes.
- Les fonctions doivent avoir un nom clair (Verbe-Nom).

Livable attendu

- organize-depot.ps1

3 — Robustesse, erreurs et traçabilité

Objectif du jour

Rendre le script **exploitable en production**, même en cas de problème.

Tâches à accomplir

1. Ajouter une gestion des erreurs :
 - dossier absent
 - fichier inaccessible
2. Mettre en place des **logs horodatés**.
3. Distinguer :
 - erreurs bloquantes
 - erreurs non bloquantes

Indications

- Utilise try / catch / finally.
- Appuie-toi sur -ErrorAction Stop.
- Les logs doivent permettre de comprendre ce qui s'est passé **sans relancer le script**.

Livrables attendus

- Script finalisé
- script.log

4 — Industrialisation et planification

Objectif du jour

Préparer le script pour une **exécution automatique et autonome**.

Tâches à accomplir

1. Supprimer toute interaction utilisateur (Read-Host interdit).
2. Tester l'exécution complète en ligne de commande.
3. Préparer la planification via le planificateur de tâches Windows.

Indications

- Tous les paramètres doivent être passés en argument.
- Le script doit fonctionner sans interface.
- Prévois un code de sortie (exit 0, exit 1).

Livrables attendus

- README-planification.txt
- Description de la tâche planifiée

5 — Restitution professionnelle

Objectif du jour

Rendre compte de ton travail comme un **technicien en situation réelle**.

Tâches à accomplir

1. Rédiger un rapport final synthétique :
 - objectifs
 - fonctionnement du script
 - difficultés rencontrées
 - améliorations possibles
2. Fournir une archive complète et propre.

Livrables attendus

- README-final.txt
- Archive .zip finale