Rapport de stage seconde année

Rôle : administrateur systeme infrastructure





Stage pratique du 09/01/2018 au 23/02/2018

Table des matières

Remerciement:1

Introduction:1

Presentation de l’entreprise2

Presentation d’Odiwi2

Organisation et activités d’Odiwi2

Activités réalisées Durant le stage3

Planning du stage3

Analyse des besoins 4

Mise en service de postes et d’outils de travail4

Poste personnel4

Poste client5

Création de nouveau users dans l’AD/Création de boîtes mail5

Recherches personnelles 6

Recherche de solution et restauration de fichiers8

Soutien technique réseau 9

Suivi de mail et d’appel / Migration des boîtes mail Leolagrange11

Récupération de données / transfert sur cloud chez OVH 13

Supervision d’un serveur Veeam à l’aide de Nagios 15

Création d’un job sous Veeam15

Installation et configuration de l’agent NSClient 17

Plugin Veeam20

Configuration client côté serveur Nagios22

Réalisation de documentations24

Conclusion24

Annexes25

# Remerciement

Je tiens à remercier ma tutrice de stage, Me. Mougoue Lucrèce, ainsi que M.Ouary Bertrand de m’avoir accepté entant que stagiaire au sein de l’entreprise Odiwi.

Je remercie plus particulierement Me. Mougoue Lucrèce, pour les connaissances et l’experience que celle-ci m’a transmise durant la totalité du stage.

Je remercie l’ensemble des collaborateurs au sein de l’entreprise Odiwi pour leur accueil et leur présence tout au long du stage, ainsi que l’ensemble du groupe HERACLES.

Introduction

Ce stage entant qu’administrateur système infrastructure, d’une durée de sept semaines consistait à garantir le bon fonctionnement d’un service technique auprès des clients ainsi qu’à déployer un service de supervision d’un logiciel de backup nommé Veeam, depuis Nagios. Ce stage avait pour but de mettre en pratique les compétences acquises durant la formation interne réalisé dans l’école privée des sciences informatiques (EPSi) et d’en développer de nouvelles en rapport avec celles exercées par un administrateur infrastructure. Ce document contient des informations sur la totalité du travail que j’ai réalisé pendant mon stage chez Odiwi dans la période allant du 09 janvier 2018 au 23 février 2018.

Présentation de l’entreprise Présentation d’Odiwi Crée en Avril 2012, Odiwi est une agence digitale spécialisée dans le développement sur-mesure de sites web, d’applications web ainsi que dans le développement d’applications mobiles (détail sur les langages). De plus, Odiwi propose des contrats de maintenances system et logiciel pour ses clients et partenaires afin de garantir une fonctionnalité et un suivi exemplaire, des solutions fournis. Récemment, Odiwi a acquis la société crpi, une entité située sur Saint-Nazaire. Celle-ci est spécialisée dans l’intégration et la maintenance dans les domaines de l’informatique et de la téléphonie. Grace à cela, le groupe Odiwi son nombre de collaborateur s’accroître. Organisation et activités d’Odiwi

Direction

Commerciale

Developpement

Communication

Infrastructure

Activités réalisées durant le stage Planning du stage

Semaine 01 : réalisation du cahier de charge, recherches, gestion des comptes AD

Semaine 02 : installation Veeam + test de scripts, restauration de fichiers, migration de Boîtes mail et gestion des appels support

Semaine 03 : restauration de fichiers, Configuration de postes client, Livraison et installation des postes, création de fichiers de configuration sur le serveur Nagios, migration de boîte mail, gestion des appels support

Semaine 04 : Remonté d’information de job Veeam sur Nagios, déplacement chez le client, récupération des fichiers transfert des données sur VM chez OVH, migration de boîtes mail et gestion des appels support

Semaine 05 : Modification du plugin, définition de niveaux d’alerte, supervision de jobs d’un serveur, configuration d’accès distant aux serveur migration de boîtes mail et gestion des appels support.

Semaine 06 : recherche type de remonté d’information Veeam, migration de boîtes mail et gestion des appels support.

Semaine 07 : déploiement du service de supervision des jobs Veeam sur le serveur de production, migration de boîtes mail, gestion des appels

Analyse des besoins

Le jour même de mon arrivée chez Odiwi, les différentes missions que j’allais réaliser durant la période de mon stage, m’ont été communiquées. La première tâche sur laquelle je me suis penché fut la réalisation d’un cahier des charges, dans le but d’estimer le temps dont j’aurai besoin dans le futur.

De plus, la réalisation d’un cahier des charges me permet, une fois qu’une tâche est terminée, de noter le temps qu’il m’a fallut pour réaliser la mission. De ce fait, je peux comparer le résultat avec l’estimation et me rendre compte de ce qui a pu se passer plus ou moins comme prévu.

Mise en service de postes et d’outils de travail

Poste personnel

Pour pouvoir me mettre au travail, il m’a été fourni un poste vierge ayant servi auparavant, j’ai donc dû le configurer à l’aide d’un CD bootable. J’ai pu installer un Windows Server 2012RE depuis le BIOS et changer de disque de stockage.

Par la suite, J’ai utilisé la fonctionnalité Hyper-V à mon serveur et je lui ai fournis une adresse IP fixe en fonction du réseau. J’ai utilisé un Windows server afin de visualiser le comportement de d’un Active Directory car plus tard j’allais m’en servir dans certaines tâches.

Sur ce serveur, j’y ai configuré deux RDS (Remote DeSktop) afin de pouvoir travailler avec un serveur Nagios de test, et un serveur de production. Le but est de mettre en place l’architecture nécessaire à la communication entre un logiciel et le serveur Nagios. Le second serveur est présent car c’est le serveur client sur lequel j’installe la configuration finale.

Poste client

Dans le cas des configurations postes clients, il m’est arrivé de réceptionner des machines ayant été achetées pour des clients, afin de les configurer entièrement au préalable. Après c’être logger en administrateur. Je crée une session utilisateur avec les droits administrateurs en suivant les normes de nommage de l’entreprise, ainsi qu’une seconde session Odiwi afin de pouvoir prendre le contrôle à distance de la machine.

Par la suite on désinstalle les logiciels inutiles, et on y configure les bureaux à distance en fonction des informations que le client nous a fourni lors de la demande de configuration des postes et on installe les différentes mises à jour (système, driver).

Puis vient le moment de la livraison des produits, il est nécessaire de communiquer avec les clients afin de leur faire comprendre ce que nous faisons lors du changement de matériel, afin qu’ils puissent utiliser par la suite sans difficulté, leurs manipulations futures. Je finis par leur configurer les imprimantes en redirection, afin de pour les utiliser même sur serveur.

Création de nouveau users dans l’AD/Création de boîtes mail

Depuis la console exchange, un outil permettant de créer des utilisateurs ainsi que des boîtes mail. Comme je l’ai précisé précédemment, il était important que je comprenne le fonctionnement de l’AD, le principe de forêt, qui se résumerait à une zone et le principe de l’O.U, qui lui serait représenté par un groupe.

Exchange est un outil Microsoft permettant de créer un utilisateur ainsi que sa boîte mail sans avoir à se logger et à agir sur le serveur AD. Une fois les deux outils rattachés, lors de la création des utilisateurs depuis l’interface Exchange, il m’a fallu remplir les formulaires les concernant, tout en respectant les normes suivies par Odiwi lors de cette réalisation. Nous y renseignons nom et prénom, puis l’adresse mail qui sera créée pour l’utilisateur ainsi que la forêt (qui correspond au domaine de l’entreprise) et l’O.U (qui renseigne le serveur SMTP sur lequel sa boîte sera stockée).

Une fois créé, le compte est ajouté dans l’AD dans la forêt et l’O.U indiqués. De base sur Exchange, une boîte est créée avec le domaine Odiwi.net, il suffit de le passer en secondaire et le supprimer puis passer la boîte qui a été créée en primaire, celle-ci aura le protocole SMTP actif.

Pour vérifier que la tâche a été bien réalisée, on se log depuis le navigateur au serveur de messagerie, si à l’entrer des informations, il nous est demandé de configurer un fuseau horaire, c’est que la manipulation s’est bien déroulée.

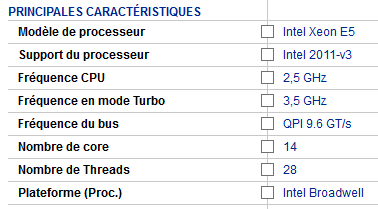
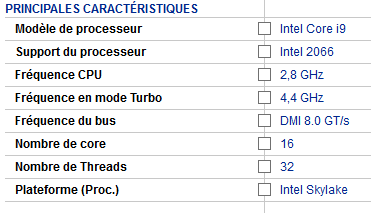
Il est nécessaire d’assurer un suivi au client une fois que la création de ses données a été réalisée, afin qu’il sache de quelle manière il pourra par la suite accéder à ses informations.

Recherches personnelles

Afin de comprendre le fonctionnement des éléments au sein d’une machine j’ai dû réaliser quelques recherches à leur sujet. Notamment, dans le but de savoir la différence entre les composants présent dans un pc portable et ceux dans un serveur.

Car en effet, des différences sont clairement présentes, ce n’est pas pour rien que la marque Intel par exemple, dispose d’une gamme de produit de processeur différente en fonction de leur utilisation.

Intel Xeon E5-2690 v4 (2.6 GHz) Intel Core i9-7960X (2.8 GHz)



Threads = nombre traitement réalisé en simultané

Un utilisateur lambda pourrait s’arrêter à la fréquence affichée dans la fiche technique des produits si présent et juger que le processeur I9 est plus performent. Cependant d’autres éléments entre en jeux, tel que le nombre de cœur, qui nous renseigne sur le nombre de thread exécutable par cœur ainsi que la mémoire cache. Il est possible de créer 2 cœurs logiques pour un cœur physique cela s’appelle l’Hyper-threading, c’est un procédé qui nous permettra d’augmenter le nombre de tâche réalisé en simultané.

Du coup, en comparant deux CPU identique, l’un avec ces cœurs logiques et l’autre n’en possédant pas, on remarque un accroissement des performances de l’élément de 30% environ.

Ces cœurs logiques sont très utiles notamment dans le cas de la virtualisation. Il sera donc possible d’attribuer à des machines virtuelles des cœurs logiques afin de laisser le reste réaliser des tâches que la machine physique aura à réaliser et ceci est très avantageux car il est important de comprendre que ces clones sont des répliques identiques des originaux.

A cela on ajoute donc la RAM et le Cache, l’un étant présent pour limiter les échanges avec le disque dur, et l’autre avec la RAM. Ces deux éléments vont stocker les données des applications, logiciels qui sont démarrés.

Sans ces deux éléments, le processeur devrait interroger directement le disque dur afin d’avoir ses informations. Avec la RAM celui-ci va interagir sans forcément communiquer avec le disque. Le Cache, lui, est directement intégré au processeur, ce qui accélère l’échange de données (et limite aussi sa taille) du fait que le processeur ait peu de circuit à parcourir pour arriver à interroger celui-ci. Si une activité est réalisée un grand nombre de fois par le processeur, celui-ci viendra à la stocker dans son cache afin d’avoir accès aux données très rapidement.

Les caches se différencies en fonction du volume de données qu’ils sont capables de stocker allant de L1 à L3 plus le niveau est bas (L1) plus le cache est petit et plus rapide est le transfert de données, du fait qu’il y a moins de volume de données à lire et plus le niveau est élevé, plus le volume sera conséquent et lent au transfert.

Je vous invite à regarder ce schéma réalisé par A.L. Fortune Secondary. Se trouvant en annexe, pour comprendre le fonctionnement de communication entre le processeur, le cache et la RAM.

Dans ce cas de comparaison on pourrait penser que le processeur série I9 serait bien plus viable pour un serveur, cependant il n’a pas été conçu pour fonctionner 24/7, la où c’est le cas pour le Xeon, qui contient en plus de ça, un cache 1.5 fois plus volumineux que celui présent sur le I9.

A noter que ce processeur, n’est sans nul doute pas à la portée de tout le monde et que son prix à lui tout seul pourrait servir à acheter toute une tour.

Recherche de solution et restauration de fichiers

Suite à de nombreux soucis que la société ATLANTA a rencontré avec ses disques durs, la mission de recherche de solution de restauration de fichier m’a été attribuée. ATLANTA est un centre axé dans la recherche des pathologies osseuses et articulaires, qu’elles soient tumorales ou non.

Au jour d’aujourd’hui, Nous sommes déjà en possession d’un outil de scan et de restauration dans l’entreprise, celui-ci se nomme diskGenius. Cependant pour scanner la totalité des secteurs du disque en profondeurs, cela nous prenait entre 20 et 24 heures. EaseUs et Active@ sont ceux que j’ai mentionné.

Suite à des tests sur les versions gratuites de ces produits, la société s’est penchée sur le logiciel Active@ car il est capable de restaurer non seulement les fichiers mais aussi l’arborescence des dossiers qui a disparu. EaseUs est lui aussi très performent si ce n’est plus, il scan et restaure les documents beaucoup plus vites, et trouve bien plus d’élément, cependant, lors des tests nous nous sommes rendu compte que le logiciel plantait si le volume du disque est trop grand, c’est-à-dire, lorsque le volume de données dépassait les 2 To.

De plus, à l’aide des informations du client concernant les données à récupérer, nous trions les différents dossiers, ainsi que les fichiers se trouvant dans les lostfiles, un répertoire listant les fichiers qui ont perdus la structure des dossiers ou alors qui ont été supprimés ultérieurement.

A l’aide des normes d’attribution de nom aux fichiers, la mission a été facilité. Dans le nom de chaque fichier se trouve une lettre et des valeurs ou des mots clés, nous permettant d’identifier le dossier auquel il appartient. La grande majorité des données ont été récupérée et les disques ont été renvoyés à l’entreprise.

Soutien technique réseau

Il se pouvait que je traite des appels de client ayant des soucis avec leur réseau, que ce soit à domicile ou en entreprise. L’important est de bien prendre en compte ce que la personne a à dire, les circonstances qu’elle a vécu avant de rencontrer le problème pour lequel elle nous a contacté.

Ainsi vient la partie de prise en main à distance avec les différents outils utilisés par Odiwi tel que TeamViewer, Ammyy ou encore LogMeIn. La plupart des soucis que j’ai eu à traiter résultait d’une configuration de proxy ou alors de DNS. Pour illustrer l’exemple du Proxy, j’ai notamment eu à faire avec une cliente qui rencontrait des soucis avec son poste fixe avec lequel elle travaille.

Le message qui lui était remonté lorsqu’elle essayait d’accéder à google via navigateur était le suivant « connexion expiré, délai dépassé » et différent élément pouvait en être la cause, problème DNS, PROXY …

J’élimine donc les possible éventualité une par une, pour vérifier si le problème venait d’un enregistrement DNS, j’ai réalisé quelques commandes tel que :

ipconfig /release : qui nous permet de libérer les connexion établis.

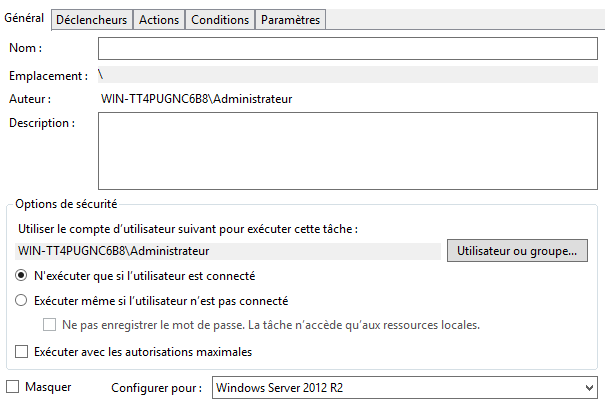
ipconfig /renew : qui permet de renouveler les informations des cartes, c’est-à-dire que cette commande informe le DHCP de fournir à nouveau à la machine une adresse IP.

ipconfig /displaydns : afin d’afficher les enregistrement DNS .

ipconfig /flushdns : pour vider le cache DNS.

ipconfig /registerdns : qui va ensuite actualiser les baux DHCP et réinscrire les nom DNS.

Malgré cela, les tests de communication avec le DNS google se font, mais toujours pas d’accès à la page [www.google.fr](http://www.google.fr), Sur google chrome et Mozilla Firefox, les informations sont très peu précises lorsque l’on souhaite résoudre des soucis réseaux. Pour cerner le problème de manière précise, j’utilise donc internet exploreur, qui lui, détecte une source proxy, j’en conclue donc que la machine de l’utilisatrice conservait les configurations réseaux de son entreprise. Depuis son navigateur j’ai donc désactivé les paramètres d’utilisation de proxy, ce qui a permis à la cliente de réaliser de nouveau ses recherches.

De plus, j’ai eu à réaliser la configuration de tâches de redémarrage pour des clients sur plusieurs serveurs Windows 2012.

Pour cela il faut accéder au planificateur de tâche depuis le menu en bas à gauche. Dans le menu actions du planificateur de tâches, on sélectionne créer une tâche. Une fois cela fait, on renseigne le nom de la tâche, ainsi que l’utilisateur qui exécutera la tâche.

Ensuite, on configure le déclencheur, c’est-à-dire la date à laquelle la tâche devra s’accomplir ainsi que la fréquence d’exécution personnellement, les configurations que je réalisais était hebdomadaire.

Et pour finir, on définit ce qui sera réalisé en sélectionnant le programme qui va bien ainsi que les arguments qui nous intéressent, la tâche consiste à réaliser un redémarrage du serveur à une certaine fréquence pour une heure donnée. Je sélectionne donc le programme shutdown.exe présent dans le dossier sys32 de Windows, qui à l’exécution de la tâche, redémarre la machine et j’attribue les arguments -r et -f qui réfèrent à « redémarrer » et « force »

-r = Redémarrage de la machine

-f = Forcer l’arrêt des programmes en cours d’utilisation avant redémarrage de la machine

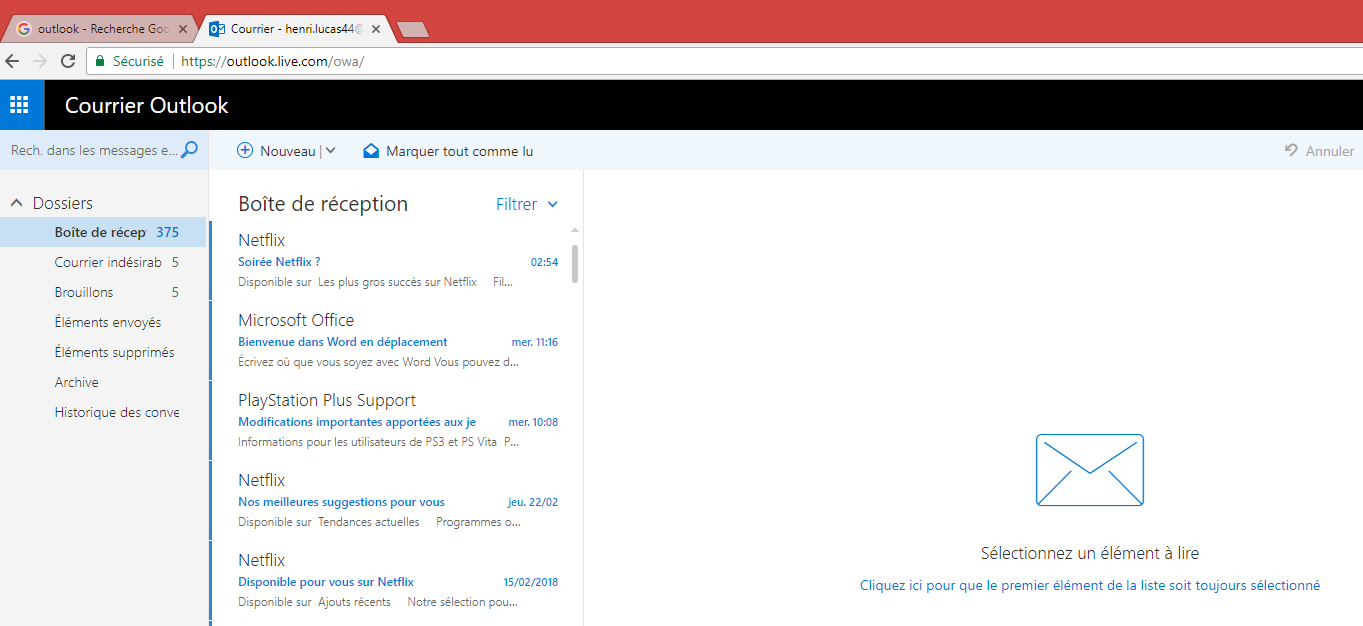
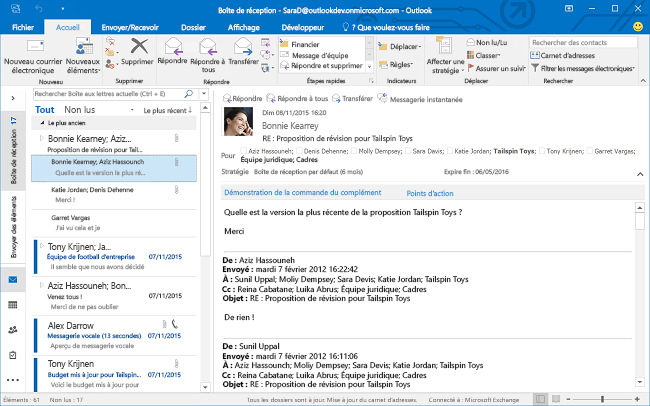
Suivi de mail et d’appel / Migration des boîtes mail Leolagrange

Le groupe LeoLagrange est une association à but non lucratif. Elle intervient dans les domaines de l’animation, la formation et l’accompagnement des personnalité public dans la mise en œuvre de la de politiques éducatives. LeoLagrange est notamment présent en Afrique, avec un total de 10 fédération.

La mission de transfert des différents collaborateurs nous a été fournis par M.Balloy Laurent. Celle-ci consiste à contacter la totalité des collaborateurs LeoLagrange dans la région, en passant par Rennes, Saint-Nazaire jusqu’à Nantes. La liste des contacts qui à joindre comportait différents niveaux de priorité allant d’un à quatre, plus la priorité est basse plus il est important d’intervenir. Cette priorité est déterminée en fonction du site ainsi qu’en fonction du volume de données à transférer.

Le principe de cette migration consiste créer une nouvelle boîte mail office365 et d’y copier les différents fichiers pst présent sur le poste de l’utilisateur. Une fois la copie réalisée et le transfert achevé, l’utilisateur a la possibilité de se connecter à sa nouvelle boîte mail directement sur le serveur de messagerie afin qu’i puisse consulter ses mails.

Ce processus permet de garantir aux utilisateurs un accès en permanence à leurs mails. En effet, si leur outil de travail où se trouve les différents fichiers pst est endommagé, ou volé avant ce transfert, ces personnes ne pourront plus traiter leurs mails.



Voici donc les différentes interfaces que l’utilisateur a rencontré, et qu’il rencontrera. A gauche nous avons l’interface applicatif d’Outlook, en fonction de la version celle-ci peut être différente, et à droite, l’interface sur laquelle il pourra travailler à l’avenir.

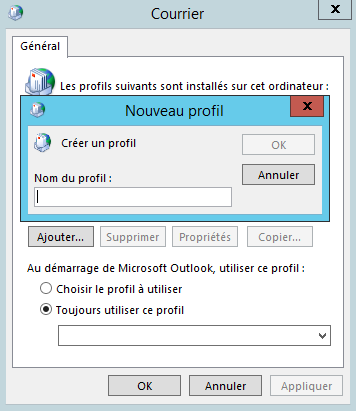
Durant la communication avec le client, ma mission est d’informer au préalable l’utilisateur du but de la migration ainsi que du temps que cela va me prendre pour la réaliser. En général, je réalise le transfert en 10 voire 15 minutes, ça s’est dans le cas où l’utilisateur a un outil de prise en main à distance comme TeamViewer sur son poste ainsi que la totalité des informations concernant son compte Outlook, c’est-à-dire adresse mail mot de passe.

La plupart des utilisateurs font mémoriser leurs données aux outils qu’ils utilisent, et certains viennent à perdre leur mot de passe ce qui est problématique. Dans ce genre de cas, je télécharge PassView, un outil qui me permet à l’aide des fichiers pst de l’utilisateur, de récupérer son mot de passe.

Une fois les données en ma possession, je vérifie l’état actuel de la configuration de la boîte mail, si le protocole de stockage des mails est en POP ou en IMAP, dans le cas où celui-ci est en POP je réalise la migration, car les mails sont stockés uniquement sur le poste et dans l’autre, je ne la réalise pas, et j’informe l’utilisateur qu’il a la possibilité d’accéder directement via navigateur au serveur de messagerie pour consulter ses mails.

La première étape consiste à désactiver la configuration de suppression des mails antérieur à 14 jours sur le serveur, puis je localise l’emplacement des fichiers pst de l’utilisateur sur le poste et enfin, Pour que l’utilisateur reçoit les mails qu’il pourrait avoir durant la manipulation.

Pour créer le nouveau compte office365, l’application Outlook doit être fermée. Je me rends dans la rubrique courrier ou mail du panneau configuration. La fenêtre suivante est donc censée s’afficher.



Après avoir renseigné le nom du profil, j’ai la possibilité d’y indiquer le mail ainsi que le mot de passe qui m’a été fournis par l’utilisateur, puis je définis le nouveau compte comme étant le compte par défaut. Vient ensuite le moment de l’import des fichiers.

Pour cela j’ouvre à nouveau l’application Outlook et je vérifie que les modifications qui ont précédés l’étape qui suit ont bien été appliquées. Si la messagerie est vide et le protocole en IMAP cela veut dire que tout s’est bien passé.

Récupération de données / transfert sur cloud chez OVH

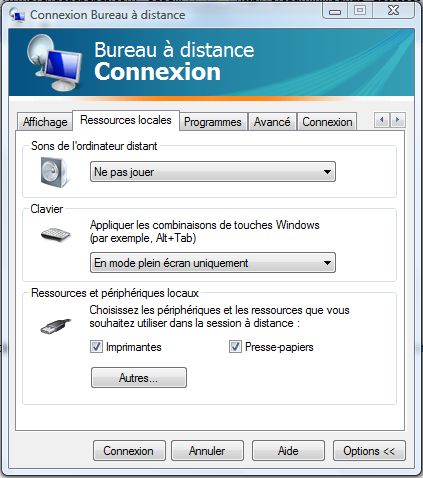
Les vignobles « Cheneau et fils » situé sur Mouzillon bourg, sont des clients de l’entreprise Odiwi, étant spécialisés dans la production de muscadet. Ils disposent d’un ensemble de solution informatique distribué par ISAGRI, qui sont conçu pour répondre aux besoins administratifs.

Le client nous a fait parvenir une demande de transfert des différentes données présentes sur leur serveur local vers le cloud car ils souhaitaient pouvoir avoir accès aux outils de travail à distance, sans forcément se trouver dans les locaux.

Le client avait une idée de ce qu’il souhaitait avoir, cependant il n’avait rien préparé, nous n’avions quasiment aucunes informations en termes de volume de données que nous allions récupérer,

Nous avons donc été vigilent, avant de nous y rendre, avec ma tutrice, nous avons préparé un disque dur 2To ainsi qu’un disque portatif de 1To afin de récupérer la totalité de leurs données sans avoir à faire d’aller-retour.

Pour travailler sur le serveur, la secrétaire se connectait en bureau à distance cependant elle se déplaçait jusqu’au serveur afin d’y connecter sa clé USB à la machine, ce qui est problématique. Dans les configurations du bureau à distance, on indique dans les ressources et périphériques locaux que l’on veut avoir accès à l’imprimante et dans autres, les périphériques USB.



Le patron, Mr.Cheneau souhaitait récupérer les logiciels ainsi que leurs licences sur serveur et les données qui vont avec. Les travaux de ses précédents collaborateurs et des collaborateurs actuels devaient l’être aussi. Concernant les précédents salariés, il souhaitait que leurs travaux soient présents dans un partage de fichiers entre les différents membres de l’entreprise, Les données des membres actuels devait être propres à leur session.

Nous avons donc commencé par déterminer le volume de données à récupérer et à transférer sur le cloud. Nous avons pris à l’aide d’un docking station sur lequel était connecté notre disque de 2 To, la totalité du disque C:\ du serveur. Sur le disque portatif, nous avons récupéré les données utilisateurs et fichiers pst des boîtes mails.

Le temps de traitement s’est avéré long, nous avons donc pris la décision de laisser notre équipement sur place, et de repasser le lendemain. Une fois les données en notre possession, Bertrand OUARY, l’un des deux directeurs d’Odiwi, c’est chargé d’acheter le serveur de l’entreprise « Cheneau et fils » chez OVH, le leader français dans les services de cloud computing et il y a copié les données que nous avions collectés.

Par la suite, nous avons configurés les différentes boîtes mails de la même manière que pour les différents utilisateurs du groupe Leolagrange.

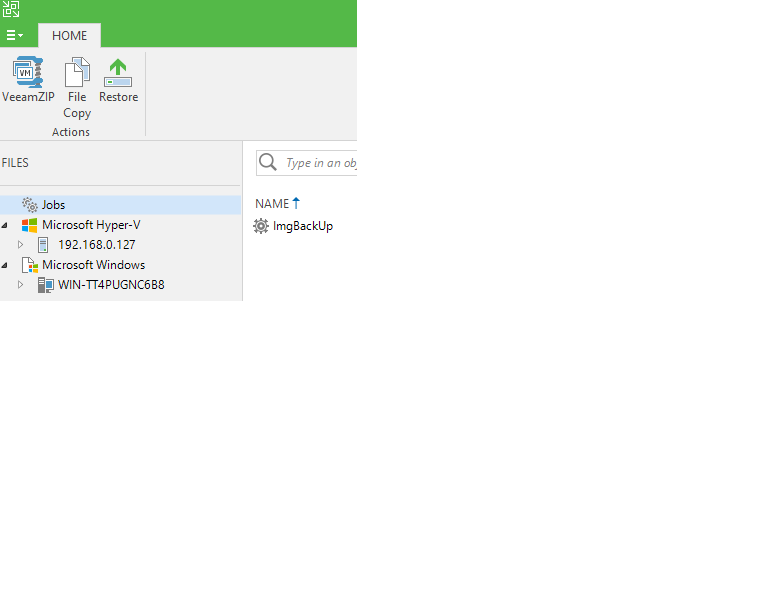
Supervision d’un serveur Veeam à l’aide de Nagios

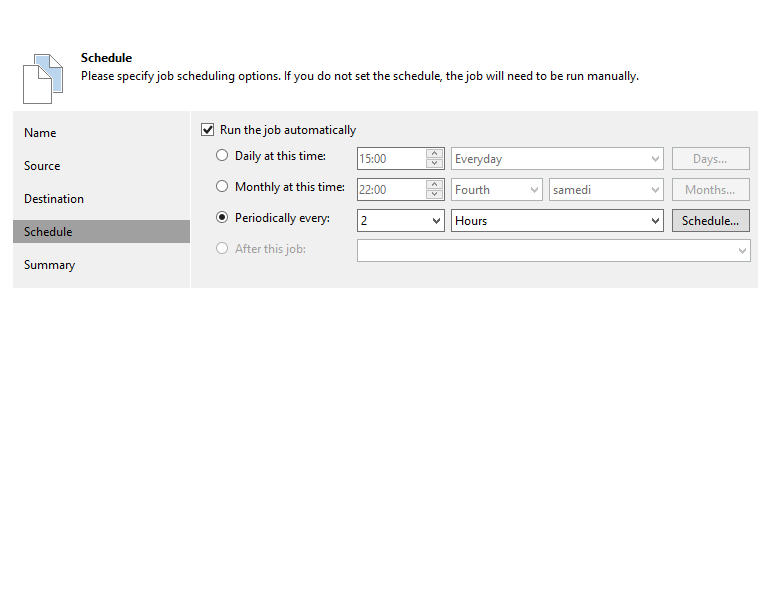
Création d’un job sous Veeam

Monitorer l’état des jobs de Veeam était l’une des missions principales que j’ai eu à réaliser. Veeam est un groupe privé spécialisé dans les différentes technologies d’informations, de sauvegardes, de récupération de données ainsi que d’administration d’environnements.

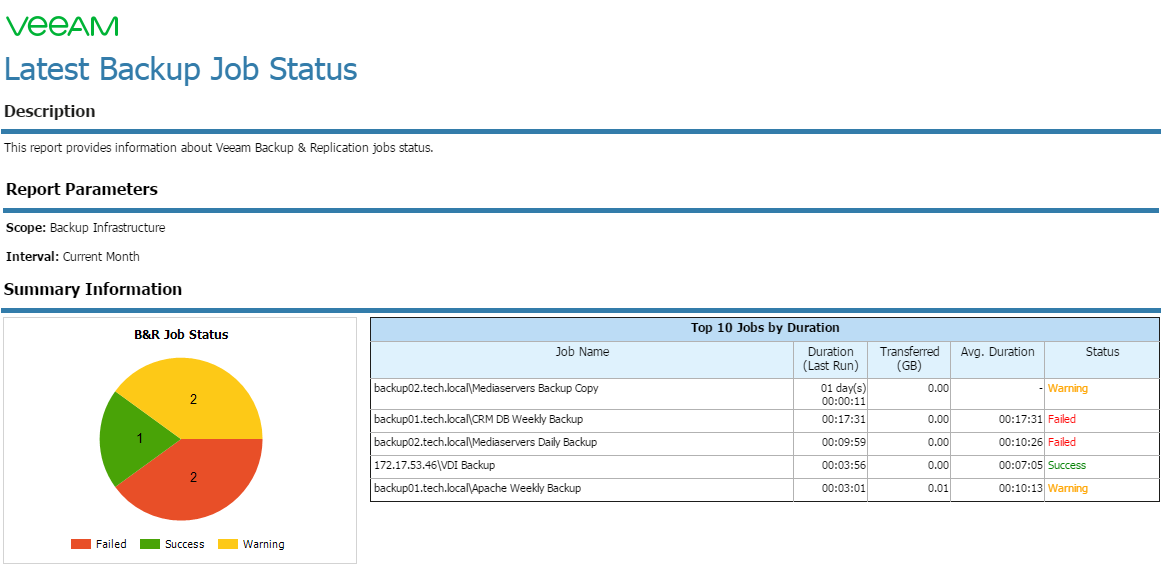
Veeam propose un panel d’offre varié, pour tout type d’utilisateur. Cependant, pour se faire la main dessus, la version gratuite (Veeam one) est celle que je recommanderai. Les exemples qui suivront seront réalisés à l’aide de la version d’essai de l’outil Veeam Backup & replication dans sa version 9.5

La création d’un job consiste à sauvegarder un élément à l’état dans lequel celui-ci était, juste avant le processus. Je me suis donc lancé sur la création d’un job file.



****La première étape est de le configurer. En effet, un job a besoin d’information pour fonctionner correctement. Celui-ci doit connaître l’emplacement ainsi que l’élément à sauvegarder. Par la suite, la destination doit être renseigné, c’est la machine, le répertoire sur où cette sauvegarde sera stockée. Enfin vient le moment où l’on doit indiquer la fréquence d’exécution du job.

Une fois le job créé, je test sont bon fonctionnement en l’exécutant et en vérifiant si l’emplacement de stockage contient bien la sauvegarde. De plus, après l’exécution d’un job, son état nous est renvoyé. Il y a trois différents états d’alertes que peuvent remonter les jobs, le SUCCESS, le FAILED et le WARNING ainsi qu’un message généré par l’outil Veeam, expliquant le déroulement lors de son exécution dans le report du job.

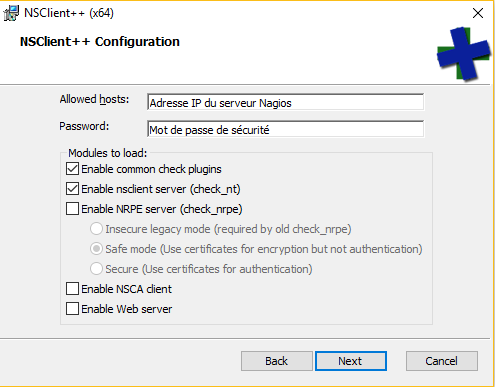


Installation et configuration de l’agent NSClient

NSClient est un agent de supervision qui à la base, est utilisé pour travailler de pair avec un serveur Nagios. Cependant il est possible de le trouver en fonctionnement avec un autre outil de supervision tel que shinken ou encore Zabbix. Cet agent s’installe côté client et fonctionne sous tout type de système d’exploitation.

L’installation est simple, sur le poste que l’on souhaite monitorer, il est nécessaire de se rendre sur le site web Veeam, c’est là que l’on retrouve les différents produits Veeam. Au lancement de l’exécutable, j’ai eu la possibilité de choisir une des trois méthodes d’installations, typical custom ou complete. Ce choix a un impact sur la méthode de travail de l’agent.

Ensuite, il est possible de sélectionner dès le départ les modules avec lesquelles nous souhaiterons travailler.

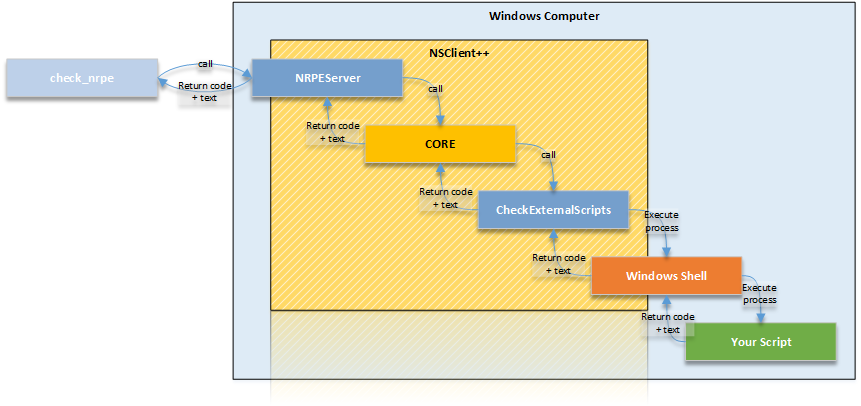


La prise en compte des différents plugins est quelque chose d’indispensable car je travaille avec un script et pour la réalisation de ma mission, l’agent doit être en mesure d’exécuter mon script. Puis les deux types de check actif son à autoriser car Odiwi souhaite avoir le moins de configuration possible à faire côté client.

En active check, c’est le serveur Nagios qui va établir la communication avec les différentes machines clientes. En fonction de sa configuration toutes les X secondes, le serveur interroge les postes à l’aide d’une commande personnalisée, et utilise un type de protocole de communication NRPE ou nt, la différence entre les deux se fait surtout dans la sécurisation des données qui sont échangées entre les serveurs que propose le NRPE.

L’agent prend en compte la commande, suite à quelques modifications dans les fichiers de configurations NSClient et exécute le script, qui lui va nous récupérer les différents statuts de nos jobs et les stocker au sein d’un tableau. Le processus continu dans le sens inverse et remonte jusqu’au serveur Nagios.

Plus en détail, le processus qui est réalisé suite à l’appelle du serveur de supervision.



Les deux derniers éléments sont NSCA\_Client qui se résume en check passif. À l’inverse de l’actif, il permet au serveur client de fournir les différents résultats des jobs, sans en avoir reçu la demande au préalable par le serveur Nagios.

Puis il y a le Web server, un module qui peut s’avérer utile, surtout lorsque l’on souhaite avoir des résultats rapidement, en effet, ce module nous donne accès à une page web en local, sur laquelle les résultats des différents checks sont affichés.

Une fois installé les fichiers NSClient se trouve généralement dans le répertoire Program Files du disque C. En installant le NSClient à l’aide de la méthode typique, différents modules de check sont installés notamment le Check\_System et Check\_Disk qui permettent de récupérer des données concernant les composants machines, CPU, DISK, fréquence de travail des éléments.

Pour que les deux serveurs puissent échanger, il est nécessaire de renseigner l’adresse des différentes machines dans les fichiers de configurations, côté client et serveur. Certains modules pouvaient ne pas être actifs, pour remédier à cela, dans le fichier de configuration NSClient, je change leur état de disabled à enabled ou alors de 0 à 1.

Le contenu du script de configuration NSClient sera présent en annexe.

Plugin Veeam

Le script Veeam qui allait récupérer les informations des jobs était fourni par la communauté Nagios exchange, on y retrouve différents plugins pouvant servir à un grand nombre d’interaction client serveur.

Le script est rédigé en PowerShell. A l’exécution de celui-ci, il se pouvait que des erreurs lors de compilation me soit remontées. En effet dans le script, une bracket ouvrante était manquante mais ce n’était pas tout, le script n’arrivait pas à récupérer ni stocker les jobs ainsi que leurs informations. Suite à quelques recherches j’ai découvert deux méthodes pour résoudre ce souci.

La première méthode consiste à exécuter le script entant qu’administrateur, dans ce cas-là je possède une part des droits administrateurs et il me faut changer les droits d’ExecutionPolicy. En exécutant la commande set-ExecutionPolicy. Plusieurs paramètres sont envisageables.

Le restricted, qui est le paramètre par défaut permet aucune exécution de script.

Le AllSigned, qui nécessite un grand niveau de sécurité, car la totalité des scripts et fichiers de configurations doivent être signés sont comptés les scripts développés en local.

Le RemoteSigned qui souhaite uniquement avoir une signature pour les fichiers et scripts provenant d’internet, donc nous n’avons pas de restriction pour les scripts développés en local.

Le Unrestricted, permet d’exécuter tout type de script, cependant, un message de confirmation sera généré.

Le Bypass, ne bloque aucun script et ne génère aucun message de confirmation.

Le Undifined, qui lui retire le paramètre ExecutionPolicy actuel pour la session powershell en cours.

La seconde méthode consiste à débloquer le script à l’aide de cette commande :

PS C:\> Unblock-File -Path "Votre fichier"

Cela ne change pas le paramètre de l’ExecutionPolicy, le script sera débloqué et exécuté sous Windows PowerShell.

Sachant que je travaille uniquement avec un script je n’ai pas eu besoin de changer de paramètre, car oui, si je change de paramètre, cela sera effectif pour la totalité des scripts, et question sécurité les conditions ne sont pas terribles. J’ai donc débloqué le script après l’avoir lu et analysé afin de savoir ce qu’il réalisait.

De base, le script récupère le job à l’aide d’une fonction Get, ajouté par le Snapin et le stock dans une variable. Par la suite un jeu de test se fait en fonction du statut du job et un message ainsi qu’un code de retour est envoyé à l’utilisateur.

Le script fournit par Nagios Exchange sera présent en annexe.

Le problème avec ce script est qu’il ne prend en compte qu’un seul job. Or, Odiwi possède plusieurs jobs au sein de leur serveur Veeam des modifications étaient donc nécessaires.

Dans mon script, je fais un Get des jobs et je les stocks dans un tableau $job. Par la suite je réalise une condition d’entré dans la suite de mon programme, si mon tableau ne contient pas de job, s’il est nul, un message informant l’utilisateur et un code de sortie approprié lui sont fournis.

Sinon, le programme continu et entre dans une boucle for, je parcours la totalité des indexes où sont stockés mes jobs et j’y définis des variables qui récupéreront le nom et le statut du job.

Par la suite, je gère plusieurs conditions au sein de ma boucle for, afin d’avoir le message adapté suite au fonctionnement du job.

Si le statut de la dernière session est équivalant à WORKING je retourne un message disant que le job est en cours de réalisation.

Si le statut est équivalant à FAILED, j’informe l’utilisateur que le job a été interrompu durant le processus de backup.

Sinon si le statut n’est pas équivalant à SUCCESS j’informe l’utilisateur que le job ne s’est pas entièrement achevé.

J’ai gardé la gestion de la date du dernier lancement des jobs de l’ancien script car celui-ci fonctionnait très bien deux variables sont utilisées, $now et $last le now contient la date du jour et le last, la date du dernier lancement, qui est récupérée à l’aide de la fonction getscheduledoption. Si la date actuelle est plus grande que la date de la dernière session j’informe l’utilisateur que la dernière session lancée date d’un certain temps.

Dans mes conditions qui sont présente dans la boucle for, j’y intègre des booléens car au sein de l’interface web de Nagios le champ réservé aux messages ne peut en contenir qu’un seul. Avec le code actuel, les messages sont envoyés mais le champ est vide car Nagios en reçoit plus qu’il n’en faut.

Donc je gère hors de ma boucle des conditions qui s’appuie sur ces booléens.

Si le booléen concernant les erreurs est à true un message global sera envoyé. Celui dira, qu’un ou plusieurs jobs ont été interrompus durant le processus de backup.

Sinon si le booléen concernant les soucis mineurs est à true et que le booléen concernant les erreurs est à false, un message est envoyé et dit, qu’un ou plusieurs jobs ne se sont pas entièrement achevés.

Sinon, la totalité des jobs s’est achevée avec succès.

Mon script est présent en annexe.

Configuration client côté serveur Nagios

Le serveur Nagios étant déjà présent dans l’infrastructure, je n’ai pas eu à l’installé ni à le configuré. Pour interroger les serveurs clients à l’aide des différents protocoles NRPE et NT, j’ai modifié le script commands.cfg et créé un script de configuration pour le poste que je souhaitais monitorer.

Dans le script commands.cfg, j’y renseigne dans le define command, les éléments qui permettront au serveur d’ordonner l’exécution de mon script Veeam et la manière dont elles s’écrivent, en voici un example.

define command{

command\_name check\_Veeam\_jobs

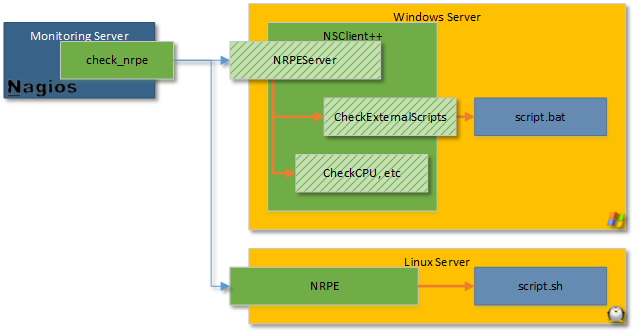
command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$

}

Puis dans le script dédié au poste, j’y renseigne dans un define host, les informations de la machine pour pouvoir communiquer depuis le serveur Nagios au poste client. C’est-à-dire que je lui donne un nom, celui-ci sera affiché sur l’interface web, l’adresse IP, et son usage.

De plus je souhaite que soit affiché les informations des différents services, donc je définis un define service pour chaque élément qui sera affiché sur l’interface web Nagios en y indiquant le nom du service, son usage, une description ainsi que la commande qui permettra d’ordonner la récupération des informations.

Cependant, seul mon service Veeam aura une configuration dans le fichier commands.cfg car l’echange se fait en NRPE et pas en nt car le Check\_NRPE permet de récupérer les informations suite au lancement d’une commande personnalisée.



La commande personnalisée est écrite dans le fichier de configuration NSClient dans la partie external script et exécute mon script une fois à l’aide de PowerShell. Le nom de la commande personnalisée doit être le même que dans le fichier commands.cfg ainsi que dans mon define service sinon des erreurs seront générées.

Réalisation de documentations

La date du stage s’approchant à grand pas, j’ai réalisé différentes documentations relatant les différentes étapes que j’ai eu à réaliser concernant la mission Veeam/Nagios. La première fut une documention personnelle, qui décrivait de manière détaillée la totalité des missions. Pour se faire, j’ai eu recours aux notes que je réalisais chaque jour, tout le long du stage.

La documentation technique, elle, avait pour but d’être un support pour ma tutrice ainsi que pour les autres collaborateurs, afin qu’ils n’aient pas à faire de recherche au sujet de la supervision d’un service Veeam et qu’ils puissent réaliser cette mission le plus rapidement possible en suivant pas à pas, les example que je leurs fournis.

Conclusion

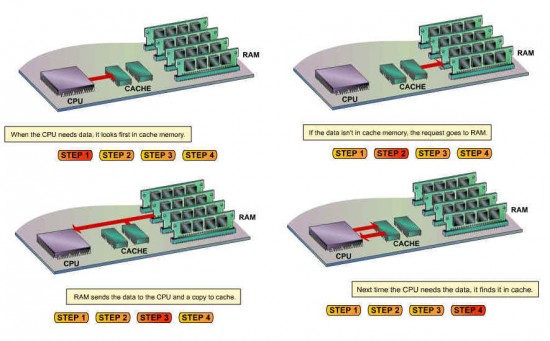
De cette expérience en entreprise, j’en ressors deux grands points majeurs concernant le poste d’administrateur infrastructure, le premier point concerne les clients, la personne en charge de ce poste est très régulièrement en communication avec les clients.

En effet, le moindre souci que ces personnes rencontreront seront principalement Résolus par cette personne, elle se doit d’avoir les connaissances, mais aussi la capacité d’assurer le service que propose son poste, j’entends par là le fait de prendre l’appel, écouter le client, résoudre les soucis, guider le client lors des manipulations et rester disponible si celui-ci en viendrait à recontacter l’administrateur.

Puis le second réside dans la notion de développement, car même si cela n’est pas le corps de métier principale de l’administrateur infrastructure, cette personne rencontrera obligatoirement des scripts, des lignes de codes, des programmes qu’il devra réaliser.

Annexes

Schéma CPU :



Script client côté Nagios server :

######Winservveeam########

# serveur : veeam test

# adresse :

##########################

define host{

use generic-host

host\_name odi-VeeamServTest

alias odi-VeeamServTest

address

}

define service {

use generic-service

host\_name odi-VeeamServTest

service\_description NSClient++ Version

check\_command check\_nt!CLIENTVERSION -s Veeam

}

define service {

use generic-service

host\_name odi-VeeamServTest

service\_description Uptime

check\_command check\_nt!UPTIME -s Veeam

}

define service {

use generic-service

host\_name odi-VeeamServTest

service\_description CPU load

check\_command check\_nt!CPULOAD!-l 5,80,90 -s Veeam

}

define service {

use generic-service

host\_name odi-VeeamServTest

service\_description Memory usage

check\_command check\_nt!MEMUSE!-w 85 -c 95 -s Veeam

}

define service {

use generic-service

host\_name odi-VeeamServTest

service\_description C:\ Drive space

check\_command check\_nt!USEDDISKSPACE!-l C -w 90 -c 95 -s Veeam

}

define service {

host\_name odi-VeeamServTest

service\_description Veeam-Backups

use generic-service

check\_command check\_nrpe!check\_veeam\_jobs! -s Veeam

}

Plugin Nagios Exchange :

#################################################################################

#################################################################################

#################### Made by Tytus Kurek on September 2012 ####################

#################################################################################

#################################################################################

### This is a Nagios Plugin destined to check the last status and last run ###

### of Veeam Backup & Replication job passed as an argument. ###

#################################################################################

#################################################################################

# Adding required SnapIn

asnp VeeamPSSnapin

# Global variables

$name = $args[0]

$period = $args[1]

# Veeam Backup & Replication job status check

$job = Get-VBRJob -Name $name

$name = "'" + $name + "'"

if ($job -eq $null)

{

Write-Host "UNKNOWN! No such a job: $name."

exit 3

}

$status = $job.GetLastResult()

if($($job.findlastsession()).State -eq "Working"){

Write-Host "OK - Job: $name is currently in progress."

exit 0

}

if ($status -eq "Failed")

{

Write-Host "CRITICAL! Errors were encountered during the backup process of the following job: $name."

exit 2

}

if ($status -ne "Success")

{

Write-Host "WARNING! Job $name didn't fully succeed."

exit 1

}

# Veeam Backup & Replication job last run check

$now = (Get-Date).AddDays(-$period)

$now = $now.ToString("yyyy-MM-dd")

$last = $job.GetScheduleOptions()

$last = $last -replace '.\*Latest run time: \[', ''

$last = $last -replace '\], Next run time: .\*', ''

$last = $last.split(' ')[0]

#changed by DY on 11/04/2014 based on comment from cmot-weasel at http://exchange.nagios.org/directory/Plugins/Backup-and-Recovery/Others/check\_veeam\_backups/details

#if ($now -gt $last)

if((Get-Date $now) -gt (Get-Date $last))

{

Write-Host "CRITICAL! Last run of job: $name more than $period days ago."

exit 2

}

else

{

Write-Host "OK! Backup process of job $name completed successfully."

exit 0

}

Mon script :

#################################################################################

#################################################################################

#################### Made by Henri PICHON ####################

#################################################################################

#################################################################################

### This is a Nagios Plugin destined to check the last status and last run ###

### of Veeam Backup & Replication job passed as an argument. ###

#################################################################################

#################################################################################

# Adding required SnapIn

asnp VeeamPSSnapin

# Global variables

$Fail = $false

$Warning = $false

$period = $args[1]

# Veeam Backup & Replication job status check

$job = Get-VBRJob

#Write-Host "Nombre de job créé :"$job.Count

# Condition d'entré dans mon programme, si aucun job

# n'a été créé, la suite du programme ne se réalise pas

if ($job -eq $null)

{

Write-Host "UNKNOWN! Aucun job de présent sur le serveur VEEAM"

exit 3

}

else

{

# On va réaliser une suite d'instruction sur chaque

# objet dans tableau

for ($i = 0; $i -lt $job.Count; $i++)

{

$jobname ="'" + $job[$i].Name + "'"

$status = $job[$i].GetLastResult()

#Write-Host $status

#write-host $jobname

if($($job.findlastsession()).State -eq "Working")

{

Write-Host "OK - Job: $jobname est en cours de réalisation."

}

if ($status -eq "Failed")

{

$Fail = $true

Write-Host "CRITIQUE! Un element a imterrompu le processus de backup de: $jobname."

}

elseif ($status -ne "Success")

{

$Warning = $true

Write-Host "ATTENTION! Le job $jobname ne s'est pas entierement acheve"

}

else

{

# Date du dernier Run des jobs

$now = (Get-Date).AddDays(-$period)

$now = $now.ToString("yyyy-MM-dd")

$last = $job.GetScheduleOptions()

$last = $last -replace '.\*Latest run time: \[', ''

$last = $last -replace '\], Next run time: .\*', ''

$last = $last.split(' ')[0]

# changed by DY on 11/04/2014 based on comment from cmot-weasel at http://exchange.nagios.org/directory/Plugins/Backup-and-Recovery/Others/check\_veeam\_backups/details

#if ($now -gt $last)

if((Get-Date $now) -gt (Get-Date $last))

{

Write-Host "ATTENTION! date du dernier lancement de: $jobname equivaut au $last"

exit 1

}

}

}

if($Fail -eq $true)

{

Write-Host "CRITIQUE! Un element a imterrompu le processus de backup d'un des jobs"

exit 2

}

elseif ($Warning -eq $true -And $Fail -eq $false)

{

Write-Host "ATTENTION! L'un des jobs ne s'est pas entierement acheve"

exit 1

}

else

{

Write-Host "OK! Le processus de backup des jobs s'est acheve avec succes"

exit 0

}

}

Fichier de configuration NSClient

# If you want to fill this file with all available options run the following command:

#   nscp settings --generate --add-defaults --load-all

# If you want to activate a module and bring in all its options use:

#   nscp settings --activate-module <MODULE NAME> --add-defaults

# For details run: nscp settings --help

; in flight - TODO

[/settings/default]

; Undocumented key

password = Veeam

allowed hosts = adresse serveur

[/settings/NSClient/Server]

port = 12489

; in flight - TODO

[/settings/NRPE/server]

port = 5666

insecure = true

allow arguments = 1

allow nasty characters=true

use SSL = 1

ssl options = no-sslv2,no-sslv3

verify mode = none

[/modules]

; Undocumented key

;CheckExternalScripts = enabled

; Undocumented key

;CheckHelpers = enable

; Undocumented key

;CheckEventLog = enabled

; Undocumented key

;CheckNSCP = enable

; Undocumented key

;CheckDisk = enabled

; Undocumented key

;CheckSystem = enabled

; Undocumented key

;NRPEServer = enabled

; Undocumented key

CheckExternalScripts = 1

; Undocumented key

CheckHelpers = 1

; Undocumented key

CheckNSCP = 1

; Undocumented key

CheckDisk = 1

; Undocumented key

CheckSystem = 1

; Undocumented key

NSClientServer = 1

; Undocumented key

CheckEventLog = 1

; Undocumented key

NSCAClient = 1

; Undocumented key

NRPEServer = 1

; Undocumented key

WEBServer = 1

[/settings/external scripts]

allow arguments = true

allow nasty characters=true

[/settings/external scripts/scripts]

check\_veeam\_jobs = cmd /c echo .\scripts\check\_veeam\_backups.ps1 ; exit $LastExitCode | powershell.exe -command -

Sources :

<https://wiki.monitoring-fr.org>

<https://exchange.nagios.org>

<https://docs.microsoft.com>

<https://docs.nsclient.org>

<https://doc.ubuntu-fr.org/>

<https://www.veeam.com/>

<https://www.easeus.fr>

<https://www.microsoft.com>

<http://www.lsoft.net/bootdisk.aspx>

<https://products.office.com/fr-fr/exchange/email>