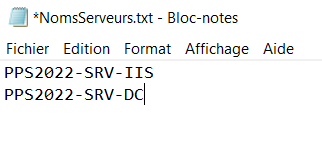
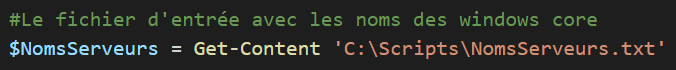
# Supervision

On commencer avec les noms de serveurs qu’on souhaite superviser. Ils sont notés à la ligne dans un document texte appelé NomsServeur.txt



Ensuite grâce à la commande Get-Content je vais prendre le contenu de ce fichier texte et je le mets dans la variable $NomsServeurs

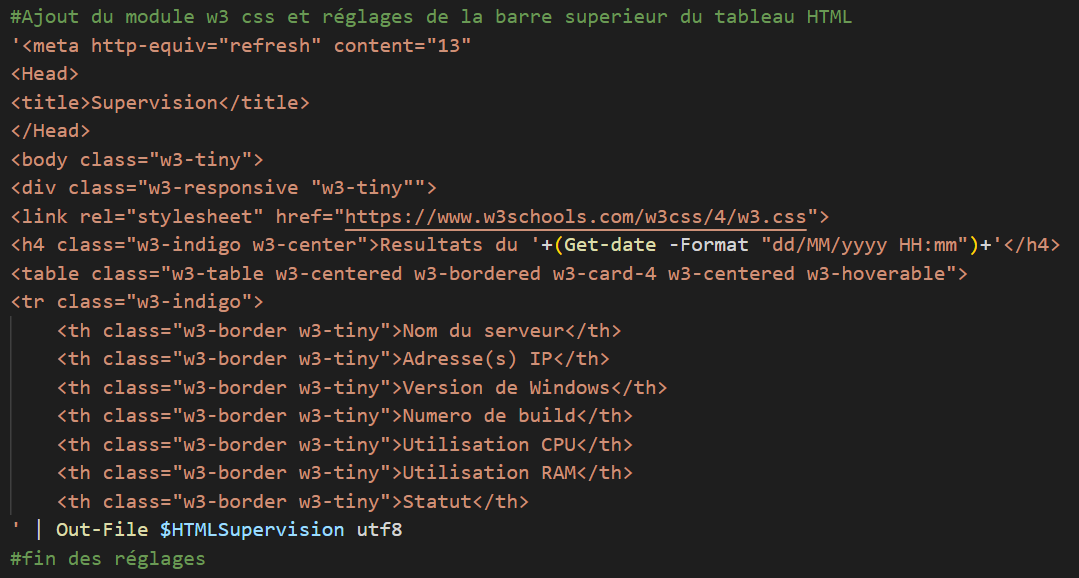


Je précise la variable $HTMLSupervision avec le fichier HTML de sortie qui sera notre page IIS

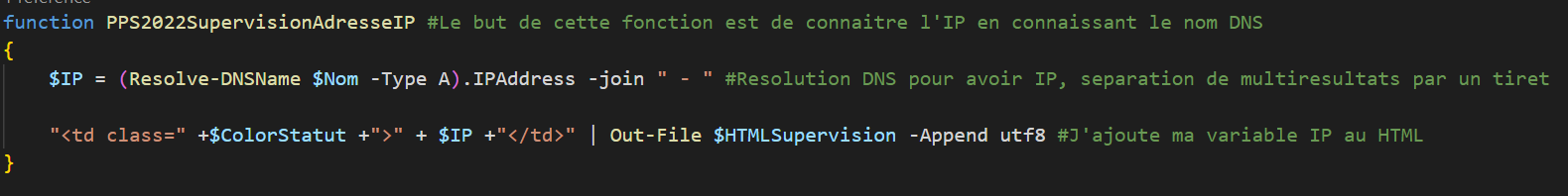


Sur cette partie-là je vais faire des réglages du HTML, comme la partie auto refresh de 13 secondes.

Je vais importer le module w3-css qui me permettra de mettre de la couleur et de créer mon entête de HTML, avec le titre, la date et l’heure, les colonnes etc. Ensuite j’exporte ça dans le HTML.

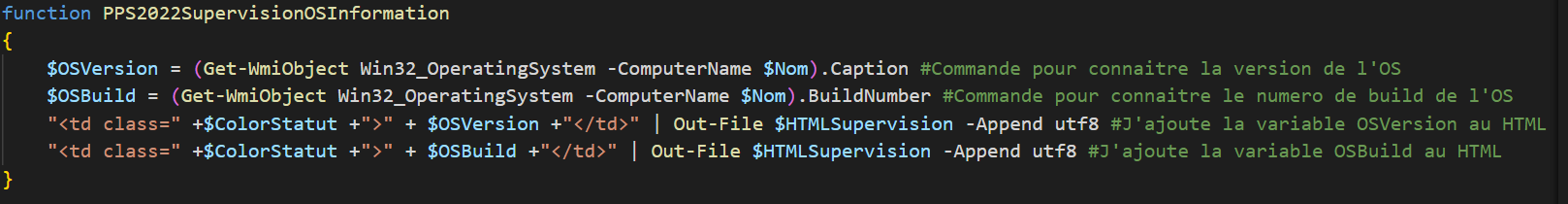


La fonction Supervision adresse IP me permet, via une resolution DNS (commande Resolve-DNSName) de connaitre l’IP du serveur en ne connaissant que son nom. J’ajoute cette valeur a la variable $IP. Enfin j’ajoute cette valeur ainsi que celle de couleur de statut (explication plus loin) dans mon fichier HTML de sortie.



La fonction Supervision OS information me permet de connaitre le système installé sur le serveur via la commande **Get-WmiObject Win32\_OperatingSystem).Caption** dont j’exporte la valeur dans une variable $OSVersion et de connaitre le build installé via la commande **Get-WmiObject Win32\_OperatingSystem).BuildNumber** donc j’exporte la valeur dans une variable $OSBuild

Ensuite j’exporte ces valeurs ainsi que la valeur de couleur de statut au fichier HTML de sortie.



La fonction Supervision CPU est la pour que je connaisse le pourcentage d’utilisation du processeur et aussi en déduire un code couleur suivant le résultat.

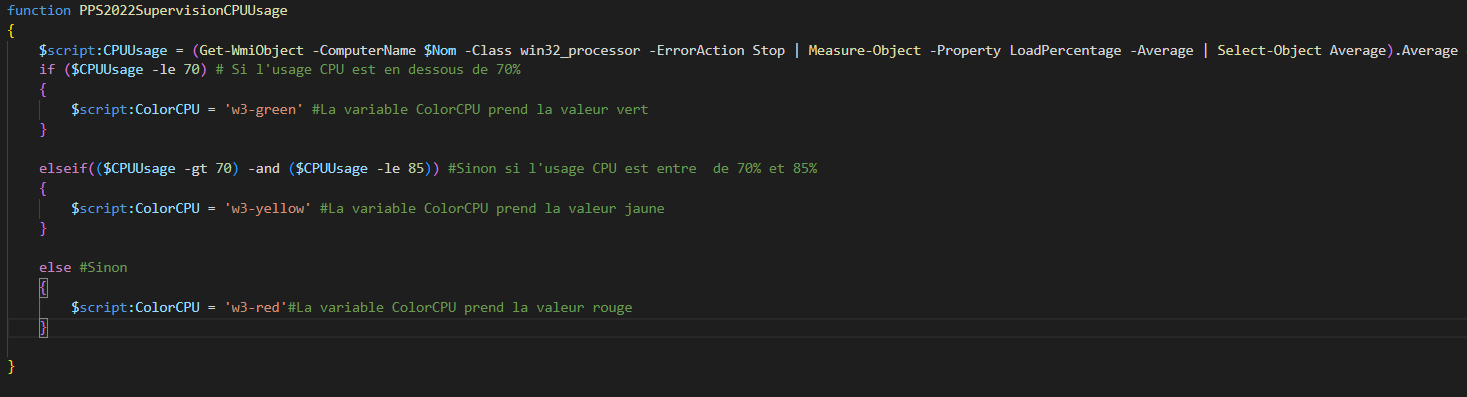
La variable $CPUUsage est scoper « script » pour que je puisse l’utiliser en dehors de la fonction (pour la fonction color statut qu’on verra plus tard)

Ensuite grâce à la commande :

(Get-WmiObject -ComputerName $Nom -Class win32\_processor -ErrorAction Stop | Measure-Object -Property LoadPercentage -Average | Select-Object Average).Average

Je récupère dans la variable $CPUUsage le pourcentage d’utilisation du processeur.

Ensuite avec une boucle IF je déduit la variable $ColorCPU en fonction de l’utilisation, vert si moins de 70% d’utilisation, jaune si entre 70% et 85% et rouge au-delà.



La fonction Supervision RAM Usage se base en partie sur le même concept que la CPU Usage.

Les deux principales différences c’est qu’en plus de la commande pour récupérer les informations techniques :

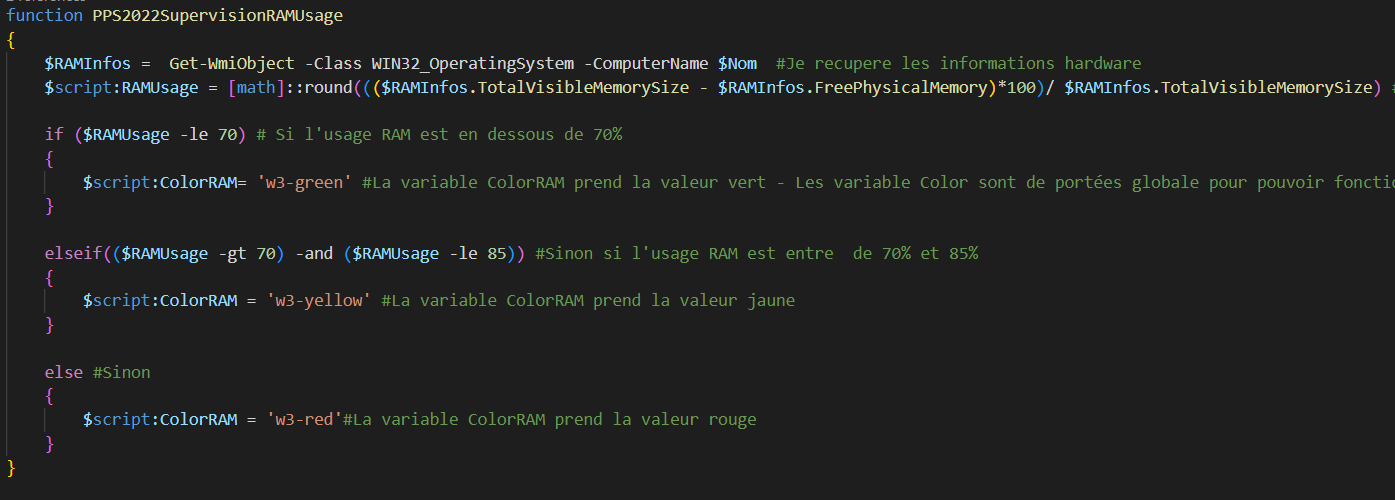
Get-WmiObject -Class WIN32\_OperatingSystem -ComputerName $Nom

Il y a une commande pour en faire une valeur exploitable sur 100, un pourcentage d’utilisation. J’utilise également [math] ::round pour avoir une valeur arrondie au decimal :

[math]::round((($RAMInfos.TotalVisibleMemorySize - $RAMInfos.FreePhysicalMemory)\*100)/ $RAMInfos.TotalVisibleMemorySize)

Je mets ensuite ces information dans une variable $RAMUsage

Enfin de la même manière que la CPU Usage, avec une boucle IF je déduit la variable $ColorRAM en fonction de l’utilisation, vert si moins de 70% d’utilisation, jaune si entre 70% et 85% et rouge au-delà.



La variable Supervision Statut elle me permet d’avoir une idée globale de l’état de la machine.

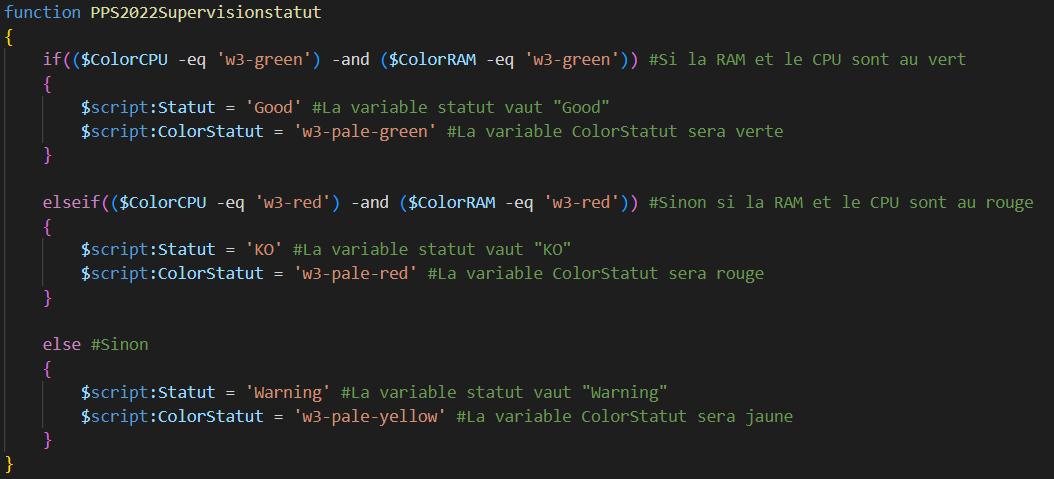
Si le CPU et la RAM sont en vert, alors c’est vert et la variable statut prend la valeur Good.

Si les deux sont en rouge, alors c’est rouge et la variable statut prend comme valeur KO.

Sinon la couleur est jaune et la variable statut prend la valeur Warning.

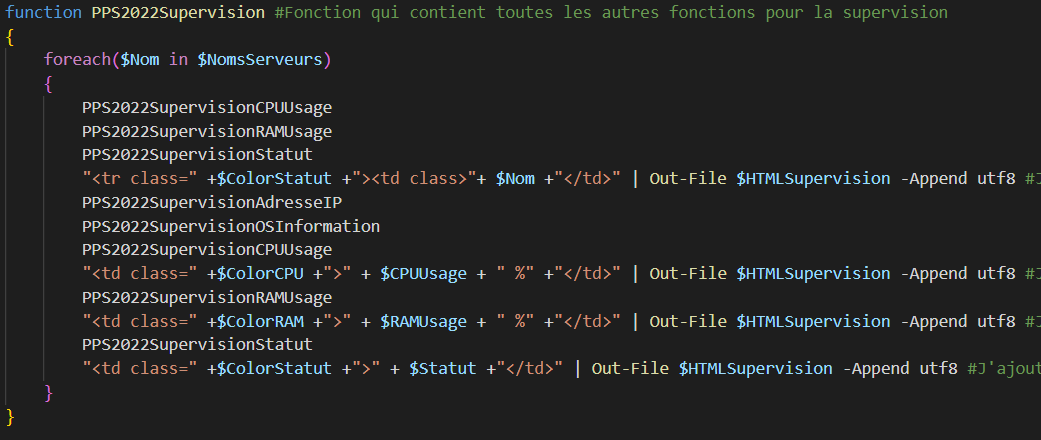
Ensuite j’exporte le résultat dans le html pour la colonne statut.

Aussi la couleur de la variable $ColorStatut est mise sur toute les cases de la ligne comme vu dans les fonctions précédentes pour avoir une vue rapide sur l’état général des serveurs.



Enfin la fonction Supervision qui fait office de fonction main. Avec une boucle foreach de chaque serveur dans le fichier texte elle va exécuter les fonctions une a une pour avoir les différentes informations et les ajouter dans le HTML de sortie.

On voit que j’exécute les fonctions CPU et RAM deux fois, pour une question d’ordre d’exécution. Une première qui me sert à avoir la couleur que je mets ensuite dans chaque case du tableau et une deuxième pour la valeur en pourcentage au moment venu. Je pense que cette partie est améliorable pour éviter la double exécution.



Améliorations envisageables :

Sur la partie supervision je pense que je pourrais améliorer :

* Eviter la double utilisation des fonctions CPU et RAM pour avoir la couleur, c’est dommage.
* Améliorer l’auto refresh pour éviter les pages en partie blanche de manière aléatoire
* On aurait pu faire en sorte qu’au déploiement d’un serveur il aille inscrire son nom automatiquement dans le fichier texte des noms de serveurs.