



**Institut Universitaire de Technologie,  
Aix-Marseille Université**

**RAPPORT DE STAGE de fin de deuxième année  
Bachelor Universitaire de Technologie  
Spécialité Réseaux et Télécommunications  
parcours cybersécurité**

**Automatisation des tâches Réseaux**

**Guillaume Halary**

**Direction Générale de l'Armement**

**Responsable entreprise : Laurent Laugier**

**Responsable académique : Corinne Houssain**

**2024**



## Table des matières

1	Introduction.....	5
2	Direction Générale de l'Armement.....	5
2.1	L'entreprise.....	5
2.2	Le service informatique.....	6
2.2	Organigramme.....	7
3	Projet Automatisation des tâches réseau.....	7
3.1	Contexte.....	7
3.2	Objectif.....	8
3.3	Maquette.....	8
3.4	Solutions.....	11
3.5	Choix de la solution.....	15
4	Tâches de suivis des administrateurs.....	21
4.1	Installation de machines Ubuntu 22.04 pour les développeurs.....	21
4.2	Utilisation de Clonezilla pour le clonage et la sauvegarde de disques.....	22
4.3	Suivi du problème du Robot de sauvegarde Dell Quantum.....	24
5	Conclusion.....	25
6	Remerciements.....	27
7	Glossaire.....	29
8	Sitographie.....	31



# 1 Introduction

L'informatique occupe une place de plus en plus importante dans le domaine aéronautique, c'est dans ce contexte que j'ai pu intégrer la Direction Générale de l'Armement (DGA) pour mon stage en informatique, spécialisé dans les réseaux et systèmes. Au cours de ce stage, j'ai participé aux différentes missions confiées au service de Système de Transmission Mécatronique (STM).

Je présenterai dans ce rapport le secteur d'activité dans lequel j'ai évolué de DGA-EV. Je détaillerai également les différentes missions qui m'ont été confiées. Je présenterai aussi les contraintes liées au contexte aéronautique de notre activité et finirai par un résumé de cette expérience très enrichissante au sein de DGA-EV.

## 2 Direction Générale de l'Armement

### 2.1 L'entreprise

La Direction Générale de l'Armement (DGA) est un acteur central au sein du ministère des Armées en France, responsable de la préparation, de l'équipement et du soutien des forces armées françaises. Fondée en 1961 par un décret présidentiel sous la présidence de Charles de Gaulle, la DGA a été créée pour centraliser les efforts de recherche, de développement et d'acquisition de matériels militaires, afin de garantir l'indépendance et la souveraineté technologique de la France en matière de défense.

La mission principale de la DGA est de fournir aux forces armées les équipements nécessaires pour assurer la sécurité nationale. Elle intervient à toutes les étapes du cycle de vie des systèmes d'armement : de la recherche et développement, à la conception, en passant par les tests, la mise en service et le soutien opérationnel. Pour ce faire, la DGA collabore étroitement avec l'industrie de la défense, les instituts de recherche et les universités, afin de développer des solutions innovantes et performantes.

La DGA joue un rôle essentiel dans la définition des besoins des forces armées et veille à ce que les matériels fournis soient en adéquation avec les exigences opérationnelles. Elle supervise également les essais et l'évaluation des équipements dans ses nombreux centres d'expertise et d'essais, garantissant ainsi la fiabilité et l'efficacité des matériels avant leur déploiement.

DGA EV est composée de deux sites : Cazaux et Istres et j'ai travaillé sur le site d'Istres

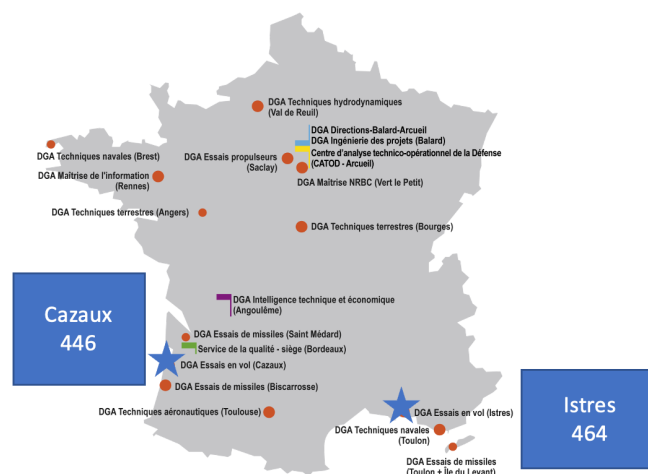


Figure 1: Deux Sites DGA-EV

La DGA (Délégation Générale pour l'Armement) joue un rôle central au sein du Ministère des Armées, sous l'autorité du ministre, aux côtés du CEMA (Chef d'État-Major des Armées) et du SGA (Secrétariat Général pour l'Administration). Elle est chargée d'équiper les forces armées de manière souveraine, de préparer les futurs systèmes de défense, de promouvoir la coopération européenne, de soutenir les exportations et de contribuer à la transformation du ministère. La DGA travaille en collaboration avec les chefs d'état-major des différentes branches des forces armées (terre, marine, air et espace) pour garantir une défense nationale moderne et efficace.



Figure 2: Aéronefs de DGA-EV

Les types d'aéronefs au sein de DGA-EV sont variés, comme on peut le voir sur l'image. On y trouve des avions de chasse comme le Rafale Air et Marine ou le Mirage 2000-5, des hélicoptères comme le NH 90 NFH et le Tigre, ainsi que des avions de surveillance comme l'E-3F AWACS et le Hawkeye. Il y a aussi des avions de transport comme l'A400M, des drones comme le SIDM, et des hélicoptères de sécurité civile comme l'EC 145. Cette diversité de véhicules témoigne de la richesse de DGA-EV, qui ne travaille pas uniquement avec les mêmes types de matériels, mais gère un large éventail d'équipements pour répondre à divers besoins et missions.

## 2.2 Le service informatique

Le service informatique compte actuellement trois administrateurs réseau, deux administrateurs systèmes et une alternante en dernière année de Master. A leurs têtes se trouve trois personnes, une qui s'occupe de trier les demandes d'interventions des clients pour les distribuées aux administrateurs, un administrateur SSI et un chef de projet.

Ma mission était d'assister les administrateurs dans un premier temps pour prendre conscience de l'importance de l'informatique au sein de l'aéronautique. Étant dans une structure de l'État, les missions se sont avérées aussi diverses que variées. En effet, ce service s'occupe de l'infrastructure réseau et système de la DGA et en assure la sécurité. Ils participent également aux projets techniques pour les clients.

Pour des raisons de sécurité, je ne peux pas vous montrer un schéma de l'infrastructure réseau de la DGA. Cependant c'est une architecture en étoile composé principalement de switches Cisco et de quelques Stormshield, qui relie plusieurs bâtiments au sein de la DGA site d'Istres.

## 2.2 Organigramme

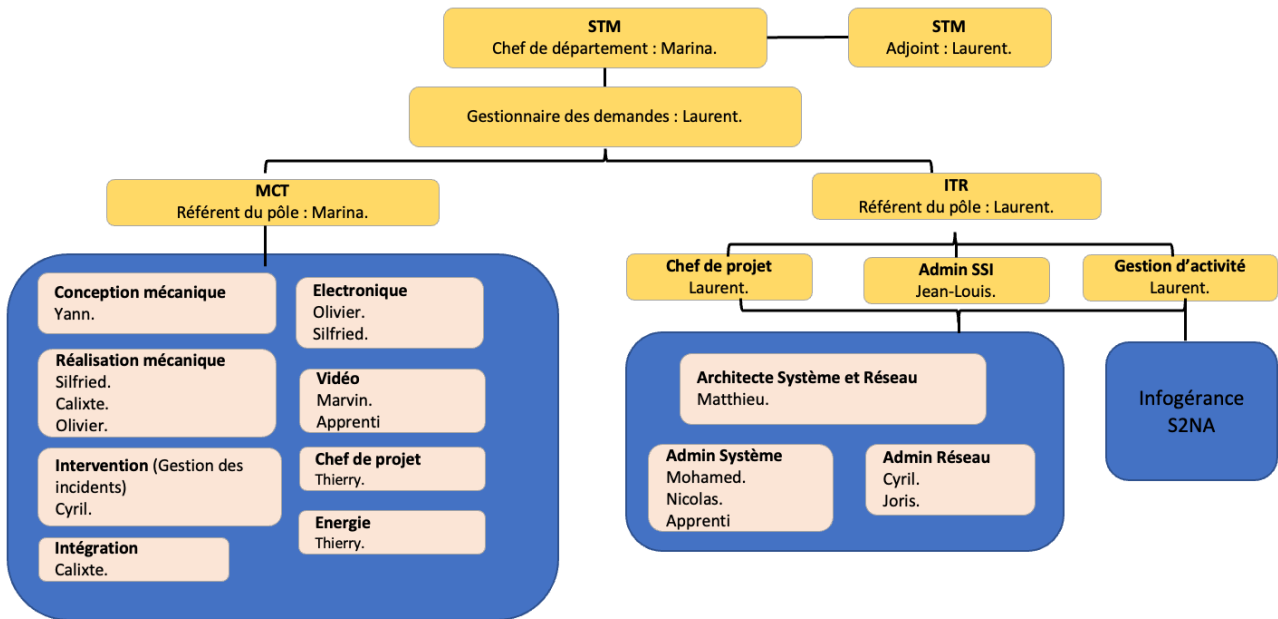


Figure 3: Organigramme

## 3 Projet Automatisation des tâches réseau

En tant que stagiaire au service informatique de la DGA, j'ai eu l'opportunité de développer de nouvelles compétences et découvrir un milieu qui ne m'était pas familier.

Tout au long de ce stage, j'ai pu effectuer diverses missions. Cependant je vais me concentrer sur le projet qui a constitué la partie principale de mon stage et je finirais par diverses tâches que j'ai eu à effectuer.

### 3.1 Contexte

Le contexte dans lequel je travaille est que le réseau est totalement isolé de l'extérieur. Par conséquent, pour installer des logiciels ou autres mises à jour, on ne peut pas simplement installer la solution depuis notre poste de travail. On doit d'abord aller installer la solution sur un poste connecté à Internet, puis passer par une station blanche qui vérifie l'innocuité des éléments installés. Pour ne pas impacter la production, on peut profiter des coupures de courant planifiées pour effectuer ces mises à jour automatiquement.

Dans le cadre d'un projet interne à l'entreprise afin d'optimiser les tâches répétitives des administrateurs réseau, j'ai été amené à mettre en place une solution d'automatisation de mise à jour des EAR. L'objectif de ce projet est la maintenance des équipements réseaux tout en réduisant les erreurs humaines du a une répétition des tâches à faire sur tous les équipements réseaux. Dans la mise en place de cette solution j'ai dû prendre en compte le fait que je n'ai pas pu travailler sur le réseau interne pour des raisons de sécurité, j'ai donc créé mon propre réseau, et le fait que je sois isolé d'internet m'as amené quelques problématiques au niveau de l'installation de ma solution.

### 3.2 Objectif

Dans le cadre d'un projet interne à l'entreprise, afin d'optimiser les tâches répétitives des administrateurs réseau, j'ai été amené à mettre en place une solution d'automatisation de mises à jour des EAR, Équipements Autonome de Réseau. L'objectif de ce projet est la maintenance des équipements réseaux tout en réduisant les erreurs humaines dû à une répétition des tâches à faire sur tous les équipements réseaux.

Lors de la mise en place de cette solution, j'ai dû prendre en compte le fait que je n'ai pas pu travailler sur le réseau interne pour des raisons de sécurité. J'ai donc créé ma plateforme de test. Le fait que je sois isolé d'internet m'a amené quelques problématiques au niveau de l'installation de ma solution.

### 3.3 Maquette

Dans cette partie, nous allons voir les principaux équipements utilisés qui m'ont permis de configurer l'automatisation et de simuler un réseau sur lequel faire de l'administration.

#### Switch

##### Catalyst C2960-CX Series

Le Catalyst C2960-CX est un switch de la marque Cisco permettant ainsi de faire communiquer entre eux les PC. L'interface du C2960-CX est simple, puisque c'est une interface comme sur tous les autres switchs ou routeurs. La configuration se fait en invite de commande.

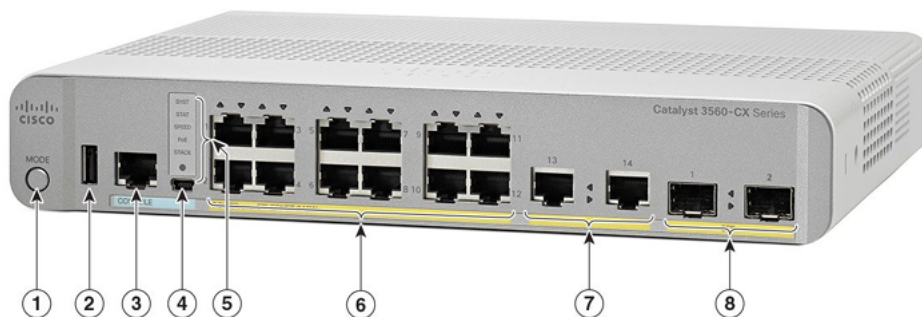


Figure 4 : Catalyst C2960-CX Series

1	Bouton mode	5	System LEDs
2	Port USB Type-A	6	10/100/1000 POE+ ports
3	Port Console RJ-45	7	10/100/1000 uplink ports
4	Port USB mini-Type B (console)	8	SFP module slots

#### Routeur

##### Cisco 891F

Le routeur Cisco 891F permet de faire le routage des données et de faire le rôle de passerelle entre mes deux réseaux. Pour simuler un accès à internet, j'ai dû mettre en place deux Vlans :

- Le vlan 2 viens joué le rôle d'un réseau WAN, auquel j'ai attribué la plage d'adresse 192.168.2.0/24

- Le vlan 3 représente mon réseau d'administration avec comme adresse 192.168.1.0/24. Il est composé de ma machine serveur pour ma solution d'automatisation, une machine serveur ESXI avec VMware qui permet l'installation de machines virtuelles.

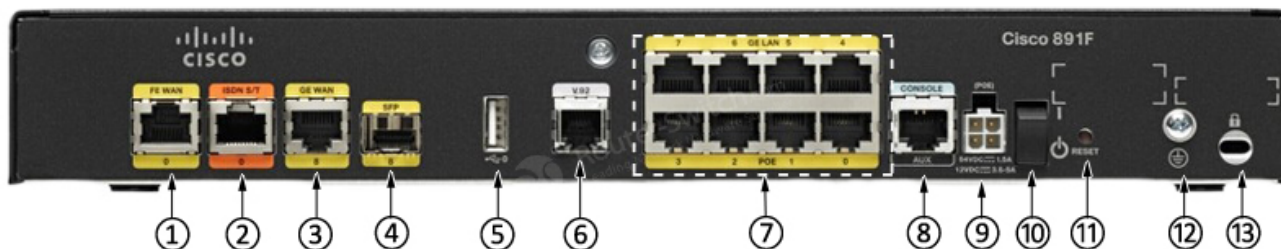


Figure 5: Routeur Cisco 891F

1	Port WAN de sauvegarde – FE WAN	8	Port console/auxiliaire
2	ISDN	9	Connecteur d'alimentation
3	Port Wan principal – GE WAN	10	Interrupteur marche/arrêt
4	SFP	11	Bouton de réinitialisation
5	Port USB	12	Connexion a la terre
6	Sauvegarde V.92	13	Emplacement de sécurité Kensington
7	Commutateur Ethernet 10/100/1000	/	/

## Hub

### Allied Telesyn AT-FS708

Le hub Allied Telesyn AT-FS708 est un dispositif de réseau conçu pour fournir des fonctionnalités de connectivité Ethernet simples et fiables. Ce modèle fait partie de la série de produits Allied Telesyn, réputée pour sa robustesse et ses performances.

Le AT-FS708 est équipé de 8 ports Ethernet Fast Ethernet (10/100 Mbps), il a aussi des ports RJ-45 pour une connectivité standard avec des câbles Ethernet.

Ce Hub est conçu pour être simple à installer avec une configuration plug-and-play, c'est à dire qui ne nécessite pas de configuration complexe. Les ports sont capables d'auto-négocier la vitesse et le mode duplex (half ou full duplex) permet une connectivité optimale.

Le boîtier est compact et robuste, ce qui le rend adapté pour une installation dans des environnements de bureau, comme sur des switches ou routeurs. Chaque port dispose de LED pour indiquer l'état de la connexion et l'activité, ce qui facilite le diagnostic et la gestion du réseau.

Le hub Allied Telesyn AT-FS708 est donc particulièrement adapté pour les petits bureaux qui nécessitent une connectivité réseau de base, aussi pour les extensions de réseau déjà existants sans pour autant nécessiter de matériel complexe. Il permet à des environnements qui ont besoins de solutions réseau d'être économiques et fiables.

Dans mon cas, ce hub était déjà installé quand je suis arrivé dans la salle, je l'ai donc laissé même si la présence d'un switch le rend obsolète.



Figure 6: Hub Allied Telesyn

## ESXI VMware

VMware ESXi est un hyperviseur de type 1, développé par VMware. Il est conçu pour gérer de manière efficace et sécurisée les ressources matérielles d'un serveur physique afin d'héberger plusieurs machines virtuelles (VM) sur un même matériel physique.

ESXi est un hyperviseur de type 1, cela signifie qu'il s'exécute directement sur le matériel physique sans nécessiter de système d'exploitation hôte. Du fait de sa proximité avec le matériel, l'ESXi offre des performances supérieures et une meilleure gestion des ressources par rapport aux hyperviseurs de type 2. Il utilise un noyau propriétaire, le VMkernel, qui gère les ressources CPU, mémoire, stockage et réseau pour les machines virtuelles.

L'interface principale de l'ESXi est le vSphere Client ce qui permet aux administrateurs de configurer, gérer et surveiller les VMs. Pour la gestion de la mémoire, l'ESXi utilise des techniques comme le partage transparent de la mémoire, le ballooning, et la compression de mémoire pour optimiser l'utilisation des ressources. En ce qui concerne l'allocation CPU et réseau, il est géré dynamiquement pour s'adapter aux besoins des VM en temps réel.

Dans mon cas, j'ai utilisé cette ESXi afin de déployer VMware pour créer et gérer des machines virtuelles dans le but d'installer et de configurer un Active Directory qui joue un rôle crucial dans la gestion des utilisateurs, des groupes et des politiques de sécurité sur un réseau.

Ainsi grâce à cet ESXi, j'ai pu tester si la solution que j'allais choisir allé pouvoir administrer des EAR mais également des postes de travail comme des machines Ubuntu ou Windows.

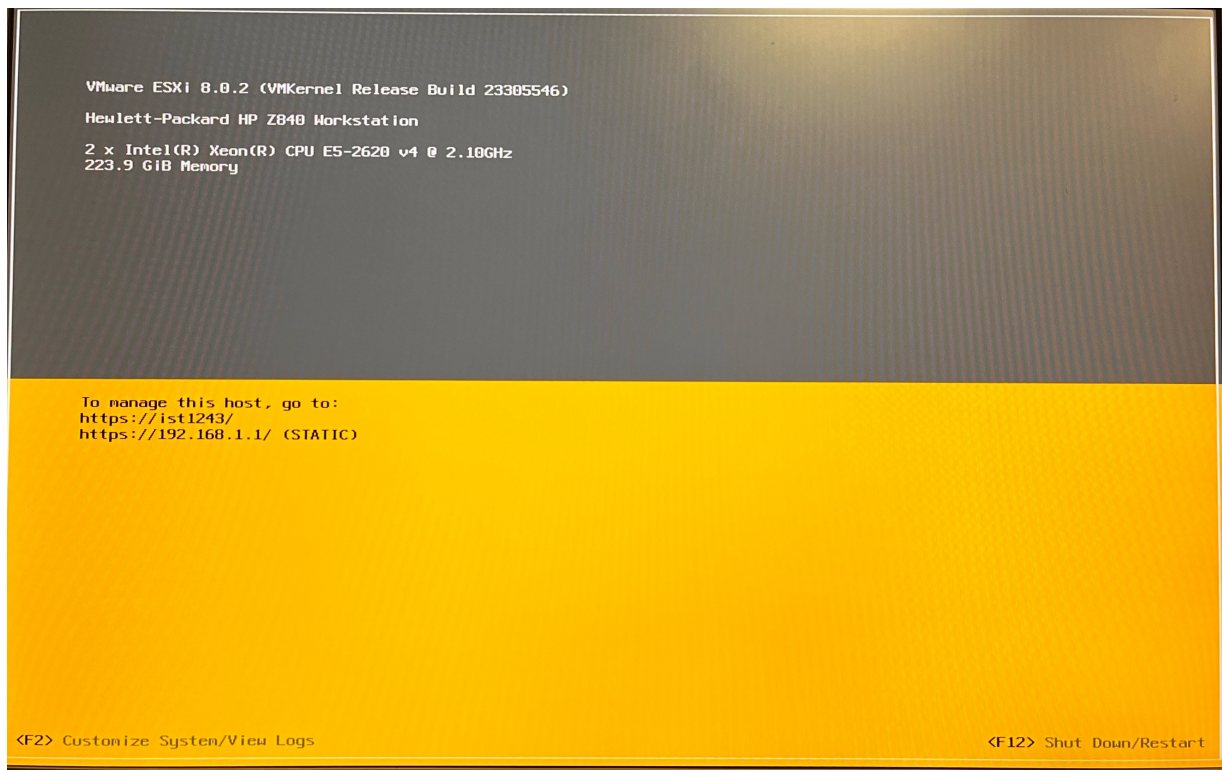


Figure 7 : ESXi VMware

### 3.4 Solutions

En ce qui concerne l'automatisation des tâches, il fallait trouver plusieurs solutions pour pouvoir les comparer et choisir la meilleur solution.

J'ai dû étudier trois logiciels d'automatisation afin de déterminer lequel serait le plus adapté à notre contexte :

- Ansible
- Cisco DNA
- Cisco Prime infrastructure

**Ansible :**



Figure 8: Logo Ansible

Ansible est un outil open-source d'automatisation informatique qui permet de gérer des configurations, de déployer des applications et d'automatiser des tâches répétitives de manière simple et efficace. Développé par Red Hat, Ansible utilise une approche sans agent, basée sur SSH (pour Linux/Unix) ou WinRM (pour Windows), ce qui simplifie grandement son déploiement et son utilisation. Ansible ne nécessite pas l'installation d'agents sur les nœuds gérés, il utilise SSH pour se connecter aux machines Linux/Unix et WinRM pour les machines Windows, ce qui réduit grandement la complexité et les besoins en maintenance.

Les fichiers de configuration d'Ansible sont écrits en YAML, un langage lisible par l'homme, ce qui facilite la compréhension et l'écriture des playbooks. Le playbooks sont des fichiers d'exécution dans

lesquels on va écrire des suites d'instructions à faire sur les machines cibles. Afin de faciliter la construction d'un playbook, Ansible inclut une vaste bibliothèque de modules prêts à l'emploi pour divers aspects de l'infrastructure.

L'application des configurations va se faire de manière idempotente, ce qui garantit que l'état final du système correspond à l'état désiré, ce qui réduit les risques de modifications inattendues ou de configurations incohérentes.

Ansible supporte à la fois les fichiers d'inventaires statiques et dynamiques, permettant de gérer facilement les environnements évolutifs ou basés sur le cloud. Les machines doivent être listées dans le fichier inventaire, elles peuvent être regroupées et des variables spécifiques peuvent être définies pour chaque groupe ou hôte, offrant une flexibilité dans la gestion des configurations.

Au niveau du réseau Ansible permet :

### 1 **Gestion de Configuration :**

- **Automatisation :** Ansible permet d'automatiser la configuration des serveurs, des réseaux et des services, assurant la cohérence et la standardisation.
- **Centralisation :** Les configurations sont centralisées dans des playbooks, facilitant la gestion et le déploiement à grande échelle.

### 2 **Déploiement d'Applications :**

- **Déploiement Automatisé :** Ansible peut automatiser le déploiement d'applications, de l'installation à la configuration en passant par la gestion des dépendances.
- **Mises à Jour et Patches :** Les mises à jour et les correctifs peuvent être appliqués de manière cohérente et contrôlée à travers tout l'environnement.

### 3 **Orchestration :**

- **Tâches Multi-Tiers :** Ansible peut orchestrer des tâches complexes impliquant plusieurs niveaux de l'infrastructure, tels que le déploiement d'une application nécessitant des modifications dans les bases de données, les serveurs web et les pare-feux.
- **Flux de Travail :** Les playbooks peuvent être enchaînés pour créer des flux de travail automatisés, améliorant l'efficacité opérationnelle.

**Cisco DNA :**



**Figure 9 : Logo Cisco DNA**

Cisco Digital Network Architecture (Cisco DNA) est une plateforme de gestion réseau innovante développée par Cisco, conçue pour simplifier, sécuriser et automatiser les opérations réseau. Cisco DNA combine les technologies logicielles et matérielles pour offrir une gestion centralisée, une visibilité approfondie et des capacités d'automatisation étendues pour les réseaux d'entreprise.

Le cœur de Cisco DNA est le Cisco DNA Center, une plateforme centralisée qui offre des fonctionnalités de gestion, d'automatisation et de contrôle des réseaux. C'est une interface utilisateur conviviale et intuitive qui permet aux administrateurs réseau de gérer facilement les configurations, les politiques et les performances, il y a aussi une orchestration des flux de travail réseau pour simplifier la gestion des tâches complexe et répétitives.

Cisco DNA offre une visibilité proactive sur les performances et la santé du réseau grâce à des outils d'analyse avancés, une détection automatique des problèmes et recommandation de solutions permet aux utilisateurs finaux de ne pas être affectés. Au niveau de la sécurité il y a une segmentation des réseaux pour isoler les différentes parties du réseau et réduire la surface d'attaque, il y a aussi l'application des politiques de sécurité cohérentes à travers l'ensemble du réseau pour protéger les données et les utilisateurs. Cisco DNA permet une gestion intégrée des environnements multicloud, facilitant la migration et la gestion des charges de travail dans le cloud.

Au niveau du réseau Cisco DNA permet :

### 1 Simplification de la Gestion Réseau :

- **Centralisation** : Centralisation de la gestion des configurations réseau, des mises à jour logicielles et des politiques de sécurité.
- **Automatisation** : Réduction de la complexité opérationnelle grâce à l'automatisation des tâches répétitives.

### 2 Amélioration de la Performance et de la Fiabilité :

- **Assurance** : Surveillance continue et proactive de la performance du réseau pour détecter et résoudre les problèmes avant qu'ils n'affectent les utilisateurs.
- **Optimisation** : Utilisation de l'analyse pour optimiser les performances réseau en temps réel.

### 3 Renforcement de la Sécurité :

- **Segmentation et Politiques** : Application de politiques de sécurité granulaires et segmentation des réseaux pour protéger les données sensibles.
- **Détection des Menaces** : Surveillance continue pour détecter et répondre aux menaces de sécurité

Cisco Prime Infrastructure :



Figure 10: Logo Cisco Prime Infrastructure

Cisco Prime Infrastructure est une solution complète de gestion réseau développée par Cisco, destinée à fournir une gestion unifiée des réseaux filaires et sans fil. Elle offre une visibilité et un

contrôle centralisés des infrastructures réseau, facilitant ainsi la gestion, le dépannage et l'optimisation des performances réseau.

Cisco Prime Infrastructure permet de gérer à la fois les réseaux filaires et sans fil à partir d'une seule interface, il fournit des tableaux de bord centralisés pour surveiller et gérer l'ensemble de l'infrastructure réseau. Cette solution offre une visibilité complète sur le réseau, incluant les appareils, les utilisateurs, les applications et les services. Il utilise des outils d'analyse avancés pour surveiller les performances du réseau et identifier les goulots d'étranglement et les problèmes potentiels. Il peut aussi automatiser le déploiement des mises à jour et des correctifs logiciels, assurant que les appareils sont toujours à jour et sécurisés. Cisco Prime Infrastructure permet de définir et d'appliquer des politiques de sécurité à travers tout le réseau pour protéger les données et les utilisateurs. Il permet aussi de gérer les politiques de contrôle d'accès pour garantir que seuls les utilisateurs et appareils autorisés peuvent accéder aux ressources réseau. Les tâches de maintenance et de dépannage peuvent aussi être gérées par cette solution comme fournir des outils de diagnostic pour identifier et résoudre rapidement les problèmes réseau ou encore de générer des rapports détaillés et des alertes pour informer les administrateurs des performances du réseau et des incidents.

Au niveau du réseau Cisco Prime Infrastructure permet :

### 1 **Gestion de la Configuration et de l'Inventaire :**

- **Centralisation :** Centralisation de la gestion des configurations réseau et de l'inventaire des appareils, facilitant la maintenance et les audits.
- **Consistance :** Assure la consistance des configurations à travers tous les appareils réseau.

### 2 **Optimisation des Performances :**

- **Analyse Proactive :** Analyse proactive des performances réseau pour identifier et résoudre les problèmes avant qu'ils n'affectent les utilisateurs.
- **Optimisation Continue :** Utilise des analyses en temps réel pour optimiser continuellement les performances du réseau.

### 3 **Sécurité et Conformité :**

- **Politiques de Sécurité :** Définition et application de politiques de sécurité pour protéger les données sensibles et garantir la conformité aux réglementations.
- **Audit :** Génération de rapports d'audit pour assurer que les configurations et les politiques de sécurité sont respectées.

Bien que Cisco Prime Infrastructure offre une gestion complète et centralisée des réseaux filaires et sans fil, cette solution ne sera pas retenue pour notre déploiement au Ministère des Armées. En effet, Cisco ne soutient plus activement cette solution, ce qui pose des risques en termes de sécurité et de support technique. Dans un environnement critique comme le nôtre, où la fiabilité et la sécurité sont primordiales, il est impératif d'utiliser des solutions activement soutenues par leurs éditeurs. Par conséquent, nous devons nous tourner vers des alternatives offrant un support et des mises à jour régulières pour garantir la sécurité et l'efficacité de notre réseau interne.

### 3.5 Choix de la solution

Pour notre déploiement au Ministère des Armées, nous avons opté pour Ansible comme solution de gestion et d'automatisation réseau. Cette décision repose sur plusieurs facteurs clés qui font d'Ansible le choix le plus adapté à nos besoins. Tout d'abord, Ansible est une solution flexible et robuste qui peut gérer et automatiser les configurations à la fois sur les équipements réseau Cisco et HP, offrant ainsi une grande interopérabilité et simplifiant la gestion de notre infrastructure hétérogène. En outre, Ansible utilise une approche sans agent et des fichiers de configuration en YAML, ce qui facilite son déploiement, son utilisation et sa maintenance.

Contrairement à Cisco DNA, dont la licence ne pouvait pas être acquise à temps pour ma période de stage, Ansible est une solution open-source qui permet une mise en œuvre rapide et économique. Cette flexibilité nous a permis de progresser rapidement dans la mise en place et l'automatisation des tâches critiques, assurant ainsi une gestion efficace et sécurisée de notre réseau pendant la durée du projet.

Un autre avantage majeur d'Ansible est sa communauté en ligne dynamique et active. En cas de message d'erreur ou de problème technique, il est possible de rechercher des solutions sur Internet, et il y aura presque toujours quelqu'un qui aura rencontré le même problème et partagé une solution. Cette ressource communautaire est inestimable pour résoudre rapidement les problèmes et optimiser l'utilisation de l'outil.

De plus, Ansible se révèle être encore plus avantageux dans le contexte de notre réseau. En effet, dans l'éventualité d'un projet interne visant à migrer notre infrastructure réseau de Cisco vers HP, Ansible s'avère être la solution idéale. Sa capacité à gérer et automatiser les configurations sur des équipements de différents fournisseurs garantit une transition en douceur et minimise les interruptions de service. Par conséquent, Ansible se révèle être la solution idéale pour répondre à nos exigences opérationnelles et sécuritaires tout en respectant les contraintes temporelles et budgétaires du ministère.

#### **Documentation Ansible**

Dans le cadre de mon projet, j'ai rédigé une documentation intitulée Ansible pour les Nuls. Cette documentation a été conçue pour expliquer de manière simple et détaillée le travail que j'ai réalisé avec Ansible. Elle est destinée à tous ceux qui souhaitent comprendre et utiliser Ansible, même sans connaissances préalables. Vous y trouverez des instructions pas à pas sur la création de playbooks, la mise en place d'un inventaire, la configuration d'Ansible, ainsi que des solutions aux problèmes courants que j'ai rencontrés. De plus, elle inclut des guides d'installation et des exemples pratiques pour faciliter votre apprentissage.

#### **Explication des codes (changement de mot de passe et mise à jour)**

**Lister les différentes tâches que j'ai automatiser avec ansible en plus de ce qui m'était demander.**

## 1. Mise à jour des firmwares

### Description

L'automatisation de la mise à jour des firmwares sur des switches et routeurs permet de garantir que tous les appareils fonctionnent avec la version logicielle la plus récente, ce qui est essentiel pour des raisons de sécurité et de performance. Le processus implique plusieurs étapes, notamment le transfert du firmware vers les appareils, la vérification de l'intégrité du fichier, et la configuration du firmware comme image de démarrage.

### Étapes du processus

- 1 **Collecte d'informations utilisateur** : Utilisation de prompts pour obtenir les détails du firmware et les hôtes cibles. Cela permet à l'administrateur de spécifier les informations nécessaires sans modifier le script.
- 2 **Configuration initiale** : Désactivation du timeout sur les lignes VTY pour éviter que les sessions ne se ferment automatiquement, garantissant ainsi que les commandes puissent s'exécuter sans interruption.
- 3 **Sauvegarde de la configuration actuelle** : Exécution de la commande `show running-config` pour sauvegarder la configuration actuelle des appareils. Cela permet de revenir à la configuration précédente en cas de problème.
- 4 **Copie du firmware** : Transfert du fichier firmware depuis une source définie par l'utilisateur vers les appareils.
- 5 **Vérification d'intégrité** : Validation du fichier firmware transféré en utilisant une vérification MD5. Cela garantit que le fichier n'a pas été corrompu pendant le transfert.
- 6 **Configuration du démarrage** : Paramétrage du firmware comme image de démarrage pour que l'appareil utilise le nouveau firmware après le redémarrage.
- 7 **Redémarrage de l'appareil** : Exécution de la commande de redémarrage sur les appareils et attente de leur retour en ligne.
- 8 **Nettoyage** : Réactivation du timeout sur les lignes VTY et désactivation du serveur SCP pour sécuriser l'appareil après le processus de mise à jour.

## 2. Changement des mots de passe sur les switches et routeurs

### Description

Le changement périodique des mots de passe est crucial pour la sécurité des appareils réseau. Cette tâche automatisée assure que tous les appareils utilisent des mots de passe à jour et conformes aux politiques de sécurité de l'organisation.

### Étapes du processus

- 1 **Collecte d'informations utilisateur** : Utilisation de prompts pour obtenir les hôtes cibles et le nouveau mot de passe. Cela permet à l'administrateur de spécifier les informations nécessaires sans modifier le script.

- 2 **Application du mot de passe** : Mise à jour du mot de passe sur les lignes VTY des switches et routeurs spécifiés. Cela garantit que les accès aux appareils sont protégés par le nouveau mot de passe.

### 3. Redémarrage des routeurs et switches

#### Description

Le redémarrage des appareils réseau est parfois nécessaire pour appliquer les mises à jour ou les modifications de configuration. Automatiser ce processus permet de minimiser les temps d'arrêt et de s'assurer que tous les appareils redémarrent de manière coordonnée.

#### Étapes du processus

- 1 **Collecte d'informations utilisateur** : Utilisation de prompts pour obtenir les hôtes cibles. Cela permet à l'administrateur de spécifier les informations nécessaires sans modifier le script.
- 2 **Redémarrage des appareils** : Exécution de la commande de redémarrage sur les appareils spécifiés. L'automatisation assure que tous les appareils redémarrent sans intervention manuelle.

### 4. Automatisation avec Cron

#### Description

L'utilisation de Cron pour planifier et automatiser l'exécution des playbooks Ansible permet de s'assurer que les tâches critiques, telles que la mise à jour des firmwares, sont effectuées de manière régulière et sans intervention manuelle. Cela augmente la fiabilité et la régularité des opérations réseau.

#### Exemple de script de Cron

Le script de Cron suivant montre comment automatiser l'exécution des playbooks Ansible :

- 1 **Lancement du premier playbook** : Le script exécute le playbook de mise à jour du firmware et enregistre les résultats dans un fichier log.
- 2 **Vérification du succès** : Si le premier playbook réussit, un message de succès est enregistré et le script attend quelques minutes avant de lancer le second playbook.
- 3 **Lancement du second playbook** : Le second playbook est exécuté pour redémarrer les appareils, et le résultat est également enregistré.
- 4 **Gestion des erreurs** : Si le premier playbook échoue, un message d'erreur est enregistré et le second playbook n'est pas exécuté.

### 5. Tâches supplémentaires automatisées

#### Collecte de configurations et de logs

Pour approfondir la compréhension d'Ansible et améliorer la gestion des appareils réseau, plusieurs tâches supplémentaires ont été automatisées, notamment la collecte de configurations et de logs. Cela permet aux administrateurs de disposer de données précieuses pour le dépannage et la gestion des appareils.

### Exemple :

- **show arp** : Collecte des tables ARP des appareils et enregistrement des résultats dans des fichiers.
- **show running-config** : Collecte des configurations en cours d'exécution et enregistrement des résultats dans des fichiers.

### Création d'un utilisateur

La création d'utilisateurs sur les appareils réseau est une tâche courante pour gérer les accès. L'automatisation de cette tâche permet de s'assurer que les nouveaux utilisateurs sont ajoutés de manière cohérente et sécurisée.

### Exemple :

- **Création d'un utilisateur** : Collecte des informations d'utilisateur via des prompts et création de l'utilisateur avec les permissions appropriées sur les appareils spécifiés.

### Vérification des logs

La vérification des logs est essentielle pour le dépannage et la surveillance des appareils réseau. L'automatisation de cette tâche permet de s'assurer que les logs sont collectés et stockés régulièrement pour une analyse ultérieure.

### Exemple :

- **show logging** : Collecte des logs des appareils et enregistrement des résultats dans des fichiers.

## Fonctionnement d'Ansible :

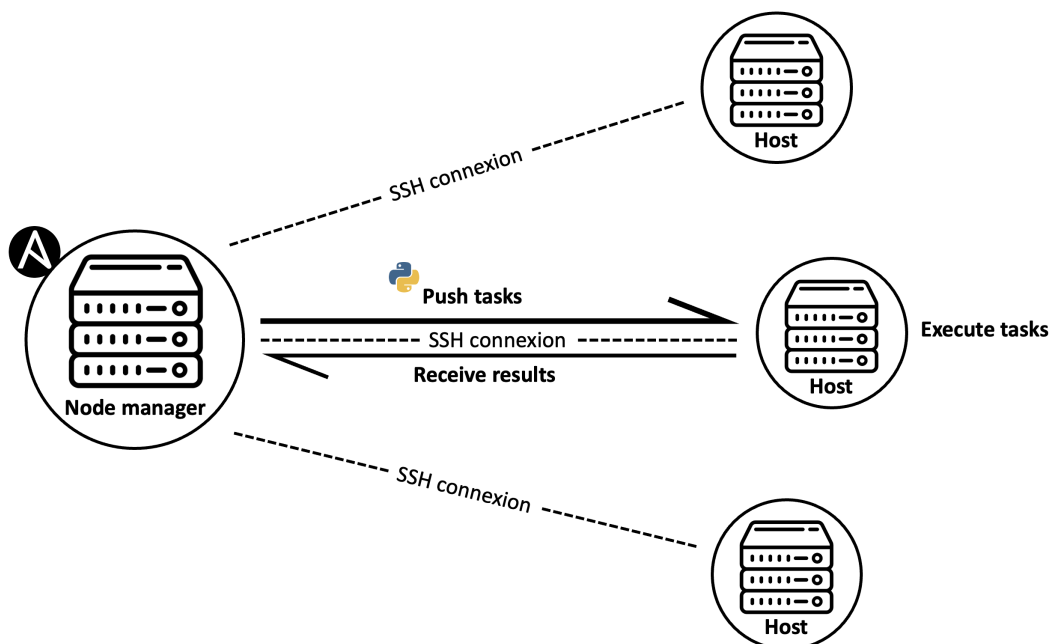


Figure 11: Schéma Fonctionnement Ansible

## Description du Schéma

Le schéma présente une architecture typique de déploiement Ansible avec les éléments suivants :

### 1 Node Manager :

- C'est le serveur central où Ansible est installé et d'où les tâches sont orchestrées. Le Node Manager contient les playbooks, qui sont des scripts définissant les tâches à exécuter.

### 2 Hosts :

- Ce sont les machines cibles sur lesquelles les tâches définies dans les playbooks Ansible seront exécutées. Les hosts peuvent être des serveurs physiques, des machines virtuelles, ou des dispositifs réseau.

### 3 SSH Connexion :

- Ansible se connecte aux hosts via SSH (Secure Shell). C'est un protocole de communication sécurisé qui permet au Node Manager de pousser les tâches et de recevoir les résultats.

### 4 Push Tasks :

- Les tâches définies dans les playbooks Ansible sont envoyées (poussées) aux hosts via SSH. Ces tâches sont écrites en YAML (Yet Another Markup Language), un format de fichier lisible et facile à écrire.

### 5 Execute Tasks :

- Une fois les tâches reçues, les hosts les exécutent localement. Les tâches peuvent inclure des configurations système, des installations de logiciels, des déploiements d'applications, etc.

### 6 Receive Results :

- Après l'exécution des tâches, les résultats sont renvoyés au Node Manager via SSH. Ces résultats sont ensuite analysés pour vérifier le succès ou l'échec des opérations.

## Fonctionnement Détaillé

### 1 Création des Playbooks :

- Les administrateurs rédigent des playbooks Ansible en YAML, décrivant les tâches à exécuter. Par exemple, un playbook peut spécifier l'installation d'un serveur web sur plusieurs machines.

### 2 Connexion via SSH :

- Le Node Manager se connecte aux machines cibles (hosts) en utilisant des connexions SSH sécurisées. Cette connexion peut être authentifiée par mot de passe ou, de manière plus sécurisée, par des clés SSH.

### 3 Envoi des Tâches :

- Une fois connecté, le Node Manager envoie les tâches définies dans les playbooks aux hosts. Ansible utilise le langage Python pour exécuter ces tâches, ce qui nécessite que Python soit installé sur les machines cibles.

### 4 Exécution Locale des Tâches :

- Les machines cibles exécutent les tâches localement. Cela peut inclure des commandes système, des scripts, des configurations de services, des mises à jour de logiciels, etc.

### 5 Retour des Résultats :

- Après exécution, les résultats des tâches (succès, échecs, erreurs) sont renvoyés au Node Manager. Ansible collecte et présente ces résultats sous forme de rapports, permettant aux administrateurs de vérifier l'état des opérations.

## Avantages d'Ansible

### 1 Sans Agent :

- Ansible ne nécessite pas l'installation d'agents spécifiques sur les machines cibles, ce qui simplifie grandement la gestion et la maintenance.

### 2 Simplicité et Flexibilité :

- L'utilisation de YAML rend les playbooks lisibles et faciles à écrire. Ansible est flexible et peut gérer une large gamme de tâches et de systèmes.

### 3 Sécurité :

- Utilise SSH pour des communications sécurisées entre le Node Manager et les hosts.

### 4 Communauté Active :

- Ansible bénéficie d'une large communauté en ligne. Pour chaque problème ou message d'erreur, il y a de fortes chances qu'une solution soit déjà disponible sur les forums ou les documentations communautaires.

## 4 Tâches de suivis des administrateurs

### 4.1 Installation de machines Ubuntu 22.04 pour les développeurs

#### Contexte et Sécurité

Dans le cadre de mon travail au sein de la base, la sécurité est une priorité absolue. Pour garantir la protection des informations et des systèmes, notre réseau est totalement isolé du monde extérieur, y compris d'Internet. Cela implique que pour installer des paquets sur une machine, nous devons d'abord les télécharger sur un disque depuis une autre machine connectée à Internet, puis les transférer et les installer sur le dépôt interne de notre réseau. Ce dépôt est mis à jour régulièrement pour permettre l'installation des paquets comme si nous avions un accès direct à Internet.

Cependant, en raison de la nature de mon travail, je suis isolé du réseau principal dans une salle spécifique. Pour contourner cette contrainte, j'ai mis en place ma propre plateforme de test comprenant un routeur, un switch et une Active Directory (AD), comme mentionné plus haut dans le rapport.

#### Installation des Machines Ubuntu

##### Préparation du Matériel

Pour commencer, nous avons installé deux disques SSD de 1 To chacun sur les deux machines destinées aux développeurs. La partition des disques a été réalisée de la manière suivante :

1. **Partition pour le GRUB :** Une petite partie du disque a été dédiée au bootloader GRUB. Cette partition est nécessaire car GRUB est le programme qui permet de démarrer le système d'exploitation et doit être placé dans un espace séparé pour garantir le bon fonctionnement du démarrage.
2. **Partitions Principales :** Le reste de l'espace libre a été partagé en deux autres partitions pour le système et les données.

##### Installation d'Ubuntu 22.04

Nous avons utilisé une clé USB bootable pour installer Ubuntu 22.04 sur les deux machines. La procédure d'installation a suivi les étapes standard pour l'installation de nouvelles machines sur notre réseau, incluant les configurations réseau spécifiques et autres paramètres requis.

##### Clonage et Déploiement

Une fois la première machine entièrement configurée et opérationnelle, nous avons créé une image de ce système à l'aide de l'outil Clonezilla. Cette image a ensuite été clonée sur la deuxième machine. Après le clonage, nous avons modifié les configurations réseau et le nom de la machine pour éviter les conflits sur le réseau.

##### Mise en Place Physique

Une fois l'installation et les configurations terminées, nous avons déplacées les machines à l'étage et nous les avons installées dans la salle des développeurs.

##### Intégration au Réseau et à l'AD

Pour finaliser l'intégration des machines dans l'environnement de travail :

1. **Configuration Réseau :** Nous avons modifié les paramètres réseau pour les adapter au réseau principal.

2. **Configurations Kernel et AD :** Des ajustements ont été nécessaires au niveau du kernel et de l'Active Directory pour assurer la compatibilité et le bon fonctionnement des machines sur le réseau sécurisé.

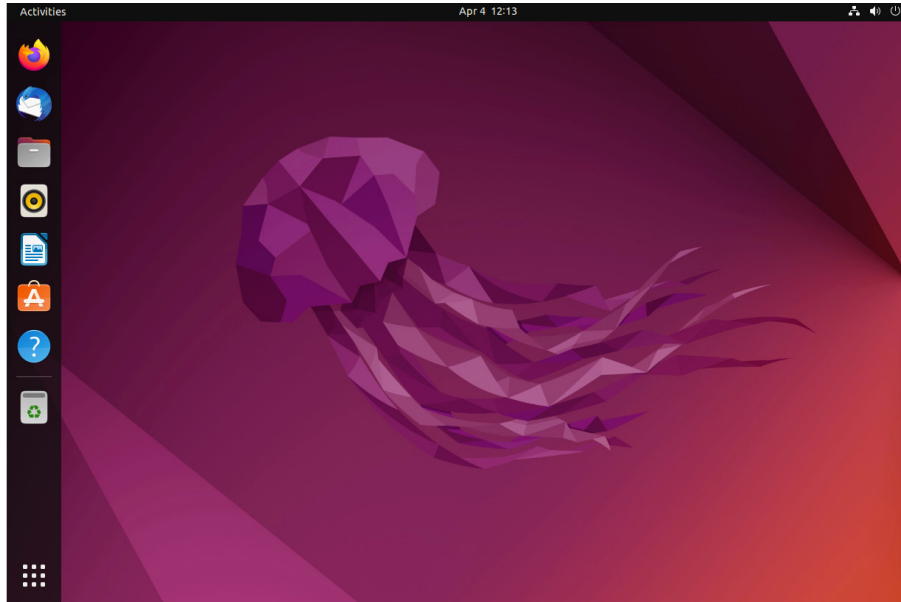


Figure 12: Ubuntu 22.04

### Principe de Clonezilla et son fonctionnement

Clonezilla est un logiciel libre et open-source de clonage et de sauvegarde de disques et de partitions. Il fonctionne en créant une image exacte du disque ou de la partition, incluant le système d'exploitation, les applications installées, les configurations et les données. Cette image peut ensuite être utilisée pour restaurer le système en cas de panne, de migration ou de duplication sur d'autres machines.

Le processus de clonage avec Clonezilla se déroule en plusieurs étapes : tout d'abord, on démarre la machine à cloner à partir d'un support bootable contenant Clonezilla (comme une clé USB ou un CD). Ensuite, Clonezilla copie bit par bit le contenu du disque source vers une image stockée sur un autre support ou directement sur un autre disque. Cette image peut ensuite être restaurée sur un disque cible, reproduisant exactement l'état du disque source au moment de la création de l'image. Clonezilla utilise des techniques de compression pour réduire la taille des images et accélérer le processus de clonage, tout en garantissant une haute-fidélité de la duplication.

## 4.2 Utilisation de Clonezilla pour le clonage et la sauvegarde de disques

Dans le cadre de mes activités, j'ai fréquemment utilisé Clonezilla pour cloner divers types de machines ainsi que pour sauvegarder des disques de données, afin de pouvoir les restaurer ultérieurement sur d'autres machines. Cet outil est essentiel pour garantir la continuité des opérations et la sécurité des données.

Voici un rapport détaillé de mon expérience avec Clonezilla, en mettant l'accent sur les défis rencontrés et les leçons apprises.

## Principe de Fonctionnement de Clonezilla

Clonezilla est un logiciel libre et open-source spécialisé dans le clonage et la sauvegarde de disques et de partitions. Il permet de créer une image exacte du disque ou de la partition, y compris le système d'exploitation, les applications, les configurations et les données. Cette image peut ensuite être utilisée pour restaurer le système, migrer des données vers de nouvelles machines ou créer des duplications exactes des systèmes existants.

Processus d'Utilisation de Clonezilla :

### • Création d'un Support Bootable :

- Clonezilla doit être exécuté à partir d'un support bootable, comme une clé USB ou un CD. Le démarrage de la machine cible se fait alors à partir de ce support pour accéder à l'interface de Clonezilla.

### • Sélection du Disque Source et Destination :

- Lors de la création d'une image, il est crucial de bien choisir le disque source (le disque à cloner) et le disque de destination (où l'image sera sauvegardée). De même, pour restaurer une image, le disque source sera celui contenant l'image sauvegardée et le disque de destination sera celui à restaurer.

### • Création d'une Image :

- Pour sauvegarder une machine, on commence par sélectionner le disque de destination où l'image sera stockée. Une fois cette sélection faite, Clonezilla copie bit par bit le contenu du disque source et le compresse pour créer une image, réduisant ainsi l'espace de stockage nécessaire et accélérant le processus.

### • Restauration d'une Image :

- Pour restaurer un système, il faut sélectionner le disque source contenant l'image sauvegardée, puis choisir le disque de destination qui recevra cette image.

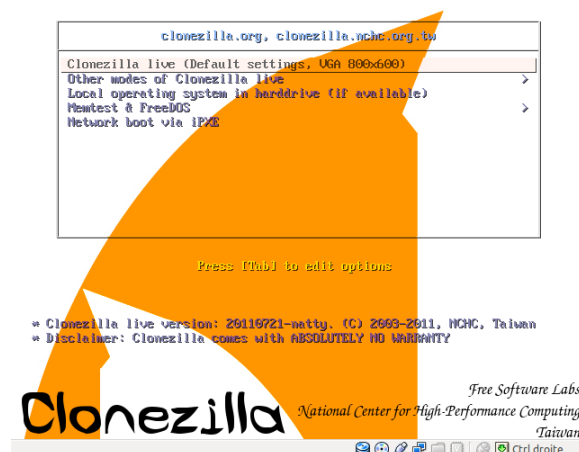


Figure 13: Clonezilla

### 4.3 Suivi du problème du Robot de sauvegarde Dell Quantum

Dans le cadre de mon stage, j'ai eu l'opportunité de suivre l'équipe système sur un problème critique concernant un robot de sauvegarde de données Dell Quantum, situé dans un bâtiment distinct du nôtre. Ce robot est essentiel pour la gestion et la sécurité des données sur le réseau. Initialement, le robot refusait de reconnaître les cassettes de sauvegarde. Pour diagnostiquer le problème, nous avons inséré plusieurs cassettes afin de vérifier si des messages d'erreur étaient générés, mais aucune anomalie n'a été détectée à ce stade.

Cependant, quelques temps plus tard, le robot a cessé de s'allumer correctement, et il n'affichait aucune information. De plus, les administrateurs n'arrivaient plus à se connecter au robot via le réseau, rendant toute tentative de diagnostic à distance impossible. Face à cette situation, les administrateurs ont pris la décision de remplacer la puce Compact Flash (CF) du robot, espérant ainsi résoudre le problème matériel. Malheureusement, ce changement n'a eu aucun effet et le robot continuait de ne pas fonctionner.

Cette mission m'a permis de mieux comprendre les défis techniques liés à la maintenance des équipements critiques et l'importance de la résilience et de la redondance dans les systèmes de sauvegarde de données.



Figure 14: Quantum Scalar i6

## 5 Conclusion

Cette expérience enrichissante fut pour moi une première réelle expérience professionnelle qui m'a permis de découvrir le monde de l'entreprise ainsi qu'un cadre de travail différent. De plus, mon stage s'est déroulé dans un milieu qui ne m'était pas familier : le secteur aéronautique. En effet, au cours de ma formation, nous avons étudié et développé des compétences sur les réseaux et les systèmes informatiques, qui m'ont été très utiles dans le cadre de mon stage. Cependant, le secteur de la défense et de l'aéronautique avec leurs contraintes spécifiques était pour moi tout nouveau.

Pouvoir évoluer dans un milieu tel que celui-ci m'a permis de découvrir différents équipements aéronautiques ainsi que l'importance des systèmes informatiques dans les essais en vol. Avoir la chance d'intégrer une équipe au sein de STM de DGA-EV m'a permis d'acquérir une vision globale allant de la mise à jour des firmwares des EAR à la sécurisation des systèmes en passant par la configuration des dispositifs nécessaires aux différents corps de métier. Cela m'a permis de mieux appréhender les contraintes ainsi que le fonctionnement d'une infrastructure réseau dans un environnement de haute sécurité.

Les différents projets qui m'ont été confiés m'ont permis de mettre en pratique mes compétences, mais aussi de redécouvrir et d'approfondir mes connaissances concernant l'automatisation des tâches réseau et la gestion des systèmes.

Cette immersion dans le secteur aéronautique au sein de DGA-EV m'a non seulement permis de développer des compétences techniques et de renforcer mes capacités en codage, mais aussi de m'ouvrir aux autres en travaillant en étroite collaboration avec les administrateurs systèmes et réseaux. Cette expérience a profondément influencé mes perspectives professionnelles, me dirigeant vers un poste d'administrateur systèmes et réseaux.

Ce souhait s'est concrétisé par une alternance l'année prochaine au sein du même service.



## 6 Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Dans un premier temps, je voudrais exprimer toute ma reconnaissance à ma tutrice de stage, Mme Houssain Corinne, professeure d'anglais à l'IUT R&T d'Aix-Marseille, pour sa disponibilité, ses précieux conseils et sa supervision de mon stage.

Je veux ensuite adresser mes remerciements à mon tuteur de stage **Laurent**, qui est adjoint au chef de département informatique. Il a su partager son expertise, me consacrer du temps et m'as permis de découvrir et d'évoluer dans le milieu aéronautique mais surtout dans le monde de l'entreprise.

Je remercie également Cyril, Joris, Mohammed, Nicolas et Mathieu qui sont des administrateurs pour leur accueil mais aussi pour l'intérêt porter à mon stage en me faisant découvrir d'autres aspects du réseau et du système.

Je tiens aussi a remercié toutes l'équipe encadrante Jean-Louis, Laurent et Marina pour leurs accueil et leurs bienveillances envers moi.



## 7 Glossaire

**BUT**, Bachelor Universitaire de Technologie

**SSI**, sécurité des systèmes d'information

**EAR**, Équipement Autonome de Réseau

**Ballooning**, technique de gestion de la mémoire

**Idempotente**, une méthode idempotente ne doit pas avoir d'effets secondaires



## 8 Sitographie

**DGA [en ligne].** Consulté tout au long du stage, disponible sur :

<https://www.defense.gouv.fr/dga/dga-essais-vol>

**Ansible [en ligne].** Consulté pour développer ma solution, disponible sur : <https://www.ansible.com/>

**RedHat [en ligne].** Consulté pour développer ma solution, disponible sur :

<https://www.redhat.com/fr/topics/automation/learning-ansible-tutorial>

**Github [en ligne].** Consulté pour développer ma solution, disponible sur : <https://github.com/>

**Cisco [en ligne].** Consulté pour les recherches de solution d'automatisation et la recherche des fichier « .bin », disponible sur :

[https://www.cisco.com/site/fr/fr/index.html?team=regional\\_campaign&medium=paid\\_search&campaign=campaign\\_independent&ccid=cc000870&dtid=pseggl000015&\\_bk=cisco&\\_bt=497257208134&\\_bm=e&\\_bn=g&\\_bg=60233088637&ds\\_rl=1290581&gad\\_source=1&ds\\_rl=1290581&gclid=CjwKCAjwps-zBhAiEiwALwsVYbuLDmpjdw7SMUbOwRywxlrtPaziu2sH9T3DqR9dnmsTqb5Qak42ORoCbQAQAvD\\_BwE&gclsrc=aw.ds](https://www.cisco.com/site/fr/fr/index.html?team=regional_campaign&medium=paid_search&campaign=campaign_independent&ccid=cc000870&dtid=pseggl000015&_bk=cisco&_bt=497257208134&_bm=e&_bn=g&_bg=60233088637&ds_rl=1290581&gad_source=1&ds_rl=1290581&gclid=CjwKCAjwps-zBhAiEiwALwsVYbuLDmpjdw7SMUbOwRywxlrtPaziu2sH9T3DqR9dnmsTqb5Qak42ORoCbQAQAvD_BwE&gclsrc=aw.ds)

**Clonezilla [en ligne].** Consulté tout au long du stage, disponible sur : <https://clonezilla.org/>

**Ubuntu [en ligne].** Consulté tout au long du stage, disponible sur : <https://www.ubuntu-fr.org/>





**Institut Universitaire de Technologie,  
Aix-Marseille Université**

**ANNEXES**

**RAPPORT DE STAGE de fin de deuxième année**  
Bachelor Universitaire de Technologie  
Spécialité Réseaux et Télécommunications  
parcours cybersécurité

**Automatisation des tâches Réseaux**

**Guillaume Halary**

**Direction Générale de l'Armement**

Responsable entreprise : Laurent Laugier

Responsable académique : Corinne Houssain

**2023**



## Code Ansible :

- Fonctionnement d'Ansible :
- Inventaire :

```
---  
Sw:  
  hosts:  
    TWR  
  vars:  
    ansible_connection: network_cli  
    ansible_network_os: ios  
    ansible_port: 22  
    ansible_user: toto  
    ansible_ssh_pass: toto  
Router:  
  hosts:  
    R1  
  vars:  
    ansible_connection: network_cli  
    ansible_network_os: ios  
    ansible_port: 22  
    ansible_user: toto  
    ansbile_ssh_pass: toto
```

## ○ Playbook :

```
---
- name: Faire un show run sur les appareils spécifiés
  hosts: all
  gather_facts: no

  vars_prompt:
    - name: "target_hosts"

      prompt: "Spécifiez les hôtes (séparés par des virgules) sur lesquels exécuter la commande"
      private: no

  tasks:
    - name: "Task 1: Faire un show run sur les appareils spécifiés"
      ios_command:
        commands: "show running-config"
      register: cli_result
      when: inventory_hostname in target_hosts.split(',')

    - name: "Task 2: Enregistrement de la sortie dans un fichier"

      copy:
        content: "{{ cli_result.stdout[0] }}"
        dest: "/etc/ansible/result_commande/not_clear/result-shrun_{{ inventory_hostname }}.txt"
      when: inventory_hostname in target_hosts.split(',')

    - name: "Task 3: Transformer la sortie avec sed"

      delegate_to: localhost
      shell: sed 's/\n/\n/g' /etc/ansible/result_commande/not_clear/result-shrun_{{ inventory_hostname }}.txt > /etc/ansible/result_commande/result_en_clair_de_shrun_{{ inventory_hostname }}.txt
      when: inventory_hostname in target_hosts.split(',')
```

## ○ Exécution d'un Playbook :

```
Spécifiez les hôtes (séparés par des virgules) sur lesquels exécuter la commande: TWR
PLAY [Faire un show run sur les appareils spécifiés] *****

TASK [Task 1: Faire un show run sur les appareils spécifiés] *****
skipping: [R1]
ok: [TWR]

TASK [Task 2: Enregistrement de la sortie dans un fichier] *****
skipping: [R1]
ok: [TWR]

TASK [Task 3: Transformer la sortie avec sed] *****
skipping: [R1]
changed: [TWR]

PLAY RECAP *****
R1                : ok=0    changed=0    unreachable=0    failed=0    skipped=3    rescued=0
ignored=0
TWR                : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

- Mise à jour des EAR.

```
---
- name: Copier le nouveau firmware sur les Switchs et Routeurs
  hosts: all
  gather_facts: no
  become: yes

  vars_prompt:
    - name: "target_hosts"
      prompt: "Spécifiez les hôtes (séparés par des virgules) sur lesquels faire ce changement"
      private: no
    - name: "firmware_src"
      prompt: "Spécifiez le chemin source du firmware"

tasks:
  - name: Désactiver le timeout sur les lignes VTY
    ios_config:
      lines:
        - "line vty 0 4"
        - "exec-timeout 30 0"
      when: inventory_hostname in target_hosts.split(',')

  - name: Task1 Faire un show run sur les appareils spécifiés
    ios_command:
      commands: "show running-config"
      register: cli_result
      when: inventory_hostname in target_hosts.split(',')

  - name: Task2 Enregistrement de la sortie dans un fichier
    copy:
      content: "{{ cli_result.stdout[0] }}"
      dest: "/etc/ansible/result_commande/not_clear/result-shrun_{{ inventory_hostname }}.txt"
      when: inventory_hostname in target_hosts.split(',')

  - name: Task3 Transformer la sortie avec sed
    delegate_to: localhost
    shell: sed 's/\\n/\\n/g' /etc/ansible/result_commande/not_clear/result-shrun_{{ inventory_hostname }}.txt > /etc/ansible/result_commande/result_en_clair_de_shrun_{{ inventory_hostname }}.txt
    when: inventory_hostname in target_hosts.split(',')

  - name: Copier le firmware sur les appareils
    net_put:
      src: "{{ firmware_src }}"
      dest: "{{ firmware_bin }}"
      when: inventory_hostname in target_hosts.split(',')
    vars:
      ansible_command_timeout: 1200
```

```

- name: Vérifier le checksum MD5 du firmware
  ios_command:
    commands:
      - "verify /md5 flash:{{ firmware_bin }}"
  register: md5_output
  when: inventory_hostname in target_hosts.split(',')

- name: Vérifier si le checksum est valide
  assert:
    that: "'{{ md5_hash }}' in md5_output.stdout[0]"
    fail_msg: "Erreur: le fichier firmware est endommagé (checksum MD5 incorrect)."
    success_msg: "Le checksum MD5 est valide."
  when: inventory_hostname in target_hosts.split(',')

- name: Configurer le firmware comme image de démarrage
  ios_config:
    lines:
      - "boot system flash:{{ firmware_bin }}"
  when: inventory_hostname in target_hosts.split(',')

- name: Exécuter la commande show boot
  ios_command:
    commands:
      - show boot
  register: boot_output
  when: inventory_hostname in target_hosts.split(',')

- name: Vérifier si le nouveau firmware est référencé
  assert:
    that: "'flash:{{ firmware_bin }}' in boot_output.stdout[0]"
    fail_msg: "Erreur: le nouveau firmware n'est pas référencé dans le BOOT path-list."
    success_msg: "Le nouveau firmware est correctement référencé dans le BOOT path-list."
  when: inventory_hostname in target_hosts.split(',')

- name: Réactiver le timeout de 5 minutes sur les lignes VTY
  ios_config:
    lines:
      - "line vty 0 4"
      - "exec-timeout 5 0"
  when: inventory_hostname in target_hosts.split(',')

```

## ○ Playbook de redémarrage :

```
###
#Redémarrage du ou des switches / routeurs
#Au lancement le playbook va vous demandez sur quels routeurs / switches faire le redémarrage
###
---
- name: Reboot les Switchs et Routeurs
  hosts: all
  gather_facts: no
  become: yes

  vars_prompt:
    - name: "target_hosts"
      prompt: "Spécifiez les hôtes (séparés par des virgules) sur lesquels faire ce changement"
      private: no

  tasks:
    - name: reboot ios device
      become: yes
      ios_command:
        commands:
          - "reload in 1\ny "
      tags:
        - reload
      vars:
        ansible_command_timeout: 1200

  when: inventory_hostname in target_hosts.split(',')
```

## ○ Scripts pour l'automatisation des Mises a Jour :

### ▪ Script exécution playbook mise à jour :

```
#!/bin/bash
ansible-playbook /etc/ansible/playbooks/jam3.yml --extra-vars "target_hosts=TWR firmware_src=/media/
adminpc/HALARY/c2960cx-universalk9-mz.152-7.E10.bin firmware_dest=flash:/c2960cx-universalk9-
mz.152-7.E/ firmware_bin=c2960cx-universalk9-mz.152-7.E10.bin
md5_hash=e5a4ad0f5fd642d76c74f560de81ad91"
echo "1er playbook finis avec succès" >> /etc/ansible/log.txt
```

- Script exécution playbook redémarrage :

```
#!/bin/bash
ansible-playbook /etc/ansible/playbooks/reboot.yml --extra-vars "target_hosts=TWR"
echo "2eme playbook finis avec succès" >> /etc/ansible/log.txt
```

- Crontab pour planifier les mises à jour :

```
00 14 * * * /etc/cron.ansible/orchestre_playbooks.sh
14 14 * * * /etc/cron.ansible/lancement_reload.sh >> /etc/ansible/log.txt
```