Solutions de centralisation et gestion de configuration de parcs informatiques hétérogènes

Rapport de stage en entreprise

Licence Professionnelle Admisys

Année 2020-2021

Auteur : Nathalie Bomfleur

Table des matières

Remerciements

Je tiens à remercier toutes les personnes qui m'ont aidée lors de la mise en œuvre du projet de stage et de la rédaction du présent rapport.

Je remercie tout d'abord Monsieur Didier Mas et toute l'équipe de la Direction du Numérique Perpignan Méditerranée de m'avoir accueillie au sein de leur structure.

Merci à mon compagnon pour ses conseils et son soutien, à mon père Christian Bomfleur pour sa révision de l'orthographe et de la grammaire, et à Monsieur Figarola que je n'ai pu m'empêcher d'embêter avec mes questions.

Merci aussi à Mme El Yacoubi et Mme Calvet pour leur compréhension durant cette situation sanitaire exceptionnelle.

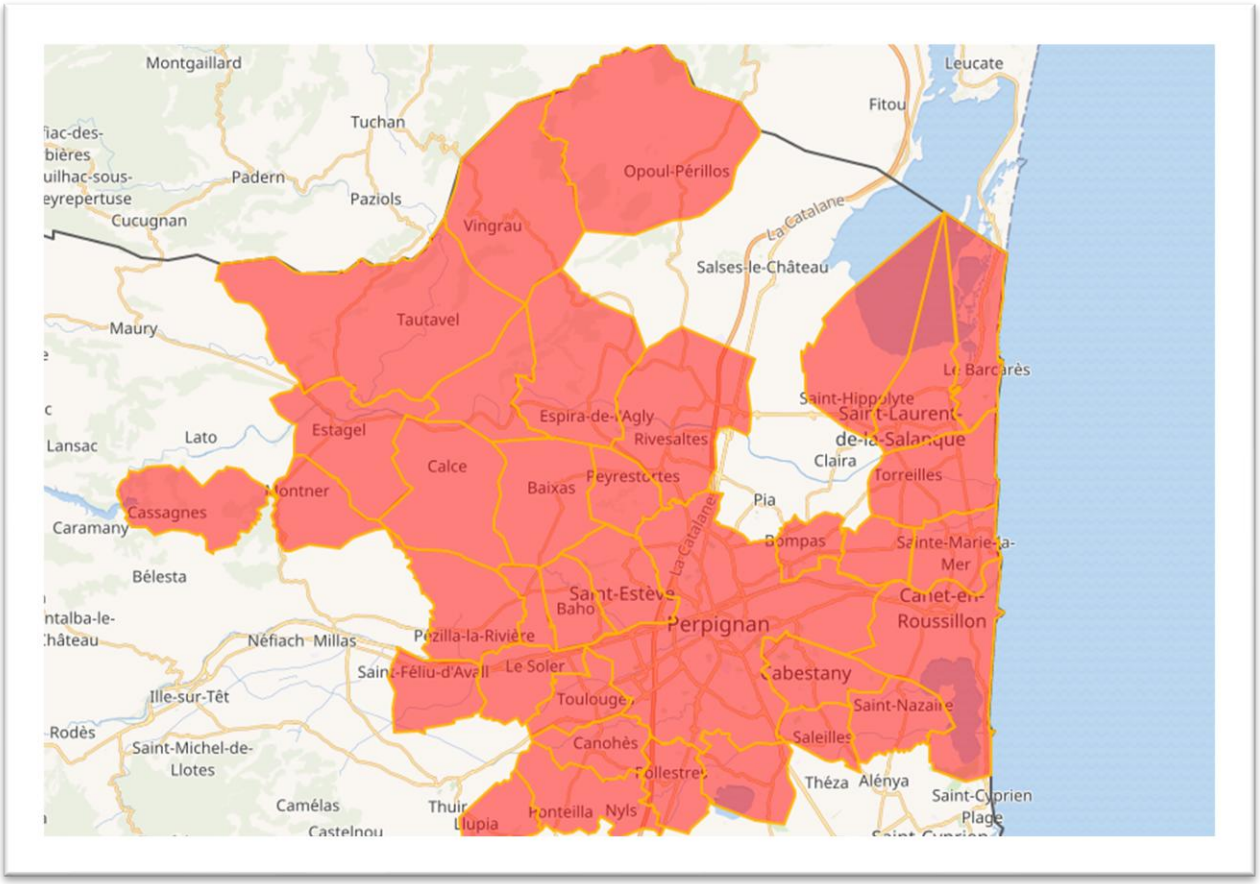
Présentation de l'entreprise

Perpignan, la plus méridionale des grandes communes françaises et préfecture du département des Pyrénées Orientales, s'étend sur 68,07 km² au cœur du Roussillon, dans la région Occitanie. L'ancienne capitale du royaume de Majorque fait aujourd'hui partie de la communauté urbaine Perpignan Méditerranée Métropole qui regroupe 36 communes limitrophes. Ce regroupement intercommunal se base sur la mutualisation de moyens humains, financiers et techniques et gère

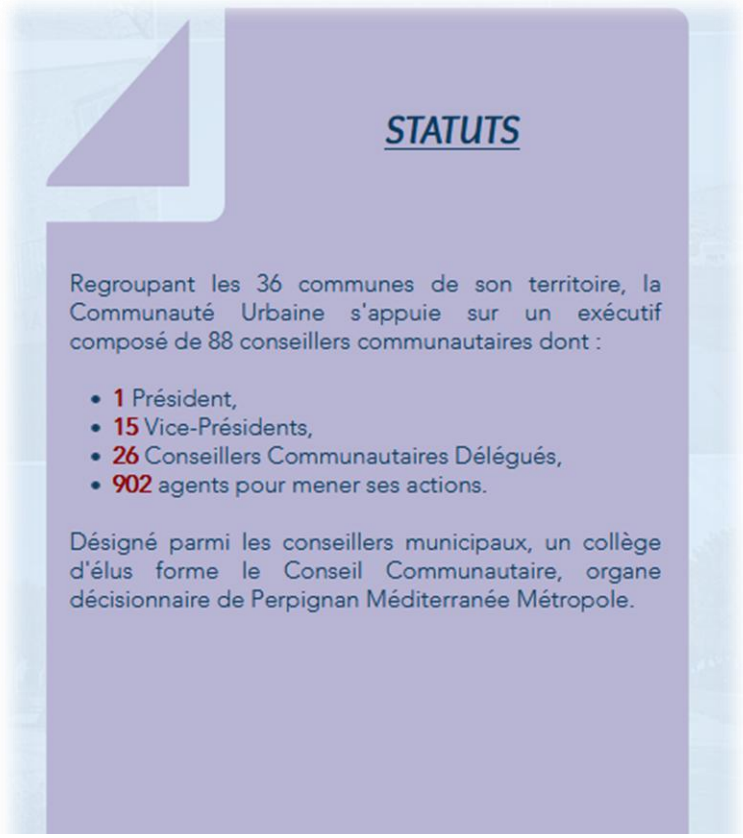
- Le développement économique, social et culturel (tourisme, conservatoire)
- L'aménagement de l'espace (espaces publics, transports en commun)
- L'équilibre social de l'habitat (logement social)
- La politique de ville (développement urbain, insertion économique et sociale, police municipale)
- Les services d'intérêt collectifs (eau, assainissement)
- Le cadre de vie (collecte de déchets ménagers, protection de sites naturels)

source: https://www.perpignanmediterraneemetropole.fr/qui-sommes-nous

Le territoire de Perpignan Méditerranée Métropole

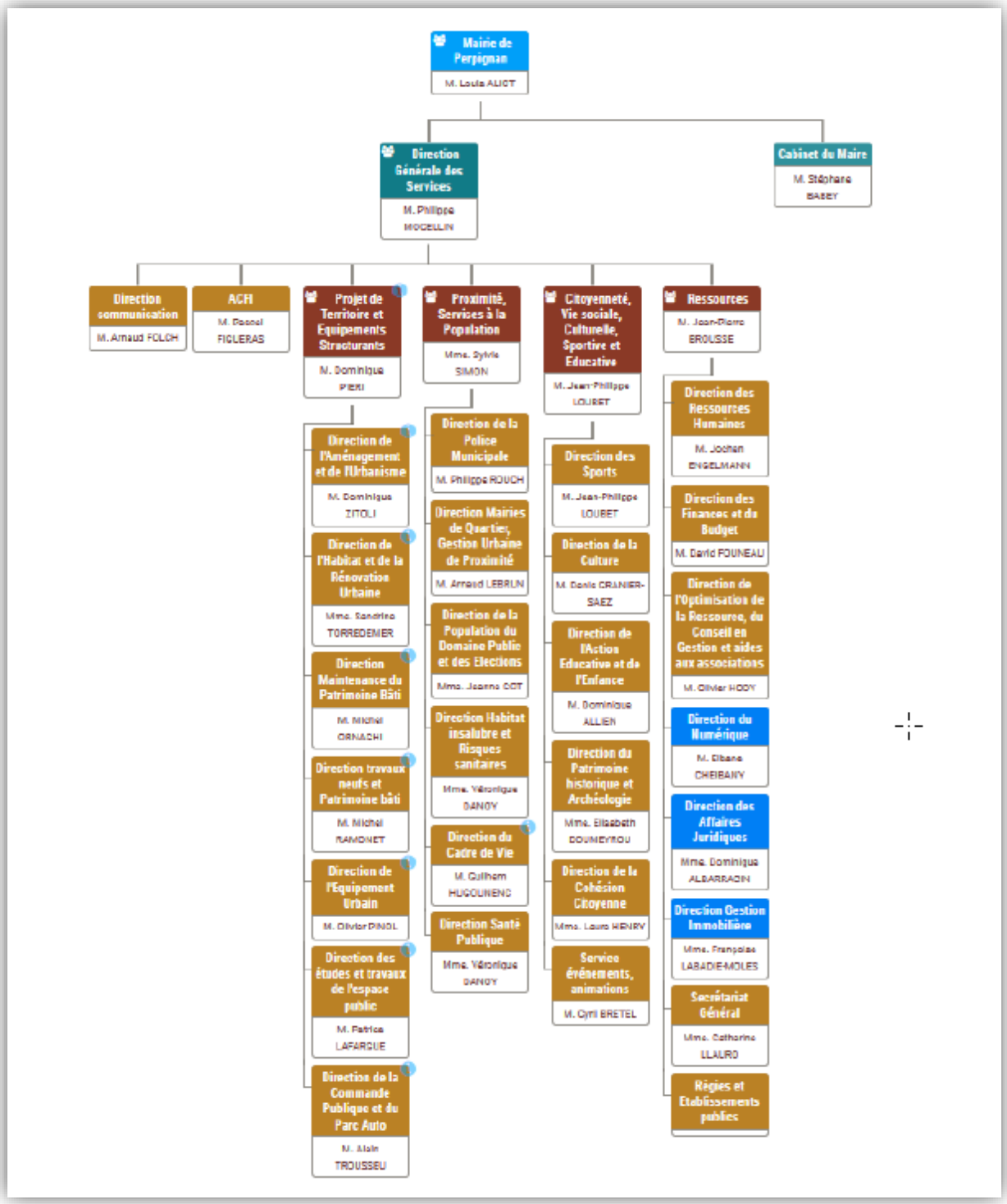


source: https://fr.wikipedia.org/wiki/Perpignan_M%C3%A9diterran%C3%A9e_M%C3%A9tropole#/map/0

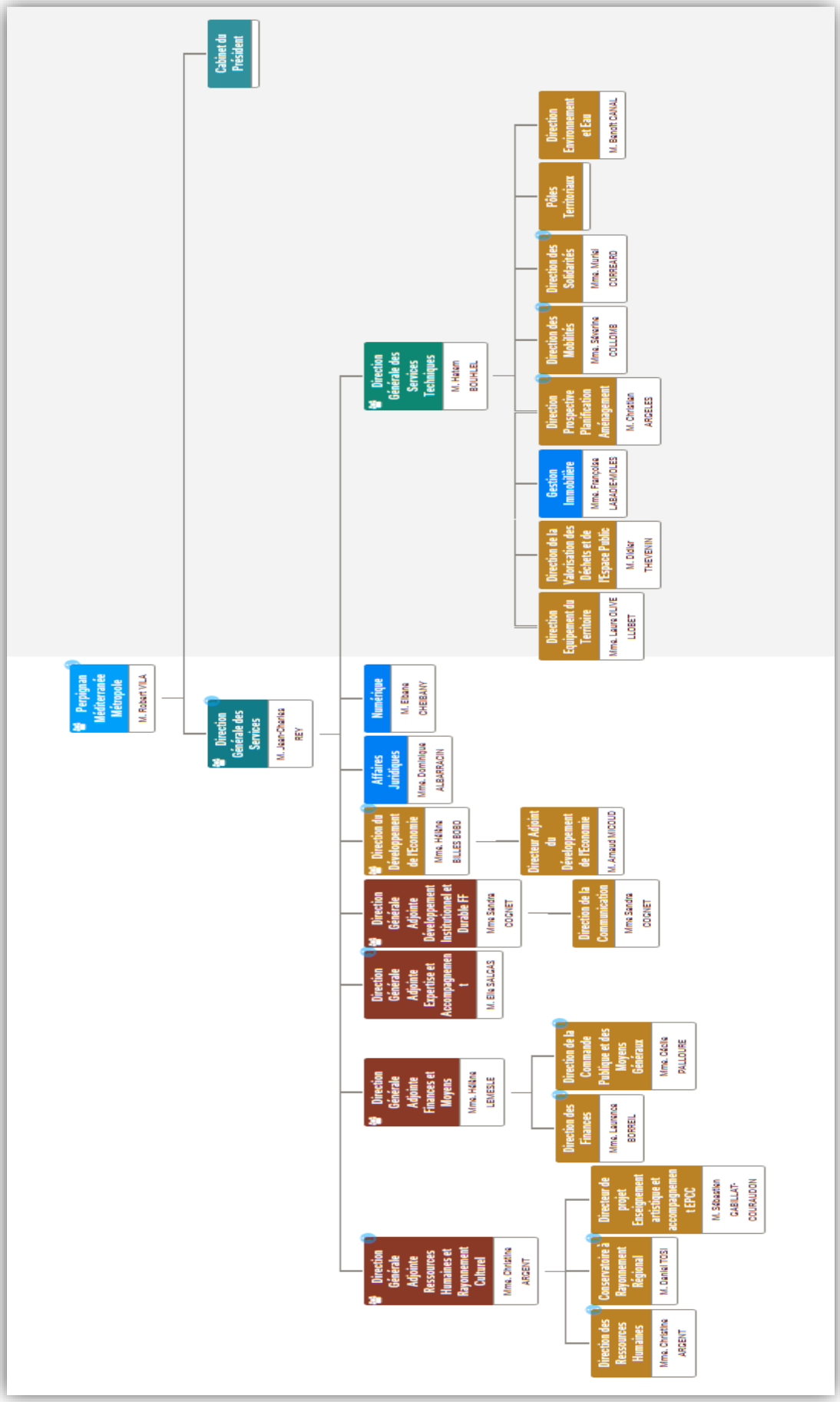


source: https://www.perpignanmediterraneemetropole.fr/qui-sommes-nous

Mairie de Perpignan



La communauté urbaine Perpignan Méditerranée Métropole



La direction du numérique Perpignan & Métropole

Mon stage s'est déroulé au sein de la direction du numérique, service mutualisé entre la ville de Perpignan et la communauté urbaine Perpignan Méditerranée Métropole depuis 2017. Ce service, qui gère les services informatiques et la télécommunication, emploie 40 agents sous la direction de Monsieur Elbane CHEIBANY. Il comporte deux divisions aux missions différentes :

La division Systèmes d'information

-validation et pilotage de projets

-formation du personnel

-documentation interne

-gestion des sites internet de la vielle et de l'office du tourisme

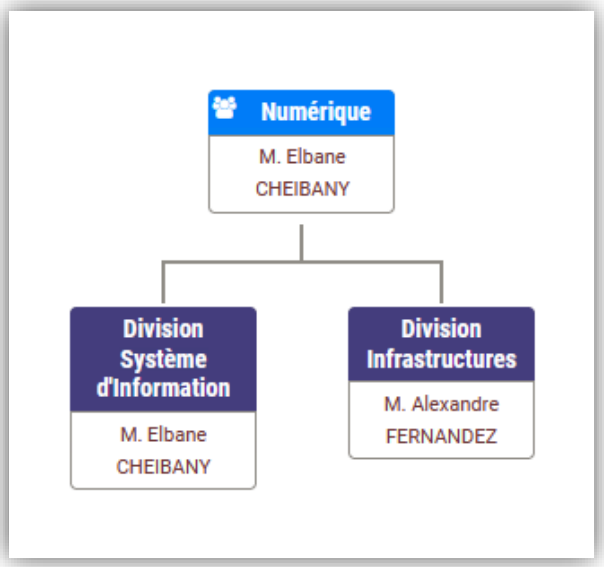
La division infrastructures

- gestion du réseau et matériel (postes de travail, serveurs)

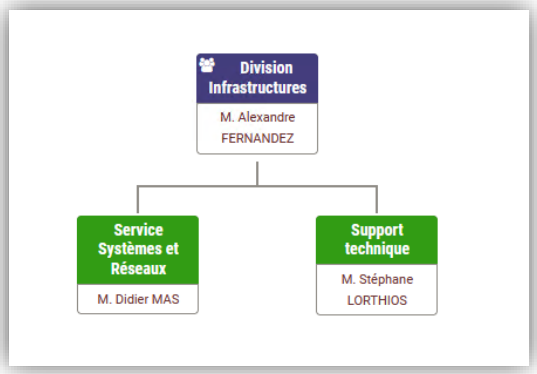
-assistance aux utilisateurs 1er et 2ème niveau

-gestion de la téléphonie

-gestion de l'exploitation



Mon premier interlocuteur et tuteur de stage a été Monsieur Didier Mas, administrateur réseau et responsable de la sous-division systèmes et réseaux.



Modalités de travail

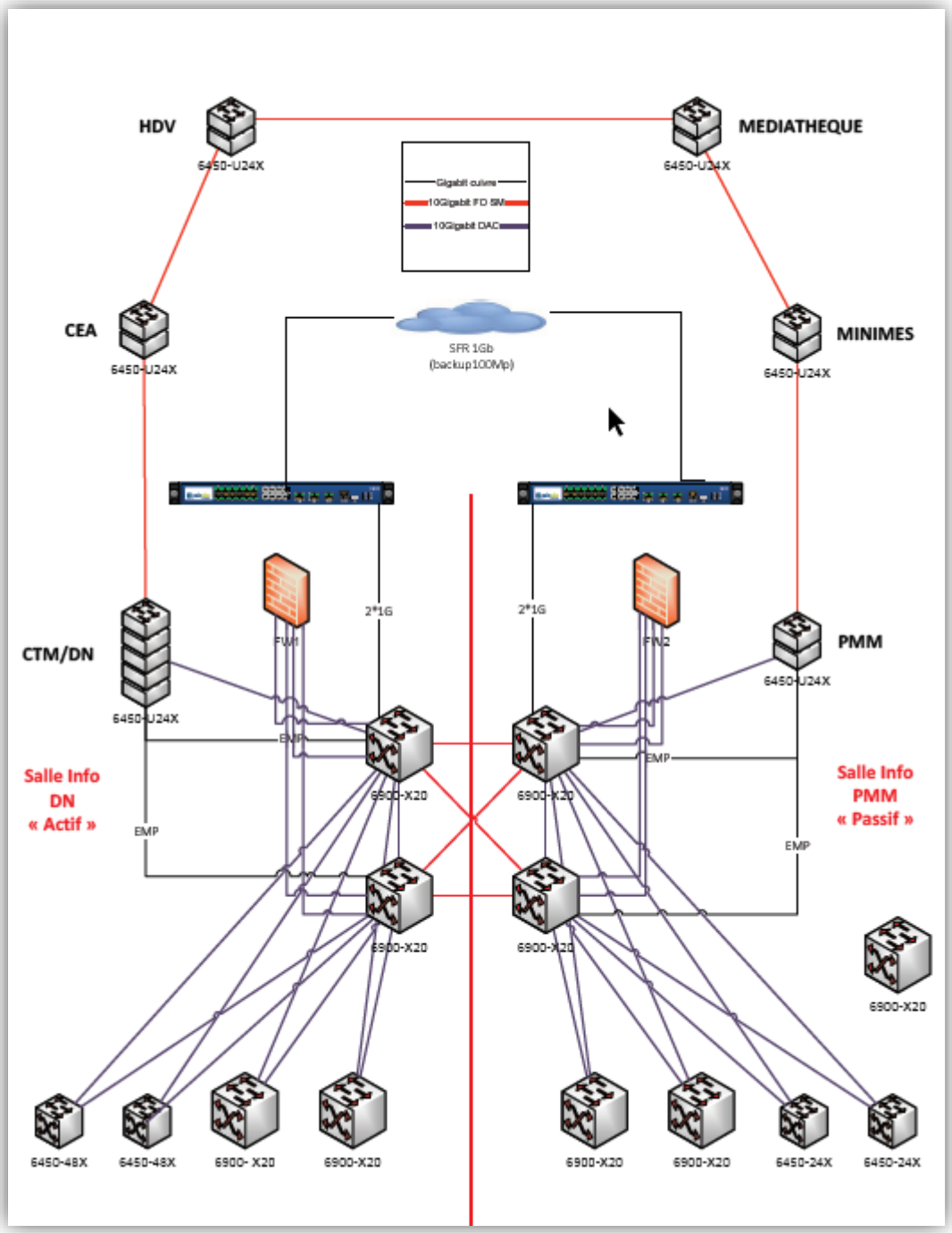
Compte tenu de la crise sanitaire, j'ai effectué un stage sous forme hybride : la majorité du travail s'est faite à distance, par connexion RDP sur le poste de travail qui m'avait été attribué et sur ma propre infrastructure privée, avec deux demi-journées de travail présentiel au centre technique.

J'ai eu à ma disposition trois machines virtuelles sous Ubuntu 20.4 et 3 machines virtuelles Windows Server 2016. Les serveurs Windows faisaient partie d'un domaine active Directory « test.lan » que j'ai créé à cet effet.

Lorsque je rencontrais des problèmes de téléchargement ou de compatibilité, j'effectuais les mêmes manipulations sur mon infrastructure personnelle que je décris dans la section suivante du rapport.

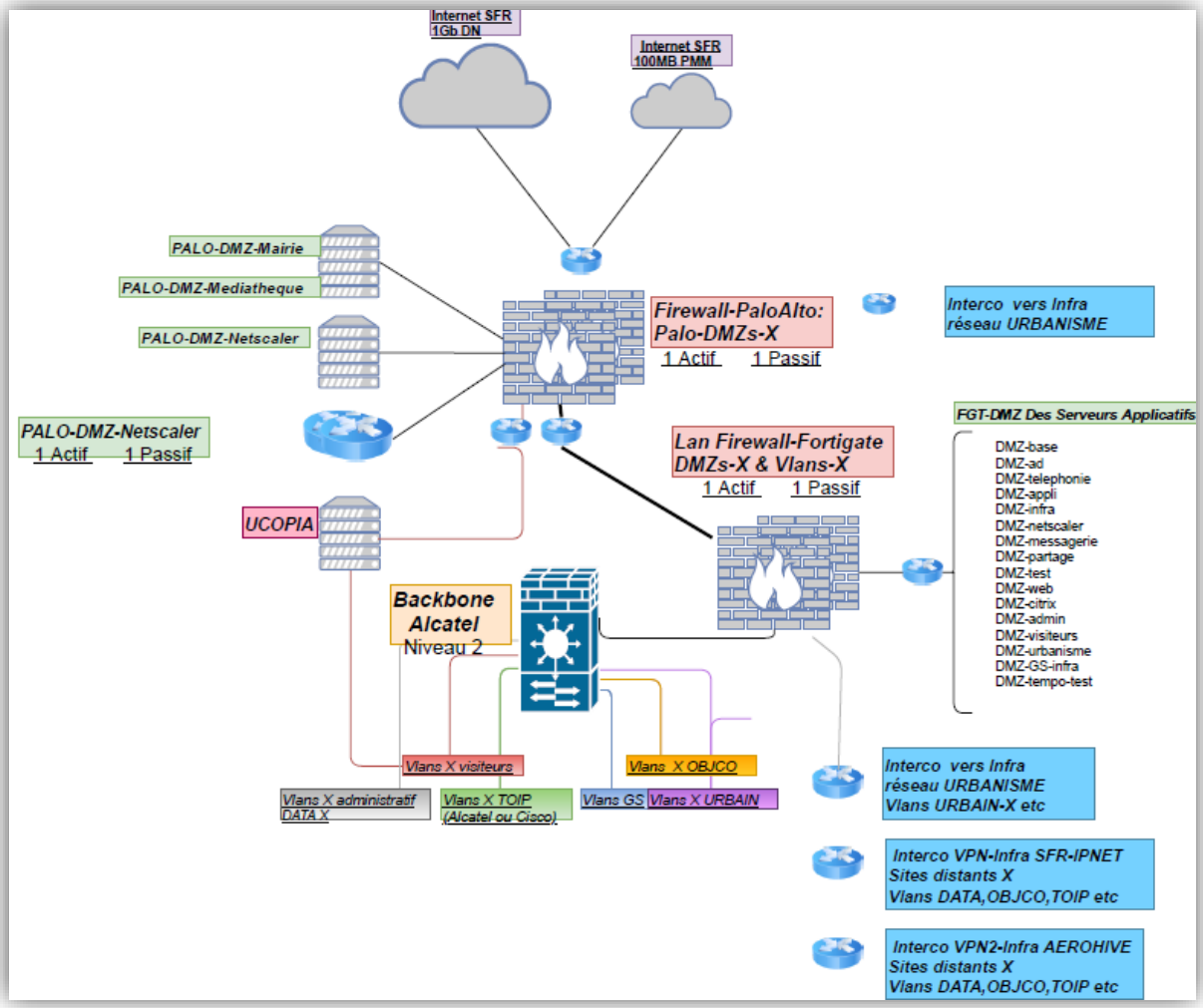
Infrastructure Informatique

Le service numérique dispose de deux salles informatiques principales sur deux lieux différents. Le réseau, fourni par SFR, est principalement basé sur la technologie Alcatel Enterprise. Le cœur de réseau est distribué entre quatre switchs physiques, raccordés entre eux en topologie maille par de la fibre optique. Les switchs se trouvent dans deux salles distinctes. Un mécanisme de tolérance de pannes permet que les équipements puissent fonctionner indépendamment et prendre le relais en cas de panne dans une des salles. Ces switchs sont raccordé à environ 15 switches « Top of Rack » et un réseau de distribution utilisant le protocole ERP. Cette « boucle métropolitaine » dessert un troisième niveau de switch « accès » auxquels sont raccordés les postes de travail, téléphones et imprimantes et

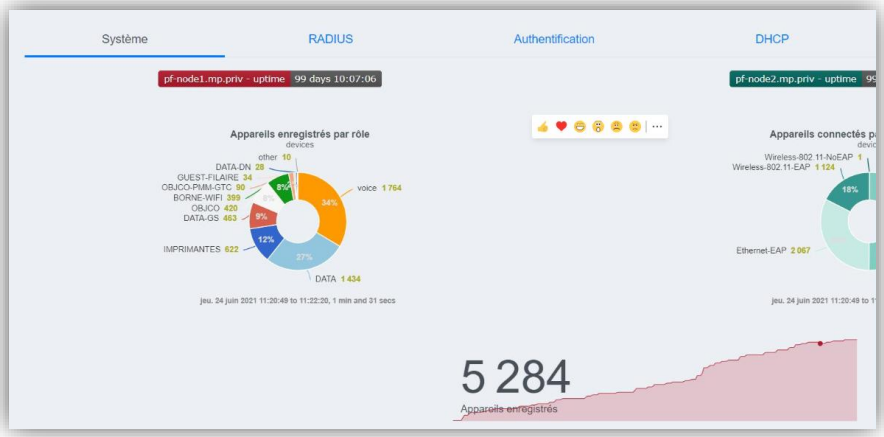


comprend tous les VLANs par des liaisons 10Gb. Aucun routage n'a lieu à l'intérieur du réseau ; il fonctionne par agrégation de VLAN.

Les switches « accès » sont principalement du matériel Cisco. Le cœur de réseau est protégé par des Pare-feu Fortinet qui gèrent le routage vers les réseaux externes. Du côté WAN, la sécurité est assurée par des firewall Palo Alto.

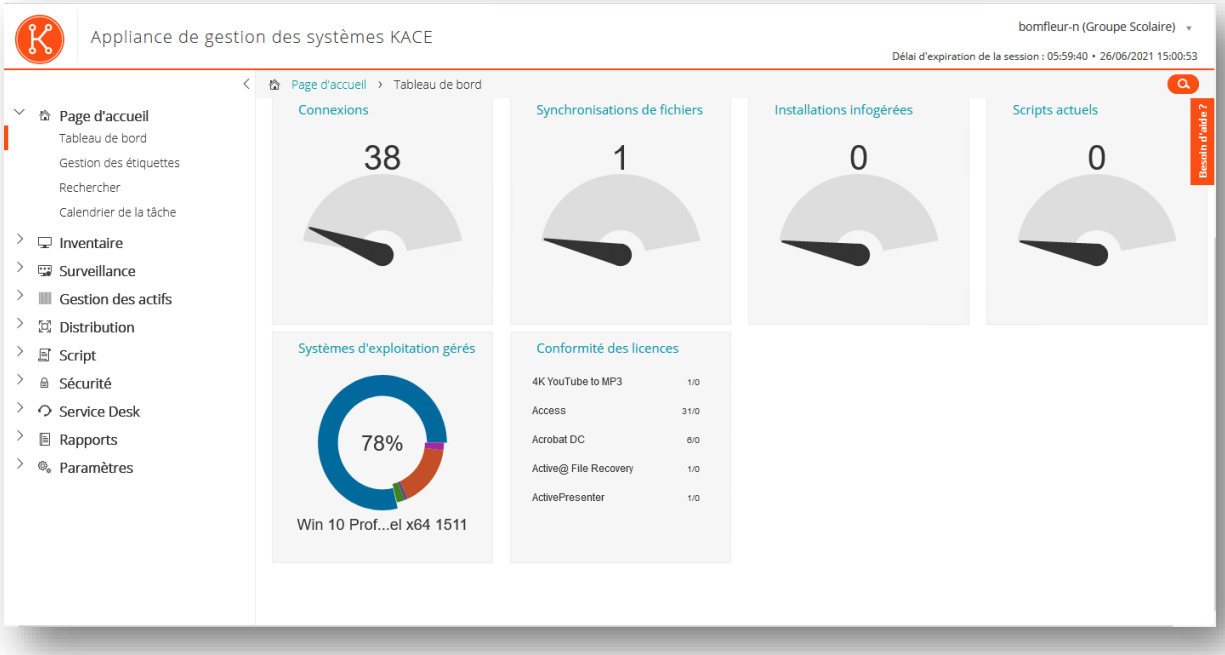


Plus de 5000 équipements pour environ 2500 utilisateurs sont connectés via RADIUS su les réseau géré par la direction du numérique.Le logiciel Packetfence permet une vue globale :



Le réseau de la direction numérique comprend 3 VPN qui permettent de séparer les accès pour les trois domaines Active Directory différents : Visiteurs, Groupes Scolaires, Mairie et bâtiments rattachés.

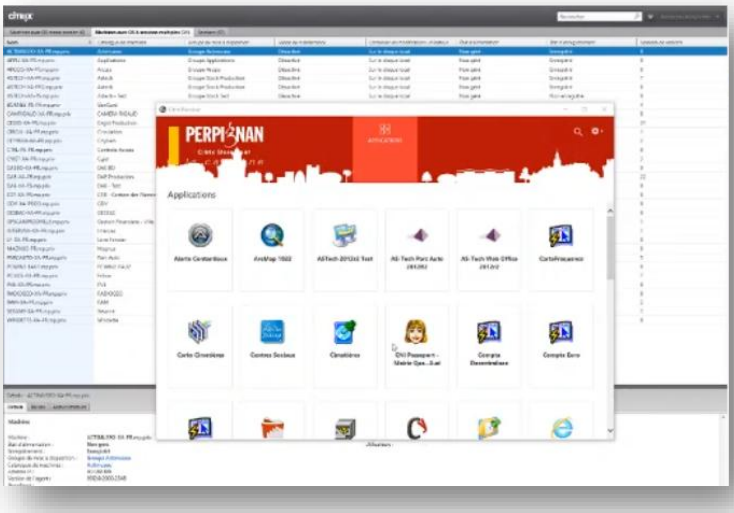
Pour gérer les postes d'utilisateurs, la direction numérique emploie le logiciel de gestion unifiée de terminaux KACE :



Cette application offre de nombreuses fonctionnalités

- Inventaire matériel et logiciel
- Déploiement d'images d'installation et de logiciels
- Configuration à distance
- Service Desk intégré
- Management de patchs
- Journalisation d'évènements

Une grande partie des applications sont mises à disposition des utilisateurs par Citrix XenApp



La plus grande partie des serveurs de la direction numérique est virtualisée sous VMWARE VSphere.

Les plus de 480 serveurs virtuels sous systèmes d'exploitation différents proposent des services divers tels que

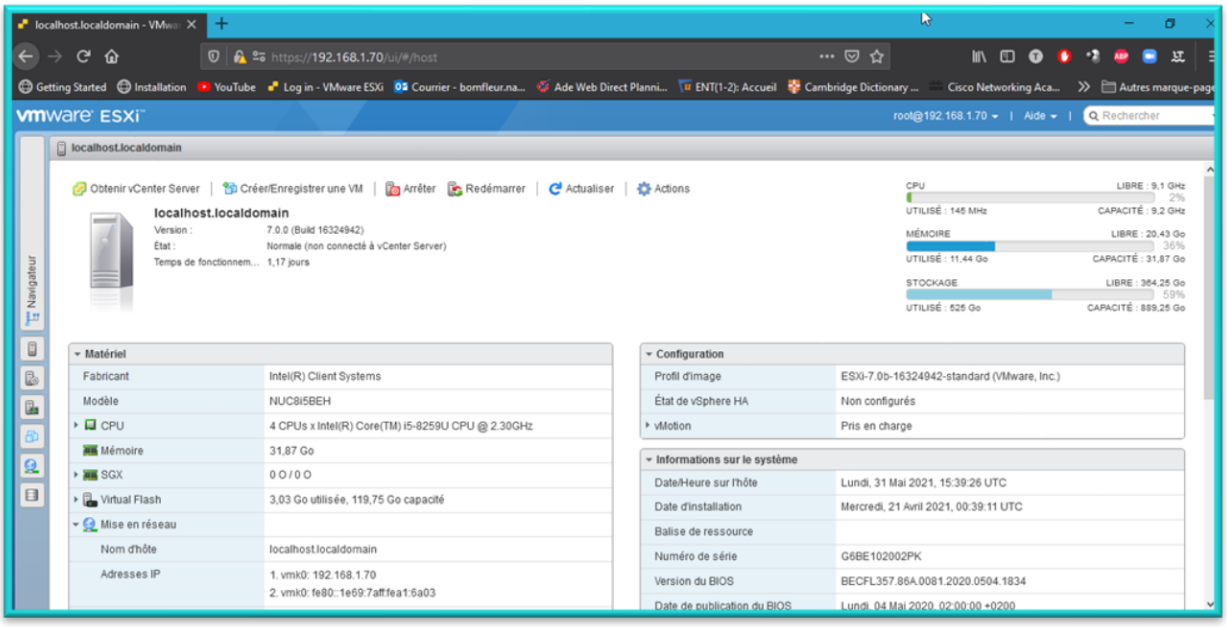
- Serveurs de fichiers et sauvegarde
- Serveurs d'applications
- Serveurs de déploiement
- Serveurs d'annuaire Active Directory
- Serveurs web

A l'heure où j'ai débuté mon stage, ces machines virtuelles étaient gérées individuellement.

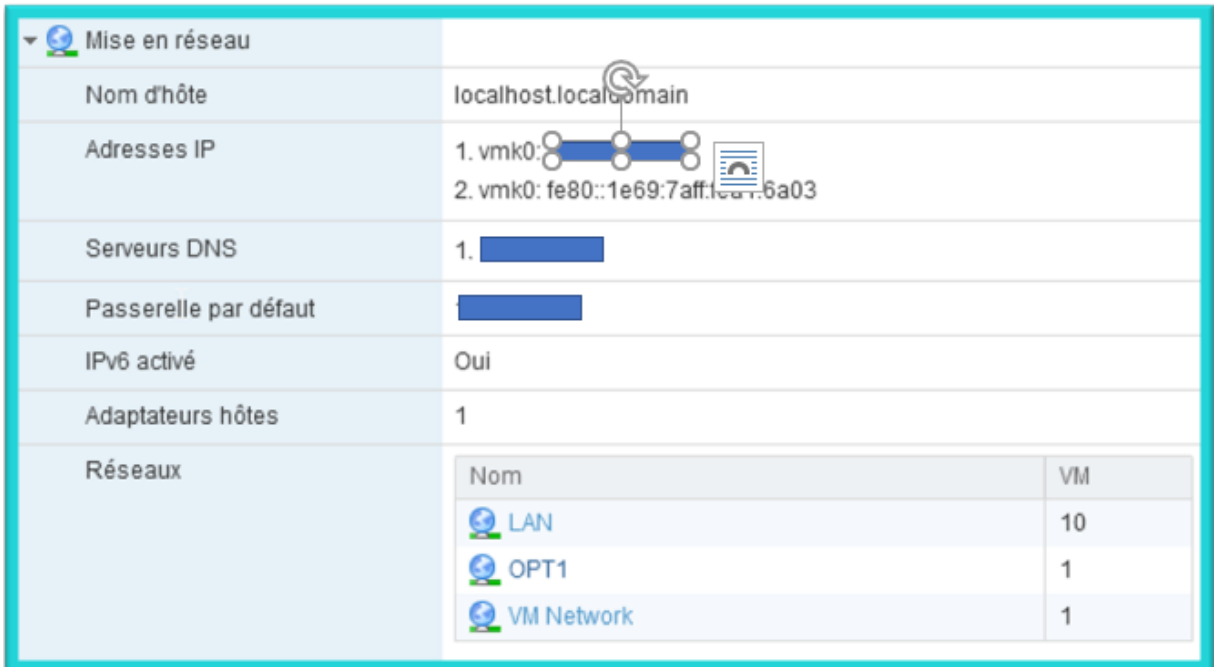
Cependant, mon tuteur envisageait la mise en place d'une solution unifiée de gestion telle que KACE. La recherche d'outils adaptés a fait partie intégrante de mon projet de stage.

Infrastructure privée

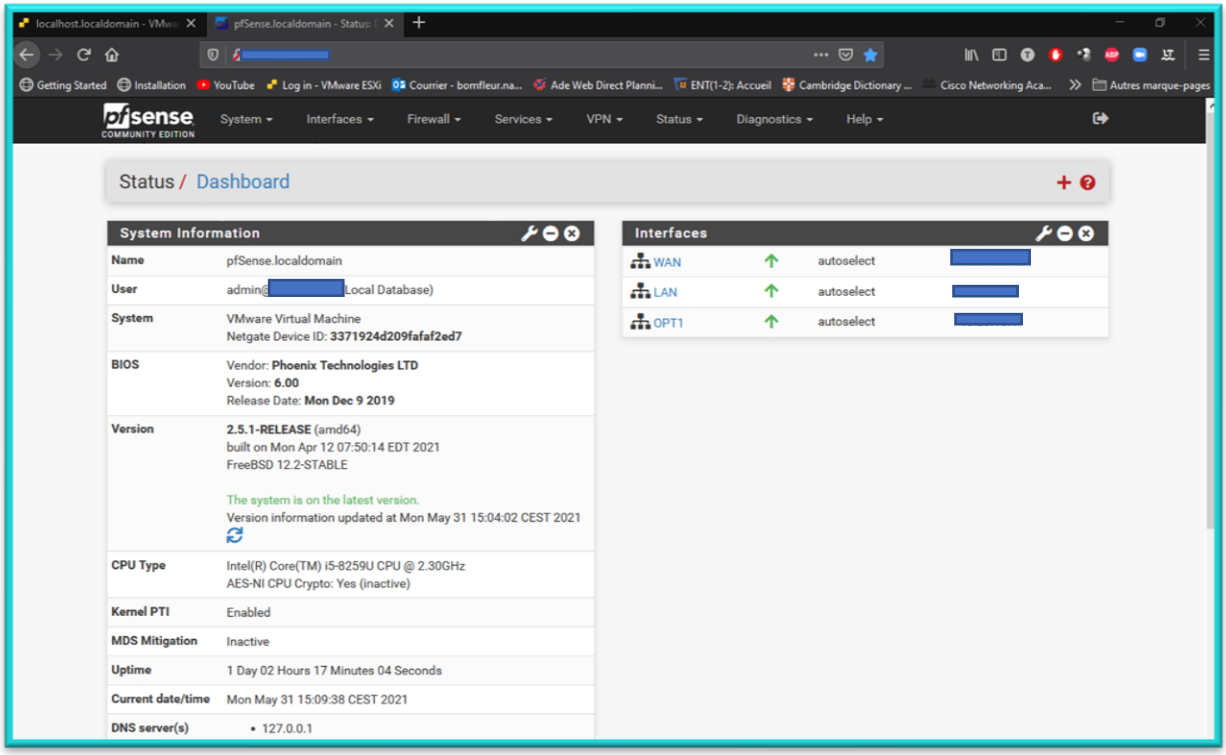
Afin de pouvoir effectuer des tests sur multiples machines simultanément sans déranger le bon fonctionnement de l'infrastructure de la mairie, j'ai créé ma propre infrastructure sur un serveur dédié à la virtualisation chez moi. J'ai installé VMWare VSphere 7.0 sur un mini-PC INTEL NUC 8i5BEH



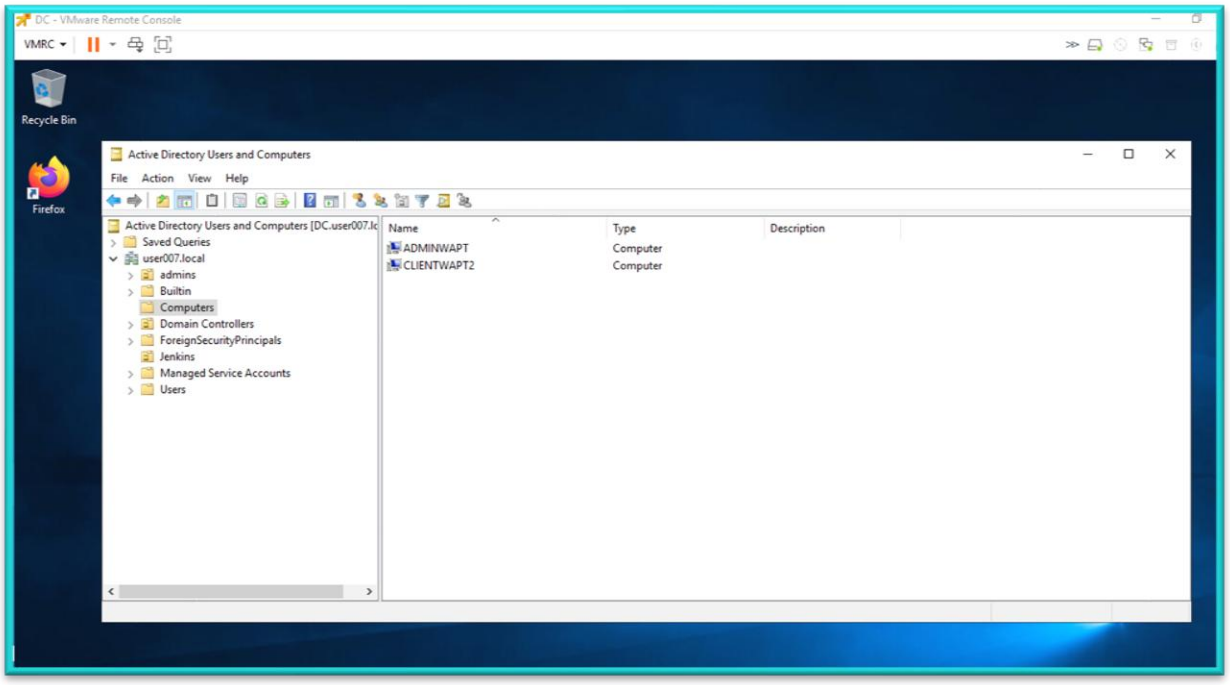
J'ai créé deux réseaux internes distincts (LAN et OPT1) sur le switch virtuel pour pouvoir séparer les trafics différents.

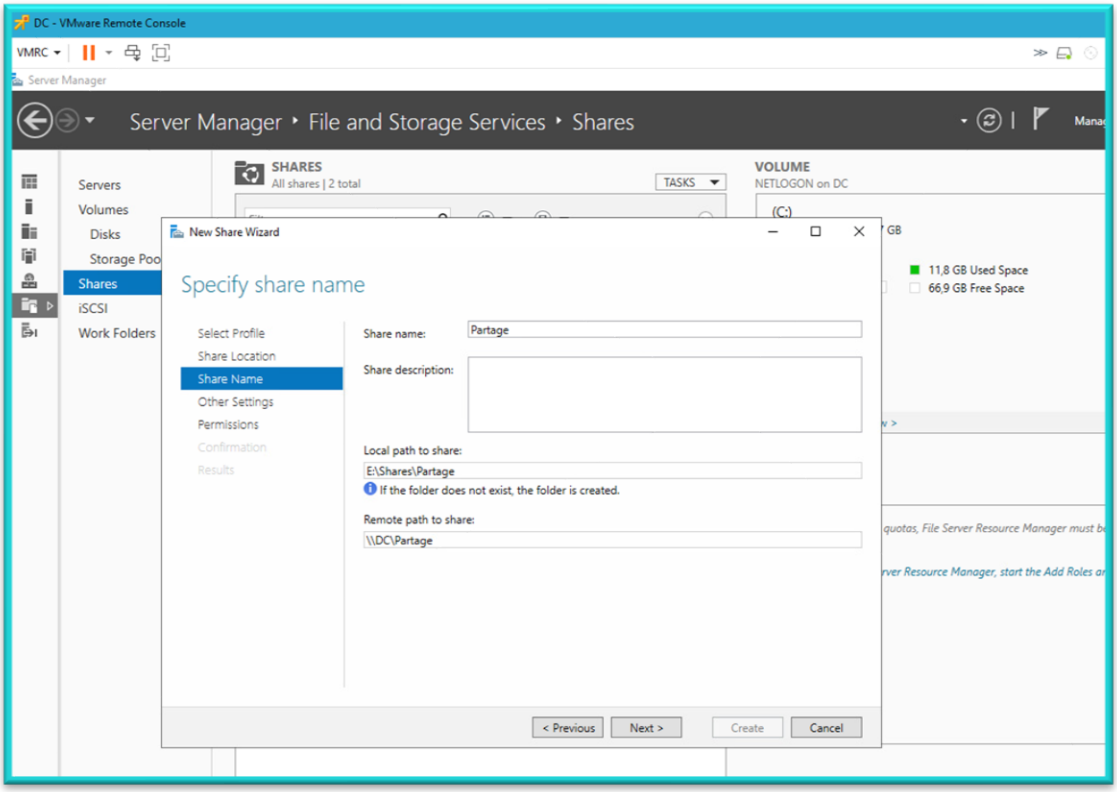


Un routeur PFSense, installé sur une VM, permet aux machines sur les réseaux internes d'accéder au réseau externe VMNetwork.



Pour simuler un environnement similaire à celui de la mairie, j'ai créé un domaine Active Directory « user007.local » avec un partage qui facilitait les échanges entre machines clients, par exemple dans les tests de Jenkins :





Afin de faciliter la distinction entre les captures d'écran qui concernent l'infrastructure de la mairie et celles qui concernent mon infrastructure privée, j'utiliserai deux couleurs différentes comme cadre des graphiques

Mairie :

Infrastructure privée :

Là où les manipulations sont exactement identiques, je n'utiliserai qu'une seule capture d'écran.

Le projet de stage

Le sujet prévu pour mon projet de stage était la mise en place d'un ordonnanceur de tâches et l'Infrastructure as Code. Quand je suis arrivée à la DISI, mon tuteur m'a demandé de tester les logiciels Cockpit et Jenkins, sans préciser le rapport direct avec mon projet. J'ai donc effectué les installations et fait des démonstrations des fonctionnalités, puis j'ai demandé à mon tuteur quelles configurations il souhaitait pouvoir effectuer précisément avec des solutions d'IaC ; c'est alors à ce moment-là que nous avons formulé des tâches concrètes.

La mairie de Perpignan dispose, comme de nombreuses entreprises, d'une infrastructure « mixte » : on y trouve des machines sous différents systèmes d'exploitation tels que des distributions Linux (dans le cas présent : Ubuntu et Debian), Windows et des hyperviseurs de type 1 comme VSphere. Le parc de serveurs comprend plus de 100 serveurs sous différentes distributions Linux et environ 370 machines sous Windows. Il est évident qu'effectuer une quelconque manipulation manuelle consomme énormément de temps, mais que les solutions de déploiement et configuration qui ne fonctionnent qu'avec un type de système d'exploitation ne seraient pas optimales non plus. Le but de ma recherche était donc de trouver des outils d'automatisation adaptés à Windows et Linux en même temps.

Les trois tâches principales que mon tuteur souhaitait voir automatiser étaient :

- Configurer les paramètres réseau, plus précisément, renseigner le serveur DNS et suffixes de recherche
- Vérifier l'installation de logiciels sur les postes clients
- Le cas échéant, lancer la mise à jour de logiciels sur les machines concernées

Après les installations initiales de Cockpit et Jenkins, j'ai fait des essais sur les deux logiciels pour voir en quelle mesure ils pouvaient satisfaire aux besoins formulés. J'ai ensuite évoqué WAPT, logiciel de déploiement que j'avais aperçu durant une activité précédente en tant que technicienne informatique dans un établissement scolaire. Ce logiciel me semblait intéressant pour vérifier l'état des logiciels installés sur les machines et donc pour répondre à deux des demandes formulées. J'ai ensuite proposé de tester un outil de management de configuration pour effectuer les manipulations concernant le réseau. Saltstack disposant de modules spécifiques pour la gestion des paramètres DNS, le choix s'est porté sur cet utilitaire.

Pour trouver un fil conducteur dans mes activités durant le stage en entreprise, je me suis interrogée sur les points communs des quatre logiciels avec lesquels j'ai travaillé : tous permettent d'avoir une vue d'ensemble sur des parcs informatiques et évitent de devoir passer d'une machine à l'autre manuellement ; mis à part Cockpit, les outils logiciels permettent de gérer des machines sous différents systèmes d'exploitation ; et tous permettent d'accélérer des paramétrages (installation de logiciel, configuration réseau, création d'utilisateurs) en centralisant leur gestion. J'ai donc décidé d'intituler mon projet

« Solutions de centralisation et d'automatisation pour parcs informatiques hétérogènes ».

Ce projet ne se veut en aucun cas un comparatif exhaustif des outils existants mais plutôt un retour sur expérience ; il m'a permis d'avoir un aperçu du domaine intéressant de la gestion de configuration que je continuerai certainement à étudier.

Cockpit

Cockpit est un outil d'administration à distance gratuit créé par RedHat, inclus dans les distributions Fedora 21 et RedHat Enterprise à partir de la version 7. Rappelant l'utilitaire Webmin, il offre également une interface graphique qui rend l'utilisation agréable pour les néophytes de Linux qui n'ont pas l'habitude de la ligne de commande. Avec une architecture modulaire, il est possible d'ajouter des fonctionnalités et de développer ses propres modules. Cockpit utilise un socket systemd et les mêmes API que les outils de ligne de commande et fonctionne donc sans surcouche logicielle.

Bien que les éditeurs précisent que Cockpit n'est pas un outil de management de configuration et d'IaC, la possibilité de pouvoir gérer plusieurs machines Linux par la même interface et de pouvoir surveiller les mises à jour entre dans la démarche sous-jacente de mon projet de stage : éviter les manipulations éparpillées et pouvoir centraliser les manipulations sur plusieurs machines au lieu de les répéter d'innombrables fois sur un parc informatique de taille importante.

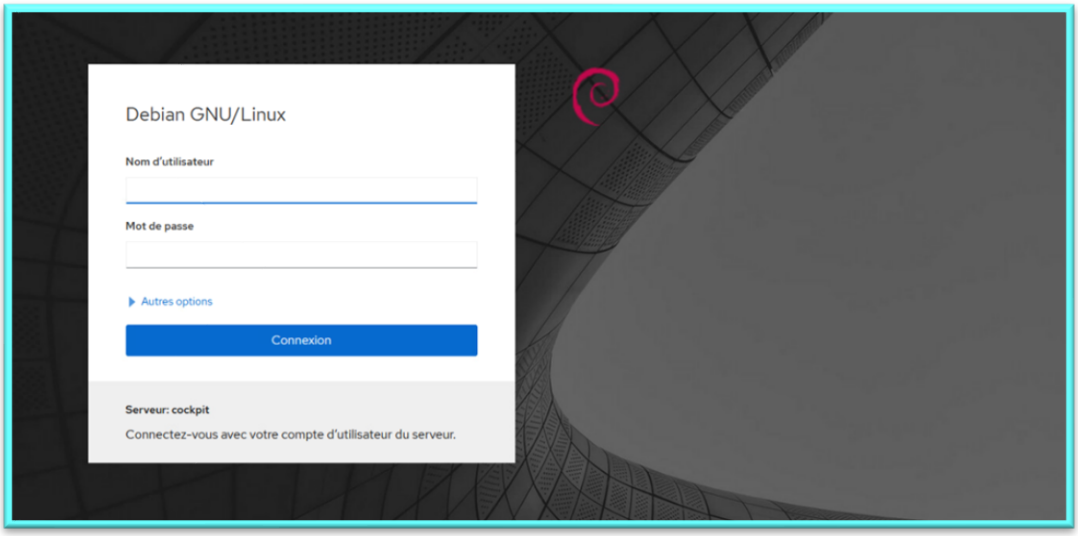
Installation

Cockpit s'installe facilement sous Ubuntu et Debian par la commande suivante

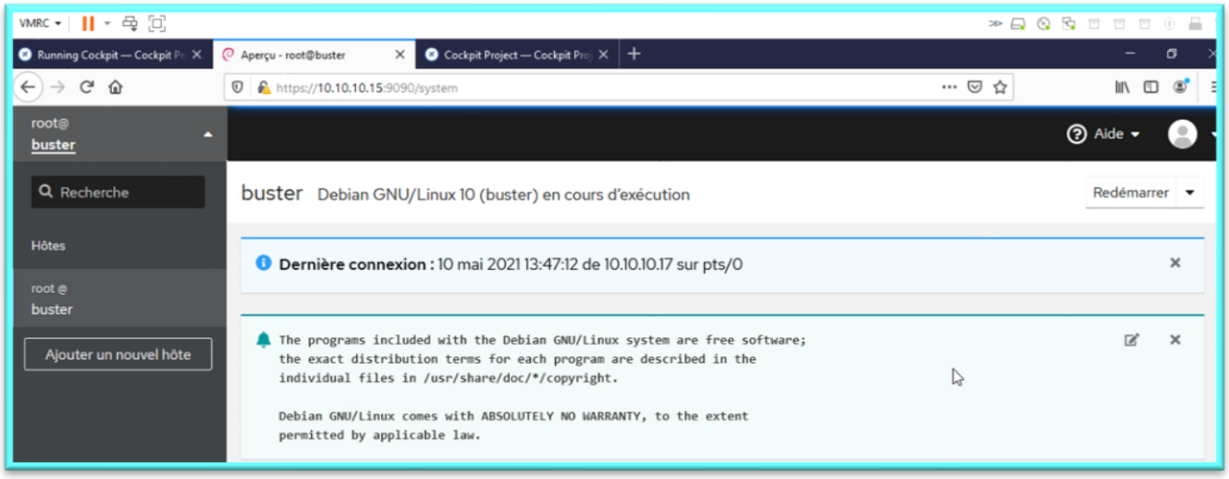
```
root@ubunsalt:/home/user# apt install cockpit
```

Le logiciel doit être installé sur toutes les machines qu'on souhaite gérer avec, qu'il s'agisse du serveur principal ou des nœuds connectés.

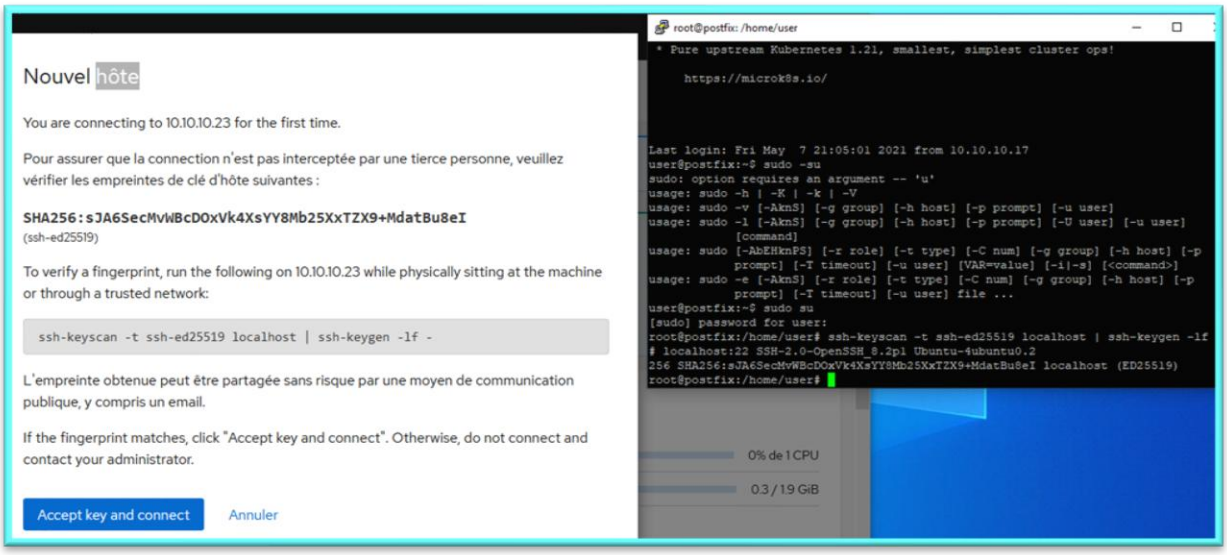
Après l'installation, on accède au système grâce à un navigateur et l'IP ou le nom d'hôte de la machine, sur le port 9090 :



Le tableau de bord permet ensuite d'ajouter d'autres machines à l'interface :

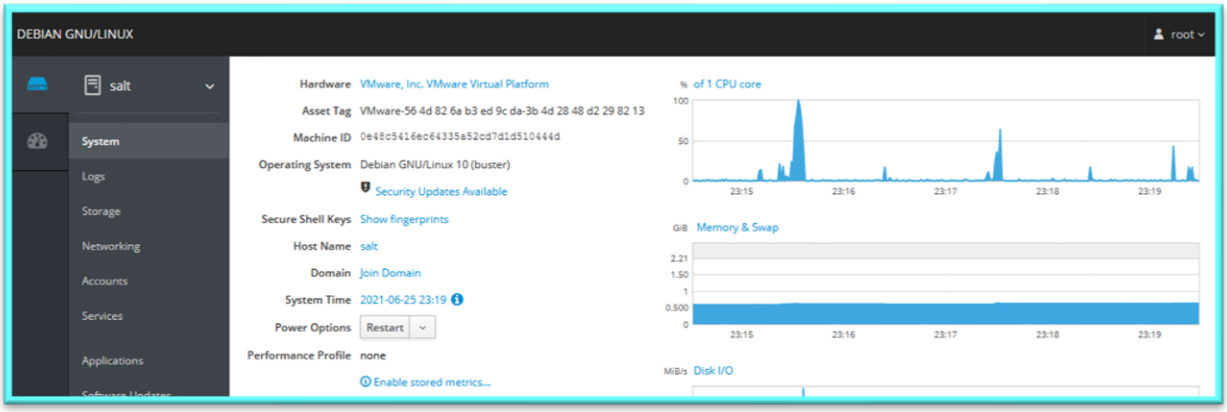


Avant la première connexion du nouvel hôte, le système interroge sur la véracité de la clé SSH et montre la commande ssh-keyscan pour vérifier la clé sur la machine distante. Une fois la clé acceptée, l'hôte ajouté apparaît dans le tableau de bord

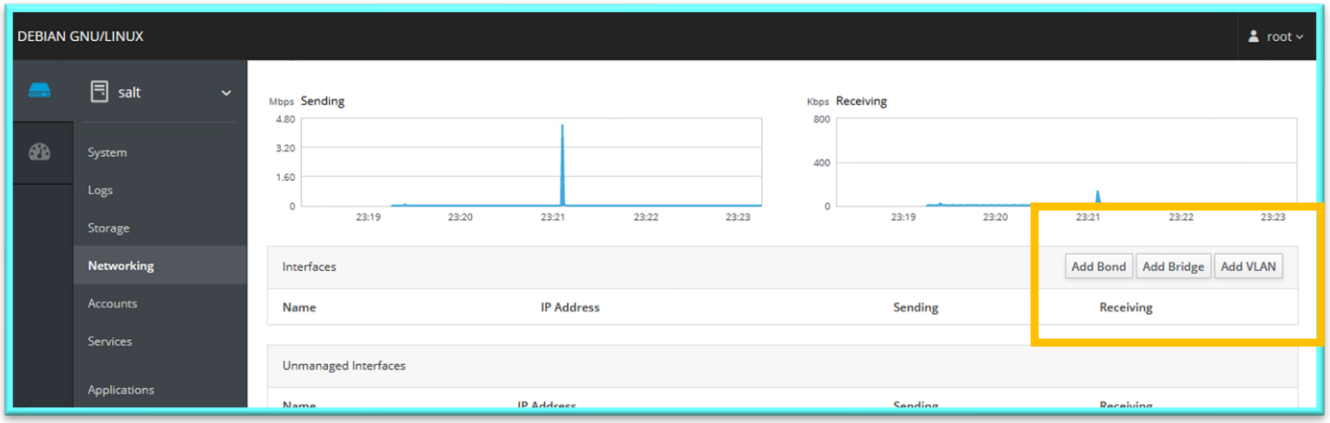


Le tableau de bord permet alors d'avoir une vue globale sur les ordinateurs connectés :

Il affiche des informations sur la santé générale du système, mais aussi sur les interfaces réseau et le stockage,

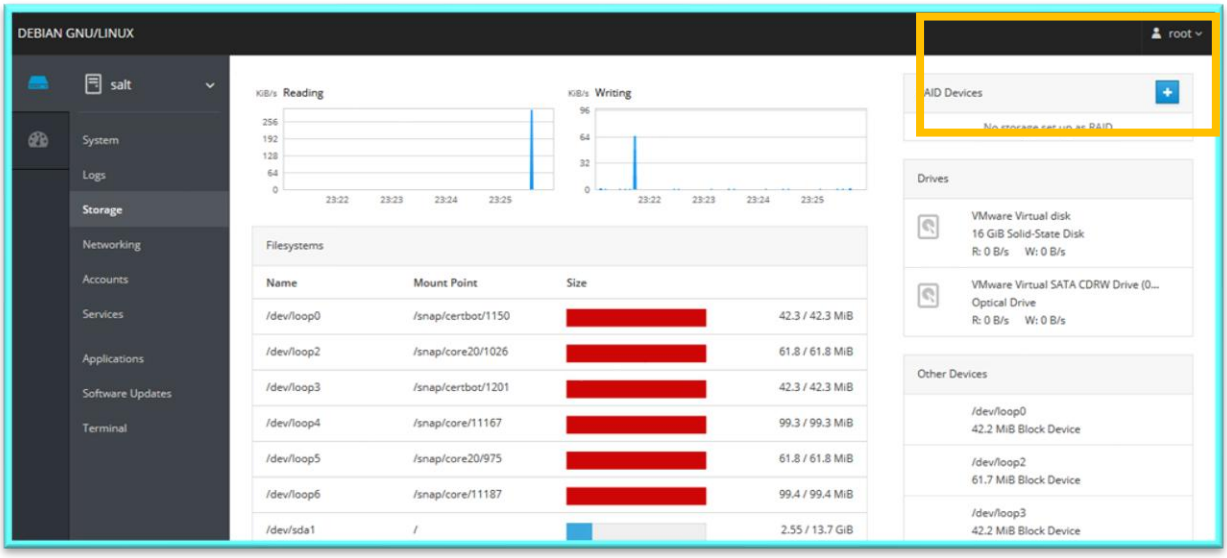


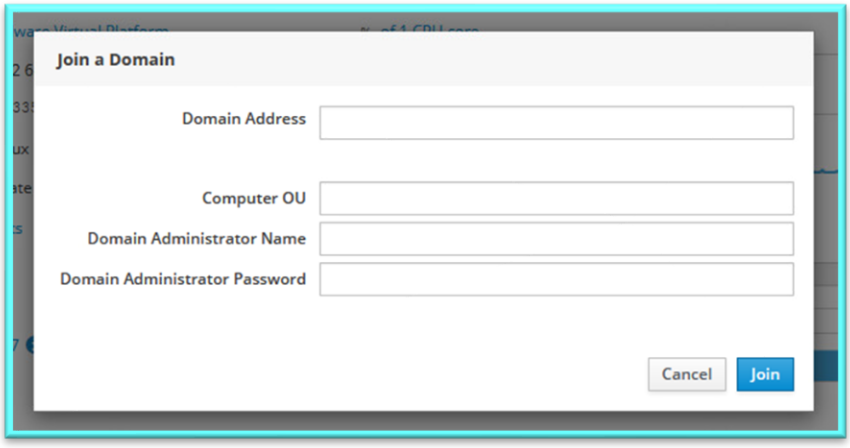
L'onglet Réseau permet d'agréger des interfaces réseau et de créer des ponts et gérer des VLAN



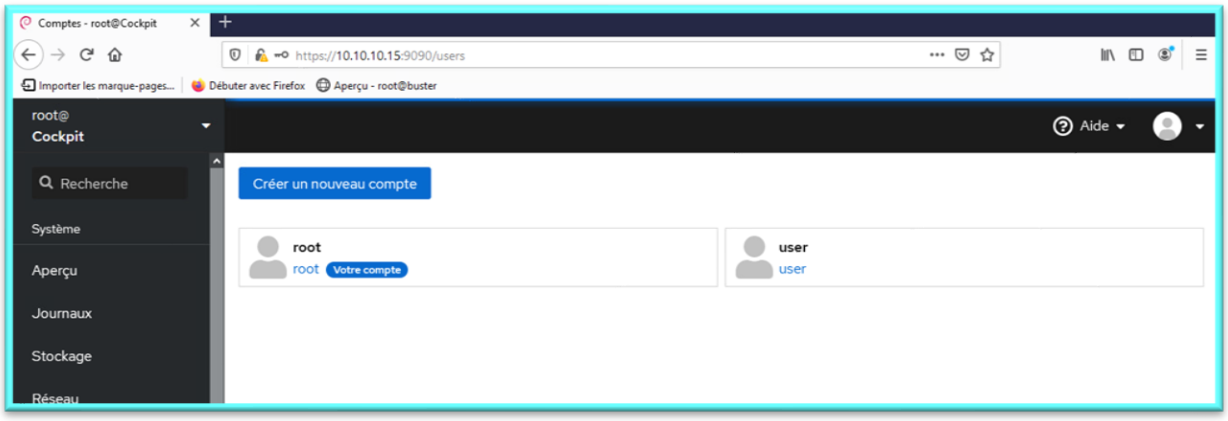
Dans l'onglet Stockage, il est possible de rajouter des RAID

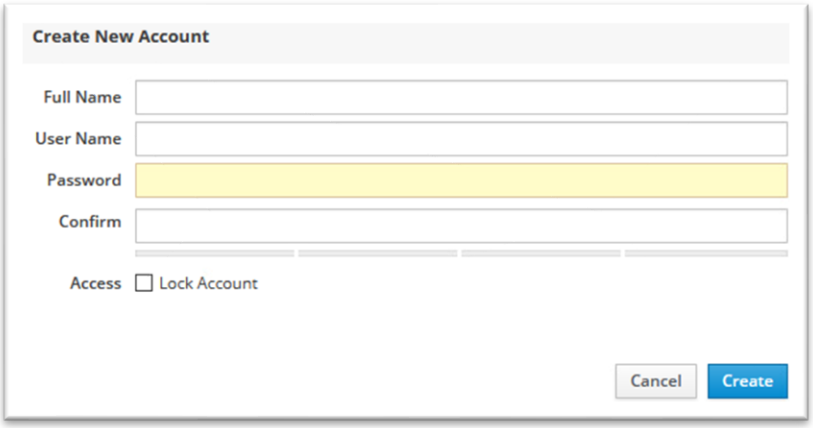
La fonction « Join Domain » dans l'onglet système installe realmd et facilite l'intégration dans un domaine



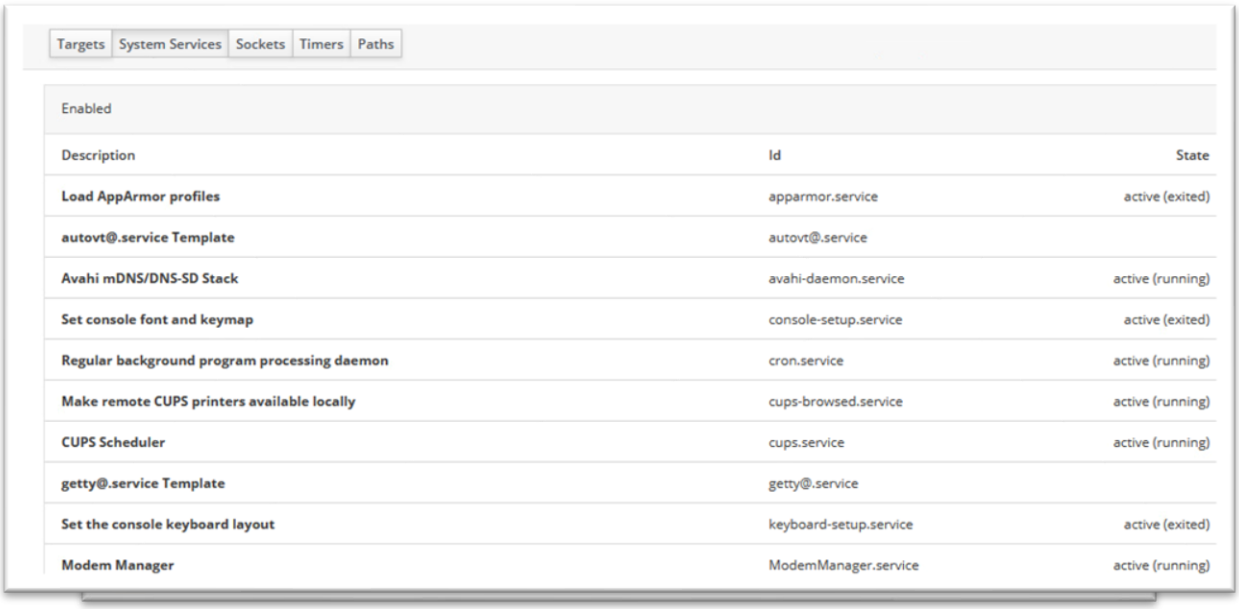


Cockpit facilite la création de nouveaux utilisateurs





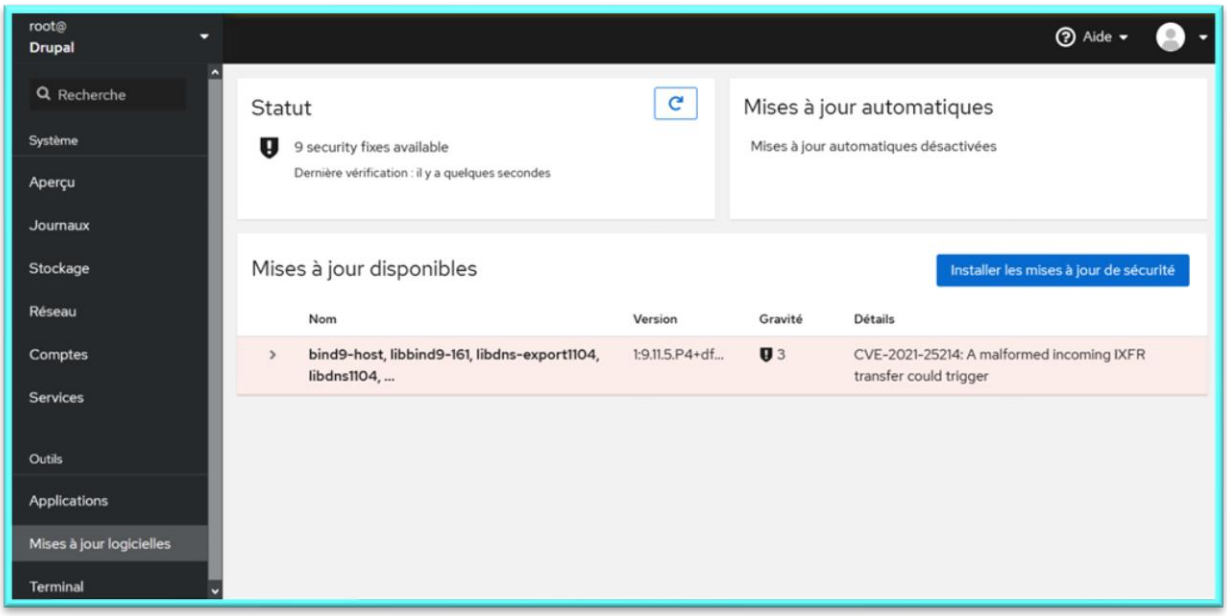
Le listing des services offre un aperçu plus clair pour les utilisateurs qui n'ont pas l'habitude de la ligne de commande



La fonction la plus intéressante dans le cadre de mon stage était la gestion des mises à jour logicielles.

Cockpit vérifie automatiquement la disponibilité de nouvelles versions logicielles. La mise à jour rapide sur plusieurs ordinateurs distants grâce à une interface unique permet d'économiser beaucoup de temps.

En tout, Cockpit est un outil intéressant pour la centralisation de gestion de machines Linux abordée au début du stage. J'ai d'ailleurs continué à l'utiliser pour vérifier l'état des machines Linux (Debian et Ubuntu) sur lesquelles je testais les autres solutions abordées dans ce rapport.



Concepts théoriques liés à l'automatisation

Infrastructure as Code

L'infrastructure as Code (IaC) est une méthodologie de travail qui repose sur le principe de gérer la configuration d'infrastructures informatiques (paramètres réseau, installations logicielles, création de machines virtuelles) par l'automatisation logicielle plutôt que par des processus manuels. On parle aussi d'infrastructure programmable ou « software-defined ». L'IaC est utilisée dans tous les domaines informatiques mais principalement dans le cloud computing et le développement de logiciels. Contrairement à la gestion par scripts qui cible certaines étapes de configuration de manière statique et qui implique la répétition de ces étapes, les outils d'IaC disposent d'un langage de niveau élevé qui permet de décrire des processus de configuration complexes. Saltstack, par exemple, peut installer et configurer un environnement de serveurs web Apache, ainsi que paramétrer des pages web.

On peut distinguer parmi les solutions

d'IaC les outils d'orchestration et les outils de management de configuration. Les premiers servent à approvisionner l'infrastructure en instances telles que serveurs et conteneurs, gérer l'espace de stockage et des clusters. On trouve ici des solutions comme Terraform, Azure Resource Manager et AWS. Les outils de management de configuration opèrent au niveau des instances et gèrent principalement l'installation de logiciels, scripts, démarrage de services etc. Parmi les logiciels les plus connus de « configuration management » (CM), on compte Ansible, Chef, Puppet et Saltstack.

Les solutions IaC diffèrent entre elles à plusieurs niveaux :

Ainsi, comme en programmation, on retrouve les approches déclaratives et impératives. La méthode déclarative décrit l'état dans lequel on souhaite retrouver l'infrastructure sans préciser les étapes nécessaires pour l'atteindre. On retrouve ce paradigme dans les langages de programmation : SLQ et Erlang, par exemple, sont des langages déclaratifs. AWS Cloud Formation suit cette approche. La méthode impérative, quant à elle, définit clairement la procédure à suivre pour atteindre l'état souhaité et s'apparente donc aux langages déclaratifs comme C++ et Python. Chef et Puppet peuvent être utilisés de façon déclarative et impérative.

On peut aussi distinguer les mécanismes par lesquels les modifications sur l'infrastructure sont effectués. Dans la méthode « push », une machine maître « pousse » la configuration souhaitée vers les systèmes concernés. Dans la méthode « pull », les systèmes lancent des requêtes de configuration vers le maître. La plupart des outils permettent les deux.

L'Infrastructure as Code offre de nombreux avantages : de nombreuses tâches répétitives et rébarbatives sont gérées automatiquement, ce qui permet aux administrateurs de système et développeurs d'utiliser ce temps pour la recherche et création de solutions plus importantes. Ceci peut aussi éviter l'erreur humaine, la négligence et les oublis. Cependant, la gestion automatique peut également avoir l'effet inverse : si les paramétrages initiaux ne sont pas planifiés et programmés avec soin, toute erreur de configuration est multipliée et répandue quasi immédiatement sur toutes les instances concernées. Il est donc primordial de bien maîtriser ces outils. Ceci, par contre, peut se révéler un vrai défi : Les outils d'IaC sont des systèmes sophistiqués, disposant de leur propre terminologie et « univers ». Il est

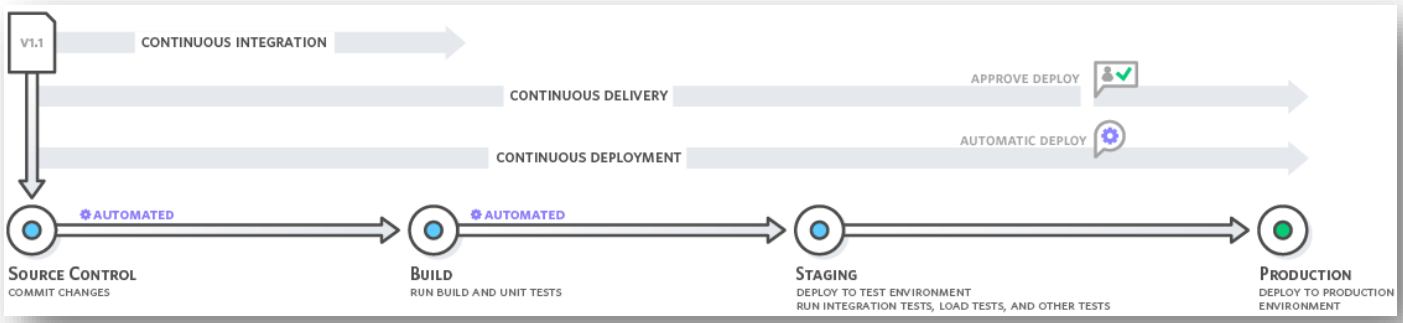
donc nécessaire d'investir assez de temps dans l'apprentissage et la formation, facteur de « coût » qui peut amener certaines entreprises à éviter de se lancer dans un projet d'IaC.

Continous Integration/Continous Delivery

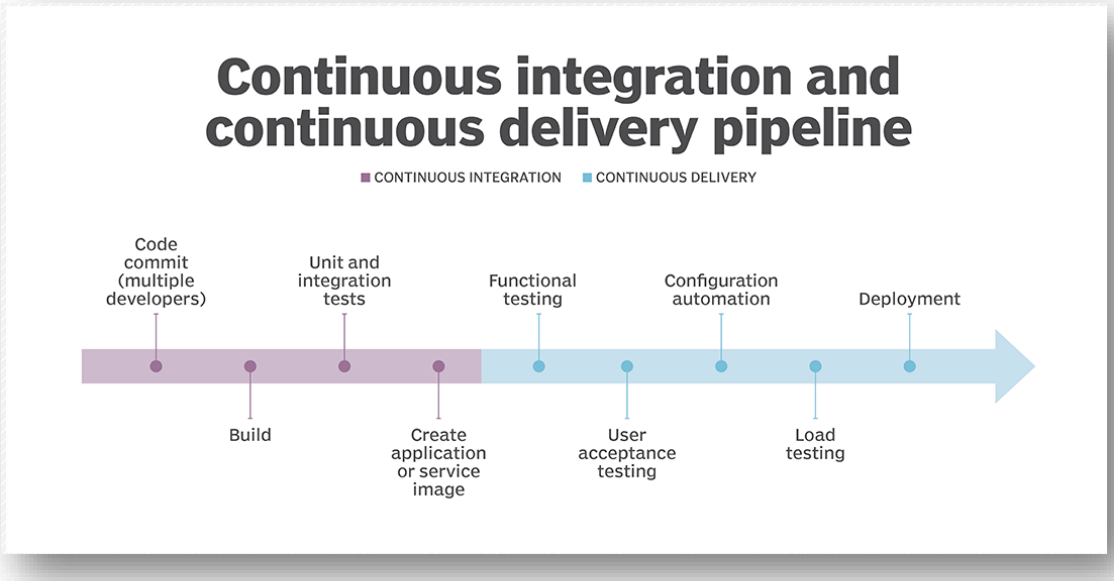
L'intégration continue est une méthode de travail utilisée en génie logiciel. Habituellement, lors du développement d'une application , plusieurs développeurs travaillent simultanément sur des éléments distincts (micro service) de l'ensemble du code. Dans l'intégration continue, ces morceaux de code sont régulièrement intégrés dans un référentiel centralisé (outil de gestion de code source tel que Git et autres), ce qui déclenche des opérations de tests et de création.

La livraison continue se base sur les mêmes principes que l'intégration continue mais concerne les étapes de déploiement du code modifié sur des environnements de test et production. On utilise ici des outils tels que Sélénium (Framework de tests) et UFT

Afin d'éviter les erreurs et surtout réaliser des économies de temps, la méthodologie CI/CD a recours à l'automatisation durant toutes les étapes de travail.



source: https://searchsoftwarequality.techtarget.com/CI-CD-pipelines-explained-Everything-you-need-to-know

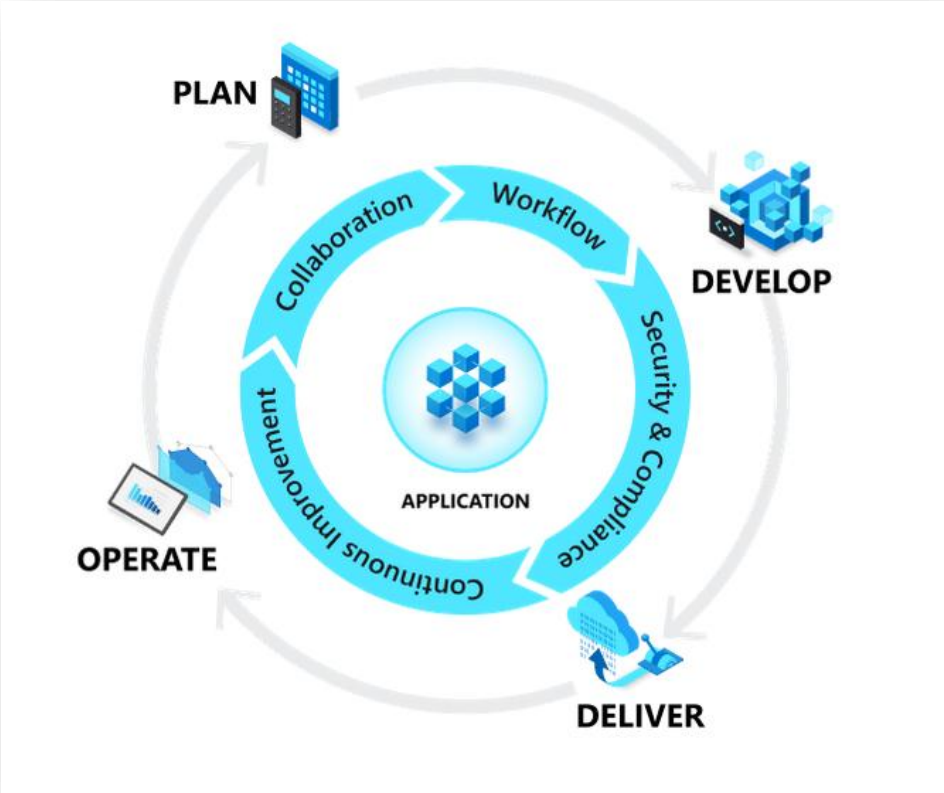


source: https://searchsoftwarequality.techtarget.com/CI-CD-pipelines-explained-Everything-you-need-to-know

DevOps

Tout comme l'intégration continue, la méthode DevOps vise la collaboration étroite et communication continue entre équipes pour unifier « DEVelopment » et « Operations » (exploitation et administration de systèmes). La vue globale sur les projets et produits et le partage de connaissances et de tâches sont des éléments clés de ce concept. Il s'agit plus d'une philosophie de travail que de pratiques précises. La méthode CI/CD est un outil de travail important de la culture DevOps.

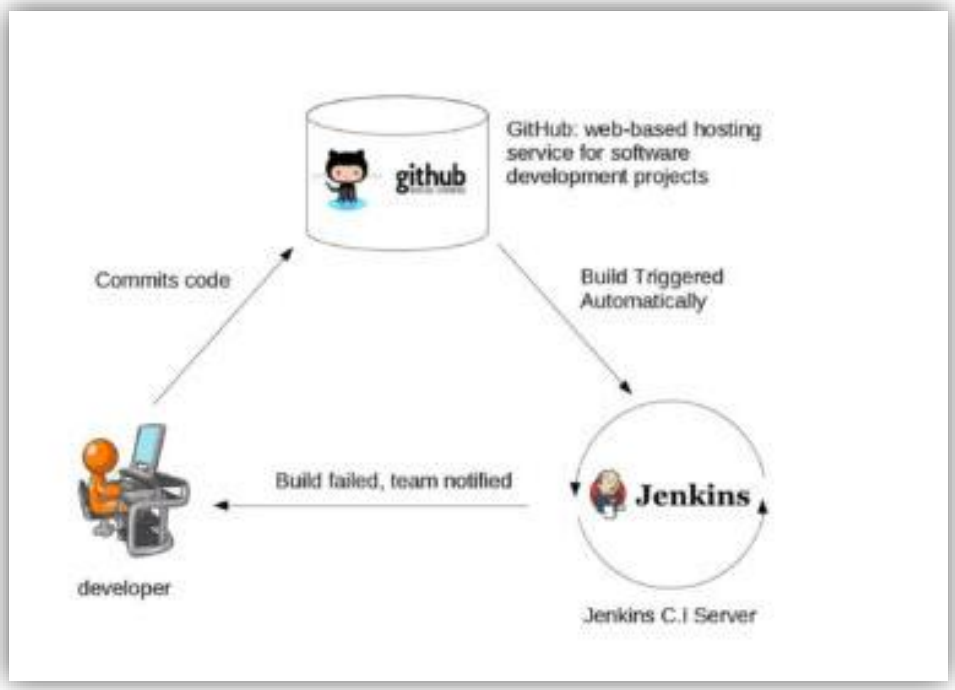
Afin de satisfaire à ses exigences de travail simultané et d'échange permanent, le DevOps a besoin de moyens de communiquer et d'agir de façon rapide pour pouvoir adapter les systèmes aux changements permanents qui accompagnent la création de nouveaux produits en équipe. L'automatisation est un des moyens techniques qui permettent d'éviter les pertes de temps liées aux manipulations répétées. Cet avantage peut bénéficier à n'importe quel domaine de l'informatique, notamment l'administration de système dans de grandes infrastructures. Les solutions adaptées au travail en DevOps et Continuous Integration ont en commun de remplacer les configurations manuelles par du code, et entrent donc dans la catégorie Infrastructure as Code.



source: https://azure.microsoft.com/fr-ca/overview/what-is-devops/

Jenkins

Jenkins est un outil open source d'IC (intégration continue) développé en Java. L'intégration continue est une pratique de travail permettant à plusieurs développeurs de travailler simultanément sur un code source dans un dossier partagé et apporter des changements fréquemment. Au lieu de rassembler tous les fragments de code à la fin du projet, les équipes de programmation transmettent régulièrement leur partie du code à l'application. Les modifications sont ensuite testées, ce qui permet de détecter d'éventuels problèmes. Jenkins fonctionne comme un serveur web suivant un modèle maître/esclave (instance légère de Jenkins) : Lors du développement d'un logiciel, le ou les développeurs transmettent leur code dans le répertoire de code source, qui peut être, par exemple, un dépôt GitHub ou Subversion. Les modifications sont détectées par Jenkins, qui lance alors la compilation et son analyse. Si des erreurs sont détectées, des notifications sont émises. Dès que le programme entier passe tous les tests fonctionnels avec succès, Jenkins peut pousser le code sur les environnements test prévus.



Bien que Jenkins soit un outil d'Intégration Continue destiné au programmeurs, ses fonctionnalités d'automatisation peuvent se révéler utiles dans le cadre de l'administration de système « de tous les jours ». Il est, par exemple, possible de l'utiliser comme ordonnanceur de tâches pour l'exécution régulière ou ponctuelle de scripts sur plusieurs machines.

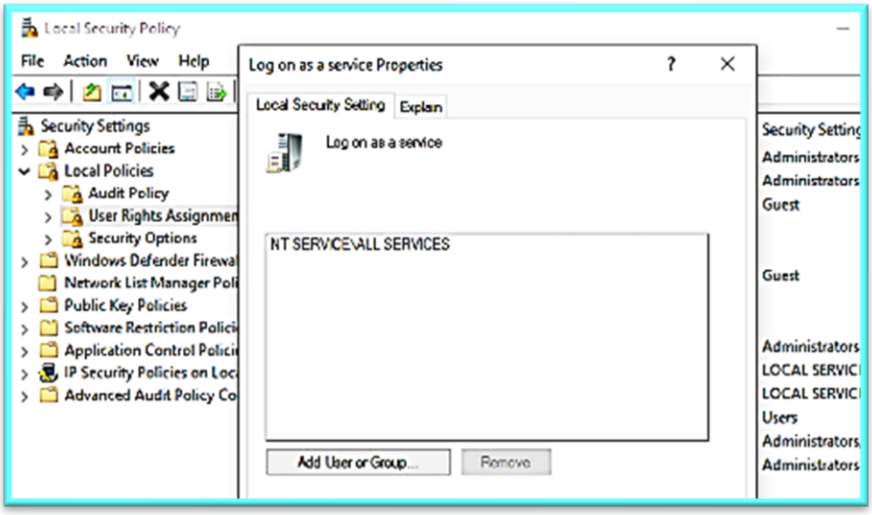
Environnement de test

Mairie : 1 Master (Windows server 2016), 3 nodes (2 Windows Server 2016, 1 Ubuntu 20.4)

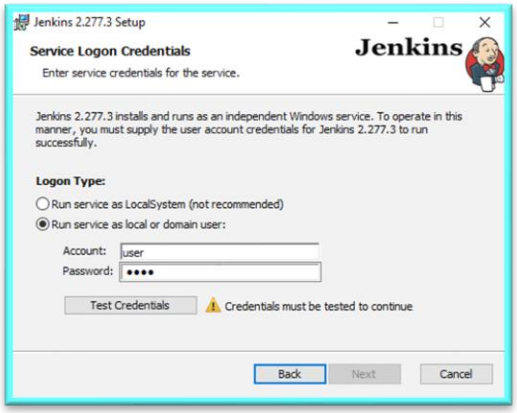
Privé:1 Master (Windows 10), 3 nodes (Windows 10, Ubuntu 20.4, Debian 10

Installation

L'installation sous Windows est très simple ; cependant, si on ne souhaite pas que Jenkins soit exécuté par le système mais par un utilisateur, il faut prévoir un compte utilisateur disposant du droit d'ouvrir une session en tant que service. Ce paramètre doit donc être modifié dans les stratégies de sécurité locales :

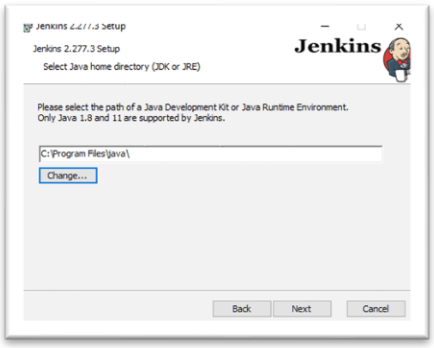


Jenkins teste ce paramètre à l'installation ; alternativement, le logiciel peut être exécuté en tant que LocalSystem



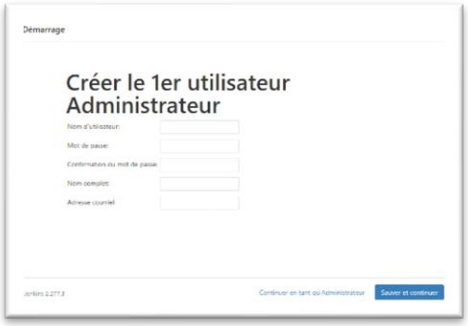
Ainsi que l'ouverture du port 8080 :

Jenkins nécessite l'installation de Java :

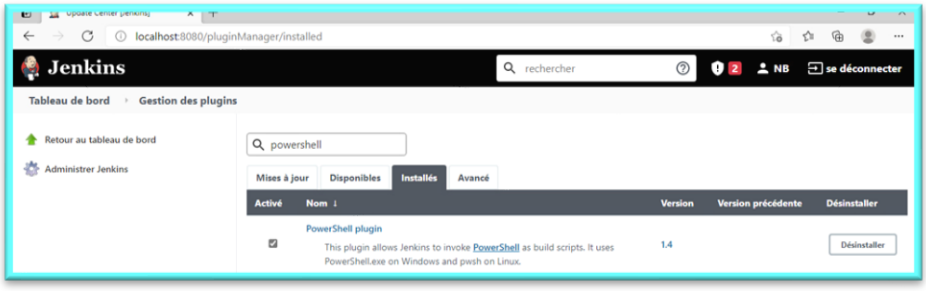


Une fois l'installation terminée, il faut configurer le gestionnaire web grâce au mot de passe initial

Bien sûr, il est recommandé de créer au moins un utilisateur



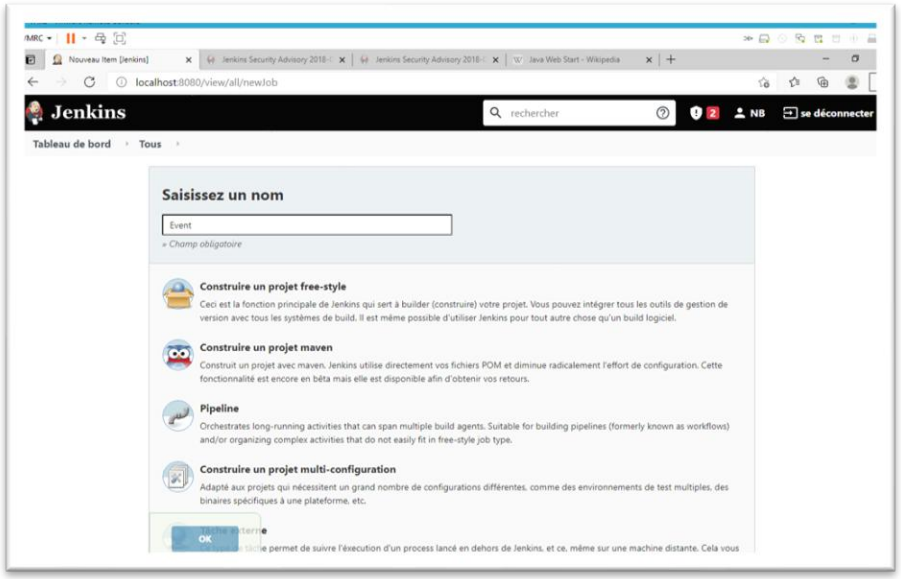
Comme je souhaitais utiliser des scripts PowerShell, il était nécessaire de télécharger un plugin supplémentaire



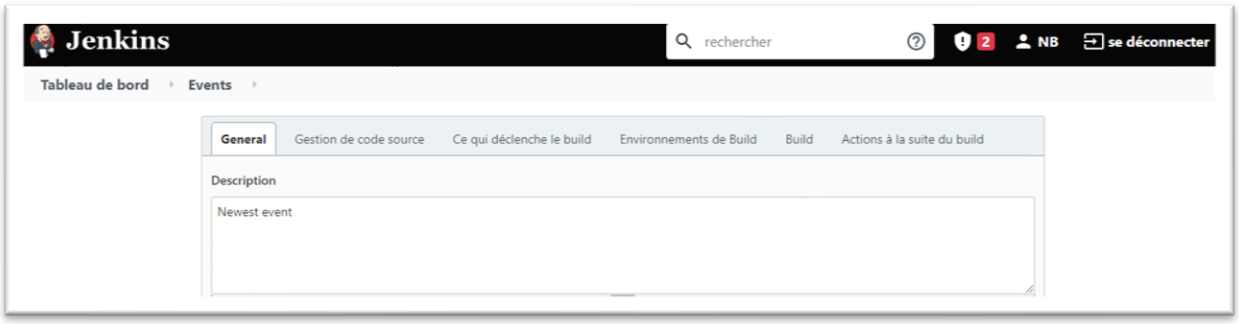
Et les tests pouvaient démarrer.

Exécution automatique d'un script PowerShell

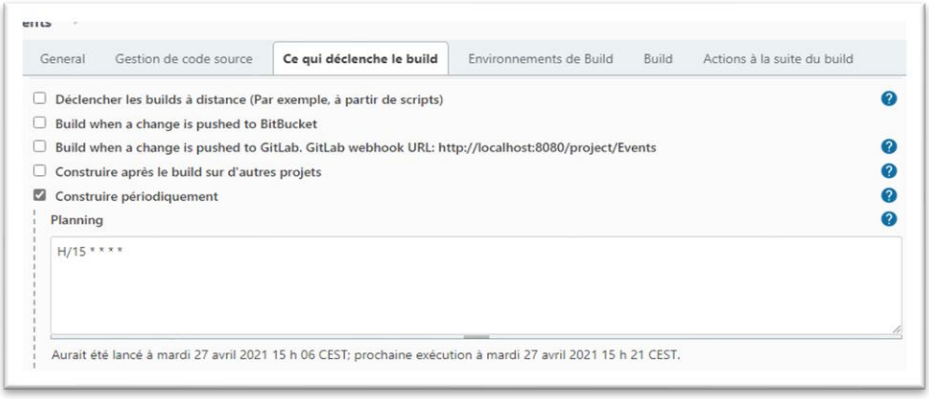
Pour faire un premier essai, j'ai créé un script PowerShell très simple qui affiche l'évènement le plus récent et redirige le contenu vers un fichier texte sans écraser son contenu. Je choisis « construire un projet free-style » sous le point « Nouveau Item » du tableau de bord



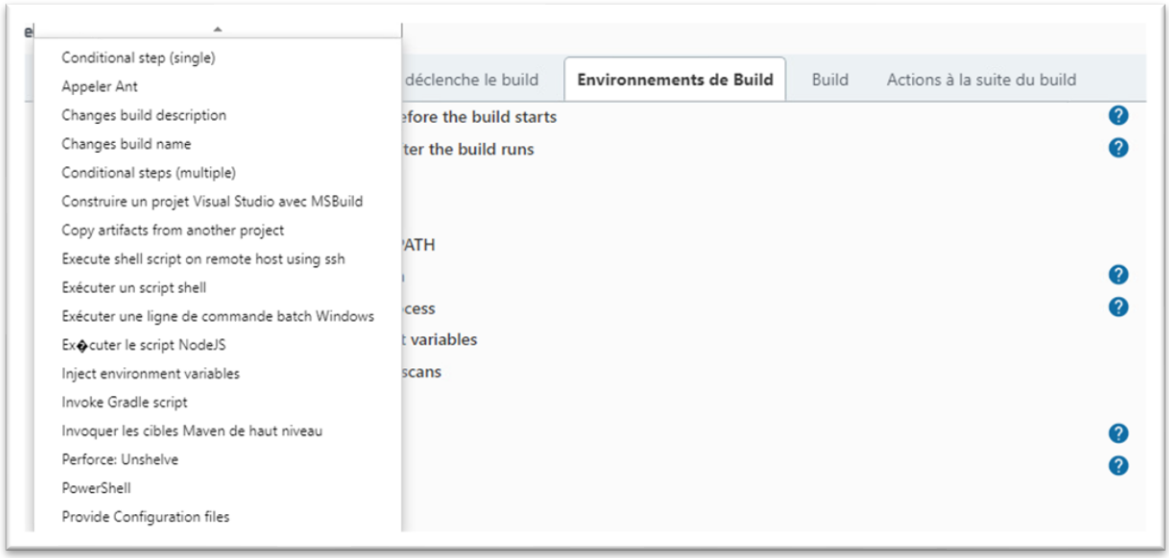
Je renseigne un nom pour ce job, et je choisis le mode de déclenchement



Pour faire exécuter la tâche régulièrement, Jenkins propose un outil de planning similaire à Crontab sous Linux. Ici, je choisis de lancer le script toutes les 15 minutes, tous les jours

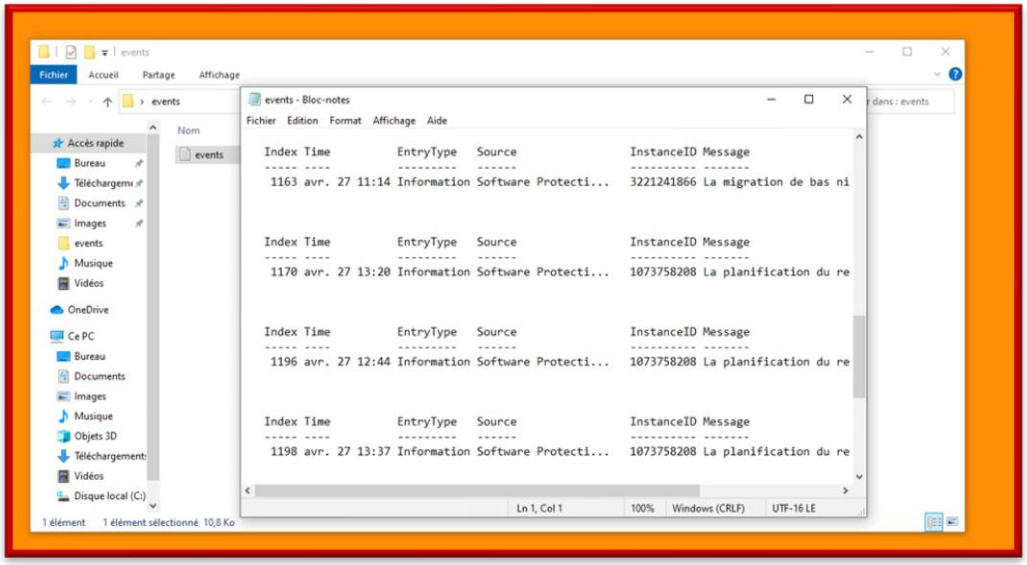


Sous le point « Build », je choisis PowerShell, puis j'entre le script directement dans la boite de dialogue



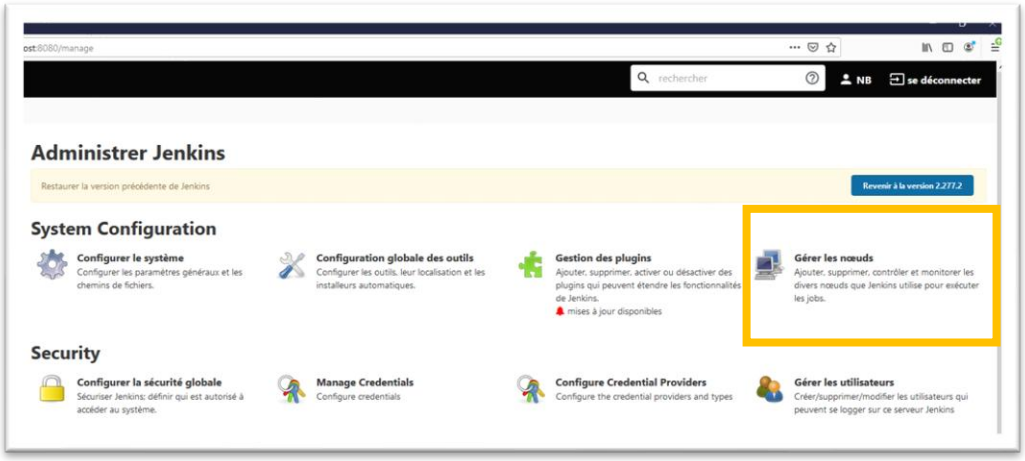
Je trouve, par la suite, des informations sur les tâches exécutées dans le tableau de bord, y compris la sortie console

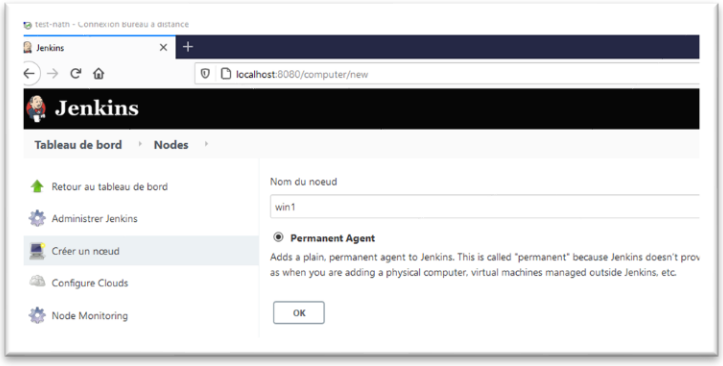
Et le résultat du script est là :



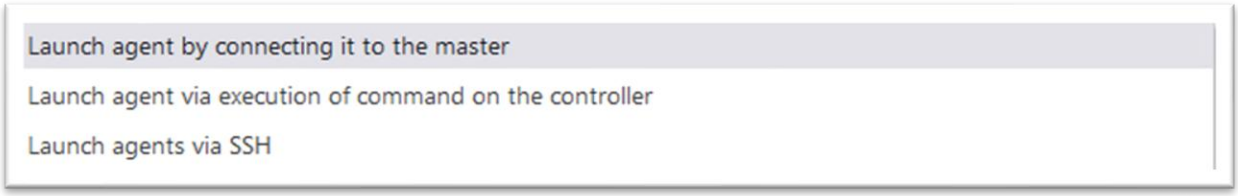
Pour exécuter les tâches automatisées sur une ou plusieurs machines distantes, il n'est pas nécessaire d'installer Jenkins sur tous les PCs. Cependant, toutes les machines doivent disposer de Java.

La première étape est d'ajouter le(s) nœud(s) distant(s) sur la machine maître Jenkins

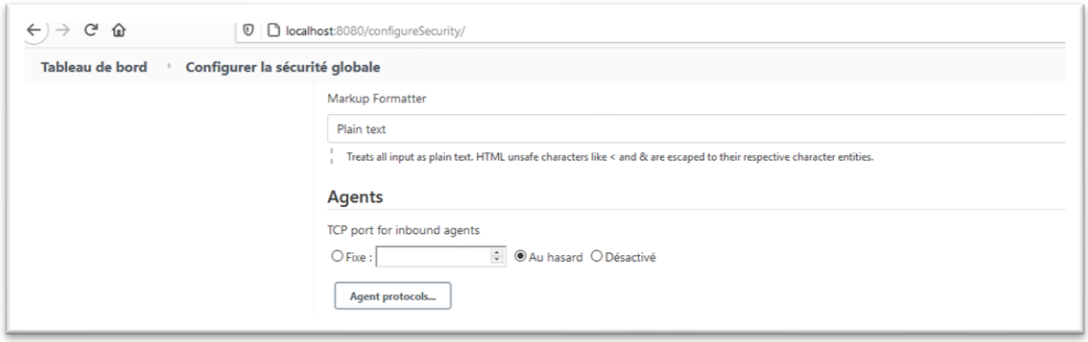




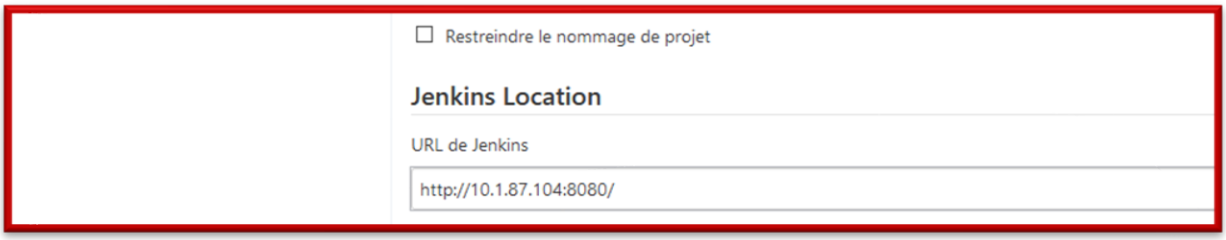
Il existe plusieurs méthodes de connexion entre maître et esclaves. J'ai commencé par tester "launch agent by connecting it to the master":



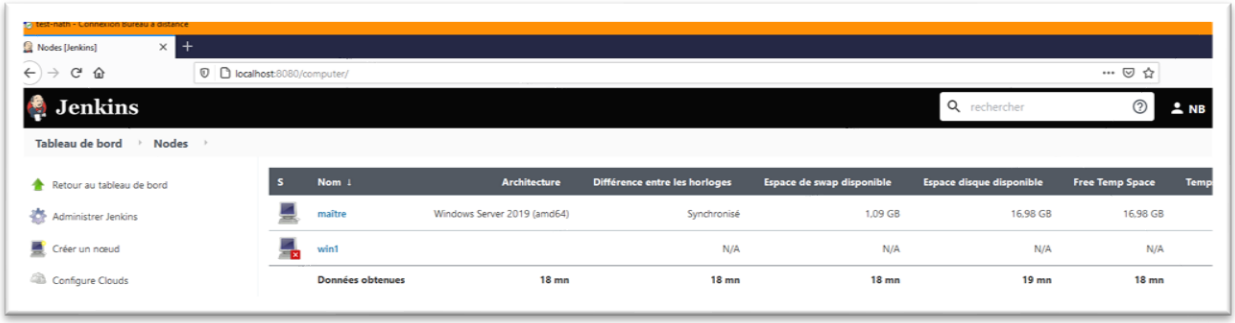
Si cette option est choisie, les connexions TCP pour les agents doivent être configurées pour les agents sur le master :



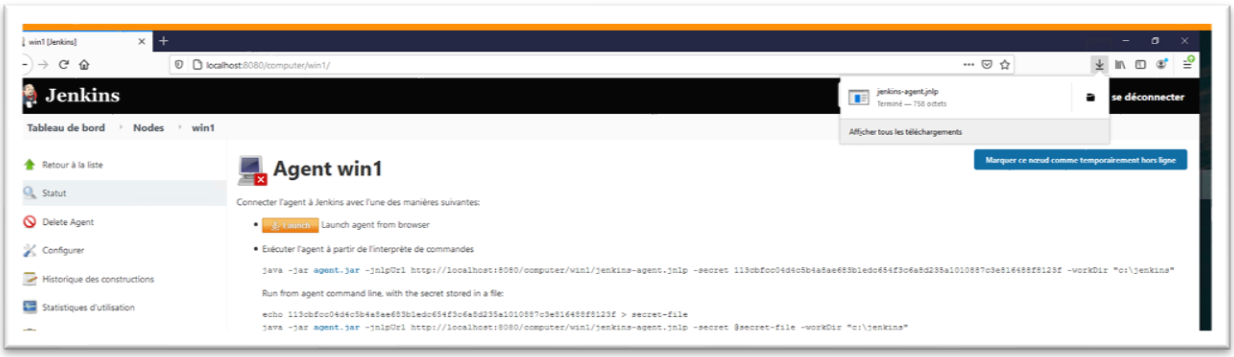
Afin que les nœuds puissent communiquer, le nom de la machine maître doit être configuré correctement (adresse IP ou nom NetBIOS)

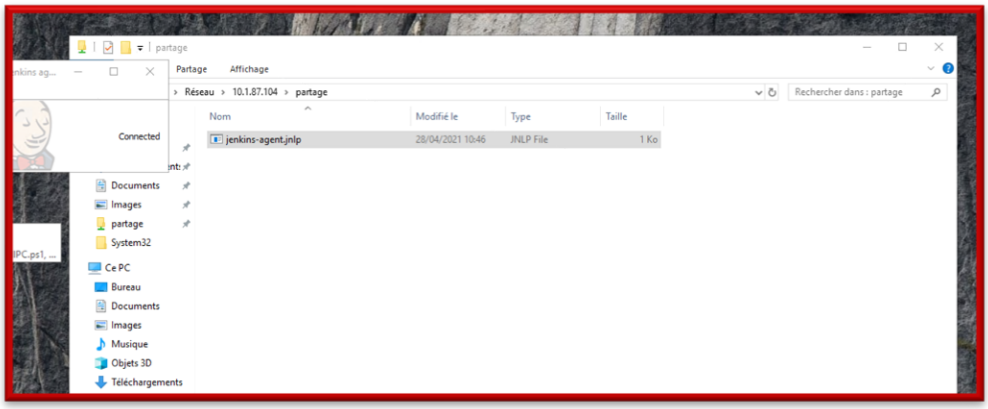


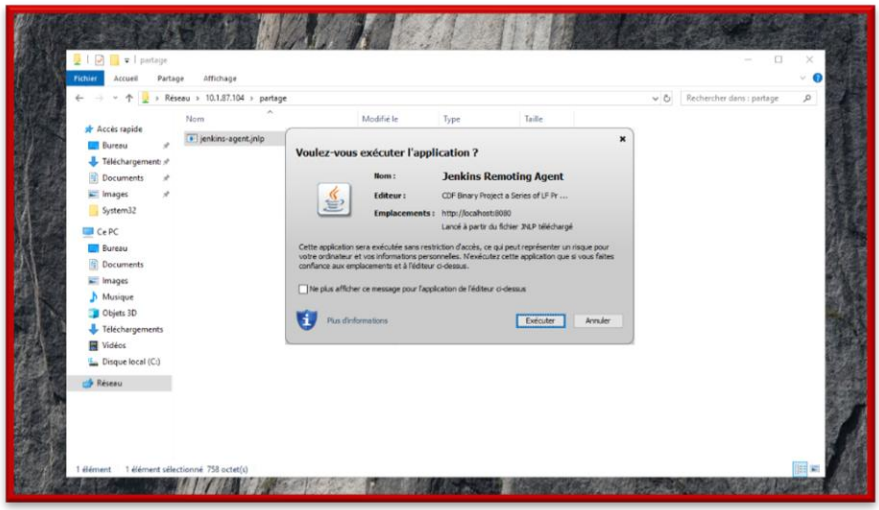
Le nœud apparait alors comme non connecté. Un clic droit permet d'accéder au téléchargement de l'agent Java de connexion pour le nœud :



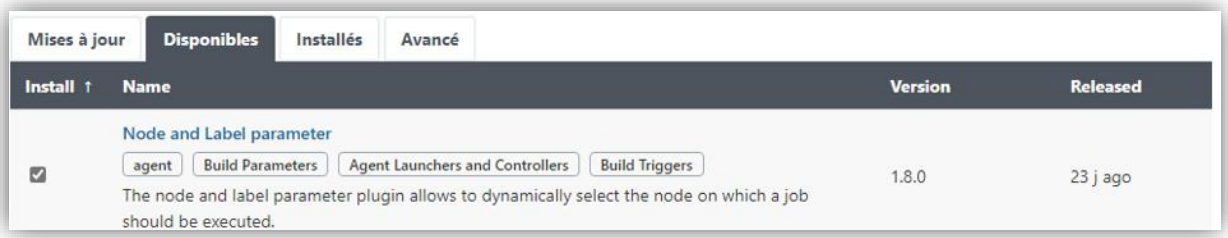
On télécharge l'agent sur la machine maître et le copie sur le nœud concerné pour pouvoir l'y installer. Dans le cas de ce test, j'ai utilisé un partage réseau pour transférer le logiciel :



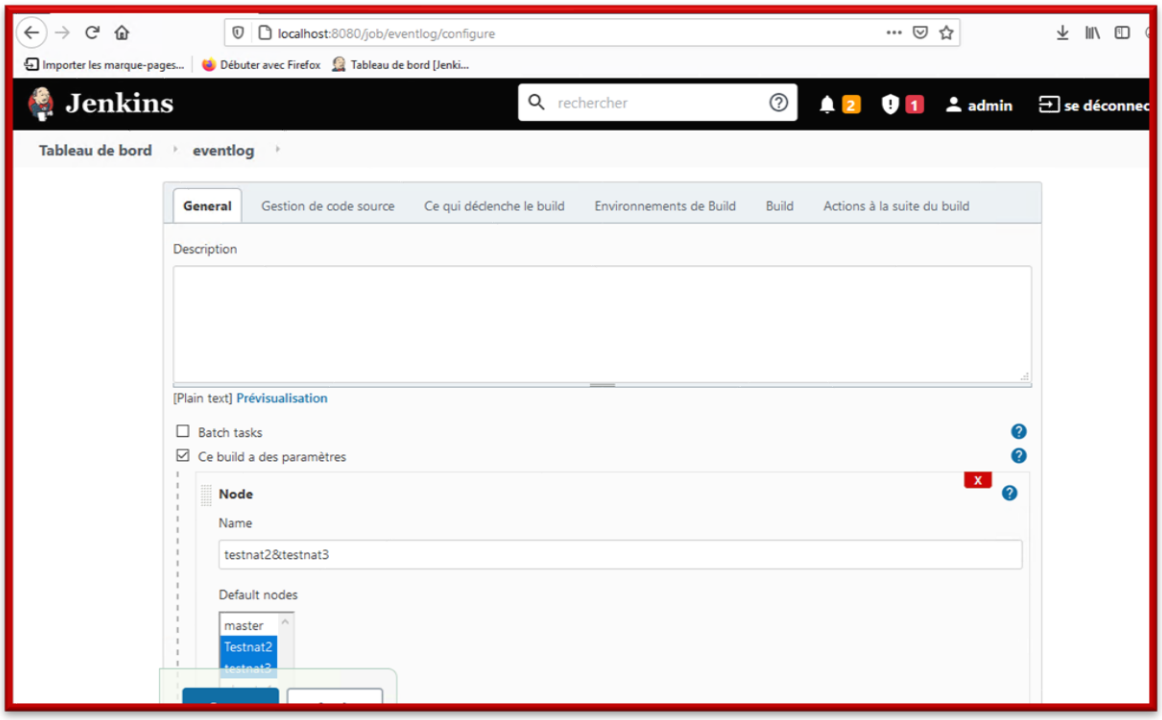




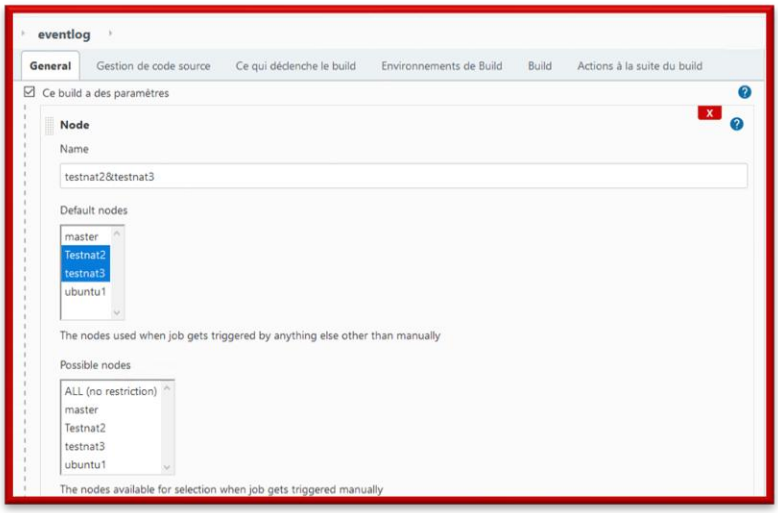
Une fois l'agent lancé sur la machine distante, il est possible d'exécuter des tâches sur celle-ci. Pour lancer une tâche sur plusieurs nœuds simultanément, il est nécessaire d'installer le plugin « Node And Label Parameter » dans le menu de gestion des plugins

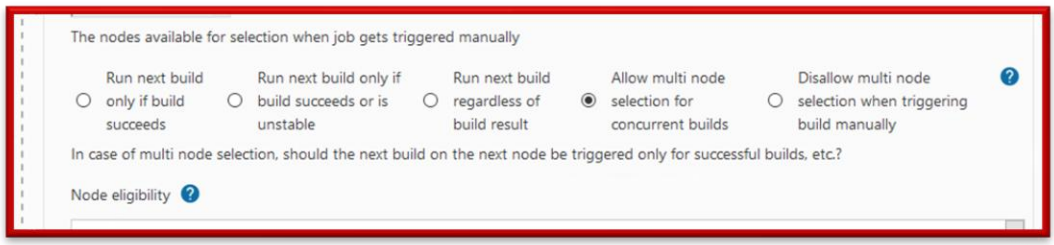


On configure alors le Job pour l'exécution simultanée sous « ce build a des paramètres »,



les nodes concernés étant le paramètre à indiquer, et permettre l'exécution simultanée

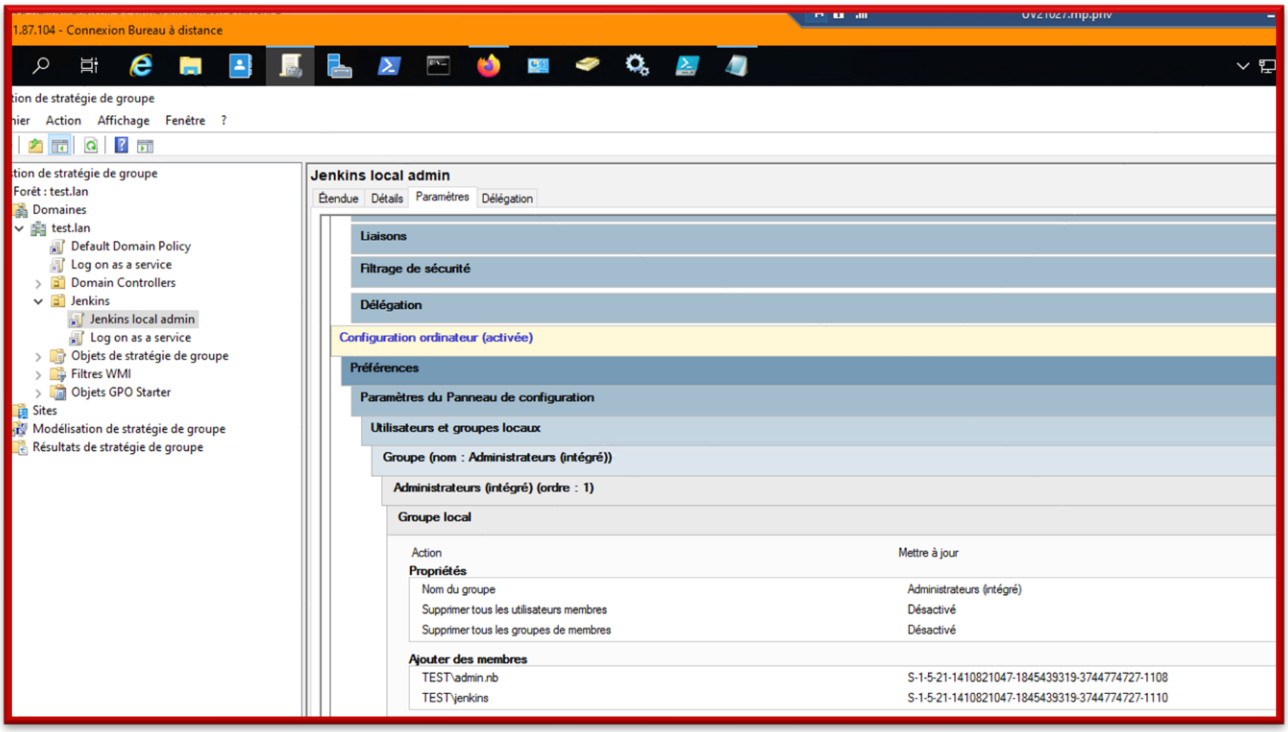


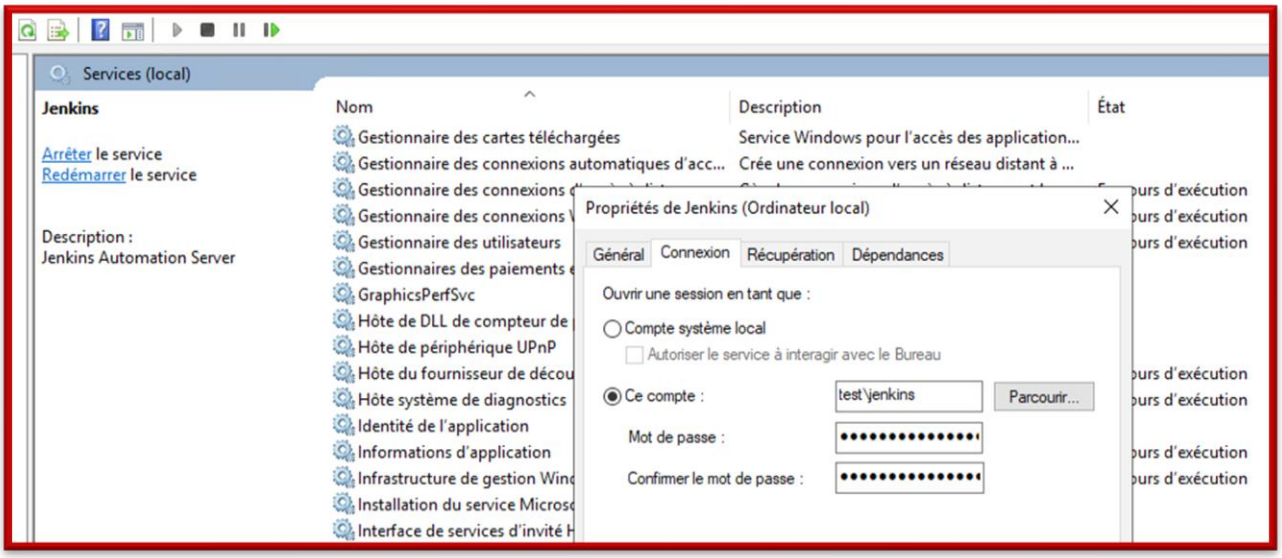


Pour une première démonstratrion ,la tâche était à nouveau une simple redirection de l'événement le plus récent affiché par PowerShell

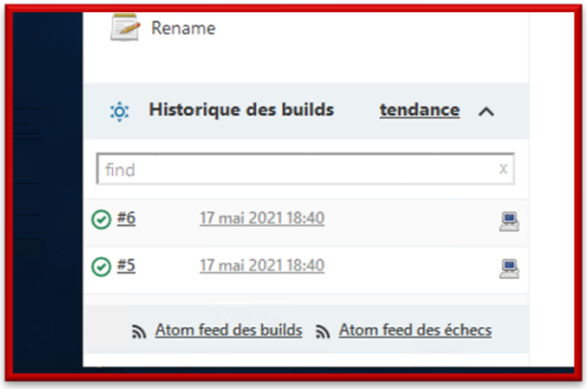
Bien évidemment, il est indispensable que les droits sur les éléments concernés par cette tâche soient accordés au compte utilisateur avec lequel s'exécute Jenkins.

Pour démontrer l'exécution d'une tâche sur plusieurs machines, j'ai modifié la configuration de Jenkins : le programme s'exécute en tant qu'utilisateur AD nommé « Jenkins » avec des droits d'administrateur local sur les machines de l'unité d'organisation « Jenkins » (paramétré par GPO).

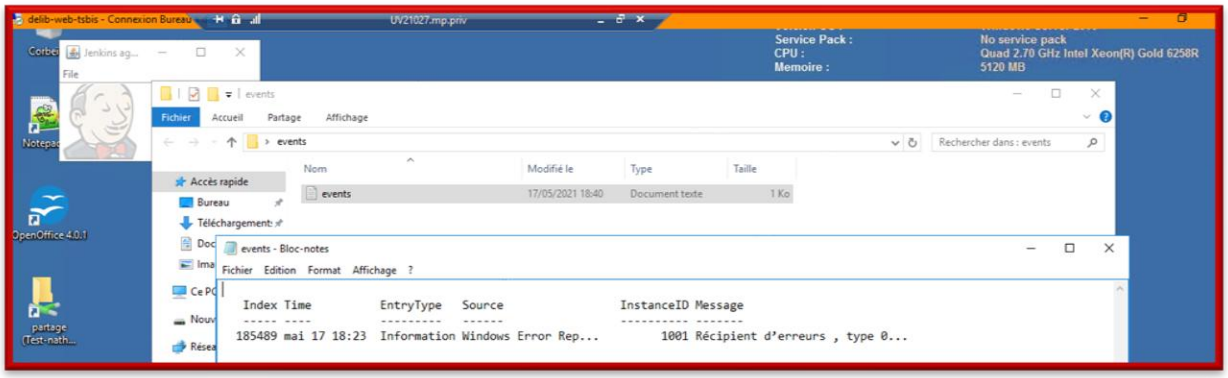


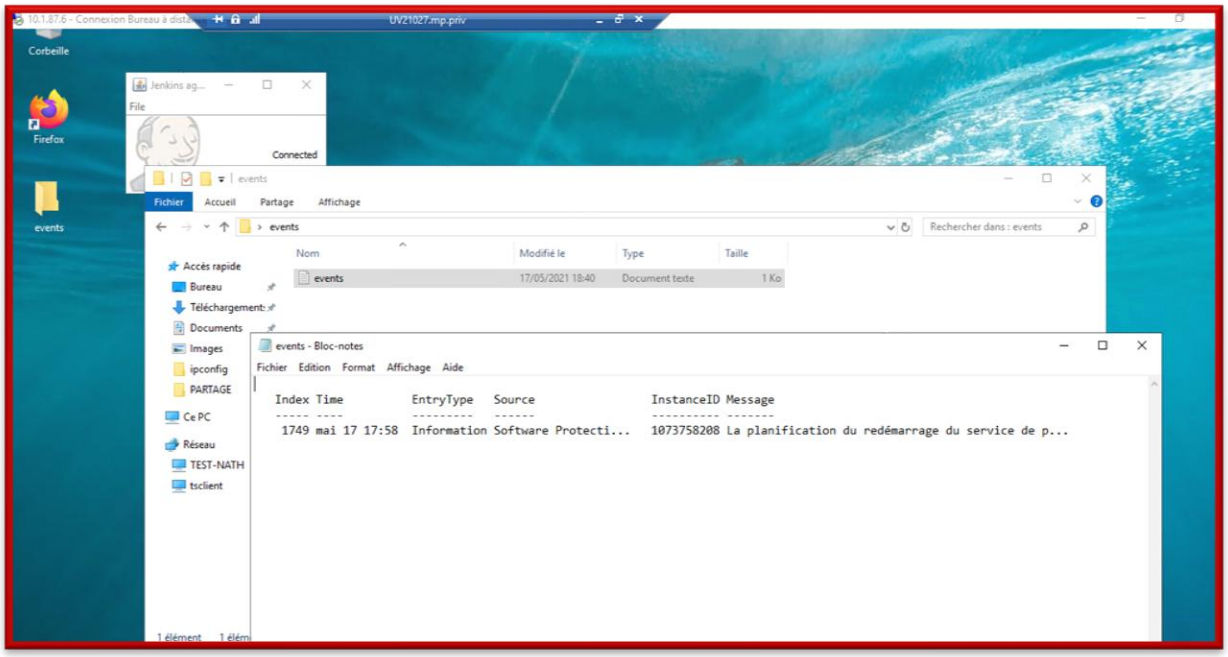


Le script Power Shell s'est alors exécuté sans problème sur les deux machines comme indiqué par la flèche verte dans l'Historique des builds



Et le fichier est bien créé sur les deux machines :





Jenkins est donc adapté pour centraliser l'automatisation de tâches par scripts PowerShell ou Bash. Parmi les demandes formulées par mon tuteur, la configuration du serveur DNS et suffixe de recherche étaient des options qui pourraient être effectuées par des scripts et donc automatisées par Jenkins. J'ai donc essayé d'automatiser ces tâches avec Jenkins.

Ne disposant pas de machines Debian à la mairie mais souhaitant simuler un environnement de Test le plus hétérogène possible, j'ai continué les tests sur Jenkins dans mon infrastructure personnelle, toujours avec un Master Jenkins sous Windows, mais avec des nodes Windows 10, Debian et Ubuntu.

Configuration des serveurs DNS et du suffixe de recherche par PowerShell (clients Windows)

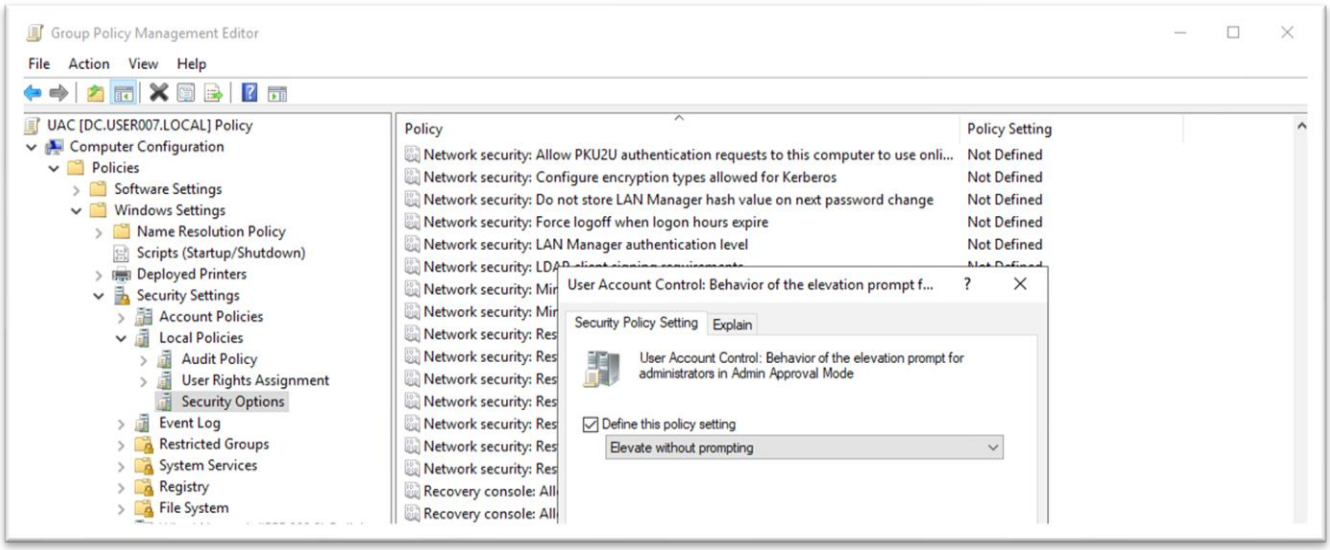
La plus grande difficulté de cette étape était de trouver un moyen de contourner l'obligation d'exécuter PowerShell en tant qu'administrateur à distance.

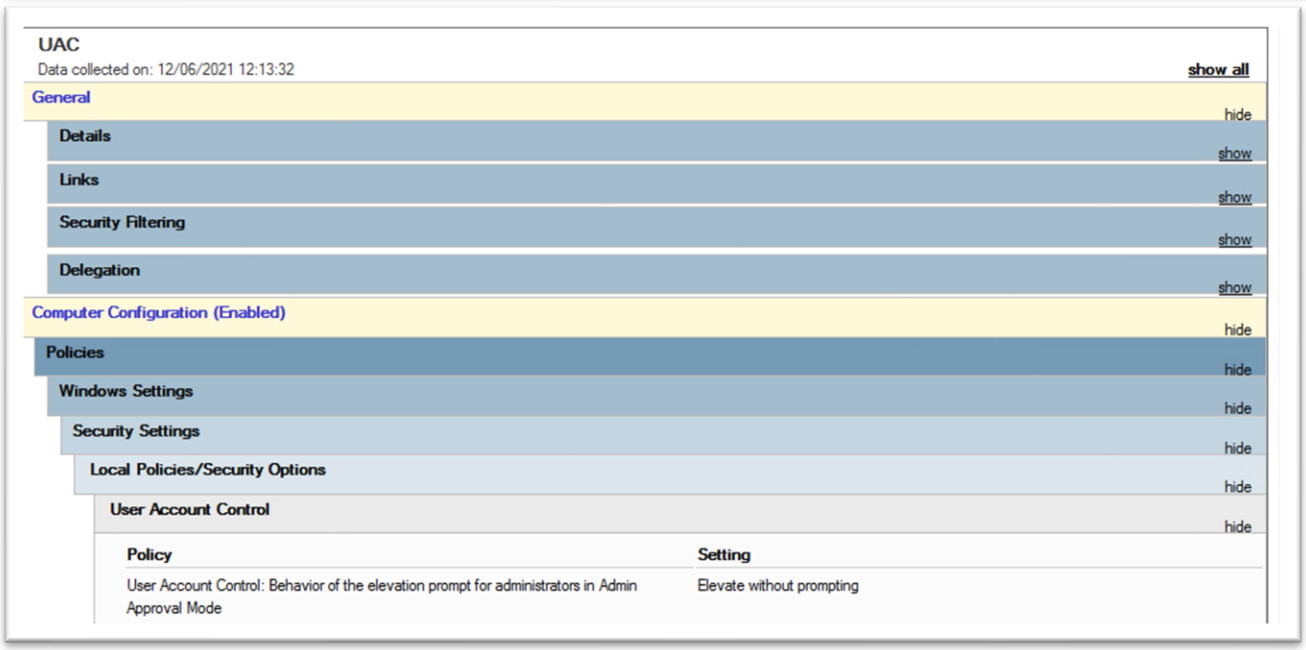
Il existe des commandes PowerShell qui permettent à un script PowerShell de s'attribuer des privilèges administrateur (source : https://www.blogabout.cloud/2020/05/1460/)

```
# vérifie le SID et Id de l'utilisateur actuel
$myWindowsID=[System.Security.Principal.WindowsIdentity]::GetCurrent()
$myWindowsPrincipal=new-object System.Security.Principal.WindowsPrincipal($myWindowsID)

# récupère le SID du compte administrateur intégré
$adminRole=[System.Security.Principal.WindowsBuiltInRole]::Administrator

# vérifie si le script s'exécute actuellement avec des privilèges élevés
if ($myWindowsPrincipal.IsInRole($adminRole))

  {
  # Si cela est le cas, modifier la couler d'arrière-plan
  $Host.UI.RawUI.WindowTitle = $myInvocation.MyCommand.Definition + "(Elevated)"
  $Host.UI.
  }
else
  {
  # exécution sans privilèges élevés, donc redémarrage du script en tant qu'administrateur

  # création d'un nouvel objet processus qui redémarre PowerShell
  $newProcess = new-object System.Diagnostics.ProcessStartInfo "PowerShell";

  # spécifie le script même en tant que paramètre
  $newProcess.Arguments = $myInvocation.MyCommand.Definition;

  # exécuter avec des privilèges élevés
  $newProcess.Verb = "runas";

  # démarrage du nouveau processus
  [System.Diagnostics.Process]::Start($newProcess);

  # quitter le processus actuel qui s'exécute sans privilèges élevés
  exit
     }
RawUI.BackgroundColor = "DarkBlue"
  clear-host
```

Ceci résout une première partie du problème, mais il reste le prompt de confirmation interactif :

Le seul moyen d'éviter cette interaction pour automatiser entièrement l'exécution du script est de désactiver les UAC (contrôles d'accès utilisateur). Ceci est risqué, l'intérêt des UAC étant d'empêcher l'exécution « aveugle » de n'importe quel code.Je me suis questionnée sur l'utilité d'employer Jenkins pour automatiser des tâches si cela requiert de créer des vulnérabilités dans le système. Pour essayer tout de même, j'ai créé une GPO qui supprime le prompt pour tous les administrateurs, et je l'ai liée à mon OU « Jenkins » :

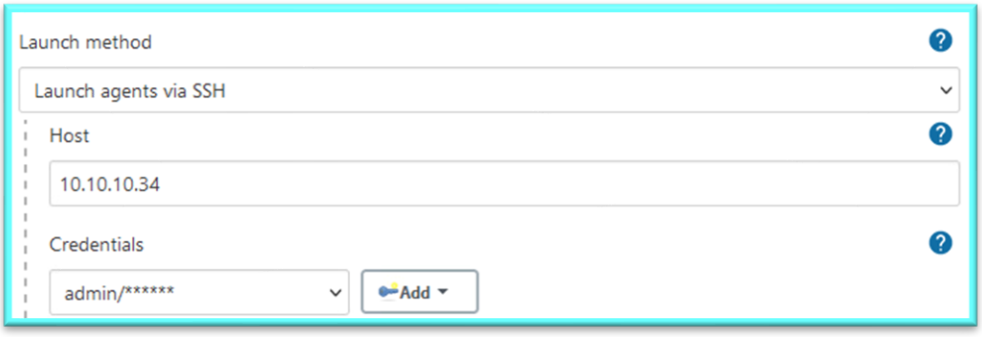




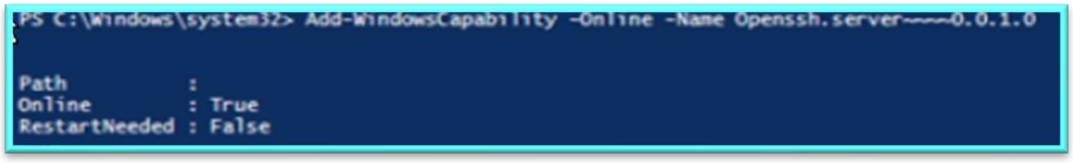
J'ai testé la modification du suffixe de recherche DNS sur les machines Windows avec le script suivant, en ajoutant juste une ligne qui ajoute le suffixe de recherche « test.lan » au code d'auto-élévation mentionné ci-dessus :

```
$myWindowsID=[System.Security.Principal.WindowsIdentity]::GetCurrent()
$myWindowsPrincipal=new-object System.Security.Principal.WindowsPrincipal($myWindowsID)

$adminRole=[System.Security.Principal.WindowsBuiltInRole]::Administrator

if ($myWindowsPrincipal.IsInRole($adminRole))
  {
  $Host.UI.RawUI.WindowTitle = $myInvocation.MyCommand.Definition + "(Elevated)"
  $Host.UI.RawUI.BackgroundColor = "DarkBlue"
  clear-host
  }
else
  {

  $newProcess = new-object System.Diagnostics.ProcessStartInfo "PowerShell";

  $newProcess.Arguments = $myInvocation.MyCommand.Definition;

  $newProcess.Verb = "runas";

  [System.Diagnostics.Process]::Start($newProcess);

  exit
  }
Set-DnsClientGlobalSetting -SuffixSearchList test.lan
```

Après de multiples tentatives sans succès, j'ai découvert que les scripts PowerShell ne s'exécutaient pas correctement quand le node concerné se connectait par l'agent. J'aidonc choisi la connexion SSH dans Jenkins



Ceci nécessitait donc l'installation et le démarrage de OpenSSH sur le master et le node



```
PS C:\Windows\system32> Start-service sshd
```

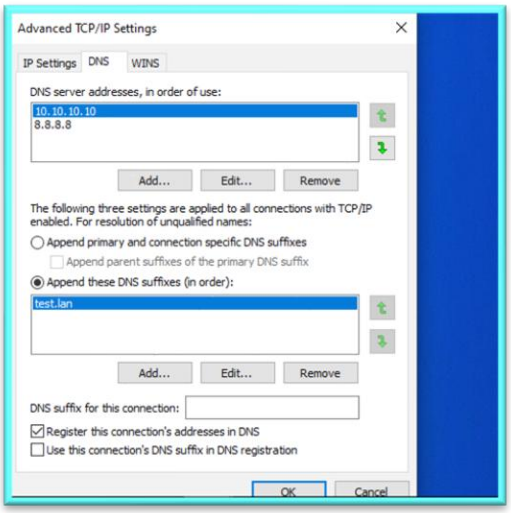
Et les pares-feu doivent permettre les communications sur le port 22

```
PS C:\Windows\system32> New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)' -Enabled True -Direction Inb
ound -Protocol TCP -Action Allow -LocalPort 22
```

La connexion au node doit être validée sur le master en ajoutant la clé publique du node au fichier « known_hosts » du master :

```
C:\Windows\system32>ssh-keyscan 10.10.10.34 >>c:\users\admin\.ssh\known_hosts
# 10.10.10.34:22 SSH-2.0-OpenSSH_for_Windows_7.7
# 10.10.10.34:22 SSH-2.0-OpenSSH_for_Windows_7.7
# 10.10.10.34:22 SSH-2.0-OpenSSH_for_Windows_7.7

C:\Windows\system32>
```

Une fois la connexion SSH établie, le script s'est alors exécuté sans problèmes, et le suffixe choisi a bien été ajouté



L'exécution de scripts avec privilèges élevés est donc possible.

Pour configurer l'adresse du serveur DNS, les étapes suivantes sont nécessaires :

Le serveur DNS est associé à une carte réseau , et la commande de configuration du ou des serveurs dns exige l'alias

source: https://docs.microsoft.com/en-us/powershell/module/dnsclient/set-dnsclientserveraddress?view=windowsserver2019-ps

S'il n'existait qu'une seule interface, il serait simple de trouver son alias. Cependant, ceci n'est quasiment jamais le cas, parce que Windows considère la boucle locale comme interface réseau: Et dans de nombreux cas, les serveurs disposent de plusieurs interfaces.

```
PS C:\Users\admin> Get-NetIPInterface

ifIndex InterfaceAlias                 AddressFamily NlMtu(Bytes) InterfaceMetric Dhcp     ConnectionState PolicyStore
------- --------------                 ------------- ------------ --------------- ----     --------------- -----------
5       Ethernet0                      IPv6                  1500              25 Enabled  Connected       ActiveStore
1       Loopback Pseudo-Interface 1    IPv6            4294967295              75 Disabled Connected       ActiveStore
5       Ethernet0                      IPv4                  1500              25 Disabled Connected       ActiveStore
1       Loopback Pseudo-Interface 1    IPv4            4294967295              75 Disabled Connected       ActiveStore
```

Un moyen serait de filtrer les Alias par appartenance à un réseau ; ceci suppose que l'adressage IP soit bien documenté pour pouvoir utiliser des expressions régulières. Admettons que l'on souhaite chercher l'interface qui appartient aux réseau 10.10.10.*

Je chercherais donc toute NIC qui ait une adresse IP contenant 10.10.10.

```
PS C:\Users\admin> $(Get-NetIPAddress | Where-Object {$_.IPAddress -like "10.10.10.*"}).InterfaceAlias
Ethernet0
```

Dans les cas présent, celle-ci s'appelle donc « Ethernet 0 ».

Cette information me servira pour créer la ligne de commande nécessaire pour Jenkins

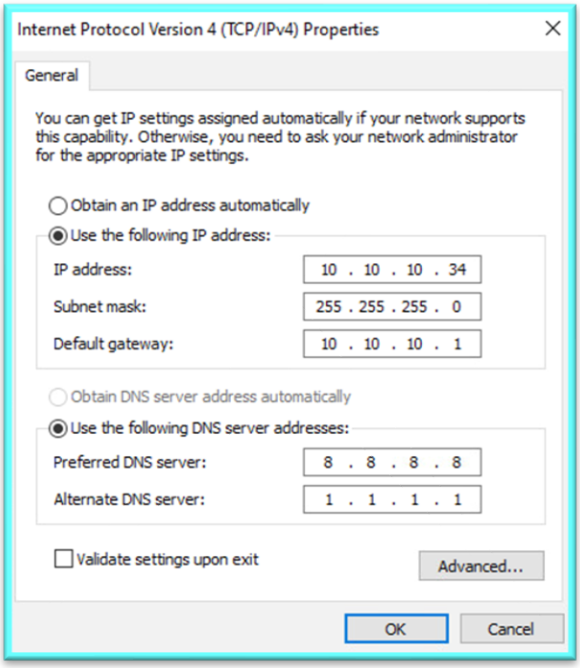
```
Set-DnsClientServerAddress -InterfaceAlias 'Ethernet0' -ServerAddresses "8.8.8.8",
  "1.1.1.1"
```

Alternativement, l'objet créé (nom d'interface) peut être inséré directement

```
Set-DnsClientServerAddress -InterfaceAlias $(Get-NetIPAddress |
Where-Object {$_.IpAddress -like "10.10.10.*"}).InterfaceAlias
-ServerAddresses "8.8.8.8",
  "1.1.1.1"
```

Le build s'est exécuté avec succès

Et les modifications ont bien été prises en compte



La configuration de suffixe de recherche et des serveurs DNS sur des nodes Windows est donc possible sous Jenkins. Cependant, il faut tenir compte des désavantages de son utilisation pour ces tâches :

Les UAC doivent être désactivées pour les administrateurs

Les interfaces réseau doivent toutes avoir le même nom ou faire partie du même réseau.

L'étape suivante était de tester la possibilité de configurer les serveurs et suffixes DNS sur des nodes Linux.

Configuration DNS sur nodes Linux

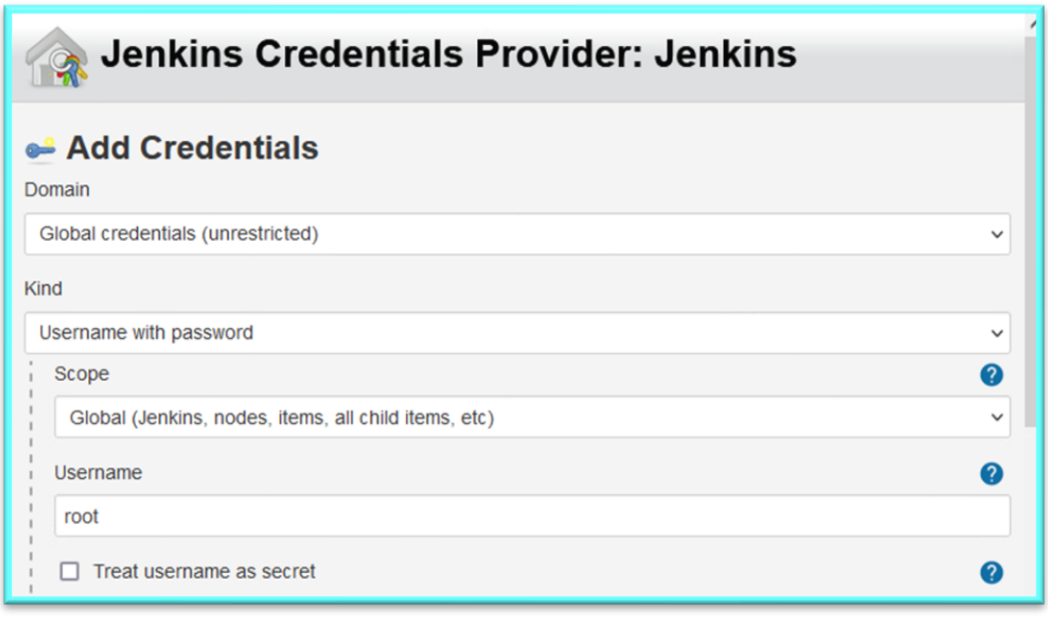
L'ajout de nodes Linux comporte les mêmes étapes que pour les nodes Windows du côté du master. Sur les nodes même, il est nécessaire d'avoir installé un client et serveur OpenSSH et Java. Les clés publiques des nodes doivent être ajoutées au fichier « known_hosts » de la même façon que pour les nodes Windows auparavant.

```
root@buster:~# java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-post-Debian-1deb10u1)
OpenJDK 64-Bit Server VM (build 11.0.11+9-post-Debian-1deb10u1, mix
ed mode, sharing)
root@buster:~#
```

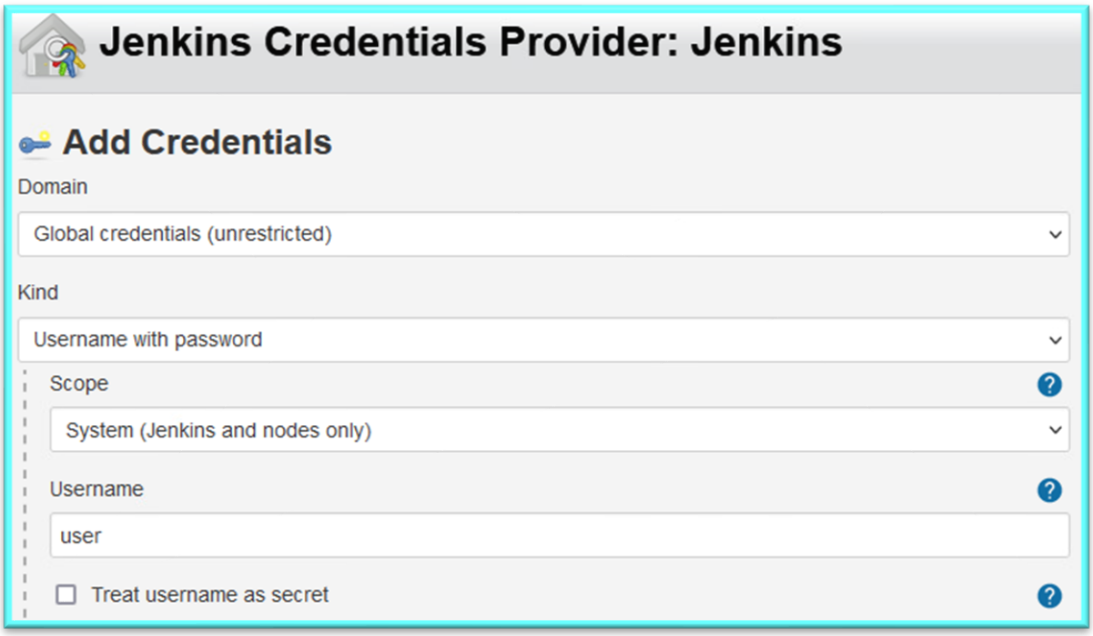
```
C:\Users\admin>ssh-keyscan 10.10.10.23 >>.ssh\known_hosts
# 10.10.10.23:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2
# 10.10.10.23:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2
# 10.10.10.23:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2

C:\Users\admin>ssh-keyscan 10.10.10.15 >>.ssh\known_hosts
# 10.10.10.15:22 SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
# 10.10.10.15:22 SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
# 10.10.10.15:22 SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
```

Comme l'utilisateur Jenkins n'existait pas pour mes machine Linux, car elles ne se trouvaient pas dans le domaine Windows, j'ai ajouté des identifiants spécifiques aux machines

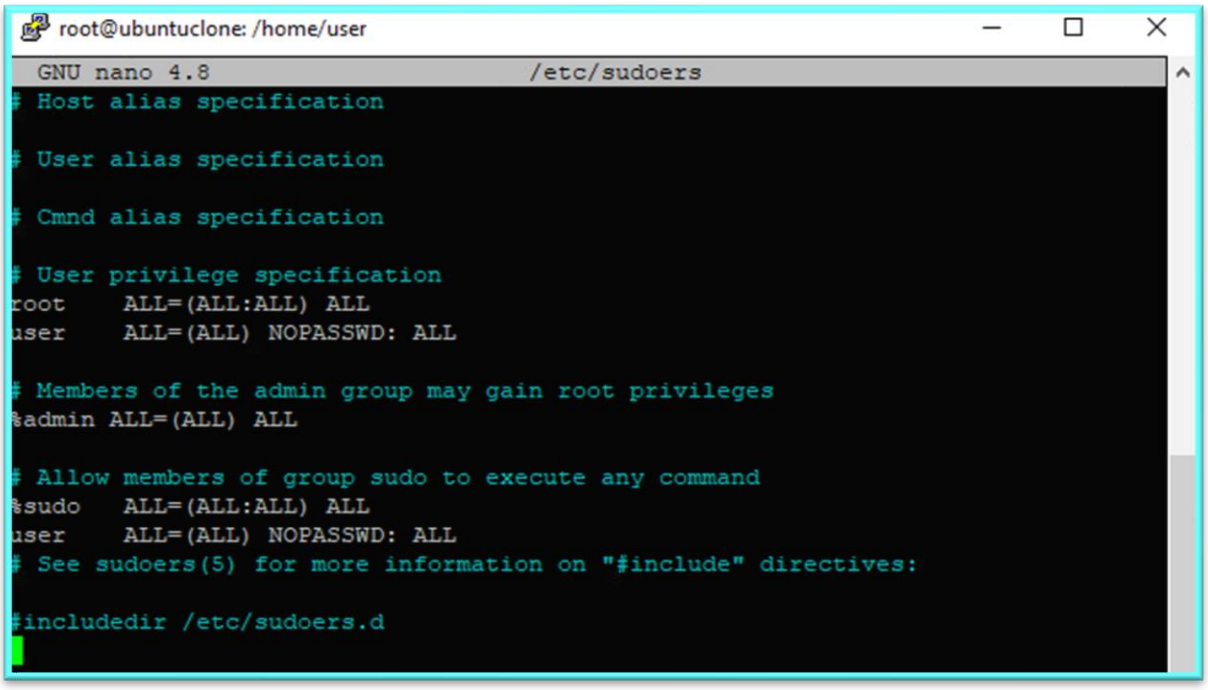
Pour Debian :



Pour Ubuntu



En sachant que cet utilisateur doit avoir des droits « root », il faut donc prévoir une manipulation supplémentaire sur toutes les machines Ubuntu pour ajouter l'utilisateur au groupe sudo



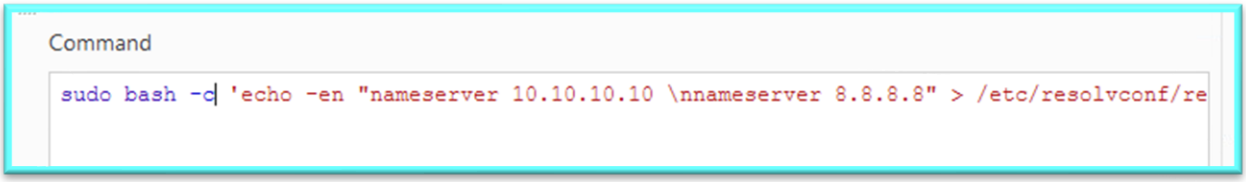
Pour modifier les serveur DNS sur les machines Debian 10 et Ubuntu 20.4, j'ai procédé de la façon suivante :

- Installation de l'utilitaire resolvconf
- Modification du fichier /etc/resolvconf/resolv.conf.d/head pour ajouter les serveurs DNS
- Modification du fichier /etc/resolvconf/resolv.conf.d/tail pour ajouter les suffixes de recherce

J’ai donc créé deux scripts pour les deux tâches.Cependant, comme l’exécution de tâches en sudo exige le mot de passe sous Ubuntu (malgré les manipulations du fichier sudo), il a fallu rajouter une commande supplémentaire aux script pour les nodes Ubuntu.

Pour les serveurs de noms sous Debian :

Sous Ubuntu

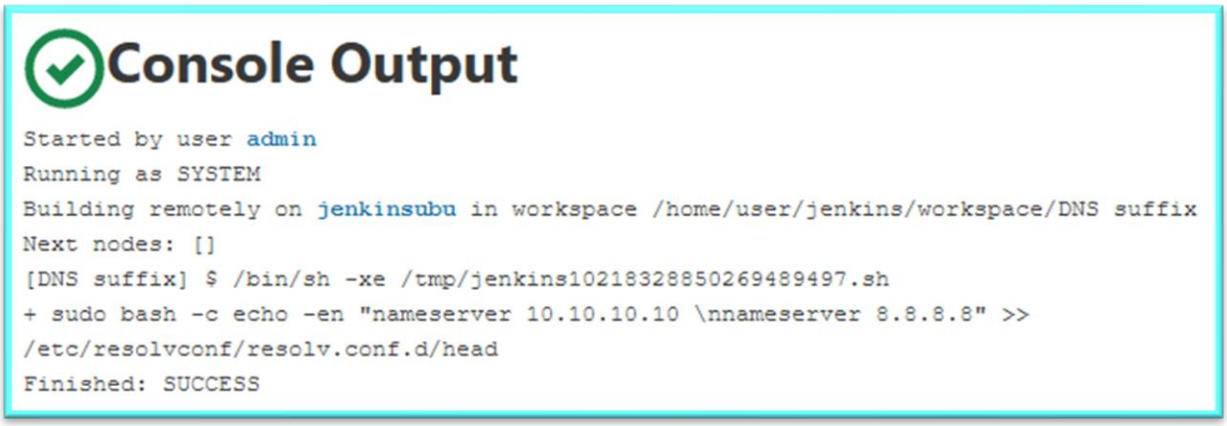


Pour les suffixes DNS sous Debian

pour Ubuntu

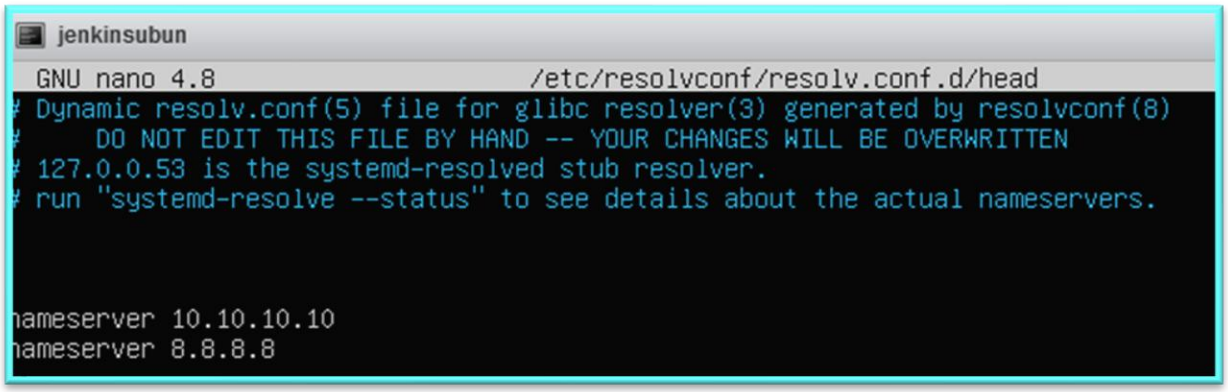
Tous les scripts se sont exécutés avec succès

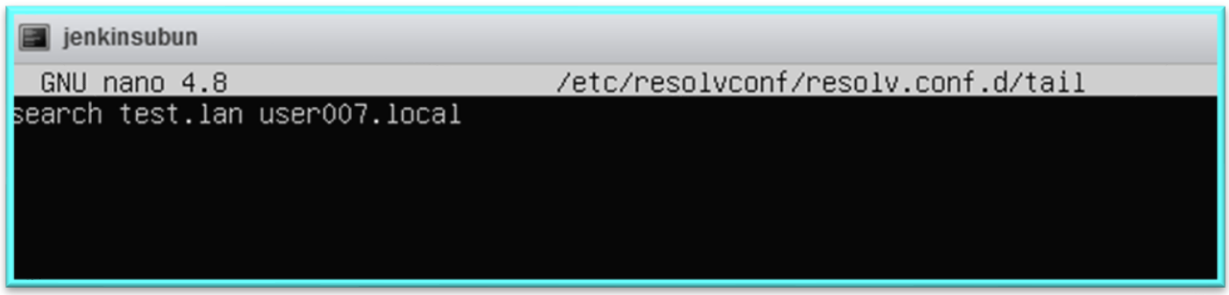
Pour l'ajout des serveurs de nom



Et pour les suffixes de recherche

La vérification sur les machines confirme les résultats :





Il est donc également possible de modifier la configuration DNS sur des machines Linux avec Jenkins.

Jenkins est pratique pour exécuter des scripts et commandes simples ne nécessitant pas de droits élevés, tels que l'export de journaux d'évènements ou le nettoyage régulier de dossiers partagés. Cependant, les configurations de systèmes par scripts se sont avérées laborieuses et demandaient des connaissances approfondies en langages de scripting. Jenkins ne semblait donc pas entièrement adapté aux demandes formulées dans le cadre de mon stage, nommément de faciliter la gestion et de permettre des économies de temps.

J'ai donc continué à étudier d'autres solutions logicielles.

WAPT

WAPT est un logiciel de configuration et de déploiement de logiciels pour les parcs Windows créé par l'entreprise TRANQUILIT. Depuis peu de temps, il y a moyen de gérer également les machines Linux. Cependant, dans le cadre des tests en entreprise, je me suis concentrée sur son fonctionnement sur les machines Windows, car la solution Cockpit-Projet que j'ai également testée s'est avérée satisfaisante pour la gestion logicielle des machines Linux du parc de la DN. J'ai testé d'intégrer une machine client Debian dans mon infrastructure privée.

Le fonctionnement de WAPT se rapproche de celui de Microsoft SCCM, mais s'inspire aussi du principe de « paquets » logiciels utilisés pour les systèmes Linux, d'où son nom WAPT (Windows -APT). Il permet de centraliser l'installation, la mise à jour et la désinstallation de logiciels à l'aide d'une interface graphique. WAPT est certifié CSPN (certification de premier niveau) par l'ANSSI.

WAPT est basé sur le principe de « paquets » logiciels mis à disposition dans une dépôt (répertoire web) grâce au service Nginx du serveur ; celui-ci peut être installé sous Windows ou Linux, mais les éditeurs recommandent fortement l'utilisation de Debian. Il est possible d'utiliser de multiples dépôts, par exemple, pour créer des pools de logiciels avec différents droits d'accès ou pour adapter le déploiement à des infrastructures multisites. Afin de limiter l'utilisation de la bande passante, les dépôts peuvent également être installés sur un client local WAPT.

Le serveur WAPT peut également faire remonter des informations sur l'inventaire matériel et logiciel des clients, fonction qui était particulièrement intéressante dans le cadre de mon projet de stage.

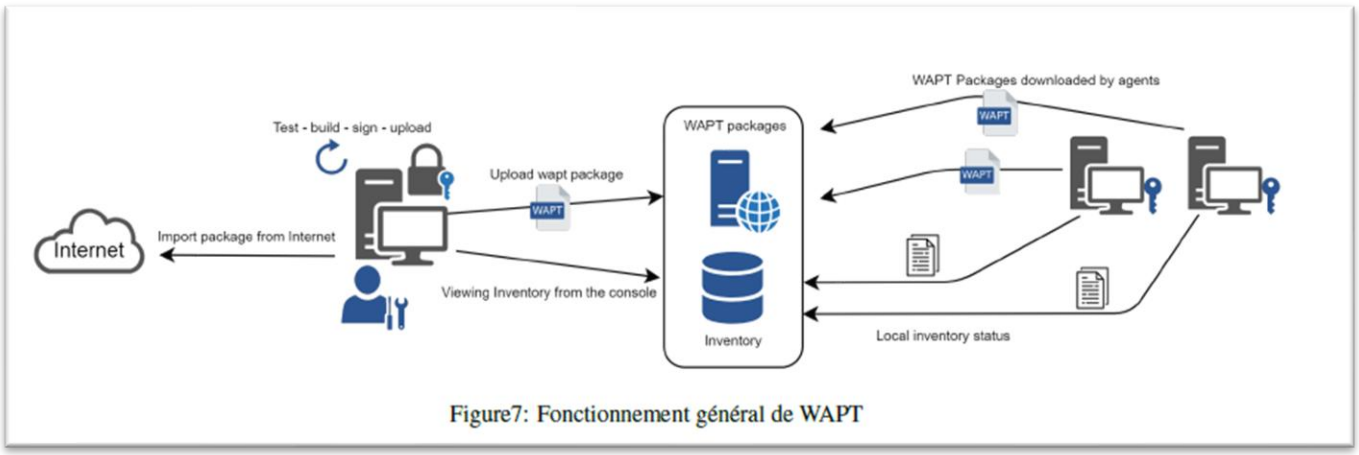
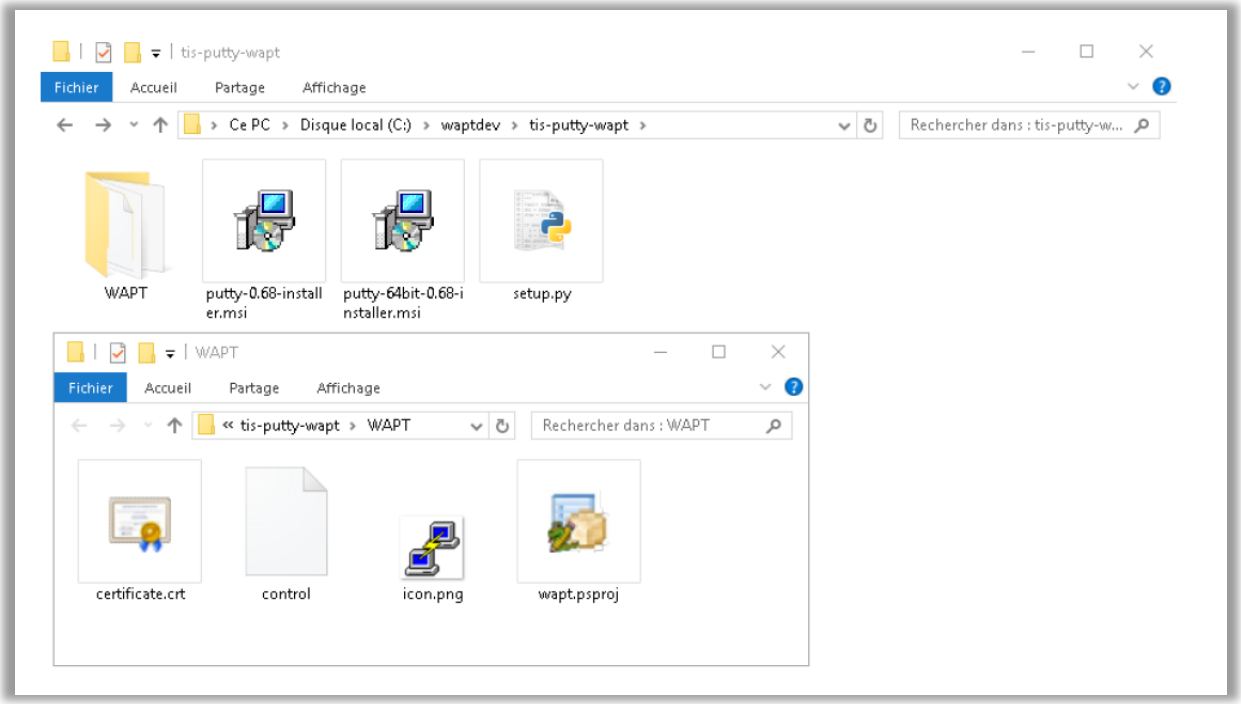


Figure7: Fonctionnement général de WAPT

source : documentation TRANQUILIT

Le Principe de paquets

Un « paquet » WAPT contient les binaires d'un logiciel et des fichiers de configuration nécessaires à son installation.



Il existe 7 types différents de paquets sous WAPT :

« **base** » : paquet de logiciel standard

« **group** » : permet de créer des groupes de logiciels destinés à des profils de machines (qui peuvent, à leur tour, faire partie d'un groupe)

« **host** » : un paquet attribué à une machine spécifique (identifié par FQDN)

« **unit** » : paquet attribué à une unité d'organisation Active Directory

« **wsus** » : paquet contenant la liste de mises à jour Microsoft autorisées et interdites. Ce paquet permet de filtrer les mises à jour Windows lors du prochain update Windows

« **selfservice** » : contient une liste de paquets et une liste d'utilisateurs (domaine ou local) qui sont autorisés à les installer

« **profile** » : paquet prévu à être installé sur les machines appartenant à un groupe Active Directory précis

Il est possible de créer des « dépendances » entre paquets comme elles existent pour les systèmes UNIX : un paquet spécifique ne sera installé que lorsque les paquets des dépendances sont également installés.

WAPT utilise le cryptage asymétrique (certificat public/clé privée) pour sécuriser les échanges entre serveur et client. Le certificat peut être issue d'une Autorité de certification (CA) interne ou commerciale. Dans le cadre de mes tests, j'ai utilisé un certificat auto-signé.

Versions du logiciel

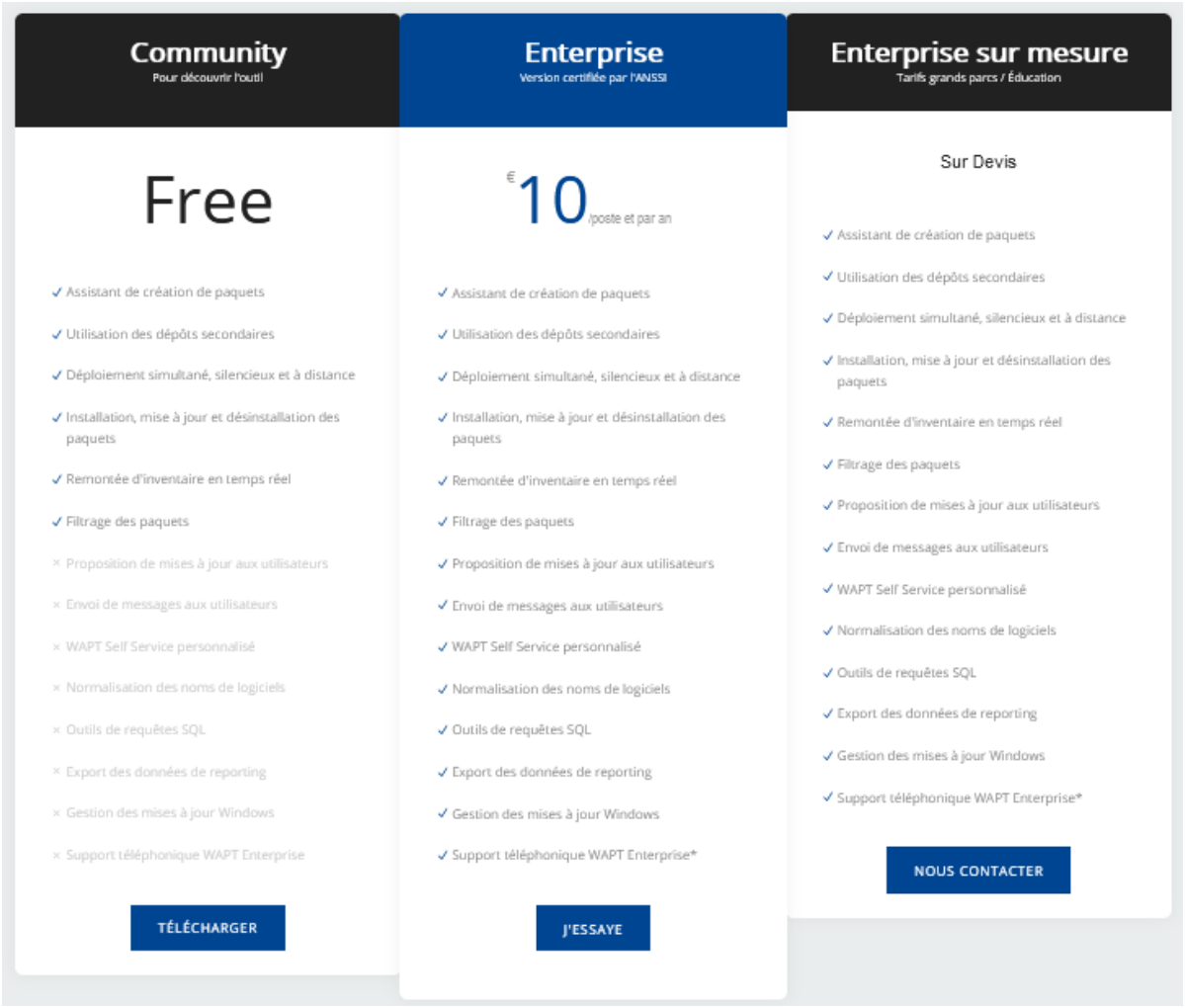
TranquillT propose actuellement trois versions de WAPT : Enterprise (payante), Discovery et Community. Le support pour la version Community sera arrête au mois de juin 2021.

Voici un comparatif des caractéristiques des trois versions :

Comparaison des caractéristiques entre les versions WAPT en date du 2021-05-12

Caractéristique	Enterprise	Discovery	Community
Deploy, update and remove software on hosts running	✓	✓	✓ [1]
Maintenance et support (voir note de bas de page pour les conditions)	équipe Tranquil IT [4]	forum Tranquil IT [4]	Communauté OpenSource
Licensed under	Proprietary	Proprietary	GPLv3
Limitation du nombre d'appareils	aucune limite	300	aucune limite
Version de Python utilisée dans le code et les paquets WAPT	3+ (actuel)	3+ (actuel)	2.7 (obsolete)
Déployer et mettre à jour les **configurations dans le contexte du SYSTÈME**v	✓	✓	✓ [1]
Déployer et mettre à jour les **configurations dans le contexte de l'UTILISATEUR**	✓	✓	✓ [1]
Obtenir un **inventaire complet** du matériel, des logiciels et des progiciels appliqués avec WAPT	✓	✓	✓
Bénéficier du **libre-service différencié** (les utilisateurs autorisés peuvent installer les logiciels autorisés à partir de magasins de paquets WAPT autorisés)	✓	✗	✗
Bénéficiez de **Mises à jour Windows simplifiées** qui fonctionnent bien mieux qu'un WSUS standard (seuls les KB requis sont téléchargés depuis Microsoft)	✓	✗	✗
Simplifiez et structurez votre charge de travail administrative en appliquant des paquets WAPT à vos UO	✓	✗	✗
Configurer et gérer facilement les relais WAPT **pour préserver la bande passante** pour les scénarios *Edge Computing*.	✓	✗	✗
Accédez à des **paquets WAPT prêts à être déployés** pour des logiciels communs gratuits	✓	✓	✓ [1]
Travailler avec des **recettes python facilement vérifiables** pour l'installation, la mise à jour et la suppression de logiciels et de configurations	✓	✓	✓ [1]
Bénéficiez de **centaines d'assistants** pour simplifier le conditionnement des logiciels	✓ [2]	✓	✓ [1]
Chiffrez vos données sensibles pour le transport (clés de licence du logiciel, login, mot de passe, FQDN des serveurs, informations API pour l'enregistrement du logiciel auprès du vendeur, etc)	✓	✗	✗
Automatisez l'audit de vos configurations pour une conformité **facile, automatisée et toujours à jour**	✓	✗	✗
Profitez de la puissance du SQL intégré à la console WAPT pour faire **les rapports dont vous avez besoin pour votre travail quotidien d'administrateur système ou dont votre organisation a besoin pour ses décisions budgétaires.**	✓	✗	✗
Authentifiez vos Administrateurs WAPT avec **Active Directory, LDAP, ou avec leurs certificats personnels**	✓	✗	✗ [3]
Bénéficiez de rôles différenciés entre vos Développeurs de Paquets et vos Gestionnaires de Déploiement afin que vous puissiez **déléguer vos pouvoirs WAPT aux personnes les plus adéquates** (les développeurs de paquets connaissent les implications en matière de sécurité, les déployeurs connaissent les besoins des utilisateurs)	✓	✗	✗
Bénéficier du mode multi-tenant et multi-client avec les ACLs pour les MSPs ou les grandes organisations multi-départementales ou internationales utilisant un mécanisme interne basé sur la PKI pour le périmètre autorisé	✓	✗	✗
Partage d'écran et prise en main à distance simple pour l'assistance aux utilisateurs, construit avec le même niveau de sécurité et de confidentialité que WAPT (nécessite un hôte supplémentaire)	✓	✗	✗
Support continu de Windows XP avec WAPT pour les machines-outils d'usine, les équipements médicaux des hôpitaux, les instruments de recherche coûteux et difficiles à remplacer, etc	✓ [5]	✗	✗
Update package directly on console with `update_package` function	✓	✗	✗
Intégration de l'inventaire WAPT aux outils Glpi ITSM populaires	✓	✗	✗
Vérifié et approuvé par l'agence de cybersécurité internationalement reconnue ANSSI  **. WAPT est le seul logiciel de déploiement au monde avec ce niveau de certification**	✓	✗	✗

Voici les tarifs de WAPT au mois de juin 2021



Comme le parc informatique de la DN comprend plus de 2000 postes, j'ai testé la version entreprise avec une licence d'essai.

Bonjour,

Nous vous remercions de l'intérêt que vous portez à notre logiciel, WAPT Enterprise.

Suite à notre échange téléphonique, veuillez trouver ci-dessous toutes les informations concernant la mise en service de votre licence d'essai pour 2 000 postes **valide jusqu'au 18/06/2021**.

Vous trouverez ci-joint :

- la procédure d'installation de WAPT Enterprise
- le fichier de licence à intégrer (installation décrite dans le pdf joint).
- la présentation complète de WAPT Enterprise
- Votre login : 51d537a1-dfbd-4a3a-87bb-c3381114be13
- Votre password : rJ6kKP5FEUFy3OSP
- Site de dépôt : https://srvwapt-pro.tranquil.it/entreprise/

En parallèle, j'ai effectué des tests sur la version Community sur mon infrastructure privée.

WAPT offre de nombreuses possibilités intéressantes pour personnaliser l'installation de logiciels et adapter le déploiement à différentes infrastructures. Cependant, l'étude complète de ce logiciel n'étant pas le sujet principal de ce rapport, je ne ferai pas une présentation exhaustive de toutes les fonctionnalités ; WAPT est une des solutions de gestion

de configuration que j'ai testées. Je présenterai donc les fonctionnalités de base : déploiement de paquets, inventaire logiciel et matériel.

Environnement de Test

Mairie : 1 serveur WAPT Enterprise sous Ubuntu 20.4, 3 clients sous Windows Server 2016, 1 client sous Ubuntu20.4

Privé : 1 Serveur WAPT Community sous Debian 10, client sous Windows 10, 1 client sous Ubuntu 20.4, 1client sous Debian 10

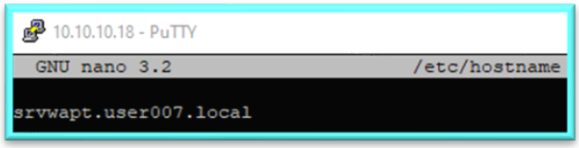
Préparations de l'environnement et du serveur

Les éditeurs de WAPT recommandent d'installer le logiciel sur une serveur Debian. Cependant, la DN ne disposant que de machines virtuelles sous Ubuntu, j'ai effectué l'installation sous Ubuntu 20.4. J'ai utilisé une machine virtuelle Debian 10 pour les tests dans mon infrastructure privée. Les étapes d'installation sont clairement décrites sur le site des éditeurs et comprennent :

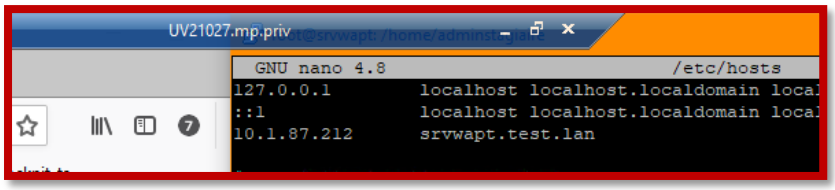
La préparation du serveur WAPT (Debian/Ubuntu) :

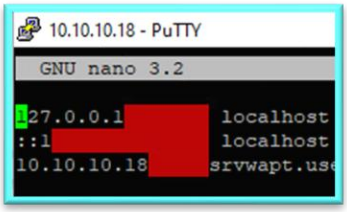
Nommage : J'ai nommé les serveurs WAPT srvwapt.domaine dans les deux infrastructures sur lesquelles j'ai travaillé. Ceci nécessitait la modification des fichiers /etc/hostname et /etc/hosts des serveurs même :

Fichier hostname :

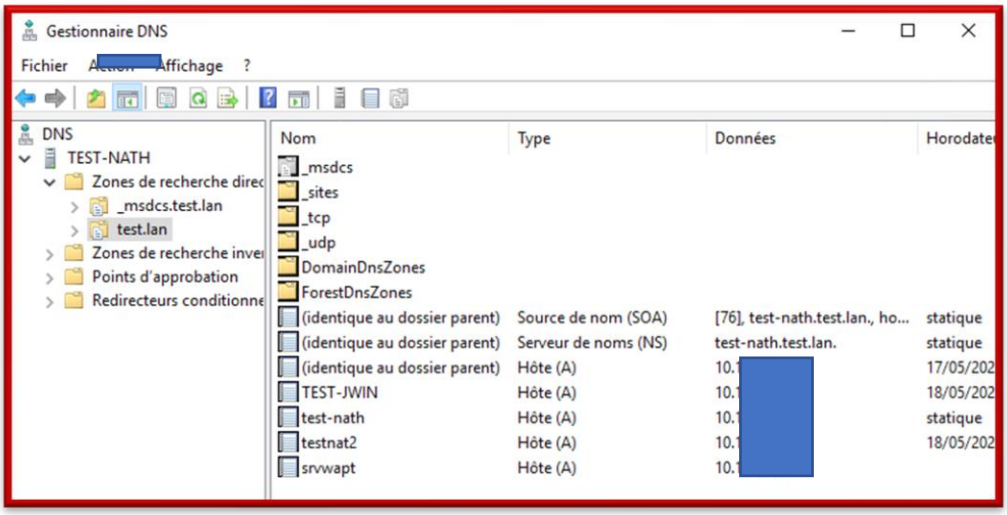


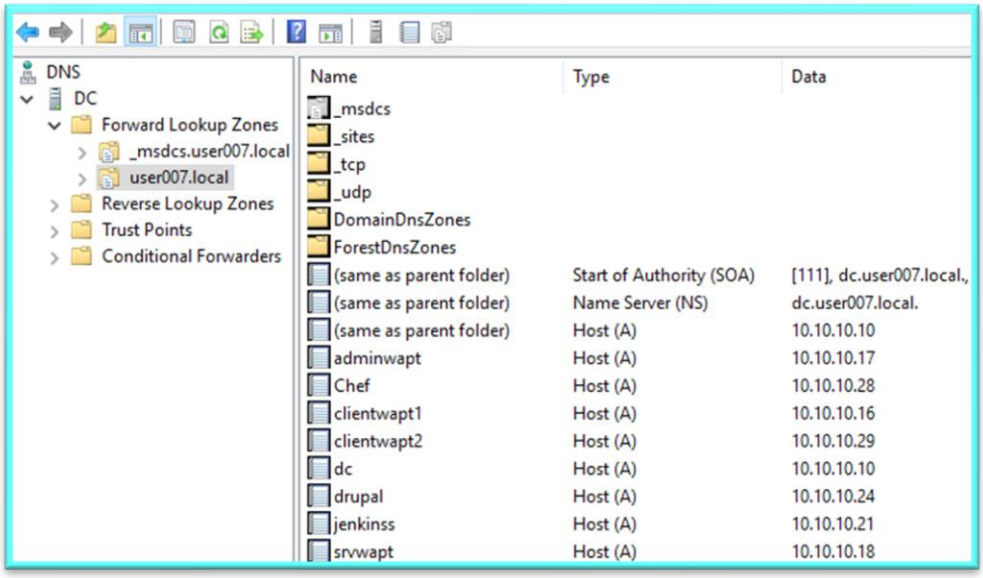
Fichiers hosts :



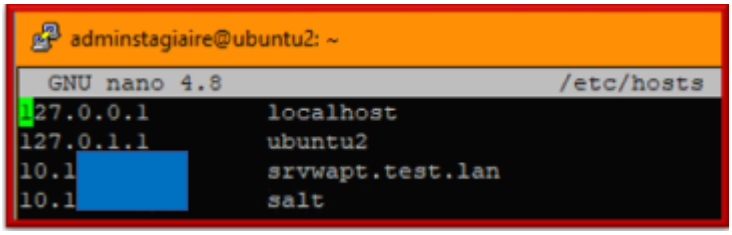


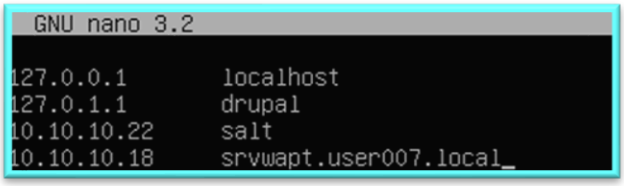
Les machines clients devant pouvoir contacter le serveur WAPT, des modifications de leur côté étaient également nécessaires. Les clients WAPT sous Windows faisaient, dans les deux infrastructures, partie d'un domaine dont le contrôleur faisait office de serveur DNS. J'ai donc créé une entrée DNS sur les serveurs DNS respectifs.





Bien évidemment, il aurait été possible de renseigner le serveur DNS du domaine sur les clients Linux. Comme j'allais utiliser ces machines dans d'autres contextes qui impliquaient des changements de DNS, j'ai préféré renseigner le nom du serveur WAPT dans les fichiers hosts respectifs





Afin de faciliter la recherche dans les fichiers logs, le système du serveur peut être configuré en Anglais

```
root@srvwapt:~# localectl set-locale LANG=en_US.UTF-8
root@srvwapt:~# localectl status
   System Locale: LANG=en_US.UTF-8
       VC Keymap: n/a
      X11 Layout: fr
       X11 Model: pc105
     X11 Variant: latin9
root@srvwapt:~#
```

Il est recommandé de bien configurer le service NTP pour assurer que le serveur soit à l'heure

```
root@srvwapt:/home/adminstagiaire# systemctl enable ntp
Synchronizing state of ntp.service with SysV service script with /lib/systemd/systemd-s
ysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ntp
root@srvwapt:/home/adminstagiaire#
```

Après ces étapes préalables peut commencer l'installation même du logiciel WAPT.

Les étapes d'installation sont les mêmes pour la version Entreprise et Community, mais les dépôts renseignés diffèrent. La version Entreprise doit être téléchargée d'un dépôt sécurisé accessible avec un login et un mot de passe fourni par les éditeurs.

Il est recommandé de mettre à jour le système et la liste de sources par

```
root@srvwapt:~# apt update && apt upgrade
```

L'installation de WAPT nécessite l'installation de gestionnaires de transport de téléchargement https et de gestionnaires de chiffrement et signature de données

```
root@srvwapt: /home/adminstagiaire
su: Authentication failure
adminstagiaire@srvwapt:~$ sudo su
[sudo] password for adminstagiaire:
root@srvwapt:/home/adminstagiaire# apt install apt-transport-https lsb-release g
nupg
```

Et des autorités de certificats, par exemple, celles fournies par Mozilla

```
root@srvwapt:~# apt install ca-certificates
```

Pour pouvoir accéder aux dépôts WAPT, il faut installer leurs clés respectives ; dans le cas de cette installation, la clé est la même pour les dépôts privés et entreprise, ainsi que pour les versions Debian et Ubuntu

```
root@srvwapt:/home/adminstagiaire# wget -O - https://wapt.tranquil.it/debian/tis
wapt-pub.gpg  | apt-key add -
```

Ensuite, l'adresse de dépôt doit être renseignée dans un fichier sources.list à part, crée à cet effet. L'adresse ci-dessous contient le login et mot de passe fournis par les éditeurs pour pouvoir accéder à la licence d'essai de la version Entreprise (login : motdepasse@srvwapt-pro.tranquilit/…)

```
adminstagiaire@srvwapt:~$ echo "deb https://51d537a1-dfbd-4a3a-87bb-c3381114be13
:rJ6kKP5FEUFy3OSP@srvwapt-pro.tranquil.it/entreprise/ubuntu/wapt-1.8/ $(lsb_rele
ase -c -s) main" > /etc/apt/sources.list.d/wapt.list
```

Adresse version Community

```
root@srvwapt:~# echo "deb https://wapt.tranquil.it/debian/wapt-1.8/ $(lsb_release -
c -s) main" > /etc/apt/sources.list.d/wapt.list
```

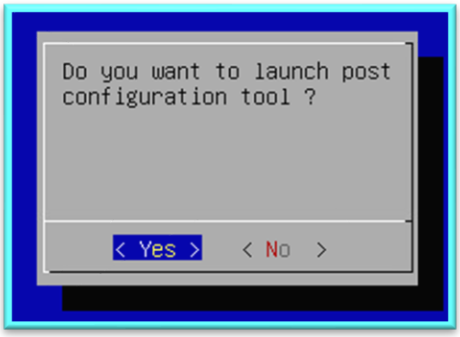
Les paquets WAPT sont installés par cette commande

```
root@srvwapt:~# apt update && apt install tis-waptserver tis-waptsetup_
```

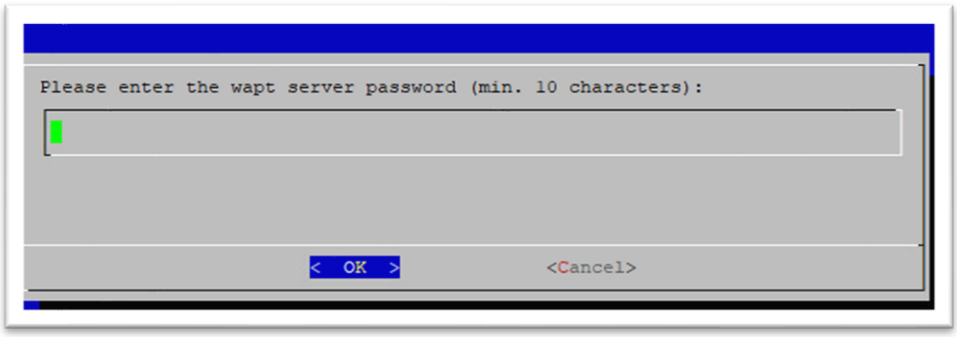
Post-configuration

La configuration du serveur doit être lancée en exécutant le script suivant

```
root@srvwapt:~# /opt/wapt/waptserver/scripts/postconf.sh
```



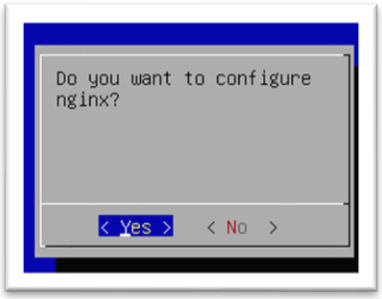
Il faut choisir le mot de passe pour le compte d'administrateur du serveur WAPT



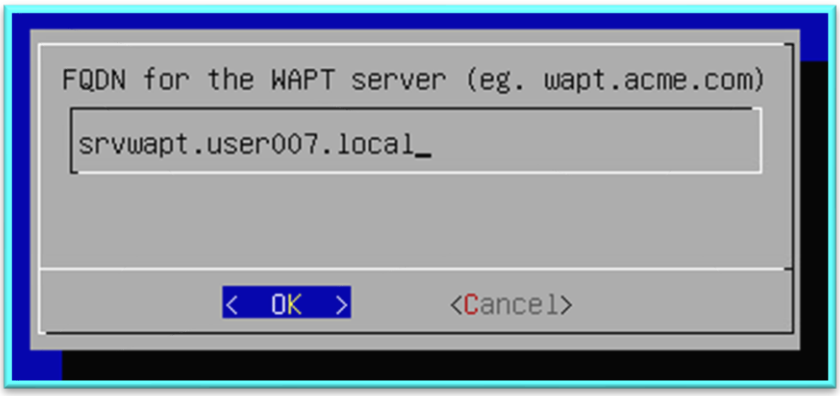
Puis choisir le mode d'authentification pour enregistrer les machines clients :

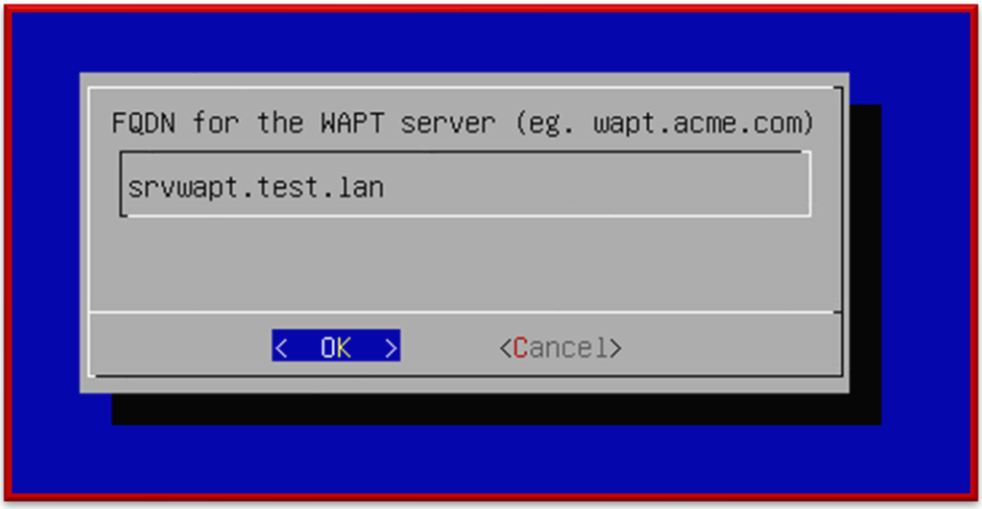
- Sans authentification
- Enregistrement initial basé sur Kerberos
- Demande de mot de passe à chaque nouvel enregistrement

L'étape suivante concerne la configuration du serveur NGinX



Renseigner le FQDN du serveur WAPT





Le serveur WAPT est alors prêt à l'emploi.

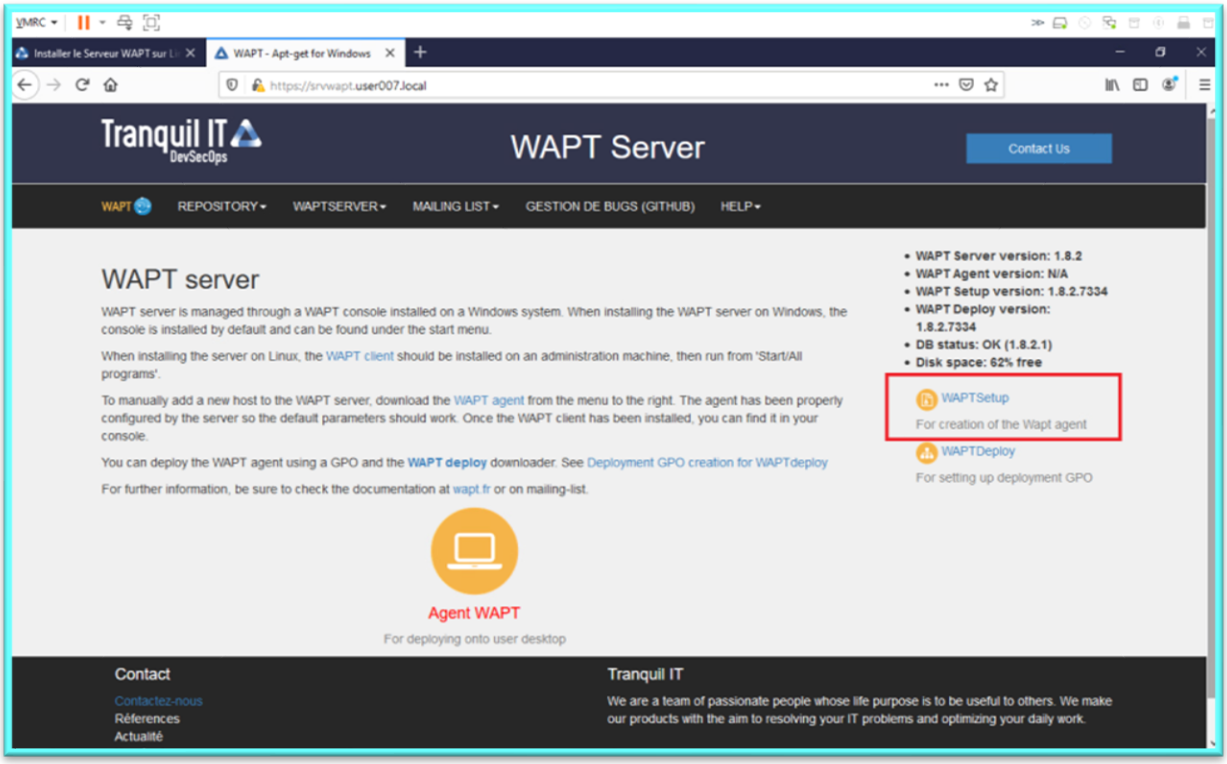
```
[*] Postconfiguration completed.
Please connect to https://srvwapt.user007.local / to access the server.
```

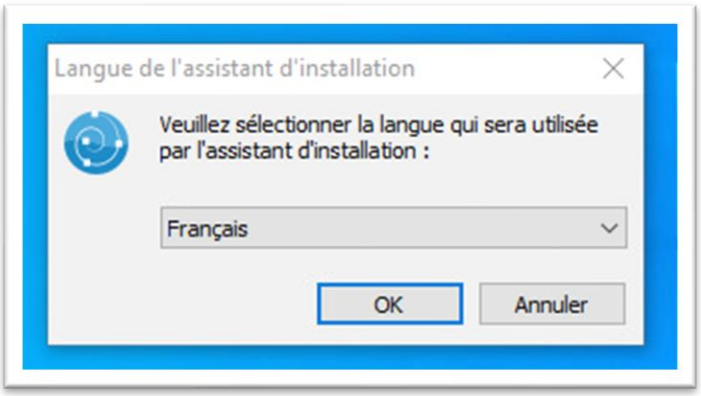
```
[*] Postconfiguration completed.
Please connect to https://srvwapt.test.lan/ to access the server.
```

Installation de la console de gestion

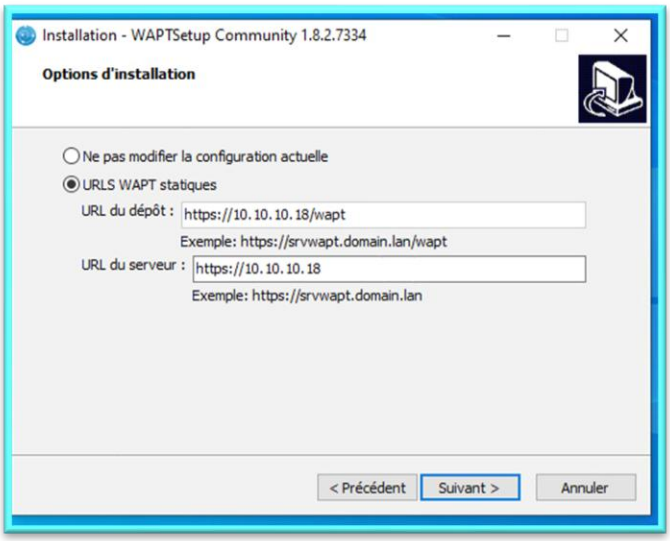
La console de gestion du serveur ne doit pas être installée sur le serveur même, mais sur une machine utilisée pour la gestion du réseau. J'ai utilisé une machine Windows 10 à cet effet.

Le fichier d'installation « WAPTsetup » doit être téléchargée sur le site du serveur, en y accédant par un navigateur web :

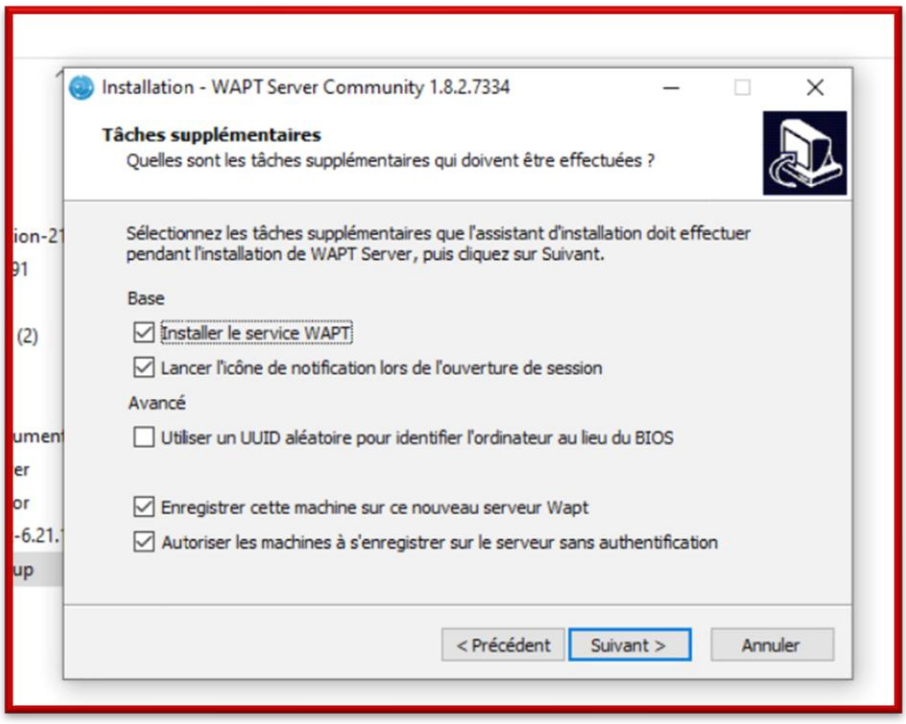




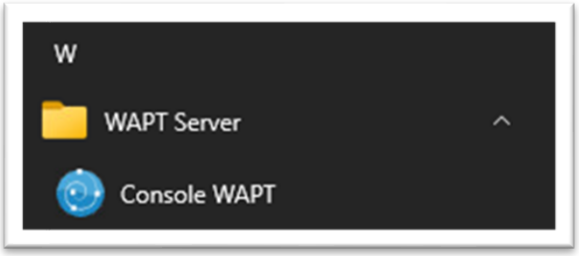
L'adresse IP ou le nom d'hôte de serveur WAPT doivent être renseignés lors de l'installation de l'agent, afin de connecter la machine d'administration :

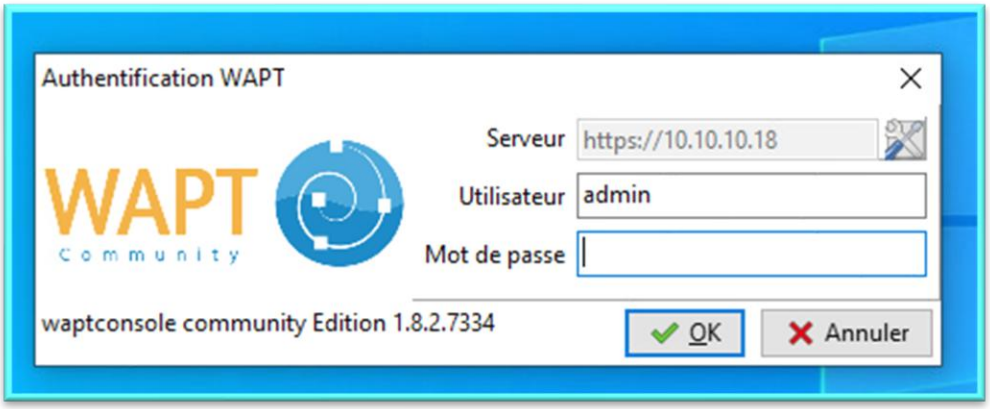


Des options supplémentaires peuvent être choisies à la fin de l'installation

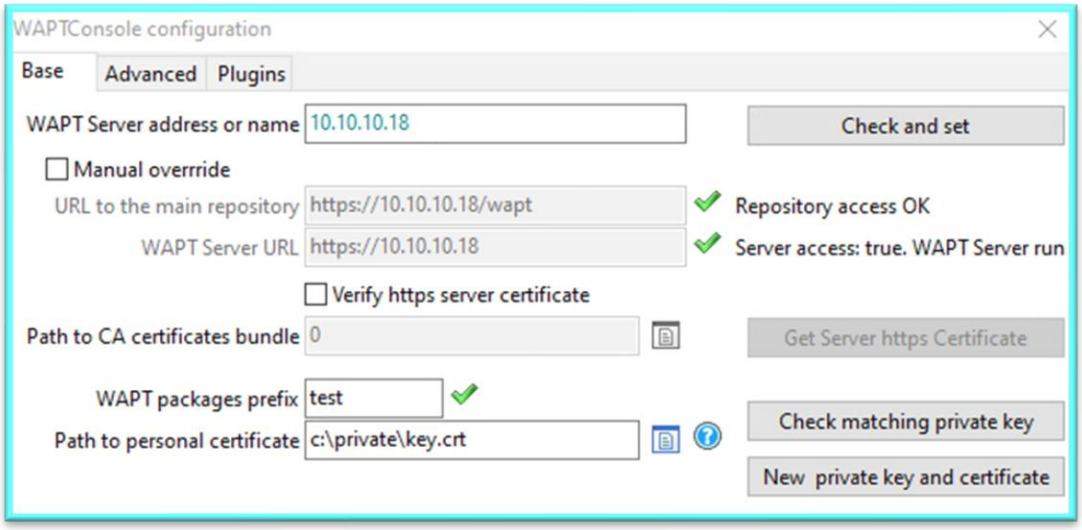


La console d'administration devient alors accessible par le menu des programmes



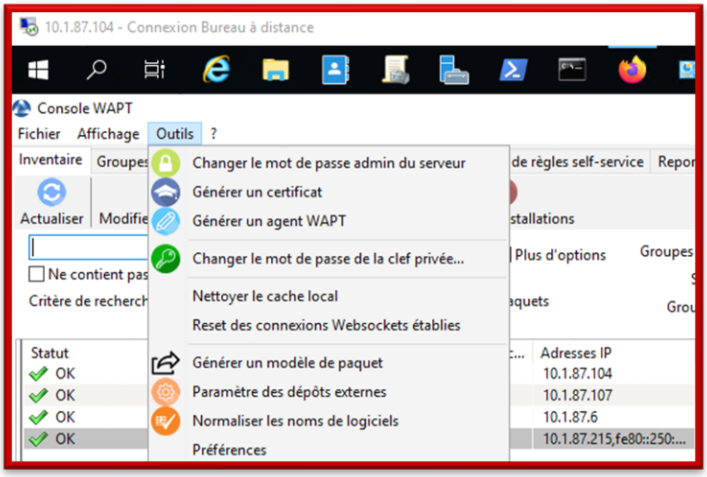


Le bouton de configuration  ouvre une boite de dialogue qui permet de générer le certificat public et la clé, de renseigner le chemin vers ces derniers, et de choisir un préfixe pour les paquets logiciels. J'ai opté pour « test »



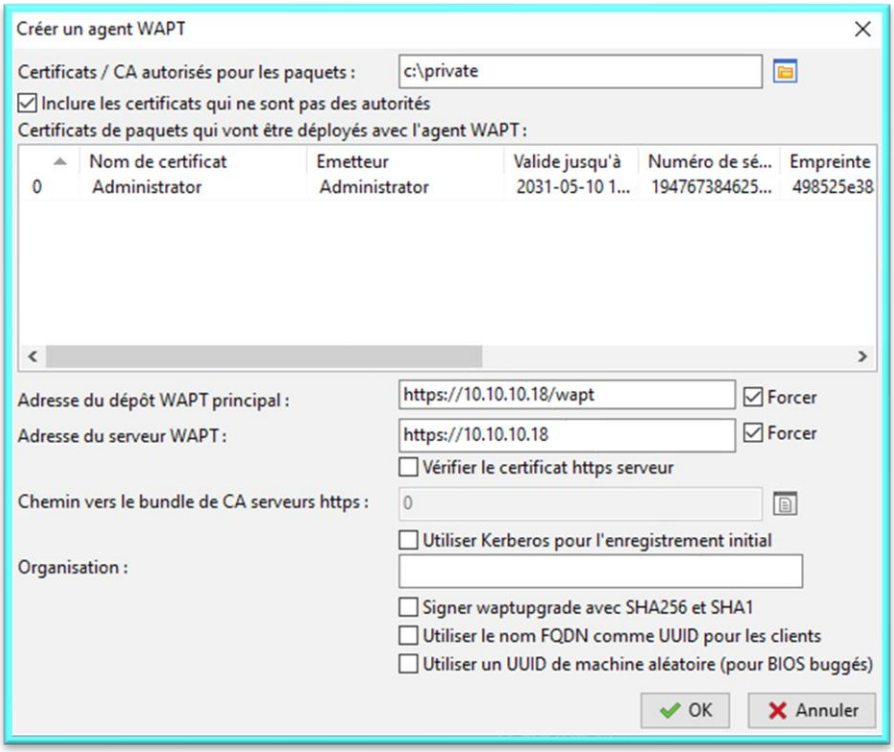
Création et déploiement de l'agent pour clients Windows

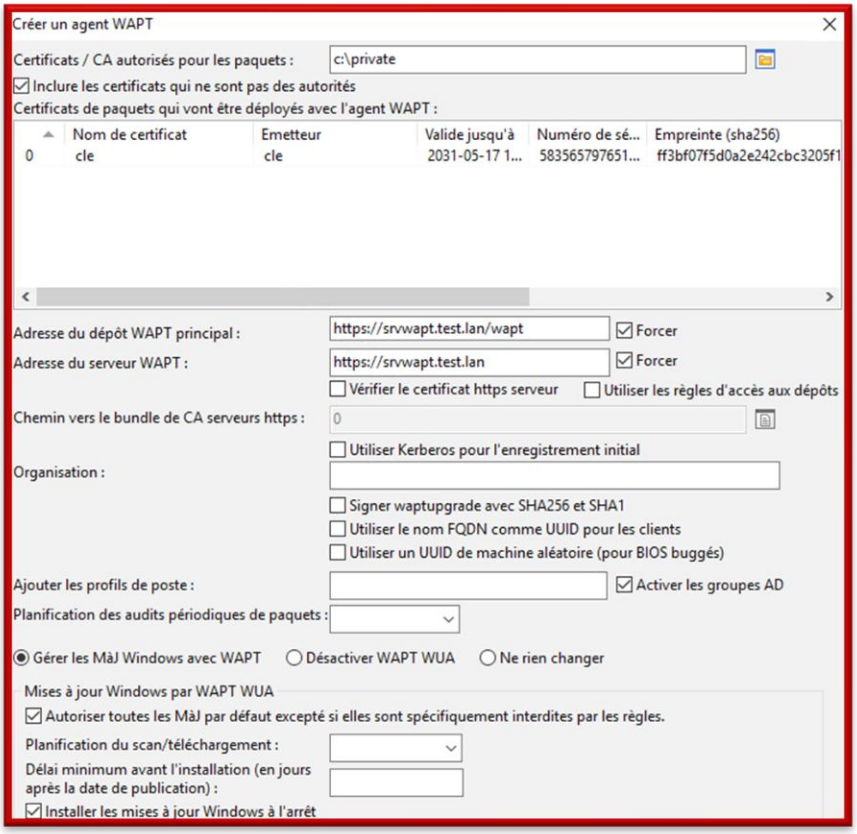
L'installeur de l'agent peut être créé à partir des outils de la console de gestion



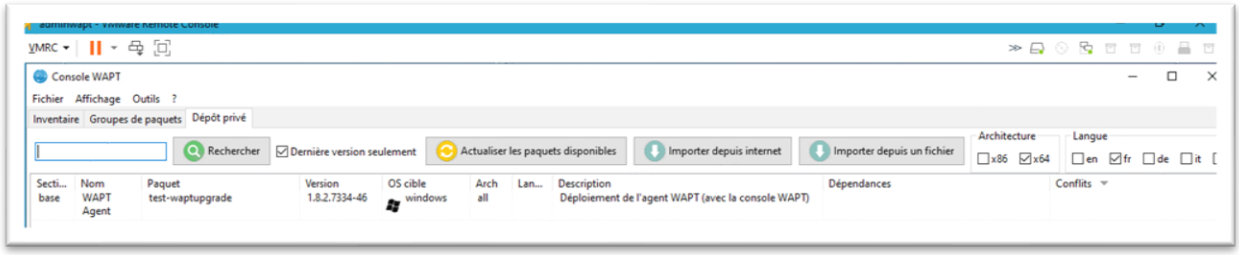
Il faut alors renseigner obligatoirement le certificat public et l'adresse du serveur et dépôt de logiciels. Les autres options sont facultatives. Pour faciliter les tests, je n'ai pas indiqué la vérification des certificats HTTPS, et comme, lors de l'installation du serveur, je n'avais pas indiqué de méthode d'authentification pour les clients, ces options n'étaient pas nécessaires non plus.

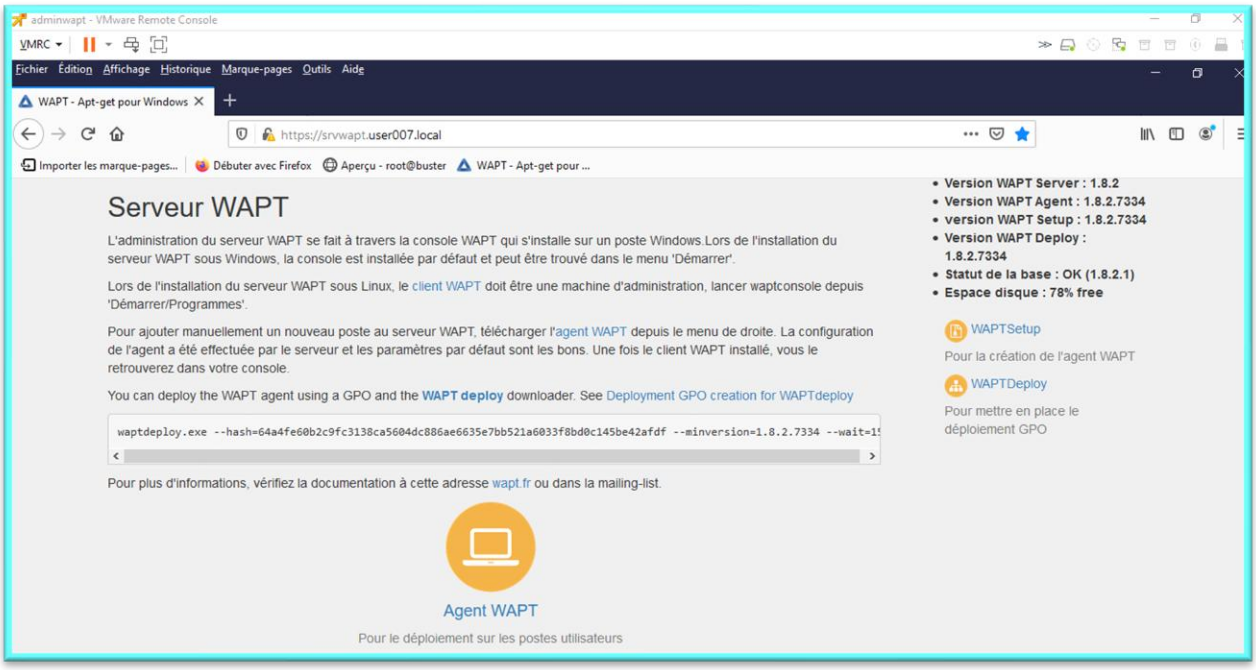
La version Community offrant moins de fonctionnalités que la version Entreprise, les installeurs ne contiennent pas les mêmes options, comme montrent les captures d'écran





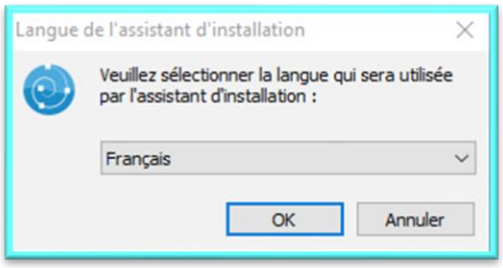
Une fois la compilation terminée, l'agent est disponible dans le dépôt logiciel et peut être téléchargé à partir du site du serveur WAPT, comme la console de gestion auparavant

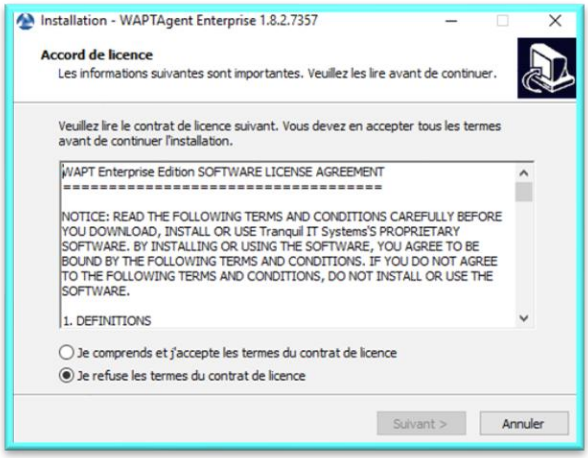


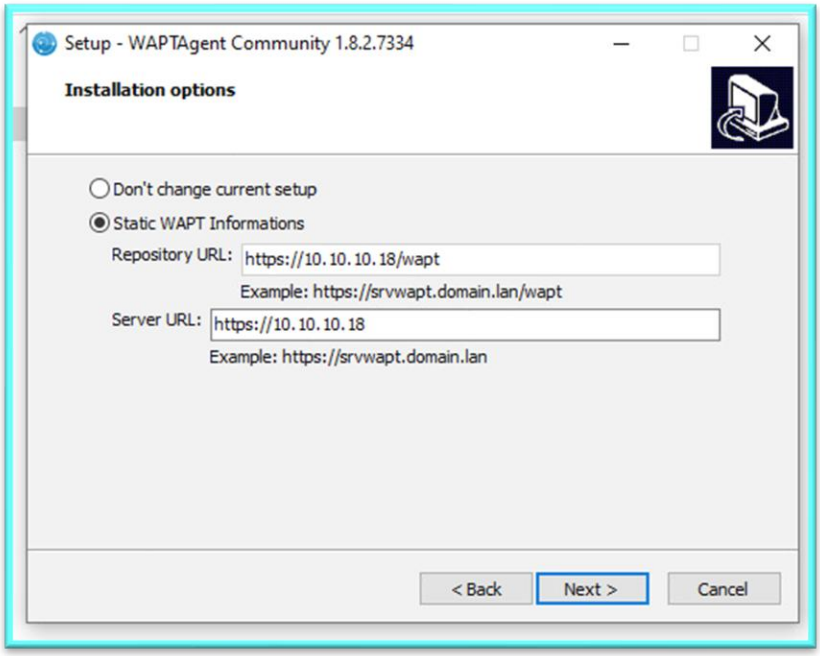


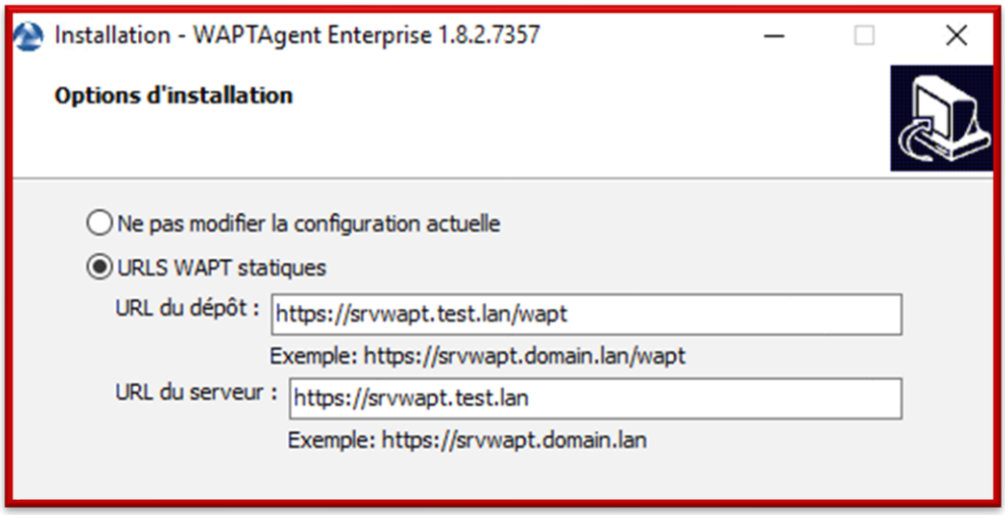
Il est également possible de déployer l'agent par GPO. N'utilisant que 2 clients, je n'ai pas testé cette solution.

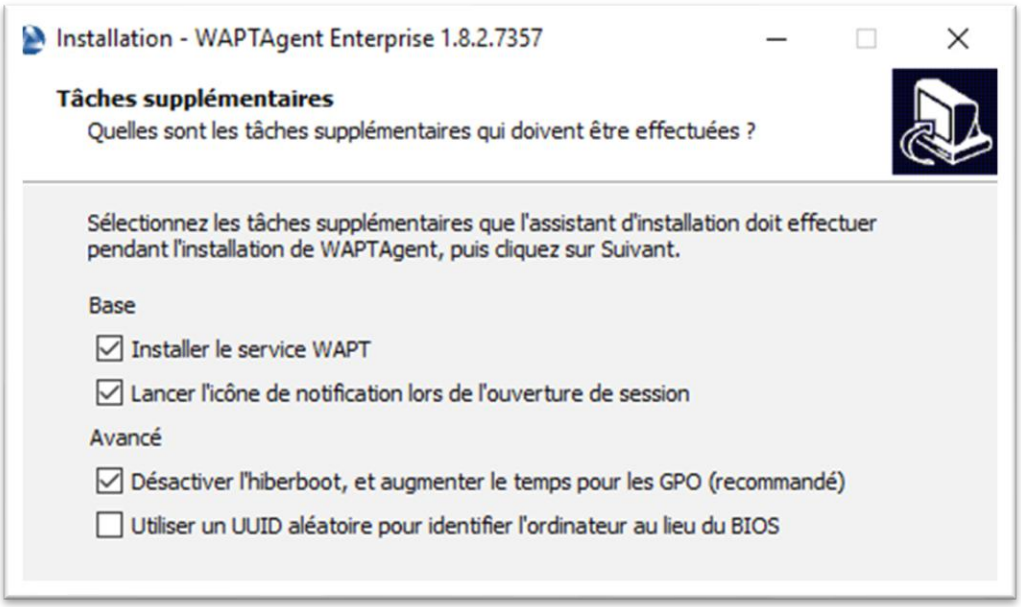
L'installation manuelle a été facile et s'est faite sans problèmes.



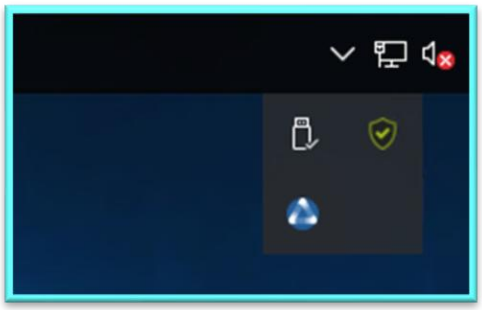



Setup - WAPTAgent Community 1.8.2.7334
Installation options
Don't change current setup
Static WAPT Informations
Repository URL: https://10.10.10.18/wapt
Example: https://srvwapt.domain.lan/wapt
Server URL: https://10.10.10.18
Example: https://srvwapt.domain.lan
< Back
Next >
Cancel

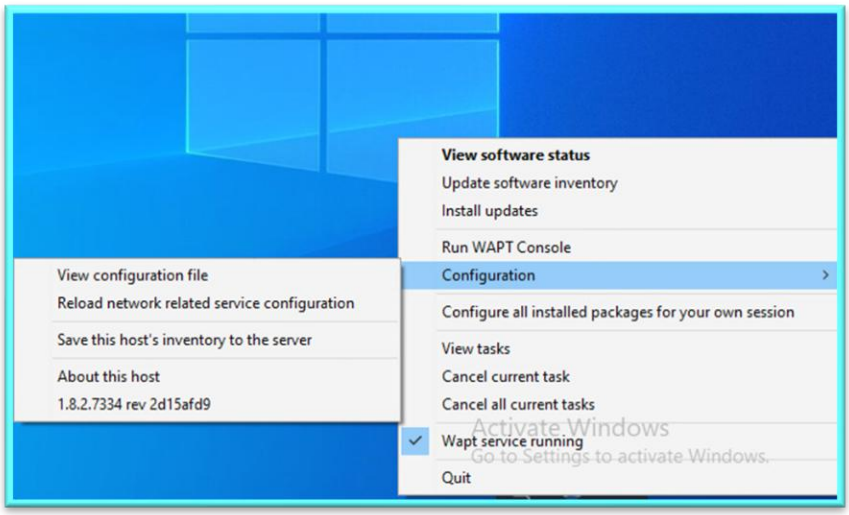

Installation - WAPTAgent Enterprise 1.8.2.7357
Options d'installation
Ne pas modifier la configuration actuelle
URLS WAPT statiques
URL du dépôt : https://srvwapt.test.lan/wapt
Exemple: https://srvwapt.domain.lan/wapt
URL du serveur : https://srvwapt.test.lan
Exemple: https://srvwapt.domain.lan


Installation - WAPTAgent Enterprise 1.8.2.7357
Tâches supplémentaires
Quelles sont les tâches supplémentaires qui doivent être effectuées ?
Sélectionnez les tâches supplémentaires que l'assistant d'installation doit effectuer pendant l'installation de WAPTAgent, puis cliquez sur Suivant.
Base
Installer le service WAPT
Lancer l'icône de notification lors de l'ouverture de session
Avancé
Désactiver l'hiberboot, et augmenter le temps pour les GPO (recommandé)
Utiliser un UUID aléatoire pour identifier l'ordinateur au lieu du BIOS

Une fois l'installation terminée, l'icône de notification WAPT devient visible sur la machine client si cette option a été choisie

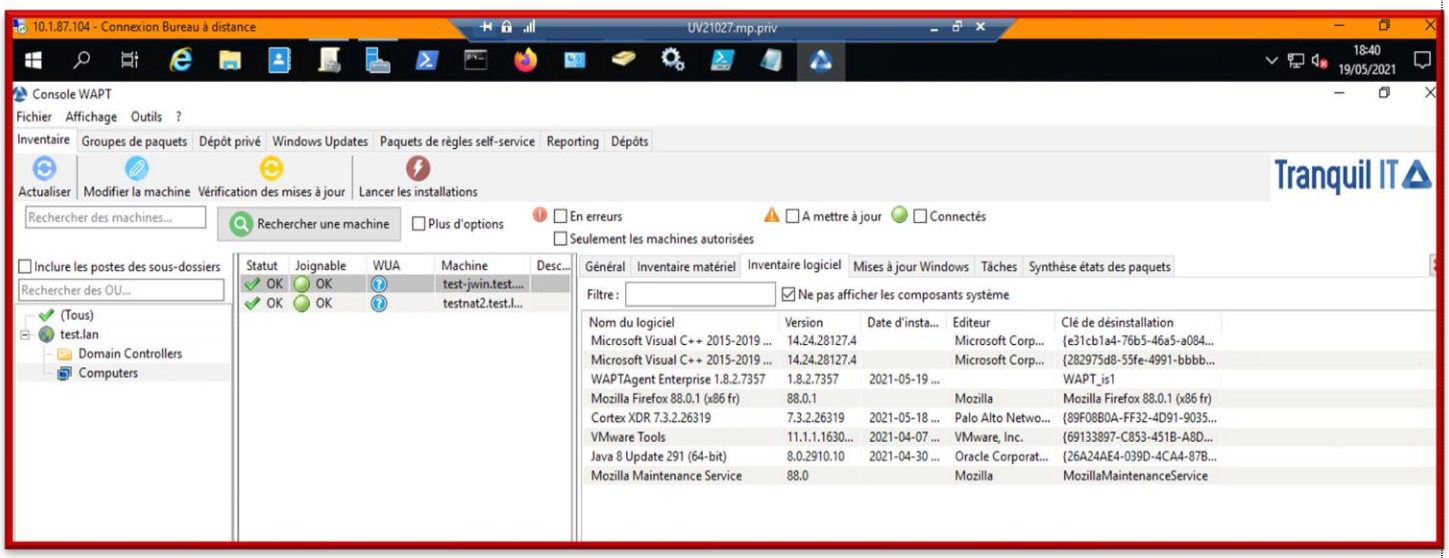


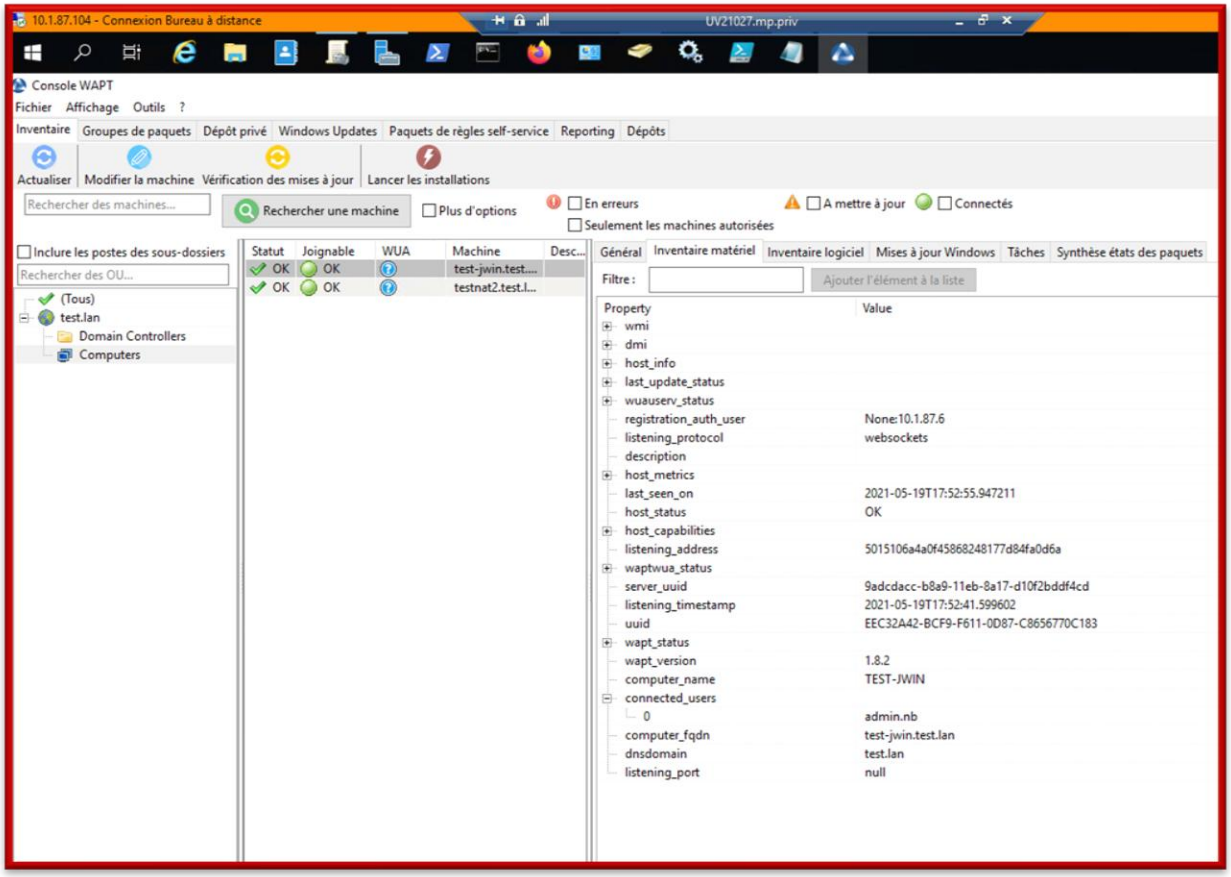
Un clic droit sur l'icône permet de sauvegarder l'inventaire du client sur le serveur



La console de gestion permet alors l'aperçu de l'inventaire matériel et logiciel du client

La version Entreprise affiche également l'arborescence Active Directory des clients



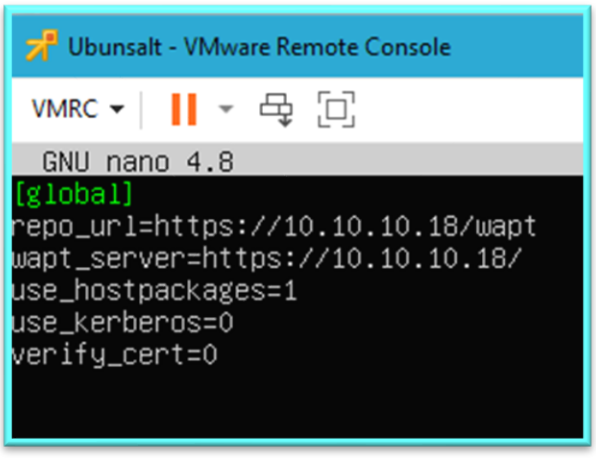


Création et déploiement de l'agent sur machines Linux (Debian Buster et Ubuntu 20.4)

Les étapes d'installation sont les mêmes que pour la création du serveur, à l'exception de la dernière commande qui n'installe que le paquet agent

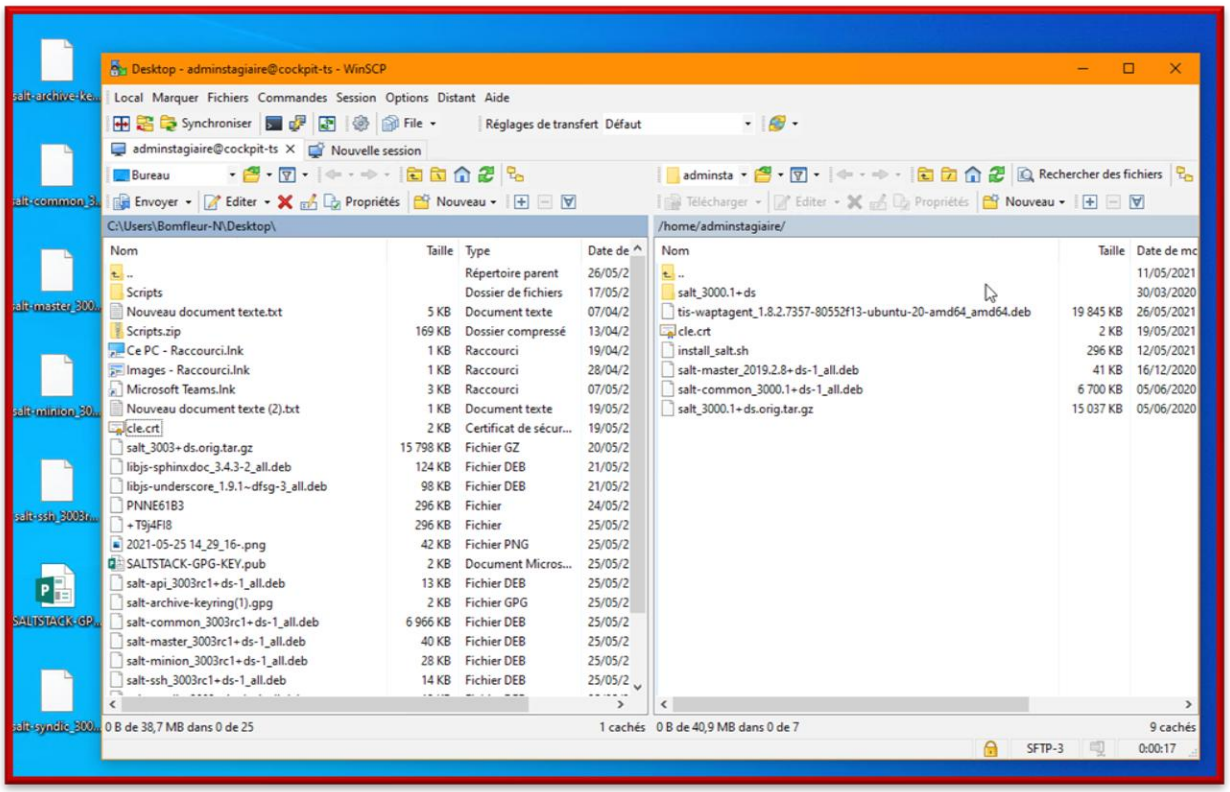
```
root@ubunsalt:/home/user# apt update && apt install tis-waptagent_
```

Après l'installation du paquet, il faut créer le fichier de configuration /opt/wapt/wapt-get.ini d'après le modèle défaut



Comme je n'utilisais pas de certificat NginX SSL/TLS, j'ai laissé la ligne verify_cert à 0.

Le certificat de signature des paquets créé sur le serveur doit être copié sur la machine client. J'ai utilisé WinSCP pour le copier de la machine d'administration Windows sur le client Linux.

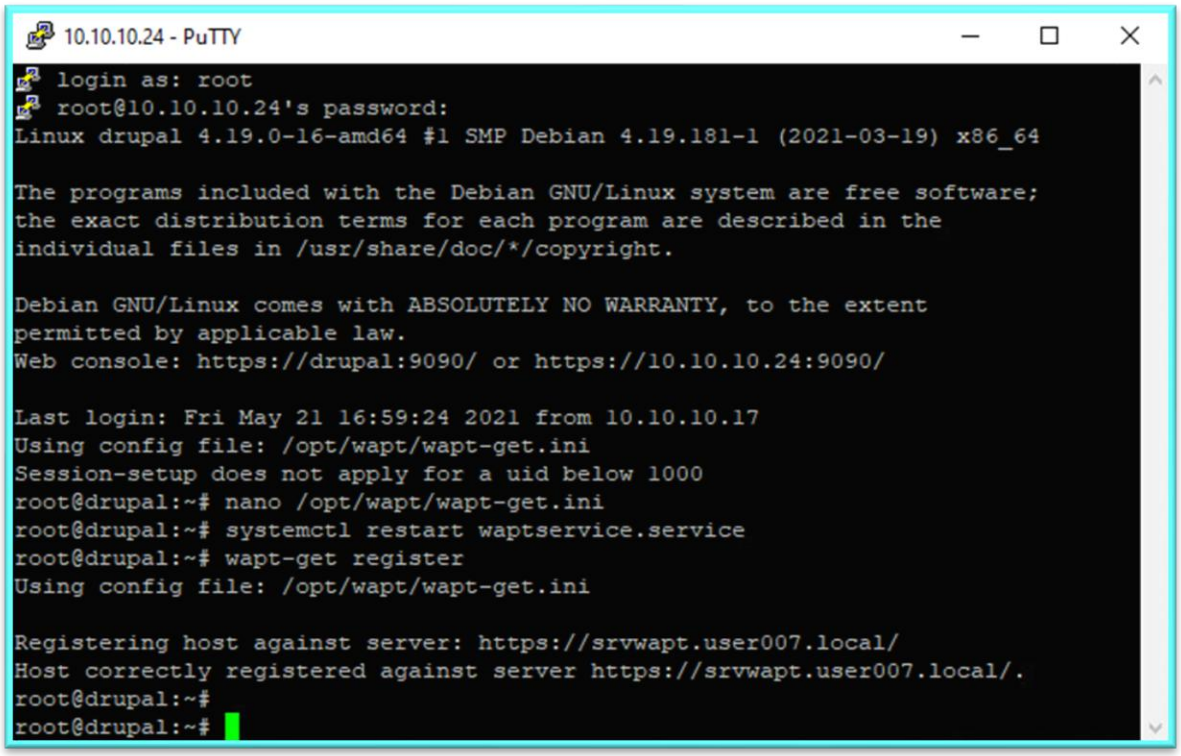


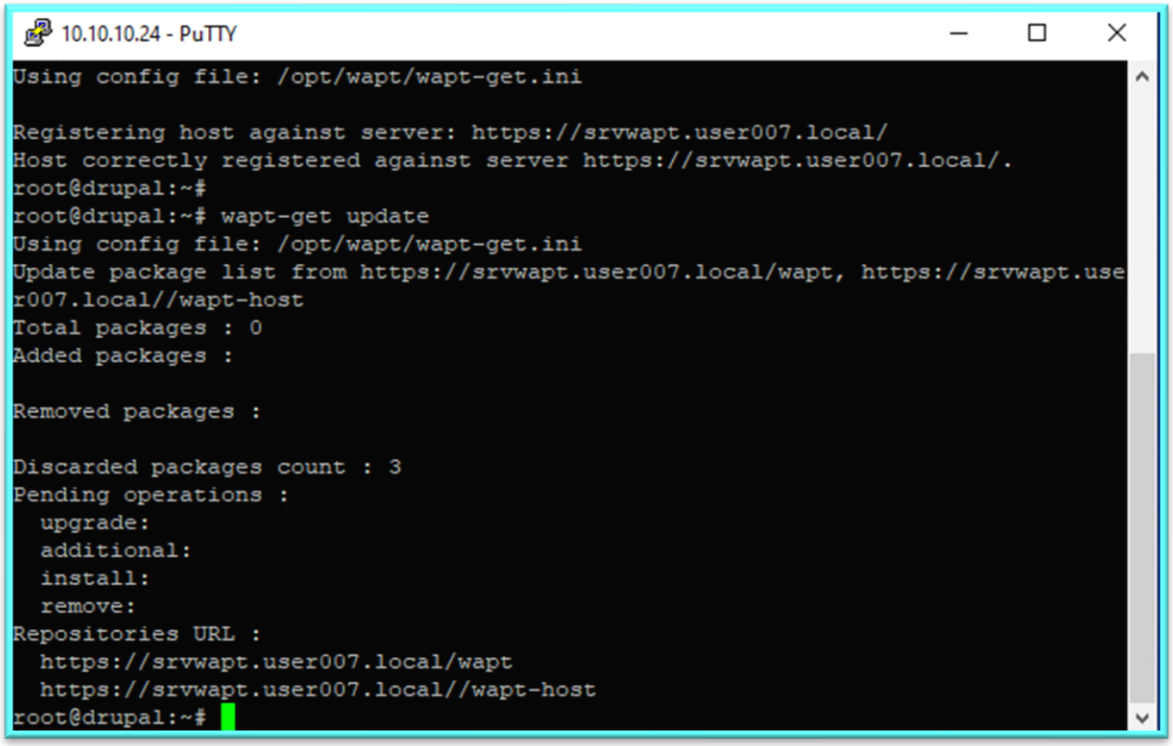
Le service WAPT sera ensuite redémarré

```
root@ubunsalt:/home/user# systemctl restart waptservice
```

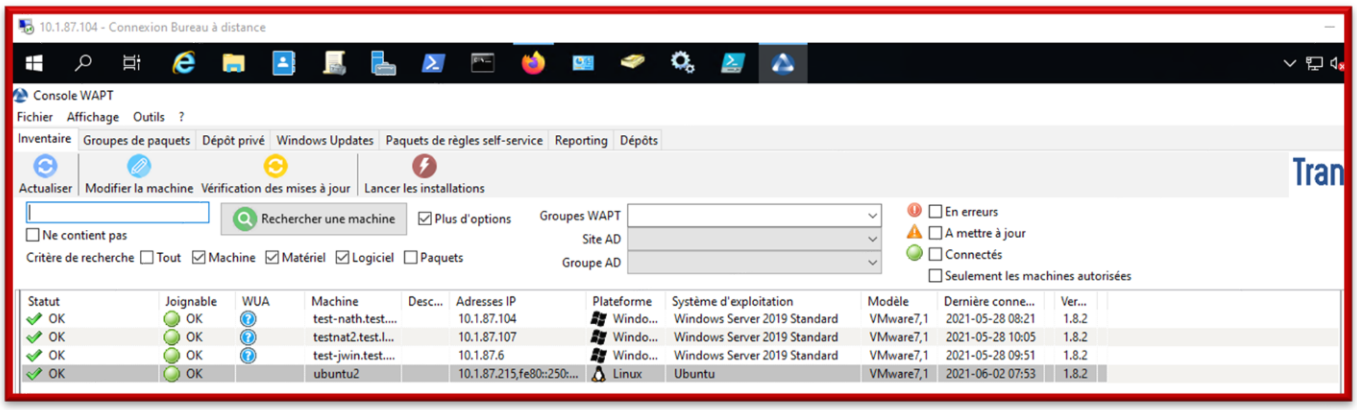
et l'enregistrement de la machine lancé par

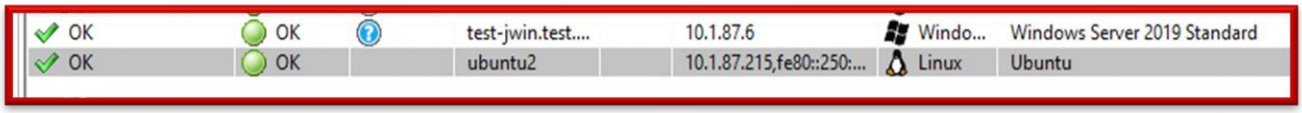
```
See 'snap info q' for additional versions.

root@ubuntu2:/home/adminstagiaire# wapt-get register
Using config file: /opt/wapt/wapt-get.ini
Registering host against server: https://srvwapt.test.lan/
Host correctly registered against server https://srvwapt.test.lan/.
root@ubuntu2:/home/adminstagiaire# wapt-get update
Using config file: /opt/wapt/wapt-get.ini
Update package list from https://srvwapt.test.lan/wapt, https://srvwapt.test.lan//w
host
Total packages : 2
Added packages :

Removed packages :

Discarded packages count : 4
Pending operations :
  upgrade:
  additional:
  install:
  remove:
Repositories URL :
  https://srvwapt.test.lan/wapt
  https://srvwapt.test.lan//wapt-host
root@ubuntu2:/home/adminstagiaire#
```

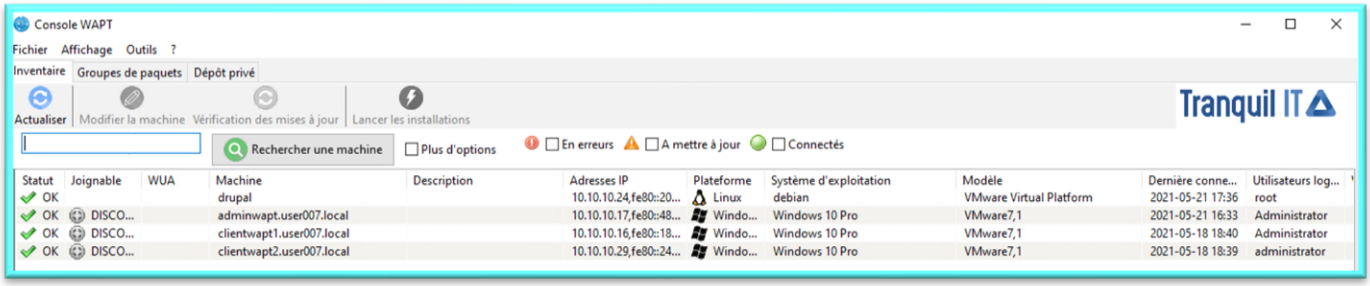

10.10.10.24 - PuTTY
login as: root
root@10.10.10.24's password:
Linux drupal 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Web console: https://drupal:9090/ or https://10.10.10.24:9090/
Last login: Fri May 21 16:59:24 2021 from 10.10.10.17
Using config file: /opt/wapt/wapt-get.ini
Session-setup does not apply for a uid below 1000
root@drupal:~# nano /opt/wapt/wapt-get.ini
root@drupal:~# systemctl restart waptservice.service
root@drupal:~# wapt-get register
Using config file: /opt/wapt/wapt-get.ini
Registering host against server: https://srvwapt.user007.local/
Host correctly registered against server https://srvwapt.user007.local/.
root@drupal:~#
root@drupal:~#

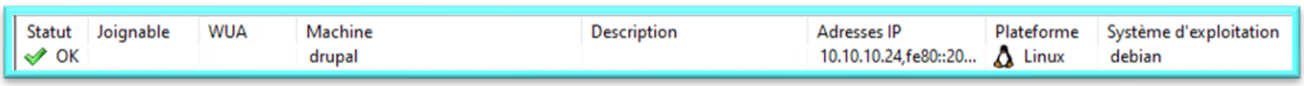

10.10.10.24 - PuTTY
Using config file: /opt/wapt/wapt-get.ini
Registering host against server: https://srvwapt.user007.local/
Host correctly registered against server https://srvwapt.user007.local/.
root@drupal:~#
root@drupal:~# wapt-get update
Using config file: /opt/wapt/wapt-get.ini
Update package list from https://srvwapt.user007.local/wapt, https://srvwapt.use
r007.local//wapt-host
Total packages : 0
Added packages :
Removed packages :
Discarded packages count : 3
Pending operations :
upgrade:
additional:
install:
remove:
Repositories URL :
https://srvwapt.user007.local/wapt
https://srvwapt.user007.local//wapt-host
root@drupal:~#

La machine devient alors visible dans la console de gestion donne un aperçu de l'inventaire matériel et logiciel des machines Linux



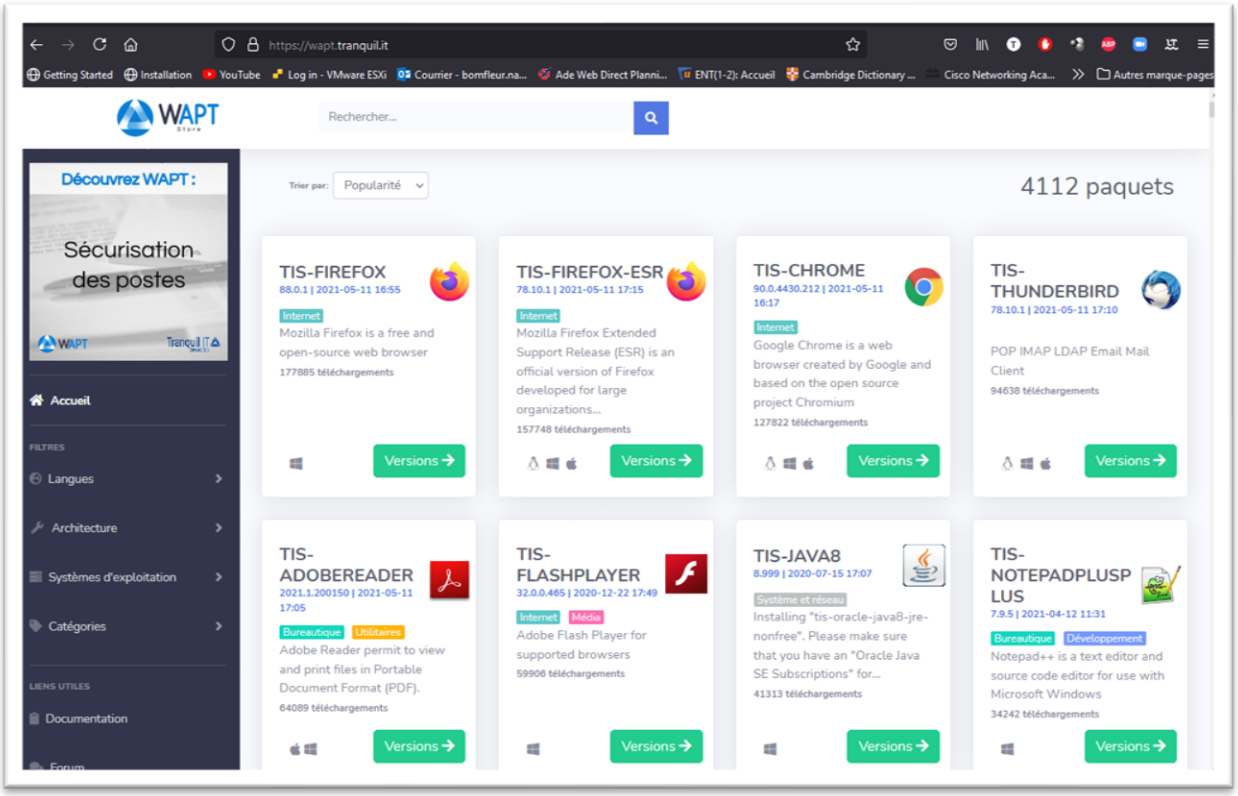


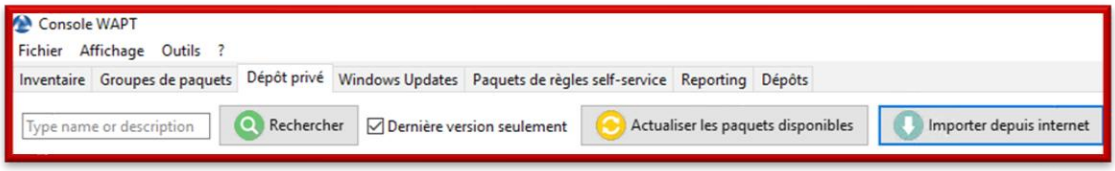




Déploiement de logiciels sur clients Windows

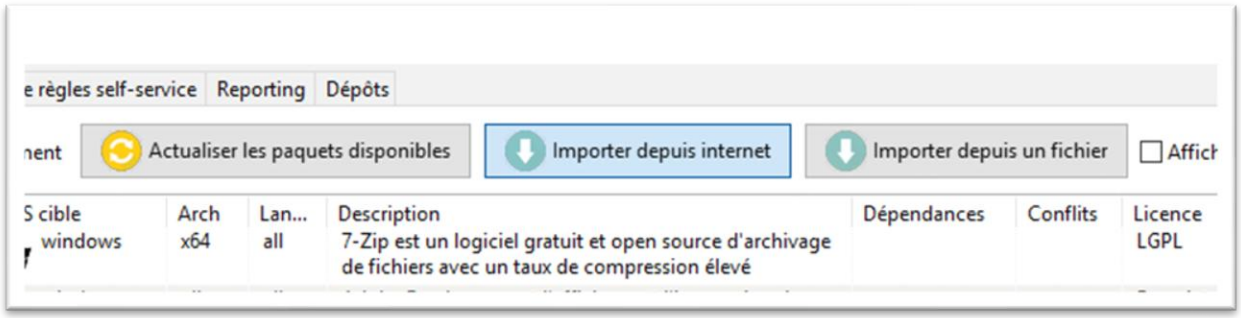
WAPT offre de nombreuses fonctionnalités pour créer et customiser des paquets logiciels à partir de fichiers .msi et .exe. Mais il existe déjà de nombreux paquets « prêts à l'emploi » sur le site dépôt de TranquilIT téléchargeables à partir de la console

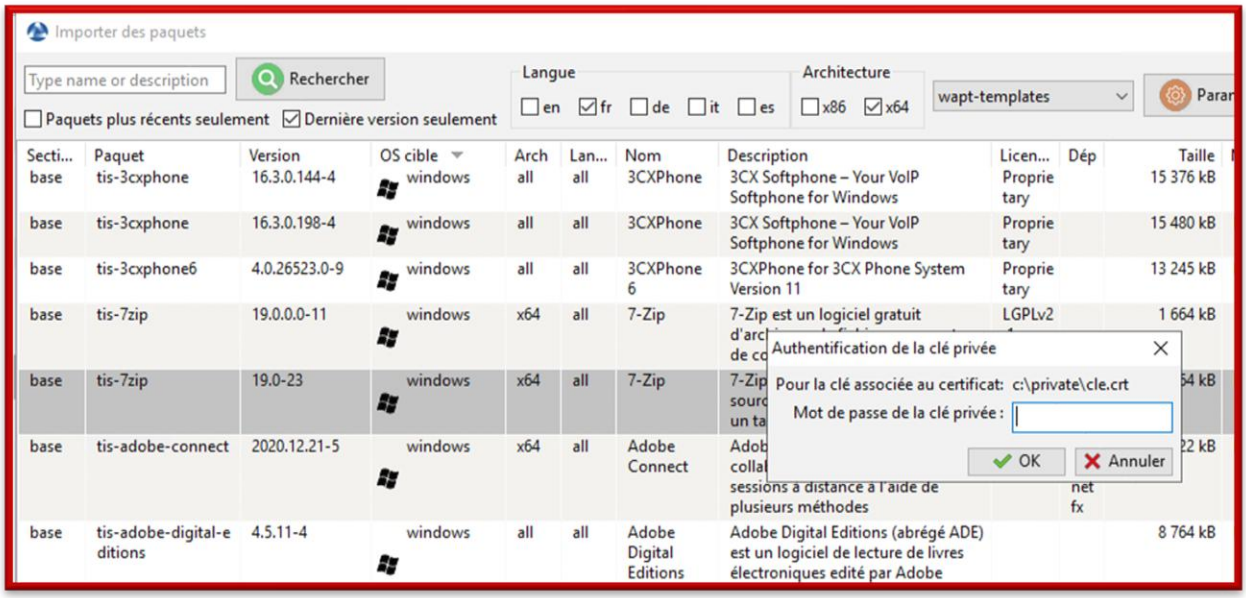




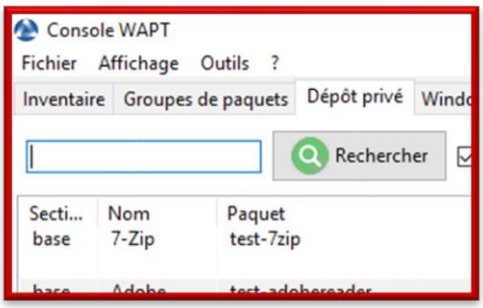
Pour les clients Windows de la mairie, j'ai testé le déploiement simultané de 7-Zip sur les deux machines appartenant à mon domaine « test.lan »

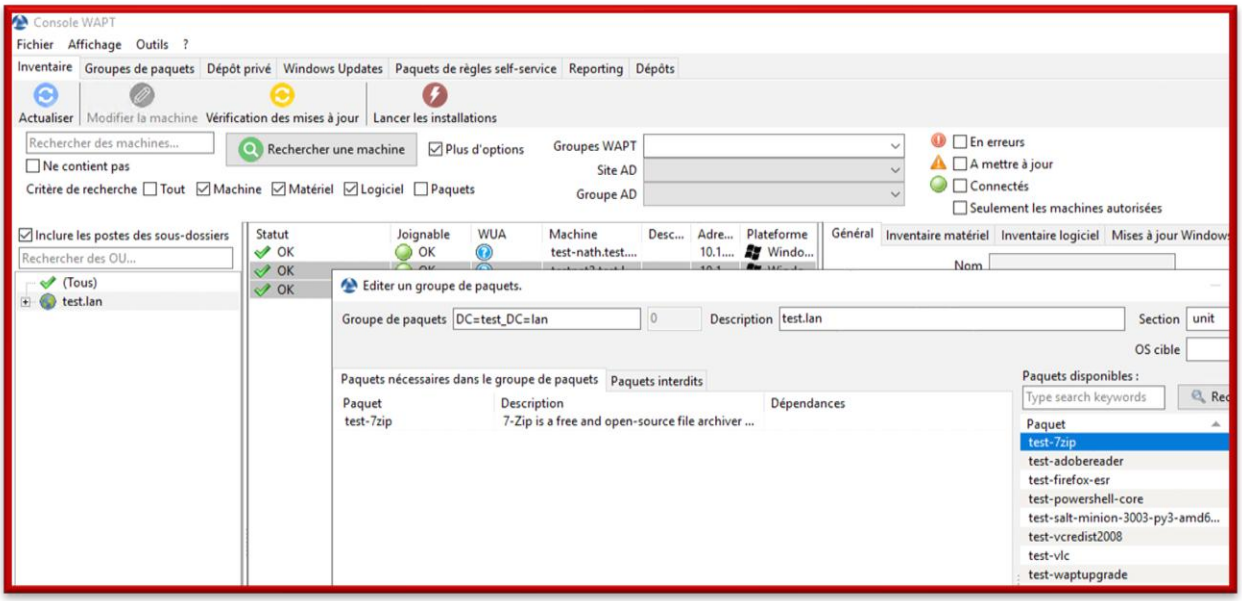
Dans un premier temps, le paquet doit être importé depuis internet et signé avec la clé du serveur WAPT



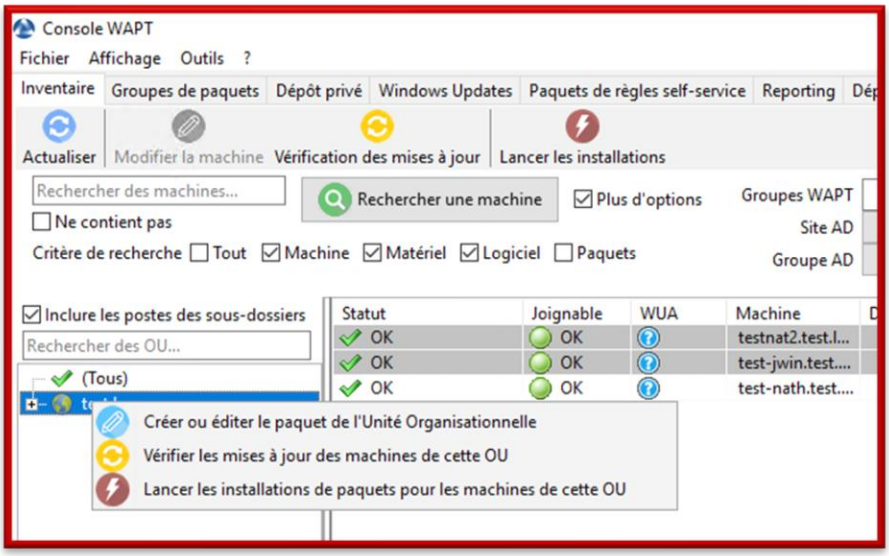


Le paquet devient alors disponible dans le dépôt privé, avec le préfixe paramétré lors de l'installation de la console de gestion

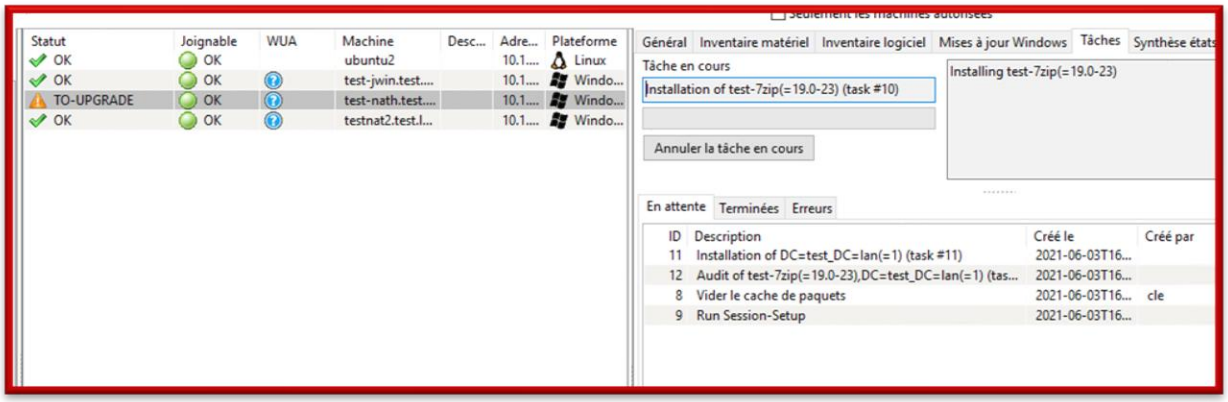




Par la suite, pour tester l'aspect centralisation de gestion, j'ai créé un paquet contenant le logiciel destiné aux machines appartenant au domaine (fonctionnalité réservée à la version entreprise)

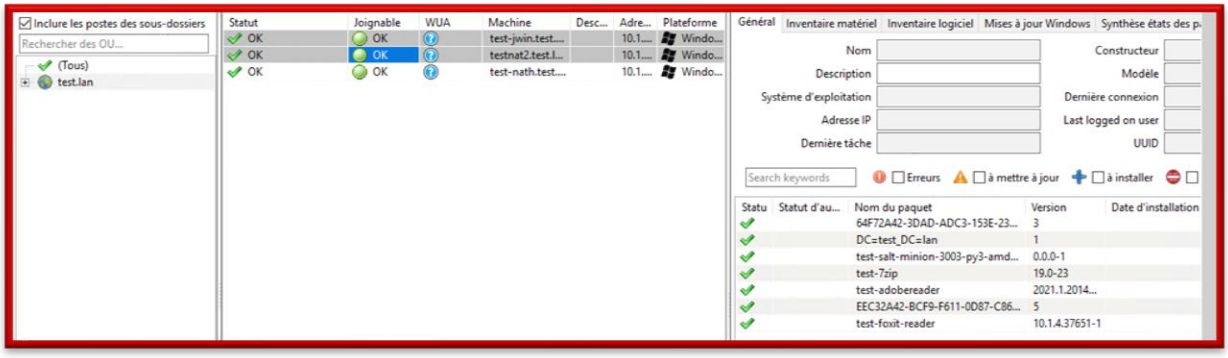


Sélectionner les machines concernées et cliquer « lancer les installations » démarre alors le procès d'installation. Les tâches effectuées sont visibles dans l'onglet « Tâches » de la console de gestion



Et l'icône de notification affiche l'activité sur l'hôte 

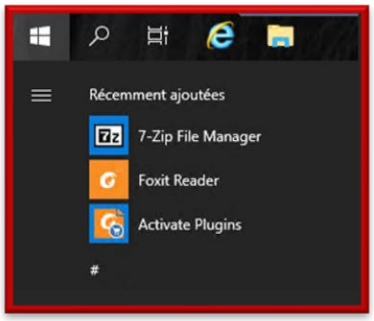
Après actualisation, le statut des paquets affiche que l'installation est réussie

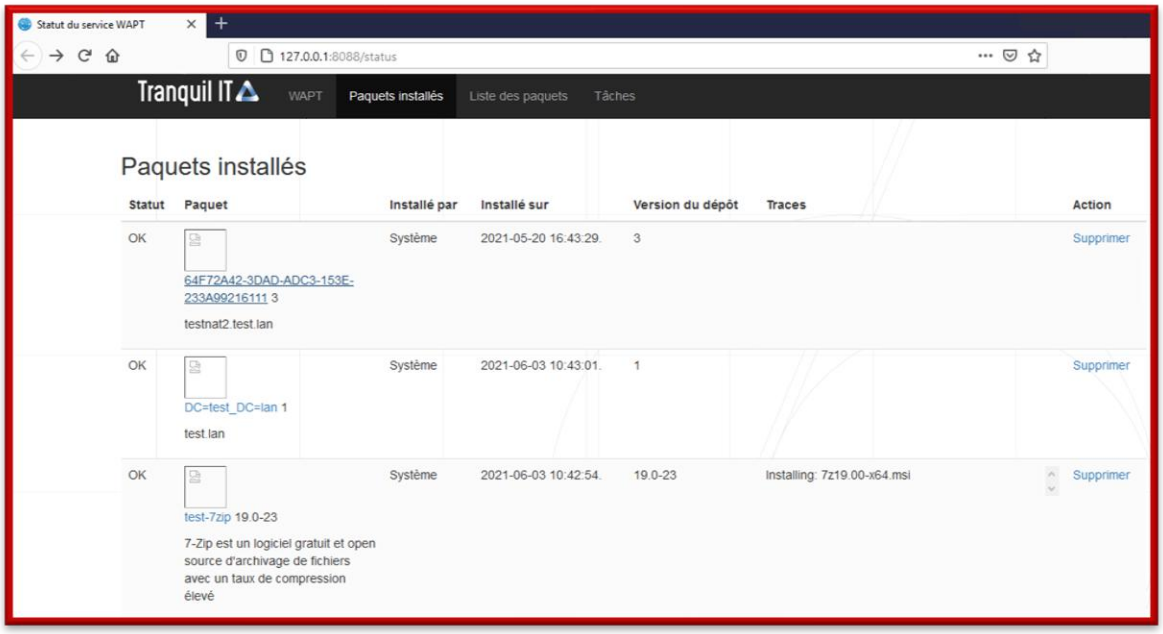


Statu	Statut d'au...	Nom du paquet	Version
✔		64F72A42-3DAD-ADC3-153E-23...	3
✔		DC=test_DC=lan	1
✔		test-salt-minion-3003-py3-amd...	0.0.0-1
✔		test-7zip	19.0-23

La nomenclature de WAPT peut prêter à confusion : « Paquet » désigne autant le paquet logiciel (test-vlc) que l'ensemble de paquets logiciels (ici :uniquement test-vlc) attribués au domaine (DC=test_DC=lan) et à la machine ciblée (UID 64F72…). Il s'agit donc en effet de trois éléments qui désignent, dans ce scénario précis, le logiciel VLC.

Le logiciel est bien disponible sur les deux machines cibles, et le statut des logiciels installés peut être visualisé grâce à l'agent WAPT





Cette page web permet également de désinstaller facilement les logiciels sur la machine concernée.

Déploiement de logiciels sur un client Linux

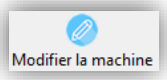
J'ai testé le déploiement sur deux distributions Linux : Ubuntu dans l'infrastructure de la mairie et Debian dans mon infrastructure privée.

Pour l'infrastructure de la mairie , j'ai déployé le paquet VLC sur une machine Ubuntu. Les étapes de déploiement étaient les mêmes que pour les déploiements de logiciels sur clients Windows :

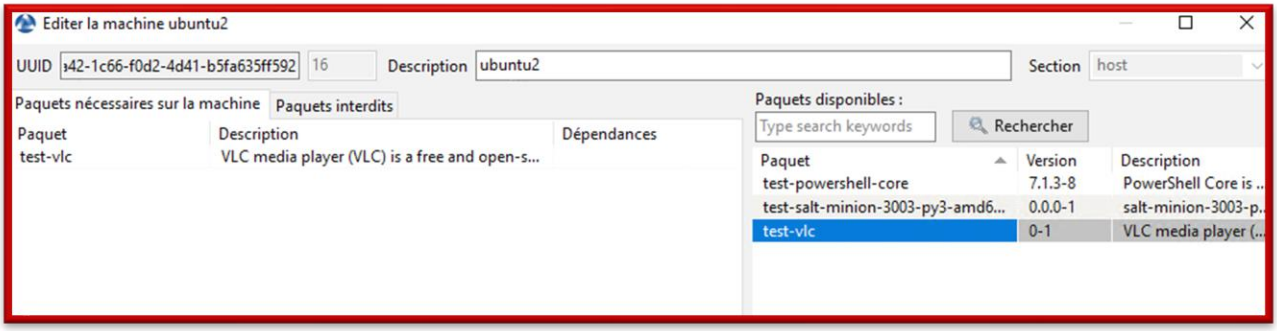
- téléchargement du paquet depuis le dépôt TranquilIIT

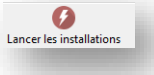
- signature avec la clé privée de l'administrateur du serveur WAPT

- attribution du paquet à une machine , un groupe ou une autre unité d'organisation

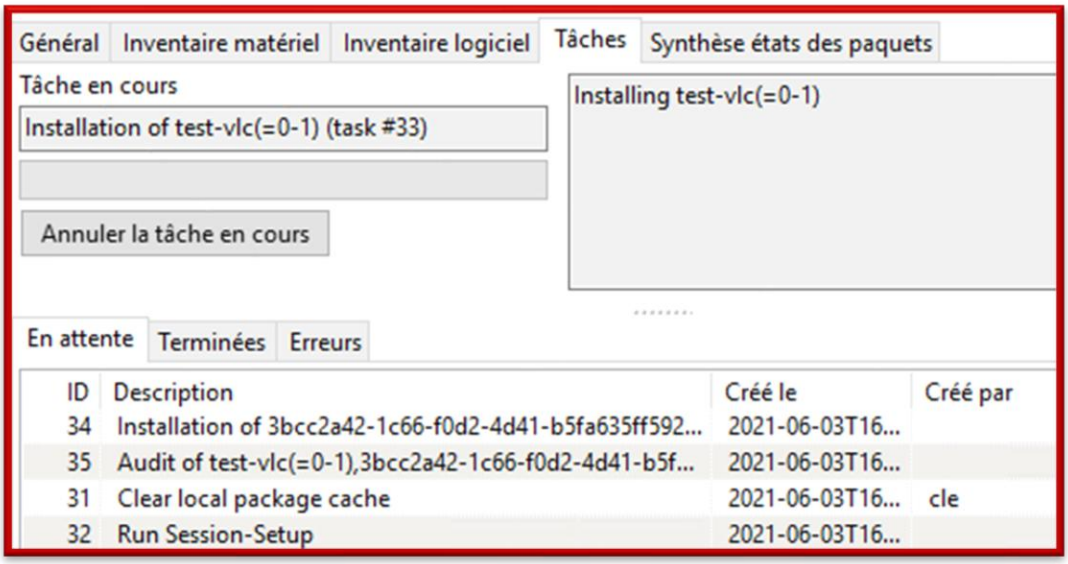
Pour la machine Ubuntu, j'ai utilisé le bouton 

qui ouvre la boîte de dialogue suivante



On lance les installations avec 

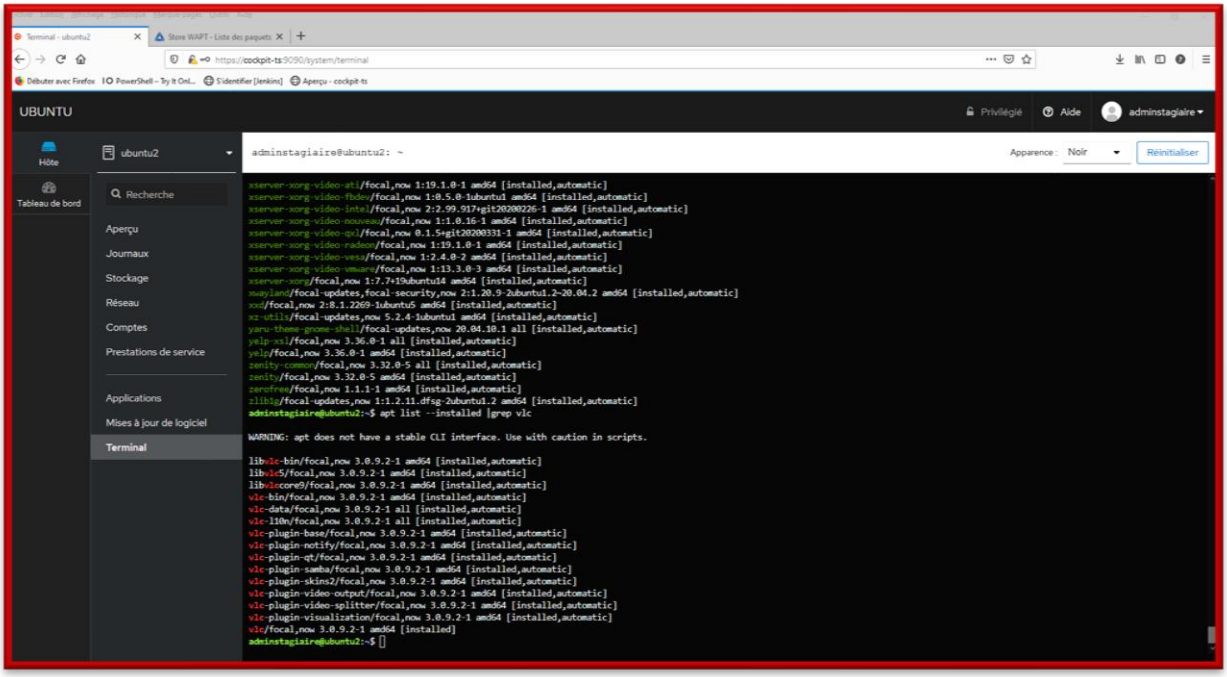
, et les processus deviennent visibles dans l'onglet « tâches »

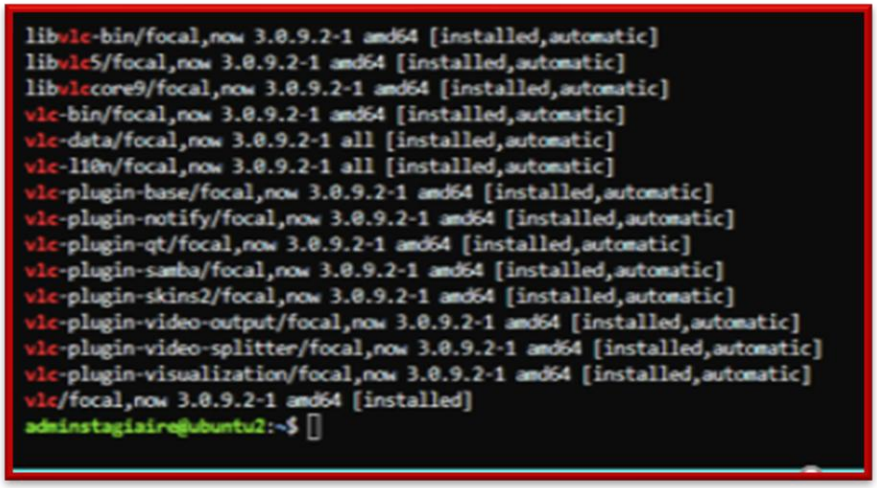


Comme je ne disposais pas d'interface graphique, j'ai vérifié l'installation en listant les paquets installés en ligne de commande.

```
adminstagiaire@ubuntu2:~$ apt list --installed |grep vlc
```

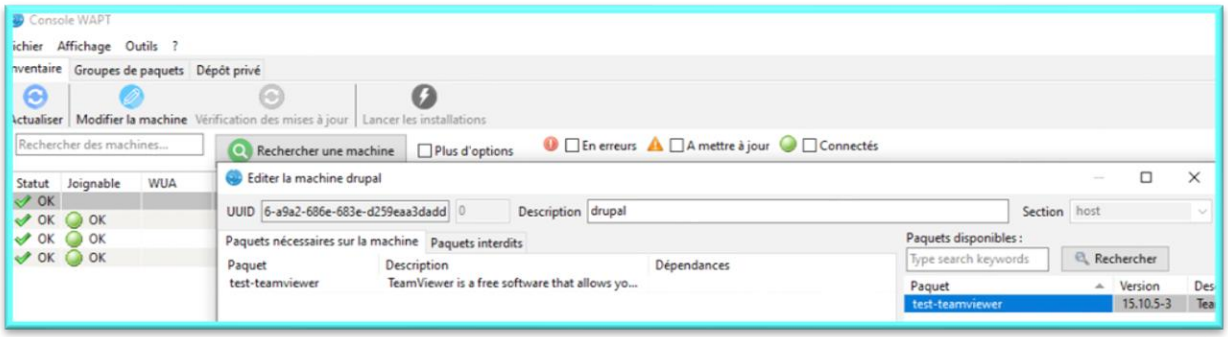
J'ai d'ailleurs utilisé le logiciel Cockpit, présenté dans un chapitre précédent, pour travailler sur plusieurs machines Linux en même temps.

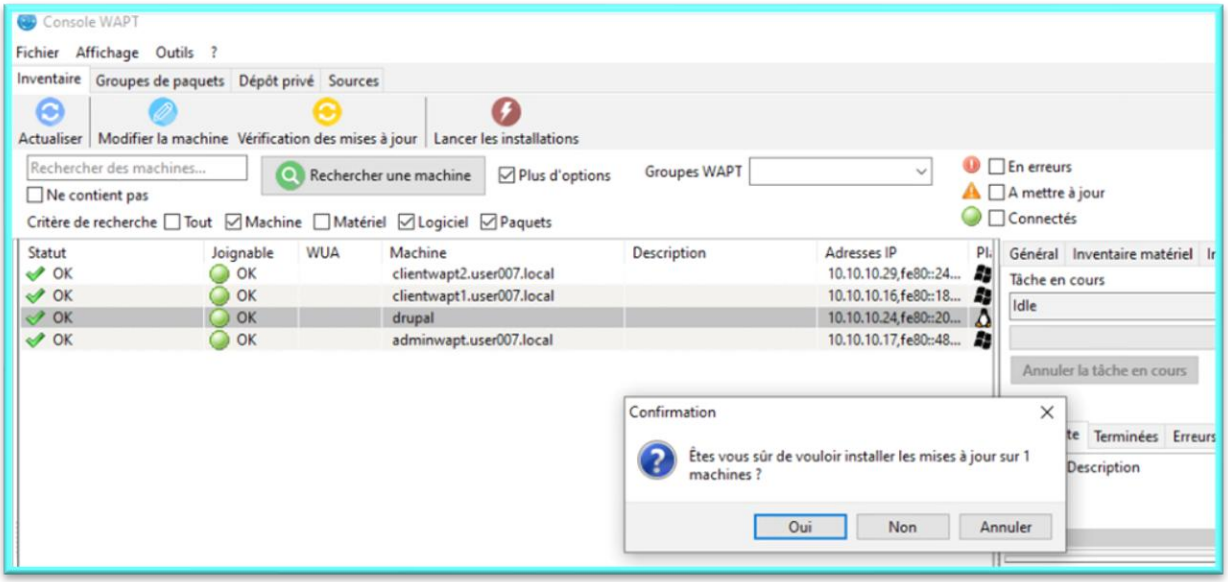




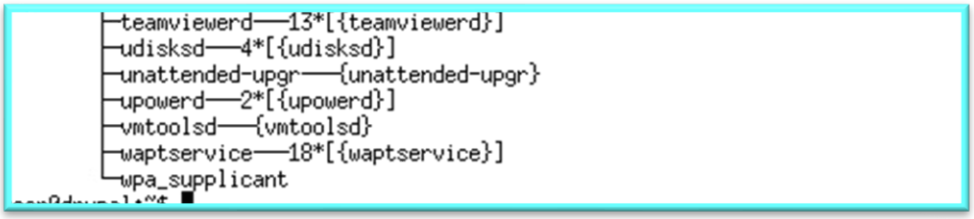
Le logiciel VLC a été installé correctement.

Pour tester le déploiement sur Debian, j'ai choisi le logiciel Teamviewer

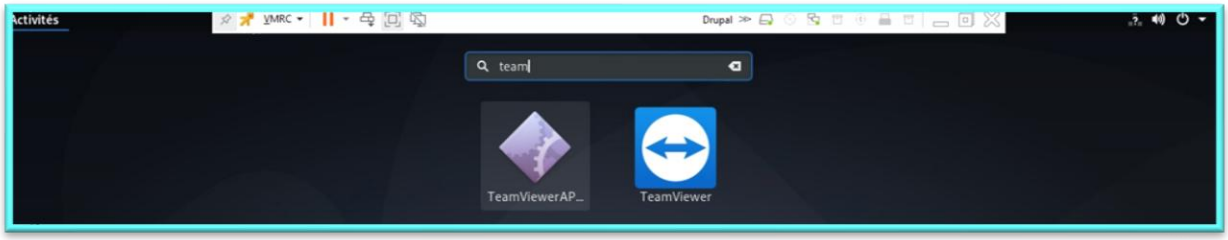




Pour avoir une vue plus claire sur les logiciels installés et pour pouvoir tester TeamViewer j'ai installé une interface graphique (Gnome) sur le client Debian. L'installation du paquet le plus récent s'est effectuée sans problème (après un premier essai infructueux avec une version antérieure du paquet). Le daemon teamviewerd est bien présent sur la machine Debian

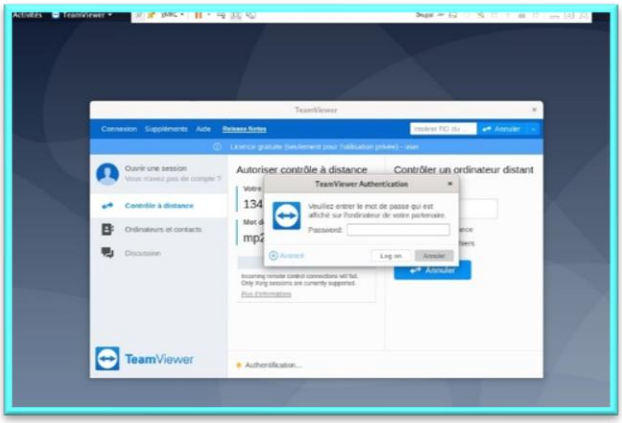


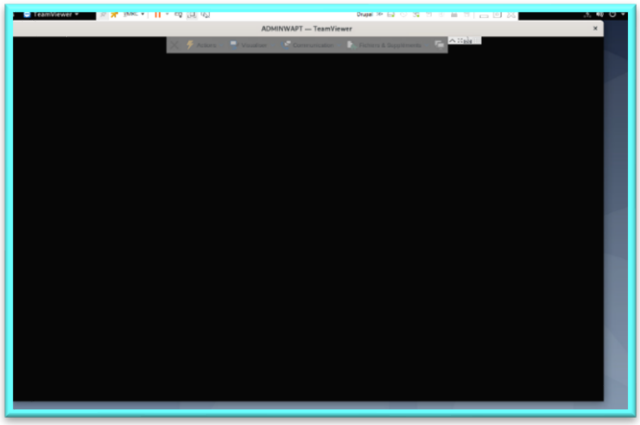
Et l'icône s'affiche bien dans « Activités »



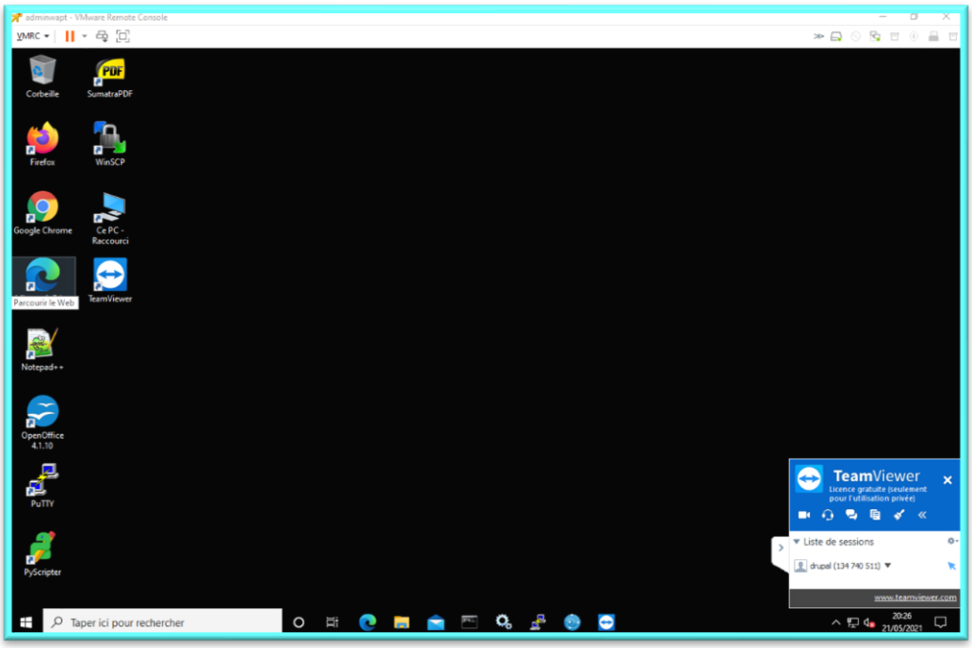
Cependant, une session de contrôle à distance sur le client Debian et un client Windows sur lequel j'ai également installé Teamviewer par l'intermédiaire de Wapt n'as pas vraiment fonctionné ; l'écran était noir, bien qu'on voie sur le client Windows que la session était démarrée

Machine Debian



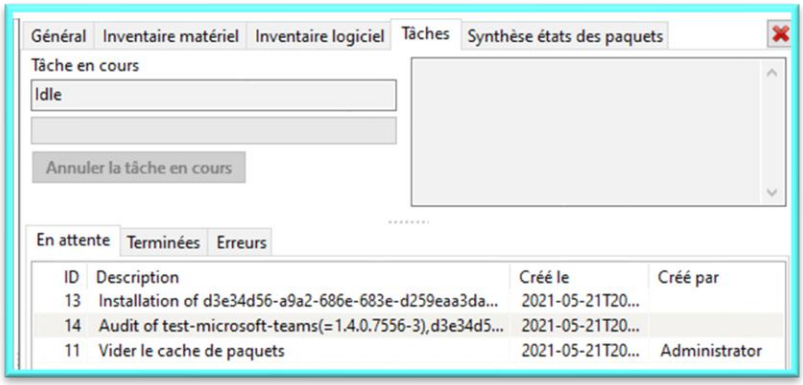


Machine Windows (accès à distance)

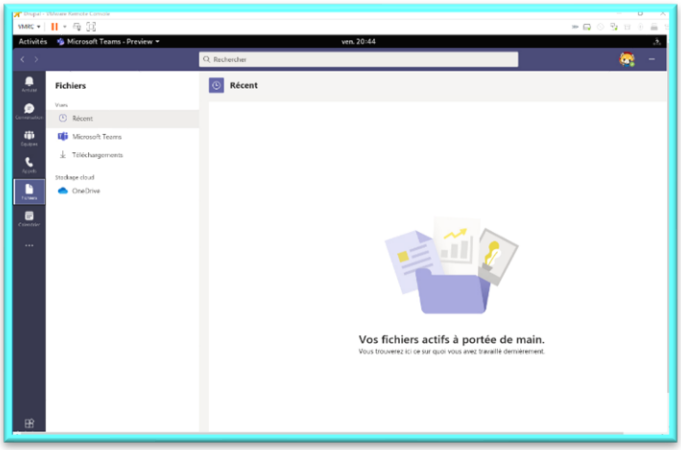


Je suppose qu'il s'agit d'un problème propre au fonctionnement de Teamviewer sur les systèmes Unix.

Un deuxième essai avec le paquet Microsoft Teams fut un plus grand succès :



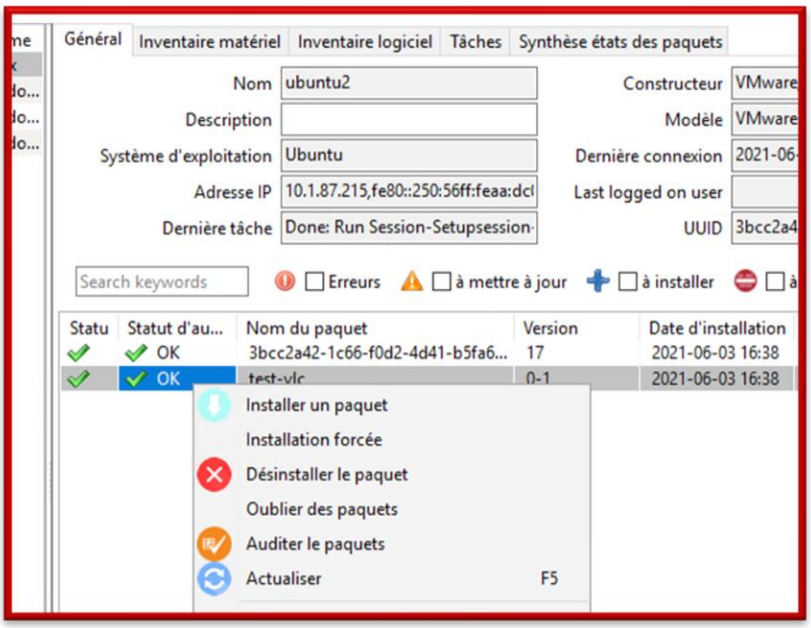
Et Microsoft Teams fonctionne bien sous Debian



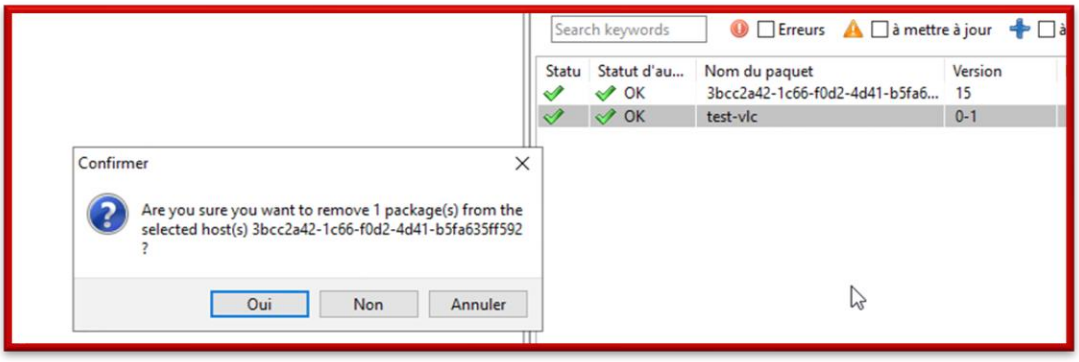
Désinstallation de paquets

La désinstallation de paquets s'est avérée facile autant pour les clients Windows que pour les clients Linux. Les étapes sont les mêmes.

Un clic droit sur le paquet concerné



Comme expliqué ci-dessus, le paquet nommé d'après l'UID du client correspond bien au paquet logiciel VLC.



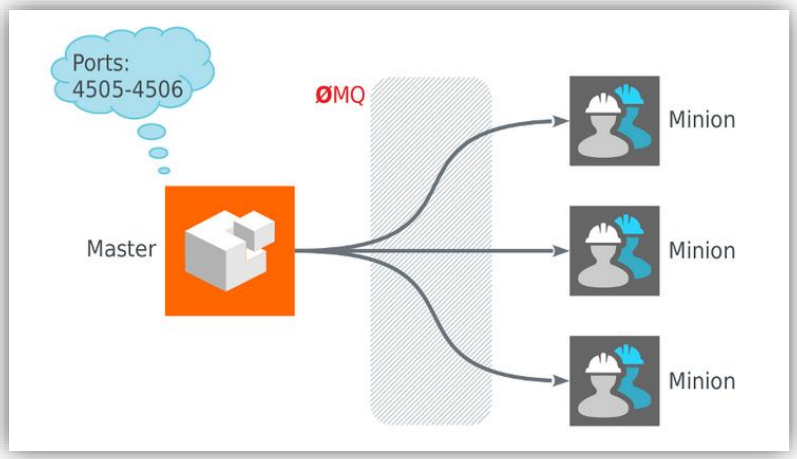
Le logiciel a bien été désinstallé. Cependant, le paquet reste disponible à l'installation et s'affiche toujours dans l'onglet « général » qui concerne le client, mais avec le statut « à installer ».

Pour conclure, WAPT offre de nombreuses possibilités intéressantes à explorer. Les éditeurs peuvent être contactés par de nombreuses voies (Discord, téléphone, mail), et le service technico-commercial a été réactif et aimable.

Cependant, la terminologie propre aux éditeurs et la documentation un peu confuse ne facilitent pas l'utilisation de ce logiciel complexe.

Saltstack

Saltstack est un moteur d'exécution à distance basé sur Python crée en 2011. Le logiciel a été racheté par VMWare en 2020. Cet outil permet d'effectuer des manipulations sur plusieurs machines distantes simultanément et facilite des tâches telles que l'installation et la configuration de serveurs web, serveurs de bases de données ou configurations réseau. L'architecture de Salt se base sur le concept maître (Salt Master) et esclave (Minion) en utilisant Zer0MQ, une bibliothèque de messagerie asynchrone performante pour les communications bidirectionnelles. Zer0Mq implique un système de clé publique pour l'authentification. Le Minion crée et transmet ses clés au Master qui, lui, peut choisir de les accepter ou non. Sur le Master, les communications sortent sur le port 4505 et entrent sur le port 4506.



source: https://duncan.codes/posts/2016- 1

Il est possible d'utiliser Saltstack dans un scénario « agent-less », c'est-à-dire sans installation d'agent sur les machines esclaves. Dans ces cas-là, la communication entre les machines et Minion fonctionne par SSH.

La version payante de Saltstack coûte environ 100$ par ordinateur géré (source : Tutorialspoint)

Les concepts de Saltstack

Salt possède de nombreuses fonctionnalités intéressantes, mais sa terminologie peut sembler complexe. Voici quelques termes clés pour bien illustrer les concepts Saltstack :

Execution Modules

Les modules d'exécution sont des modules Python qui peuvent être lancés directement en ligne de commande. Dans la terminologie de Saltstack, ils font partie des fonctionnalités de « Flow » : des commandes à distance ponctuelles et éphémères, comparé au « State » qui représente une déclaration d'état de configuration souhaité pour un système. Les informations obtenues en lançant les modules sont renvoyées en format JSON (JavaScript Objet Notation), un langage d'échange de données qui permet une représentation structurée de données facilement lisible pour l'utilisateur. Le site des Saltstack propose une grande librairie de modules pour effectuer différentes tâches diverses, mais aussi un tutoriel pour écrire ses propres modules. La structure des commandes Saltstack est la suivante :

```
Salt 'cible' module.fonction  argument

Salt '*' pkg.install  resolvconf
```

Cible: le Minion ciblé par la commande. Il peut s'agir d'un seul PC ou de groupes de machines définis par des expressions régulières (ici : tous les Minions)

Module : module Python qui englobe des fonctions apparentées (ici : gestion de paquets)

Function : une fonction liée au module (ici : installation)

Argument : valeur nécessaire à l'exécution de la fonction (ici : l'utilitaire resolvconf)

State Modules

Les State Modules décrivent un état de configuration de système souhaité durable dans le temps. Ils sont idempotents ; peu importe le nombre de fois qu'on exécute le module, le résultat doit toujours être le même : si, par exemple, l'état souhaité est défini comme « serveur Apache installé sur tous les systèmes Linux », les actions nécessaires, téléchargement et installation, seront exécutées sur les machines Linux ne disposant pas encore de l'utilitaire Apache ; les machines disposant déjà d'Apache resteront dans leur état.

Les state modules sont écrits en langage YAML et portent l'extension .sls. Ils comportent les sections suivantes

Nom du fichier : sample_state.sls

```
Sample_state :          Identifiant arbitraire de l'état de configuration
  Pkg.installed :       Fonction appelée pour attendre l'état (installation
                        de paquets
    -pkgs :             Paramètres de la fonction (paquet Apache)
      - Apache
```

Pour configurer l'état souhaité sur les systèmes, le fichier state est appelé en tant que paramètre pour la fonction « state.apply » et intégré dans un commande Salt :

```
Salt 'clients linux' state.apply sample_state
```

Formula

Il est possible de regrouper plusieurs « states » et d'intégrer les Minions cibles de ces configurations dans des « formules », écrites en YAML. Les « top files » qui comprennent ces formules contiennent les informations suivantes :

Grains

Salt rassemble des informations clés sur les Minions, principalement sur leurs systèmes d'exploitation. Ces ensembles de données sont accessibles sur l'interface « grains » et peuvent servir à cibler un groupe particulier de Minions (serveurs Debian)

Pillar

L'interface Pillar génère et stocke également des données sur les Minons

sous forme clé -valeur sur le master

Mine

Des données sur les résultats des commandes exécutées sont régulièrement stockées dans un fichier « mine » sur le Master

Reactor

Cette fonctionnalité permet de déclencher des actions suite à des événements définis préalablement

Runner

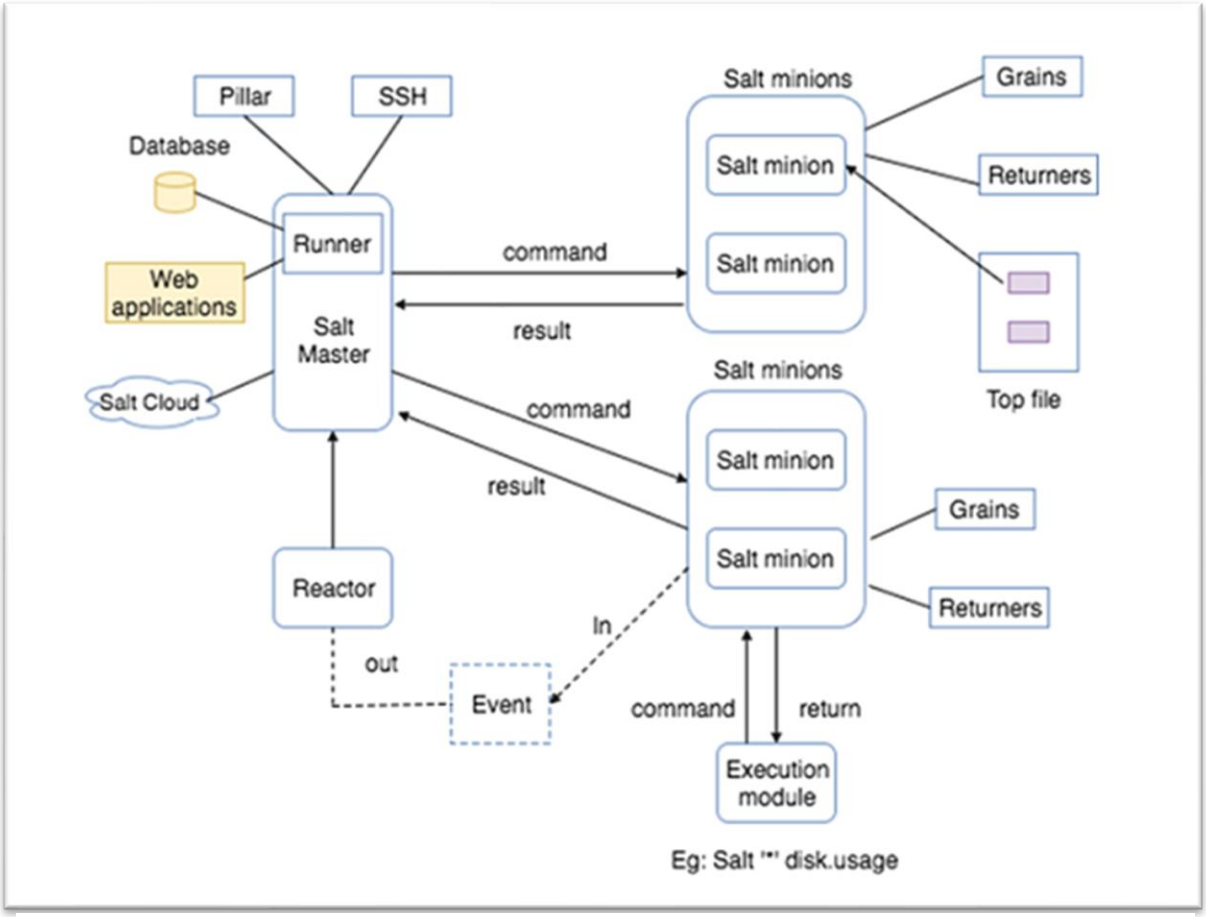
Certains modules, le Runners, s'exécutent sur le Master et non sur les Minions

Returner

Les Returners sont des utilitaires qui peuvent rediriger les données renvoyées par les Minions vers une autre destination que les Mines

Afin de tester l'automatisation de configurations réseau sur mes infrastructures, j'ai utilisé les fonctionnalités d'exécution module et state module.

L’architecture de Saltstack



source: https://www.tutorialspoint.com/saltstack

Environnement test mairie :

- machine Master sous Ubuntu 20.4 : « salt »

-machines Minions sous Windows Server : « winion6, winion104 »

-machines Minion sous Ubuntu 20.4 : « ubu215, ubu93 »

Environnement test privé :

-machine master sous Debian 10 : « salt »

-Minions sous Windows 10 : « saltwin », « winion”

-Minion sous Ubuntu 20.4: “ubunsalt”

-Minion sous Debian 10 : « saltdeb »

Installations

Si Saltstack peux configurer des Minions Windows et Linux, le Master doit être installé sur un système Linux. A la mairie, j'ai utilisé une machine virtuelle Ubuntu 20.4 ; dans mon infrastructure personnelle, j'ai opté pour un serveur Debian 10. Les étapes d'installation sont les mêmes pour les deux systèmes, mais il faut disposer de droits root.

Linux

L'installation de Saltstack sous Linux peut s'effectuer de façons différentes, en ajoutant les dépôts logiciels ou en exécutant un script bootstrap téléchargeable sur le site des éditeurs. J'ai testé les deux solutions sans constater de différences.

Méthode d'installation par ajout des dépôts

sous Ubuntu

```
add-apt-repository ppa:saltstack/salt
```

sous Debian

```
root@salt:~# echo "deb [signed-by=/usr/share/keyrings/salt-archive-keyring.gpg]
https://repo.saltproject.io/py3/debian/10/amd64/latest buster main" | tee /etc/a
pt/sources.list.d/salt.list
```

Téléchargement de la clé du dépôt

```
root@salt:~# curl -fsSL -o /usr/share/keyrings/salt-archive-keyring.gpg https://
repo.saltproject.io/py3/debian/10/amd64/latest/salt-archive-keyring.gpg
```

```
root@salt:~# apt-get install salt-master
```

```
root@salt:~# apt-get install salt-minion
```

Après un apt-update && upgrade, installation des démons principaux, maître et minion

Méthode d'installation par script bootstrap (multi-plateforme)

```
root@ubuntu2:/etc/salt# curl -fsSL https://bootstrap.saltproject.io -o install_s
alt.sh
```

téléchargement du script

exécution du script avec les arguments qui téléchargent la version souhaitée du répo git et installent les démons master et minion

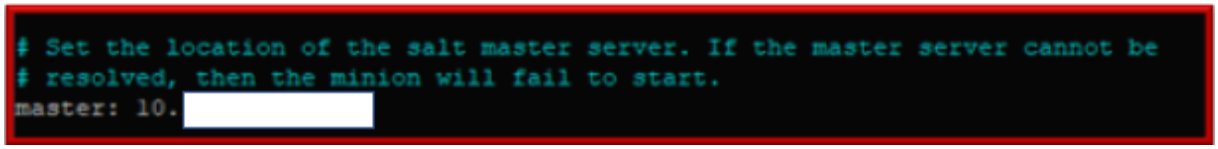
```
root@buster:~#  sh install_salt.sh -P -M git v2015.8.0
```

Configuration de base

Afin que les machines puissent communiquer entre elles, certains éléments doivent être renseignés dans les fichiers principaux de configuration **/etc/salt/master** sur le Master et **/etc/salt/minion** sur les Minions.

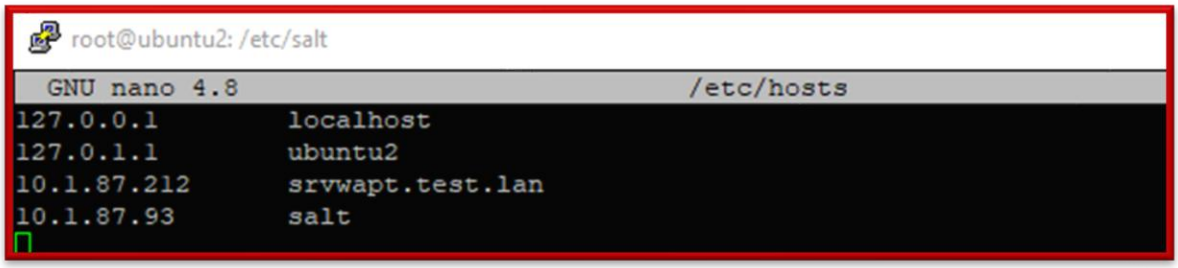
/etc/salt/master : l'interface sur laquelle écoute le Master doit être configurée. Si on souhaite utiliser la seule interface disponible, la boucle locale sera adaptée :

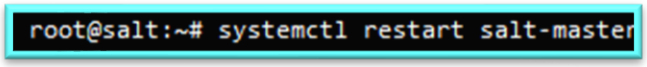
/etc/salt/minion : le master doit être renseigné, soit par adresse IP, soit par nom d'hôte (auquel cas il est, évidemment, nécessaire de renseigner également le fichier hosts sur le Minion) :



Fichier **/etc/hosts** sur les Minions

Après avoir redémarré les services sur le Master et les Minions par les commandes





les clés générées par les Minions doivent être validées sur le Master. On commence par afficher les clés des Minions ayant contacté le Master par la commande salt-key –list-all :

```
root@salt:/home/adminstagiaire# salt-key --list-all
Accepted Keys:
Denied Keys:
Unaccepted Keys:
TEST-NATH.test.lan
test-chef.mp.priv
test-jwin.test.lan
ubuntu1
ubuntu2
Rejected Keys:
```

la commande **Salt-key -accept-all** valide toutes les clés et accepte ainsi les connexions

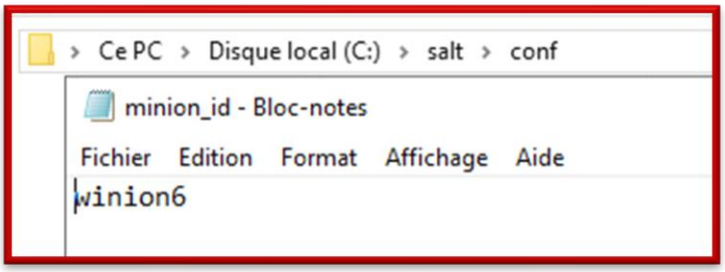
```
root@salt:/home/adminstagiaire# salt-key --accept-all
The following keys are going to be accepted:
Unaccepted Keys:
TEST-NATH.test.lan
test-chef.mp.priv
test-jwin.test.lan
ubuntu1
ubuntu2
Proceed? [n/Y] y
Key for minion test-chef.mp.priv accepted.
Key for minion test-jwin.test.lan accepted.
Key for minion TEST-NATH.test.lan accepted.
Key for minion ubuntu1 accepted.
Key for minion ubuntu2 accepted.
root@salt:/home/adminstagiaire#
```

Par défaut, les noms de Minion des machines correspondent à leurs noms NETBIOS. Pour faciliter la gestion, je les ai renommées par des modifications dans le fichier

/etc/salt/minion_id pour les machines Linux

```
adminstagiaire@ubuntu2: ~

 GNU nano 4.8                                    /etc/salt/minion_id
ubu215
```

Et le fichier **minion_id** dans **c:\salt\conf sous Windows**



La commande salt-key –list-all affiche toutes les clés acceptées

```
root@salt:/home/adminstagiaire# salt-key --list-all
Accepted Keys:
ubu215
ubu93
winion104
winion107
winion6
```

```
root@salt:~# salt-key --list-all
Accepted Keys:
saltdeb
saltwin
ubunsalt
winion
```

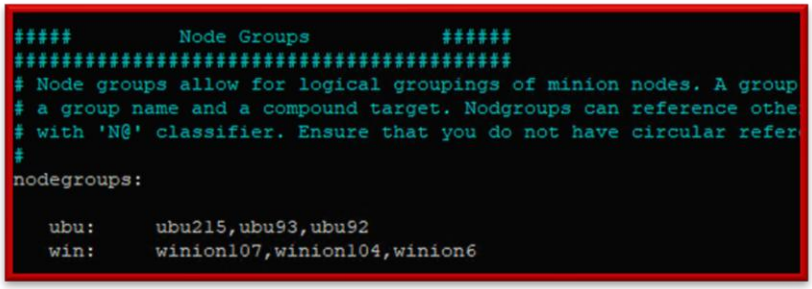
L'environnement Salt est prêt, la première commande peut être lancée.

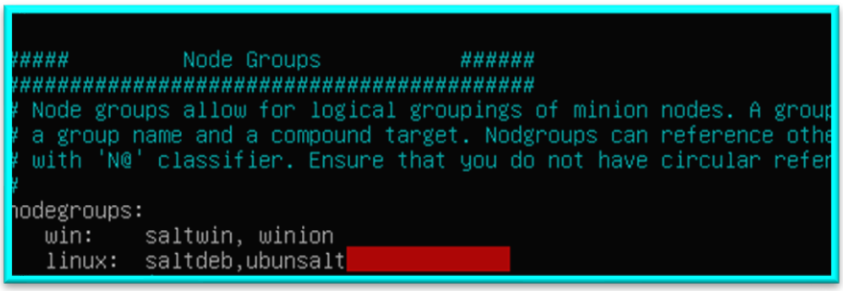
La commande **salt « * » test.ping** permet de vérifier si tous les Minions sont accessibles. Il ne s'agit pas d'un ping IMCP mais d'une fontion Python qui renvoie une valeur booléenne

```
root@salt:/home/adminstagiaire# salt "*" test.ping
ubu93:
    True
ubu215:
    True
winion6:
    True
winion104:
    True
```

```
root@salt:~# salt "*" test.ping
saltdeb:
    True
ubunsalt:
    True
saltwin:
    True
winion:
    True
```

Pour mieux cibler les machines Windows et Linux séparément, j'ai créé des groupes de nodes dans le fichier de configuration : **etc/salt/master**

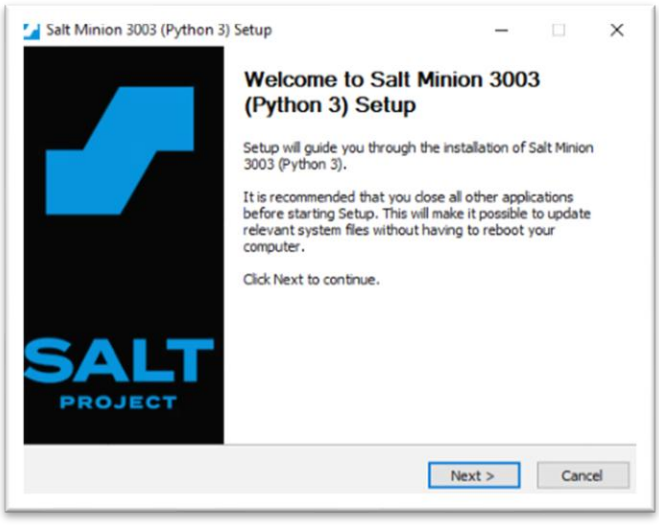


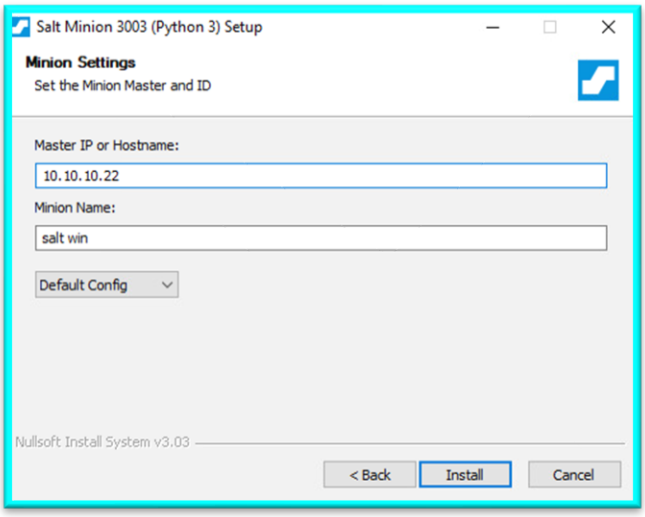


Les nodegroups peuvent être ciblés avec la commande salt -N « nom du nodegroup »

Installation sous Windows

L'installation de Saltstack sous Windows est très simple. Une fois le fichier .msi téléchargé, il suffit de suivre les étapes d'installation, en renseignant correctement les informations sur les Master





La deuxième capture rappelle d'ailleurs une des fenêtres d'installation de WAPT ; on remarque les similarités des architectures maître-esclave / serveur-client.

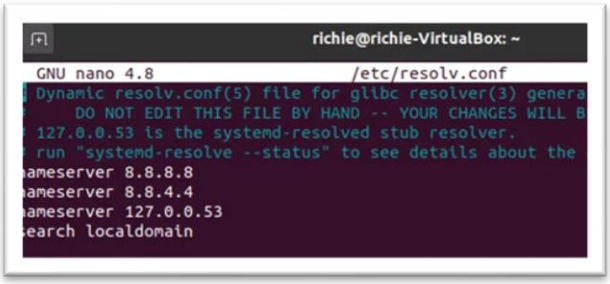
Configuration des serveurs DNS et suffixes de recherche sous Linux

Depuis Debian 10 et Ubuntu 18.4, le fichier de configuration /etc/resolv.conf peut être géré par l'utilitaire resolvconf. Afin d'éviter que le paramétrage soit réinitialisé au démarrage suivant, deux fichiers doivent être modifiés :

/etc/resolvconf/resolv.conf.d/head pour renseigner le(s) serveur(s) de noms

/etc/resolvconf/resolv.conf.d/tail pour renseigner le(s) suffixe(s) de recherche

Les fichiers de configuration tels que resol.conf sont généralement lu de haut en bas par le système. ; dans une liste de serveurs de nom, par exemple, le système intérrogera d'abord le premier qui figure sur la liste



source: https://www.ricmedia.com/set-permanent-dns-nameservers-ubuntu-debian-resolv-conf/

Les serveurs suivants sont intérrogés si le premier serveur ne répond pas. Par contre, s'il répond, que la réponse soit positive ou négative, le système n'interrogera pas les serverus suivants. Ceci peut poser un problème dans un scénario où on souhaiterait utiliser un serveur DNS interne et un serveur DNS public , dans l'idée de pouvoir résoudres des adresses publiques même si le DNS interne primaire est défaillant. Un serveur DNS interne proprement configuré contient une entrée «forwarder » : si le serveur même ne connaît pas le nom DNS qu'on lui demande de résoudre, il redirige sa demande à un ou plusieurs serveurs susceptibles de connaître la réponse. Dans le scénario du DNS interne défaillant, l'utilisateur n'obtiendra pas de réponse s'il recherche un poste interne, mais pourra toujours résoudre des noms de sites publics tels que google.com parce que le serveur DNS public prend le relais. Si, par contre, le serveur DNS publique, par exemple 8.8.8.8, est renseigné en premier et que l'utilisateur cherche à résoudre un nom d'hôte interne, il obtiendra la réponse que l'hôte n'existe pas,car le serveur DNS public ne cherchera pas à rediriger la demande de résolution vers le serveur DNS interne (qu'il ne connaît pas). Il est donc important de respecter l'ordre des entrées dans la liste.

Un module qui écrase le contenu d'un fichier et le remplace par du texte est une première solution : file.write . Mais dans le cas où l'utilisateur aurait souhaité conserver les entrées de serveurs de noms mais rajouter un serveur principal, j'avais donc besoin de trouver un module Saltstack qui puisse rajouter du texte au début du fichier. Ce module existe, dans les modules d'exécution et les modules state :File.prepend. J'ai testé les deux fonctions de façon différente.

Pour essayer le module d'exécution file.write je l'ai intégré dans la commande suivante

```
root@salt:~# salt -N 'linux' file.write /etc/resolvconf/resolv.conf.d/head  'nameserver 8.8.8.8 '
```

En plus de la syntaxe de base décrite plus haut elle contient le chemin vers le fichier à modifier (/etc/resolv/resol.conf.d/head) et le texte à ajouter entre quotes (nameserver 8.8.8.8)

La commande s'est exécutée sans problèmes :

```
saltdeb:
    Wrote 1 lines to "/etc/resolvconf/resolv.conf.d/head"
ubunsalt:
    Wrote 1 lines to "/etc/resolvconf/resolv.conf.d/head"
```

Et le fichier a été modifié sur les machines ciblées :

```
root@saltdeb:~# cat /etc/resolv.conf
nameserver 8.8.8.8
```

```
root@ubunsalt:/home/user# cat /etc/resolv.conf
nameserver 8.8.8.8
```

Par la suite, j'ai créé un fichier state en précisant deux entrées. Le chemin par défaut des fichiers state est /srv/salt sur le Master qui joue alors le rôle d'un serveur de fichiers.

Les fichiers .sls sont écrits en YAMl et doivent respecter sa syntaxe :

- Une indentation de deux espaces pour chaque niveau de données, pas de tabulations
- Les objets d'une liste sont séparés par un tiret
- Un ensemble paramètre et valeur est représenté par *paramètre :valeur*

Le fichier state pour rajouter du texte au début du fichier pour configurer les adresses des serveurs DNS nommé dns.sls se présente donc comme ceci :

root@salt: /srv/salt

```
GNU nano 4.8                                          dns.sls
set_dns_servers:
  file.prepend:
    - name: /etc/resolvconf/resolv.conf.d/head
    - text:
       - nameserver 10.
       - nameserver 10.
```

Le fichier contient un identifiant (set_dns_servers) la fonction (file.prepend) et les paramètres (chemin du fichier à modifier, contenu à insérer).

Pour utiliser le fichier, il doit être intégré dans une commande salt. Cette commande lance le module state.apply qui « applique les états » décrits dans les fichiers.Le nom du fichier devient alors l' argument

```
root@salt:/home/adminstagiaire# salt -N 'ubu' state.apply dns
```

Quand la commande s'est exécutée avec succès, Saltstack renvoie des informations sur les modifications effectuées sur les minions :

```
Summary for ubu93
------------
Succeeded: 1 (changed=1)
Failed:    0
------------
Total states run:     1
Total run time:   9.809 ms
ubu92:
----------
          ID: set_dns_servers
    Function: file.prepend
        Name: /etc/resolvconf/resolv.conf.d/head
      Result: True
     Comment: Prepended 2 lines
     Started: 10:27:29.872375
    Duration: 16.087 ms
     Changes:
              ----------
              diff:
                  ---
                  +++
                  @@ -0,0 +1,2 @@
                  +nameserver 10.
                  +nameserver 10.

Summary for ubu92
------------
Succeeded: 1 (changed=1)
Failed:    0
------------
Total states run:     1
Total run time:   16.087 ms
```

Les modifications ont été prises en compte et ne sont pas écrasées après redémarrage.

Cette stratégie semblait donc fonctionner. Par conséquent, j'ai utilisé la même fonction pour modifier le fichier gérant les suffixes de recherche en créant ce fichier state :

```
GNU nano 4.8                    /srv/salt/suffix.sls
search_domain_suffix:
  file.prepend:
    - name: /etc/resolvconf/resolv.conf.d/tail
    - text:
       - search mp.priv
       - search gs.mp.priv
       - search vis.mp.priv
```

La commande entière :

```
root@salt:/home/adminstagiaire# salt -N 'ubu' state.apply suffix
```

Les informations renvoyées :

```
Summary for ubu93
------------
Succeeded: 1 (changed=1)
Failed:    0
------------
Total states run:     1
Total run time:  10.584 ms
ubu92:
----------
          ID: search__domain_suffix
    Function: file.prepend
        Name: /etc/resolvconf/resolv.conf.d/tail
      Result: True
     Comment: Prepended 3 lines
     Started: 10:36:47.101217
    Duration: 14.667 ms
     Changes:
              ----------
              diff:
                  ---
                  +++
                  @@ -0,0 +1,3 @@
                  +search mp.priv
                  +search gs.mp.priv
                  +search vis.mp.priv
```

Et le résultat de la commande, persistant après redémarrage :

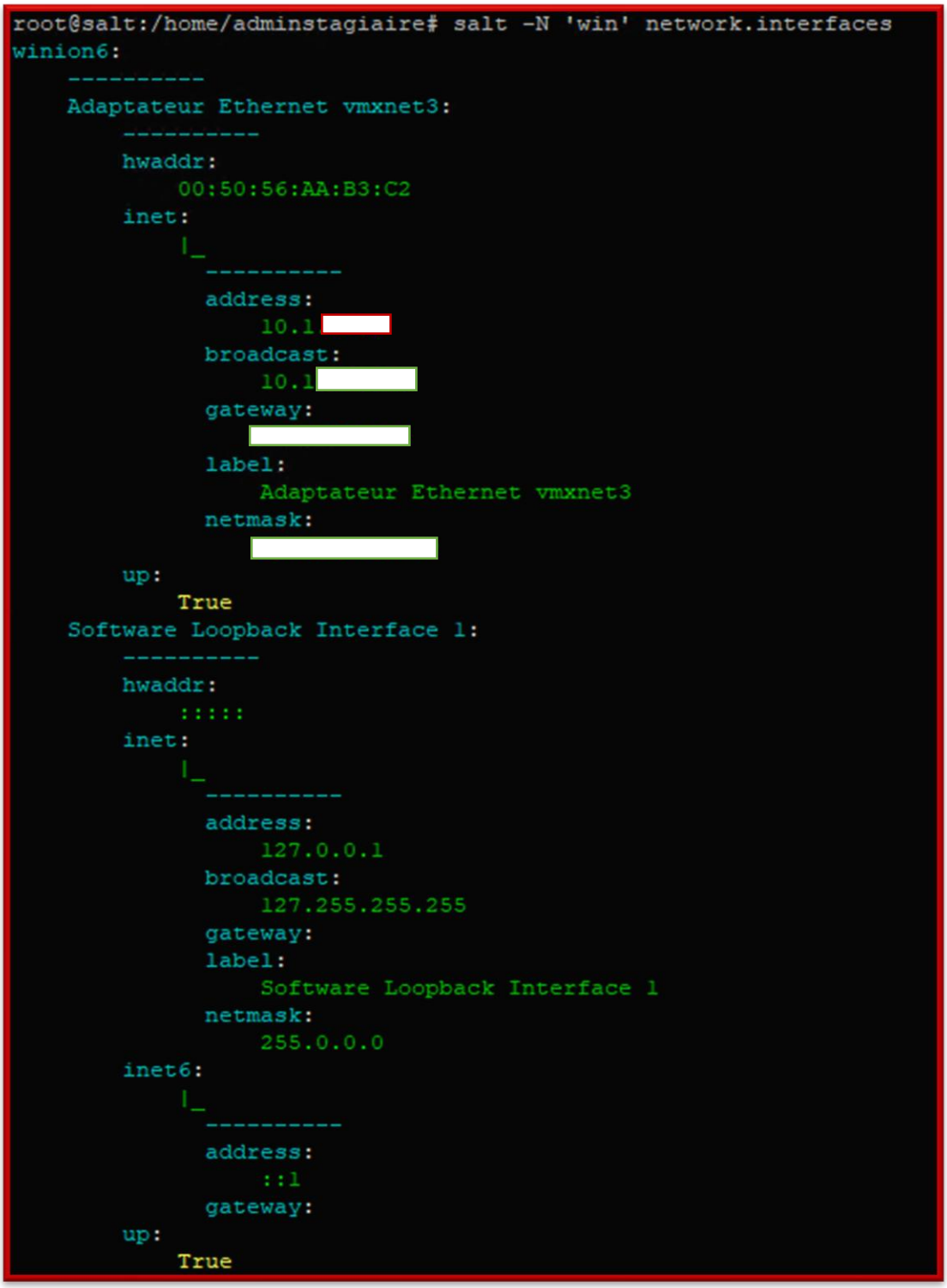
```
 GNU nano 4.8                                  /etc/resolvconf/resolv.conf.d/ta
search mp.priv
search gs.mp.priv
search vis.mp.priv
```

```
 GNU nano 4.8                                  /etc/resolvconf/resolv.conf.d/head
nameserver 10.
nameserver 10.
```

La configuration DNS des machines Ubuntu était donc terminée. Je suis donc passée à la configuration DNS des Minons sous Windows.

Configuration DNS sous Windows

La libraire de salt contient de nombreux modules très intéressants pour configurer des systèmes Windows. Je n'ai donc pas eu à faire beaucoup d'efforts pour ajouter les serveurs DNS.Il fallait tout de même connaître les noms des cartes réseau, ce qui peut poser problème quand les machines disposent de plusieurs NIC. Saltstack dispose d'un module qui renvoie des informations sur toutes les interfaces réseau



Mais dans ce cas précis, il renvoie le nom de l'adaptateur virtuel qui ne correpond pas au nom de l'interface dans le système d'exploitation.

Dans ce cadre, le script Powershell utilisé préalablement dans les essais Jenkins qui retourne le nom d'interface par appartenance à un réseau peut se révéler très utile.

```
root@salt:/home/adminstagiaire# salt -N 'win' cmd.run_all '$(Get-NetIPAddress |
Where-Object {$_.IPAddress -like "10.1.*"}).InterfaceAlias' shell=powershell
winion6:
    ----------
    pid:
        4848
    retcode:
        0
    stderr:
    stdout:
        Ethernet0
winion104:
    ----------
    pid:
        7252
    retcode:
        0
    stderr:
    stdout:
        Ethernet0
```

Les interfaces s'appellent donc toutes Ethernet0

Un premier module permet de vérifier la configuration actuelle des serveurs DNS sur la machine ciblée

```
root@salt:/home/adminstagiaire# salt 'winion104' win_dns_client.get_dns_servers 'Ethernet0'
winion104:
    - 10.1.87.104
    - 8.8.8.8
```

Le module win_dns_client.add_dns ajoute une serveur DNS à l'interface

```
root@salt:/home/adminstagiaire# salt 'winion104' win_dns_client.add_dns 8.8.4.4 'Ethernet0'
winion104:
    True
```

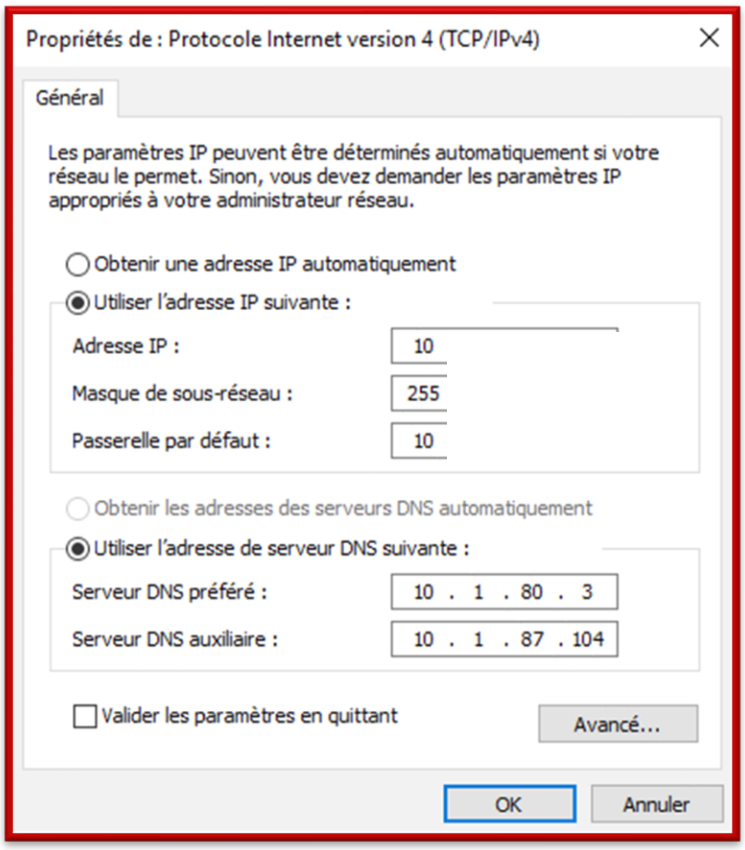
Il est aussi possible de créer un state

```
GNU nano 4.8                    /srv/salt/dnswin.sls
change_dns:
  win_dns_client.dns_exists:
    - replace: True
    - servers:
      - 10.1.80.3
      - 10.1.87.104
    - interface: Ethernet0
```

Application du state et retour

```
root@salt:/home/adminstagiaire# salt -N 'win' state.apply dnswin
winion6:
----------
          ID: change_dns
    Function: win_dns_client.dns_exists
      Result: True
     Comment: DNS Servers have been updated
     Started: 22:59:48.115044
    Duration: 2218.708 ms
     Changes:
              ----------
              Servers Added:
                  - 10.1.80.3
                  - 10.1.87.104
              Servers Removed:
                  - 8.8.8.8
                  - 8.8.4.4
              Servers Reordered:

Summary for winion6
------------
Succeeded: 1 (changed=1)
Failed:    0
```

La vérification montre que les modifications ont été prises en compte



Configuration du suffixe de recherche

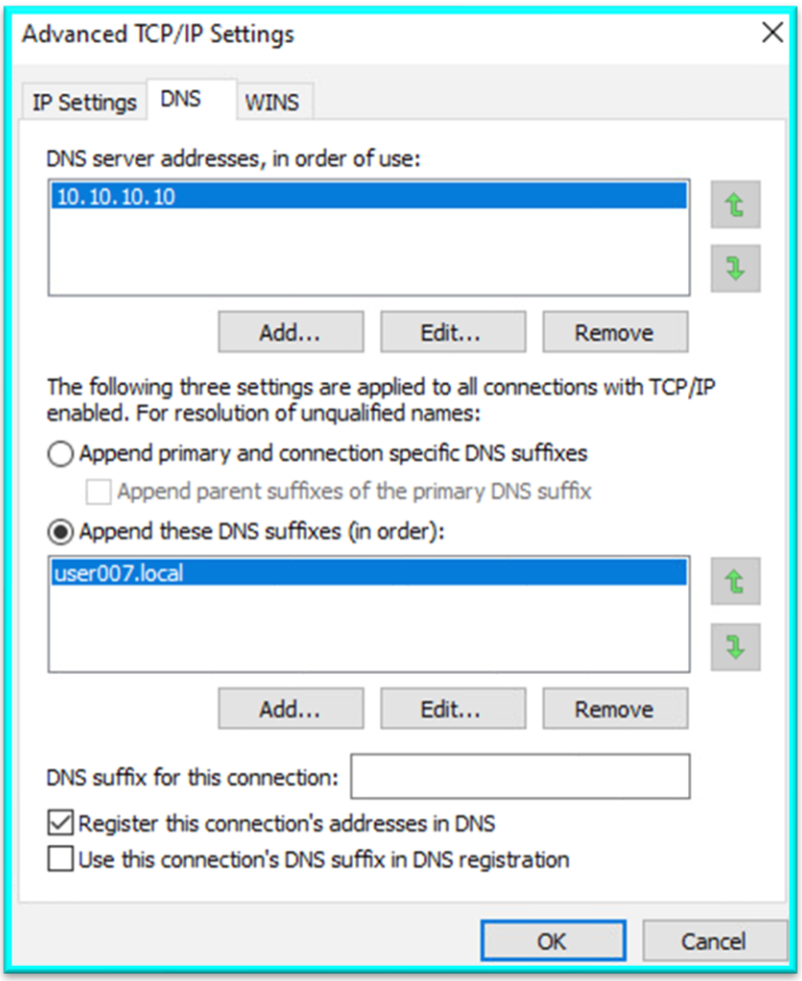
Au moment de ma recherche, il n'existait pas de module prédéfini pour ajouter des sufixes de recherche. J'ai donc utilisé la même commande Powershell que pour Jenkins, en l'exécutant avec le module **cmd.run_all**

```
root@salt:~# salt 'saltwin' cmd.run_all 'Set-DnsclientGlobalSetting -SuffixSearchList user007.local' shell=powers
hell
```

La commande s'est exécutée sans problèmes

```
saltwin:
    ----------
    pid:
        5848
    retcode:
        0
    stderr:
    stdout:
```

Et après vérification , les paramètres sont bien configurés



Saltstack s'est avéré très pratique et efficace pour configurer les paramètres réseau aussi bien sur des machines Windows que Linux.

Comme il existe de modules Saltstack qui peuvent afficher des informations sur les logiciels installés, j'ai jeté un coup d'œil sur ces fonctionnalités.

Gestion de logiciels sous Windows

Afin de pouvoir déployer des logiciels sur les Minions sous Windows, il est nécessaire d'ajouter un dépôt logiciel pour Windows au Master. Cette commande ajoute le dépôt par défault

```
root@salt:~# salt-run winrepo.update_git_repos
https://github.com/saltstack/salt-winrepo-ng.git:
    /srv/salt/win/repo-ng/salt-winrepo-ng
https://github.com/saltstack/salt-winrepo.git:
    /srv/salt/win/repo/salt-winrepo
```

Le module pkg permet alors de lister les versions disponibles d'un logiciel compatible avec les Minions sélectionnés. Dans ce cas précis, il s'agit de VLC

```
root@salt:~# salt -N 'win' pkg.list_available vlc
winion:
    - 2.2.6
    - 2.2.8
    - 3.0.0
    - 3.0.1
    - 3.0.2
    - 3.0.3
    - 3.0.4
    - 3.0.5
    - 3.0.6
    - 3.0.7
    - 3.0.7.1
    - 3.0.8
    - 3.0.9
    - 3.0.9.1
    - 3.0.9.2
    - 3.0.10
    - 3.0.11
    - 3.0.12
```

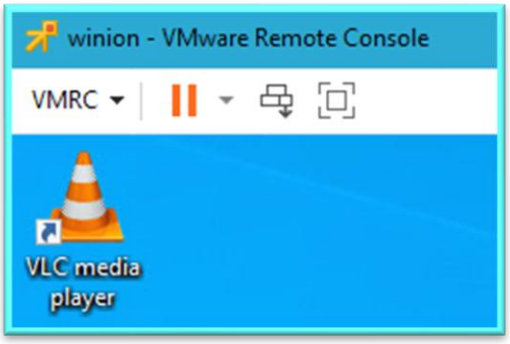
La fonction pkg.install installe le logiciel (il est possible d'en installer plusieurs à la fois)

```
root@salt:~# salt -N 'win' pkg.install vlc
winion:
    ----------
    vlc:
        ----------
        new:
            3.0.12
        old:
saltwin:
    ----------
    vlc:
        ----------
        new:
            3.0.12
        old:
```

Et la fonction pkg.version vérifie la version installée et renvoie un string vide si le logiciel n'est pas présent sur le Minion

```
root@salt:~# salt -N 'win' pkg.version vlc
winion:
    3.0.12
saltwin:
    3.0.12
```

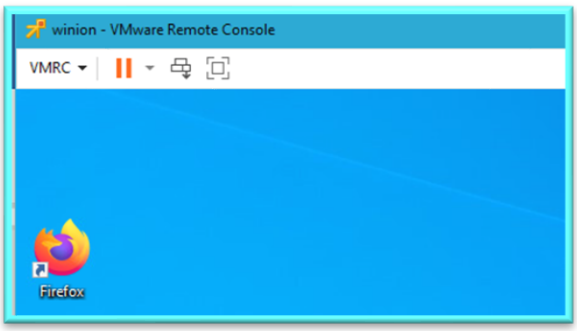
Le logiciel est bien présent sur le Minion



La désinstallation des logiciels est également possible avec la fonction pkg.remove

```
root@salt:~# salt -N 'win' pkg.remove vlc
winion:
    ----------
    vlc:
        ----------
        new:
        old:
            3.0.12
saltwin:
    ----------
    vlc:
        ----------
        new:
        old:
            3.0.12
```

Le logiciel a été désinstallé et le raccourci supprimé



Pour effectuer les même tâches d'installation, vérification et désinstallation sur de minions linux, Saltstack propose différentes fonctions selon les distributions. Les fonctions d'installation et de vérification de version portent le même nom pour Windows , Ubuntu et Debian

Pour installer le logiciel libroffice-calc

```
root@salt:~# salt -N 'linux' pkg.install libreoffice-calc
saltdeb:
    ----------
    libreoffice-calc:
        ----------
        new:
            1:6.1.5-3+deb10u7
        old:
ubunsalt:
    ----------
    libreoffice-calc:
        ----------
        new:
            1:6.4.7-0ubuntu0.20.04.1
        old:
```

Pour vérifier la version installée

```
root@salt:~# salt -N 'linux' pkg.version libreoffice-calc
saltdeb:
    1:6.1.5-3+deb10u7
ubunsalt:
    1:6.4.7-0ubuntu0.20.04.1
```

Seule la fonction de désinstallation s'appelle purge au lieu de remove

```
root@salt:~# salt -N 'linux' pkg.purge libreoffice-calc
saltdeb:
    ----------
    installed:
        ----------
        libreoffice-calc:
            ----------
            new:
            old:
                1:6.1.5-3+deb10u7
    removed:
        ----------
ubunsalt:
    ----------
    installed:
        ----------
        libreoffice-calc:
            ----------
            new:
            old:
                1:6.4.7-0ubuntu0.20.04.1
    removed:
        ----------
```

Il est donc possible de vérifier et gérer les installations de logiciels sur des minions Windows et Linux avec Saltstack. Cependant, les manipulations ne se font pas avec le même « confort » d'une interface graphique ni la même facilité de recherche que sous WAPT.

Conclusion

Lors de ma période d'activité en entreprise, j'ai été amenée à me former dans un domaine que je ne connaissais pas du tout : L'infrastructure as Code. Bien qu'il ne soit pas nécessaire d'être développeur pour maitriser les outils liés à ce domaine, plusieurs situations m'ont obligée à creuser et approfondir mes connaissances basiques en langages de Scripting et programmation, et je ne le regrette pas.

J'ai eu l'occasion d'essayer quatre logiciels différents qui avaient un point en commun : ils visaient à faciliter la gestion de grands parcs informatiques : Cockpit permet un aperçu et des fonctionnalités de gestion de plusieurs machines Linux. Jenkins facilite l'exécution centraliser de tâches d'automatisation comme l'exécution de scripts. Les personnes très compétentes en Scripting peuvent effectuer de nombreuses tâches de configuration par cet intermédiaire. Cependant, pour gérer plusieurs machines sous différents systèmes d'exploitation, il est nécessaire de planifier les droits d'accès et authentification de façon rigoureuse et de bien se documenter pour pouvoir passer le relais dans le cas ou plusieurs administrateurs de système doivent intervenir sur l'infrastructure. La gestion de configuration est nettement plus facile avec des outils prévus à cet effet comme Saltstack : des nombreux modules gèrent les permissions automatiquement, et il est tout de même possible d'utiliser des scripts personnalisés ou même d'écrire ses propres modules. La gestion du déploiement peut également être confiée à Saltstack ; cependant, pour les personnes qui préfèrent avoir une interface graphique qui offre une meilleure visibilité sur l'inventaire logiciel et matériel du parc, WAPT offre un plus grand confort et de nombreuses fonctionnalités intéressantes. Rien n'empêche d'utiliser plusieurs de ces outils parallèlement.

Tous les outils que j'ai testés ont également en commun la possibilité de créer des modules soi-même ; le fait de pouvoir personnaliser ses solutions est un point attractif.

Cependant, plus un outil informatique est sophistiqué, plus il est nécessaire de « plonger » dans son univers : les terminologies et concepts propres à ces systèmes demandent un certain temps d'apprentissage. Ceux qui ont la possibilité d'investir le temps et les efforts

pour apprendre trouveront ces connaissances très enrichissantes.

Sources

@expta, Jeff Guillet -. “How to Self-Elevate a PowerShell Script.” *The EXPTA {Blog}*, 1 Jan. 1970, blog.expta.com/2017/03/how-to-self-elevate-powershell-script.html.

“CI/CD : Comprendre L'essentiel En 8 Minutes.” *YouTube*, YouTube, 15 Mar. 2021, www.youtube.com/watch?v=ws1qGuFMYlc.

Cockpit Project, cockpit-project.org/.

“Continuous Integration : Définition De L'intégration Continue.” *IONOS Digitalguide*, www.ionos.fr/digitalguide/sites-internet/developpement-web/integration-continue/.

“Documentation De WAPT¶.” *Documentation De WAPT - Documentation WAPT 1.8.2*, www.wapt.fr/fr/doc-1.8/.

“Endpoint Management: KACE by Quest.” *Endpoint Management | KACE by Quest*, www.quest.com/kace/.

Freeman, Emily. *DevOps*. Wiley & Sons Canada, Limited, John, 2019.

Gillis, Alexander S. “Was Ist Kontinuierliche Integration (Continuous Integration)? - Definition Von WhatIs.com.” *ComputerWeekly.de*, ComputerWeekly.com/De, 20 Dec. 2020, www.computerweekly.com/de/definition/Kontinuierliche-Integration-Continuous-Integration?_ga=2.189922347.1671774306.1624348889-1848333254.1623663056.

Gillis, Alexander S. “Was Ist Kontinuierliche Integration (Continuous Integration)? - Definition Von WhatIs.com.” *ComputerWeekly.de*, ComputerWeekly.com/De, 20 Dec. 2020, www.computerweekly.com/de/definition/Kontinuierliche-Integration-Continuous-Integration?_ga=2.189922347.1671774306.1624348889-1848333254.1623663056.

HALL, JOSEPH. *MASTERING SALTSTACK*. PACKT Publishing Limited, 2017.

“Infrastructure as Code: Chef, Ansible, Puppet, or Terraform?” *IBM*, www.ibm.com/cloud/blog/chef-ansible-puppet-terraform.

JasonGerend. “DnsClient Module.” *Module | Microsoft Docs*, docs.microsoft.com/en-us/powershell/module/dnsclient/.

Jenkinsci, director. *Jenkins Is the Way to Build, Test, and Deploy*. *YouTube*, YouTube, 11 Dec. 2020, www.youtube.com/watch?v=_MXtbjwsz3A&t=3s.

Kiarie, James, et al. “James Kiarie.” *Tecmint*, 28 Aug. 2019, www.tecmint.com/install-free-ssl-certificate-for-nginx-on-debian-10/.

Kili, Aaron, et al. “Aaron Kili.” *Tecmint*, 6 July 2020, www.tecmint.com/set-permanent-dns-nameservers-in-ubuntu-debian/comment-page-2/#comment-1517223.

Kim, Gene, et al. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, LLC, 2017.

L, +Bastien. “Infrastructure as Code : Qu'est-Ce Que C'est Et à Quoi Ça Sert ?” *LeBigData.fr*, 5 Jan. 2019, www.lebigdata.fr/infrastructure-as-code-definition.

“Le DevOps Expliqué En Emojis.” *YouTube*, YouTube, 8 Jan. 2017, www.youtube.com/watch?v=M6F6GWcGxLQ.

Morris, Kief. *Infrastructure as Code: Managing Servers in the Cloud*. O'Reilly Media, 2016.

Myers, Colton. *Learning SaltStack - Second Edition*. Packt Publishing, 2016.

“Perpignan Méditerranée Métropole - QUI SOMMES-NOUS ?” *Perpignan*, www.perpignanmediterraneemetropole.fr/qui-sommes-nous.

“Qu'est-Ce Que L'intégration Continue ?” *OpenClassrooms*, openclassrooms.com/fr/courses/2035736-mettez-en-place-lintegration-et-la-livraison-continues-avec-la-demarche-devops/6182691-quest-ce-que-lintegration-continue.

Randell, Beverley. “Fr.” *Amazon*, Nelson Cengage Learning, 1996, aws.amazon.com/fr/devops/continuous-integration/.

“SaltStack - Quick Guide.” *Tutorialspoint*, www.tutorialspoint.com/saltstack/saltstack_quick_guide.htm.

Spam, No, et al. “How to Disable/Change User Account Control with Group Policy?” *Windows OS Hub*, 4 Dec. 2020, woshub.com/user-account-control-slider-and-group-policy-settings/.

Swartout, Paul. *Continuous Delivery and DevOps: a Quickstart Guide*. Packt Publishing Limited, 2014.

“What Is Infrastructure as Code? Difference of Infrastructure as Code Tools.” *YouTube*, YouTube, 28 Aug. 2020, www.youtube.com/watch?v=POPP2WTJ8es.

Young, Austin. *Infrastructure as Code: A Comprehensive Guide to Managing Infrastructure as Code* . Independently Published , 2019.

Zamot, Michael. “An Introduction to Cockpit, a Browser-Based Administration Tool for Linux.” *Enable Sysadmin*, Red Hat, Inc., 14 Apr. 2020, www.redhat.com/sysadmin/intro-cockpit.