

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de BEJAIA

Faculté des Sciences Exactes

Département d'informatique



Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master

Professionnel en Informatique

Option : Administration et Sécurité des Réseaux

Thème

**Etude et mise en place d'une solution
d'automatisation d'infrastructure Réseau et
système**

Cas d'étude : Tchîn-Lait Candia Bejaia

Réalisé par :

M^{lle}. KAOUD Sonia

et

M^{lle}. SAADA Noura

Devant le jury composé de :

Président

M^m. BATAT Nadia

U. A/Mira Bejaia.

Examineur

M^r. BOUCHEBBAH Fatah

U. A/Mira Bejaia.

Encadrant

M^r. CHEKRID Mohamed

U. A/Mira Bejaia.

Co-Encadrant

M^r. BAROUTDJI Raid

Tchîn-Lait Candia

Année Universitaire : 2023/2024

Remerciement

Avant tout, nous remercions Dieu tout puissant de nous avoir donné le Courage et la patience de terminer ce travail.

Nous souhaitons exprimer toute notre gratitude aux personnes qui ont contribué au succès de notre mémoire et qui nous ont accompagnés tout au long de ce projet. Leurs conseils, leur soutien et leur expertise ont été inestimables.

Nous tenons à remercier chaleureusement Notre directeur de mémoire : M.CHEKRID Mohamed, pour son encadrement, ses retours constructifs et sa disponibilité. Ses conseils ont grandement enrichi notre réflexion

Non sincères remerciements vont à Mr BAROUDJI Raid notre encadrant de l'entreprise Tchén-Lait CANDIA, pour son encadrement avec patience. Son encouragement et ses remarques pertinentes nous ont permis de mieux structurer ce travail. Nous le remercions aussi de nous avoir fait profiter de ses expériences, ses orientations et ses conseils nous ont énormément aidés.

Sans oublier de remercier également, les membres du jury qui ont accepté D'examiner et de juger ce modeste travail.

Dédicace

Je rends grâce au bon Dieu de m'avoir donné la force, la volonté et la sagesse afin de parvenir à cette conclusion de mon cycle.

Dans cet espace je souhaiterai dédier ce travail à mes très chers parents

En premier lieu mes dédicaces vont droit à ma chère mère. Tes encouragements et tes prières ont été d'un grands soutien pour moi je te remercie infiniment.

A mon père, décédé trop tôt, qui m'a toujours poussé et motivé dans mes études, j'aurais bien aimé si il est entre nous maintenant pour voir sa fille réussit, que dieu l'accueil son vaste paradis.

A mes chers frères M'hamed, Lounes et M'heni que je porte dans mon cœur, je vous aime très fort, je ne vous souhaite que de la réussite et que Dieu vous protège inchallah

A ma très chère sœur Kahina et son mari Meziane pour qui je souhaite une vie pleine de joie, réussite et beaucoup de bonheur.

A ma très chère copine et binôme Sonia avec qui j'ai passé des agréables et inoubliables moments.

NOURA

Dédicace

Je rends grâce au bon dieu de m'avoir donné la force, la volonté et la sagesse afin de parvenir à cette conclusion de mon cycle.

Je remercie mon cher père pour sa présence dans ma vie, de son soutien et ces sacrifices. J'espère avoir réussi à te rendre fiers chose que je tache de continuer à faire et que ce travail soit l'accomplissement de tes vœux et le fruit de ton soutien infailible.

A la mémoire de ma mère qui nous a quittées très tôt, qui reste toujours dans mon cœur et mon esprit.

A mes deux sœurs Samia et Katia et son mari Ghilas, pour leur soutien moral et encouragement, je vous souhaite une vie pleine de succès et d'amour.

Aussi mon cher petit frère Islam que j'aime beaucoup, merci pour ta présence dans ma vie.

A ma chère copine et binôme Noura avec qui j'ai passé tout mon parcours universitaire.

Sonia

Table des matières

LISTE DES FIGURES	I
LISTE DES TABLEAUX	III
LISTE DES ABREVIATIONS	IV
INTRODUCTION GENERALE.....	1
I. Chapitre I : Contexte général du projet	3
1. Introduction	3
2. Présentation de l'organisme d'accueil Candia Tchín-Lait	3
3. Localisation de Candia Tchín-Lait.....	3
4. Réseau de distribution du groupe Tchín-Lait.....	4
5. La structure de l'entreprise	4
6. Les missions de l'entreprise.....	5
7. Structure informatique.....	6
8. Les sites de l'entreprise Candia Tchín-Lait.....	6
8.1 Les différents équipements d'interconnexions de site de Bejaia.....	7
9. Présentation de projet.....	8
9.1 Contexte de projet.....	8
9.2 Problématique.....	8
9.3 Objectif de projet	8
9.4 Démarche du projet.....	9
10. Conclusion	10
II. Chapitre II: Automatisation des réseaux informatiques	11
1. Introduction	11
2. Les réseaux informatiques.....	11
2.1 Définition d'un réseau	11
2.2 Définition d'un réseau informatique d'entreprise	11
2.3 Intérêt des réseaux d'entreprise	12
2.4 Types de réseaux informatiques.....	12
2.5 Les topologies des réseaux	13
2.6 Les types d'architectures	14
2.7 Composants matériels d'un réseau d'entreprise	15

TABLE DES MATIERES

2.8	Modèle OSI et TCP/IP	16
3.	Définition d'automatisation des réseaux.....	20
4.	Pourquoi automatiser un réseau ?	21
5.	Avantages d'automatisation d'un réseau	21
6.	Etude sur les outils d'automatisation	22
6.1	Chef	22
6.2	Puppet	23
6.3	Ansible	24
6.4	Comparaison et synthèse	25
7.	La solution proposée	26
7.1	Pourquoi Ansible	26
8.	Conclusion	27
III.	Chapitre III: Généralité sur ANSIBLE	28
1.	Introduction	28
2.	Définition.....	28
3.	Historique.....	28
3.1	Red Hat.....	29
4.	Cas d'usage d'Ansible	29
5.	Architecture et composants	30
5.1	Architecture.....	30
5.2	Composants.....	31
5.3	Comment fonctionne ANSIBLE	34
5.4	ANSIBLE sans agents.....	35
6.	Ansible pour des commandes ad hoc.....	35
7.	Le modèle Jinja2.....	35
8.	Langage de programmation utilisé	36
8.1	Présentation du modèle YAML	36
8.2	Syntaxe YAML.....	37
9.	Conclusion	37
IV.	Chapitre IV: Réalisation du projet.....	38
1.	Introduction	38
2.	Environnement de travail.....	38
2.1	La virtualisation	38
2.2	Avantages de la virtualisation.....	39
2.3	La virtualisation des serveurs	39
3.	Présentation de VMWare Workstation Pro	40

TABLE DES MATIERES

3.1	Installation.....	40
3.2	Installation des machines virtuelles.....	41
4.	Présentation de GNS3	44
4.1	Installation.....	45
5.	Conception de la topologie.....	47
5.1	Installation des équipements Cisco sur GNS3.....	47
5.2	Réalisation de l'architecture réseau	48
5.3	Attribution des adresses IP aux équipements	49
5.4	La configuration de base des équipements	50
5.5	Vérification de la connectivité	52
6.	Installation et configuration d'ANSIBLE	53
6.1	Installation.....	53
7.	Le protocole Secure Shell	56
7.1	SSH sur les équipements Cisco	56
7.2	SSH sur la machine virtuelle Ubuntu.....	57
7.3	Teste de connexion à distance via SSH	57
8.	Déploiement de la solution	58
8.1	Démarche de déploiement	58
8.2	Exécution et vérification	62
9.	Conclusion	65
	Références	68

LISTE DES FIGURES :

Figure I.1 La localisation de l'usine via Google maps	4
Figure I.2 Réseau de distribution du groupe Tchén-Lait	4
Figure I.3 Organigramme générale de Tchén-Lait.....	5
Figure I.4 Architecture réseau globale des 4 sites de l'entreprise	6
Figure II.1 Les types des réseaux informatiques	12
Figure II.2 Les topologies physiques	14
Figure II.3 Les topologies logiques	15
Figure II.4 Les différentes couches du modèle OSI.....	19
Figure III.1 Architecture structurelle d'ANSIBLE	31
Figure III.2 Architecture d'un inventaire [14]	32
Figure III.3 Architecture d'un Playbook [14]	33
Figure III.4 Fonctionnement d'Ansible	34
Figure IV.2 Site d'installation de VMWare Workstation Pro	41
Figure IV.3 L'interface graphique initiale de VMWare Workstation Pro	41
Figure IV.4 Installation de PfSense.....	42
Figure IV.5 La page d'accueil d'authentification.....	43
Figure IV.6 Les interfaces configurées sur le pare-feu	44
Figure IV.7 Installation de Ubuntu sur VMware	44
Figure IV.8 Page de téléchargement de GNS3.....	45
Figure IV.9 L'interface graphique initiale de GNS3	46
Figure IV.10 Importer la VM sur VMWare Workstation	47
Figure IV.11 importé la VM sur GNS3.....	47
Figure IV.12 Liste des machines installée.....	48
Figure IV.13 Topologie du réseau.....	49
Figure IV.14 Création des VLANS	50
Figure IV.15 Affectation des VLANS aux interfaces	51
Figure IV.16 Routage statique des VLANS	52
Figure IV.17 Resultat de Ping (PC2-Pare-Feu).....	52
Figure IV.18 Résultat du Ping (pare-feu - internet)	53
Figure IV.19 installation des mis à jour UBUNTU.....	54
Figure IV.20 Installation du paquet.....	54
Figure IV.21 Ajouter le PPA d'Ansible.....	55
Figure IV.22 Installation d'Ansible	55
Figure IV.23 Vérification d'installation Ansible	56
Figure IV.24 Configuration du protocole SSH sur le switch manager.....	57
Figure IV.25 Installation de l'Open SSH server	57
Figure IV.26 : Test de connectivité via SSH.....	58
Figure IV.27 Contenu d'Ansible	58
Figure IV.28 Le fichier hosts	59
Figure IV.29 Fichier d'inventory	60
Figure IV.30 Contenu de repertoire roles	60

TABLE DES MATIERES

Figure IV.31 TASK de configuration des vlan	61
Figure IV.32 Playbook	62
Figure IV.33 Commande d'exécution du playbook.....	63
Figure IV.34 Résultat de playbook	63
Figure IV.35 Listes des vlans ajouté sur le switch manager	64
Figure IV.36 Liste de modifications sur les ports	64

LISTE DES TABLEAUX :

Tableau I-1 La liste des équipements du site de Bejaia.....	8
Tableau II-1 comparaison des outils	26
Tableau IV-1 Tableau d'adressage	50

LISTE DES ABREVIATIONS :

- **API** : Application Programming Interface
- **ARP** : Address Resolution Protocol
- **ASCII** : American Standard Code for Information Interchange
- **ATM** : Asynchronous Transfer Mode
- **CLI** : Command-Line Interface
- **DNS** : Domain Name System
- **DSL** : Domain-Specific Language
- **EBCDIC** : Extended Binary Coded Decimal Interchange Code
- **FTP** : File Transfer Protocol
- **GNOME** : GNU Network Object Model Environment
- **GNS3** : Graphical Network Simulator-3
- **HTTP** : Hypertext Transfer Protocol
- **IGMP** : Internet Group Management Protocol
- **IMAP** : Internet Message Access Protocol
- **IOS** : Internetwork Operating System
- **IP** : Internet Protocol
- **IPv4** : Internet Protocol version 4
- **IPv6** : Internet Protocol version 6
- **ISO** : Organisation internationale de normalisation
- **IT** : Information Technology
- **JSON** : JavaScript Object Notation
- **LAN** : Local Area Network
- **LDAP** : Lightweight Directory Access Protocol
- **LLC** : Contrôle de Liaison Logique
- **MAC** : Media Access Control (Contrôle d'Accès aux Médias)
- **MAN** : Metropolitan Area Network
- **NetDevOps** : Network Development and Operations
- **NTP** : Network Time Protocol
- **PAN** : Personal Area Network
- **PPA** : Personal Package Archives
- **POP** : Post Office Protocol
- **POS** : Personal Operating Space
- **QoS** : Qualité de service
- **RIP** : Routing Information Protocol
- **RSA** : Rivest-Shamir-Adleman
- **SARL** : Société à Responsabilité Limitée
- **SaaS** : Software as a Service
- **SMTP** : Simple Mail Transfer Protocol
- **SNMP** : Simple Network Management Protocol
- **SSH** : Secure Shell
- **TCP** : Transmission Control Protocol
- **TCP/IP** : Transmission Control Protocol/Internet Protocol
- **UDP** : User Datagram Protocol
- **UHT** : Ultra Haute Température
- **VLAN** : Virtual Local Area Network

Liste des abréviations

- **WAN** : Wide Area Network
- **WinRM** : Windows Remote Management
- **YAML**: Yet Another Markup Language.

Introduction générale

De nos jours, les réseaux modernes se caractérisent par une prolifération d'équipements hétérogènes, engendrant une complexité croissante en matière de configuration et de gestion. Au-delà de la simple configuration, il est désormais crucial de s'adapter aux exigences spécifiques de chaque client. Les responsables réseau et sécurité au sein d'une entreprise sont sans doute confrontés à une multitude de tâches manuelles effectuées via l'interface de ligne de commande (CLI), des outils comme Ansible offrent aux entreprises des solutions puissantes et accessibles même aux utilisateurs non-experts, permettant de répondre efficacement aux besoins croissants en matière de configuration réseau.

L'automatisation, définie comme l'utilisation de technologies pour exécuter des tâches habituellement réalisées par des humains, vise à optimiser l'efficacité et la productivité en accomplissant plus de choses, de meilleure qualité et plus rapidement, avec une intervention humaine minimale, voire nulle. Dans le domaine des réseaux, l'automatisation consiste à automatiser les processus de configuration, de gestion, de test, de déploiement et d'exploitation des périphériques physiques et virtuels composant le réseau. En automatisant les tâches quotidiennes et en gérant les processus répétitifs de manière autonome, l'automatisation du réseau améliore significativement la disponibilité du service.

Concrètement, l'automatisation des réseaux s'appuie sur des logiques programmables pour créer des tâches reproductibles. Ces tâches permettent de gérer les ressources et les services réseau à l'aide de scripts programmés exécutés via l'interface en ligne de commande d'un système d'exploitation ou d'un logiciel d'automatisation dédié. L'avènement des réseaux virtualisés et des services Cloud renforce l'importance de l'automatisation, qui s'impose comme une stratégie indispensable pour les équipes réseau et sécurité. Elle permet de fournir des services plus rapides, plus cohérents et plus sécurisés, tout en limitant l'intervention humaine.

Face à la complexité croissante des infrastructures réseaux, composées de centaines d'équipements, les équipes réseau et sécurité font face à des pressions considérables pour assurer le bon fonctionnement de ces environnements critiques. La surveillance constante est nécessaire pour garantir la performance des infrastructures, tandis que les tâches répétitives quotidiennes peuvent engendrer une surcharge de travail et une fatigue importante. L'automatisation de la gestion des infrastructures permet de réduire considérablement le temps et les efforts consacrés à ces tâches manuelles, tout en minimisant le risque d'erreurs. Ansible s'inscrit précisément dans cette démarche, en offrant un outil puissant pour automatiser ces tâches et optimiser l'efficacité des équipes réseau et sécurité.

Notre mémoire consiste en la conception et la mise en place d'une solution d'automatisation d'infrastructure réseau à l'aide de l'outil Ansible, en prenant comme cas d'études l'entreprise Tchir-Lait Candia de Bejaia. Cette solution permet de gérer les ressources et les services des réseaux, de manière rapide et efficace, à l'aide des scripts programmés. Avec l'ère des réseaux virtualisés et des services Cloud, l'automatisation devient une stratégie plus qu'indispensable pour aider les équipes de réseaux et sécurité à fournir des services offrant plus de rapidité et de sécurité, avec une intervention humaine beaucoup réduite.

Ce mémoire se compose de quatre chapitres :

Introduction

Le premier chapitre fournit une présentation détaillée de l'organisme d'accueil, offrant ainsi une compréhension complète du contexte dans lequel s'inscrit ce projet ;

Le deuxième chapitre est consacré à l'automatisation des réseaux, en mettant en lumière les concepts clés des réseaux informatiques et les divers outils d'automatisation et leurs comparaisons. Cette analyse comparative a permis de justifier le choix d'Ansible comme solution optimale utilisée dans notre mémoire ;

Le troisième chapitre offre une vue d'ensemble sur Ansible, présentant ses principales caractéristiques, ses avantages, et son fonctionnement ;

Le dernier chapitre est consacré à la mise en œuvre pratique de la solution proposée. Cette étape comprend plusieurs éléments, à savoir la virtualisation, l'environnement de travail, l'installation des équipements réseau, et à la conception de la topologie réseau. La simulation présentée en fin de chapitre démontre l'efficacité de la solution proposée.

I. Chapitre I : Contexte général du projet

1. Introduction :

Dans ce chapitre, nous présentons le contexte général du projet. Nous commençons par présenter l'organisme d'accueil, suivi de l'exposition du projet et de sa problématique, ensuite nous abordons les objectifs et la démarche que nous avons suivie tout au long de la réalisation, ainsi que la planification mise en place pour atteindre nos buts.

2. Présentation de l'organisme d'accueil Candia Tch-

Lait :

Tchin-Tchin, à l'origine une entreprise familiale spécialisée dans les boissons gazeuses depuis 1952, a accumulé une vaste expérience dans le conditionnement de produits liquides. La marque CANDIA est présente en Algérie depuis de nombreuses années grâce à ses exportations de lait liquide, qui ont été interrompues en 1998 en raison d'une forte hausse des taxes douanières. Plusieurs industriels algériens se sont spontanément tournés vers CANDIA pour pénétrer le marché du lait.

Le projet de l'entreprise Tch-Lait a attiré l'attention de CANDIA, qui l'a choisi. Implantée sur l'ancien site de la limonadière Tchin-Tchin, à l'entrée de la ville de Bejaia, Tch-Lait produit et commercialise du lait longue conservation UHT (Ultra Haute Température) sous la marque CANDIA. Tch-Lait est une société privée de droit algérien, constituée juridiquement en SARL. Elle est principalement détenue par M. Fawzi BERKATI, gérant de la société.

Les installations des machines ont été réalisées par la société française TetraPak. L'unité est équipée d'un matériel ultramoderne de grande capacité, sous la marque Candia. Des tests de contrôle sont effectués quotidiennement de manière permanente et régulière par le laboratoire Tch-Lait pendant tout le cycle de fabrication. De plus, le lait UHT est conservé pendant 72 heures avant sa commercialisation, garantissant ainsi sa stérilité.



3. Localisation de Candia Tch-Lait :

L'entreprise Candia Tch-Lait construite sur une superficie totale de 6000 m², localisée sur la route nationale N°12 à l'entrée ouest de la ville de Bejaïa (Bir-Slam).



Figure I.1 La localisation de l’usine via Google maps

4. Réseau de distribution du groupe Tchin-Lait :

Candia Tchin-Lait dispose une architecture de distribution suivante :

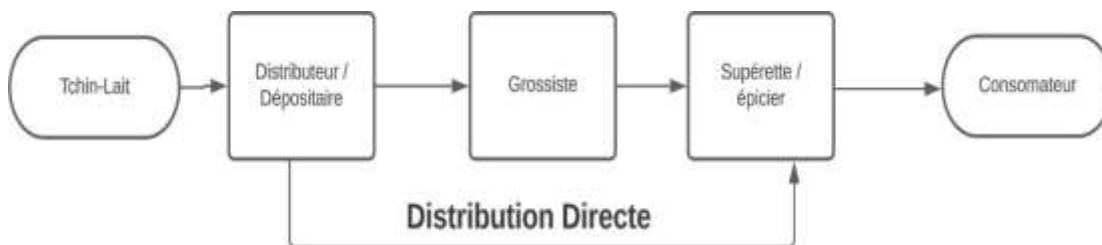


Figure I.2 Réseau de distribution du groupe Tchin-Lait

5. La structure de l’entreprise :

La structure de l’entreprise est subdivisée en plusieurs directions :

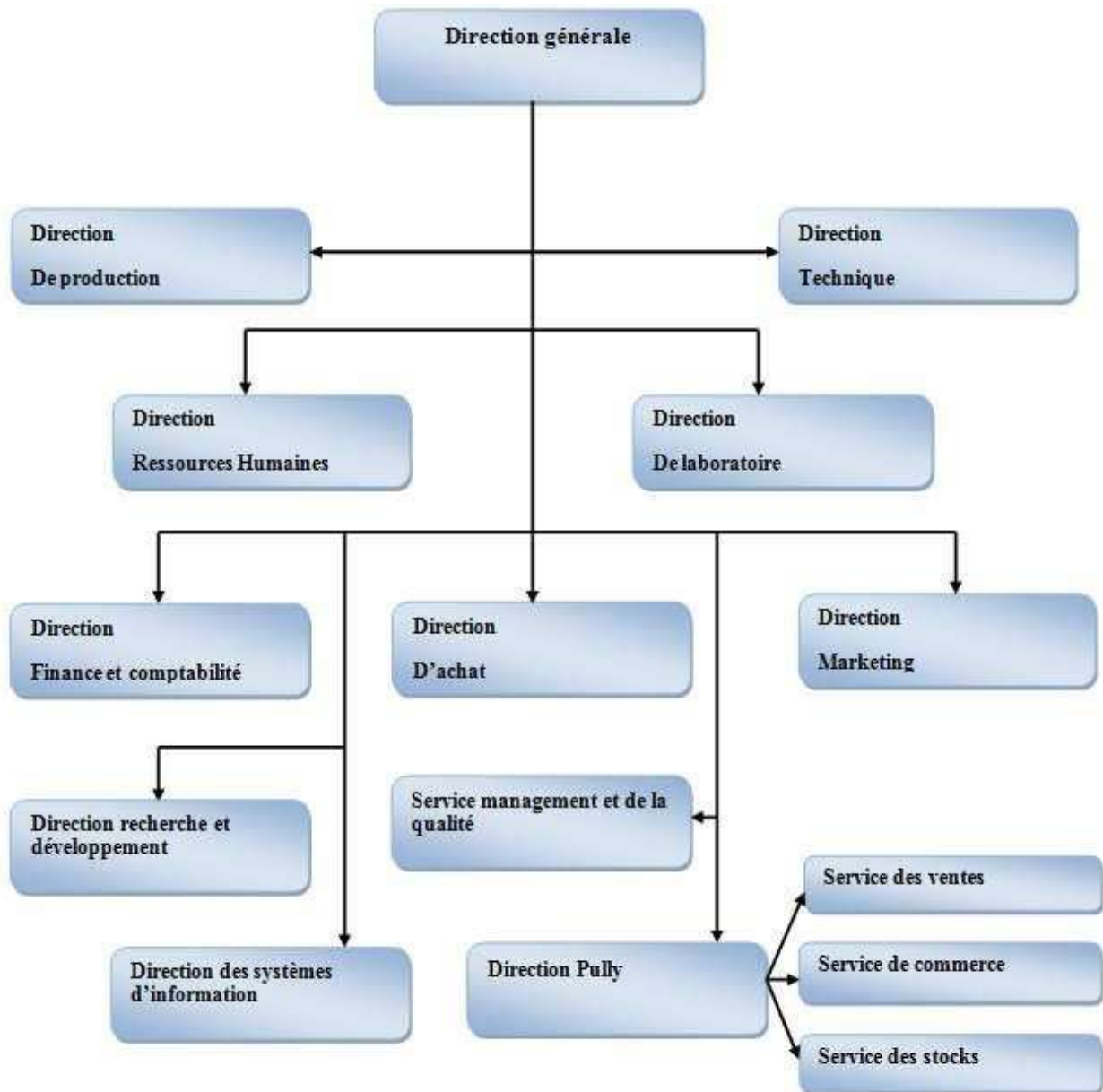


Figure I.3 Organigramme générale de Tchou-Lait

6. Les missions de l'entreprise :

La mission d'entreprise consiste à exprimer la raison d'être de l'entreprise ainsi que la manière dont elle prévoit d'atteindre ses objectifs. Tchou-Lait a pour mission principale de :

- Mobiliser les ressources internes en motivant les employés qui peuvent s'identifier à des valeurs fortes.
- Aligner les décisions et actions prises au quotidien par l'ensemble du personnel.
- Communiquer une image forte et claire aux clients et aux actionnaires de l'entreprise.
- Forcer les managers à se poser des questions fondamentales sur les valeurs et les Comportements qu'ils doivent chercher à promouvoir.

7. Structure informatique :

L'infrastructure informatique d'une entreprise englobe toutes les activités coordonnées liées à la recherche, au traitement, à la distribution et à la protection des informations. Elle met en œuvre les technologies informatiques et les réseaux pour servir le personnel et la clientèle de l'entreprise. Le département informatique est composé des rôles suivants :

- Un chef de département.
- Un administrateur réseau et système.
- Un administrateur de bases de données.
- Un ingénieur support.
- Un ingénieur réseau et système.

8. Les sites de l'entreprise Candia Tchir-Lait :

Candia Tchir-Lait est une grande entreprise qui est composée de plusieurs sites :

Site d'Oued Ghir, Bejaia, Alger, Sétif et Beraki, qui sont tous reliés à Algérie Telecom avec une fibre optique, la figure suivante explique la manière dont les quatre sites sont reliés :

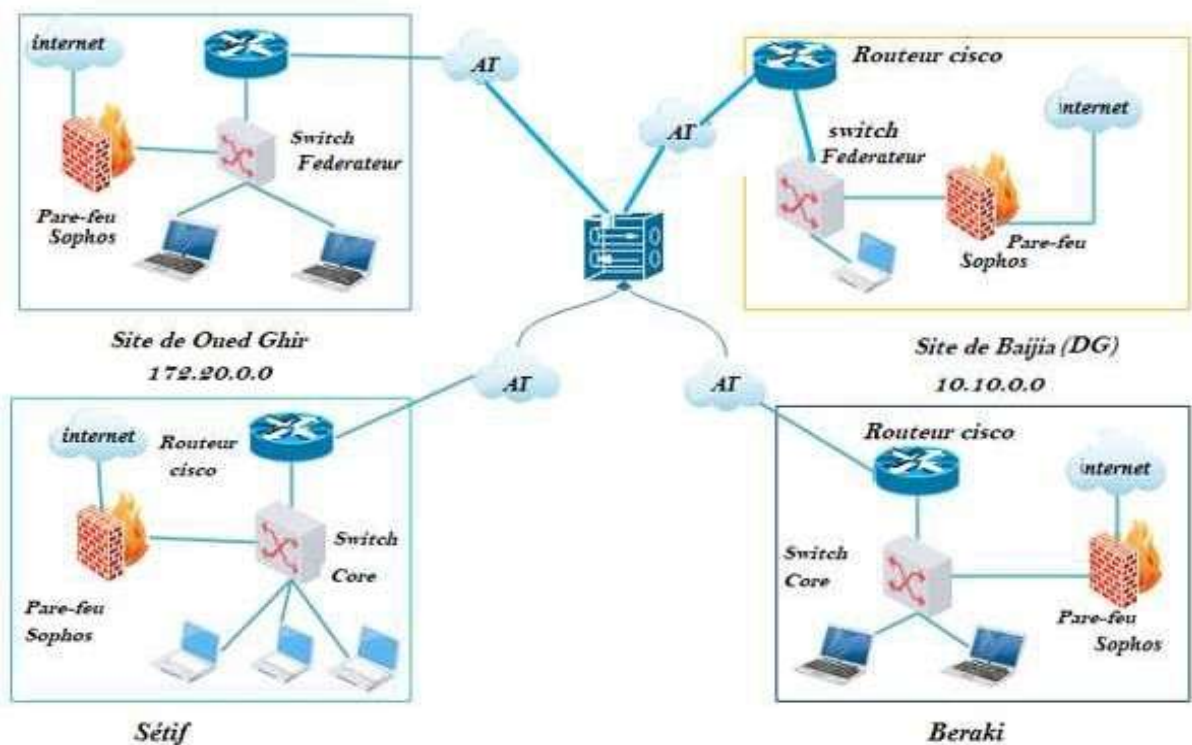


Figure I.4 Architecture réseau globale des 4 sites de l'entreprise

8.1 Les déferents équipements d'interconnexions de site de Bejaia :

Le site de Bejaia dispose un nombre important des équipements d'interconnexion qui sont reliés entre eux, ce tableau ci-dessous montre la liste des switches, routeurs, pare-feu, et contrôleur Wi-Fi :

Equipement	Marque	Modèle	IP	Nombre de port	Emplacement	Remarque
Switch	Cisco	WS-C3750G-48TS-S	10.10.10.1	48	Data Center	Switch Core
Switch	Cisco	WS-C2960X-24PS-L	10.10.10.2	24	Data Center	
Switch	Cisco	WS-C2960-48TC-C	10.10.10.3	48	DRH	
Switch	NetGear	GS724T	10.10.10.4	24	DC	
Switch	Cisco	WS-C3560E-48TD-S	10.10.10.5	48	Technique	
Switch	Cisco	WS-C3750G-12S-S	10.10.10.6	12	Technique	
Switch	NetGear	GS724T	10.10.10.7	24	Salle d'archive	
Switch	Cisco	WS-C3560E-48TD-S	10.10.10.8	48	CDB	
Switch	NetGear	GS724T	10.10.10.9	24	Dépôt PF	
Switch	HP		10.10.10.10	24	Nouveau labo	
Routeur	Cisco	C892FSP-K9	192.168.10.1	2	Data Center	Liaison LS 10 M
Routeur	Cisco	CISCO1941/K9	10.10.100.1		Data Center	Liaison VPV MPLS
Routeur	Cisco	CISCO1941/K9	10.10.200.1	2	Data Center	Liaison VPV WiMax

Contrôleur Wi-Fi	Aruba	Aruba7030	10.10.1.20	2		
Pare-feu	Sophos	XG-330	10.10.128.1		Data Center	

Tableau I-1 La liste des équipements du site de Bejaia

9. Présentation de projet :

9.1 Contexte de projet :

À mesure que les technologies évoluent et que la taille de l'infrastructure réseau augmente, les défis se multiplient. Dans ce contexte, le projet ANSIBLE proposé à l'entreprise Candia vise à répondre aux nouveaux enjeux des réseaux et de la sécurité en simplifiant la gestion du réseau grâce à l'automatisation. Cette approche permet d'éliminer les étapes manuelles liées à la configuration, à la gestion, aux tests, au déploiement et au fonctionnement des périphériques physiques et virtuels au sein du réseau. L'objectif est d'accomplir davantage, plus efficacement et plus rapidement, avec une intervention humaine minimale, voire nulle.

9.2 Problématique :

Lorsque les équipes de Réseaux et Sécurité sont confrontées à des tâches, elles rencontrent certains problèmes qu'elles résolvent manuellement. Ce processus devient ainsi une boucle, ce qui peut entraîner des contraintes de temps et, surtout, des erreurs dans la configuration.

Dans l'ensemble, compte tenu des problèmes mentionnés ci-dessus, il est nécessaire de trouver une solution évolutive capable de résoudre ces problèmes. C'est pourquoi, dans ce projet, nous avons adopté l'outil NetDevOps Ansible.

9.3 Objectif de projet :

Ce projet a pour les objectifs suivants :

1. Objectif de l'étude : Comparer les outils d'automatisation disponibles (qu'ils soient payants ou open source) afin de prendre une décision éclairée pour notre projet.

2. Facilité d'utilisation : Nous cherchons une solution d'automatisation qui puisse suivre le rythme de la gestion de configuration sans ralentir le processus.

3. Analyse du profil existant : Étudier les principes de fonctionnement des outils actuellement utilisés.

4. Mise en place d'Ansible : Déployer et configurer Ansible pour automatiser les tâches quotidiennes au sein de notre grande infrastructure réseau.

5. Importance et avantages : Mettre en évidence les points forts d'Ansible et son impact sur le marché du travail.

6. Sécurité et contrôles : Appliquer les meilleures pratiques de sécurité tout au long du cycle de développement du projet.

9.4 Démarche du projet :

La démarche de réalisation du projet se décompose comme suit :

- **Proposition et étude de la solution Ansible :** Dans cette étape, nous analysons et évaluons la pertinence d'Ansible comme solution pour automatiser la configuration des équipements réseau.
- **Étude des technologies utilisées dans les Pare-feu et IOS (Routeurs et Switches) :** Nous examinons en détail les spécificités techniques des pare-feu sophos et des équipements IOS (routeurs et commutateurs) afin de mieux comprendre leurs fonctionnalités et leurs exigences.
- **Installation d'Ansible sur une distribution Linux Debian :** Nous mettons en place Ansible sur une machine Linux Debian pour pouvoir l'utiliser dans notre environnement de simulation.
- **Automatisation de la configuration des pare-feu :** Nous créons des playbooks Ansible pour automatiser la configuration des pare-feu.
- **Automatisation de la configuration des routeurs :** Nous utilisons Ansible pour automatiser la configuration des routeurs.
- **Automatisation de la configuration des commutateurs :** Nous appliquons également l'automatisation à la configuration des commutateurs.
- **Tests et vérification :** Enfin, nous testons et vérifions que notre solution fonctionne correctement.

10. Conclusion

Dans ce premier chapitre, nous avons minutieusement exploré le contexte global de notre projet de fin d'études. Nous avons commencé par présenter l'organisme d'accueil, puis nous avons plongé dans les détails du réseau sur lequel nous avons concentré nos efforts. Ensuite, nous avons abordé le contexte général du projet, en mettant en lumière sa problématique, ses objectifs et les spécifications du cahier des charges. Enfin, nous avons élaboré une planification pour guider notre progression tout au long du projet.

II. Chapitre II: Automatisation des réseaux informatiques

1. Introduction :

L'automatisation des réseaux informatiques représente une révolution dans la gestion et l'optimisation des infrastructures technologiques. Après avoir exploré le contexte de l'entreprise Candia Tchir-Lait et son environnement opérationnel dans le premier chapitre, ce deuxième chapitre se consacre à l'examen approfondi de l'automatisation des réseaux. Nous aborderons les outils existants qui facilitent cette automatisation, en mettant un accent particulier sur l'outil Ansible pour sa capacité à simplifier les processus complexes et à accroître l'efficacité opérationnelle. Ce chapitre posera les bases nécessaires pour comprendre l'importance cruciale de l'automatisation dans les infrastructures réseau modernes et comment Ansible se positionne comme une solution de choix dans ce domaine en pleine évolution.

2. Les réseaux informatiques :

2.1 Définition d'un réseau :

Un réseau est un ensemble d'entités interconnectées de manière ordonnée. Il permet la circulation d'éléments matériels ou immatériels entre ces entités, en suivant des règles spécifiques. Selon le type d'objets interconnectés, on peut trouver différents types de réseaux tels que les réseaux téléphoniques, les réseaux sociaux, les réseaux électriques et les réseaux informatiques. [1]

2.2 Définition d'un réseau informatique d'entreprise :

Cette infrastructure de communications et d'administration s'appuie sur des services réseaux essentiels tels que les annuaires LDAP, le DNS et le routage du trafic. Dans ce contexte, les ressources en réseau, comme les serveurs, sont responsables de la gestion des accès, du stockage et de la collaboration. [1]

2.3 Intérêt des réseaux d'entreprise :

- Les réseaux informatiques d'entreprise permettent le partage de données et de logiciels entre tous les membres d'une entreprise.
- Ils simplifient également la gestion, la sauvegarde et le stockage des données, ainsi que la configuration des différents droits pour assurer la sécurité.
- Sans le réseau informatique, il serait impossible de fonctionner de manière efficace.
- Par conséquent, les professionnels spécialisés dans ce domaine sont d'une importance cruciale au sein des entreprises de toutes tailles. [1]

2.4 Types de réseaux informatiques :

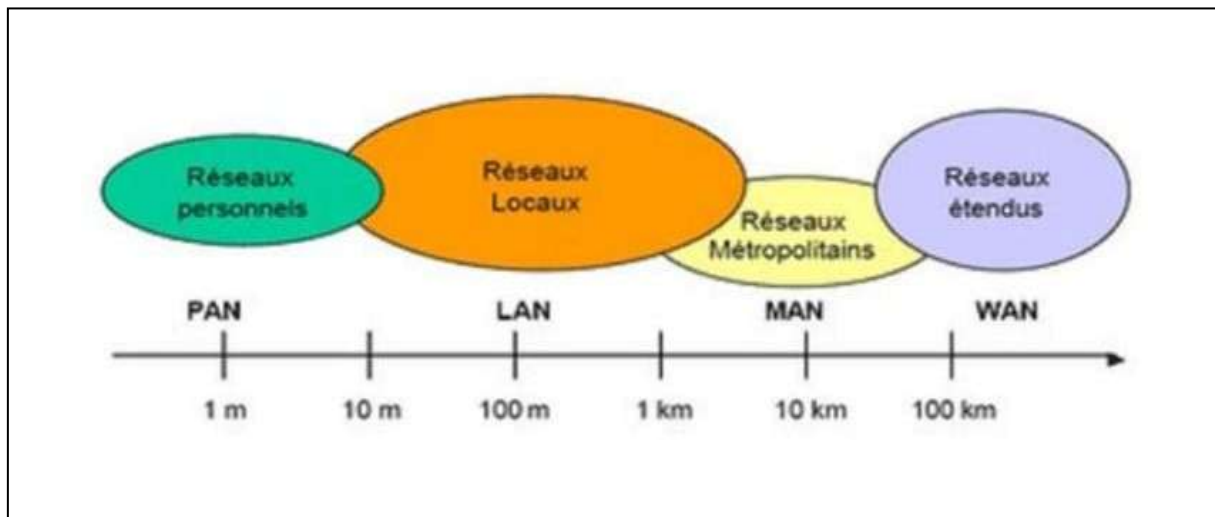


Figure II.1 Les types des réseaux informatiques

2.4.1 Le réseau personnel :

C'est la plus petite étendue de réseau. En anglais, on l'appelle Personal Area Network (PAN). Centré sur l'utilisateur, il désigne une interconnexion d'équipements informatiques dans un espace d'environ une dizaine de mètres autour de celui-ci, appelé Personal Operating Space (POS). D'autres termes pour ce type de réseau sont réseau individuel et réseau domestique. [1]

2.4.2 Le réseau local :

(Local Area Network - LAN), de taille supérieure, s'étend sur quelques dizaines à quelques centaines de mètres. Il relie entre eux des ordinateurs, des serveurs, et d'autres équipements. Le LAN est couramment utilisé pour le partage de ressources communes telles que les périphériques, les données ou les applications. [1]

2.4.3 Le réseau métropolitain :

Le réseau métropolitain (Metropolitan Area Network - MAN), également appelé réseau fédérateur, permet des communications sur de plus longues distances. Il interconnecte souvent plusieurs réseaux LAN et peut servir à relier différents bâtiments distants de quelques dizaines de kilomètres via des liaisons privées ou publiques. [1]

2.4.4 Les réseaux étendus :

Les réseaux étendus (Wide Area Network - WAN), constitués de réseaux de type LAN ou même MAN, sont capables de transmettre des informations sur des milliers de kilomètres à travers le monde entier. Le WAN le plus célèbre est le réseau public Internet, dont le nom provient de son rôle d'interconnexion mondiale. [2]

2.5 Les topologies des réseaux :

Une topologie définit la disposition des différents équipements réseau les uns par rapport aux autres. On distingue deux types de topologies : [2]

2.5.1 Les topologies physiques :

➤ Topologie en bus :

Dans cette configuration, les ordinateurs sont disposés le long d'un câble principal appelé "bus". Le support de transmission utilisé est généralement un câble coaxial. Lorsqu'un ordinateur envoie une information, tous les autres ordinateurs du réseau la reçoivent, mais seule la machine à laquelle l'information est destinée l'utilise.

➤ Topologie en anneau :

Dans cette topologie, les ordinateurs sont connectés en boucle et communiquent les uns après les autres. Les informations circulent dans une direction unique, d'un ordinateur à un autre.

➤ Topologie en étoile :

Dans cette configuration, les ordinateurs du réseau sont reliés à un équipement central appelé concentrateur (ou hub) ou un commutateur (switch). Ce dispositif assure la communication entre les différents ordinateurs connectés à lui.

➤ Topologie en maillage :

Chaque ordinateur est directement relié à tous les autres dans cette topologie. Ainsi, lorsqu'un ordinateur souhaite envoyer une information à un autre, il le fait directement sans passer par un équipement spécifique.

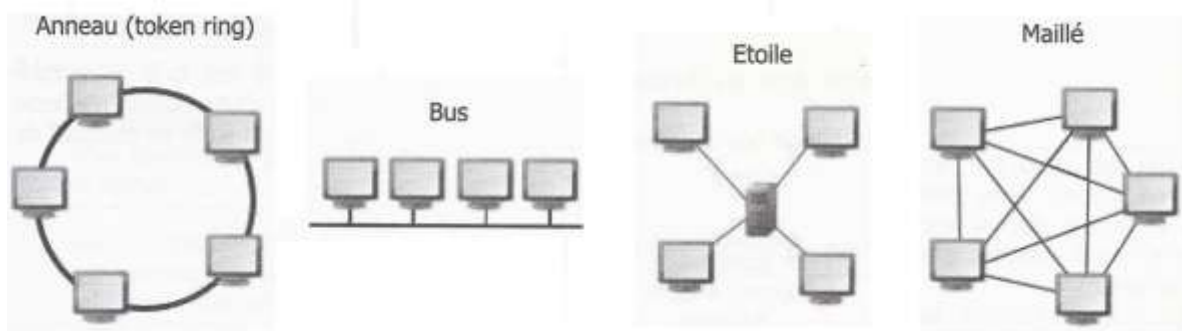


Figure II.2 Les topologies physiques

2.5.2 Les topologies logiques :

Représente la façon selon laquelle les données transitent dans les câbles, les plus courantes sont : Ethernet, Token Ring et FDDI. [3]

2.6 Les types d'architectures :

On distingue également deux catégories de réseaux :

2.6.1 Les réseaux Post à post (peer to peer= P2P) :

Dans un réseau poste à poste, les ordinateurs sont connectés directement les uns aux autres, sans qu'il y ait d'ordinateur central, chaque ordinateur joue à la fois le rôle de serveur et de client comme illustré dans la figure 1.1. L'avantage majeur de ce type d'installation réside dans son faible coût en matériel (chaque poste de travail nécessite uniquement une carte réseau). Cependant, dès que le réseau commence à comporter plusieurs machines, sa gestion devient difficile voire impossible. [4]

2.6.2 Les réseaux client-serveur :

Dans un réseau à architecture client/serveur, tous les ordinateurs (clients) sont connectés à un ordinateur central (le serveur du réseau). Ce serveur est généralement une machine très puissante en termes de capacité. Ce type d'architecture est principalement utilisé pour le partage de connexion Internet et de logiciels centralisés. Il offre une administration plus aisée lorsque le réseau est important, car l'administration est centralisée. Cependant, il nécessite l'utilisation d'un logiciel coûteux spécialisé pour l'exploitation du réseau. [4]

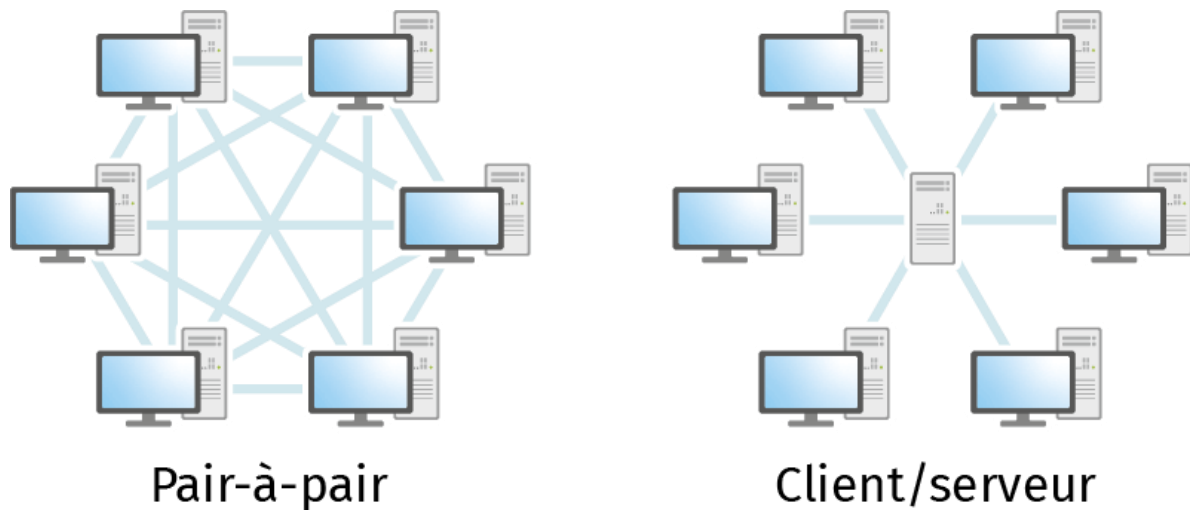


Figure II.3 Les topologies logiques

2.7 Composants matériels d'un réseau d'entreprise :

2.7.1 Equipements d'interconnexion :

Voici les équipements qui peuvent rentrer dans la composition d'un réseau d'entreprise :

➤ Carte réseau :

Est l'interface physique entre l'ordinateur et le câble réseau. Elle permet le transfert des données entre le câble et l'ordinateur sous forme de paquets. Souvent intégrée à la carte mère, elle gère les informations d'emplacement et les données des utilisateurs. [2]

➤ Concentrateur (ou Hub) :

Le concentrateur relie plusieurs ordinateurs entre eux. Son rôle est de diffuser les données sur l'ensemble des ports. Contrairement au commutateur, il ne connaît pas les adresses physiques des machines connectées. [2]

➤ Répéteur (ou Repeater) :

Le répéteur régénère le signal entre deux nœuds du réseau, permettant d'étendre la distance du réseau.

Il est utilisé pour relier des câbles de même type ou de types différents. [2]

➤ Pont (ou Bridge) :

Le pont relie des réseaux travaillant avec le même protocole. Il analyse les adresses de l'émetteur et du destinataire pour diriger les trames vers la machine appropriée. [2]

➤ Commutateur (ou Switch) :

Le commutateur relie également plusieurs ordinateurs entre eux. Contrairement au concentrateur, il connaît l'adresse physique des machines connectées. Il analyse les trames reçues pour les diriger vers la machine de destination [2]

➤ Passerelle :

Une passerelle permet la communication entre différentes architectures réseau. Elle joue le rôle d'un interprète, notamment lorsque deux réseaux sont physiquement connectés mais nécessitent une passerelle pour traduire les données qu'ils échangent. [2]

➤ Routeur :

Le routeur relie différents réseaux et achemine les informations d'un émetteur vers un destinataire en suivant un itinéraire. Il analyse l'en-tête de chaque paquet pour déterminer le meilleur chemin par lequel acheminer le paquet. Le routeur connaît l'itinéraire de tous les segments du réseau grâce aux informations stockées dans sa table de routage. [2]

➤ Modem (modulateur-démodulateur) :

Le modem est un périphérique qui permet de transmettre et de recevoir des données sous forme de signaux. Il transforme les signaux analogiques en signaux numériques et vice versa. Ces signaux sont acheminés via une ligne téléphonique. [2]

2.8 Modèle OSI et TCP/IP :

2.8.1 Le modèle OSI (Open System Interconnexion) :

Le modèle OSI (Open System Interconnexion) a été développé en 1977 dans le but d'éviter que chaque fournisseur de solutions informatiques (réseaux et systèmes) ne propose sa propre implémentation de protocoles liés à des services. L'ISO (Organisation internationale de normalisation) a donc défini le modèle OSI de manière à ce qu'il puisse s'adapter à un large éventail d'implémentations sans favoriser un constructeur spécifique. Le modèle OSI a été intentionnellement conçu de manière abstraite par rapport aux réalités techniques, principalement pour répondre à des besoins de normalisation plutôt qu'à des contraintes techniques spécifiques. Il contient sept couches ces couches remplissant une tâche bien spécifique : [5]

7. Couche d'Application :

La couche d'application du modèle OSI interagit directement avec les applications logicielles pour fournir des fonctions de communication selon les besoins. Elle est la plus proche des utilisateurs finaux.

La fonction principale de la couche d'application consiste à vérifier la disponibilité des partenaires de communication et des ressources nécessaires pour prendre en charge tout transfert de données.

Cette couche définit également des protocoles pour les applications finales, tels que le système de noms de domaine (DNS), le protocole de transfert de fichiers (FTP), le protocole de transfert hypertexte (HTTP), le protocole d'accès aux messages Internet (IMAP), le protocole de bureau

de poste (POP), le protocole de transfert de courrier simple (SMTP), le protocole de gestion de réseau simple (SNMP) et Telnet (une émulation de terminal).

6. Couche de Présentation :

La couche de présentation vérifie les données pour s'assurer qu'elles sont compatibles avec les ressources de communication.

Elle traduit les données dans un format accepté par la couche d'application et les couches inférieures. Elle gère également le formatage des données et les conversions de code, telles que la conversion d'un fichier de texte codé en EBCDIC en un fichier de texte codé en ASCII.

La couche de présentation prend également en charge la compression et le cryptage des données. Par exemple, elle comprime les appels vidéo pour une transmission plus rapide et crypte les données sensibles, comme les mots de passe.

5. Couche de Session :

La couche de session contrôle les dialogues (connexions) entre les ordinateurs.

Elle établit, gère, entretient et met fin aux connexions entre l'application locale et l'application distante.

Les logiciels de la couche de session gèrent également les fonctions d'authentification et d'autorisation. Cette couche vérifie également que les données sont correctement fournies.

Elle est généralement appliquée explicitement dans les environnements d'application qui utilisent des appels de procédure à distance.

4. Couche de Transport :

La couche de transport fournit les moyens de transférer des séquences de données d'une source à un hôte de destination via un ou plusieurs réseaux.

Elle assure la qualité de service (QoS) et garantit la livraison complète des données. Les protocoles TCP et UDP sont essentiels à cette couche.

TCP offre une communication fiable entre dispositifs, tandis qu'UDP fournit un service sans connexion et peu fiable.

3. Couche Réseau :

La couche réseau gère le routage des paquets via des fonctions d'adressage et de commutation logiques.

Chaque nœud sur le réseau a une adresse, et lorsqu'un nœud doit transférer un message à d'autres nœuds, il fournit le contenu du message et l'adresse du nœud de destination.

Si le message est trop long, le réseau peut le diviser en segments, les envoyer séparément et les réassembler au niveau d'un autre nœud.

2. Couche de Liaison de Données :

La couche de liaison de données assure le transfert entre nœuds, c'est-à-dire le lien entre deux nœuds directement connectés.

Elle gère l'encapsulation et la décapsulation des données dans les trames. En d'autres termes, elle enveloppe les données dans un format compréhensible pour les dispositifs réseau et les décompose lors de la réception.

Cette couche définit également le protocole permettant d'établir et de terminer une connexion entre deux dispositifs physiquement connectés. Par exemple, le protocole point à point (PPP) est utilisé pour établir des connexions entre deux équipements via une ligne série.

La couche de liaison de données est généralement divisée en deux sous-couches :

- **La couche de Contrôle d'Accès aux Médias (MAC):** Contrôle la manière dont les appareils d'un réseau accèdent au support de transmission et sont autorisés à transmettre des données.
- **La couche de Contrôle de Liaison Logique (LLC):** Elle est responsable de l'identification et de l'encapsulation des protocoles de la couche réseau. Elle gère également la vérification des erreurs et la synchronisation des trames.

1. Couche Physique :

La couche physique définit les spécifications électriques et physiques de la connexion des données.

Par exemple, elle spécifie la disposition des broches du connecteur, les tensions de fonctionnement d'un câble électrique, les caractéristiques des câbles à fibres optiques et la fréquence des appareils sans fil.

La couche physique est responsable de la transmission et de la réception de données brutes non structurées sur un support physique.

Le contrôle du débit binaire (la vitesse à laquelle les données sont transmises) est effectué au niveau de cette couche.

En résumé, la couche physique concerne les aspects matériels des équipements réseau de base et n'est pas directement liée aux protocoles ou aux éléments des couches supérieures.

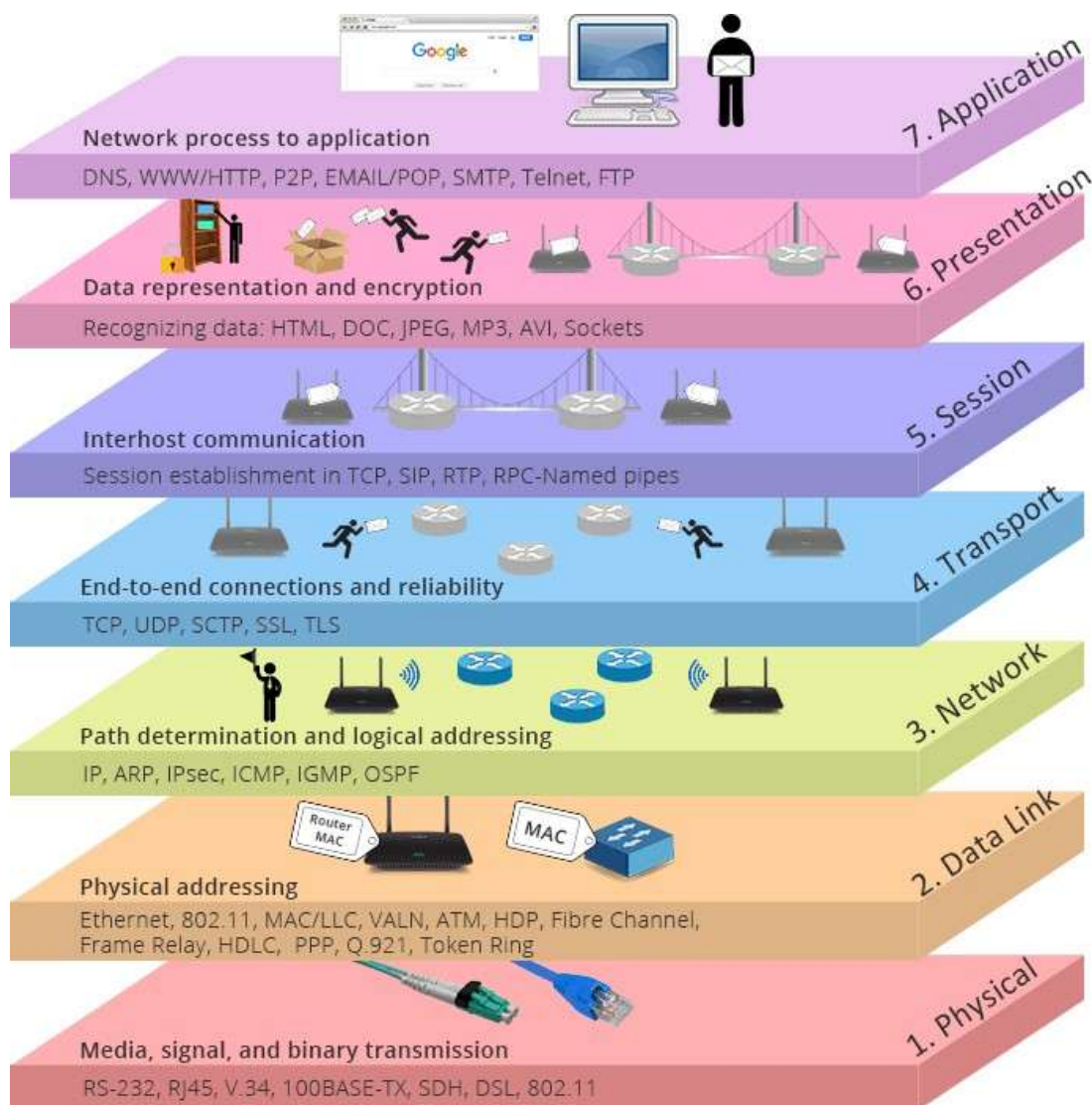


Figure II.4 Les différentes couches du modèle OSI

2.8.2 Le modèle TCP/IP :

Le modèle TCP/IP, également appelé “suite de protocoles Internet”, est un modèle de référence en couches. Contrairement au modèle OSI, qui comporte sept couches, le modèle TCP/IP se compose de quatre couches principales. Ces couches sont conçues pour gérer la transmission des données sur un réseau. Les deux protocoles dominants dans ce modèle sont TCP (Transmission Control Protocol) et IP (Internet Protocol). [5]

1. Couche d'Application :

La couche d'application du modèle TCP/IP permet aux applications d'accéder aux services des autres couches. Elle détermine également les protocoles que les applications utilisent pour échanger des données. Parmi les protocoles de cette couche, on trouve les plus connus tels que HTTP, FTP, SMTP, Telnet, DNS, SNMP et le protocole RIP (Routing Information Protocol).

2. Couche de Transport :

La couche de transport, également appelée couche de transport hôte-à-hôte, fournit à la couche d'application des services de communication de session et de datagramme. Les protocoles de base de cette couche sont TCP (Transmission Control Protocol) et UDP (User Datagram Protocol).

TCP offre un service de communication fiable entre dispositifs, axé sur la connexion. Il gère la séquence, la reconnaissance de réception des paquets et la récupération des paquets perdus en transmission.

UDP, quant à lui, fournit un service de communication sans connexion et peu fiable entre dispositifs. Il est généralement utilisé lorsque la quantité de données à transférer est faible (par exemple, lorsque les données tiennent dans un seul paquet).

3. Couche Internet :

La couche Internet est responsable des fonctions d'adressage, de conditionnement et de routage des hôtes.

Les principaux protocoles de cette couche sont :

IP (Internet Protocol) : Protocole routable qui gère l'adressage IP, le routage, la fragmentation et le réassemblage des paquets.

ARP (Address Resolution Protocol) : Responsable de la découverte des adresses de la couche d'accès au réseau, telles que les adresses matérielles associées à une adresse IP donnée.

ICMP (Internet Control Message Protocol) : Fournit des fonctions de diagnostic et signale les erreurs liées à la livraison des paquets IP.

IGMP (Internet Group Management Protocol) : Gère les groupes de multidiffusion (multicast) IP.

Dans cette couche, l'IP ajoute un en-tête aux paquets, ce qui est connu sous le nom d'adresse IP. Il existe désormais une adresse IPv4 (32 bits) et une adresse IPv6 (128 bits).

4. La couche d'accès réseau :

(Ou couche de liaison) est responsable de placer et de recevoir les paquets TCP/IP dans et en dehors du support réseau. TCP/IP est conçu pour être indépendant de la méthode d'accès au réseau, du format de trame et du support. En d'autres termes, il est indépendant de toute technologie de réseau spécifique. De cette façon, TCP/IP peut être utilisé pour connecter différents types de réseaux, tels que l'Ethernet, Token Ring, et Asynchronous Transfer Mode (ATM)

3. Définition d'automatisation des réseaux :

L'automatisation du réseau est le processus d'automatisation de la configuration, de la gestion, des tests, du déploiement et du fonctionnement des périphériques physiques et virtuels au sein d'un réseau. Grâce à l'automatisation, les tâches et les fonctions quotidiennes du réseau sont exécutés automatiquement, et les processus répétitifs sont contrôlés et gérés sans intervention humaine. En conséquence, la disponibilité du service réseau s'améliore. [6]

4. Pourquoi automatiser un réseau ?

- ✓ L'un des plus grands problèmes pour les gestionnaires de réseau est la croissance des coûts informatiques pour l'exploitation du réseau. La croissance des données et des périphériques commence à dépasser les capacités informatiques, rendant les approches manuelles presque impossibles.
- ✓ Jusqu'à 95 % des changements apportés au réseau sont aujourd'hui effectués manuellement.
- ✓ Les changements manuels entraînent des erreurs de configuration et des incohérences dans le réseau.
- ✓ Le développement à grande échelle des changements apportés au réseau peut être problématique.
- ✓ Les temps d'arrêt du réseau et les temps de dépannage non distants sont préjudiciables. [6]

5. Avantages d'automatisation d'un réseau :

Les avantages d'un réseau entièrement automatisé sont nombreux. Ce qui pourrait sembler être un long processus déployé par étapes finira par donner à l'organisation plusieurs avantages. Même les employés qui ne sont généralement pas informés de la situation dans le service informatique en seront informés. [7]

1. Maîtrise des risques :

L'automatisation permet de minimiser les erreurs humaines lors de la configuration et de la gestion des réseaux. En utilisant des scripts automatisés, les équipes peuvent garantir une cohérence et une précision accrues, réduisant ainsi les risques d'incidents réseaux.

Les processus automatisés suivent des règles prédéfinies, ce qui réduit les chances d'introduire des vulnérabilités ou des configurations incorrectes.

2. Effectuer des changements plus rapides :

L'automatisation permet de déployer rapidement des modifications de configuration, des mises à jour de sécurité et des correctifs.

Les tâches manuelles prennent du temps, mais l'automatisation permet d'appliquer des changements de manière instantanée et cohérente.

3. Réseau plus fiable :

L'automatisation garantit une configuration uniforme et cohérente sur l'ensemble du réseau. Cela réduit les problèmes de compatibilité.

Les mécanismes de surveillance automatisés détectent rapidement les pannes et les problèmes de performance, permettant ainsi une réaction plus rapide.

4. Optimiser les performances :

L'automatisation permet d'ajuster dynamiquement les ressources réseau en fonction des besoins. Par exemple, elle peut augmenter la bande passante pour les applications critiques ou réduire la latence pour les services sensibles.

Les outils automatisés surveillent les performances et identifient les goulots d'étranglement, permettant ainsi d'optimiser les performances globales du réseau.

5. Simplifier la gestion du réseau :

L'automatisation réduit la charge de travail manuelle pour les équipes d'exploitation réseau. Les tâches répétitives telles que la configuration initiale, la gestion des utilisateurs et la surveillance sont automatisées.

Les interfaces de gestion automatisées simplifient la visualisation et la gestion de l'ensemble du réseau.

6. Etude sur les outils d'automatisation :

De nombreux outils d'automatisation sont disponibles pour simplifier la gestion des configurations. Ces outils visent à alléger la complexité et à minimiser le temps nécessaire pour configurer et entretenir les réseaux, en particulier les vastes réseaux comportant des centaines d'équipements. Parmi les outils spécialisés dans la gestion de configuration et l'automatisation, trois se distinguent dans le domaine NetDevOps pour leur notoriété et leur adoption par les sociétés : Ansible, Puppet et Chef. Dans cette partie, nous examinerons ces outils et discuterons des avantages et inconvénients d'Ansible par rapport à Chef et Puppet, afin de choisir les bons outils de gestion de configuration pour nos besoins. [8]

6.1 Chef :

Chef, ou Progress Chef, est un outil de gestion de configuration qui gère efficacement votre infrastructure. Chef vous permet d'utiliser Ruby pour créer des configurations système, appelées recettes, qui décrivent l'état optimal de votre infrastructure, tel que le serveur qui devrait exécuter quel service, quels logiciels devraient être installés, quels fichiers devraient être écrits, et ainsi de suite. Avec ces configurations, Chef garantira que votre infrastructure est correctement configurée et réparera automatiquement toutes les ressources qui ne fonctionnent pas dans un état optimal.



6.1.1 Avantages :

- Entièrement programmable, de sorte que l'étendue de la manipulation et de la personnalisation est très élevée
- Le chef exécute également les commandes dans un ordre séquentiel, ce qui est très facile à comprendre.
- La communauté des chefs est très active et dispose d'une documentation et d'un support solides.
- L'une des solutions les plus flexibles pour la gestion des systèmes d'exploitation et des middlewares.
- Le chef est bien mûr et stable pour un déploiement à grande échelle.
- Une version SaaS de Chef est disponible, ce qui est très utile pour l'analyse et la création de rapports. [9]

6.1.2 Inconvénients :

- Les débutants ont demandé un apprentissage énorme et c'est très difficile pour eux.
- La configuration du chef et les configurations initiales sont complexes.
- La configuration basée sur l'extraction attendra la prochaine interrogation planifiée pour obtenir la configuration du serveur. [9]

6.2 Puppet :

Puppet est un autre outil populaire de gestion de configuration de serveurs qui vous permet de configurer et de surveiller plusieurs serveurs en même temps. Il utilise son propre langage déclaratif pour décrire les configurations système, et il nécessite seulement que l'utilisateur ait une quantité limitée de connaissances en programmation pour l'utiliser.



6.2.1 Avantages :

- L'installation et la configuration initiales sont très faciles
- La console Web UI nous aidera à prendre en charge facilement de nombreuses tâches de configuration, de reporting et de gestion des nœuds en temps réel.
- Puppet est très robuste et a la capacité native de travailler avec des constructions au niveau du shell.

- Système très stable et mature pour les DevOps pour gérer une infrastructure à grande échelle
- La communauté des marionnettes est également très active et dispose d'une documentation et d'un soutien solides. [9]

6.2.2 Inconvénients :

- Parfois, il est difficile pour les débutants d'apprendre Puppet DSL ou Ruby,
- Nous avons besoin de CLI pour effectuer des tâches avancées.
- Le code Ruby DSL peut devenir grand lorsque nous augmentons la taille et cela devient compliqué.
- Comme toujours, le système basé sur le pull suit un travail planifié pour des tâches qui nous feront attendre la configuration.
- Puppet DSL est un peu différent de Ruby, de sorte que le Ruby pur ne fonctionnera pas parfois. [9]

6.3 Ansible :

Ansible est un produit relativement récent, mais il a gagné en popularité depuis son acquisition par RedHat en 2015. Il vous permet d'automatiser la fourniture de logiciels, la gestion de configuration et le déploiement d'applications. Ansible utilise YAML pour créer des configurations système, décrivant l'état optimal de votre infrastructure.



6.3.1 Avantages :

- L'installation d'Ansible est très simple et facile à configurer.
- L'écriture d'un script dans le fichier YAML est très simple et puissante.
- L'exécution des commandes à distance facilite l'accessibilité du système cible.
- Les commandes Ansible sont exécutées dans un ordre séquentiel afin que la compréhension de l'exécution du script soit facile.
- Ansible peut être exécuté N nombre de serveurs cibles selon le fichier d'inventaire.
- Partage les faits entre plusieurs serveurs, afin qu'ils puissent s'interroger mutuellement.
- Le déploiement sans agent permet d'établir des connexions plus rapides par rapport à un modèle basé sur un agent.
- Par rapport aux protocoles de connexion d'autres outils, les connexions SSH et winRM sont sécurisées par rapport aux autres modèles. [9]

6.3.2 Inconvénient :

- Les communications SSH sont lentes, ce qui peut entraîner plus de temps d'arrêt.
- Par rapport à d'autres outils, des fonctionnalités limitées comme la surveillance, la disponibilité des API.
- La syntaxe des Playbooks et des Templates est différente, ce qui peut être un peu difficile pour les débutants. [9]

6.4 Comparaison et synthèse :

Dans ce tableau, nous comparerons en détail les trois plateformes pour trouver la meilleure option. La comparaison sera basée sur les critères suivants : [10]

Catégorie	Chef	Puppet	Ansible
Version initiale	2009	2005	2012
Langue de configuration	Rubis DSL	Ruby, Puppet DSL, Embedded Ruby (ERB), DSL	Python, YAML
Utilisabilité	Modéré	Modéré	Facile
Architecture	Maître, Agent	Maître, Agent	Nœud de contrôle, nœud géré
Disponibilité	Serveur de sauvegarde	Maître alternatif	Instance secondaire
Ensemble de fonctionnalités	Haut	Haut	Bas
Processus de configuration	Complexe ; principalement en raison de la configuration de Chef Workstation	Complexe ; en raison de la signature du certificat maître-agent	Relativement facile
Gestion de la configuration	Tirer	Tirer	Pousser et tirer
Évolutivité	Haut	Haut	Très élevé
Interopérabilité	- Serveur : prise en charge Linux / Unix uniquement. - Client : Windows et Linux	- Puppet Master (serveur) : prise en charge Linux / Unix uniquement. - Agent : Windows et Linux	- Nœud de contrôle : Linux, Ubuntu, Windows (avec WSL)

			- Nœud géré : n'importe quel appareil.
Soutien aux entreprises	Grand	Grand	Petit
Déploiement d'applications	Non	Complexe	Oui

Tableau II-1 comparaison des outils

7. La solution proposée :

Dans le cadre de l'automatisation des réseaux, il est essentiel de choisir un outil qui non seulement répond aux exigences techniques mais qui s'aligne également avec la vision stratégique du projet. Ansible, avec sa simplicité d'utilisation et sa capacité à déployer des configurations de manière idempotente, se distingue comme une solution de choix. Son architecture sans agent et son langage de configuration déclaratif permet une intégration et une gestion aisées des infrastructures réseau complexes. En optant pour Ansible, nous nous orientons vers une plateforme qui favorise la reproductibilité des déploiements et l'évolutivité des systèmes, tout en réduisant les risques d'erreurs humaines. Ce choix stratégique est le reflet d'une approche moderne de l'automatisation, où la fiabilité, l'efficacité et la sécurité sont au cœur des opérations réseau.

7.1 Pourquoi Ansible :

Lors de notre stage pratique au sein de l'entreprise Candia, nous avons minutieusement évalué plusieurs outils d'automatisation de la configuration, dont Ansible, Puppet et Chef. Notre choix s'est finalement porté sur Ansible.

Ansible se distingue comme un outil d'automatisation de choix pour de nombreuses entreprises grâce à son architecture sans agent, sa facilité de configuration et sa gestion efficace des systèmes complexes. Contrairement à Puppet et Chef qui nécessitent l'installation d'agents sur les systèmes cible, Ansible réduit les frais généraux et simplifie le processus d'automatisation. De plus, Ansible utilise YAML pour ses configurations, ce qui le rend plus accessible et facile à comprendre pour les équipes, comparativement au langage déclaratif propre à Puppet et aux recettes basées sur Ruby de Chef. En termes de compatibilité réseau, Ansible excelle dans l'automatisation des tâches NetOps complexes, étendant ses capacités des datacenters aux sites d'edge computing, et gérant une variété d'équipements réseau avec une seule plateforme unifiée.

8. Conclusion :

La définition de l'automatisation des réseaux, ainsi que l'examen approfondi des outils disponibles, mettent en lumière la diversité et la richesse des solutions existantes. La comparaison minutieuse entre ces outils et la mise en avant de la solution Ansible, en particulier dans le contexte de l'entreprise Candia, soulignent l'importance de choisir une solution adaptée aux besoins spécifiques de l'entreprise. Ce chapitre pose les bases nécessaires pour comprendre les enjeux et les bénéfices de l'automatisation des réseaux, ouvrant la voie à une exploration plus détaillée d'Ansible qui est l'objet du chapitre suivant.

III. Chapitre III: Généralité sur ANSIBLE

1. Introduction :

Dans ce chapitre, nous présentons de manière détaillée l'outil choisi pour la mise en œuvre de notre solution à savoir Ansible, qui est devenu incontournable dans le domaine de l'automatisation des réseaux. Nous débutons par un voyage à travers l'histoire d'Ansible, en découvrant comment et pourquoi cet outil a émergé comme une solution privilégiée pour les administrateurs systèmes et réseaux. Ensuite, nous détaillerons les composants qui constituent l'architecture d'Ansible, en mettant en lumière la simplicité et l'efficacité qui sous-tendent son design et son fonctionnement.

2. Définition :

Ansible est un moteur d'automatisation informatique écrit en Python open source qui automatise le provisionnement, la gestion de la configuration, le déploiement des applications, l'orchestration et de nombreux autres processus informatiques. Son utilisation est gratuite et le projet bénéficie de l'expérience et de l'intelligence de ses milliers de contributeurs.

Ansible s'exécute sur n'importe quelle infrastructure. Grâce à des scripts lisibles par l'homme appelés playbooks, vous pouvez gérer la configuration système, déployer des logiciels, effectuer des mises à jour continues et éliminer les redondances. Ansible fonctionne sans agent, réduisant ainsi les frais de maintenance, et utilise une syntaxe YAML simple pour décrire l'état souhaité du système. Il est également évolutif et flexible, prenant en charge divers systèmes d'exploitation, plateformes cloud et périphériques réseau. Enfin, Ansible garantit l'idempotence et la prévisibilité : même si le playbook s'exécute plusieurs fois, le système reste dans l'état défini. [9]

3. Historique :

Ansible, l'outil d'automatisation système incontournable aujourd'hui, est né en 2012 de l'esprit de Michael DeHaan, également connu pour son application de provisionnement de serveurs Cobbler. A cette époque, le mouvement DevOps prenait son essor et des outils comme Puppet et Chef régnaient sur le marché. DeHaan, fort de son expérience chez Puppet, était convaincu qu'il manquait une solution d'automatisation plus simple et accessible à tous.

Lassé de la complexité des outils existants qui freinaient l'adoption des pratiques DevOps, DeHaan a conçu Ansible en misant sur la simplicité et l'accessibilité. Basé sur le langage YAML et doté d'une architecture "sans agents", Ansible s'est rapidement démarqué par sa facilité d'utilisation et son efficacité.

En 2015, Red Hat, reconnaissant le potentiel d'Ansible, a racheté l'outil. Cette intégration a permis à Ansible de bénéficier des ressources et de la communauté d'une entreprise majeure dans le monde des logiciels libres, propulsant ainsi son adoption à un niveau encore plus élevé.

Aujourd'hui, Ansible s'impose comme l'un des outils de gestion de configuration les plus populaires, avec le plus grand nombre de forks et de stars sur GitHub. Son succès fulgurant témoigne de sa pertinence dans le paysage DevOps actuel, répondant ainsi à la vision initiale de DeHaan d'une automatisation accessible à tous. [11]

3.1 Red Hat :

Red Hat est une entreprise de logiciel spécialisée dans les solutions open source, le site web redhat.com propose une variété de produits et services basés principalement sur le système d'exploitation Linux, notamment RHEL ou Red Hat Entreprise Linux.

Red Hat offre également des solutions en matière de Cloud Computing, de virtualisation, de gestion de systèmes. [12]

4. Cas d'usage d'Ansible :

Ansible, un outil qui a la puissance d'automatiser plusieurs processus informatique, tels que : [13]

❖ Gestion de la configuration :

Ansible transforme la complexité de la gestion et de l'orchestration des configurations en un processus simplifié et automatisé. Sa conception repose sur la simplicité et la cohérence, permettant aux utilisateurs, même ceux avec une expérience informatique limitée, de déployer rapidement des configurations à travers des infrastructures. Les configurations sont exprimées dans un langage clair, rendant les playbooks Ansible faciles à lire et à écrire. Ils décrivent l'état souhaité des systèmes, que ce soit pour l'installation de logiciels, la mise à jour de paramètres ou la gestion des utilisateurs. Avec Ansible, les tâches répétitives deviennent des modèles réutilisables, ce qui réduit les erreurs et augmente l'efficacité. Par exemple, pour mettre à jour un logiciel sur plusieurs serveurs, il suffit de lister les adresses IP dans un inventaire, de créer un playbook définissant les tâches à exécuter, et de lancer l'exécution du playbook depuis un poste de contrôle. Ansible s'occupe du reste, en appliquant les changements de manière idempotente, garantissant que le résultat final est toujours conforme à la configuration définie, peu importe l'état initial des systèmes ciblés.

❖ L'orchestration :

L'orchestration avec Ansible est une méthode puissante pour gérer et coordonner des systèmes informatiques complexes. À l'image d'un chef d'orchestre qui dirige chaque instrument pour créer une symphonie harmonieuse, Ansible orchestre l'ensemble des composants d'une infrastructure IT pour assurer une exécution fluide et synchronisée des tâches. En déployant des applications, par exemple, il ne suffit pas de lancer les services front-end et back-end ; il faut également configurer les bases de données, les réseaux et le stockage, tout en veillant à ce que chaque opération soit effectuée dans l'ordre approprié. Ansible simplifie cette complexité grâce

à des playbooks, qui sont des scripts YAML décrivant les tâches à effectuer. Ces playbooks, grâce à leur nature déclarative, permettent de définir l'état souhaité de l'infrastructure, rendant les processus reproductibles et cohérents, peu importe l'environnement. Ainsi, l'orchestration d'Ansible assure non seulement l'automatisation des flux de travail mais garantit également la portabilité des systèmes, éléments clés dans le monde agile et en constante évolution de la technologie informatique.

❖ **Le déploiement d'application :**

Ansible est un outil puissant pour le déploiement d'applications, permettant aux équipes de définir et de gérer l'infrastructure et les déploiements de manière cohérente et reproductible. En utilisant Ansible et Ansible Tower, il est possible de gérer le cycle de vie complet d'une application, simplifiant le passage du développement à la production. Avec Ansible, il n'est pas nécessaire d'écrire du code spécifique pour automatiser les systèmes. Au lieu de cela, les tâches sont définies dans un playbook, qui est une sorte de recette qu'Ansible suit pour mettre les systèmes dans l'état désiré. Cela élimine le besoin de configuration manuelle sur chaque machine, car Ansible utilise SSH pour communiquer avec les machines distantes et exécuter les tâches spécifiées. Ainsi, Ansible rend le processus de déploiement plus efficace et moins sujet aux erreurs humaines, ce qui est essentiel pour maintenir la stabilité et la fiabilité des environnements de production.

❖ **Le provisionnement :**

Le provisionnement avec Ansible est un processus clé pour la gestion et l'automatisation des infrastructures informatiques. En définissant l'état désiré de vos systèmes via des playbooks Ansible, vous pouvez assurer une configuration cohérente et répétable de vos environnements. Que ce soit pour déployer des serveurs dans le cloud, configurer des machines virtuelles, ou initialiser des dispositifs réseau, Ansible simplifie ces tâches en les rendant automatisables et sans intervention manuelle. Ainsi, Ansible contribue à réduire les erreurs humaines et accélère le déploiement de vos applications.

5. Architecture et composants :

5.1 Architecture :

Ansible dispose une architecture simple et compréhensible :

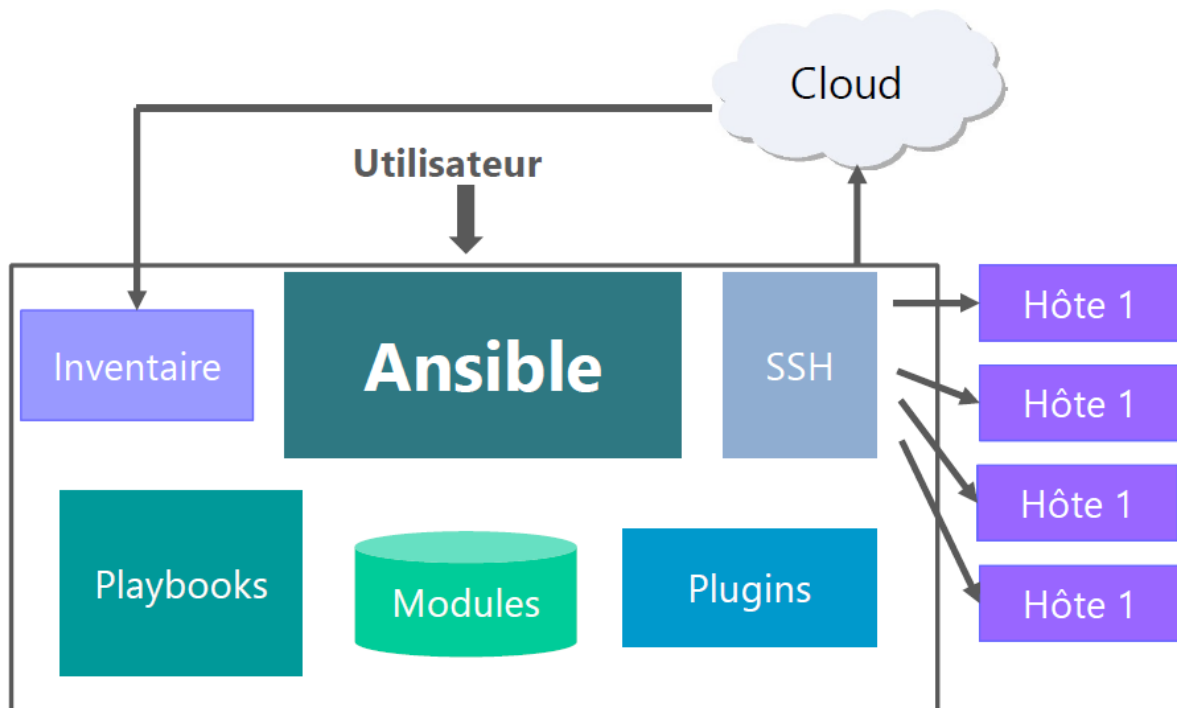


Figure III.1 Architecture structurelle d'ANSIBLE

5.2 Composants :

Dans cette section, on va donner un aperçu très rapide du fonctionnement d'Ansible afin de voir comment les pièces s'emboîtent.

Le moteur d'automatisation Ansible se compose de différents composants, comme décrit ci-dessous : [14]

5.2.1 Inventaire :

Ansible organise les systèmes qu'il contrôle en utilisant un fichier d'inventaire (sous formats INI, YAML, etc.), permettant de regrouper les machines selon vos préférences. L'ajout de nouvelles machines est simplifié car il n'est pas nécessaire de gérer un serveur de signature SSL, évitant ainsi les complications liées à des problèmes de NTP ou DNS. Ansible est également capable de se connecter à d'autres sources de données au sein de l'infrastructure. Un fichier d'inventaire typique pourrait ressembler à ceci :

```
[webservers]
www1.example.com
www2.example.com
[dbservers]
db0.example.com
db1.example.com
```

Après avoir listé les hôtes dans l'inventaire, il est possible de leur assigner des variables soit via des fichiers textes séparés situés dans les répertoires 'group_vars/' ou 'host_vars/', soit directement dans le fichier d'inventaire. Alternativement, un inventaire dynamique peut être utilisé pour tirer ces informations de sources de données.

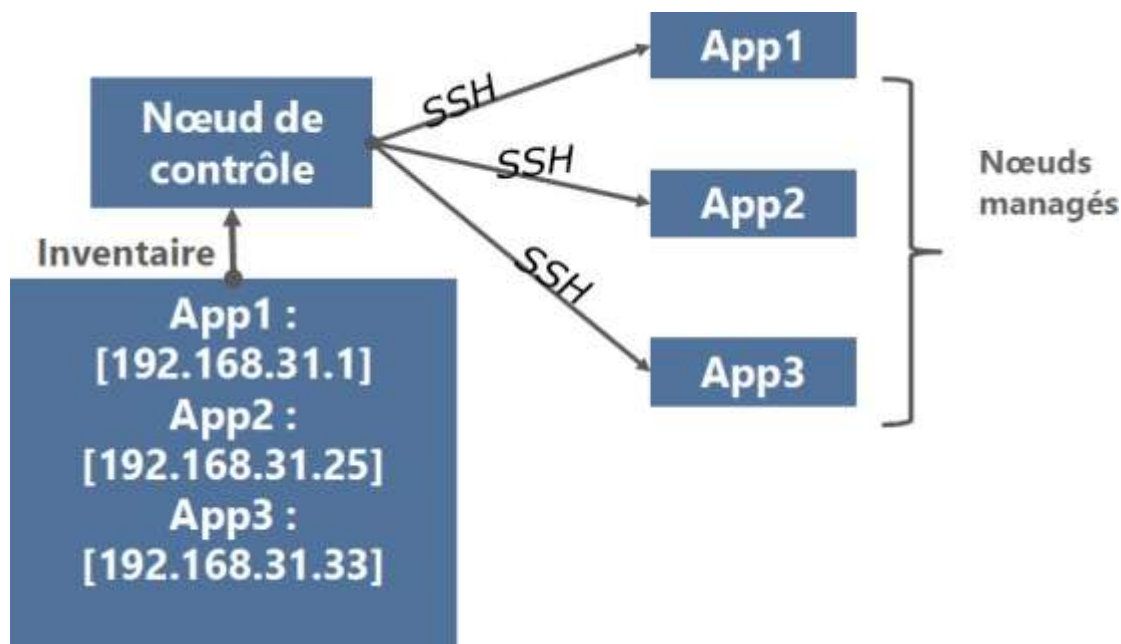


Figure III.2 Architecture d'un inventaire [14]

5.2.1 PlayBooks :

Les Playbooks Ansible permettent une orchestration précise de divers segments de l'architecture du système, offrant une maîtrise pointue du nombre d'unités informatiques gérées simultanément. C'est dans ce contexte qu'Ansible révèle tout son potentiel.

La philosophie d'Ansible en termes d'orchestration repose sur une élégance dans la simplicité. Nous sommes convaincus que le code d'automatisation écrit doit rester clair et compréhensible sur le long terme, sans nécessiter une mémoire extensive de la syntaxe ou des fonctionnalités avancées.

Exemple d'un Playbook :

- hosts: webservers

serial: 5 # update 5 machines at a time

roles:

- common

- webapp

- hosts: content_servers

roles:

- common

- content

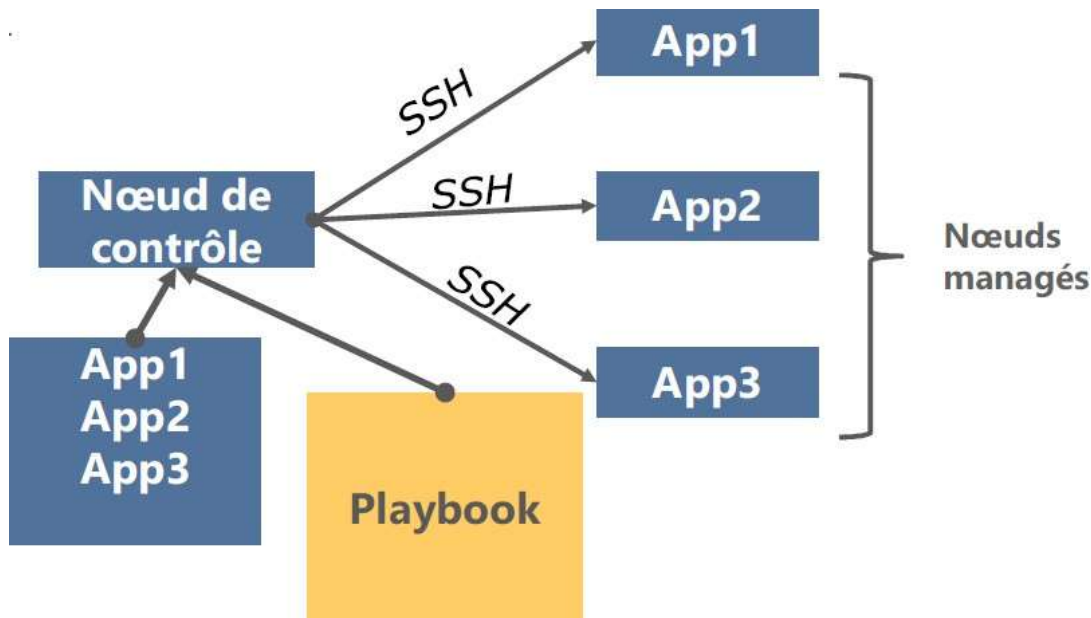


Figure III.3 Architecture d'un Playbook [14]

5.2.2 Modules :

Ansible opère en établissant une connexion avec les nœuds et en y déployant des scripts nommés "modules Ansible". Ces modules, qui acceptent généralement des paramètres définissant l'état cible du système, sont exécutés par Ansible (habituellement via SSH) et sont retirés après leur exécution. La collection de modules peut être hébergée sur n'importe quel appareil, sans nécessiter de serveurs, de démons ou de bases de données.

Il est possible de concevoir des modules personnalisés, cependant, il convient de réfléchir à la nécessité de cette démarche. Typiquement, le processus implique l'utilisation d'un terminal, d'un éditeur de texte et, souvent, d'un système de gestion de versions pour le suivi des modifications du contenu. Les modules peuvent être écrits dans tout langage capable de produire du JSON (tels que Ruby, Python, bash, etc.).

Pour les utilitaires de module, Ansible centralise les fonctions communes à plusieurs modules afin de réduire la redondance et faciliter la maintenance. Ainsi, le code pour l'analyse des URL est situé dans `lib/Ansible/module_utils/url.py`. Il est également permis de créer ses propres utilitaires de module, qui doivent être rédigés en Python ou PowerShell.

5.2.3 Plugings :

Les extensions de fonctionnalités d'Ansible sont principalement réalisées par l'intermédiaire de plugings. Contrairement aux modules, qui opèrent sur le système cible via des processus distincts, souvent sur une machine distante, les plugings fonctionnent directement sur le nœud de contrôle central. Ils enrichissent Ansible avec des capacités supplémentaires telles que la transformation de données, la capture de sorties, la gestion de l'inventaire, entre autres. Ansible est fourni avec une gamme de plugings utiles, mais il est également possible de développer des plugings personnalisés. Par exemple, un plugin d'inventaire personnalisé peut être créé pour se connecter à toute source de données retournant du JSON. Pour l'écriture de plugings, la maîtrise du langage Python est requise.

5.3 Comment fonctionne ANSIBLE :

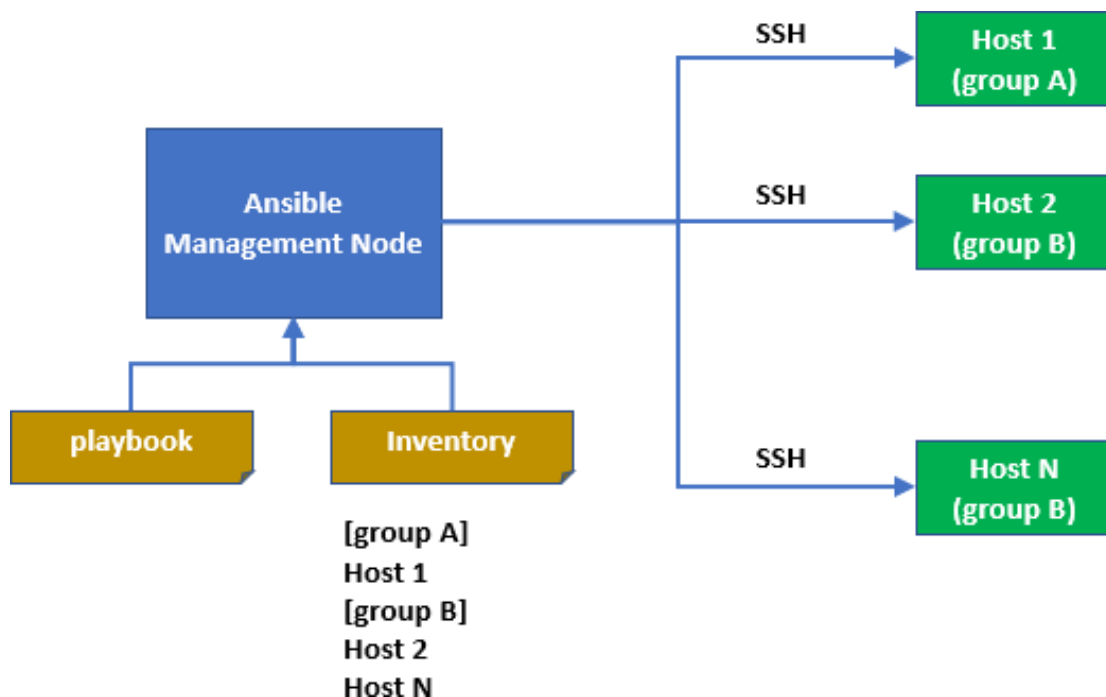


Figure III.4 Fonctionnement d'Ansible

Le fonctionnement d'Ansible se base principalement sur deux types de machines :

- **Le serveur Ansible** : couramment appelé la machine de contrôle ou Management Node, c'est la machine Ansible.

- **Les Hôtes** : appelés aussi nœuds ce sont les machines sur lesquelles Ansible effectuera des tâches.

Ansible fonctionne en se connectant aux hôtes ou aux réseaux en SSH et en y poussant de petits programmes, appelés modules. Ces modules sont définis dans un fichier nommé le Playbook. Le Nœud de contrôle se base sur un fichier d'inventaire qui fournit la liste des hôtes sur lesquels les modules Ansible doivent être exécutés.

Cette configuration permet à Ansible de gérer efficacement de multiples serveurs, en automatisant des tâches répétitives et en assurant la cohérence des configurations à travers différents environnements. [15]

5.4 ANSIBLE sans agents :

Ansible est une technologie sans agent, ce qui signifie qu'elle n'installe aucun logiciel sur les nœuds qu'elle gère. Elle lit les informations concernant les machines qu'on souhaite gérer à partir de l'inventaire. Ansible dispose d'un fichier d'inventaire par défaut, mais on peut créer un autre et y indiquer les serveurs à gérer.

Pour se connecter aux serveurs et exécuter des tâches, Ansible utilise le protocole SSH. Par défaut, Ansible utilise des clés SSH avec Ss-agent et se connecte aux machines distantes à l'aide de nom d'utilisateur actif. Pas besoin d'une connexion root, on peut se connecter avec le compte de n'importe quel utilisateur, puis utiliser la commande « su » ou « sudo » pour passer à un autre compte utilisateur.

Une fois connectée, la technologie Ansible transfère les modules requis par une commande ou playbook Ansible vers les machines distantes pour leur exécution. [12]

6. Ansible pour des commandes ad hoc :

Ansible, ce n'est pas que des playbooks, On peut aussi l'utiliser pour lancer des commandes en direct sur un ou plusieurs serveurs, un peu comme si on se connectait en SSH. C'est pratique pour des tâches ponctuelles, comme installer un logiciel en urgence ou vérifier un fichier spécifique. Il suffit d'utiliser la commande « Ansible » suivie de la cible (un serveur ou un groupe de serveurs), du module à exécuter et des options. [12]

7. Le modèle Jinja2 :

Ansible s'appuie sur les modèles Jinja2 pour permettre des expressions dynamiques et l'accès à des variables et faits divers. Ces modèles sont utilisables via le module 'template'. Ils permettent, par exemple, de générer un fichier de configuration adapté, puis de le déployer dans différents environnements en y intégrant les données spécifiques nécessaires (telle que l'adresse IP, le nom de l'hôte, la version). La création de modèles peut aussi se faire directement dans les playbooks, pour personnaliser les noms des tâches, entre autres. Jinja2 offre une variété de filtres et tests standards, que Ansible complète avec des filtres exclusifs pour la manipulation

et la transformation des données, ainsi que des tests pour l'évaluation des expressions de modèles et des plugins de recherche pour l'acquisition de données depuis des sources externes, comme des fichiers, des API ou des bases de données.

L'ensemble du processus de modélisation est réalisé sur le nœud de contrôle d'Ansible avant l'envoi et l'exécution des tâches sur les machines cibles. Cette méthode réduit les besoins en termes de packages sur les machines cibles, car Jinja2 est uniquement nécessaire sur le nœud de contrôle. Elle limite aussi le volume de données transmises aux machines cibles. Ansible traite les modèles sur le nœud de contrôle et envoie seulement les informations essentielles pour l'exécution des tâches, évitant ainsi le transfert et le traitement de données superflues sur les machines cibles. [14]

8. Langage de programmation utilisé :

Ansible utilise des modèles YAML lisibles par l'homme. Ainsi, les utilisateurs peuvent automatiser des tâches répétitives sans avoir à apprendre un langage de programmation avancé.

8.1 Présentation du modèle YAML :

YAML est un langage de sérialisation des données lisible par l'utilisateur qui est souvent utilisé pour coder des fichiers de configuration. Pour certains, YAML est l'acronyme de Yat Annotera Markup Language, pour d'autres, c'est l'acronyme récursif de YAML Ain't Markup Language (YAML n'est pas un langage de balisage), ce qui souligne que le langage YAML s'utilise pour représenter des données plutôt que des documents.

YAML est un langage de programmation fréquemment utilisé, car il est conçu pour être parfaitement lisible et compréhensible. Il peut également être utilisé en association avec d'autres langages de programmation. En raison de sa flexibilité et de son accessibilité, YAML est utilisé par ANSIBLE pour créer des processus d'automatisation, sous la forme de playbooks Ansible.

Le langage YAML est principalement utilisé pour créer des fichiers de configuration. Il est recommandé de coder les fichiers de configuration en YAML plutôt qu'en JSON, car même si ces langages peuvent être utilisés de manière interchangeable dans la plupart des cas, YAML reste plus facile à lire et à utiliser. [16]



8.2 Syntaxe YAML :

- ❖ Les fichiers YAML utilisent une extension `.yaml` ou `.yml`, et suivent des règles de syntaxe spécifiques.
- ❖ YAML comporte des fonctions issues de Perl, C, XML, HTML et d'autres langages de programmation. Comme il s'agit également d'un surensemble du langage JSON, les fichiers JSON sont des fichiers YAML valides.
- ❖ Les commentaires sont introduits par le symbole de la livre (£) ou le dièse (#).
- ❖ YAML ne prend pas en charge les commentaires sur plusieurs lignes, Chaque ligne doit être suivie du symbole £.
- ❖ En YAML, ils marquent le début d'un document par trois tirets (---), et signalent la fin du document par trois points (...).
- ❖ YAML comporte également des scalaires, qui sont des données arbitraires (codées en Unicode) pouvant être utilisées comme valeurs, telles que des chaînes, des entiers, des dates, des nombres ou des booléens.
- ❖ un fichier YAML est structuré sous la forme d'un mappage ou d'une liste qui respecte une hiérarchie basée sur l'indentation et la définition des valeurs clés.
- ❖ Les mappages permettent d'associer des paires clé-valeur. Chaque clé doit être unique, et l'ordre importe peu. Cela ressemble à un dictionnaire Python ou à une affectation de variable dans un script bash. [16]

9. Conclusion :

Dans ce chapitre, nous avons exploré les concepts fondamentaux d'Ansible et les étapes détaillées de son installation sur Ubuntu. Cette plateforme puissante simplifie la gestion des configurations et automatise les processus, ce qui est essentiel dans le déploiement d'infrastructures informatiques modernes. Avec cette base théorique solide, nous sommes désormais prêts pour la création d'un environnement de travail virtuel et l'installation des logiciels nécessaires à la mise en place de notre solution, qui seront abordés dans le chapitre qui suit.

IV. Chapitre IV: Réalisation du projet

1. Introduction :

Dans ce chapitre, nous explorons l'environnement de travail virtuel utilisé pour la réalisation de notre solution. Nous aborderons les outils et logiciels mis en place pour simuler un réseau et automatiser les tâches de configuration réseau à l'aide d'Ansible. Cette section détaille les outils logiciels et les plateformes de virtualisation, notamment VMware Workstation et GNS3, qui ont été utilisés pour simuler les réseaux dans le cadre de ce projet. Noté que l'impossibilité d'accéder au réseau réel de l'entreprise a conduit à l'adoption de ces solutions virtuelles, permettant ainsi une expérimentation flexible et contrôlée.

2. Environnement de travail :

2.1 La virtualisation :

2.1.1 Définition :

La virtualisation est une technologie qui permet de diviser un ordinateur physique en plusieurs machines virtuelles indépendantes grâce à un logiciel nommé hyperviseur. Ce dernier gère la répartition des ressources matérielles entre les différentes machines virtuelles, appelées "invités", qui fonctionnent comme si elles étaient sur des systèmes distincts. L'ordinateur qui exécute l'hyperviseur est désigné comme « l'hôte ». Cette technologie optimise l'utilisation des ressources et améliore la flexibilité et l'efficacité des systèmes informatiques en permettant un meilleur contrôle et une allocation dynamique des capacités de calcul selon les besoins. [17]

2.1.2 Les éléments clé de la virtualisation :

a. Machines virtuelles (VM) :

Les machines virtuelles (VM) sont des environnements virtuels qui simulent un ordinateur physique sous la forme d'un logiciel. Elles comprennent normalement plusieurs fichiers contenant la configuration de la machine virtuelle, le stockage du disque dur virtuel, et quelques instantanés de la machine virtuelle qui maintiennent son état à un point de cohérence. [18]

b. Hyperviseurs :

Un hyperviseur est la couche logicielle qui coordonne les machines virtuelles. Il sert d'interface entre la machine virtuelle et le matériel physique sous-jacent, garantissant que chacun a accès aux ressources physiques dont il a besoin pour s'exécuter. Il garantit également que les VM n'interfèrent pas les unes avec les autres en empiétant sur leur espace mémoire ou cycle de calcul respectifs.

Il existe deux types d'hyperviseurs :

- **Les hyperviseurs « bare-metal » ou de type 1** : interagissent avec les ressources physiques sous-jacentes, remplaçant complètement le système d'exploitation traditionnel. Ils apparaissent le plus souvent dans des scénarios de serveur virtuel.
- **Les hyperviseurs de type 2** : s'exécutent comme une application sur un système d'exploitation existant. Plus couramment utilisés sur les points de terminaison pour exécuter des systèmes d'exploitation alternatifs, ils doivent utiliser le système d'exploitation hôte pour accéder aux ressources matérielles sous-jacentes et les coordonner. [18]

2.2 Avantages de la virtualisation :

- Centralisation des ressources physiques pour une standardisation du matériel.
- Optimisation des infrastructures coûteuses par la virtualisation pour un support élargi d'applications.
- Suppression de l'obligation de certification matérielle individuelle pour chaque application.
- Simplification de la migration des machines virtuelles pour maintenir les opérations sans interruption.
- Facilité de création ou de duplication de bancs d'essai pendant les tests de régression.
- Élimination du besoin d'équipements dédiés ou de serveurs de développement redondants.
- Possibilité d'optimisation des environnements virtualisés pour augmenter les capacités et la densité avec une formation adéquate. [18]

2.3 La virtualisation des serveurs :

La virtualisation offre la capacité de faire fonctionner divers serveurs virtuels, tels que des serveurs de messagerie, de gestion commerciale et comptable, ou de bases de données, sur une seule machine physique. Cette technologie facilite aussi l'attribution autonome des ressources essentielles comme la mémoire et l'espace disque, permettant ainsi à chaque serveur de fonctionner avec les applications dans des conditions idéales. [19]

Il existe de nombreux hyperviseurs de virtualisation :

- VirtualBox
- VMWare Workstation Pro/Player
- GNOME machine
- KVM

3. Présentation de VMWare Workstation Pro :

Dans le cadre de notre projet, nous allons implémenter VMware Workstation Pro comme solution de virtualisation. Cette plateforme sera utilisée pour créer et gérer des machines virtuelles, ce qui permettra une simulation efficace des environnements de réseau et de système d'exploitation. L'intégration de VMware Workstation Pro contribuera à la robustesse et à la flexibilité de notre infrastructure informatique.

VMware Workstation Pro™ permet aux professionnels de l'informatique de développer, tester, présenter et déployer des logiciels en exécutant simultanément plusieurs systèmes d'exploitation x86, tels que Windows et Linux, sur un même ordinateur. Il est possible de reproduire des environnements de serveurs, de bureaux et de tablettes dans une machine virtuelle, et d'allouer plusieurs cœurs de processeur, des giga-octets de mémoire principale et de mémoire graphique à chaque machine virtuelle, que celle-ci se trouve sur un PC personnel ou dans un cloud d'entreprise privé. [20]



3.1 Installation :

Pour installer le logiciel VMWare Workstation Pro sur une machine Windows on doit d'abord consulter le site de téléchargement www.vmware.com. La figure ci-dessous montre la possibilité d'utiliser ce logiciel soit avec le système Windows soit avec le système Linux.

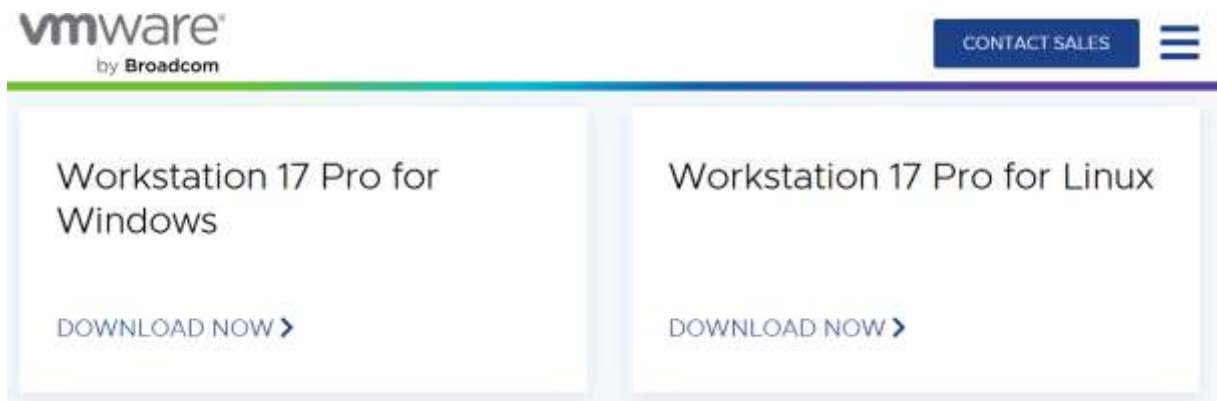


Figure IV.1 Site d'installation de VMWare Workstation Pro

En suivant les étapes d'installation, on arrive à l'interface finale du logiciel qui donnée par la figure ci-dessous.

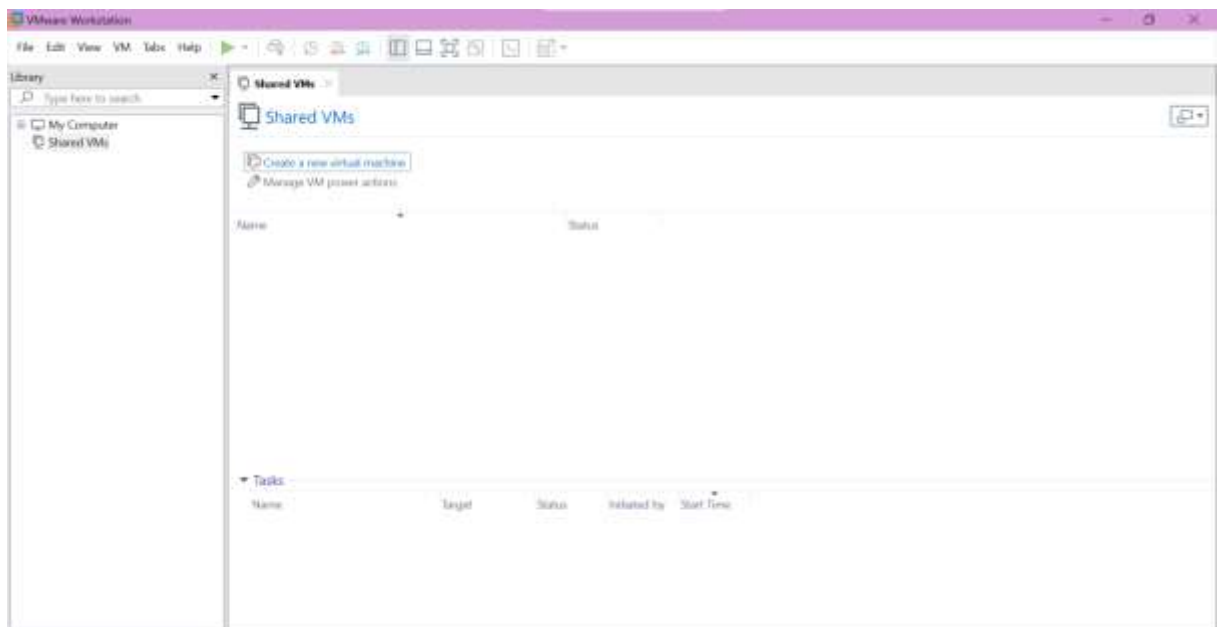


Figure IV.2 L'interface graphique initiale de VMWare Workstation Pro

3.2 Installation des machines virtuelles :

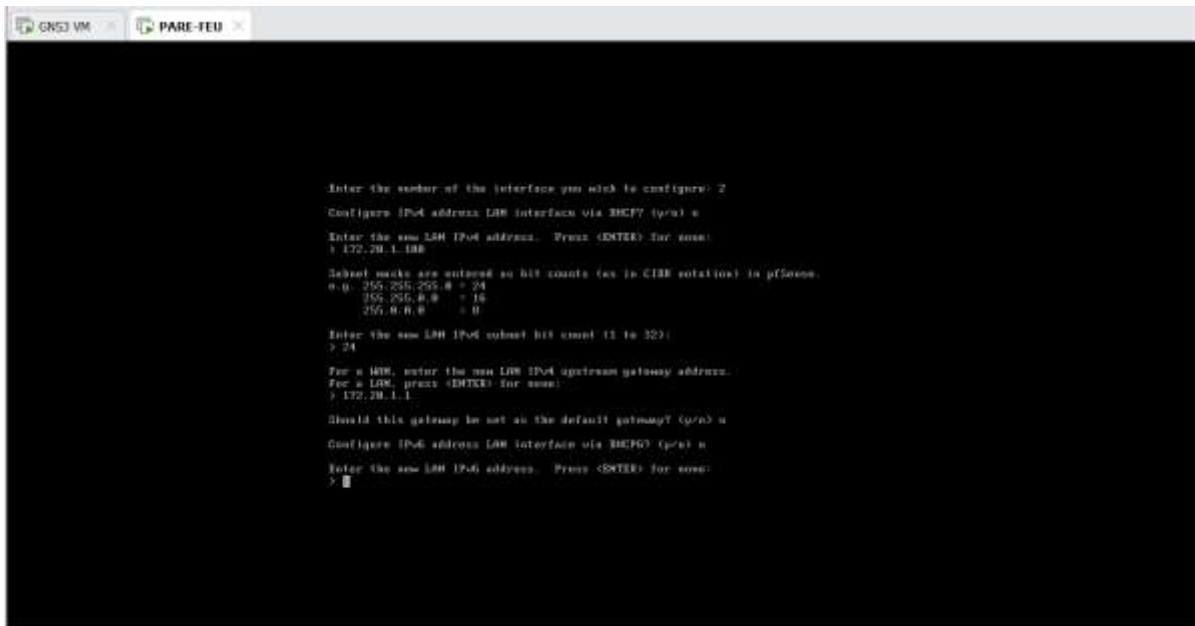
L'installation des machines virtuelles sur VMware Workstation consiste à télécharger l'image iOS de chaque machine souhaité puis suivre les étapes suivant :

1. Ouvrez VMware Workstation.
2. Cliquez sur Fichier (ou File).
3. Sélectionnez Nouvelle machine virtuelle (ou New Virtual Machine).
4. Choisissez le type de configuration (par exemple, **Typique** ou **Custom**).
5. Sélectionnez le support d'installation (ISO ou CD) pour le système d'exploitation que vous souhaitez installer.
6. Configurez les paramètres matériels tels que la mémoire, le processeur et le réseau.

7. Suivez les étapes pour créer la machine virtuelle et installez le système d'exploitation.

3.2.1 Installation de pare-feu pfSense :

PfSense est une distribution open-source basée sur FreeBSD qui permet de créer un pare-feu et un routeur. Il offre des fonctionnalités avancées de sécurité, de filtrage, de gestion du trafic et de VPN.



```
Enter the number of the interface you wish to configure: 2
Configure IP address LAN interface via DHCP? (yes) n
Enter the new LAN IP address. Press <ENTER> for none:
> 172.20.1.100

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
n.b. 255.255.255.0 = 24
      255.255.0.0   = 16
      255.0.0.0    = 8

Enter the new LAN IP address subnet bit count (3 to 32):
> 24

For a WAN, enter the new WAN IP address gateway address.
For a LAN, press <ENTER> for none:
> 172.20.1.1

Should this gateway be set as the default gateway? (yes) n
Configure IP address LAN interface via DHCP? (yes) n
Enter the new LAN IP address. Press <ENTER> for none:
> █
```

Figure IV.3 Installation de PfSense

La seconde consistera à configurer le pare-feu pfsense que nous avons déjà installé ou la configuration de la page d'authentification est nécessaire :

- Tout d'abord il faut se rendre dans le site de pare-feu on tapant l'adresse IP de notre première interface « 172.20.1.100 » dans un navigateur web de la machine locale.

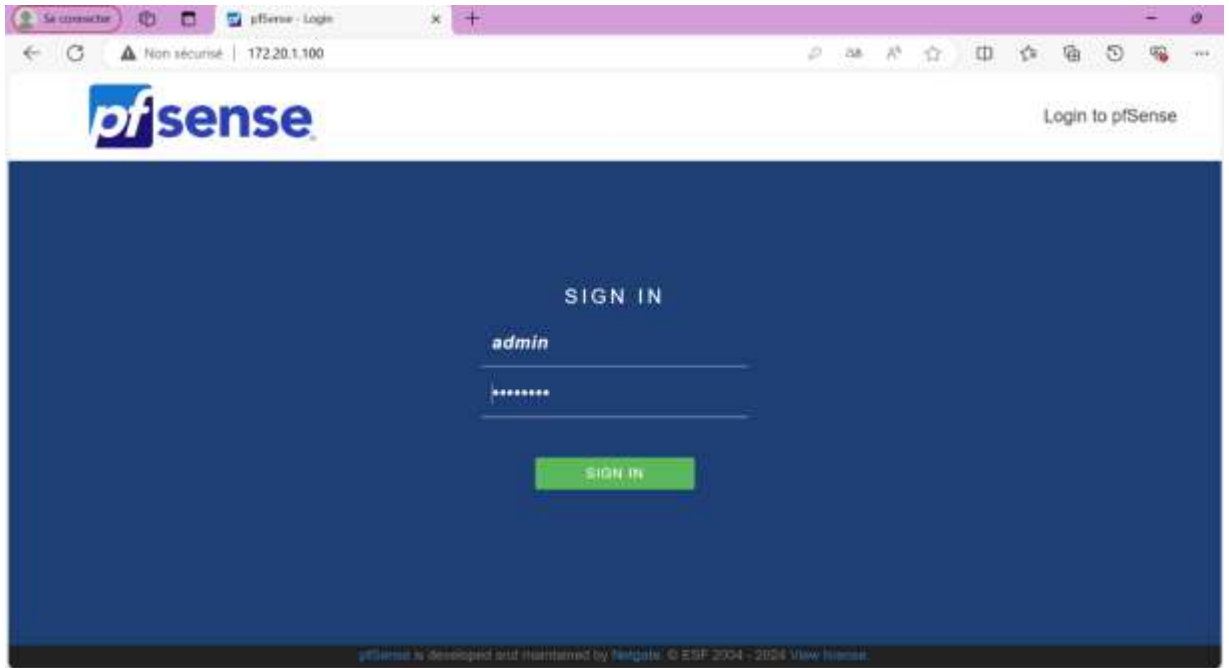


Figure IV.4 La page d'accueil d'authentification

Pour commencer, il faudra d'abord configurer notre pare-feu pfSense par rapport à la topologie qu'on souhaite créer. Les interfaces ajoutées sont montrés dans la figure ci-dessous.



Figure IV.5 Les interfaces configurées sur le pare-feu

3.2.2 Installation du système d'exploitation :

L'installation d'une machine linux est nécessaire pour le bon fonctionnement de notre solution Ansible. Pour cela on a choisi d'installer le système d'exploitation Ubuntu afin de l'utiliser comme un serveur de contrôle où Ansible sera installé plus tard.

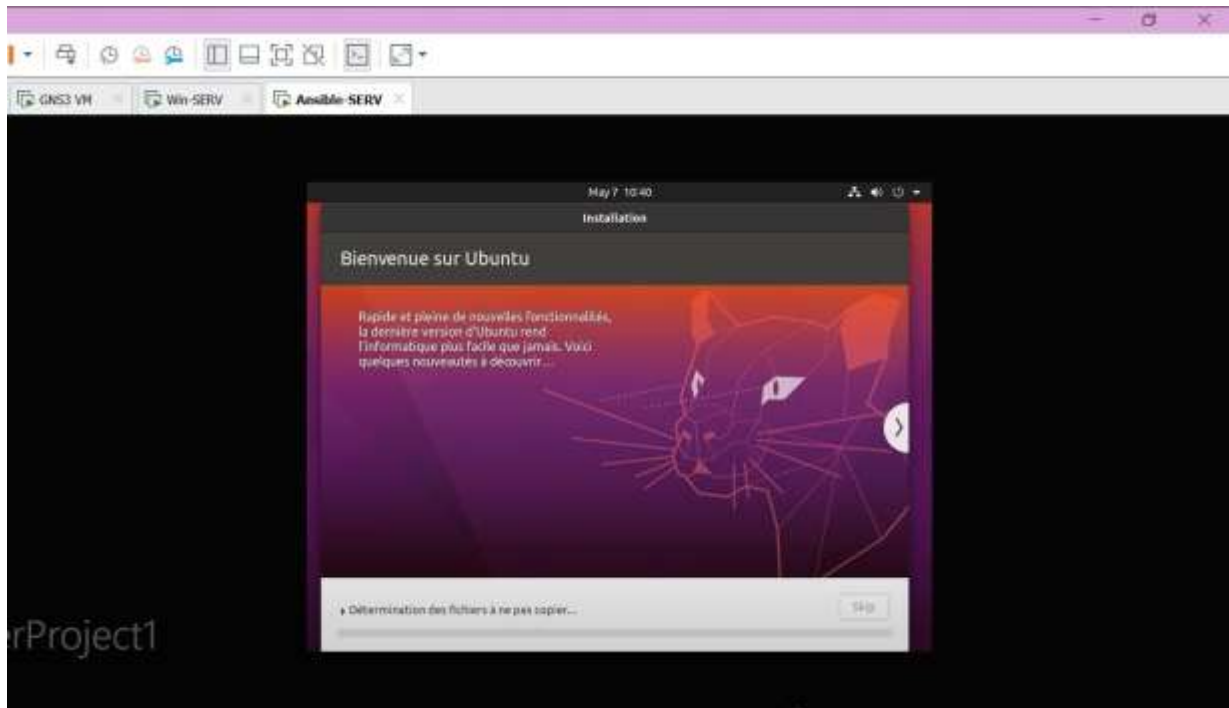


Figure IV.6 Installation de Ubuntu sur VMware

4. Présentation de GNS3 :

GNS3 est un logiciel open source qui permet de simuler des réseaux complexes. Il est largement utilisé par les ingénieurs réseau pour tester des configurations de réseau avant de les déployer en production. Avec GNS3, vous pouvez émuler des équipements réseau de divers fabricants et créer des topologies de réseau qui peuvent être utilisées pour la formation, les tests ou la démonstration. C'est ce que nous a amenés à adopter GNS3 comme un outil de base afin de réaliser notre projet pratique.



4.1 Installation :

L'installation de GNS3 sur Windows est un processus simple. Il suffit de télécharger le logiciel depuis le site officiel de GNS3 <https://gns3.com> et l'installer.



Figure IV.7 Page de téléchargement de GNS3

Une fois téléchargé, l'utilisateur doit suivre les instructions d'installation qui incluent l'installation des logiciels prérequis et optionnels. Il est important de s'assurer que le système d'exploitation est compatible et que la virtualisation est activée dans. La figure ci-dessus montre l'interface graphique après l'installation finale :

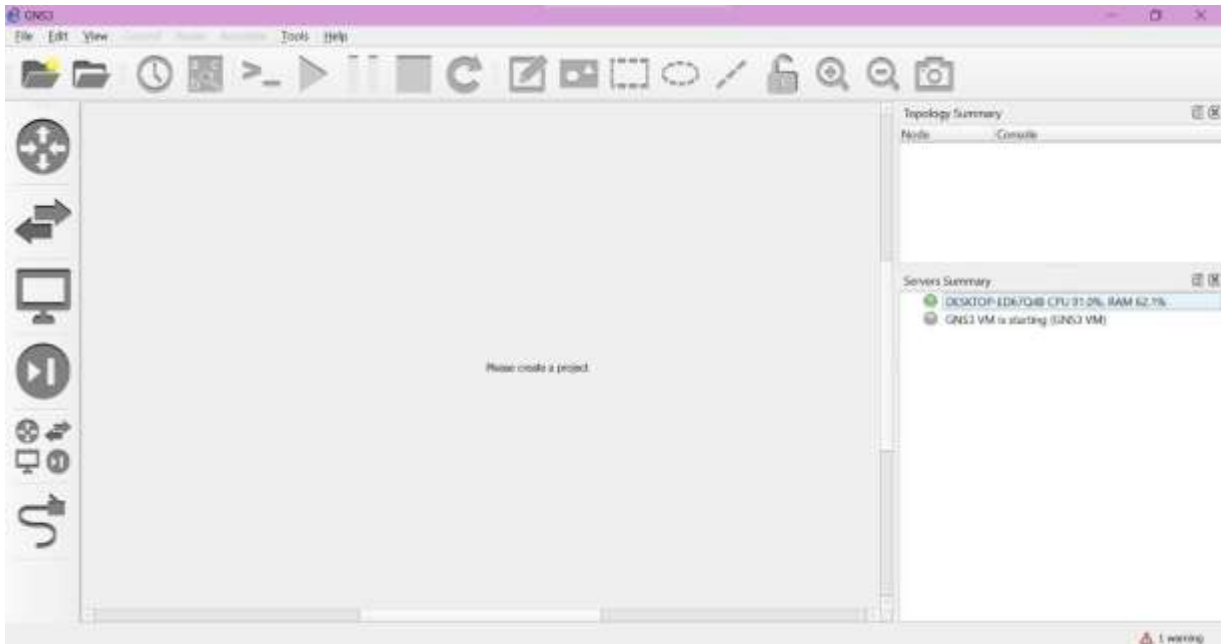


Figure IV.8 L'interface graphique initiale de GNS3

Après avoir installé le GNS3 et le VMWARE Workstation, Maintenant on va faire la liaison entre ces derniers en implémentant la machine virtuelle.

La machine virtuelle GNS3.VM.VMware.Workstation peut être téléchargée à partir du site officiel.

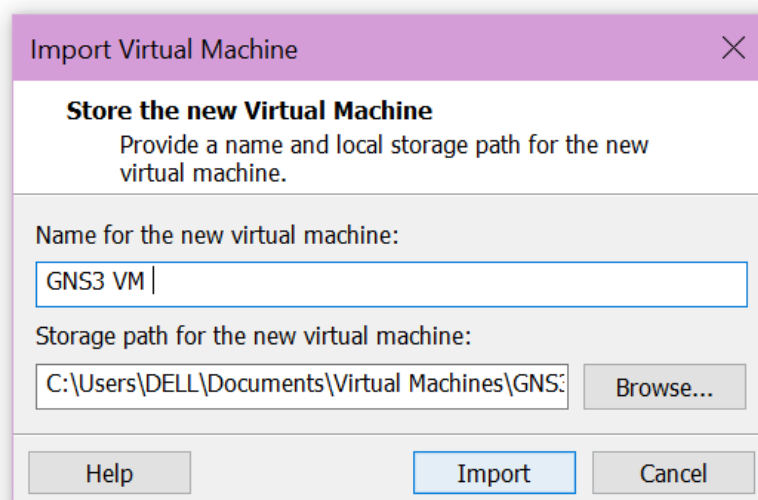


Figure IV.9 Importer la VM sur VMWare Workstation

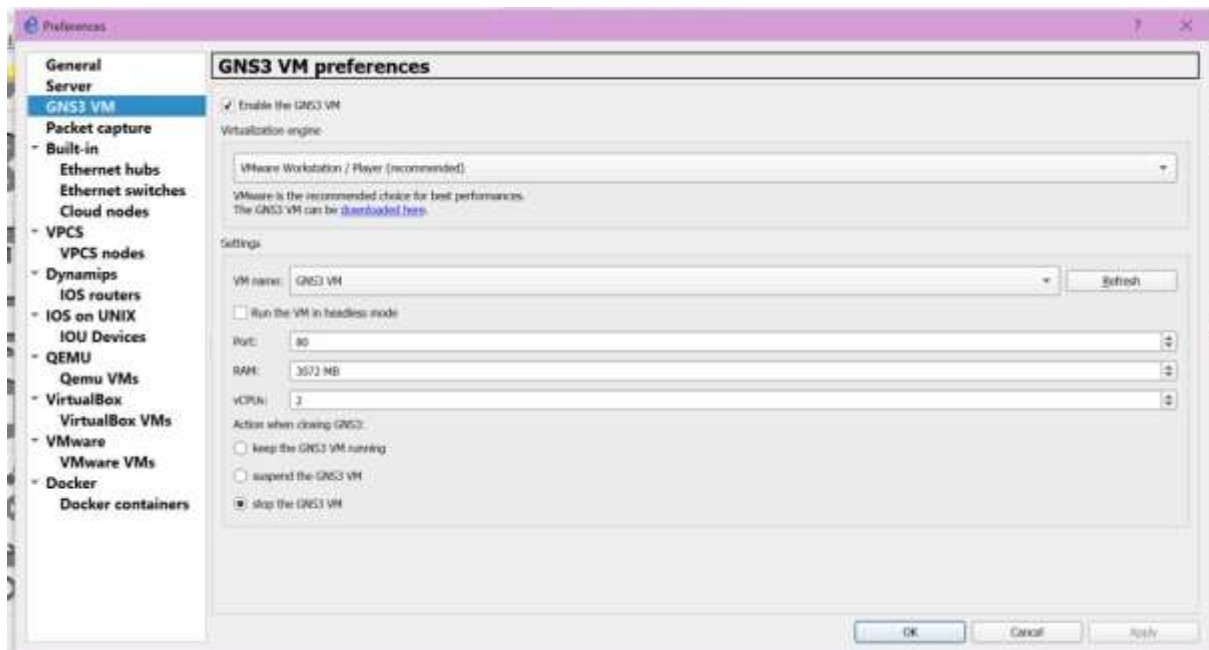


Figure IV.10 importé la VM sur GNS3

A la fin on peut lancer le GNS3 et créer des topologies réseaux et des machines virtuelles.

Pour cela on doit d'abord ajouter les équipements dont nous aurons besoin pour la réalisation de notre maquette.

5. Conception de la topologie :

5.1 Installation des équipements Cisco sur GNS3 :

La réalisation d'une topologie réseau sur GNS3 consiste d'une installation des équipements nécessaires, dans notre cas nous avons besoin d'installer les machines montré sur la figure ci-dessous :

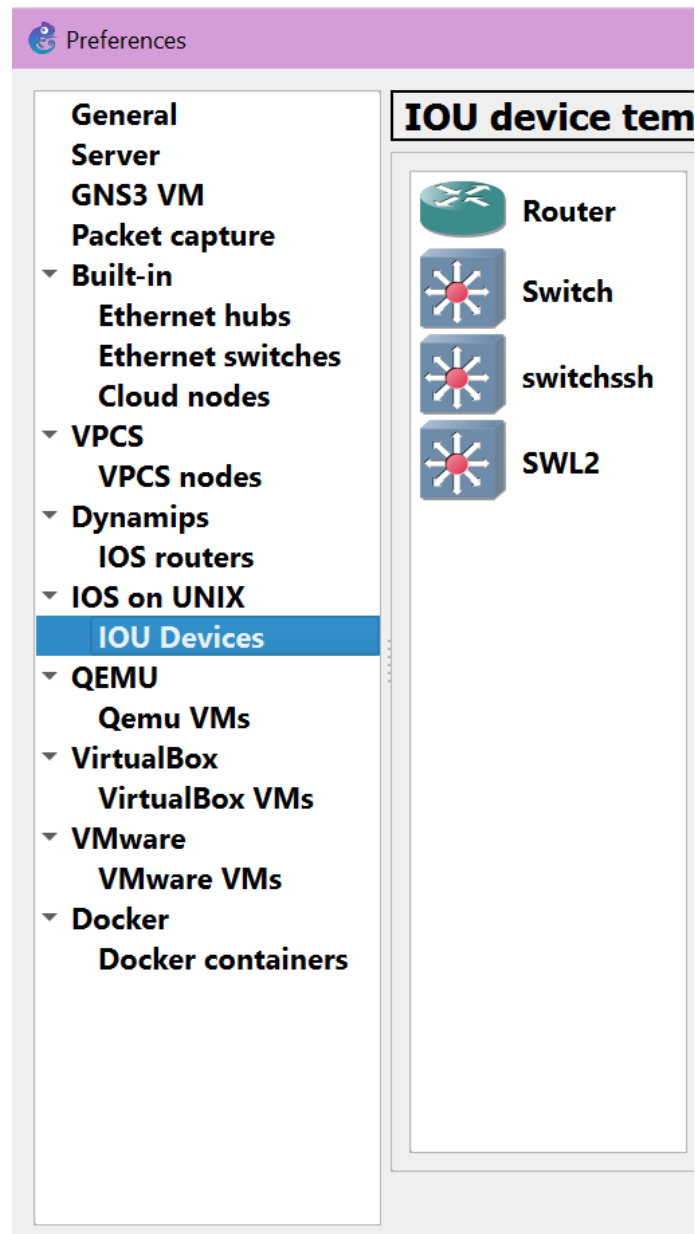


Figure IV.11 Liste des machines installée

5.2 Réalisation de l'architecture réseau :

Nous avons créé notre réseau informatique à base des informations que nous avons récolté durant notre stage chez Candia, Ce modèle comporte des routeurs, des commutateurs et des ordinateurs virtuels qui travaillent ensemble comme un seul système. Nous avons d'abord configuré les connexions entre eux, ensuite on a ajouté un pare-feu pour gérer la sécurité et les connections Internet. De plus, nous avons ajouté un autre réseau supplémentaire nommé DMZ, ce réseau a été conçu au but de pratiquer notre solution.

Ce modèle montré dans la figure ci-dessous.

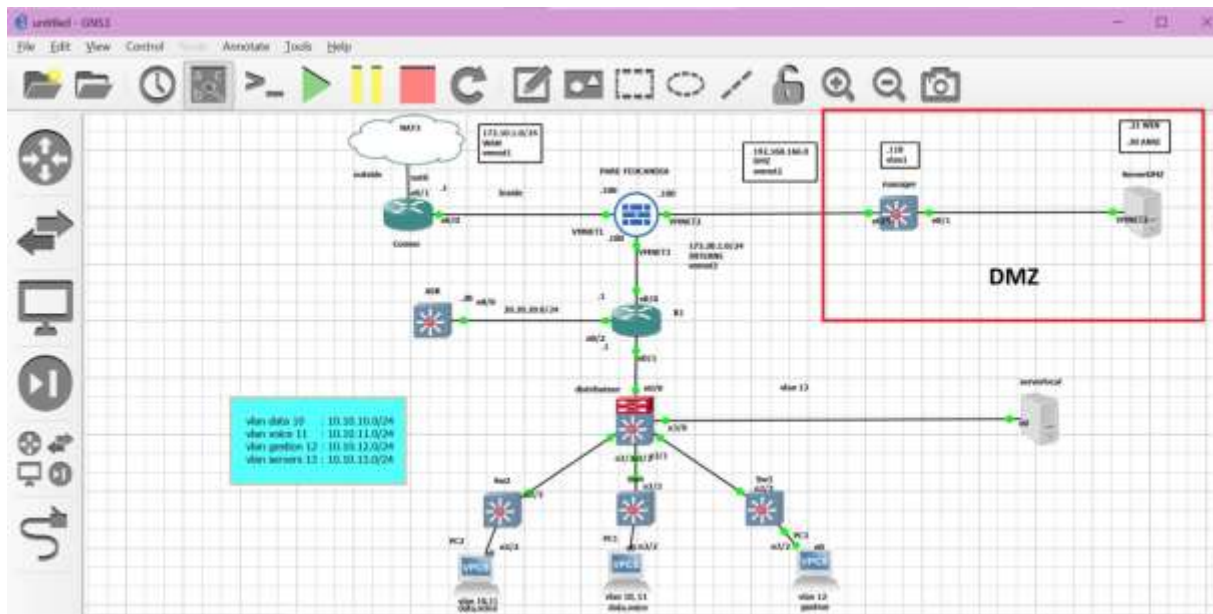


Figure IV.12 Topologie du réseau

5.3 Attribution des adresses IP aux équipements :

Equipement	Interface	Adresse IP	Masque	Passerelle	VMNET
Router conver	e0/0	172.10.1.1	24	/	WAN vmnet1
Pare-feu	Em0	172.10.1.100	24	172.10.1.1	Vmnet1
	Em1	172.20.1.100	24	172.20.1.1	Vmnet3
	Em2	192.168.166.100	24	/	Vmnet2
Switch manager	Vlan1	192.168.166.110	24	/	Vmnet2
ServerDMZ	VMNET2	192.168.166.20	24	192.168.166.110	Vmnet2
R1	E0/0	172.20.1.1	24	/	Vmnet3
	E0/1.10	10.10.10.1	24	/	/
	e0/1.11	10.10.11.1	24	/	/
	e0/1.12	10.10.12.1	24	/	/
	e0/1.13	10.10.13.1	24	/	/

Machine	VLAN	DHCP	Gateway
Pc1	Data10 , Voice11	10.10.10.11/24	10.10.10.1
Pc2	Data10 , Voice11	10.10.10.12/24	10.10.10.1
Pc3	Gestion 12	10.10.12.11/24	10.10.12.1
ServerLocal	Servers 13	10.10.13.11/24	10.10.13.1

Tableau IV-1 Tableau d'adressage

5.4 La configuration de base des équipements :

Afin d'assurer la connectivité entre les différentes machines de notre topologie réseau, nous affectons la configuration nécessaire à chaque équipement :

5.4.1 Création des vlans sur les switches :

La création des Vlan est faite au niveau du switch « distributeur »

```
distributeur#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Et0/1, Et0/2, Et0/3, Et1/0 Et1/1, Et1/2, Et1/3, Et2/0 Et2/1, Et2/2, Et2/3
10	data	active	
11	voice	active	
12	gestion	active	
13	servers	active	Et3/0

Figure IV.13 Création des VLANS

Mettre les interfaces de switch en mode trunk et associer chaque vlan a son interface

```

!
interface Ethernet3/0
  switchport access vlan 13
  switchport mode access
!
interface Ethernet3/1
  switchport trunk allowed vlan 1,10-13
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
interface Ethernet3/2
  switchport trunk allowed vlan 1,10-13
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
interface Ethernet3/3
  switchport trunk allowed vlan 1,10-13
  switchport trunk encapsulation dot1q
  switchport mode trunk
!
interface Vlan1
  no ip address
  shutdown

```

Figure IV.14 Affectation des VLANS aux interfaces

5.4.2 Configuration du pare-feu :

Afin de mettre les connexions entre les interfaces du pare-feu, il est nécessaire d'établir un routage statique pour les Vlan à partir de la page d'accueil.

La figure ci-dessous montre les résultats de la configuration.

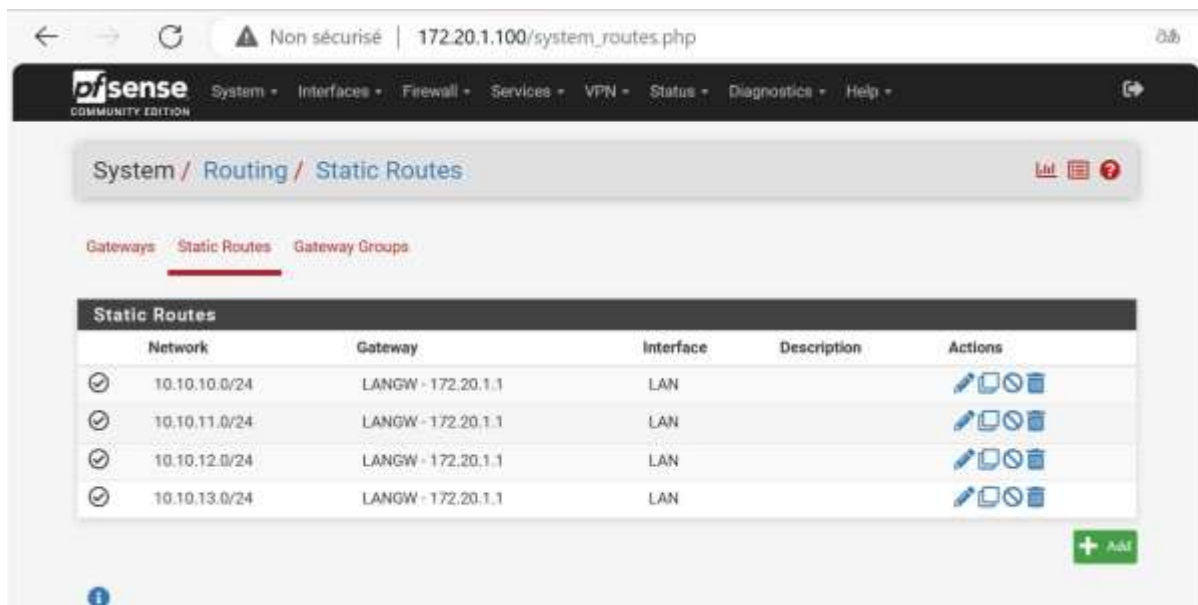


Figure IV.15 Routage statique des VLANS

5.5 Vérification de la connectivité :

Le test de connectivité entre les différentes machines de notre réseau consiste à établir des Ping en utilisant des requêtes ICMP et attendre des réponses REPLY.

Voici quelques tests effectués :

- Ping de PC2 vers Pare-Feu :

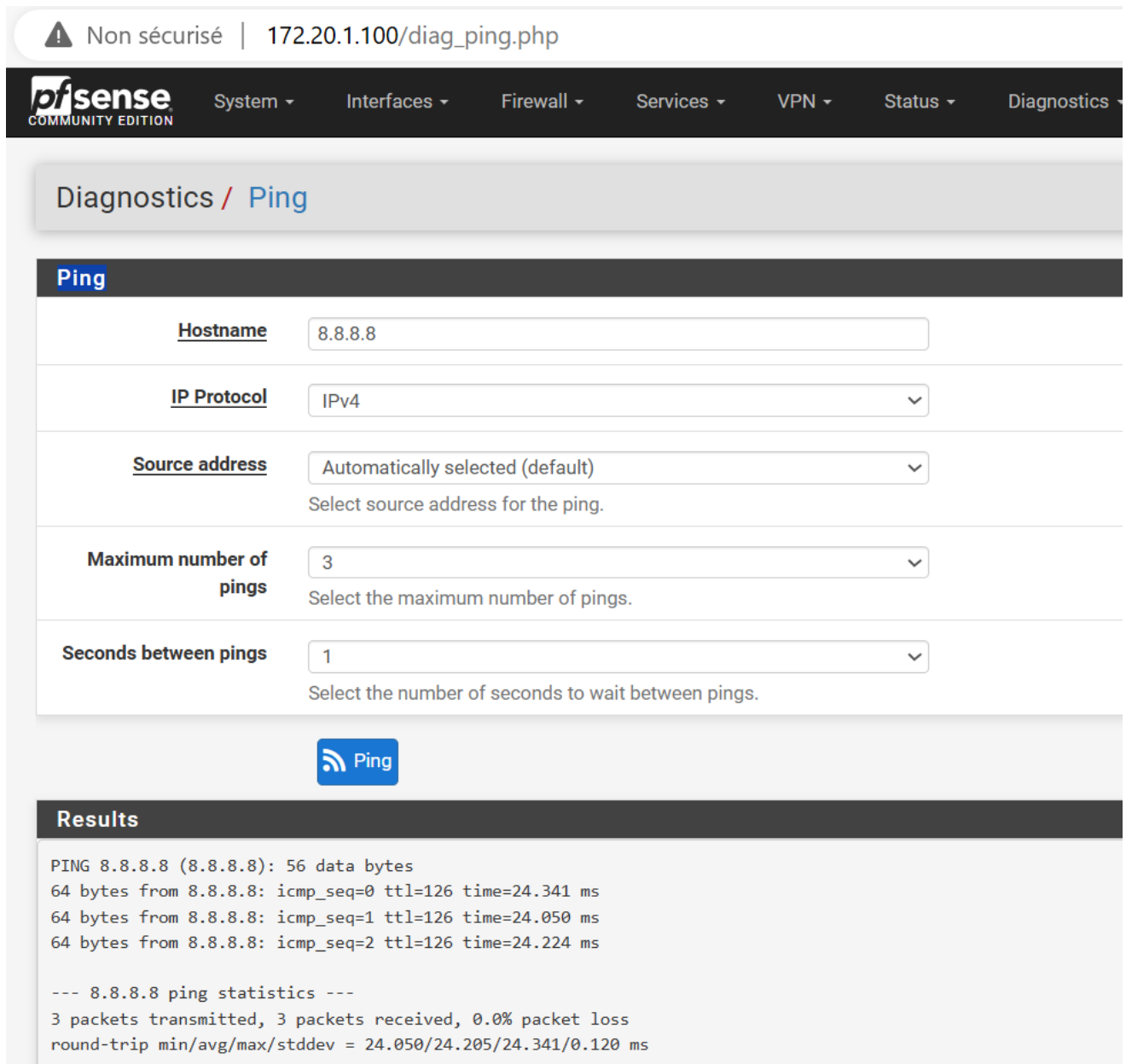
```
PC2> ip dhcp
DDORA IP 10.10.10.11/24 GW 10.10.10.1

PC2> ping 172.20.1.100
84 bytes from 172.20.1.100 icmp_seq=1 ttl=63 time=2.323 ms
84 bytes from 172.20.1.100 icmp_seq=2 ttl=63 time=2.739 ms
84 bytes from 172.20.1.100 icmp_seq=3 ttl=63 time=2.467 ms
84 bytes from 172.20.1.100 icmp_seq=4 ttl=63 time=2.710 ms
84 bytes from 172.20.1.100 icmp_seq=5 ttl=63 time=2.723 ms

PC2>
```

Figure IV.16 Resultat de Ping (PC2-Pare-Feu)

- Ping de Pare-Feu vers internet (NAT) :



The screenshot shows the pfSense web interface. At the top, there is a navigation bar with the pfSense logo and menu items: System, Interfaces, Firewall, Services, VPN, Status, and Diagnostics. The main content area is titled "Diagnostics / Ping". Below this, there is a "Ping" section with several configuration fields:

- Hostname:** 8.8.8
- IP Protocol:** IPv4
- Source address:** Automatically selected (default)
- Maximum number of pings:** 3
- Seconds between pings:** 1

Below the configuration fields is a "Ping" button. Underneath, there is a "Results" section displaying the output of the ping command:

```
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=126 time=24.341 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=126 time=24.050 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=126 time=24.224 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 24.050/24.205/24.341/0.120 ms
```

Figure IV.17 Résultat du Ping (pare-feu - internet)

6. Installation et configuration d'ANSIBLE :

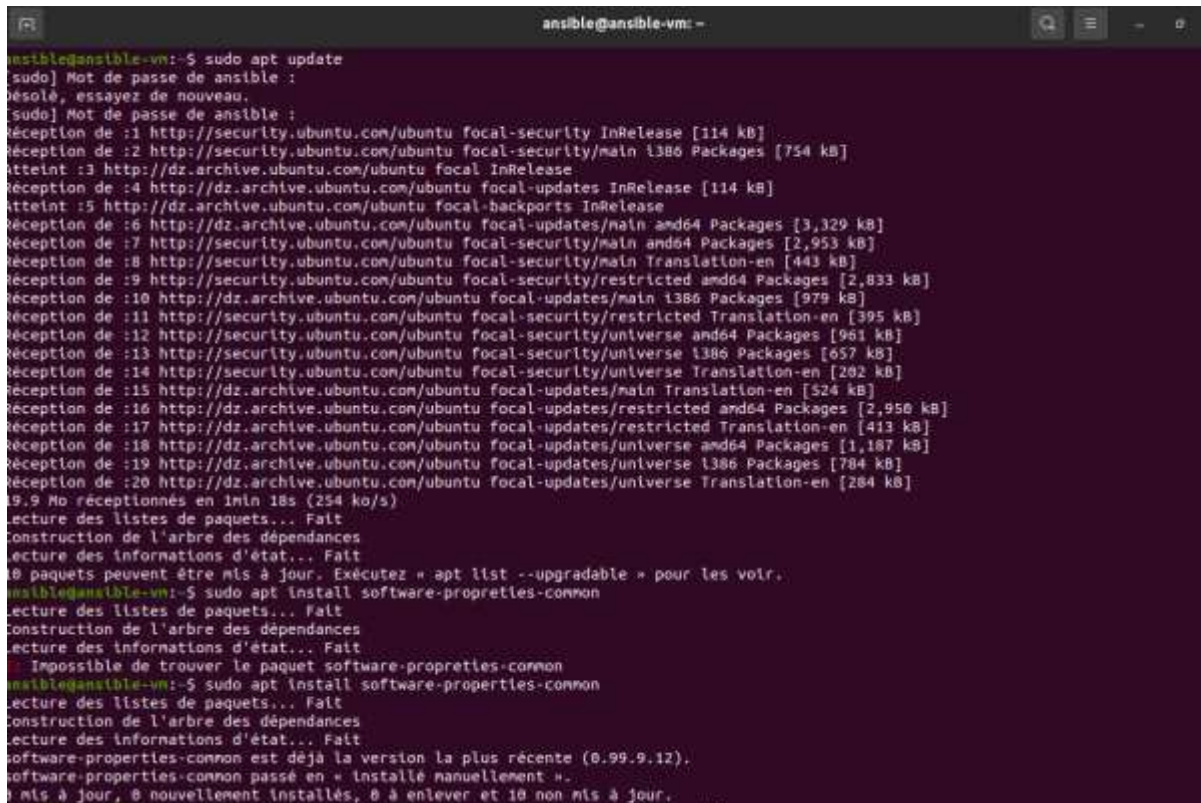
6.1 Installation :

Ansible peut être installé sur une large gamme de systèmes d'exploitation, y compris Linux, macOS et Windows.

Ansible s'installe facilement et uniquement sur serveur de gestion, alors pour installer Ansible sur Debian 10, vous devez effectuer les trois étapes simples suivantes :

6.1.1 Etape 1 : Mettre a jour le système :

Avant d'installer Ansible sur Ubuntu, il faut d'abord le mettre à jour avec les commandes présentées ci-dessous :

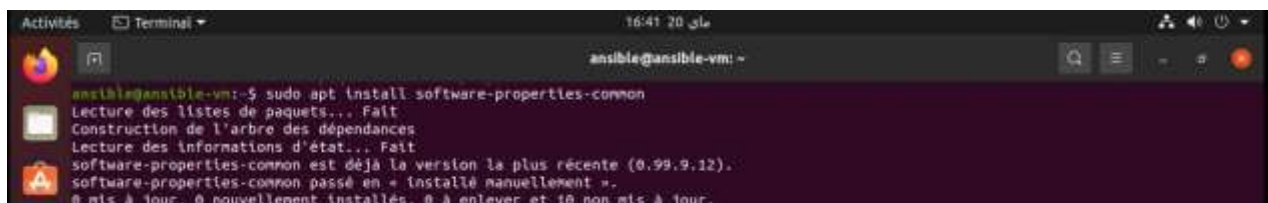


```
ansible@ansible-vm: ~$ sudo apt update
[sudo] Mot de passe de ansible :
[sudo] Mot de passe de ansible :
Réception de :1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Réception de :2 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [754 kB]
Réception de :3 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2,953 kB]
Réception de :4 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [443 kB]
Réception de :5 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [2,833 kB]
Réception de :6 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [395 kB]
Réception de :7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [961 kB]
Réception de :8 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [657 kB]
Réception de :9 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [202 kB]
Réception de :10 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [524 kB]
Réception de :11 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :12 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :13 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :14 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :15 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :16 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :17 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :18 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :19 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
Réception de :20 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [413 kB]
19.9 Mo réceptionnés en 1min 18s (254 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
10 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
ansible@ansible-vm: ~$ sudo apt install software-properties-common
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Impossible de trouver le paquet software-properties-common
ansible@ansible-vm: ~$ sudo apt install software-properties-common
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
software-properties-common est déjà la version la plus récente (0.99.9.12).
software-properties-common passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 10 non mis à jour.
```

Figure IV.18 installation des mis à jour UBUNTU

6.1.2 Etape 2 : installation du paquet software-properties-common :

L'installation de ce paquet peut fournir des scripts utiles pour ajouter et supprimer des référentiels PPA (Personal Package Archives). Pour cela, on utilise la commande montrée ci-dessous :

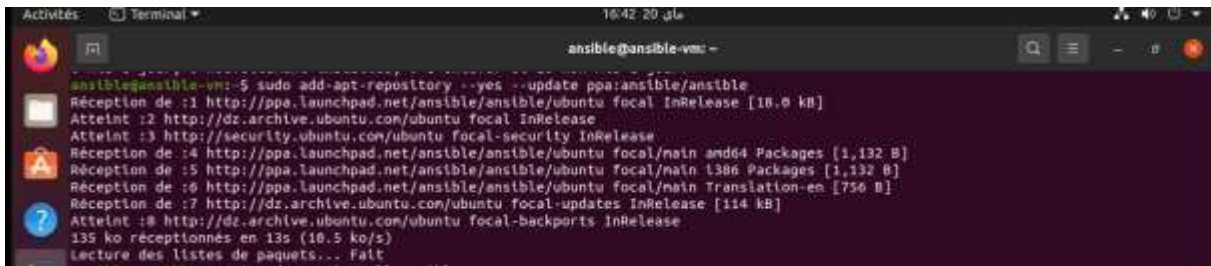


```
ansible@ansible-vm: ~$ sudo apt install software-properties-common
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
software-properties-common est déjà la version la plus récente (0.99.9.12).
software-properties-common passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 10 non mis à jour.
```

Figure IV.19 Installation du paquet

6.1.3 Etape 3 : Ajouter un PPA ANSIBLE sur le système :

La commande ci-dessous permet d'ajouter le référentiel PPA d'Ansible à notre système Ubuntu, ce qui permet d'installer Ansible et d'accéder aux dernières versions du logiciel.



```

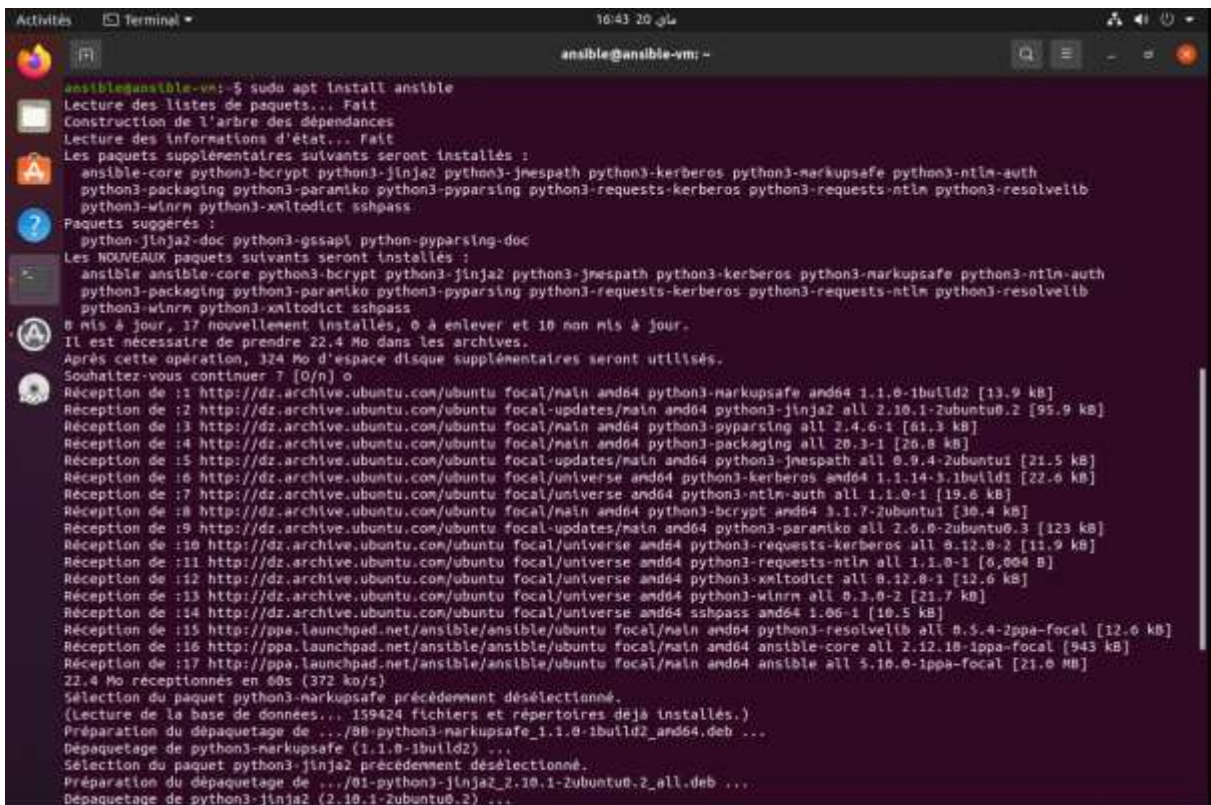
ansible@ansible-vm: ~$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Réception de :1 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease [18.0 kB]
Atteint :2 http://dz.archive.ubuntu.com/ubuntu focal InRelease
Atteint :3 http://security.ubuntu.com/ubuntu focal-security InRelease
Réception de :4 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 Packages [1,132 B]
Réception de :5 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main i386 Packages [1,132 B]
Réception de :6 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main Translation-en [750 B]
Réception de :7 http://dz.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Atteint :8 http://dz.archive.ubuntu.com/ubuntu focal-backports InRelease
135 ko réceptionnés en 13s (10.5 ko/s)
Lecture des listes de paquets... Fait

```

Figure IV.20 Ajouter le PPA d'Ansible

6.1.4 Etape 4 : Installation d'Ansible sur le système

Une fois que la mise à jour du système est terminée, vous pouvez installer Ansible sur Ubuntu avec la commande indiquée ci-dessous :



```

ansible@ansible-vm: ~$ sudo apt install ansible
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  ansible-core python3-bcrypt python3-jinja2 python3-jmespath python3-kerberos python3-markupsafe python3-ntlm-auth
  python3-packaging python3-paramiko python3-pyparsing python3-requests-kerberos python3-requests-ntlm python3-resolvlib
  python3-wlrm python3-xltdict sshpass
Paquets suggérés :
  python-jinja2-doc python3-gssapi python-pyparsing-doc
Les NOUVEAUX paquets suivants seront installés :
  ansible ansible-core python3-bcrypt python3-jinja2 python3-jmespath python3-kerberos python3-markupsafe python3-ntlm-auth
  python3-packaging python3-paramiko python3-pyparsing python3-requests-kerberos python3-requests-ntlm python3-resolvlib
  python3-wlrm python3-xltdict sshpass
0 mis à jour, 17 nouvellement installés, 0 à enlever et 10 non mis à jour.
Il est nécessaire de prendre 22,4 Mo dans les archives.
Après cette opération, 324 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://dz.archive.ubuntu.com/ubuntu focal/main amd64 python3-markupsafe amd64 1.1.0-1build2 [13.9 kB]
Réception de :2 http://dz.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jinja2 all 2.10.1-2ubuntu0.2 [95.9 kB]
Réception de :3 http://dz.archive.ubuntu.com/ubuntu focal/main amd64 python3-pyparsing all 2.4.6-1 [61.3 kB]
Réception de :4 http://dz.archive.ubuntu.com/ubuntu focal/main amd64 python3-packaging all 20.3-1 [26.8 kB]
Réception de :5 http://dz.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-jmespath all 0.9.4-2ubuntu1 [21.5 kB]
Réception de :6 http://dz.archive.ubuntu.com/ubuntu focal/universe amd64 python3-kerberos amd64 1.1.14-3.1build1 [22.6 kB]
Réception de :7 http://dz.archive.ubuntu.com/ubuntu focal/universe amd64 python3-ntlm-auth all 1.1.0-1 [19.6 kB]
Réception de :8 http://dz.archive.ubuntu.com/ubuntu focal/main amd64 python3-bcrypt amd64 3.1.7-2ubuntu1 [30.4 kB]
Réception de :9 http://dz.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-paramiko all 2.6.0-2ubuntu0.3 [123 kB]
Réception de :10 http://dz.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Réception de :11 http://dz.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-ntlm all 1.1.0-1 [6,004 B]
Réception de :12 http://dz.archive.ubuntu.com/ubuntu focal/universe amd64 python3-xltdict all 0.12.0-1 [12.6 kB]
Réception de :13 http://dz.archive.ubuntu.com/ubuntu focal/universe amd64 python3-wlrm all 0.3.0-2 [21.7 kB]
Réception de :14 http://dz.archive.ubuntu.com/ubuntu focal/universe amd64 sshpass amd64 1.06-1 [10.5 kB]
Réception de :15 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 python3-resolvlib all 0.5.4-2ppa-focal [12.6 kB]
Réception de :16 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 ansible-core all 2.12.10-1ppa-focal [943 kB]
Réception de :17 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 ansible all 5.10.0-1ppa-focal [21.0 MB]
22,4 Mo réceptionnés en 68s (372 ko/s)
sélection du paquet python3-markupsafe précédemment désélectionné.
(Lecture de la base de données... 159424 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../98-python3-markupsafe_1.1.0-1build2_amd64.deb ...
Dépaquetage de python3-markupsafe (1.1.0-1build2) ...
Sélection du paquet python3-jinja2 précédemment désélectionné.
Préparation du dépaquetage de .../01-python3-jinja2_2.10.1-2ubuntu0.2_all.deb ...
Dépaquetage de python3-jinja2 (2.10.1-2ubuntu0.2) ...

```

Figure IV.21 Installation d'Ansible

6.1.5 Etape 5 : Vérification et confirmation de l'installation d'Ansible

Afin que nous soyons sûrs que l'installation d'Ansible a été bien effectuée, nous allons vérifier en tapant une partie de la commande 'Ansible' puis « Tab ».



Figure IV.22 Vérification d'installation Ansible

7. Le protocole Secure Shell :

Le SSH (Secure Shell) est un protocole de communication sécurisé utilisé pour accéder à distance à des systèmes informatiques.

7.1 SSH sur les équipements Cisco :

7.1.1 Activer SSH sur le switches :

Pour configurer SSH sur les switches, il est nécessaire de suivre des étapes méthodiques. Tout d'abord, il faut renommer l'équipement pour assurer une identification facile. Ensuite, la création d'un compte utilisateur sécurisé par un mot de passe. Il est également important de définir un nom de domaine spécifique à l'équipement.

Ensuite, on passe à la génération de clés RSA. Pour cela, il est recommandé de choisir une longueur de clés de 1024 bits pour une sécurité optimale.

La configuration de la ligne virtuelle VTY est nécessaire pour permettre les connexions à distance. L'utilisation du compte utilisateur précédemment créé est requise pour ces connexions. Ces étapes garantissent une configuration sécurisée et efficace du protocole SSH sur les switches.


```
manager(config)#ip domain-name candia.ssh
manager(config)#crypto key generate rsa
The name for the keys will be: manager.candia.ssh
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

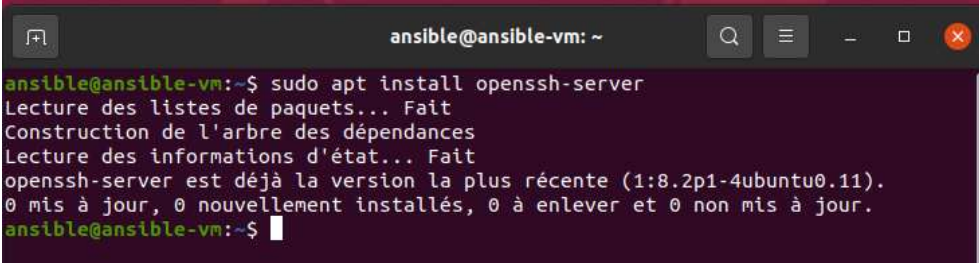
How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

manager(config)#
*Jun 25 14:55:43.145: %SSH-5-ENABLED: SSH 1.99 has been enabled
manager(config)#username candia privilege 15 password candia
manager(config)#ip ssh version 2
manager(config)#line vty 0 4
manager(config-line)#transport input ssh
manager(config-line)#login local
manager(config-line)#end
```

Figure IV.23 Configuration du protocole SSH sur le switch manager

7.2 SSH sur la machine virtuelle Ubuntu :

Pour activer le protocole SSH sur machine UBUNTU il suffit de télécharger le paquet ver« Open SSH Server », pour cela on doit exécuter la commande suivante : « sudo apt install openssh-server »

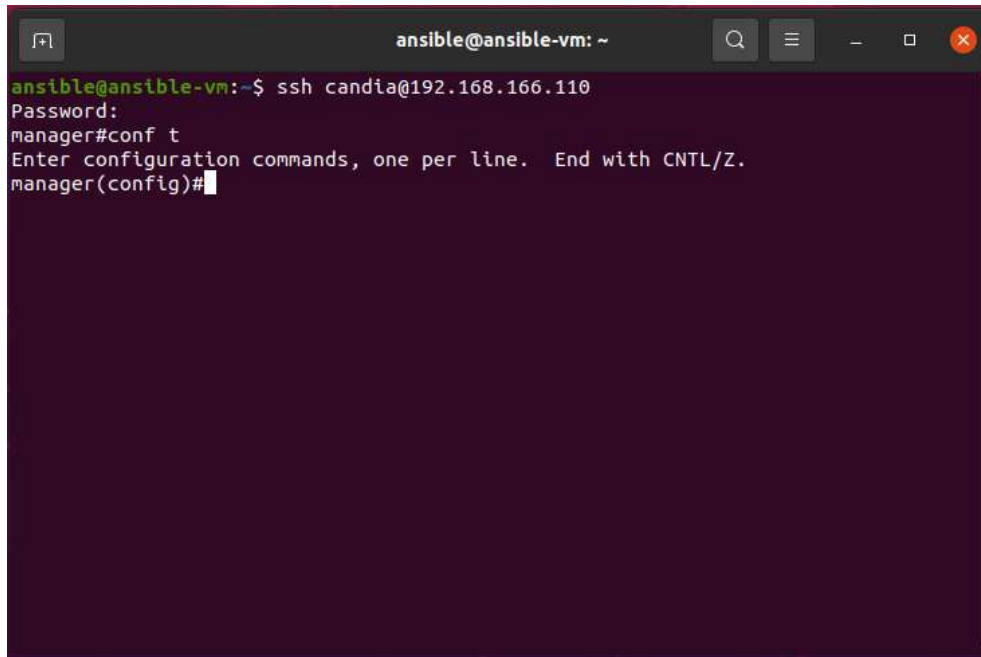
A screenshot of a terminal window titled 'ansible@ansible-vm: ~'. The terminal shows the command 'sudo apt install openssh-server' being executed. The output indicates that the package is already installed and up-to-date. The terminal text is as follows:

```
ansible@ansible-vm:~$ sudo apt install openssh-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
openssh-server est déjà la version la plus récente (1:8.2p1-4ubuntu0.11).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
ansible@ansible-vm:~$
```

Figure IV.24 Installation de l'Open SSH server

7.3 Teste de connexion à distance via SSH :

Pour pouvoir accéder à notre machine via le protocole SSH, il suffit d'introduire ces commandes comme illustrées sur la figure suivante :



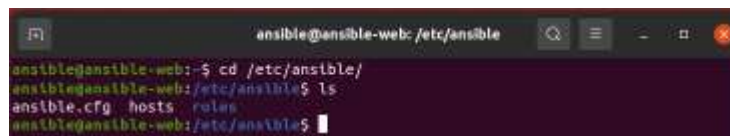
```
ansible@ansible-vm: ~  
ansible@ansible-vm:~$ ssh candia@192.168.166.110  
Password:  
manager#conf t  
Enter configuration commands, one per line.  End with CNTL/Z.  
manager(config)#
```

Figure IV.IV.25 : Test de connectivité via SSH

8. Déploiement de la solution :

8.1 Démarche de déploiement :

L'installation de l'outil Ansible fournit un ensemble des fichiers et répertoires par défaut tel que :



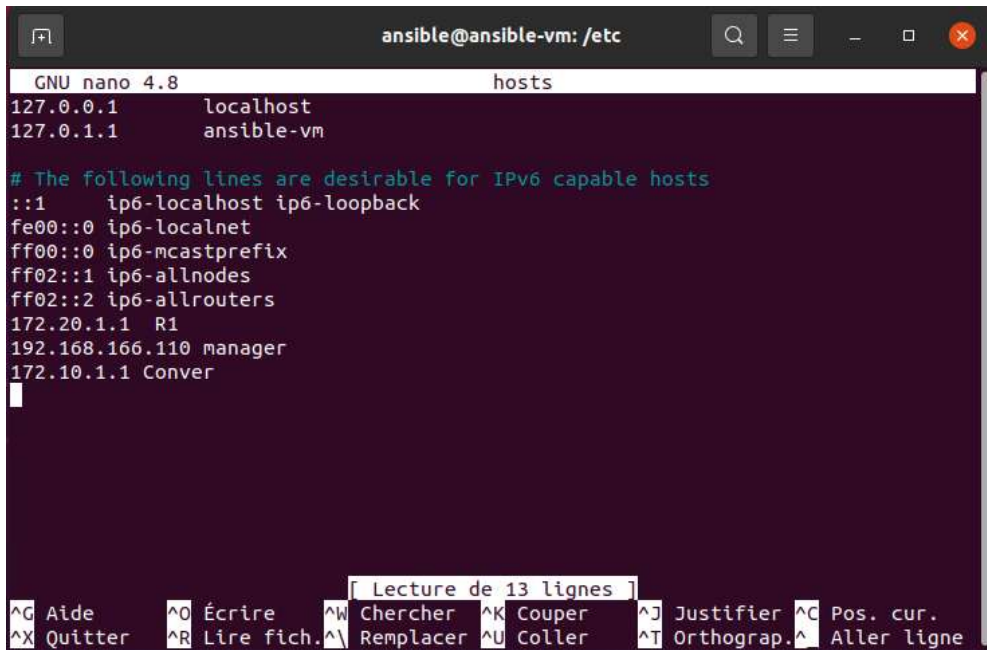
```
ansible@ansible-web: /etc/ansible  
ansible@ansible-web:~$ cd /etc/ansible/  
ansible@ansible-web:/etc/ansible$ ls  
ansible.cfg  hosts  roles  
ansible@ansible-web:/etc/ansible$
```

Figure IV.26 Contenu d'Ansible

- **Le fichier Ansible.cfg :** Ce fichier permet de modifier certains paramètres Ansible, la configuration de base est souvent suffisante, mais il se peut que certaines modifications des paramètres soient nécessaires afin d'assurer la connectivité et le bon fonctionnement de l'automatisation.
- **Le fichier hosts :** Il s'agit généralement du fichier d'inventaire où vous définissez les hôtes et les groupes d'hôtes sur lesquels les commandes, modules et tâches d'un playbook sont exécutés.
- **Le répertoire roles :** Les rôles dans Ansible sont des ensembles de tâches (Tasks), de fichiers et de variables regroupés de manière logique. Ils permettent d'organiser et de réutiliser du code de manière modulaire.

8.1.1 Creation des inventaires :

Dans l'emplacement /etc/ de notre machine UBUNTU on trouve le fichier 'hosts', Il s'agit d'un fichier texte simple qui associe des adresses IP statiques à des noms d'hôtes spécifiques. Ce fichier est consulté avant que le système n'accède au serveur DNS pour résoudre les noms de domaine. Il peut être utilisé pour bloquer l'accès à des sites web, on a utilisé ce fichier pour associer chacun des équipements : les routeurs 'R1' et 'Conver', le switch 'manager' à leurs adresses IP, cela facilite plusieurs tâches et surtout celle de créations d'inventaire.



```

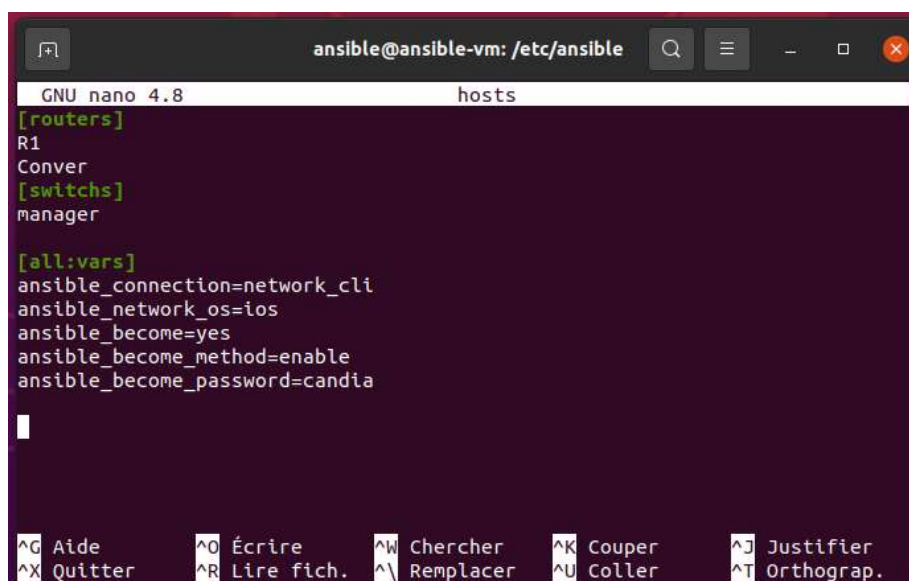
ansible@ansible-vm: /etc
GNU nano 4.8 hosts
127.0.0.1 localhost
127.0.1.1 ansible-vm

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.20.1.1 R1
192.168.166.110 manager
172.10.1.1 Conver

```

Figure IV.27 Le fichier hosts

Maintenant la création d'un fichier inventory sera plus simple, on a choisi de travailler avec le fichier « hosts » situé dans l'emplacement /etc/Ansible/



```

ansible@ansible-vm: /etc/ansible
GNU nano 4.8 hosts
[routers]
R1
Conver
[switchs]
manager

[all:vars]
ansible_connection=network_cli
ansible_network_os=ios
ansible_become=yes
ansible_become_method=enable
ansible_become_password=candia

```


Figure IV.28 Fichier d'inventary

Dans ce fichier, les termes entre les crochets [] définissent un groupe, par exemple [routers] est un groupe qui contient les routeurs R1 et Converg, [switchs] est un groupe qui contient les switchs manager, notant qu'un groupe [all : vars] reprend tous les groupes et tous les hosts définie dans l'inventary.

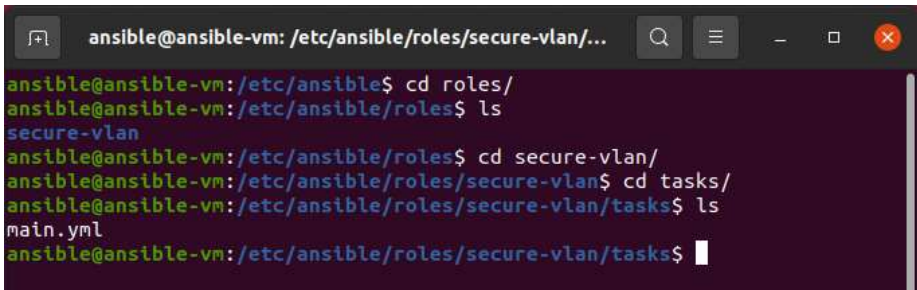
Il y a deux méthodes d'automatiser les configurations, La première méthode est d'écrire les playbooks de la manière normale et la seconde méthode est d'utiliser les rôles.

Dans notre cas pratique, on utilisera la méthode des Rôles, car cette méthode est plus pratique pour gérer et exécuter plusieurs playbooks en même temps. En d'autres termes, les rôles sont des playbooks génériques, qui peuvent être intégrés dans d'autres playbooks. Ce concept est essentiel pour créer des tâches complexes. En fait, pour rendre les playbooks lisibles, il vaut mieux construire des rôles qui peuvent être partagés, distribués et assemblés, plutôt que créer un playbook qui sera difficile à maintenir et complexe à utiliser et peu lisible.

8.1.2 Création des rôles :

Dans le répertoire '/etc/Ansible/roles/', EN utilisant la commande « mkdir » nous avons créé un nouveau répertoire intitulé 'secure-vlan'.

Ce répertoire contient plusieurs sous-dossiers, dont un nommé 'tasks'. Dans le dossier 'tasks', nous avons ajouté un fichier 'main.yml' où seront écrits les différents modules Ansible nécessaires pour implémenter notre configuration sécurisée du VLAN.



```
ansible@ansible-vm: /etc/ansible/roles/secure-vlan/...
ansible@ansible-vm:/etc/ansible$ cd roles/
ansible@ansible-vm:/etc/ansible/roles$ ls
secure-vlan
ansible@ansible-vm:/etc/ansible/roles$ cd secure-vlan/
ansible@ansible-vm:/etc/ansible/roles/secure-vlan$ cd tasks/
ansible@ansible-vm:/etc/ansible/roles/secure-vlan/tasks$ ls
main.yml
ansible@ansible-vm:/etc/ansible/roles/secure-vlan/tasks$
```

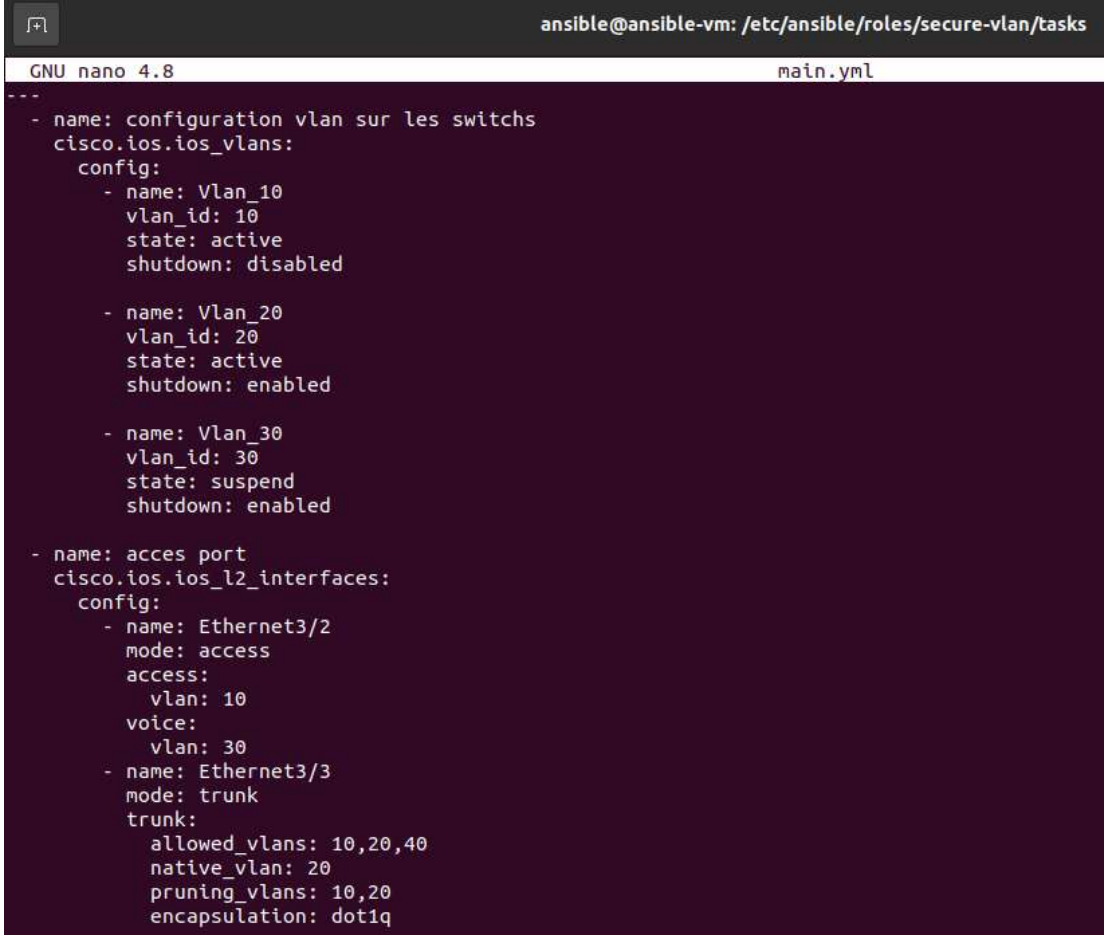
Figure IV.29 Contenu de repertoire roles

Cette structuration permet de gérer nos tâches de manière organisée et modulaire, ce qui facilite la maintenance et la réutilisation des configurations.

Maintenant que nous avons notre fichier « main.yml » on peut commencer à ajouter des modules. Le site de la documentation officiel d'Ansible fourni l'ensemble entière des modules qu'on pourra utiliser sur divers équipements.

Nous avons donc consulté le site afin de prendre les instructions nécessaires pour la création et la configuration des vlan.

Le résultat est montré dans la figure suivante :



```
ansible@ansible-vm: /etc/ansible/roles/secure-vlan/tasks
GNU nano 4.8 main.yml
--
- name: configuration vlan sur les switches
  cisco.ios.ios_vlans:
    config:
      - name: Vlan_10
        vlan_id: 10
        state: active
        shutdown: disabled

      - name: Vlan_20
        vlan_id: 20
        state: active
        shutdown: enabled

      - name: Vlan_30
        vlan_id: 30
        state: suspend
        shutdown: enabled

- name: acces port
  cisco.ios.ios_l2_interfaces:
    config:
      - name: Ethernet3/2
        mode: access
        access:
          vlan: 10
        voice:
          vlan: 30
      - name: Ethernet3/3
        mode: trunk
        trunk:
          allowed_vlans: 10,20,40
          native_vlan: 20
          pruning_vlans: 10,20
          encapsulation: dot1q
```

Figure IV.30 TASK de configuration des vlan

Ce TASK est utilisé pour configurer les Vlans en deux étapes en effectuant deux modules, le premier module intitulé : "configuration vlan sur les switches" est de créer des vlans dans tous les switches, et le deuxième module "access port" permet de la configuration Access port par affectation des interfaces à des Vlans, plus le mode trunk qui permet de configurer l'agrégation entre les switches pour transporter le trafic de plusieurs Vlans.

8.1.3 Creation des playbooks :

La création des playbooks consiste à créer un fichier format .yml dans le répertoire Ansible.

On le nomme et on suit sa syntaxe de création qui est montré dans la figure suivante :

The image shows a terminal window with a dark background. At the top, the prompt is 'ansible@ansible-vm: /etc/ansible'. Below that, the terminal title is 'GNU nano 4.8' and the file name is 'playbook-secure-vlan.yml'. The content of the file is as follows:

```
- -
- name: configuration de base des routeurs et switchs cisco
  hosts: switches
  gather_facts: false

  pre_tasks:
    - debug:
        msg: 'debut de la configuration.'
```

Figure IV.31 Playbook

En général, un playbook est composé de plusieurs play « jeu », ici on a créé un seul play nommé « configuration de base des routeurs et switchs »

Ce playbook va permettre d'exécuter le role (secure-vlan) sur l'ensemble des hosts contenue dans le groupe [switchs], ce groupe comprend le switch « manager ».

Nous terminons cette partie par un résumé de toutes les étapes nécessaires pour établir la communication et le droit de lancer des playbooks :

On termine cette partie par un résumé de toutes les étapes nécessaires pour établir la communication et le droit de lancer des playbooks :

- 1- Vérifier que Python est installé sur le serveur Linux.
- 2- installer Ansible sur la distribution Linux.
- 3- Configurer SSH sur tous les périphériques de l'infrastructure.
- 4- Créer un utilisateur avec privilège 15 et password sécurisé sur tous les périphériques.
- 5- Vérifier le ping entre le serveur Ansible et les périphériques.
- 6- Vérifier la connexion SSH entre le server Ansible et les périphériques.
- 7- Définir un inventory des périphériques.
- 8- Vous devez écrire les playbooks.
- 9- Finalement, vous pouvez lancer le Playbook.

8.2 Exécution et vérification :

Une fois le playbook est prêt, nous injectons le fichier « playbook-secure-vlan.yml » sur un groupe de nœuds situé dans le fichier d'inventaire « hosts.yml » en utilisant la commande « Ansible-playbook ».

```
ansible@ansible-vm:/etc/ansible$ ansible-playbook playbook-secure-vlan.yml -u candia -k
```

Figure IV.32 Commande d'exécution du playbook

Les résultats de cette automatisation apparaissent quelques secondes après l'exécution de cette commande.

On peut voir que l'exécution a été bien effectuée :

```
ansible@ansible-vm:/etc/ansible/roles$ cd ..
ansible@ansible-vm:/etc/ansible$ ansible-playbook playbook-secure-vlan.yml -u candia -k
SSH password:

PLAY [configuration de base des routeurs et switchs cisco] *****
TASK [debug] *****
ok: [manager] => {
  "msg": "debut de la configuration."
}
TASK [secure-vlan : configuration vlan 10 et 11 sur les switchs] *****
ok: [manager]
TASK [secure-vlan : Merge provided configuration with device configuration] *****
changed: [manager]
TASK [debug] *****
ok: [manager] => {
  "msg": "hosts configurés."
}
PLAY RECAP *****
manager                : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ansible@ansible-vm:/etc/ansible$
```

Figure IV.33 Résultat de playbook

Pour vérifier la réussite de notre simulation, on vérifie si le switch « manager » a eu des changements dans sa configuration en utilisant les commandes suivantes :

```

manager#show vlan brief

VLAN Name                Status    Ports
-----
1    default                active    Et0/2, Et0/3, Et1/0, Et1/1
                                           Et1/2, Et1/3, Et2/0, Et2/1
                                           Et2/2, Et2/3, Et3/0, Et3/1
                                           Et3/2, Et3/3

10   Vlan_10                 active
20   Vlan_20                 act/lshut
30   Vlan_30                 sus/lshut
1002 fddi-default           act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default       act/unsup
1005 trnet-default         act/unsup
manager#
manager#

```

Figure IV.34 Listes des vlans ajouté sur le switch manager

```

manager#show interfaces trunk

Port      Mode           Encapsulation  Status        Native vlan
Et0/0     desirable     n-isl          trunking      1
Et0/1     desirable     n-isl          trunking      1
Et3/3     on             802.1q         trunking      20

Port      Vlans allowed on trunk
Et0/0     1-4094
Et0/1     1-4094
Et3/3     10,20,40

Port      Vlans allowed and active in management domain
Et0/0     1,10
Et0/1     1,10
Et3/3     10

Port      Vlans in spanning tree forwarding state and not pruned
Et0/0     1,10
Et0/1     none
Et3/3     10
manager#

```

Figure IV.35 Liste de modifications sur les ports

En tapant la commande « show vlan brief » pour afficher la liste des vlans sur le switch manager, on remarque que les vlan10, vlan20 et vlan30 ont été bien ajouté, ensuite on vérifie l'affectation des interfaces à des vlans et le mode truck. Cela confirme la réussite de notre simulation.

9. Conclusion :

Ce chapitre illustre concrètement la mise en œuvre de notre solution à savoir l'automatisation de réseau et système qui comporte : la virtualisation, l'installation des équipements réseau, et la configuration de la topologie réseau. L'utilisation d'Ansible pour la simulation a démontré son efficacité en fournissant une gestion plus agile et une meilleure réactivité face aux exigences évolutives des environnements informatiques modernes. On peut donc conclure qu'Ansible offre une excellente solution pour l'automatisation des infrastructures réseaux et systèmes.

CONCLUSION GENERALE

Vue la complexité croissante des infrastructures réseaux et systèmes, composées de centaines d'équipements hétérogènes, les administrateurs de réseaux informatiques font face à de fortes pressions pour assurer un fonctionnement optimal de ces environnements critiques. Pour cela, il devient plus qu'indispensable de fournir aux administrateurs réseaux des outils qui leur permettent d'effectuer certaines tâches de manière automatique et efficace. L'automatisation de la gestion des infrastructures permet de réduire considérablement le temps et les efforts consacrés à des tâches manuelles, tout en minimisant le risque d'erreurs. Ansible s'inscrit précisément dans cette démarche, en offrant un outil puissant pour automatiser ces tâches et optimiser l'efficacité des équipes réseau et sécurité.

Dans ce mémoire nous avons mise en place une solution d'automatisation d'infrastructure réseau à l'aide d'un outil très puissant qui est Ansible, en prenant comme cas d'études l'entreprise Tchén-Lait Candia de Bejaia. Notre solution permet la gestion des ressources et des services réseaux, de manière rapide et efficace, notamment en termes de gain de temps, de réduction des erreurs humaines et de simplification des tâches répétitives. Cette expérience a non seulement enrichi et approfondi nos notions théoriques et pratiques acquises durant notre cursus mais a aussi apporté une valeur pratique et concrète à l'entreprise Tchén-Lait Candia.

Cette étude nous a permis d'acquérir une certaine expérience pratique dans le domaine d'automatisation d'infrastructure réseau et système en utilisant l'outil Ansible. Ce dernier est un outil très puissant et flexible, capable de répondre aux exigences des environnements réseaux modernes et complexes. Enfin, nous espérons que les résultats et les méthodes présentés dans ce mémoire pourront servir de référence pour de futures initiatives d'automatisation.

A l'avenir, nous aspirons procéder à l'automatisation d'autres éléments de l'infrastructure réseau tels que les routeurs, et implémenter notre solution sur une infrastructure réseau réelle.

RESUME

Ce mémoire a été réalisé tout au long du semestre en vue d'obtenir le diplôme de Master en Administration et sécurité des réseaux informatiques à l'université de Bejaia. Dans ce cadre, nous avons effectué un stage au sein de l'entreprise Tchén-Lait CANDIA. L'objectif de ce stage était d'étudier l'architecture du réseau informatique de l'entreprise et de proposer une solution d'automatisation en utilisant l'outil Open Source Ansible dans un environnement virtuelle VMWARE PRO 17 et GNS3. Nous avons installé et configuré notre architecture réseau, En la sécurisant avec la segmentation de réseau LAN en VLANs et en configurant les protocoles VTP, TRUNK. Ensuite, Nous avons installé et configuré notre solution ANSIBLE, et finalement nous avons simulé la solution.

Mots clés: VMWARE PRO 17, GNS3, LAN, VLANs, VTP, TRUNK, ANSIBLE.

ABSTRACT

This thesis was carried out throughout the semester with the aim of obtaining a Master's degree in Administration and Security of Computer Networks at the University of Bejaia. As part of this, we completed an internship at the company Tchén-Lait CANDIA. The objective of this internship was to study the architecture of the company's computer network and to propose an automation solution using the Open Source tool Ansible in a VMWARE PRO 17 and GNS3 virtual environment. We installed and configured our network architecture, securing it with LAN network segmentation into VLANs and configuring the VTP and TRUNK protocols. Then, we installed and configured our ANSIBLE solution, and finally, we simulated the solution.

Keywords: VMWARE PRO 17, GNS3, LAN, VLANs, VTP, TRUNK, ANSIBLE.

Références

- [1] B. Mr Salim, «Etude et conception d'une plateforme de réseau informatique couplant entre sécurité et supervision pour l'entreprise ENIEM .,» 2013.
- [2] J. DORDOIGNE, «DORDOIGNE , José . Notions fondamentales (Protocoles, Architectures, Réseaux sans fil, Virtualisation, Sécurité ,IP v6...)».
- [3] M. L. L. N. Hans., cours sur Les Topologies Physiques des réseaux informatiques, ecole normale supérieur du cameroun., ecole normale supérieur du cameroun., 2016.
- [4] D.Zouatine., "outage multicast à travers un backbone maillé sans fil"., Oum El Bouaghi: Université Larbi Ben M'hidi, 2017..
- [5] <https://community.fs.com/fr/article/tcpip-vs-osi-whats-the-difference-between-the-two-models.html>.
- [6] https://www.cisco.com/c/fr_ca/solutions/automation/network-automation.html.
- [7] <https://www.fortinet.com/fr/resources/cyberglossary/network-automation>.
- [8] <https://betterstack.com/community/comparisons/chef-vs-puppet-vs-Ansible/>.
- [9] https://docs.Ansible.com/Ansible/latest/getting_started/introduction.html.
- [10] S. Asif, «Chef Vs Ansible Vs Puppet: Top DevOps Tools Compared.,» 5 sep 2023.
- [11] <https://devopssec.fr/article/introduction-cours-complet-Ansible>, «introduction-cours-complet-Ansible».
- [12] <https://www.redhat.com/fr/topics/automation/learning-ansible-tutorial>.
- [13] <https://medium.com/@snsavithrik1/an-introduction-to-Ansible-aa62559d97de>. [Accès le 13 05 2024].
- [14] A. IDI, Ansible LE GUIDE COMPLET DU DEBUTANT, ALPHORM.COM, 2021.
- [15] <https://geekflare.com/fr/ansible-basics/>. [Accès le 12 04 2024].

Références

- [16] <https://www.redhat.com/fr/topics/automation/what-is-yaml>. [Accès le 03 05 2024].
- [17] <https://www.redhat.com/fr/topics/virtualization>. [Accès le 02 03 2024].
- [18] <https://www.ibm.com/fr-fr/topics/virtualization>. [Accès le 1 06 2024].
- [19] <https://www.mbi85.fr/serveur-informatique-talmont-saint-hilaire.html>. [Accès le 15 04 2024].
- [20] <https://docs.vmware.com/>. [Accès le 03 06 2024].
- [21] «<https://www.redhat.com/fr/topics/automation/what-is-yaml>,» .

