

Maintenance de stockage via PowerShell

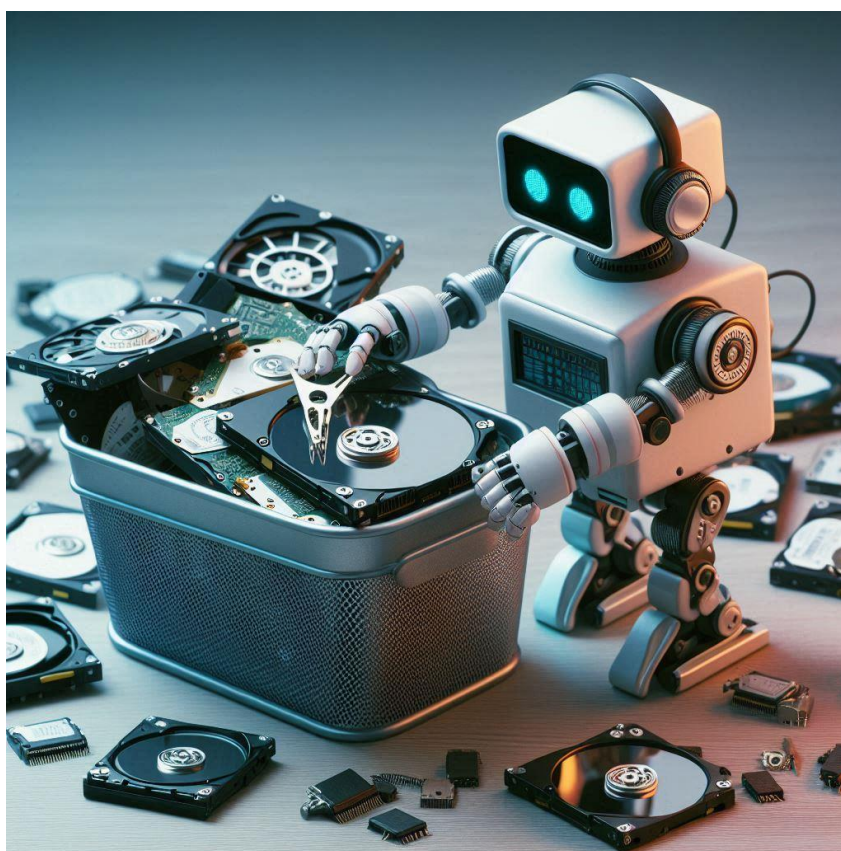


Figure 1 - Image générée par Bing Copilot représentant le projet

Yann Mangiagli – FIN2
Vennes
88H
Gaël Sonney
Experts : Daniel Berney, Gabriel Maret

Table des matières

1	RÉSUMÉ DE LA DOCUMENTATION / RÉSUMÉ DU RAPPORT	4
1.1.1	<i>Situation de départ</i>	4
1.1.2	<i>Mise en œuvre</i>	4
1.1.3	<i>Résultat</i>	4
2	SPÉCIFICATIONS.....	5
2.1	TITRE.....	5
2.2	DESCRIPTION.....	5
2.3	MATÉRIEL ET LOGICIELS À DISPOSITION	5
2.4	PRÉREQUIS	5
2.5	CAHIER DES CHARGES.....	6
2.5.1	<i>Objectifs et portée du projet (objectifs SMART et CQQCOQP)</i>	6
2.5.2	<i>Caractéristiques des utilisateurs et impacts.....</i>	6
2.5.3	<i>Fonctionnalités requises (du point de vue de l'utilisateur)</i>	7
2.5.4	<i>Contraintes.....</i>	7
2.5.5	<i>Travail à réaliser par l'apprenti.....</i>	7
2.5.6	<i>Si le temps le permet</i>	8
2.5.7	<i>Méthodes de validation des solutions</i>	8
2.6	LES POINTS SUIVANTS SERONT ÉVALUÉS	8
2.7	VALIDATION ET CONDITIONS DE RÉUSSITE.....	8
3	PLANIFICATION INITIALE.....	8
3.1.1	<i>Méthode des 6 pas</i>	8
3.1.2	<i>Mises à jour du planning.....</i>	9
3.1.3	<i>Diagramme de Gantt.....</i>	10
3.1.4	<i>Bilan graphique</i>	15
4	ANALYSE.....	16
4.1	OPPORTUNITÉS	16
4.1.1	<i>Approfondir</i>	16
4.1.2	<i>Difficultés potentielles</i>	16
4.1.3	<i>Matériel à exploiter</i>	16
4.1.4	<i>Recherche d'informations particulière.....</i>	16
4.1.5	<i>Solutions possibles.....</i>	16
4.2	DOCUMENT D'ANALYSE ET CONCEPTION	17
4.2.1	<i>Schéma de conception</i>	17
4.2.2	<i>Script 1</i>	17
4.2.3	<i>Script 2</i>	21
4.3	CONCEPTION DES TESTS	23
4.4	PLANIFICATION DÉTAILLÉE	23
5	RÉALISATION	27
5.1	DOSSIER DE RÉALISATION	27
5.1.1	<i>Version des outils logiciels utilisés.....</i>	27
5.1.2	<i>Comptes utilisés</i>	28
5.1.3	<i>Configuration spéciale des serveurs</i>	28
5.1.4	<i>Déploiement et exécution détaillée.....</i>	32
5.1.5	<i>En tête commune aux 2 scripts</i>	33
5.1.6	<i>Réalisation du script de setup.....</i>	33
5.1.7	<i>Réalisation du script de maintenance</i>	37
5.2	MODIFICATIONS	42

6	TESTS	42
6.1	DOSSIER DES TESTS.....	42
7	CONCLUSION.....	43
7.1	BILAN DES FONCTIONNALITÉS DEMANDÉES.....	43
7.2	BILAN DE LA PLANIFICATION	44
7.3	BILAN PERSONNEL	44
7.3.1	Remerciements.....	45
8	DIVERS.....	45
8.1	JOURNAL DE TRAVAIL	45
8.2	WEBOGRAPHIE.....	51
8.3	TABLE DES ILLUSTRATIONS.....	52
9	GLOSSAIRE	54
10	ANNEXES.....	55

1 RÉSUMÉ DE LA DOCUMENTATION / RÉSUMÉ DU RAPPORT

1.1.1 Situation de départ

Deux machines distantes et une machine locale sont reliées par Ethernet sur un réseau LAN. Deux scripts doivent être créés, un pour la maintenance du stockage des pc, l'autre pour sa mise en place. Le script de maintenance, appelé script 1 dans ce rapport, permet de supprimer les fichiers OVA de plus de 3 Go. Le script de mise en place, surnommé script 2, sert à copier à distance le script 1 en utilisant les adresses IP du réseau en paramètre pour identifier l'ordinateur de destination.

1.1.2 Mise en œuvre

La mise en œuvre du projet commence par un travail fait en amont en paramétrant 2 serveurs Windows Server 2022, qui ont été reliés via le réseau de l'école par câble Ethernet. Les pare-feux ont ensuite été désactivés et le Windows remote activé (avec ajout en hôte trusté via PowerShell). Par la suite une planification initiale a été réalisée, ainsi qu'un schéma illustrant comment les scripts vont procéder. Les scripts sont ensuite réalisés petit à petit et postés sur GitHub afin d'avoir un versioning. Le rapport est rédigé en parallèle du projet, comme le journal de travail.

1.1.3 Résultat

Les 2 scripts fonctionnent, malgré certains chemins en dur pour la copie du script. Le script 2 copie le script 1 dans le dossier C:\Scripts que le script 2 crée si le répertoire n'existe pas déjà. Le script 2 crée une tâche planifiée à 16h qui lance le script de maintenance tous les jours à l'heure convenue. Le script 1 analyse le stockage de l'ordinateur sur lequel il est. Si le taux de stockage atteint plus de 80%, il nettoie le disque en supprimant tous les fichiers OVA de plus de 3Go du disque C. Il envoie dans le même temps un mail d'avertissement à l'adresse électronique tpiymetml@gmail.com. Le mail contient la version de l'OS, la version des mises à jour, l'espace libre en octets et l'espace utilisé en octets.

2 SPÉCIFICATIONS

2.1 Titre

Maintenance automatique et régulière du stockage des postes de travail à distance

2.2 Description

Les étudiants de la section informatique sont amenés à utiliser de nombreuses machines virtuelles pendant leur formation qu'ils et elles laissent sur leur lecteur (H :) qui leur servent de répertoire utilisateur sans les utiliser de nouveau. Ces machines virtuelles sont donc stockées en vain sur les machines de l'ETML plutôt que sur le SSD des étudiants et prennent beaucoup de place avec le temps. Le but de ce projet est de limiter l'impact des machines virtuelles sur le stockage des postes de l'école en avertissant quand le stockage d'un poste atteint 80% et en envoyant un mail à l'administrateur qui pourra identifier la machine pendant que le script supprime les fichiers OVA de plus de 3Go des répertoires des élèves s'étant connecté à cette machine.

2.3 Matériel et logiciels à disposition

PC standard de l'ETML sous Windows 10

PowerShell

PowerShell ISE

Deux machines distantes (WINSERV2022) pour le déploiement du script à distance avec les droits d'administrateur configurées par l'apprenti en amont.

2.4 Prérequis

Module 431 - Planification de projets et de tâches

Module 122 – PowerShell

Module 117 – Réseau (Windows Server 2022, Câblage simple, Remote Desktop)

2.5 Cahier des charges

2.5.1 Objectifs et portée du projet (objectifs SMART et CQQCOQP)

CQQCOQP

Combien : Le projet ne coûte rien à produire, il rapporte de l'efficacité en supprimant une tâche redondante et permet aux administrateurs système de faire autre chose et augmenter leur productivité. Cela consomme peu car la tâche est faite une fois par jour brièvement, mais il ne faut pas oublier que les mails restent dans une boîte mail et qu'une trop grande quantité est polluante. C'est pour cela qu'il faut se débarrasser des mails qui ne sont plus utiles.

Quoi : Les actions à effectuer sont la création de 2 scripts différents. L'un des scripts permet d'installer l'autre, pendant que celui-ci surveille le stockage des élèves.

Qui : Un administrateur système va réaliser le projet. Cela concerne les élèves ayant des machines virtuelles sur leur disque et les administrateurs système, pour faciliter la vie des administrateurs système du service informatique de l'ETML.

Comment : Utilisation des compétences acquises durant le CFC.

Où : Les ordinateurs distants sont à l'étage au-dessus de la salle A13 dans la salle des serveurs de Vennes. La machine locale est en salle A13.

Quand : Du 13 mai au 3 juin 2024.

Pourquoi : Améliorer la productivité et soulager la charge de travail du service informatique de l'ETML tout en libérant de l'espace de stockage pour les élèves.

SMART

Spécifique : Réaliser 2 scripts dont un capable de copier l'autre script sur une autre machine à distance, pendant que l'autre script analyse le stockage du pc et envoie des informations par mail sur le pc sur lequel s'exécute le script.

Mesurable : Chaque jour le progrès est enregistré sur un journal de travail, puis tous les mardi et vendredi un rendu « complet » est fait.

Atteignable : Utilisation des compétences apprises durant le CFC.

Réaliste : Tous les élèves doivent quand même pouvoir poser des fichiers volumineux sans que ceux-ci ne se fassent supprimer s'ils ne sont pas des ova.

Temporel : Le projet doit être fait en 88H, du lundi 13 mai 2024 au lundi 3 juin 2024.

2.5.2 Caractéristiques des utilisateurs et impacts

Les utilisateurs du script seraient des administrateurs système et réseau désirant optimiser le stockage des postes de travail supportant des machines virtuelles, comme le service informatique de l'ETML. Le but est donc d'automatiser les scripts tout en laissant aux administrateurs le moyen de personnaliser l'utilisation du script avec des paramètres. Par exemple, choisir la taille maximum tolérée par le script (par défaut 3Go comme mentionné plus haut).

2.5.3 Fonctionnalités requises (du point de vue de l'utilisateur)

Pour le premier script, l'utilisateur doit pouvoir récupérer les informations concernant l'espace disque utilisé ainsi que l'espace libre, le nom de l'ordinateur, la version de son système, ainsi que les versions des mises à jour de l'OS. Le script envoie un courriel sur le mail d'entreprise de l'administrateur un avertissement quand 80% du stockage est utilisé, et doit pouvoir supprimer les fichiers trop volumineux, avec éventuellement un paramètre permettant de décider ce qui est considéré comme « trop volumineux ». Le deuxième script est plus simple et contient moins de fonctionnalités. L'utilisateur a juste besoin de le lancer pour qu'il copie le script 1 sur une machine à distance ne le possédant pas pour qu'il crée la tâche planifiée correspondante. Le script peut gérer une liste de 10 ordinateurs en simultané pour son exécution.

2.5.4 Contraintes

Plusieurs contraintes peuvent exister avec des scripts comme ceux-là. Par exemple, le mail envoyé par le script a une chance non négligeable de finir dans les spam, car il n'y a pas le temps d'installer un serveur SMTP et c'est pour cela qu'une adresse Gmail sera utilisée, car il est plus simple et plus rapide de se connecter via PowerShell sur ce serveur SMTP. Comme la sécurité n'est pas mentionnée dans le cahier des charges, la sécurité du script ne sera pas forcément optimale car cela ne sera pas une priorité. Par exemple, pour automatiser l'enregistrement des identifiants en utilisant la commande « get-credential », il faudra passer par écrire le mot de passe en clair dans le script pour ne pas perdre du temps là-dessus. Cela pourrait être un point à améliorer s'il reste du temps quand les 2 scripts seront prêts.

2.5.5 Travail à réaliser par l'apprenti

L'apprenti doit réaliser 2 scripts ainsi qu'un rapport, une planification et un journal de travail. Le premier script doit récupérer des informations système comme le nom, version de l'OS, Espace disque libre & utilisé, version des mises à jour et envoyer un mail contenant les informations permettant d'identifier le poste de travail à l'administrateur quand le stockage utilisé égale ou dépasse les 80%. Il sera ensuite possible de supprimer en priorité les fichiers de machine virtuelles, en particulier les fichiers avec une extension ova. Le deuxième script est une automatisation. Il permet de sélectionner une liste jusqu'à 10 ordinateurs distants, et d'implémenter le premier script afin qu'il puisse scanner et surveiller le stockage des postes de travail sur lesquels il vient de s'installer, tout en créant une tâche planifiée qui planifiera ce scan. Le rapport doit contenir toute la documentation qu'il a créée durant le projet ainsi que les tests réalisés.

2.5.6 Si le temps le permet ...

Il serait judicieux d'améliorer la sécurité pour rendre le script utilisable par une entreprise ou une école, ainsi qu'installer un serveur SMTP pour ne pas avoir à utiliser une adresse Gmail mais une adresse eduvaud dans l'exemple de l'ETML utilisé comme prétexte pour ce TPI.

2.5.7 Méthodes de validation des solutions

Chaque fonctionnalité demandée dans le script sera testée en premier lieu seule pour être sûr que la solution fonctionne avant de l'incorporer dans le script 1 ou 2 suivant l'instruction(s) testée(s). Selon la commande, elle devra être testée sur une machine distante, comme la commande de suppression des fichiers ova de 3Go. Le maximum de résultats de tests seront sujet à une capture d'écran montrant le résultat. Ils seront aussi documentés dans la partie réalisation du rapport.

2.6 Les points suivants seront évalués

- Le rapport
- Les planifications (initiale et détaillée)
- Le journal de travail
- Le code et les commentaires
- Les documentations de mise en œuvre et d'utilisation

2.7 Validation et conditions de réussite

- Compréhension du travail
- Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
- Etat de fonctionnement du produit livré

3 PLANIFICATION INITIALE

Le seul jour spécial est pentecôte, ce qui rend le lundi 20 mai 2024 chômé. Le lendemain, l'expert 2 vient voir l'avancée du projet vers 13h30.

3.1.1 Méthode des 6 pas

Le projet commence le 13 mai 2024 et se termine le lundi 3 juin 2024. La méthode des 6 pas a été choisie pour ce projet. Cette méthode de planification consiste à se fixer des objectifs

S'informer :

Comprendre et enregistrer la tâche à accomplir, se renseigner sur le travail, sur ce qui est attendu, ce qui est « libre ». C'est dans ce moment-là qu'on se renseigne sur l'échéance, sur si on est capable ou non de réaliser ce projet, le matériel à disposition... Dans ce projet, on peut faire le parallèle avec la prise de connaissance du cahier des charges, les questions posées au chef de projet et aux experts en rapport avec ce document, la configuration des deux machines Windows server 2022 utilisées dans le projet comme machines distantes.

Planifier :

C'est tout ce qui concerne la planification du projet pour en arriver à bout. Il faut donc se fixer des priorités dans le projet, faire en premier lieu ce qui est le plus important comme les fonctionnalités principales du script avant

la gestion des erreurs par exemple. En gros, quelles tâches réaliser en premier pour avoir le plus vite possible un prototype fonctionnel. Il est aussi décidé quelle méthode de gestion de projet est utilisée, comme agile, 6 pas, etc. Il faut aussi estimer le temps que chaque tâche prendra. C'est un exercice très difficile car on ne peut pas prévoir tous les problèmes qu'un projet va inévitablement rencontrer, ni leur gravité, comme une perte totale des données par exemple.

Décider :

Que le projet soit à réaliser seul(e) ou à plusieurs, il est impératif d'avoir un consensus avec tous les partis impliqués dans le projet. À plusieurs, il faut être d'accord entre collègues, pour pas que chacun fasse quelque chose de différent qui fasse juste perdre du temps.

Réaliser / exécuter :

Partie principale du projet. C'est à ce moment qu'on travaille sur le projet en le réalisant. Dans l'idéal, il faudrait que ça se déroule exactement comme dans la planification. Pour faire un parallèle avec ce projet, c'est la création des scripts.

Contrôler :

On regarde le travail réalisé plus haut et on le vérifie. On reteste, on relit la documentation et on la modifie si besoin.

Évaluer :

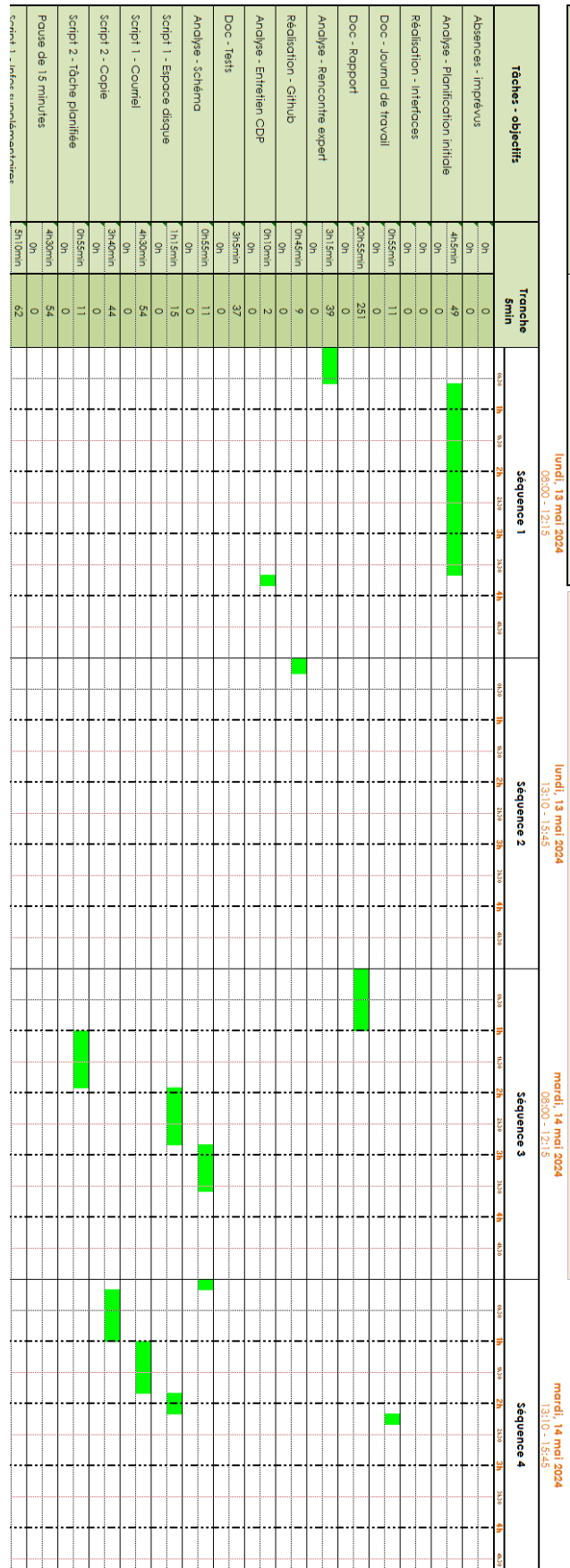
Passage en revue du projet effectué, on parle de ce qui a été bien fait, de ce qui a été mal fait, de ce qui aurait pu être amélioré. Dans ce rapport on pourrait assimiler cette étape au bilan des fonctionnalités demandées dans la conclusion.

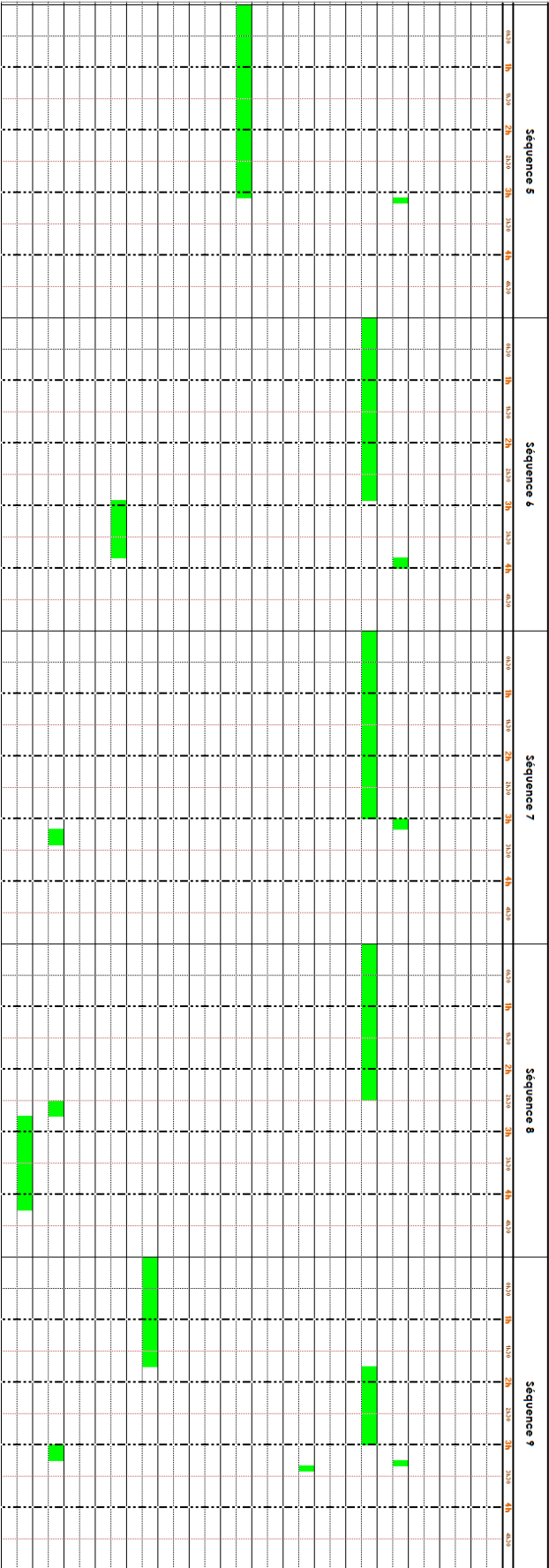
3.1.2 Mises à jour du planning

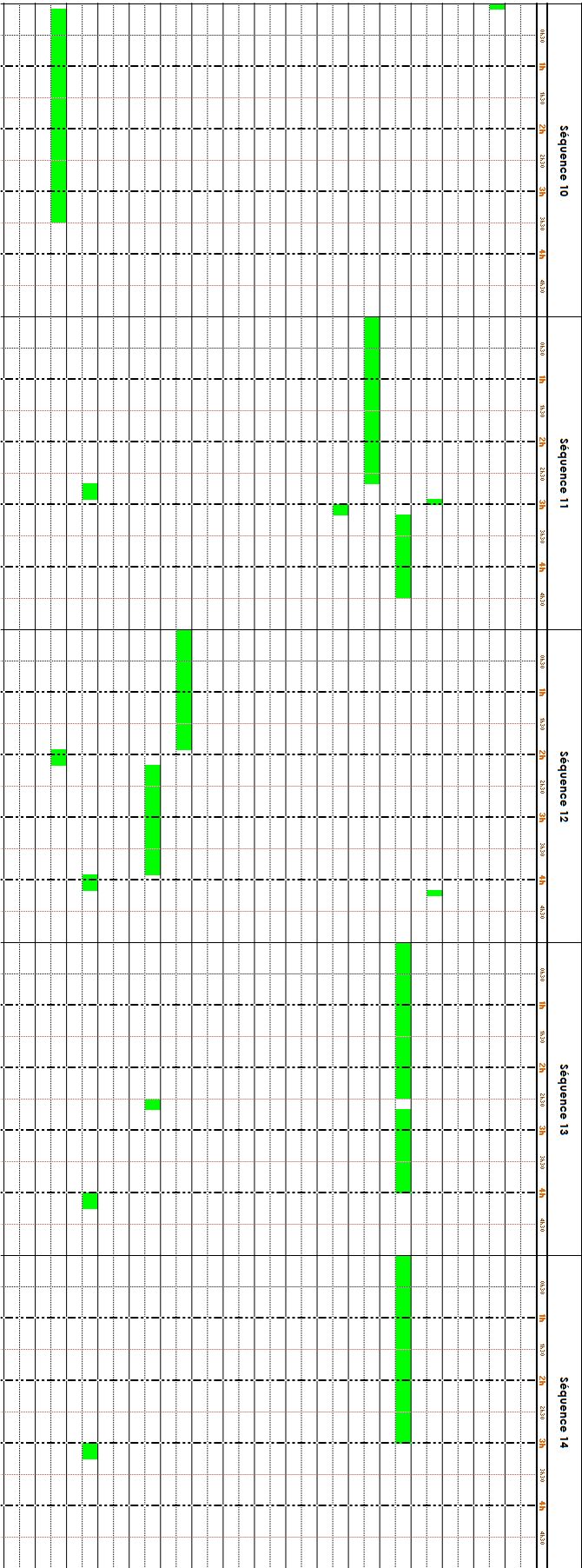
14.05.24 :

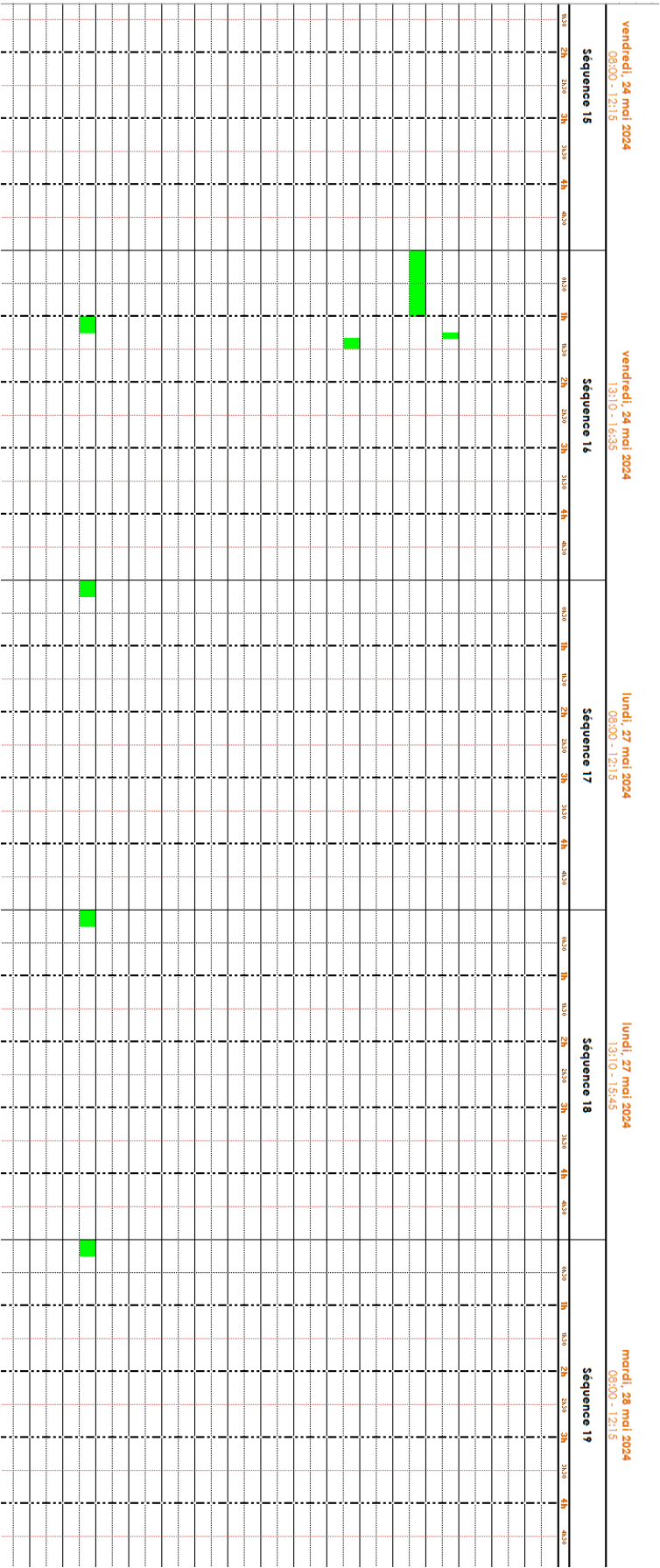
Modification de la rencontre avec expert 2, décidée arbitrairement dans la planification initiale par l'apprenti pour le 23 mai avant d'être modifié au 21 quand l'expert 2 et l'élève se sont mis d'accord.

3.1.3 Diagramme de Gantt

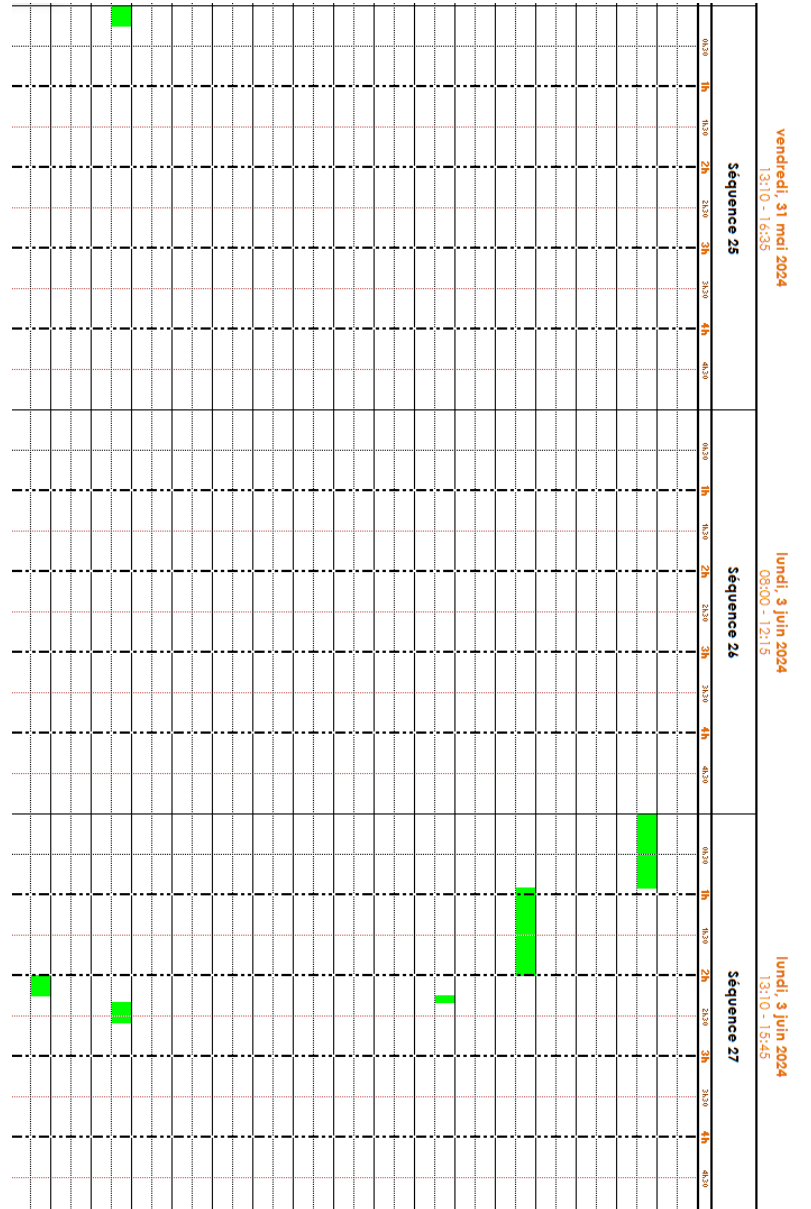




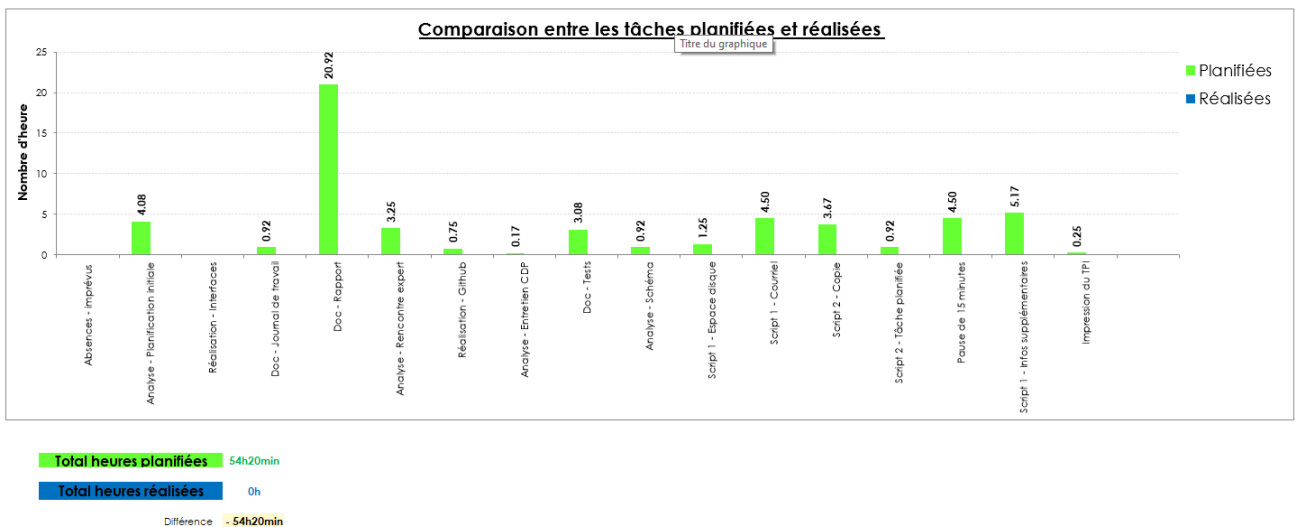




Auteur :Yann Mangiagli
Modifié par : Yann Mangiagli
Version: 7 du 03.06.2024 14:21



3.1.4 Bilan graphique



4 ANALYSE

4.1 Opportunités

4.1.1 Approfondir

Il faudra approfondir certaines connaissances en PowerShell pour mener à bien le projet.

- Commande « New-ScheduledTaskAction » à apprendre à utiliser
- Apprendre à exécuter les scripts depuis un autre ordinateur (Copy-Item - ToSession)

4.1.2 Difficultés potentielles

Plusieurs difficultés peuvent survenir, surtout à cause d'imprévus.

- 1) Malgré le script d'envoi de mail fonctionnel réalisé en amont, rien ne promet qu'il fonctionnera dans un contexte différent
- 2) Comment vérifier que le script 1 s'exécute bien à 80% alors que les machines à disposition ont des disques vides ?
- 3) La copie du script sur une liste de machines n'a pas l'air simple à mettre en place
- 4) Pour le script de copie, il faut copier un script sur la machine principale et le coller sur la machine distante
- 5) Un serveur qui ne réagit pas comme l'autre
- Il faut internet pour envoyer un mail

4.1.3 Matériel à exploiter

Le matériel exploitable peut lui aussi causer des difficultés, par exemple un serveur mal paramétré en amont, ou encore si le réseau de l'école est changé. C'est pour cela qu'il est impossible d'utiliser le script depuis une machine distante sur la machine locale, car elle a besoin de droit administrateur que je ne possède pas, et elle est protégée derrière docker kubernetes.

4.1.4 Recherche d'informations particulière

Comme PowerShell et ses erreurs sont bien documentées, il y'a peu de chances de rencontrer des difficultés de ce côté-ci. En revanche, un problème d'environnement rendrait tout de suite la tâche plus ardue.

4.1.5 Solutions possibles

- 2) Réduction du quota de 80% à 3% par exemple
- 3) Une boucle « foreach » sur un tableau ayant la liste des ordinateurs
- Supprimer Invoke-Command, car on essayait d'accéder à la session depuis la session, c'est donc normal que ça ne fonctionnait pas
- 5) Les pare-feux des 2 serveurs n'étaient pas identiques. Le premier était désactivé et pas l'autre.

4.2 Document d'analyse et conception

4.2.1 Schéma de conception

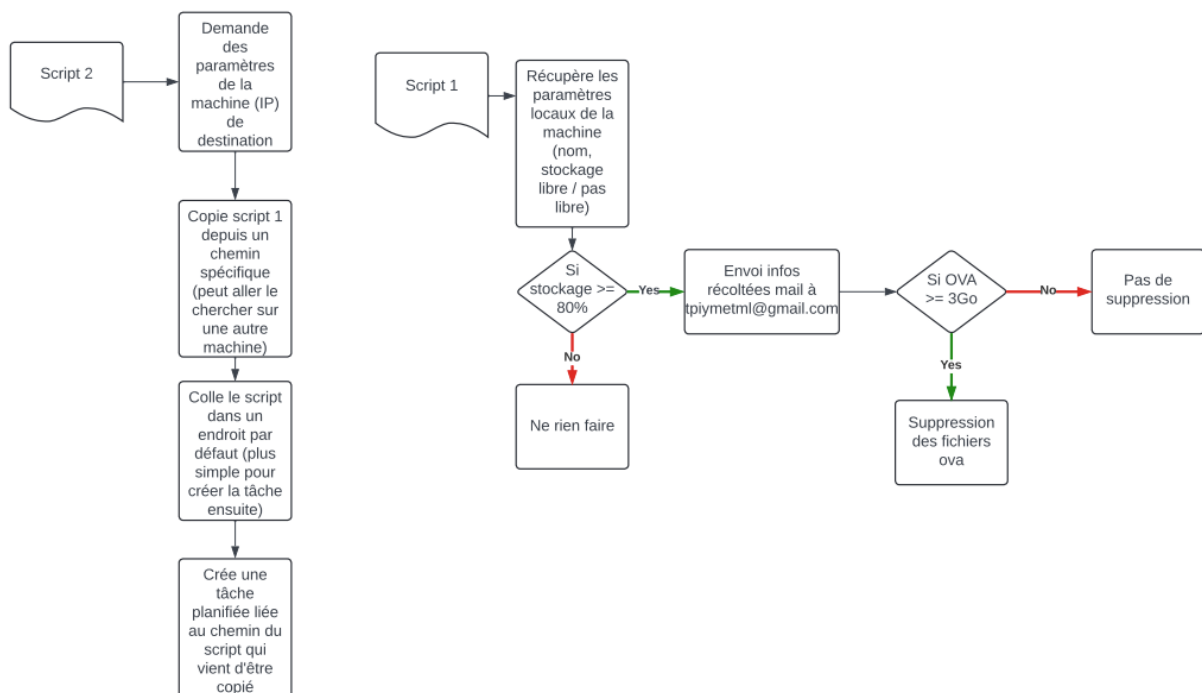


Figure 2 - Schéma démontrant comment les scripts ont été pensés

4.2.2 Script 1

Pour le script 1, des tests ayant été faits durant un projet précédent ont mis en exergue le fait qu'utiliser Send-MailMessage était obsolète et n'était plus supporté par Microsoft. Malgré le fait que des alternatives existent (MimeKit & MailKit), il a été décidé de rester sur cette façon de faire car elle est plus simple bien que moins sécurisée. Je suis parti du principe que vu que le script 2 permettra de copier le script à distance et de le faire s'exécuter périodiquement via une tâche planifiée, le script 1 n'a pas besoin d'être utilisable à distance ; il suffira de lancer le script 2 avant pour le copier sur la machine distante.

```

29     # Tableau de valeurs
30     $valuesArray = @{
31         MachineNom = "x";
32         VersionOS = "x";
33         EspaceUtilise = "x";
34         EspaceLibre = "x";
35         VersionMAJ = "x";
36     }
  
```

Figure 3 - Création du tableau

Un tableau associatif est créé afin de lier chaque valeur récupérée dans le script pour pouvoir tout imprimer dans le mail d'un coup grâce à une boucle foreach. Bien qu'il soit possible de directement mettre les valeurs dans le tableau, j'ai préféré les rajouter plus tard afin d'avoir un code plus lisible.

```
38 # Récupération du nom de l'ordinateur
39 $valuesArray.MachineNom = $env:COMPUTERNAME
```

Figure 4 - Variable d'environnement du nom de l'ordinateur

Le nom de la machine est stocké dans une variable qu'on appelle variable d'environnement, d'où le \$env : devant. Elles existent déjà dans le système, et la variable \$env:COMPUTERNAME donne le nom de l'ordinateur qui exécute le script. Elle est utilisée ici pour être entrée dans le tableau associatif la liant à l'indice MachineNom.

```
41 # Récupération de la version de l'OS
42 $valuesArray.VersionOS = $env:OS + " " + [environment]::OSVersion.Version.Build
```

Figure 5 - Variables d'environnement de la version de l'OS

Même concept que pour la valeur de la ligne 39, mais avec deux valeurs. \$env:OS donne le nom du système d'exploitation (dans le contexte de la machine sur laquelle je fais mes tests, cela équivaut à Windows NT, tandis que la seconde valeur contient le numéro de build de la version). Cela permet d'avoir un résultat plus esthétique, comme suit sans les guillemets : « Version de l'OS : Windows_NT 19045 ».

```
44 # Récupération de l'espace disque libre et de la lettre
45 $tempvalue = Get-PSDrive | Select-Object Free
46 $valuesArray.EspaceLibre = $tempvalue.Free
```

Figure 6 - Enregistrement de l'espace libre

Une valeur temporaire est créée à la ligne 45 qui sera réinstanciée plus tard pour économiser la mémoire en ne créant pas trop de variables inutiles. Cette instruction récupère l'espace libre stocké dans Get-PSDrive puis stocke la partie Free du tableau de PSDrive dans \$valuesArray avec la clé EspaceLibre.

```
48 # Récupération de l'espace disque utilisé
49 $tempvalue = Get-PSDrive | Select-Object Used
50 $valuesArray.EspaceUtilisé = $tempvalue.Used
```

Figure 7 - Enregistrement de l'espace utilisé

Comme la valeur temporaire a été utilisée, et que sa valeur a été inscrite dans le tableau, on peut la réutiliser sans perdre les données. Le script utilise donc les mêmes instructions que pour l'espace libre mais on remplace la valeur recherchée par Used. On aura donc l'espace utilisé dans les disques.

```
52 # Récupération des versions de mise à jour de l'OS
53 $tempvalue = Get-HotFix -ComputerName $valuesArray.MachineNom | Select HotFixID
54 $valuesArray.VersionMAJ = $tempvalue.HotFixID
```

Figure 8 - Enregistrement de l'historique des versions de mise à jour

La variable \$tempValue est de nouveau réinitialisée en mettant une nouvelle valeur à l'intérieur qui représente la liste de toutes les mises à jour réalisées sur le système.

```
58 # Addition des valeurs d'espace utilisé
59 $usedSpace = $valuesArray.EspaceUtilisé | Measure-Object -Sum
```

Figure 9 - Calcul du stockage utilisé

Dans l'intervalle des lignes de l'image ci-dessus (56-57) se trouve une valeur permettant de lancer un mail même si le stockage n'atteint pas 80% à des fins de débogage. Pour la ligne 59, l'instruction renvoie plusieurs valeurs.

```
EspaceUtilise {210161176576, 330605309952, 646030389248}
```

Figure 10 - Contenu de `$usedSpace`

Il faudra donc rajouter un « .Sum » à la fin de `$usedSpace` pour avoir la valeur voulue.

```
61 # Addition des valeurs d'espace libre
62 $freeSpace = $valuesArray.EspaceLibre | Measure-Object -Sum
```

Figure 11 - Calcul de l'espace libre entre tous les disques

La même stratégie est appliquée pour l'espace libre.

```
64 # Addition des 2 valeurs pour avoir la valeur totale
65 $totalSpace = $usedSpace.Sum + $freeSpace.Sum
```

Figure 12 - Calcul du stockage total

Vient ensuite l'addition de la somme du stockage plein et du stockage vide afin d'avoir le stockage total pour calculer le pourcentage du stockage utilisé en octets, qui est ensuite stocké dans une variable.

```
67 # Calcul du pourcentage d'utilisation du disque
68 $diskPercent = ($usedSpace.Sum*100)/$totalSpace
```

Figure 13 - Calcul en % de l'espace utilisé par rapport au stockage total

Le calcul pour récupérer le pourcentage est simple. On multiplie par 100 la somme du stockage utilisée, puis on divise par le stockage total. Par exemple, sur la machine de l'école en conditions réelles, le stockage est utilisé à environ 63.62%.

```
70 # Si l'espace disque est plus utilisé ou égal à 80%, entrer
71 if($diskPercent -ge 80){
```

Figure 14 - Condition pour faire fonctionner la suite du script

Le script effectue une vérification du pourcentage de stockage utilisé. S'il est de 80% ou plus, la suite du script s'exécute, sinon le script se termine. Tout ce qui suit est donc contenu dans le bloc de la condition. On pourrait améliorer les performances du script en faisant en sorte que le script aille chercher les infos que si le pourcentage de stockage autorisé est dépassé, comme ça pour la plupart du temps le script se fera plus vite.

```
73 # Crédentiels du serveur SMTP
74 $server = "smtp.gmail.com"
75 $port = 587
```

Figure 15 - Informations du serveur SMTP

On donne ici les premières informations importantes sur le serveur comme le port qui doit être à 587 pour fonctionner avec -UseSSL afin que Gmail permette d'envoyer le mail via leur serveur smtp.

```
77 # Détails de l'email
78 $sender = "papro2mail@gmail.com"
79 $recipient = "tpiymetml@gmail.com"
```

Figure 16 - Informations de l'expéditeur

Pour ce qui est de l'expéditeur et du destinataire, il s'agit juste de 2 strings classiques qui seront affectés par la suite dans le `Send-MailMessage` à la fin du script pour aérer un peu cette dernière commande.

```

80     $subject = "Report - Disque presque plein"
81     $body = "L'ordinateur " + $valuesArray.MachineNom + " est victime de surcharge.`n
82     Version de l'OS: " + $valuesArray.VersionOS + "`n
83     Versions de MAJ: " + $valuesArray.VersionMAJ+ "`n
84     Utilisation de l'espace (octets): " + $usedSpace.Sum + "`n
85     Espace libre (octets):" + $freeSpace.Sum + "`n"

```

Figure 17 - Sujet et corps de mail

Cette partie du code est la partie où le mail est rédigé \$ body pourrait être optimisé avec une boucle foreach.

```

87     # Identifiants (mot de passe d'application google)
88     $password = "fxkj gfff ebmt jprz"

```

Figure 18 - Mot de passe d'application Google utilisé

La variable du mot de passe est composée d'un mot de passe d'application généré aléatoirement par Google qui permet de se connecter à son compte Google depuis n'importe quelle application.

```

90     #Sécurisation du mot de passe
91     $securePassword = ConvertTo-SecureString -string $password -AsPlainText -Force

```

Figure 19 - Sécurisation du mot de passe

Le mot de passe Google est converti en string sécurisé, un type créé exprès pour les mots de passe. Sans ça, Google considérera que la connexion n'est pas sécurisée même avec SSL et le mail ne s'enverra pas. Comme le mot de passe est en clair dans le code ce n'est pas sécurisé.

```

93     $creds = New-Object System.Management.Automation.PSCredential -argumentlist $sender, $securePassword

```

Figure 20 - Création d'un objet de crédentité

Un objet spécial pour les logins est créé prenant en paramètre le nom d'utilisateur et le mot de passe d'un compte. Ici, le compte est l'adresse électronique, d'où le \$sender qui est utilisé, suivi par le mot de passe d'application Google.

```

95     # Envoi du mail
96     Send-MailMessage -SmtpServer $server -Port $port -UseSsl -Credential $creds -From $sender -To $recipient -Subject $subject -Body $body

```

Figure 21 - Envoi du mail via la commande Send-MailMessage

Envoi du mail en précisant le serveur smtp, son port, l'utilisation de SSL, l'adresse email et son mot de passe, puis une précision de l'expéditeur ainsi que du destinataire. Le sujet du mail et le corps sont définis aussi.

```

98     # Supprime les fichiers de plus de 3Gb contenant l'extension ova
99     Get-Childitem -Path "C:\\" -Recurse | Where-Object { $_.Extension -eq '.ova' -and $_.Length -gt 3GB } | Remove-Item -Force

```

Pour finir, le script supprime les fichiers OVA de plus de 3 Go de tout le disque.

```

101     Write-Host "Travail effectué, vous pouvez consulter votre boîte mail"
102 }

```

Le script se termine sur un message annonçant que le script est terminé.

4.2.3 Script 2

Le deuxième script se focalise sur une façon d'utiliser le premier. En effet, celui-ci copie le premier script depuis la machine locale et le colle dans un répertoire spécifique sur la ou les machines distantes (jusqu'à un maximum de 10). C'est par celui-ci que le projet a été commencé, jugé plus simple à implémenter et tester.

```
30      # Cree un paramètre tableau
31      param([string[]]$remoteip)
```

Figure 22 - Instanciation du tableau

Un tableau est créé comme paramètre afin de pouvoir mettre plusieurs adresses IP.

```
33      # Creation tableau
34      $computerList= New-Object System.Collections.ArrayList(10)
```

Figure 23 - Instanciation d'un deuxième tableau

Un autre tableau d'une taille de 10 éléments est créé afin de pouvoir stocker les paramètres entrés dans \$remoteip.

```
36      # Ajout de tous les ordinateurs dans le tableau
37      foreach($computerip in $remoteip){
38          $computerList.Add($computerip) | Out-Null
39      }
```

Figure 24 - placeholder - boucle foreach

La boucle foreach permet de naviguer dans le tableau \$remoteip et de récupérer toutes les adresses IP pour les ajouter dans \$computerList mentionné ci-dessus. Out-Null est utilisé afin de masquer l'output fait par la boucle à chaque passage dans celle-ci.

```
41      # Entrée du mot de passe à utiliser dans les credentials
42      $password = ConvertTo-SecureString(".Etml-123") -AsPlainText -Force
```

Figure 25 - Conversion du mot de passe

C'est dans cette partie du script que le mot de passe est choisi et stocké pour l'utiliser par la suite. Cette instruction cause des problèmes de sécurité car avoir un mot de passe en clair permet à n'importe quelle personne ayant accès au script de voir le mot de passe. Pour pallier à cette problématique, une solution serait de créer un fichier avec les mots de passe hachés à l'intérieur, puis de venir le récupérer avec une commande spécifique.

```
44      # Renseignement des credentials automatique
45      $credentials = new-object -typename System.Management.Automation.PSCredential -argumentlist "Administrateur", $password
```

Figure 26 - Récupération du nom d'utilisateur

Le nom d'utilisateur est récupéré ici. Comme le mot de passe et le nom d'utilisateur sont identiques pour les 2 machines distantes, ils sont renseignés en dur dans le script. Si ce n'était pas le cas, des paramètres auraient été

utilisés. Cela crée un problème de sécurité similaire à celui du mot de mot de passe mentionné précédemment.

```
47 # Instanciation des chemins (start path = $PWD + \x-yanmangiagl-monitoring.ps1)
48 $startPath = "D:\FIN22-24\FIN2\00-TPI\tpi\scripts\x-yanmangiagl-monitoring.ps1"
49 $destinationPath = "C:\Scripts"
```

Figure 27 - Implantation des chemins

Pour ce qui est des chemins, ils sont aussi en dur. Pour régler ce problème il faudra créer une valeur avec le nom du fichier et faire une concaténation avec \$PWD pour rendre le script plus indépendant de la machine utilisée. \$startpath est le chemin à optimiser, pour que le script respecte entièrement le cahier des charges. C'est lui qui va chercher le script 1 pour le copier plus tard. \$destinationPath est le chemin de destination où sera stocké le script copié plus tôt. La lettre C:\ a été choisie car c'est la lettre par défaut d'un stockage. On pourrait améliorer ça en permettant à l'administrateur de choisir le chemin de destination via un paramètre.

```
74 try{
75     foreach($computer in $computerList){
76         # Connexion à ordinateur distant
77         $remotename = [System.Net.Dns]::GetHostByAddress($computer).Hostname
```

Figure 28 - Début de la boucle

Cette boucle foreach commence par aller chercher le nom DNS des machines qui sera utile pour créer une session par la suite. C'est dans cette boucle qu'on utilise le tableau \$computerList créé plus tôt.

```
55 # Création d'une session
56 $session = New-PSSession -ComputerName $remotename -Credential $credentials
```

Une session est créée ici pour chaque passage dans la boucle décidé par le nombre d'éléments dans le tableau \$computerList cette session contient un id, un nom d'ordinateur, l'état de la connexion, le nom de la configuration qui s'apparente à comment la session a été configurée, et si la connexion est disponible ou non, comme le montre la capture d'écran ci-dessous.

```
PS H:\> $session
```

Id	Name	ComputerName	ComputerType	State	ConfigurationName	Availability
1	WinRM1	yann2k22srv	RemoteMachine	Opened	Microsoft.PowerShell	Available

Figure 29 - Informations d'une session ouverte

```
58 # Copie depuis machine locale du fichier et colle sur la machine distante
59 Copy-Item $startPath -Destination $destinationPath -ToSession $session
```

Cette instruction copie le fichier situé au chemin de départ et se connecte à l'ordinateur distant grâce à l'instruction -ToSession, puis colle le script dans le chemin de destination situé dans l'ordinateur distant.

```
64 # Concatenation du chemin
65 $fullpath = '-File ' + '"C:\Scripts\x-yanmangiagl-monitoring.ps1'"
```

Figure 30 - Argument servant à lancer la tâche planifiée

Écriture du chemin complet du script dans une variable afin de pouvoir l'utiliser comme argument pour créer une tâche planifiée par la suite à l'intérieur de « Invoke-Command ». Celle-ci permet d'exécuter une commande sur la machine distante d'une manière plus efficace qu'Enter-PSSession, car Enter PSSession doit d'abord se connecter à la machine

distante, tandis que `invoke-command` permet de rester sur la machine locale. Les guillemets simples permettent d'écrire les guillemets doubles et que ce ne soit pas considéré comme la façon d'écrire un string.

```
65     # Creation d'une tâche planifiée
66     $task = New-ScheduledTaskAction -Execute 'powershell.exe' -Argument $fullpath
```

Figure 31 - Création de la tâche planifiée

Une variable de tâche dénommée \$task est créée en utilisant New-ScheduledTaskAction. On précise que la tâche utilisera PowerShell en le notant dans -Execute et on précise le script qui sera lancé grâce à son chemin.

```
67 $taskInterval = New-ScheduledTaskTrigger -Daily -At 4PM
68 $taskName = "MonitoringAuto4PM"
```

Figure 32 - Création de la répétitivité de la tâche

`$taskInterval` consiste à décider de la fréquence à laquelle le script sera exécuté. Il s'exécutera donc tous les jours à 16h00. La suite du script donne un nom à la tâche et le stocke dans une variable du nom de `$taskName`.

```
69     $taskDesc = "Récupère les données d'un ordinateur, envoie des mails et supprime les fichiers trop volumineux"
70
71     # Enregistrement de la tâche
72     Register-ScheduledTask -TaskName $taskName -Action $task -Trigger $taskInterval -Description $taskDesc
73 }
74 }
```

Figure 33 - Description de la tâche et enregistrement de cette tâche

Le script donne une description à la tâche, puis l'enregistre définitivement ensuite. Les 2 accolades permettent de refermer invoke-command ainsi que le foreach.

4.3 Conception des tests

La stratégie de test employée est triviale ; Chaque ligne est testée indépendamment des autres pour avoir une suite d'instructions qui fonctionnent. Une fois ces instructions mises ensembles, elles sont retestées par groupe jusqu'à qu'elles fonctionnent elles-aussi. Cette opération sera effectuée tant que le script ne fonctionne pas entièrement.

4.4 Planification détaillée

La planification se lit de bas en haut, car c'est comme cela qu'apparaît la planification dans le journal de travail

[illegible]

Séquence 25			Date: vendredi, 31 mai 2024		13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Pause de 15 minutes	3	0h15min	Rapporter dans le rapport ce qui a été fait le matin		
			journal de travail		
			dépôt git + mail du rapport		
Total tranche	3	0h15min			

[illegible]

Séquence 23		Date: jeudi, 30 mai 2024		13:10 - 16:35
Tâche	Tranche [5min]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
	20	1h40min	Rapporter dans le rapport ce qui a été fait le matin dans l'analyse 4.1	
	17	1h25min	Avancer dans le bilan des fonctionnalités demandées 6.1	
	1	1		
Pause de 15 minutes	3	0h15min	Journal de travail	
Total tranche	41	3h25min		

Séquence 22			Date: jeudi, 30 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Pause de 15 minutes	3	0h15min		
Total tranche	3	0h15min		

Séquence 21			Date: mercredi, 29 mai 2024		08:00 - 11:25
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?		Liens, références, ...
Pause de 15 minutes	3	0h15min			
Séance de classe	9	0h45min	Séance de classe		
	1	0h5min	journal de travail		
Total tranche	13	1h5min			

Séquence 20			Date: mardi, 28 mai 2024		13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Pause de 15 minutes	3	0h15min	Rapporter dans le rapport ce qui a été fait le matin		
			journal de travail		
			dépôt git + mail du rapport		
Total tranche	3	0h15min			

[illegible]

Séquence 10			Date: mardi, 21 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Analyse - Planification initiale	1	0h5min	Analyse de ce qui a été fait la semaine précédente et comparaison avec la planification initiale	
Script 1 - Infos supplémentaires	41	3h25min	Se focaliser sur la façon de récupérer les maj effectuées et comment l'incorporer dans le script 1	
Total tranche	42	3h30min		

Séquence 9			Date: vendredi, 17 mai 2024	13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Script 1 - Courriel	21	1h45min	Avancée dans le script sur ce qui bloque	
Doc - Rapport	15	1h15min	Avancée dans le rapport	
Pause de 15 minutes	3	0h15min	Pause de 14h45 à 15h00	
Doc - Journal de travail	1	0h5min	journal de travail	
Réalisation - Github	1	0h5min	dépôt git + mail du rapport	
Total tranche	41	3h25min		

Séquence 8			Date: vendredi, 17 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	30	2h30min	Rédaction du tableau des tests à compléter tout le long du projet (ajout des futurs tests ainsi que des tests déjà effectués)	
Pause de 15 minutes	3	0h15min	Pause	
Script 1 - Infos supplémentaires	18	1h30min	Récupération du nom de l'ordinateur, de la version du système d'exploitation, l'espace disque utilisé et les versions des mises à jour du système d'exploitation	
Total tranches	61	4h15min		

Séquence	7		Date: jeudi, 16 mai 2024	13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	21	1h45min	Rédaction partie Réalisation	
Doc - Rapport	15	1h15min	Remplir le champ modification selon ce qui a été envoyé mardi 14 mai	
Doc - Journal de travail	2	0h10min	journal de travail	
Pause de 15 minutes	3	0h15min	Pause de 14h45 à 15h00	
Total tranche	41	3h25min		

Séquence	6		Date: jeudi, 16 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	35	2h55min	Si pas commencé plus tôt, commencer à rédiger partie analyse du rapport (3.1 à 3.3)	
Script 2 - Copie	11	0h55min	Documentation sur comment créer une liste d'ordinateurs sur laquelle avancer dans le script + test à part	
Doc - Journal de travail	2	0h10min	Journal de travail + dépôt / envoi mail du rapport	
	3	0h15min	Pause de 9h35 à 9h50	
Total tranche	51	4h15min		

Séquence 5			Date: mercredi, 15 mai 2024	08:00 - 11:25
Tâche	Tranche [5min]	Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Tests	9	0h45min	Tester si ce qui a été fait la veille sur les 2 scripts fonctionne bien	
Doc - Tests	3	0h15min	Tenter d'envoyer un mail quand le stockage est utilisé à plus de 1% (au lieu des 80) pour tester si ça marche bien	
Doc - Tests	16	1h20min	Regroupement de tout ce qui a été testé dans son script respectif et corriger les éventuels problèmes	
Doc - Journal de travail	1	0h5min	Journal de travail	
	3	0h15min	Pause	
Séance de classe	9	0h45min		
Total tranche	41	3h25min		

Séquence 4			Date: mardi, 14 mai 2024		13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Schéma	2	0h10min	Fin de la création du schéma conceptuel		
Script 2 - Copie	10	0h50min	Documentation sur les prérequis afin de créer une copie d'un script sur un poste à distance et tester		
Script 1 - Courriel	10	0h50min	Se remettre sur le script du mail fait pendant p_appro2 pour voir s'il marche en essayant d'envoyer un mail		
Script 1 - Espace disque	4	0h20min	Se renseigner sur comment récupérer le taux de stockage utilisé sur un disque		
Doc - Journal de travail	2	0h10min	Total du temps pris sur le journal de travail en modifiant au fur à mesure + dépôt git / mail rapport		
	3	0h15min	Pause		
Total tranche	31	2h35min			

Séquence 3			Date: mardi, 14 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	4	0h20min	Création d'une page de synthèse résumant le rapport à mettre dans celui-ci		
Doc - Rapport	8	0h40min	Partie Spécifications		
Script 2 - Tâche planifiée	11	0h55min	Documentation sur les prérequis afin de créer une tâche planifiée avec powershell et faire un petit script de test		
Script 1 - Espace disque	11	0h55min	Documentation sur les prérequis afin de supprimer des fichiers de plus de 3Go avec l'extension OVA et faire un petit script de test		
Analyse - Schéma	9	0h45min	Création du schéma conceptuel des 2 scripts (2 schémas différents se rejoignant)		
	8	0h40min	Stratégie de test et rédaction de l'analyse du tpi		
Total tranche	51	4h15min			

Séquence 2			Date: lundi, 13 mai 2024		13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Réalisation - Github	3	0h15min	Création du repo git (public) et l'organisation de celui-ci		
	1	0h5min	Lecture du mail de l'expert 1 avec toutes les informations discutées le matin		
	11	0h55min	Rédaction de la partie de la planification initiale dans le rapport de projet		
	10	0h50min	Continuation de la planification initiale		
	1	0h5min	Envoi de la planification initiale par mail		
	2	0h10min	Total du temps pris sur le journal de travail en modifiant au fur à mesure		
	3	0h15min	Pause		
Total tranche	31	2h35min			

Séquence 1			Date: lundi, 13 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Rencontre expert	7	0h35min	Rencontre avec l'expert 1, questions et revue du travail à accomplir		
Analyse - Planification initiale	37	3h5min	Création et estimation de la planification initiale		
Analyse - Entretien CDP	2	0h10min	Rencontre avec chef de projet pour confirmer ce qui a été discuté avec expert 1 et poser de plus amples questions		
	2	0h10min	Création du mail pour envoyer la planification initiale à 15h45		
	3	0h15min	Pause		
Total tranche	51	4h15min			

5 RÉALISATION

5.1 Dossier de Réalisation

5.1.1 Version des outils logiciels utilisés

Machine locale

Logiciel	Version
PowerShell Desktop	5.1.19041.4291

Yann2k22srv

Logiciel	Version
PowerShell Desktop	5.1.20348.558

Yann2k22srv2

Logiciel	Version
PowerShell Desktop	5.1.20348.558

5.1.2 Comptes utilisés

Utilisation	Compte	Mot de passe
Envoi du mail automatisé	pappro2mail@gmail.com	.Etml-123
Réception du mail automatisé	tpiymetml@gmail.com	.Etml-123
Compte serveur YANN2K22SRV	Administrateur	.Etml-123
Compte serveur YANN2K22SRV2	Administrateur	.Etml-123

5.1.3 Configuration spéciale des serveurs

Tout ce qui n'est pas précisé est un paramètre par défaut. Les 3 machines ont la police d'exécution déterminée sur « Unrestricted »

Les 2 machines distantes sont configurées de la même manière, la seule différence est que yann2k22srv2 a une autre adresse ip (192.168.10.53). Toutes les captures d'écran proviennent de yann2k22srv. Les deux machines ont leur pare-feu désactivé.

ID de périphérique	E914073B-C091-44DC-BC04-31E596B1AC4F
ID de produit	00454-40000-00001-AA735
Type du système	Système d'exploitation 64 bits, processeur x64
Stylet et fonction tactile	La fonctionnalité d'entrée tactile ou avec un stylet n'est pas disponible sur cet écran

Copier

Renommer ce PC

Spécifications de Windows

Édition	Windows Server 2022 Standard Evaluation
Version	21H2
Installé le	26.04.2024
Build du système d'exploitation	20348.587

Copier

Figure 34 - Spécifications de Windows Server

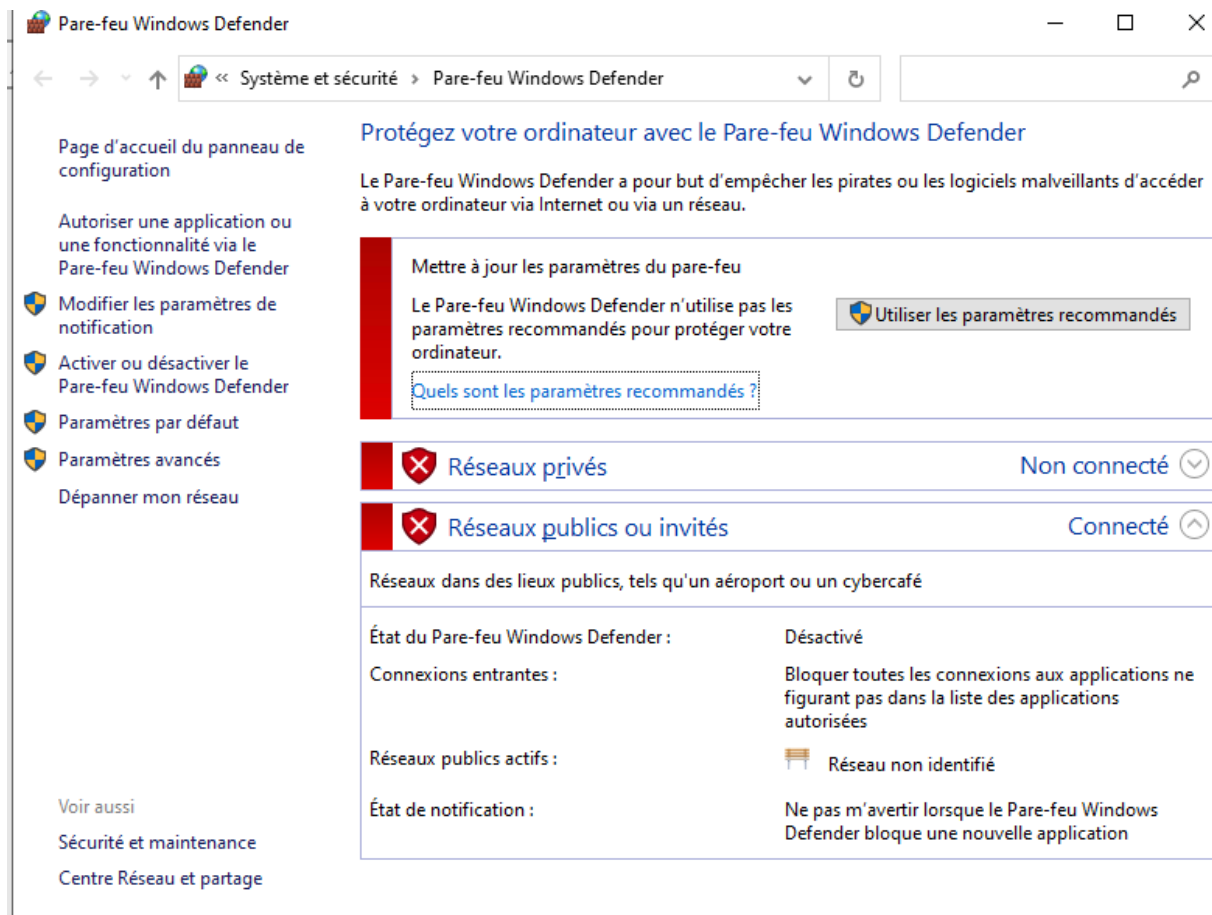


Figure 35 - Configuration du pare-feu

Les pare-feux sont désactivés car ils bloquent l'utilisation de la commande

```
$remotenam = [System.Net.Dns]::GetHostByAddress($remoteip).Hostname
```

Cela empêche donc le script 2 de résoudre le DNS et de récupérer le nom de la machine qui est utilisé par la suite pour créer une session dans laquelle il sera possible de coller le script 1. En bloquant le pare-feu, on s'affranchit des problèmes du genre. Malgré tout, cela cause d'énormes problèmes de sécurité. Le plus optimal serait de créer une règle qui autoriserait la commande de s'exécuter, et dans le même temps il faudrait désactiver toutes les règles qui coupent le script.

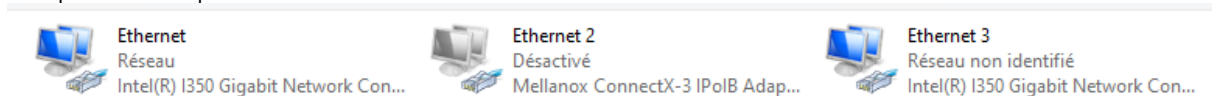


Figure 36 - Configuration des cartes réseau

La partie réseau est décidée par le **réseau** de l'école et la partie hôte a été décidée arbitrairement. Comme c'est un masque /24, seulement la dernière partie est une partie hôte. Voici un tableau regroupant les adresses IP utilisées et à quelles machines elles correspondent :

yann2k22srv	192.168.10.51
INF-A13-M215	192.168.10.52
yann2k22srv2	192.168.10.53

La carte réseau activée a 2 connexions : une pour que la machine soit connectée à internet et l'autre pour avoir un réseau LAN avec la machine physique.

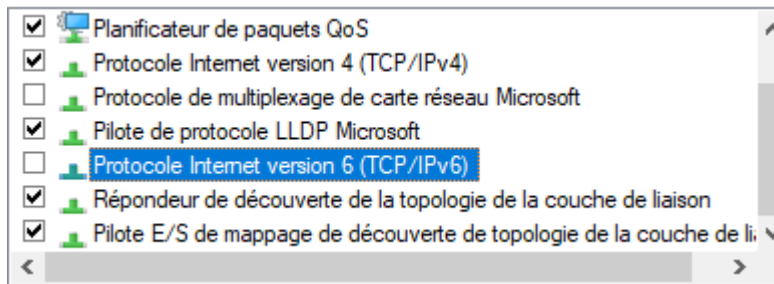


Figure 37 - Protocole IPV6 désactivé

Le protocole IPV6 est désactivé afin d'éviter les problèmes d'incompatibilité qui pourraient survenir en utilisant winRM.

Bureau à distance

Le Bureau à distance vous permet de vous connecter à ce PC et de le contrôler à partir d'un appareil à distance à l'aide d'un client Bureau à distance (disponible pour Windows, Android, iOS et macOS). Vous pourrez travailler à partir d'un autre appareil comme si vous travailliez directement sur ce PC.

Activer le Bureau à distance



Activé



Garder mon PC prêt pour la connexion quand il est branché

[Afficher les paramètres](#)



Rendre mon PC détectable sur des réseaux privés pour permettre la connexion automatique à partir d'un périphérique distant

[Afficher les paramètres](#)

Figure 38 - Activation du bureau à distance

Il a fallu activer le bureau à distance sur les serveurs physiques afin de pouvoir s'y connecter plus tard.

```
PS C:\Users\Administrateur\desktop> Get-Item WsMan:\localhost\Client\TrustedHosts

WSManConfig : Microsoft.WSMan.Management\WSMan::localhost\Client

Type      Name              SourceOfValue      Value
----      -
System.String TrustedHosts      *

```

Figure 39 - Configuration de WinRM

Pour configurer celui-ci, il suffit d'ajouter les autres machines comme des hôtes de confiance (TrustedHosts avec PowerShell). Si le pare-feu n'est pas désactivé, cela ne fonctionnera pas. La valeur est une étoile pour autoriser n'importe quelle machine. La prochaine étape est de créer un mot de passe d'application pour utiliser un compte Google. Pour créer un mot de passe d'application, il faut accéder à l'A2F de Google dans les réglages du compte, puis sélectionner « mot de passes d'application ». Il suffit ensuite d'entrer un nom et d'appuyer sur « créer »

Vos mots de passe d'application

TPI	Date de création : 21 mai, dernière utilisation : 08:31	🗑️
-----	---	----

Pour créer un mot de passe spécifique à une appli, indiquez son nom ci-dessous.

Créer

Figure 40 - Page d'accueil de la page de création de mots de passe d'application Google

Google générera un mot de passe composé de 19 caractères en comptant les espaces comme celui ci-dessous.

Mot de passe d'application généré

Mot de passe d'application pour votre appareil

fxkj gfff ebmt jprz

Comment l'utiliser ?
Accédez aux paramètres de votre compte Google dans l'application ou l'appareil que vous essayez de configurer. Remplacez le mot de passe par celui de 16 caractères indiqué ci-dessus.
Tout comme votre mot de passe classique, ce mot de passe spécifique à une application permet d'accorder un accès complet à votre compte Google. Étant donné que vous n'avez pas besoin de le mémoriser, ne le notez nulle part ni ne le partagez avec personne.

OK

Figure 41 - Mot de passe généré

C'est un mot de passe comme celui-ci qu'il faut entrer dans le script pour que celui-ci puisse se connecter à Google.

5.1.4 Déploiement et exécution détaillée

Maintenant que l'environnement est apte à accueillir et exécuter tout ce que les scripts sont susceptibles de lui demander, il faut effectuer la procédure suivante :

- 1) Avoir les 2 scripts sur la machine dans le même répertoire de la machine locale.
- 2) Lancer ensuite le script nommé « x-yanmangiagl-setup.ps1 » avec en argument les adresses IP des machines sur lesquelles vous voulez que le script 1 s'installe.
- 3) Par défaut, le script 2 crée un dossier nommé Scripts se trouvant dans la partition C. (C:\Scripts). Pour le changer il faut aller changer sa valeur dans le script directement.
- 4) Si aucune adresse n'est renseignée, le script 2 copiera le script monitoring dans C:/Scripts sur la machine qui exécute le script 2.
- 5) Le script 1 va se copier dans le dossier créé et créer une tâche planifiée qui fera fonctionner le script tous les jours à 16h00.

5.1.5 En tête commune aux 2 scripts

```
1 <#
2 .NOTES
3     Nom: x-yanmangiagl-setup.ps1
4     Auteur: Yann Mangiagli
5     Date de création: 15 mai 2024
6     Date de modification 1 : 17 mai 2024
7     Raison: Changement de commentaires + changements légers pour respecter les conventions
8     Date de modification 2 : 24 mai 2024
9     Raison: Debug pour faire apparaître le chemin du script correctement dans les arguments
10    + ajouts de try catch
11    Date de modification 3 : 27 mai 2024
12    Raison: Ajout de Write-Host pour dire que l'action a été réalisée
13
14 .SYNOPSIS
15     Copie / colle un script et crée une tâche planifiée
16
17 .DESCRIPTION
18     Copie un script nommé à un chemin spécifique et le colle dans un pc distant
19     Crée une tâche planifiée permettant de lancer le script installé plus tôt
20
21 .EXAMPLE
22     .\x-yanmangiagl-setup.ps1 -remoteip "192.168.10.51"
23
24 .EXAMPLE
25     .\x-yanmangiagl-setup.ps1 -remoteip "192.168.10.51","192.168.10.53"
26
27 .PARAMETER ComputerList
28     Liste des ordinateurs sur lesquels le script effectuera la copie
29
30 .LINK
31     https://github.com/Yann-Mangiagli/tpi/tree/main/scripts
32 #>
```

Figure 42 - En-tête du script 2

L'en-tête est utilisé pour fournir des explications sur le script à la personne qui l'utilise. Le .Notes communique des informations sur l'auteur, les modifications du script ainsi que les raisons. .SYNOPSIS donne ce que fait le script en gros, tandis que .DESCRIPTION décrit comment le script effectue le synopsis. .EXAMPLE donne un exemple, .PARAMETER les paramètres s'ils existent et .LINK un lien important pour le script. L'en-tête se trouve dans les 2 scripts, avec quelques différences. En effet, contrairement au script 1, le script 2 contient des paramètres. Cela veut dire que le .Parameter n'est pas nécessaire dans l'en-tête du script 1.

5.1.6 Réalisation du script de setup

Comme dit plus haut dans la partie analyse, le script 2 a été réalisé avant le premier, ce qui a causé un problème au moment de la réalisation du script 1 mais ce sera expliqué plus tard dans ce rapport. Le script a d'abord été fait avec le minimum requis. C'est-à-dire la copie d'un item dans une session, la création d'une tâche et des chemins en dur. J'ai eu des

problèmes en réalisant la partie de la copie, car une erreur mentionnant un chemin vide était affichée. J'ai d'abord essayé de créer une liste d'arguments dans le invoke-command, mais ça ne fonctionnait pas non plus. Il s'est avéré que le problème venait du fait que la commande qui posait problème devait être en dehors du bloc d'invoke-command, et que je devais utiliser ToSession pour faire la copie du fichier plutôt qu'essayer de me connecter à la session depuis cette même session, vu que c'est comme ça qu'Invoke-Command fonctionne.

```
PS D:\FIN22-24\FIN2\00-TPI\tpi\scripts> Invoke-Command -Session $session -ScriptBlock {Copy-Item $startPath -Destination $destinationPath}
Impossible de lier l'argument au paramètre « Path », car il a la valeur Null.
+ CategoryInfo          : InvalidData : (:) [Copy-Item], ParameterBindingValidationException
+ FullyQualifiedErrorId : ParameterArgumentValidationErrorNullNotAllowed,Microsoft.PowerShell.Commands.CopyItemCommand
+ PSComputerName        : YANN2K22SRV

PS D:\FIN22-24\FIN2\00-TPI\tpi\scripts> Invoke-Command -Session $session -ScriptBlock {Copy-Item -Path $startPath -Destination $destinationPath}
Impossible de lier l'argument au paramètre « Path », car il a la valeur Null.
+ CategoryInfo          : InvalidData : (:) [Copy-Item], ParameterBindingValidationException
+ FullyQualifiedErrorId : ParameterArgumentValidationErrorNullNotAllowed,Microsoft.PowerShell.Commands.CopyItemCommand
+ PSComputerName        : YANN2K22SRV
```

Figure 43 - Le message d'erreur qui apparaissait avant de modifier copy-item

```
30 + # Cree un parametre tableau
31 + param([string[]]$remoteip)
32 +
33 + # Creation tableau
34 + $computerList= New-Object System.Collections.ArrayList(10)
35 +
36 + # Ajout de tous les ordinateurs dans le tableau
37 + foreach($computerIP in $remoteip){
38 +     $computerList.Add($computerIP) | Out-Null
39 + }
40 +
41 + # Entrée du mot de passe à utiliser dans les credentials
42 + $password = ConvertTo-SecureString(".Etm1-123") -AsPlainText -Force
43 +
44 + # Renseignement des credentials automatique
45 + $credentials = new-object -typename System.Management.Automation.PSCredential -argumentlist "Administrateur", $password
46 +
47 + # Instanciation des chemins (start path = $PWD + \x-yanmangiagl-monitoring.ps1) à changer après si y'a le temps
48 + $startPath = "D:\FIN22-24\FIN2\00-TPI\tpi\scripts\x-yanmangiagl-monitoring.ps1"
49 + $destinationPath = "C:\Scripts"
50 +
51 + foreach($computer in $computerList){
52 +     # Connexion à ordinateur distant
53 +     $remotename = [System.Net.Dns]::GetHostByAddress($computer).Hostname
54 +
55 +     # Création d'une session
56 +     $session = New-PSSession -ComputerName $remotename -Credential $credentials
57 +
58 +     # Copie depuis machine locale du fichier et colle sur la machine distante ou la liste d'ordinateur
59 +     Copy-Item $startPath -Destination $destinationPath -ToSession $session
60 +
61 +     Invoke-Command -Session $session -ScriptBlock {
62 +         # Concatenation du chemin
63 +         $fullpath = $destinationPath + '\x-yanmangiagl-monitoring.ps1'
```

Figure 44 - Première partie du code source original

```
65 + # Creation d'une tâche planifiée
66 + $task = New-ScheduledTaskAction -Execute 'powershell.exe' -Argument $fullpath
67 + $taskInterval = New-ScheduledTaskTrigger -Daily -At 4PM
68 + $taskName = "MonitoringAuto4PM"
69 + $taskDesc = "Récupère les données d'un ordinateur, envoie des mails et supprime les fichiers trop volumineux"
70 +
71 + # Enregistrement de la tâche
72 + Register-ScheduledTask -TaskName $taskName -Action $task -Trigger $taskInterval -Description $taskDesc
73 + }
74 + }
```

Figure 45 - Seconde partie du code source original

On peut voir dans cette première version plusieurs problèmes ; les chemins en dur ne respectent pas le cahier des charges et il n'y a aucune gestion d'erreur. Les exemples aussi sont faux, c'est la première chose qui est modifiée.

17	.EXAMPLE	17	.EXAMPLE
18	- .\x-yanmangiagl-setup.ps1 -ComputerList "yann2k22srv"	18	+ .\x-yanmangiagl-setup.ps1 -remoteip "192.168.10.51"
19		19	
20	.EXAMPLE	20	.EXAMPLE
21	- .\x-yanmangiagl-setup.ps1 -ComputerList "yann2k22srv","yann2k22srv2"	21	+ .\x-yanmangiagl-setup.ps1 -remoteip "192.168.10.51","192.168.10.53"

Figure 46 - Modification des exemples

Cependant, un problème inattendu s'est déclaré qui a forcé une nouvelle version du script. En effet, le script 1 était bien ajouté en argument de la tâche planifiée, mais elle n'avait pas le bon format. Celui-ci devait non seulement contenir le chemin entier, mais en plus il devait contenir un argument nommé -File pour que le script s'exécute bien correctement.

62	# Concatenation du chemin	64	# Concatenation du chemin
63	- \$fullpath = \$destinationPath + '\x-yanmangiagl-monitoring.ps1'	65	+ \$fullpath = '-File "' + "C:\Scripts\x-yanmangiagl-monitoring.ps1"

Figure 47 - Modification de l'argument du script 2

J'ai su que je devais faire ça en créant une tâche planifiée manuellement sur yann2k22srv, et dans les options du déclencheur de tâche, voilà ce qui était montré :

Programme/script :

powershell.exe Parcourir...

Ajouter des arguments (facultatif) : -File "C:\scripts\x-yanmangiagl-monitoring.ps1"

Figure 48 - Résultat voulu de \$fullpath

Une fois cette erreur corrigée, j'en ai profité pour mettre en place la gestion d'erreurs et les limites du script 2.

```

40 + if($remoteip -eq '')
41 + {
42 +     # Donne l'adresse de l'ordinateur local si aucune
      adresse ip n'est ajoutée.
43 +     $remoteip = "127.0.0.1"
44 +
45 +     # Convertit l'adresse IP en nom d'hôte
46 +     $remotename =
      [System.Net.Dns]::GetHostByAddress($remoteip).Hostname
47 }

```

Figure 49 - Ajout de gestion d'erreurs pour la conversion DNS

Si aucune adresse IP n'est spécifiée, le diagnostic est fait sur la machine locale. Attention cependant, cela ne fonctionnera pas sur la machine locale de l'école INF-A13-M215 car elle reste gérée par l'école qui bloque la requête du script.

```

49 + if($remoteip.Count -le 10){
50 +     # Ajout de tous les ordinateurs dans le tableau
51 +     foreach($computerip in $remoteip)
52 +     {
53 +         $computerList.Add($computerip) | Out-Null
54 +     }
55 + }else{
56 +     Write-Error "Maximum 10 adresses"
57 + }

```

Figure 50 - Ajout d'une limite de 10 machines

Comme il est demandé dans le cahier des charges que le nombre d'adresses maximum soit de 10 machines, le script limite le tableau d'adresses IP à 10 éléments maximum.

<pre> 49 - # Instanciation des chemins !/\ à changer après 50 - \$startPath = "D:\FIN22-24\FIN2\00-TPI\tpi\scripts\x- ynmangiagl-monitoring.ps1" </pre>	<pre> 65 + # 66 + \$scriptName = "\x-ynmangiagl-monitoring.ps1" 67 + 68 + # Instanciation des chemins 69 + \$startPath = \$PSScriptRoot + \$scriptName </pre>
---	---

Figure 51 - Modification d'un chemin en dur en chemin automatique

Ici, le chemin a été changé pour être automatique avec la variable `$PSScriptRoot` qui récupère le chemin duquel le script est exécuté, ce qui permet de se débarrasser d'un chemin en dur. De plus, ce chemin en dur causait des problèmes pour créer la tâche planifiée comme il faut. Un test a été fait pour s'assurer qu'on pouvait concaténer les valeurs ce qu'on ne peut pas faire avec `$PWD` par exemple.

```

PS D:\FIN22-24\FIN2\00-TPI\tpi\scripts> $testvalue = '$PWD + D:\x-p_appro2-serverproof2.ps1'
PS D:\FIN22-24\FIN2\00-TPI\tpi\scripts> $testvalue
$PWD + D:\x-p_appro2-serverproof2.ps1
PS D:\FIN22-24\FIN2\00-TPI\tpi\scripts> $PSScriptRoot

```

Figure 52 - La concaténation ne se fait pas

```

PS D:\FIN22-24\FIN2\00-TPI\tpi\scripts> D:\FIN22-24\FIN2\00-TPI\tpi\scripts\scriptdetest.ps1
D:\FIN22-24\FIN2\00-TPI\tpi\scripts + test
PS D:\FIN22-24\FIN2\00-TPI\tpi\scripts> D:\FIN22-24\FIN2\00-TPI\tpi\scripts\scriptdetest.ps1
D:\FIN22-24\FIN2\00-TPI\tpi\scriptstest

```

Figure 53 - La concaténation se fait

<pre> 53 - foreach(\$computer in \$computerList){ </pre>	<pre> 72 + # Vérifie que la connexion peut être faite 73 + try{ 74 + foreach(\$computer in \$computerList){ </pre>
--	--

Figure 54 - Ajout d'une gestion d'erreurs pendant la connexion

Une gestion d'erreur est effectuée avec un try catch pour sécuriser tout le code, même si cela ralentira un peu l'exécution du code.

```

88 +      # Vérifie que la tâche existe pas et l
      elle existe
89 +      $pathExists = Test-Path -Path
      "C:\Windows\System32\Tasks\MonitoringAuto4PM"
90 +      if($pathExists){
91 +      Remove-Item -Path
      "C:\Windows\System32\Tasks\MonitoringAuto4PM"
92 +      }
93 +
94 +      # Création de la tâche

```

Figure 55 - Vérification de si la tâche existe déjà, et la supprime si c'est le cas

Avant de créer la tâche, j'ai incorporé un test permettant de checker si la tâche existe déjà en me basant sur le chemin par défaut de Windows.

```

103 + }
104 + }catch{
105 +     Write-Error "Au moins une adresse IP est incorrecte."
106 +     Exit

```

Figure 56 – Bloc de code si une adresse IP est incorrecte

Si le script a une erreur, par exemple une adresse IP non valide, et bien un message d'erreur a été rajouté. Je pourrais améliorer ça en faisant en sorte que le script mette de côté cette adresse IP et fasse quand même les adresses IP valides plutôt que juste quitter le programme. Le script est donc fonctionnel comme démontré ci-dessous :

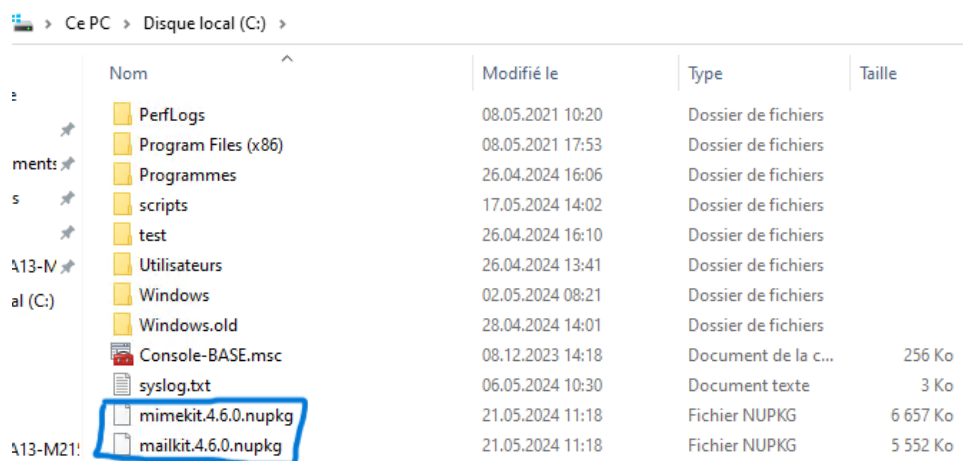
TaskPath	TaskName	State	PSComputerName
-----	-----	-----	-----
\\	MonitoringAuto4PM	Ready	YANN2K22SRV
\\	MonitoringAuto4PM	Ready	YANN2K22SRV2

Figure 57 - Sortie du script une fois les tâches créées

5.1.7 Réalisation du script de maintenance

Le script de maintenance est le script qui se charge de vérifier le stockage et de récupérer les variables qui seront envoyées par mail si le stockage atteint 80%. Il y'a eu pas mal de problèmes au début pour faire fonctionner l'envoi de mail qui m'a bloqué pendant quasiment une journée complète. En effet, j'avais réalisé en amont du projet un script qui envoyait un mail sécurisé avec MimeKit et MailKit en utilisant Nuget, qui est un gestionnaire de package pour .NET ¹. Nuget a donc été installé avec MailKit et MimeKit.

¹ Citation venant de Microsoft Learn



Ce PC > Disque local (C:) >				
	Nom	Modifié le	Type	Taille
	PerfLogs	08.05.2021 10:20	Dossier de fichiers	
	Program Files (x86)	08.05.2021 17:53	Dossier de fichiers	
	Programmes	26.04.2024 16:06	Dossier de fichiers	
	scripts	17.05.2024 14:02	Dossier de fichiers	
	test	26.04.2024 16:10	Dossier de fichiers	
	Utilisateurs	26.04.2024 13:41	Dossier de fichiers	
	Windows	02.05.2024 08:21	Dossier de fichiers	
	Windows.old	28.04.2024 14:01	Dossier de fichiers	
	Console-BASE.msc	08.12.2023 14:18	Document de la c...	256 Ko
	syslog.txt	06.05.2024 10:30	Document texte	3 Ko
	mimekit.4.6.0.nupkg	21.05.2024 11:18	Fichier NUPKG	6 657 Ko
	mailkit.4.6.0.nupkg	21.05.2024 11:18	Fichier NUPKG	5 552 Ko

Figure 58 - Installation de MimeKit et MailKit

Cependant, une erreur proclamait qu'il manquait des dépendances et le script s'arrêtait. Il a donc été décidé que Send-MailMessage serait utilisé, mais ce n'était pas plus facile pour autant.

```

25 + # Tableau de valeurs
26 + $valuesArray = @{
27 +     MachineNom = "x";
28 +     VersionOS = "x";
29 +     EspaceUtilise = "x";
30 +     EspaceLibre = "x";
31 +     VersionMAJ = "x"
32 + }
33 +
34 + # Récupération du nom de l'ordinateur
35 + $valuesArray.MachineNom = $env:COMPUTERNAME
36 +
37 + # Récupération de la version de l'OS
38 + $valuesArray.VersionOS = $env:OS + " " + [environment]::OSVersion.Version.Build
39 +
40 + # Récupération de l'espace disque libre
41 +
42 + # Récupération de l'espace disque utilisé
43 +
44 + # Récupération des versions de mise à jour de l'OS
45 +
46 + # Calcul du pourcentage d'utilisation du disque
47 +
48 + # Si l'espace disque est plus utilisé ou égal à 80%, entrer
49 +
50 +     # Crédentiels du serveur SMTP
51 +     $server = "smtp.gmail.com"
52 +     $port = 587
53 +
54 +     # Détails de l'email
55 +     $sender = "pappro2mail@gmail.com"
56 +     $recipient = "tpiymetml@gmail.com"
57 +     $subject = "Report - Disque saturé"
58 +     $body = ""

```

Figure 59 - Première partie du script 1

```

60 +     # Identifiants
61 +     $password = "fxkj gfff ebmt jprz"
62 +
63 +     #Sécurisation du mot de passe
64 +     $securePassword = ConvertTo-SecureString -string $password -AsPlainText -Force
65 +
66 +     $creds = New-Object System.Management.Automation.PSCredential -argumentlist $sender, $securePassword
67 +
68 +     # Envoi du mail
69 +     Send-MailMessage -SmtpServer $server -Port $port -UseSsl -Credential $creds -From $sender -To
        $recipient -Subject $subject -Body $body

```

Figure 60 - Seconde partie du script 1

Dans cette première version du script, seules les variables d'environnement et les valeurs nécessaires à l'envoi du mail sont réalisées, car c'est ce que j'ai eu le temps de faire à cause du blocage créé à cause de ma première idée avec NuGet, puis les problèmes qui ont suivi avec Send-MailMessage.

"Send-MailMessage : The SMTP server requires a secure connection or the client was not authenticated.
The server response was: 5.7.0 Authentication Required."

Figure 61 - Capture d'écran du message d'erreur dû à Google

Ce message d'erreur a été un grand ralentissement dans ma progression, car j'avais renseigné le bon port, le bon mot de passe, le bon nom d'utilisateur, et spécifié le SSL, désactivé le pare-feu du serveur et la machine était connectée à Internet. Ce qui faisait bloquer le mail était qu'il fallait en faire un secure-string. J'avais pensé que l'instruction créait un objet identitaire le faisait nativement, mais non. Google ne considérait la connexion sans sécurité, même si elle était en SSL. J'ai donc dû continuer le script le jour suivant.

```
31 + VersionMAJ = "x";
32 + #LettreDisk = "x"
```

Figure 62 - Enlèvement de la lettre du disque

J'ai commencé par enlever la lettre du disque car il n'y en a pas besoin dans l'immédiat, puis j'ai ajouté les valeurs qui manquaient pour l'espace libre et utilisé.

```
41 + # Récupération de l'espace disque libre et de la lettre
42 + $valuesArray.EspaceLibre = Get-PSDrive | Select-Object free # tester -ExcludeProperty jeudi
43
44 # Récupération de l'espace disque utilisé
45 + $valuesArray.EspaceUtilise = Get-PSDrive | Select-Object Used
46
47 # Récupération des versions de mise à jour de l'OS
48 + $valuesArray.VersionMAJ = Get-HotFix -ComputerName $valuesArray.MachineNom | Select HotFixID
49
50 # Calcul du pourcentage d'utilisation du disque
51 + $diskPercent = 80
```

Figure 63 - Ajout des valeurs dans le script

Je ne l'ai pas implémenté correctement, il fallait une variable ou préciser, par exemple \$valuesArray.EspaceUtilise.Used, mais ça faisait répétitif et ce n'était pas très lisible. J'ai corrigé cela plus tard.

```
54 + if($diskPercent -ge 80){
55 +
```


Figure 64 - Création de la condition de 80%

Un début de code pour la suppression des fichiers volumineux est créé. C'est dans ce bloc que sera rajouté plus tard toutes les commandes de suppression.

```
64 + $body = "L'ordinateur " + $valuesArray.MachineNom + " est surchargé.`n
65 + Version de l'OS: " + $valuesArray.VersionOS + "`n
66 + Versions de MAJ: " + $valuesArray.VersionMAJ+ "`n" #Mettre valeur qui a été créée dans un foreach plus haut
```


Figure 65 - Création du corps du mail

Le corps du mail est rempli pour ne plus être vide, et on y met déjà les valeurs qu'on veut recevoir dans le mail.

 **papro2mail@gmail.com**
 À tpiymetml ▾
 L'ordinateur INF-A13-M215 est surchargé?

 Version de l'OS: Windows_NT 19045

 Versions de MAJ: x

 **papro2mail@gmail.com**
 À tpiymetml ▾

 L'ordinateur INF-A13-M215 est surchargé?

 Version de l'OS: Windows_NT 19045

 Versions de MAJ:

Figure 66 - Mails de tests reçus

Dans ce mail cependant, on peut remarquer un problème d'encodage. Pour contourner le problème, j'ai décidé de reformuler mon texte pour ne pas avoir à me soucier de ce problème handicapant, comme dans cet exemple.

79 - \$body = "L'ordinateur " + \$valuesArray.MachineNom + " est surchargé.`n"	78 + \$body = "L'ordinateur " + \$valuesArray.MachineNom + " est victime de surcharge.`n"
---	---

Figure 67 - Contournement des problèmes d'encodage

Les autres changements sont plus importants ; L'ajout des versions de mise à jour et la suppression des fichiers OVA de plus de 3 Gb ².

80 + Versions de MAJ: " + \$valuesArray.VersionMAJ+ "`n"

Figure 68 - Ajout des versions de MAJ

83 + Versions de MAJ: " + \$valuesArray.VersionMAJ+ "`n
 84 + Utilisation de l'espace (octets): " + \$usedSpace.Sum + "`n
 85 + Espace libre (octets): " + \$freeSpace.Sum + "`n"

Figure 69 - Corps du mail final

L'espace libre et l'utilisation de l'espace sont bien notés d'une manière qui fonctionne, et plus lisible qu'avant.

94 + Get-ChildItem -Path "C:\\" -Recurse | Where-Object { \$_.Extension -eq '.ova' -and
 \$_.Length -gt 3GB } | Remove-Item -Force

Figure 70 - Commande permettant de supprimer les fichiers ova de plus de 3Go

La commande cherche dans tous les dossiers que contient C : ainsi que C : lui-même tous les fichiers dont l'extension est .ova, et qui pèsent plus de 3Gb, puis les supprime. Le -Force est là pour effacer les .ova qui pourraient être cachés. Pour améliorer le script il serait possible de mettre une valeur plutôt que '.ova', pourquoi pas dans un foreach pour rajouter d'autres extensions, comme les .avi par exemple.

² Instruction de suppression générée par Bing Copilot et modifiée pour convenir au contexte

5.2 Modifications

Aucune modification n'a été demandée par le chef de projet ou les experts. Il y'a eu un conseil de mettre en forme les mises à jour en tant que liste dans le mail plutôt que la forme « à la suite » qui est donnée afin de rendre le tout plus lisible. Après avoir essayé de l'implémenter sans succès, il a été décidé de revenir dessus plus tard s'il restait du temps.

6 TESTS

6.1 Dossier des tests

Numéro	Description	Procédure	Résultat attendu	Résultat obtenu
1	Vérification synchronisation GitHub – GitHub desktop	Faire un commit GitHub en utilisant GitHub Desktop afin de voir si le commit passe bien sur GitHub	OK	OK
2	Lancement d'un script depuis son chemin à l'intérieur d'un autre script	Entrer la commande ./x-pappromail.ps1 pour voir si le script s'exécute	KO	OK
3	Affichage de l'aide (script 2)	Dans l'en-tête des scripts, renseigner .SYNOPSIS, .DESCRIPTION, .EXAMPLE et .PARAMETER, puis faire les commandes Get-Help x-yanmangiagl-setup.ps1 et Get-Help yanmangiagl-setup.ps1 -Examples	L'aide s'affiche sans problème	L'aide s'affiche sans problème
4	Déplacement des variables (script 2)	Faire bouger les variables \$startPath et \$destinationPath dans le scriptblock d'invoke command pour ne plus avoir de x dite \$null alors qu'elle ne l'est pas	Les variables sont instanciées et permettent au script de passer à la suite	Les 2 variables ont l'air de fonctionner maintenant mais \$session est \$null
5	Supprimer d'invokecommand copy-item	Bouger Copy-Item du bloc d'invoke-command pour l'isoler	OK	OK
6	Tenter 192.168.10.53 sans l'autre comme paramètre du script 2 (.51 fonctionne)	Restart-Service winrm, suivi d'un redémarrage du serveur, puis effectuer la commande	Marche pour toujours	Fonctionne seulement une fois après un redémarrage du serveur si la commande est faite assez vite
8	Désactiver pare-feu serveur 2	La premier serveur a un pare-feu éteint et pas le deuxième, tester de le désactiver pour voir si ça fonctionne	Marche pour toujours	Marche pour toujours

9	Envoi de mail avec Send-MailMessage	Lancer la commande permettant d'envoyer un mail	La commande peut s'exécuter sans problème	La commande ne s'exécute pas et retourne une erreur
10	Encoder avec Powershell	Rajouter [System.Text.Encoding ::UTF8] sur la variable \$body	Affichage de éà dans le mail	Affichage de ?? dans le mail
11	Test du script 1 fini	Mise en commentaire de la commande mettant le vrai pourcentage et lancement du script 1 depuis la machine locale	Le script fonctionne comme prévu et le mail est envoyé	Le script fonctionne comme prévu et le mail est envoyé
12	Exécution du script depuis la tâche planifiée	Même test mais sur yann2k22srv	Le script fonctionne comme prévu et le mail est envoyé	Le mail n'arrive pas > Problème vient sûrement de la création de la tâche dans script 2
13	Création d'une tâche manuelle	Création d'une tâche qui exécute le script manuellement pour comprendre d'où vient le problème en comparant les 2 tâches après	Dans les arguments on voit c:\scripts...	Dans les arguments on voit -File c:\scripts...
14	Calcul du stockage correct	Appel des valeurs \$usedSpace.Sum et \$totalspace pour calculer en % le stockage utilisé	Le pourcentage apparaît normalement	Le pourcentage apparaît normalement
15	Connecter serveur à internet	Réactiver une des cartes réseau	OK	OK
16	Lancer un mail depuis les 2 machines en même temps	Lancer le script 2 avec les 2 machines en paramètres avec le script de debug pour voir comment se lance les mails	Chaque machine fait son propre mail	Chaque machine fait son propre mail
17	Test taille réelle	OVA de 3 Go, OVA de 2 Go, deux serveurs prêts à accueillir les scripts	OVA de 3go suppr, ova de 2 go gardé, mail envoyé	-Path est null pour le truc que j'ai réglé dans les 5 min, tout le reste a fonctionné

7 CONCLUSION

7.1 Bilan des fonctionnalités demandées

Pour ce qui est des fonctionnalités demandées dans le cahier des charges, un point sur toutes les fonctionnalités va être effectué.

Exécution sans erreur sur une machine quelconque avec indication de la bonne exécution des tâches : Fonctionnel à 100%

Envoi d'un courriel avec les informations de la machine : Fonctionnel à 100%, le mail s'envoie parfaitement sans délai et ne se retrouve pas dans les spam de la boîte de réception.

Récupération des données pertinentes : Fonctionnel à 100%, l'interface pourrait être plus conviviale, mais toutes les informations demandées sont consultables dans le mail. Une amélioration souhaitable serait de convertir en Gb les octets utilisés pour rendre le mail plus lisible.

Suppression des dossiers trop volumineux : Fonctionnel à 100%, les fichiers OVA de plus de 3GB sont supprimés.

Exécution de la tâche planifiée avec succès : Fonctionnel à 100%, la tâche planifiée est créée et s'exécute tous les jours à 16h00 sans accroc.

Copie sur une machine distante : Fonctionnel à 100%, le script 2 copie le script 1 correctement sans erreur.

Procédure de déploiement et exécution détaillée dans le rapport : 90 % Je considère avoir dit ce qu'il y'avait à dire dans le rapport, mais j'ai l'impression d'oublier quelque chose à ce sujet.

7.2 Bilan de la planification

La planification n'a pas vraiment été suivie. J'ai été trop optimiste sur le temps que prendrait les tâches à accomplir et suis parti du principe que je n'aurai pas d'erreur en créant les scripts, ce qui était peu probable. En conséquence, beaucoup de tâches ont été réalisées en plus de temps que prévu, et des tâches ont même dues être rajoutées. Les tâches qui ont causé du retard sur ma planification sont toutes les tâches importantes, comme la création de scripts ou encore la réalisation du rapport. La tâche où j'ai pris de l'avance était la rencontre avec l'expert 2, où j'avais pensé que je devrais lui montrer tout le projet dans le détail et que ça prendrait beaucoup plus de temps que la réalité. Avec une semaine de plus, je pense que j'aurais pu réussir à magnifier le script en rajoutant les chemins en paramètre du script 2 ainsi que des paramètres pour l'heure voulue pour la tâche planifiée. Pour ce qui est du temps réalisé, j'ai travaillé pendant les séances de classe des mercredis ainsi que chez moi 20 minutes le samedi 1^{er} juin, ce qui expliquerai pourquoi j'arrive à plus de 90h sur les 88 demandées. Le journal de travail se trouve dans les divers du rapport.

7.3 Bilan personnel

Si le projet devait être refait, j'accorderai plus de temps au code, et plus d'importance aux chemins. J'ai peut-être passé trop de temps sur le rapport. Il y'a aussi beaucoup de captures d'écran que j'ai faites qui ne sont pas utilisées alors qu'elles pourraient et devraient être utiles. De plus, la dernière semaine j'ai eu beaucoup de problèmes à rester concentré sur mon travail pour finir le rapport. En revanche, ce que je voudrais garder telle quelle c'est ma capacité à résoudre les problèmes de script plutôt rapidement, et avoir su poser des questions à mon chef de projet et aux experts quand j'en avais besoin. Ce projet m'a appris à mieux gérer mon temps de travail, de nouvelles méthodes pour envoyer un mail via PowerShell, debugger efficacement sur PowerShell et oser poser des questions, ce qui n'était pas forcément toujours le cas pendant les projets précédents. Pour améliorer le projet, il y'a plusieurs voies : d'abord il faudrait améliorer les chemins des scripts, bien que ceux-ci marchent sur les machines testées (toutes les machines testées avaient leur partition principale sur la lettre C) Une autre chose à améliorer serait d'avoir une meilleure attitude de travail. En effet, à plusieurs reprises il m'est arrivé de lâcher prise et de faire d'autres choses pendant mon travail, ou alors encore de regarder dans le vide sans avancer. Cela arrive souvent après que je sois bloqué sur un élément, comme un script qui ne fonctionne pas comme attendu. Là où je le ressens le plus est pendant la rédaction du rapport, où

je pouvais mettre 20 minutes pour écrire une phrase simple. Les difficultés principales que j'ai rencontrées, c'était que pour certains problèmes, il était compliqué de savoir si le problème venait du script ou de l'environnement dans lequel il évoluait.

7.3.1 Remerciements

Mr Gaël Sonney, mon chef de projet qui m'a aidé à de nombreuses reprises.

Mr Daniel Berney, mon second expert venu me rendre visite mardi 21 mai pour s'assurer que le projet se déroulait correctement.

Mr Gabriel Maret, mon premier expert qui s'est assuré que le cahier des charges soit bien compris et qui a répondu à mes questions.

Mr Leonar Dupuis, un camarade de classe qui m'a aidé à arranger mon journal de travail et qui a relu mon travail.

Mr Cédric Schaffter, qui m'a aidé à me préparer au TPI pendant les projets d'approfondissement.

8 DIVERS

8.1 Journal de travail

Le journal de travail a été réalisé en parallèle du projet pendant tout son déroulement. Dès que je sentais un petit blocage dans mon travail, je profitais pour regarder l'heure et noter ce qui avait été fait depuis le dernier petit blocage, ce qui rend les tranches très fidèles à la réalité.

Séquence 27			Date: lundi, 3 juin 2024		13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	1	0h5min	Génération d'une meilleure image de projet	Bing Copilot	
Doc - Rapport	1	0h5min	Dernière actualisation du journal de travail dans le rapport		
Doc - Rapport	10	0h50min	Peaufinement et ajout de la procédure de déploiement		
Impression du TPI	15	1h15min			
Pause de 15 minutes	3	0h15min			
Réalisation - Github	1	0h5min			
				Ressenti de la journée : Plutôt bonne, j'ai réussi à avancer et finir ce que je désirais faire	
Total tranche	31	2h35min			
Séquence 26			Date: lundi, 3 juin 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	3	0h15min	Résumé	Mr Sonney	
Analyse - Entretien CDP	4	0h20min	RDV Chef de projet pour répondre aux questions du mail par message + visite en présentiel		
Doc - Rapport	7	0h35min	Planification détaillée		
Doc - Rapport	2	0h10min	Recherche écriture pare-feu au pluriel (Word considère faux) et correction	Académie française	
Doc - Journal de travail	1	0h5min			
Pause de 15 minutes	3	0h15min			
Doc - Tests	29	2h25min	Relecture du TPI d'un camarade qui relit le mien + relecture de mon tpi -> Ajout de l'explication de l'entête dans la partie réalisation	Leonar Dupuis	
Doc - Journal de travail	2	0h10min	Correction des erreurs dans le compte des heures si trouvée (censé arriver à 88h) -> trouvé. Mercredi nous commençons dès que la séance de classe était terminée		
Total tranche	51	4h15min			
Séquence 25			Date: vendredi, 31 mai 2024		13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	2	0h10min	Amélioration de l'affichage du tableau des tests		
Doc - Rapport	1	0h5min	Correction et révision de certains tests		
Doc - Rapport	15	1h15min	Résumé		
Doc - Rapport	21	1h45min	Réalisation du glossaire, recherche de tous les mots du rapport qui pourraient avoir besoin d'explications	Youtube	
Doc - Journal de travail	1	0h5min			
Réalisation - Github	1	0h5min	Envoi github et mail		
Doc - Rapport	3	0h15min	Samedi : Continuation du glossaire pour être sûr de finir dans les temps + modification des mots déjà existants pour leur ajouter des majuscules		
Absences - imprévus	1	0h5min	Samedi : Ajout du ressenti de vendredi		
Total tranche	45	3h45min			

Séquence 24			Date: vendredi, 31 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	12	1h	Rédaction de la conclusion, partie bilan personnel, création du chapitre sur l'exécution détaillée du script		
Doc - Journal de travail	6	0h30min	Inscription dans le journal de travail ce qui a été réussi ou non lors des séquences précédentes		
Doc - Tests	7	0h35min	Ajout de tests dans le tableau de tests du rapport en fonction des captures d'écran prises		
Doc - Tests	3	0h15min	Ajout dans réalisation en fonction de captures d'écran prises		
Pause de 15 minutes	3	0h15min			
Doc - Rapport	20	1h40min	Planification détaillée + résumé -> échec des 2		
				Ressenti: J'ai réussi à avancer un moment après avoir trop réfléchi pour le résumé	
Total tranche	51	4h15min			
Séquence 23			Date: jeudi, 30 mai 2024		13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	6	0h30min	Suite de la rédaction de la webographie	Scribbr	
Analyse - Entretien CDP	2	0h10min	Discussion avec chef de projet par rapport au message du matin		
Doc - Rapport	13	1h5min	Suite de la rédaction du résumé		
Pause de 15 minutes	3	0h15min			
Doc - Rapport	11	0h55min	Rédaction du journal de travail dans le rapport		
Doc - Rapport	5	0h25min	Continuation de la conclusion		
Doc - Journal de travail	1	0h5min			
				Ressenti : je suis bloqué au résumé, je ne sais pas quoi dire de pertinent, j'avance dans le reste	
Total tranche	41	3h25min			
Séquence 22			Date: jeudi, 30 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	2	0h10min	Relecture en bref du rapport pour voir ce qui doit encore être ajouté.		
Doc - Rapport	18	1h30min	Rédaction dans le rapport de la partie Conclusion		
Pause de 15 minutes	3	0h15min			
Doc - Rapport	10	0h50min	Fin de rédaction de la planification initiale		
Doc - Rapport	12	1h	Rédaction du résumé		
	3	0h15min	Relecture de la grille d'évaluation		
Doc - Rapport	2	0h10min	Webographie	Scribbr	
Analyse - Entretien CDP	1	0h5min	Envoi message au chef de projet à propos de la planification initiale		
Total tranche	51	4h15min			
Séquence 21			Date: mercredi, 29 mai 2024		08:00 - 11:25
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Séance de classe	3	0h15min			
Doc - Rapport	2	0h10min	Finalisation du premier jet de la partie réalisation finale		
Doc - Rapport	33	2h45min	Remplir ce qui a été mis de côté dans le rapport (objectifs smart, CQCCOQP)	MMDFrance	
Pause de 15 minutes	3	0h15min		PrevenirPourDevenir	
				ressenti : idem	
Total tranche	41	3h25min			

Séquence 20			Date: mardi, 28 mai 2024	13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	28	2h20min	Continuation de la partie réalisation + maj des images de la partie précédente vu que le code a été mis à jour	
Pause de 15 minutes	3	0h15min		
				Ressenti: Je sens que comme je suis sur la fin du projet je me laisse aller et met longtemps à faire des tâches
Total tranche	31	2h35min		
Séquence 19			Date: mardi, 28 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	47	3h55min	Continuation de la partie réalisation + ajout des légendes aux images n'en ayant pas	Microsoft Learn
Pause de 15 minutes	3	0h15min		
Doc - Journal de travail	1	0h5min		
Total tranche	51	4h15min		
Séquence 18			Date: lundi, 27 mai 2024	13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	24	2h	Partie réalisation du script 2	
Analyse - Entretien CDP	3	0h15min	Démonstration au chef de projet sur le fonctionnement des scripts	Mr Sonney
Pause de 15 minutes	3	0h15min		
Script 2 - Copie	1	0h5min	Debug pour si fichier Scripts existe pas	
				Ressenti: Avancée OK mais aurait pu mieux faire
Total tranche	31	2h35min		
Séquence 17			Date: lundi, 27 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Journal de travail	1	0h5min	Ouverture des documents + journal de travail	
Script 2 - Copie	1	0h5min	Ajout de nommage d'action	
Doc - Tests	15	1h15min	Tests des 2 scripts en vue d'une démonstration au chef de projet (mail en même temps [9:00:00] -> mails pas envoyé -> srv2 avait pas dossier Scripts et créait un fichier script plutôt qu'un dossier	
Pause de 15 minutes	3	0h15min		
Script 2 - Copie	2	0h10min	Correction problème de dossier	
Doc - Rapport	29	2h25min	Commencement de la partie réalisation	RDR.IT
Total tranche	51	4h15min		

Séquence 16			Date: vendredi, 24 mai 2024		13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Script 2 - Copie	11	0h55min	Ajout de gestion des erreurs dans le script 2 (nombre d'adresses > 10, nombre d'adresses = 0, adresse ip incorrecte)		
Script 2 - Tâche planifiée	10	0h50min	Supprime la tâche si elle existe déjà -> ne marche pas totalement, comme pas demandé dans cdc, mis de côté		
Pause de 15 minutes	3	0h15min			
Script 2 - Copie	14	1h10min	Changement des chemins codés en dur en chemin dynamique pour les rendre utilisable avec n'importe quelle machine	StackExchange	
Réalisation - Github	1	0h5min	Rendu Github + mail		
Analyse - Entretien CDP	1	0h5min	Chef de projet venu pour m'apporter des précisions sur ce qui a été discuté la veille	Mr Sonney	
Doc - Journal de travail	1	0h5min			
				Bof - Pertes de concentration nombreuses, j'aurais pu faire beaucoup plus aujourd'hui	
Total tranche	41	3h25min			
Séquence 15			Date: vendredi, 24 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Script 2 - Tâche planifiée	10	0h50min	Lancement de Powershell ISE + réfléchir à pourquoi le c:\ n'apparaît pas dans les arguments qu'il faut dans la tâche -> réussi à le faire apparaître et donc faire marcher le script mais chemin en dur		
Doc - Rapport	1	0h5min	Mise à jour de la partie du script 2 concernant le \$fullpath		
Doc - Tests	2	0h10min	Copie d'un fichier vdi depuis le SSD jusqu'au serveur pour le supprimer via script (réussi)	Superuser (StackExchange)	
Script 1 - Effacement	6	0h30min	Avancée dans le script 1	Bing Copilot pour la façon d'appréhender la	
Pause de 15 minutes	3	0h15min			
Doc - Rapport	27	2h15min	Rédaction du chapitre sur la méthode des 6 pas, à quoi cela sert et comment l'appliquer à mon projet		
Doc - Tests	2	0h10min	Création ova de moins de 3 gb pour le mettre sur rm et voir qu'il ne disparaît pas		
Total tranche	51	4h15min			
Séquence 14			Date: jeudi, 23 mai 2024		13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	33	2h45min	Avancée dans le rapport dans la partie réalisation du script 1 et modification de l'image des cartes réseaux activées		
Pause de 15 minutes	3	0h15min			
Script 2 - Tâche planifiée	12	1h	Tentatives de faire fonctionner la tâche planifiée depuis une machine distante, activation d'une carte réseau -> Problème trouvé, le C:\ disparaît	Neptunet	
Analyse - Entretien CDP	3	0h15min	Démonstration du script 2 au chef de projet + discussion travail à faire sur la méthode des 6 pas dans analyse du rapport	Mr Sonney	
				Ressenti de la journée : OK	
Total tranche	51	4h15min			
Séquence 13			Date: jeudi, 23 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	36	3h	Avancée de la partie Réalisation pour le script 1.		
Script 1 - Espace disque	12	1h	Continuation du script pour récupérer l'espace disque - inspiration d'un ancien script réalisé	pastebin pour trouver le script réalisé	
Doc - Journal de travail	1	0h5min			
Absences - imprévus	1	0h5min	Création compte pastebin pour mettre une référence de l'ancien script		
Script 1 - Espace disque	1	0h5min	Mise en route de la réation de la partie suppression Ova du script		
Total tranche	51	4h15min			

Séquence 12			Date: mercredi, 22 mai 2024	08:00 - 11:25
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Séance de classe	1	0h5min	Séance de classe	
Séance de classe	4	0h20min	Rédaction du demande de congé pour le mercredi 29 mai dès 13:10 afin de m'inscrire à la formation SPARC à Zürich	site web de sparc
Script 1 - Espace disque	20	1h40min	Calcul de l'espace disque	Microsoft Learn
Analyse - Entretien CDP	1	0h5min	Envoi message Chef de projet à propos du CDC sur l'espace disque (tous les disques ou disque principal)	Mr Sonney
Script 1 - Infos supplémentaires	6	0h30min	Versions des mises à jour, du nom de l'ordinateur et de la version de l'OS + test envoi mail réussi mais encodage mauvais é -> ?	Tech2Tech
Pause de 15 minutes	3	0h15min		Ressenti OK, Avancée mais pas dans le rapport et calcul espace disque a pris beaucoup de temps pour
Doc - Journal de travail	2	0h10min	Rédaction du journal de travail	
Script 1 - Infos supplémentaires	4	0h20min	Versions de mise à jour -> ne passent pas le mail	
Total tranche	41	3h25min		

Séquence 11			Date: mardi, 21 mai 2024	13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Script 1 - Courriel	15	1h15min	Essai d'envoyer un mail et correction des problèmes d'installation de Nuget -> abandon retour sur sendmail-message	
Analyse - Rencontre expert	6	0h30min	Entretien avec expert 2 - Mise au point sur le travail rendu vendredi	
Script 1 - Courriel	9	0h45min	Envoi mail de test réussi (avant ce qui marchait pas était mot de passe pas passé dans sécurité suppl) + avancée script	Sharepointdiary
Réalisation - Github	1	0h5min	Pull github + mail	
				Ressenti de la journée: OK, bon début de journée mais compliqué par la suite résolution du problème en fin
Total tranche	31	2h35min		

Séquence 10			Date: mardi, 21 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Journal de travail	1	0h5min	Lancement de tous les logiciels qui seront utilisés + journal de travail	
Doc - Rapport	17	1h25min	Rapport : réalisation	
Pause de 15 minutes	3	0h15min		
Script 1 - Courriel	3	0h15min	Début de la rédaction du script 1 (va sûrement être refaire car mimekit et mailkit pas sûr que ce soit la bonne idée)	
Script 1 - Courriel	12	1h	Tentative de faire marcher send-mailmessage	
Script 1 - Courriel	15	1h15min	Installation de Nuget, MimeKit et MailKit	Microsoft Learn
Total tranche	51	4h15min		

Séquence 9			Date: vendredi, 17 mai 2024	13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Rapport	30	2h30min	Rapport analyse et conception	
Doc - Journal de travail	1	0h5min	journal de travail + création issue github	
Pause de 15 minutes	3	0h15min		
Script 2 - Copie	2	0h10min	Changement de commentaires dans le script 2 + ajustements mineurs	
Script 2 - Copie	3	0h15min	Vérification de l'utilité de Out-Null dans la boucle foreach	
Réalisation - Github	2	0h10min	Enregistrement des fichiers et envoi aux experts par mail et par github	
				Ressenti de la journée: Mauvais, beaucoup de pertes de concentration, peine à rester focus sur mon travail - perte de temps
Total tranche	41	3h25min		

Séquence 8			Date: vendredi, 17 mai 2024	08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Doc - Journal de travail	1	0h5min		
Doc - Rapport	7	0h35min	Rédaction des points Matériel à exploiter, recherche d'informations particulières et solutions possibles. Mise à jour de la table des matières et de certains champs du bas de page	
Doc - Rapport	40	3h20min	Dossier de réalisation - Version des logiciels utilisés et Configuration spéciale des outils	
Pause de 15 minutes	3	0h15min		
Total tranche	51	4h15min		

Séquence 7			Date: jeudi, 16 mai 2024		13:10 - 16:35
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Journal de travail	1	0h5min			
Analyse - Entretien CDP	3	0h15min	Mise au clair avec chef de projet à propos de certaines + déblocage paramètre avec multiples valeurs	Mr Sonney	
Script 2 - Tâche planifiée	27	2h15min	Changements dans le script pour intégrer le tableau, problème bloquant, le serveur 2 a un problème DNS qui empêche une connexion malgré que le script fonctionne -> Problème venait du pare-feu		
Réalisation - Github	10	0h50min	commit sur github du script 2 - Bug d'encodage		
				Ressenti de la journée : Bon, j'ai senti avancer	
Total tranche	41	3h25min			

Séquence 6			Date: jeudi, 16 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Script 2 - Copie	7	0h35min	Continuation de l'avancée dans le script 2 - Faire fonctionner les chemins qui retournent null alors qu'ils sont instanciés	Découverte de -ToSession et -FromSession	
Pause de 15 minutes	3	0h15min	9:35 - 9:50		
Doc - Journal de travail	1	0h5min	Journal de travail		
Analyse - Entretien CDP	2	0h10min	Rédaction de toutes les questions à poser au chef de projet dans un fichier		
Script 2 - Copie	3	0h15min	Documentation et réalisation pour regarder comment mettre une liste (d'adresses ip) comme paramètre au script	Reddit	
Script 2 - Tâche planifiée	3	0h15min	Création d'une tâche planifiée	Tuto pour créer une tâche planifiée	
Doc - Tests	32	2h40min	Mise en œuvre de ce qui a été trouvé dans la documentation du point 5 dans un script de test		
Total tranche	51	4h15min			

Séquence 5			Date: mercredi, 15 mai 2024		08:00 - 11:25
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Séance de classe	9	0h45min	Séance de classe		
Script 2 - Copie	1	0h5min	Ecriture des entêtes du script 2 (x-yanmangia-gl-setup)		
Script 2 - Copie	24	2h	Script 2: Début de scripting : Connexion à l'ordinateur et collage d'un fichier quelconque pour le moment		
Script 2 - Copie	3	0h15min	Documentation pour la commande copy-item		
	3	0h15min	Pause		
	1	0h5min	Mise à jour du journal de travail		
				Ressenti de la journée: OK. Avancée mais pas incroyable	
Total tranche	41	3h25min			

Séquence 4			Date: mardi, 14 mai 2024		13:10 - 15:45
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Journal de travail	4	0h20min	Tentative de création du bilan, crash excel et messages d'erreur, camarade m'a aidé à réparer le problème	Leonar Dupuis	
Pause de 15 minutes	3	0h15min	14:45 - 15:00		
Doc - Rapport	1	0h5min	Rédaction des points opportunités qui n'est pas encore terminé	Mr Sonney m'a aidé pour le paragraphe de get-	
Doc - Tests	1	0h5min	Test depuis machine serveur yann2ksrv pour tenter de lancer un script depuis un autre après avoir relancé le serveur		
Réalisation - Github	1	0h5min	Création des fichiers de script et mise en ligne sur github		
Analyse - Schéma	19	1h35min	Début de la construction du schéma		
Réalisation - Github	2	0h10min	Pull + commit de mardi contenant la v2 du journal de travail et de la planification et la v1 du rapport	Ressenti de la journée : Bon, j'ai bien avancé	
Total tranche	31	2h35min			

Séquence 3			Date: mardi, 14 mai 2024		08:00 - 12:15
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Rencontre expert	2	0h10min	Ajout du rendez-vous de l'expert 2 à la planification initiale suite à réception de son mail		
Doc - Journal de travail	2	0h10min	Ajout de ce qui a été fait la veille dans le journal de travail		
Réalisation - Github	1	0h5min	Installation de Github Desktop + premier commit pour tester si le repo fonctionne correctement + création fichier captures où mettre tous les screens qui serviront au projet (pas sur github)	git/doc	
Doc - Rapport	42	3h30min	Début de la rédaction du rapport titre, description, matériel, prérequis, cahier des charges, caractéristiques des utilisateurs, fonctionnalités requises, contraintes, travail à réaliser, si le temps le permet, validation des solutions, mises à jour du planning, début du résumé, opportunités	Rapport sur Github (/fr/télécharge le document)	
Pause de 15 minutes	4	0h20min	9:35 - 9:50 + 11:25 - 11:30		
Total tranche	51	4h15min			

Poséidon. (2022, January 27). [Tips] Exécuter un script Powershell depuis le planificateur de tâches de Windows – NEPTUNET.FR. <https://neptunet.fr/task-powershell/>

Powershell to delete all files with a certain file extension. (n.d.). Super User. <https://superuser.com/questions/1091344/powershell-to-delete-all-files-with-a-certain-file-extension>

Concatenate filename with current directory path. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/57610821/concatenate-filename-with-current-directory-path>

Romain. (2023, September 12). Créer un dossier en PowerShell - RDR-IT. RDR-IT. <https://rdr-it.io/creer-un-dossier-en-powershell/>

JonDouglas. (n.d.). Documentation de NuGet. Microsoft Learn. <https://learn.microsoft.com/fr-fr/nuget/>

Prévenir pour Devenir. (n.d.). Fichier de planification d'objectifs SMART à télécharger. Prévenir Pour Devenir. <https://prevenirpourdevenir.ca/fichier-de-planification-dobjectifs-smart-a-telecharger/>

Masucci, F. (2023, February 13). La méthode CQCOQP en Mind Mapping. MMD France. <https://mmdfrance.fr/methode-cqqcoqp/>

SJ Abgrall. (2017, April 3). Créer un Index ou un Glossaire sous Word [Video]. YouTube. <https://www.youtube.com/watch?v=srlNOQmQ4lg>

8.3 Table des illustrations

Figure 1 - Image générée par Bing Copilot représentant le projet	1
Figure 2 - Schéma démontrant comment les scripts ont été pensés	17
Figure 3 - Création du tableau	17
Figure 4 - Variable d'environnement du nom de l'ordinateur	18
Figure 5 - Variables d'environnement de la version de l'OS	18
Figure 6 - Enregistrement de l'espace libre	18
Figure 7 - Enregistrement de l'espace utilisé	18
Figure 8 - Enregistrement de l'historique des versions de mise à jour.....	18
Figure 9 - Calcul du stockage utilisé	18
Figure 10 - Contenu de \$usedSpace	19
Figure 11 - Calcul de l'espace libre entre tous les disques.....	19
Figure 12 - Calcul du stockage total	19
Figure 13 - Calcul en % de l'espace utilisé par rapport au stockage total	19
Figure 14 - Condition pour faire fonctionner la suite du script	19
Figure 15 - Informations du serveur SMTP	19
Figure 16 - Informations de l'expéditeur	19
Figure 17 - Sujet et corps de mail	20
Figure 18 - Mot de passe d'application Google utilisé	20
Figure 19 - Sécurisation du mot de passe	20
Figure 20 - Création d'un objet de crédentité	20
Figure 21 - Envoi du mail via la commande Send-MailMessage	20
Figure 22 - Instanciation du tableau	21
Figure 23 - Instanciation d'un deuxième tableau	21
Figure 24 - placeholder - boucle foreach	21
Figure 25 - Conversion du mot de passe	21
Figure 26 - Récupération du nom d'utilisateur	21

Figure 27 - Implantation des chemins.....	22
Figure 28 - Début de la boucle.....	22
Figure 29 - Informations d'une session ouverte	22
Figure 30 - Argument servant à lancer la tâche planifiée	22
Figure 31 - Création de la tâche planifiée	23
Figure 32 - Création de la répétitivité de la tâche	23
Figure 33 - Description de la tâche et enregistrement de cette tâche.....	23
Figure 34 - Spécifications de Windows Server	28
Figure 35 - Configuration du pare-feu	29
Figure 36 - Configuration des cartes réseau	29
Figure 37 - Protocole IPV6 désactivé	30
Figure 38 - Activation du bureau à distance	30
Figure 39 - Configuration de WinRM	30
Figure 40 - Page d'accueil de la page de création de mots de passe d'application Google	31
Figure 41 - Mot de passe généré	31
Figure 42 - En-tête du script 2.....	33
Figure 43 - Le message d'erreur qui apparaissait avant de modifier copy-item	34
Figure 44 - Première partie du code source original	34
Figure 45 - Seconde partie du code source original	34
Figure 46 - Modification des exemples.....	35
Figure 47 - Modification de l'argument du script 2	35
Figure 48 - Résultat voulu de \$fullpath.....	35
Figure 49 - Ajout de gestion d'erreurs pour la conversion DNS.....	35
Figure 50 - Ajout d'une limite de 10 machines.....	36
Figure 51 - Modification d'un chemin en dur en chemin automatique	36
Figure 52 - La concaténation ne se fait pas	36
Figure 53 - La concaténation se fait	36
Figure 54 - Ajout d'une gestion d'erreurs pendant la connexion	36
Figure 55 - Vérification de si la tâche existe déjà, et la supprime si c'est le cas	37
Figure 56 - Bloc de code si une adresse IP est incorrecte	37
Figure 57 - Sortie du script une fois les tâches créées.....	37
Figure 58 - Installation de MimeKit et MailKit.....	38
Figure 59 - Première partie du script 1	39
Figure 60 - Seconde partie du script 1	39
Figure 61 - Capture d'écran du message d'erreur dû à Google	40
Figure 62 - Enlèvement de la lettre du disque	40
Figure 63 - Ajout des valeurs dans le script	40
Figure 64 - Création de la condition de 80%	40
Figure 65 - Création du corps du mail	40
Figure 66 - Mails de tests reçus.....	41
Figure 67 - Contournement des problèmes d'encodage	41
Figure 68 - Ajout des versions de MAJ	41
Figure 69 - Corps du mail final	41
Figure 70 - Commande permettant de supprimer les fichiers ova de plus de 3Go	41

9 GLOSSAIRE

A

Administrateur système	
Personne responsable des serveurs d'une entreprise	6

C

CFC	
Certificat fédéral de capacité	6

D

Débogage	
Action de corriger des bugs informatique	18
Docker kubernetes	
Outil de managment de docker	16

E

Environnement	
Ensemble matériel et logiciels système	16
Ethernet	
Protocole de communication	4

H

Hachage	
Algorithme permettant de convertir en suite de nombres une chaîne de caractère	21

M

Machine virtuelle	
Machine simulée sur de vraies machines	5

O

OVA	
Extension de fichiers souvent volumineux utilisés pour les machines virtuelles	4

P

PowerShell	
Solution multiplateforme d'automatisation de tâches	5

S

Script	
Ensemble d'instructions dans un langage de programmation	4
Serveur SMTP	
Ordinateur qui gère les courriels sortants	7
Session	
Connexion d'un pc à un autre via PowerShell	22
SSL	
Protocole de sécurité Internet	20
Système d'exploitation	
Ensemble de programmes spécialisés qui travaillent ensemble (WIndows, Linux)	18

T

Tableau associatif

Tableau en programmation liant une clé à une valeur

17

10 ANNEXES

Code source du script 1

Code source du script 2

Journal de travail