

**Soutenue le 29/06/2024 , devant le jury :**

* **Réalisé à :** SII Services Maroc
* **Réalisé par :** Abbou Aymane

Mise en place d'un réseau operateurs IP MPLS capable de transporter les flux Data/Voix de façon automatiser

**MEMOIRE DE PROJET DE FIN D’ETUDES**

**Pour l’obtention du diplôme d’Ingénieur d’Etat en Génie réseau, système et service programmable**

****

Dédicace

Je dédie ce travail à toutes les familles qui ont joué un rôle essentiel dans ma vie et à qui je dois tout. Vos sacrifices ont été nombreux et j'espère que ce travail reflète ma profonde gratitude, mon amour et mon affection envers vous. Je suis éternellement reconnaissant pour votre patience et pour tout ce que vous avez fait pour moi. Ce moment que vous avez attendu depuis longtemps est enfin arrivé et je suis très heureux de vous le partager. Que Dieu vous accorde une bonne santé et une longue vie. Je tiens également à exprimer mon profond respect et mon amour à tous mes amis et mes proches.

Je souhaite à leurs familles une vie heureuse et prospère. Mes sentiments envers eux sont difficiles à exprimer par des mots, mais je les remercie infiniment pour leur soutien et leur affection

Remerciements

Avant tout, j’aimerai rendre grâce à Dieu tout puissant pour ses bienfaits trop souvent négliges.

J’adresse mes remerciements à M. AZIZ Tarik, directeur général de SII SERVICES MAROC, qui m’a permis d’effectuer ce stage de fin d’études au sein de l’organisme.

Je tiens aussi à adresser mes plus vifs remerciements à Mme MOUNIR ASSIA, notre Scrum Master qui m’a offert l’opportunité de mettre en pratique mes connaissances tout en évoluant dans un milieu professionnel stimulant.

Je tiens à exprimer toute ma gratitude à mes encadrants M. AYOUB AMINE ET M.HOUSSAM HICHAMI qui m’a accueilli dans son équipe et m’a permis de mener à bien ce travail. Nos discussions, toujours très fructueuses, ont beaucoup compté dans l’orientation de mes recherches et l’aboutissement de ces six mois d’´études.

Mes remerciements sincères vont également à tout le personnel de SII, pour leur accueil chaleureux, leur bonne humeur quotidienne, leur capacité d’équipe exemplaire ainsi que leur présence quand j’avais besoin de leur aide.

Je remercie vivement M. EL KABTANE HAMADA d’avoir accepté d’être le tuteur de mon PFE. Ses remarques et conseils, tous très constructifs, m’ont beaucoup aidé.

Que messieurs les membres de jury trouvent ici l’expression de ma reconnaissance pour avoir accepté de juger mon travail.

Bien évidemment, je remercie ma famille et mes amis qui tout au long de ce stage, mais aussi tout au long de mes études ont su me soutenir et m’accompagner dans tous ces moments de doute, mais aussi dans tous ces moments de joie. Merci à tous !

Résumé

Le projet « Mise en place d’un Réseau opérateurs IP/MPLS capable de Transporter les flux Data/Voix/Vidéo » consiste en un cadre théorique et une recherche empirique. La recherche a été initiée par la collecte et l'utilisation de données et de théories existantes sur le sujet.

Les solutions VPN MPLS d'un fournisseur de services ont été étudiées dans le cadre d'une approche pragmatique.

Le sujet de projet était d'étudier l'implémentation de solutions VPN MPLS À proximité du fournisseur de services. L'objectif principal de cette étude était de comparer le contenu des cours du fournisseur de services, qui traite principalement de la technologie VPN MPLS sur les deux équipements Alcatel Nokia et Cisco XRv.

Pour l'introduction de la technologie VPN MPLS dans le réseau du fournisseur de services utilisant les solutions VPN MPLS sur Nokia et Cisco, le cours offre un bon environnement d'apprentissage, y compris la théorie, les guides de configuration et un environnement virtuel pour la formation.

Ce cours pourrait utiliser les résultats de cette étude comme ressource supplémentaire pour que les étudiants puissent en savoir plus sur l'environnement Nokia. Un impact plus significatif peut être réalisé en fournissant un environnement virtuel, où les étudiants peuvent pratiquer, par ex. les exemples de configuration présentés dans cette étude. En outre, le résultat de l'étude est un guide simplifié et pratique des solutions Nokia MPLS VPN, principalement destiné aux professionnels des télécommunications juniors ou aux stagiaires débutant leur carrière dans l'environnement des fournisseurs de services.

Abstract

The actual need for secure data communication is enormous. past that Using a leased line is the most common way to connect two remote locations. today, Virtual private networks steal the show with their much lower costs.

However, Layer 2 VPNs are difficult and poorly scalable to implement. despite this, VPNs with MPLS have recently appeared on the market and offer higher performance No data encryption required.

The concept of MPLS VPN relies on separate contexts and routing tables on the border Routers for each VPN. Forward packets in the core network with additional labels Define VPN membership. Every customer doesn't know the MPLS cloud Address space separation. However, a high level of trust in the service provider is essential.

Given its growing popularity, ensure MPLS VPN The technology is truly secure and will never allow hacking into the core network or the core network VPN. Again, the above should provide isolation and security for the user Security, protecting information from interception or modification.

MP-BGP is a variant of the BGP protocol used to pass VPN routing information Detailed analysis. Our results show that MPLS VPN architectures are inherently secure There are no configuration errors. Finally, advice and recommendations It is recommended to avoid attack attempts.

*Keywords : MPLS, TE, Ansible, VPN*

Table des matières

[Dédicace 3](#_Toc169786334)

[Remerciements 4](#_Toc169786335)

[Résumé 5](#_Toc169786336)

[Abstract 6](#_Toc169786337)

[Liste des figures 13](#_Toc169786338)

[Définition des termes et abréviations 17](#_Toc169786339)

[Liste des tableaux 19](#_Toc169786340)

[Introduction générale 20](#_Toc169786341)

[Chapitre 1 : contexte général du projet 21](#_Toc169786342)

[1 Introduction : 22](#_Toc169786343)

[2 Organisme d’accueil 22](#_Toc169786344)

[2.1 Le groupe SII : 22](#_Toc169786345)

[3 SII Services Maroc : 23](#_Toc169786346)

[3.1 Secteurs d’activités : 23](#_Toc169786347)

[3.2 Clients SII : 24](#_Toc169786348)

[4 Architecture du réseau Bouygues Telecom : 24](#_Toc169786349)

[5 Contexte générale de projet : 25](#_Toc169786350)

[5.1 Contexte générale 25](#_Toc169786351)

[5.2 Problématique 25](#_Toc169786352)

[5.3 Objectifs 26](#_Toc169786353)

[5.4 Diagramme de GANTT : 26](#_Toc169786354)

[6 Conclusion : 27](#_Toc169786355)

[Chapitre 2 : L’état de l’art 28](#_Toc169786356)

[1 Introduction : 29](#_Toc169786357)

[2 Le réseau d’opérateur Backbone : 29](#_Toc169786358)

[3 Généralité sur MPLS au sein du Backbone de SII 31](#_Toc169786359)

[3.1 Fonctionnement de MPLS 32](#_Toc169786360)

[3.1.1 Le Data Plane 32](#_Toc169786361)

[3.1.2 Le Contrôle Plane 32](#_Toc169786362)

[3.1.3 Commutation Labels 33](#_Toc169786363)

[3.1.4 Implicite Routing (LDP) 34](#_Toc169786364)

[3.1.5 Explicit Routing 35](#_Toc169786365)

[3.2 Les Application de MPLS 36](#_Toc169786366)

[4 Ingénierie de trafic avec MPLS (MPLS-TE) 36](#_Toc169786367)

[4.1 Le type de service (QoS) 36](#_Toc169786368)

[4.2 L’agrégation de flux 37](#_Toc169786369)

[5 Protocoles de routage 37](#_Toc169786370)

[5.1 Protocole de routage IGP 37](#_Toc169786371)

[5.1.1 Le protocole ISIS 37](#_Toc169786372)

[5.1.2 Le protocole OSPF 38](#_Toc169786373)

[5.1.3 Comparaison entre IS-IS et OSPF 39](#_Toc169786374)

[5.2 Protocole BGP 40](#_Toc169786375)

[5.2.1 Généralité 40](#_Toc169786376)

[5.2.2 Autonomes System 41](#_Toc169786377)

[5.2.3 BGP Split Horizon 41](#_Toc169786378)

[5.2.4 BGP peering full mesh 42](#_Toc169786379)

[5.2.5 Router réflecteur principe 42](#_Toc169786380)

[6 Les services MPLS (Les VPNs basés sur MPLS) 43](#_Toc169786381)

[6.1 VPN VPRN L3 43](#_Toc169786382)

[6.2 Epipe VPN L2 44](#_Toc169786383)

[6.3 VPLS VPN L2 45](#_Toc169786384)

[7 Les VRFs (Virtual Routing and Forwarding) 46](#_Toc169786385)

[8 Qualité de service 46](#_Toc169786386)

[8.1 Classification des applications et besoin de QoS 46](#_Toc169786387)

[8.2 Paramètres-clé de la QoS 47](#_Toc169786388)

[8.3 Techniques de traitement de la QoS 47](#_Toc169786389)

[8.4 Les Modèles de la QoS 48](#_Toc169786390)

[8.4.1 IntServ 49](#_Toc169786391)

[8.4.2 DiffServ 49](#_Toc169786392)

[9 L'automatisation via Ansible 51](#_Toc169786393)

[9.1 L’architecture d’Ansible 51](#_Toc169786394)

[9.2 Le fonctionnement en général 52](#_Toc169786395)

[9.3 Les composantes d’Ansible 52](#_Toc169786396)

[9.3.1 Inventaire (Inventory) 52](#_Toc169786397)

[9.3.2 Les Groupes 53](#_Toc169786398)

[9.3.3 Variables 53](#_Toc169786399)

[9.3.4 Playbook 54](#_Toc169786400)

[9.3.5 Les tache (Tasks) 55](#_Toc169786401)

[9.3.6 Rôle 55](#_Toc169786402)

[9.3.7 Les modules 55](#_Toc169786403)

[9.4 Les structures de contrôle 55](#_Toc169786404)

[9.4.1 Facts 55](#_Toc169786405)

[9.4.2 Les conditions (When) 55](#_Toc169786406)

[9.4.3 Les filtres Jinja 56](#_Toc169786407)

[9.5 YAML 56](#_Toc169786408)

[Chapitre 3 : Étude de l’existant 57](#_Toc169786409)

[1 Introduction 58](#_Toc169786410)

[2 Architecture existante 58](#_Toc169786411)

[3 Critique de l’architecture 59](#_Toc169786412)

[4 L'étude des solutions possible pour notre besoin : Chef, Puppet, Terraform, NSO . 61](#_Toc169786413)

[4.1 Terraform 61](#_Toc169786414)

[4.2 Puppet 62](#_Toc169786415)

[4.3 Chef 63](#_Toc169786416)

[4.4 NSO 64](#_Toc169786417)

[5 La solution 64](#_Toc169786418)

[5.1 Ansible est plus cohérent 64](#_Toc169786419)

[5.1.1 Introduction générale 65](#_Toc169786420)

[5.1.2 Ansible est un bon outil pour l’automatisation 66](#_Toc169786421)

[Chapitre 4 : La Réalisation 68](#_Toc169786422)

[1 Introduction 69](#_Toc169786423)

[2 Logiciel et les équipements utilisé 69](#_Toc169786424)

[2.1 Eve- NG 69](#_Toc169786425)

[2.2 VMWare Workstation 69](#_Toc169786426)

[2.3 Ansible 70](#_Toc169786427)

[2.4 Visual studio Code 70](#_Toc169786428)

[2.5 SecureCRT 70](#_Toc169786429)

[3 Les équipements utilisés 71](#_Toc169786430)

[3.1 NOKIA VSR 7750 71](#_Toc169786431)

[3.2 CISCO XRv 71](#_Toc169786432)

[4 Mise en œuvre de la topologie réseau (la maquette du backbone) 72](#_Toc169786433)

[4.1 Création de la maquette 72](#_Toc169786434)

[4.2 Adressage de chaque routeur 72](#_Toc169786435)

[5 Méthodologie d’approche 77](#_Toc169786436)

[5.1 Configuration des interfaces 77](#_Toc169786437)

[5.2 Réseau de transport 78](#_Toc169786438)

[5.2.1 Mise en place du protocole OSPF 78](#_Toc169786439)

[5.2.2 Activation du protocole MPLS et LDP et RSVP sur les routeurs du cœur de réseau : 80](#_Toc169786440)

[5.2.3 Mise en place du protocole MP-BGP à base de RR 82](#_Toc169786441)

[5.3 Réseau mobile 84](#_Toc169786442)

[5.3.1 Mise en place du protocole CE-PE 84](#_Toc169786443)

[5.3.2 Configuration de INTERNAL BGP entre les PTNs 85](#_Toc169786444)

[5.3.3 Gestion de la redistribution respective des préfixes sur le routeur PTN1-REN 85](#_Toc169786445)

[5.4 Réseaux entreprise 86](#_Toc169786446)

[5.4.1 Mise en place du protocole CE-PE 86](#_Toc169786447)

[5.4.2 Configuration de IBGP entre les ENT de Réseaux entreprise 86](#_Toc169786448)

[5.5 Configuration des services : VPRN – EPIPE – VPLS (9) 87](#_Toc169786449)

[5.5.1 Configurer les interfaces entre PE et CE 87](#_Toc169786450)

[5.5.2 Service VPRN 87](#_Toc169786451)

[5.5.3 Service VPLS 89](#_Toc169786452)

[5.5.4 Service EPIPE (VPWS) 90](#_Toc169786453)

[6 Automatisation de configuration et le test 91](#_Toc169786454)

[6.1 Configuration du fichier d’inventaire 91](#_Toc169786455)

[6.2 Test de connexion 91](#_Toc169786456)

[6.3 Connectivite de serveur ansible avec le LAB via SSH 92](#_Toc169786457)

[6.4 Création des rôles 93](#_Toc169786458)

[6.5 Création des taches 93](#_Toc169786459)

[6.6 Extraction des données 94](#_Toc169786460)

[6.7 Création de PLAYBOOK : 95](#_Toc169786461)

[6.8 Automatisation de test 97](#_Toc169786462)

Liste des figures

[Figure 1: Le Logo du groupe SII 18](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777768)

[Figure 2 : SII dans le monde 19](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777769)

[Figure 3:Les secteurs d’activités du groupe SII 19](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777770)

[Figure 4 : Les clients de SII 20](#_Toc169777771)

[Figure 5: Architecture des socles d’hébergement Bouygues Telecom 20](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777772)

[Figure 6 : diagramme de gantt 22](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777773)

[Figure 7: Architecture Backbonière 25](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777774)

[Figure 8: Architecture des opérateurs des différents niveaux 26](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777775)

[Figure 9: L’architecture générale d’un réseau MPLS 27](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777776)

[Figure 10 : Le modèle de fonctionnement de MPLS 28](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777777)

[Figure 11 : Une architecture simplifiée des équipements de MPLS 29](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777778)

[Figure 12 : La structure de l’entête MPLS 30](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777779)

[Figure 13 : Le fonctionnement de protocole LDP 31](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777780)

[Figure 14 : Le fonctionnement de LSP 31](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777781)

[Figure 15 : Architecture générale en IS- IS 35](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777782)

[Figure 16 : les différentes zones en OSPF 36](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777783)

[Figure 17 : Architecture de protocole BGP 37](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777784)

[Figure 18 : Le rôle de la notion AS 38](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777785)

[Figure 19 : Fonctionnement de Split Horizon 38](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777786)

[Figure 20 : Architecture full mesh 39](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777787)

[Figure 21 : Fonctionnement de Routeur Réflecteur 39](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777788)

[Figure 22 : Exemple de VPN L3 40](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777789)

[Figure 23 : Le fonctionnement de VPN L3 VPRN 41](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777790)

[Figure 24 : Exemple de VPN L2 41](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777791)

[Figure 25: Fonctionnement de service VPN L2 VPLS 42](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777792)

[Figure 26 : Le fonctionnement de VRF 43](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777793)

[Figure 27: Les exemples de classes de service 44](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777794)

[Figure 28 : Entête QoS 44](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777795)

[Figure 29: Le fonctionnement de IntServ 46](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777796)

[Figure 30: Le fonctionnement de DiffServ 47](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777797)

[Figure 31: Comparaison d’entête IP et l’entête MPLS 47](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777798)

[Figure 32: L’entête DSCP 48](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777799)

[Figure 33: L‘architecture de fonctionnement Ansible 49](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777800)

[Figure 34: exemple de fichier inventaire 50](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777801)

[Figure 35:Exemple d’un playbook 51](#_Toc169777802)

[Figure 36 : Un exemple de condition When 52](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777803)

[Figure 37: Un filtre en langage Jinja 53](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777804)

[Figure 38: Une Playback écrit en YAML 53](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777805)

[Figure 39 : L’architecture de la solution existante 55](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777806)

[Figure 40: un exemple d’une partie des configurations des interface pour un équipement 57](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777807)

[Figure 41: L’architecture de fonctionnement de Terraform 58](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777808)

[Figure 42: L’architecture de fonctionnement de Puppet 59](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777809)

[Figure 43 : L’architecture de fonctionnement de Chef 60](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777810)

[Figure 44 : Architecture NSO 61](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777811)

[Figure 45: Les avantages de l’automatisation des réseaux 63](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777812)

[Figure 46:EVE-NG 66](#_Toc169777813)

[Figure 47: VM WARE 67](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777814)

[Figure 48: la solution Ansible 67](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777815)

[Figure 49:logiciel Vs code 67](#_Toc169777816)

[Figure 50 : secureCRT 68](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777817)

[Figure 51: L’équipement NOKIA VSR 7750 68](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777818)

[Figure 52: L’équipement CISCO XRv 68](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777819)

[Figure 53 : Architecture du projet 69](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777820)

[Figure 54 : configuration des interfaces sur Cisco 74](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777821)

[Figure 55: configuration des interfaces sur Nokia 74](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777822)

[Figure 56 : test des interfaces sur Cisco 75](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777823)

[Figure 57 : test des interfaces sur Nokia 75](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777824)

[Figure 58 : configuration de OSPF sur Cisco 75](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777825)

[Figure 59 : configuration de OSPF sur Nokia 76](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777826)

[Figure 60 : test OSPF sur Cisco 76](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777827)

[Figure 61 : test OSPF sur Nokia 76](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777828)

[Figure 62 : configuration RSVP sur Cisco 77](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777829)

[Figure 63 : configuration LDP sur Cisco 77](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777830)

[Figure 64 : configuration MPLS sur Nokia 77](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777831)

[Figure 65 : configuration LDP sur Nokia 78](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777832)

[Figure 66: configuration RSVP sur Nokia 78](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777833)

[Figure 67: test des interfaces de RSVP sur cisco 79](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777834)

[Figure 68 : test mpls par ping sur CISCO 79](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777835)

[Figure 69 : configuration de IBGP ENTRE PE ET RR 79](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777836)

[Figure 70: test de BGP SUMMARY entre PE ET RR 80](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777837)

[Figure 71 : test de connectivité entre le RR ET les PE 80](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777838)

[Figure 72 : configuration ISIS sur le site MOBILE 81](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777839)

[Figure 73 : test de protocole ISIS 81](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777840)

[Figure 74 : configuration de IBGP sur le site mobile 82](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777841)

[Figure 75 : configuration de POLICY option sure la zone mobile 82](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777842)

[Figure 76 : configuration de OSPF sur le site entreprise 83](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777843)

[Figure 77 : configuration de IBGP sur le site entreprise 83](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777844)

[Figure 78 : configuration des interfaces entre le PE et le CE 84](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777845)

[Figure 79 : création des clients 84](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777846)

[Figure 80 : configuration de service VPRN sur le site mobile 84](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777847)

[Figure 81 : création de sous interface virtuelle 85](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777848)

[Figure 82 : configuration de VRF sur le site entreprise 85](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777849)

[Figure 83 : le test de service VPRN 85](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777850)

[Figure 84 : configuration de SDP 86](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777851)

[Figure 85 : configuration de service VPLS 86](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777852)

[Figure 86 : test par ping de service VPLS 87](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777853)

[Figure 87 : configuration de service EPIPE 87](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777854)

[Figure 88 : test par ping de service EPIPE 87](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777855)

[Figure 89 : déclaration d'adresse IP d'équipement sur le fichier hosts 88](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777856)

[Figure 90 : test de connectivité entre ansible et le routeur 88](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777857)

[Figure 91: configuration ssh sur le routeur 89](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777858)

[Figure 92 : test de connectivité par SSH 89](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777859)

[Figure 93 : création des rôles 90](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777860)

[Figure 94 : contenu de fichier roles/deploy-interface /task 91](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777861)

[Figure 95 : contenu de fichier template jinja2 91](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777862)

[Figure 96 : contenu de fichier DCE 92](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777863)

[Figure 97 : library xls\_TO\_FACT 93](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777864)

[Figure 98 : création d'un PLAYBOOK 93](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777865)

[Figure 99 : lancement de PLAYBOOK 94](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777866)

[Figure 100 : contenu de fichier tasts d'un rôles check-conf 94](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777867)

[Figure 101 : contenu de dossier files 94](file:///C:\Users\hp\Downloads\RAPPORT%20VER%2002.docx#_Toc169777868)

Définition des termes et abréviations

|  |  |
| --- | --- |
| ABR | Area Border Router |
| API | Application Programming Interface |
| AS | Autonomous System |
| ASBR | Autonomous System Border Router |
| BGP | Border Gateway Protocol |
| CE | Customer Edge |
| CLI | Command Line Interface |
| COS | Class of Service |
| CR-LDP | Constraint-Based Routing with Label Distribution Protocol |
| DSCP | Differentiated Services Code Point |
| E-BGP | External Border Gateway Protocol |
| EGP | Exterior Gateway Protocol |
| E-LSR | Edge-Label Switching Router |
| EOS | End of Support |
| Epipe | Ethernet Pipe |
| ER-LSP | Explicit Route Label Switched Path |
| FAI | First Article Inspection |
| FEC | Forwarding Equivalence Class |
| IGP | Interior Gateway Protocol |
| IP | Internet Protocol |
| IS-IS | Intermediate System to Intermediate System |
| IXP | Internet Exchange Point |
| L2 | Layer 2 |
| L3 | Layer 3 |
| LAN | Local Area Network |
| LDP | Label Distribution Protocol |
| LFIB | Label Forwarding Information Base |
| LSR | Label Switching Router |
| LSU | Link State Update |
| MPLS | Multiprotocol Label Switching |
| OSI | Open Systems Interconnection |
| OSPF | Open Shortest Path First |
| P | Provider |
| PE | Provider Edge |
| PHP | Penultimate Hop Popping |
| PWP | seudo-Wire |
| QOS | Quality of Service |
| RIP | Routing Information Protocol |
| RSVP | Resource Reservation Protocol |
| SDP | Session Description Protocol |
| SLA | Service Level Agreement |
| TCP | Transmission Control Protocol |
| TE | Traffic Engineering |
| TOS | Type of Service |
| UDP | User Datagram Protocol |
| VLAN | Virtual Local Area Network |
| VPLS | Virtual Private LAN Service |
| VPRN | Virtual Private Routed Network |
| VRF | Virtual Routing and Forwarding |
| WAN | Wide Area Network |
| YAML | Yet Another Markup Language |
| YANG | Yet Another Next Generation |

Liste des tableaux

[Tableau 1 : Les signification des priorités 40](#_Toc169185201)

[Tableau 2: Le plan d'adressage 66](#_Toc169185202)

Introduction générale

L'automatisation des réseaux est devenue une préoccupation majeure pour de nombreuses organisations en raison de l'augmentation exponentielle du nombre d'équipements réseau nécessaires pour répondre aux besoins de l'entreprise. Ces équipements peuvent inclure des routeurs, des commutateurs, des pares-feux, des points d'accès sans fil, des dispositifs IoT et bien plus encore, ils sont souvent très critiques pour le fonctionnement de l'entreprise, car ils permettent de connecter les employés, les clients et les fournisseurs, ainsi que de garantir la sécurité et la disponibilité des données et les divers services inclus dans l’architecture. Toutefois, la configuration manuelle de ces équipements, qui doit être effectuée équipement par équipement, peut s'avérer fastidieuse et chronophage, surtout si le réseau est important.

En outre, l’utilité de l’automatisation qui joue un rôle poly-valant pour les différents domaines et utilités, et nous allons voir dans ce projet que l’automatisation vient avec d’autre bénéfices tel que le temps, le taux d’erreur, et la sécurité.

De plus, le nombre des services qu’on peut configurer à la fois ne pose aucun problème ainsi que la possibilité de modifier la configuration en termes de temps, des services déployés, et les équipements que nous veillons les configurés, et nous ne pouvons pas négliger le nombre d'équipements réseau nécessaires pour prendre en charge les besoins de l'entreprise augmente rapidement.

Des solutions diverses existent pour répondre aux enjeux de l'automatisation réseau, mais leur mise en place doit être soigneusement réfléchie. Au-delà d'outils, c'est toute l'architecture qui est concernée. Un outil qui devra générer des configurations spécifiques, équipement par équipement, ne sera pas efficace. Il est possible de constater après deux ans d'efforts et de personnalisation de Playbook Ansible, que le résultat final ressemble finalement à une solution clé en main proposée par un éditeur de logiciels. Cette situation peut être attribuée à un manque d'étude préalable pour identifier les besoins spécifiques de l'entreprise et les options disponibles sur le marché.

Dans ce contexte, l'automatisation des réseaux est devenue une solution essentielle pour les organisations qui souhaitent gérer efficacement leur infrastructure réseau. L'automatisation permet de configurer les équipements de manière centralisée, de manière à simplifier et à accélérer les processus de gestion de réseau, tout en garantissant une cohérence et une précision accrues. Elle peut également contribuer à améliorer la sécurité et la résilience du réseau, ainsi qu'à réduire les coûts et les risques associés à la gestion manuelle des équipements

Chapitre 1 : contexte général du projet

# Introduction :

Ce chapitre vise à fournir une vue d'ensemble du projet en mettant en avant l'organisme d'accueil, à savoir "SII Services Maroc". Il présente également le contexte général du projet ainsi que la problématique qui a été identifiée. Enfin, il expose la méthodologie adoptée pour la gestion du projet et présente le schéma de planification qui a été élaboré.

Il est important de souligner que ce chapitre permet de situer le projet dans son environnement global et de comprendre les différentes étapes qui seront suivies pour sa réalisation. Il offre une vue d'ensemble des objectifs, des parties prenantes impliquées et des contraintes éventuelles. De plus, il fournit des indications sur la manière dont le projet sera géré, notamment en termes de méthodologie et de planification.

Ce chapitre constitue une introduction essentielle au projet, en fournissant aux lecteurs les informations nécessaires pour comprendre le contexte et l'approche qui seront adoptés tout au long du rapport.

# Organisme d’accueil

Dans la première partie de ce chapitre, nous donnerons quelques informations sur le groupe SII et nous décrivons ses secteurs d’activités et ses clients. Ainsi qu’une vue organisationnelle de SII Services Maroc (filiale du groupe SII) et une description des projets.

## Le groupe SII :

Fort de plus de 40 ans d’expertise dans le conseil en technologie et l’intégration de systèmes, le groupe SII possède l’ensemble des compétences nécessaires à l’accompagnement de ses clients dans leurs besoins d´évolution technologique. SII offrent des différents types d’intervention pour couvrir l’intégralité du cycle d’un projet :

* Le conseil et les études en amont.
* L’ingénierie de d´enveloppement.
* Les tests, le déploiement et la maintenance.

**Une image contenant Graphique, logo, Police, symbole

Description générée automatiquement**

Figure 3: Le Logo du groupe SII

Une image contenant carte, texte, atlas

Description générée automatiquement

Figure 4 : SII dans le monde

# SII Services Maroc :

SII Maroc est créé en 2008 comme base d’offshore à Casablanca, dirigé par Monsieur AZIZI TARIK et en 2014 SII Services Maroc adresse ´également le marché local. Les consultants de SII Maroc interviennent sur des projets de développement logiciel, des projets de sécurité numérique et des projets pour le secteur bancaire et assurance. En 2008, avec plus de 70 collaborateurs, SII Service Maroc a réalisé un CA de 2 millions d’euros.

## Secteurs d’activités :

SII réalise son chiffre d’affaires sur des segments de marché très diversifiés. La diversité des projets et des clients adressés par SII confère au groupe un positionnement d’acteur majeur qui nécessite la maîtrise d’une large gamme de compétences technologiques sur les m´étiers de l’ingénieur.

Une image contenant texte, Police, capture d’écran, conception

Description générée automatiquement

Figure 5:Les secteurs d’activités du groupe SII

## Clients SII :

Le nombre des clients de SII dépasse 72 clients qui ont des domaines d’activité différents, parmi eux ceux qui sont représentés dans la figure suivante :

Une image contenant texte, Police, diagramme, capture d’écran

Description générée automatiquement

Figure 6 : Les clients de SII

# Architecture du réseau Bouygues Telecom :

Bouygues Telecom possède une architecture organisée en plusieurs Backbone et zones d’hébergement appelées socles dont les équipements sont repartis sur plusieurs sites. Un socle est défini soit par le type de flux transitant, le service fournit, ou l’infrastructure dédiée. Ces socles peuvent tous communiquer entre eux grâce aux raccordements qu’ils ont avec le Backbone principal appelé ORION. Tous les flux provenant des différentes zones d’hébergement du réseau se retrouvent plus particulièrement dans une VRF (Virtual Routing and Forwarding) de ORION appelée FUSION.

Une image contenant texte, diagramme, cercle, Police

Description générée automatiquement

Figure 7: Architecture des socles d’hébergement Bouygues Telecom

Le schéma présente l’ensemble des socles du réseau Bouygues Télécom. Le raccordement au Backbone principal dépend des socles, certains sont connectés aux routeurs PE via une CE sans filtrage, d’autres qui sont protégés par des firewalls. Il existe aussi des socles comme la zone de transit, qui sont protégés par un firewall mais qui sont directement raccordés au Backbone sans passer par un CE. Ces choix d’architecture dépendent des mesures de sécurité appliquées aux différents types de flux.

# Contexte générale de projet :

## Contexte générale

Dans un monde où les technologies de l'information et de la communication évoluent rapidement, l'automatisation des opérations est essentielle pour une gestion efficace des environnements réseau. Les opérateurs réseau doivent gérer et maintenir des infrastructures sophistiquées, nécessitant des configurations précises et des mises à jour fréquentes. Ces opérations manuelles peuvent être fastidieuses, sujettes aux erreurs humaines et consommatrices de temps.

C'est dans ce contexte que notre projet prend forme. Nous visons à explorer l'automatisation des opérations sur un environnement réseau avec une architecture MPLS/IP d'opérateur en utilisant les outils Ansible et Python. Notre objectif principal est de simplifier et d'accélérer l’implémentation et la configuration des services dans une topologie Backbone MPLS/IP. Ansible, un outil populaire de gestion de la configuration et d'automatisation, combiné à Python, un langage de programmation polyvalent, offre un ensemble puissant pour automatiser les opérations réseau. En alliant la simplicité et la flexibilité d'Ansible à la puissance et la polyvalence de Python, nous cherchons à créer une solution efficace et évolutive pour gérer les opérations réseau.

## Problématique

Les réseaux traditionnels des opérateurs reposent principalement sur la commutation de circuits, avec leurs avantages et inconvénients. Cependant, l'émergence de nouvelles technologies et l'explosion de l'offre de services ont posé de nombreux défis au secteur des Télécommunications et des Médias. Ces défis incluent la nécessité d'optimiser le réseau de transport pour acheminer efficacement les flux sans congestion, offrir une qualité de service supérieure, assurer une évolutivité et intégrer continuellement de nouveaux services.

Dans le domaine des réseaux informatiques, la configuration des infrastructures réseau est essentielle mais souvent laborieuse. Les architectures Backbone MPLS/IP sont particulièrement complexes, nécessitant des configurations précises et des ajustements fréquents pour garantir un fonctionnement optimal. Ces besoins requièrent de nombreuses configurations manuelles et un temps considérable pour maintenir les protocoles et services existants sans perturbation. Cela soulève la question de savoir comment visualiser un tel réseau, mettre en œuvre des changements et des améliorations, et les adapter à des besoins spécifiques, ainsi que l'intérêt de concevoir un réseau aussi complexe.

Pour atteindre ces objectifs, l'automatisation des opérations réseau devient un outil crucial pour permettre aux opérateurs de répondre rapidement, de manière fiable et sécurisée, aux besoins de leurs clients. Elle permet également aux opérateurs de rester compétitifs en offrant des services de qualité supérieure tout en réduisant les coûts d'exploitation.

Grâce à l'automatisation, les opérateurs peuvent également réduire les risques d'erreurs humaines, améliorer la sécurité et la fiabilité du réseau, et permettre une plus grande évolutivité pour répondre à une demande en constante évolution.

## Objectifs

En utilisant notre approche, nous visons à améliorer l'efficacité opérationnelle de l'entreprise, réduire les erreurs humaines et permettre une évolutivité optimale de l'infrastructure réseau. Cette étude a pour objectif d'explorer les avantages et les défis des configurations manuelles afin de proposer une solution d'automatisation des opérations sur un environnement réseau d'opérateur à l'aide d'Ansible et de Python.

En mettant l'accent sur la simplicité, la fiabilité et la scalabilité, nous espérons que notre projet apportera des contributions significatives à la gestion des réseaux informatiques dans le domaine des télécommunications. Parmi les objectifs d'amélioration et d’optimisation de la configuration des services d’une architecture Backbone MPLS/IP, on trouve :

* Proposition d’une architecture Backbone MPLS/IP.
* Configuration manuelle de SSH et de l’interface manager.
* Description des divers services VPN et des protocoles utilisés dans une architecture Backbone.
* Automatisation de la technologie Backbone MPLS/IP, ainsi que des services VPRN, Epipe et VPLS.
* Collecte des données concernant les performances des interfaces et la QoS.

## Diagramme de GANTT :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquementCe diagramme de Gantt illustre le planning d'un projet qui s'étend de mars à juin. Voici la description détaillée des différentes phases et tâches prévues :

Figure 8 : diagramme de gantt

Chaque tâche est planifiée sur une période spécifique avec une continuité logique et séquentielle pour assurer que toutes les étapes nécessaires sont couvertes, de l'intégration initiale à la présentation finale.

# Conclusion :

Ce chapitre introductif a pour but de se familiariser avec le contexte global du projet, en présentant l’organisme d’accueil, d´entaillant les problématiques et les enjeux soulevées par la situation, et mettant évidence la méthodologie de travail suivie et les étapes franchies afin de mener `à bien un projet d’une telle envergure.

Chapitre 2 : L’état de l’art

# Introduction :

Au cours des dernières années, les réseaux d'accès ont connu une popularité croissante, contribuant ainsi à la convergence des services offerts aux utilisateurs. Cette convergence permet aux utilisateurs de bénéficier en permanence d'un accès à une multitude de services, indépendamment de leur emplacement géographique ou des appareils qu'ils utilisent. Dans ce contexte, le futur réseau sera possibilité d’interconnecter des réseaux des divers sites des clients et assurer une communication privée, offrant un large éventail de services à travers le cœur de réseau IP/MPLS. Dernièrement le monde de réseau reconnaît un voyage à l’automatisation de ces services, cette tendance joue un rôle essentiel dans la gestion des réseaux, en simplifiant et en accélérant les opérations grâce à l'automatisation des tâches répétitives.

Dans ce chapitre, nous explorerons en détail les technologies MPLS et leurs services associés. Nous examinerons comment ces technologies permettent la convergence des services et l'interconnexion des réseaux d'accès hétérogènes. Nous mettrons également en évidence l'importance de l'automatisation des services réseau dans la gestion efficace de ces réseaux, en présentant les différentes tâches automatisables et les avantages qu'elles apportent. Nous discuterons des cas d'utilisation spécifiques où l'automatisation des services réseau est particulièrement pertinente et bénéfique.

# Le réseau d’opérateur Backbone :

Le réseau d’une façon générale est composé de liaisons d’accès qui acheminent le trafic vers des routeurs qu’ont une large bande passante et accompagnent le trafic de la source vers sa destination en empruntant le meilleur chemin disponible c’est exactement pour le réseau internet, alors l’interconnexion de ces réseaux individuels de fibres optiques à haut débit qui créent la dorsale ou bien une Backbone internet.

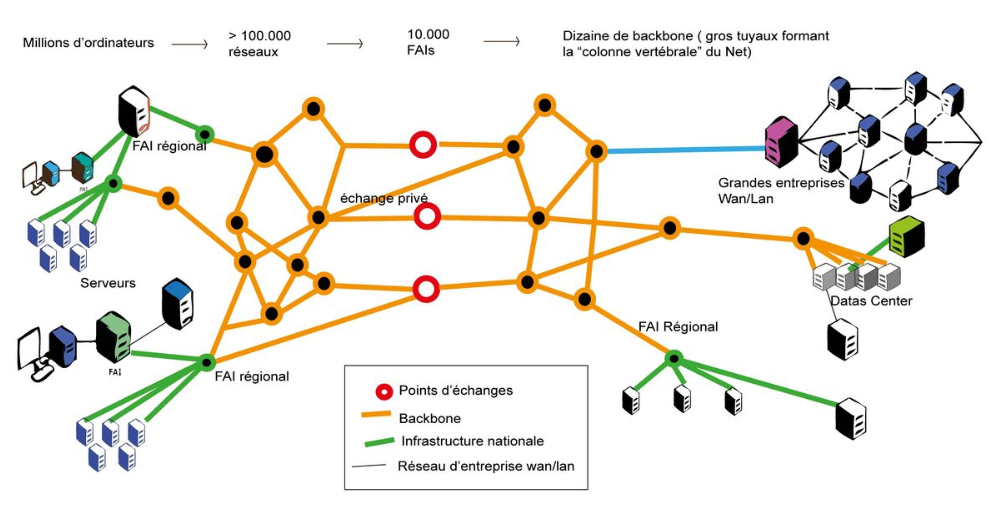
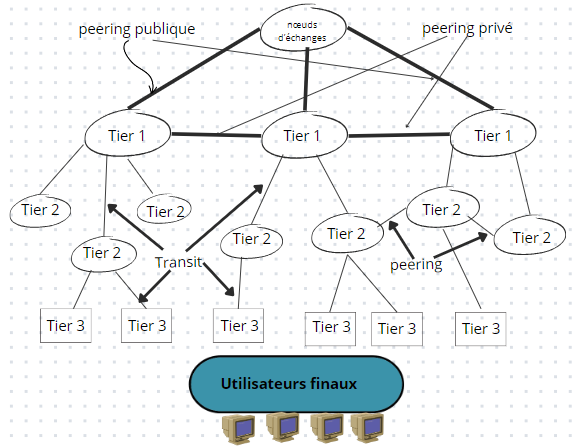


Figure 9: Architecture Backbonière

Chaque réseau appartient à des opérateurs Internet de niveau 1 (Tier 1), ce sont généralement de très gros opérateurs privés dont les réseaux sont reliés entre eux. Le regroupement de ces réseaux longue distance de ces opérateurs internet niveau 1 créent un réseau mondial unique qui permet à chacun d’entre eux d’accéder à l’ensemble de la table de routage internet pour qu’ils puissent acheminer efficacement le trafic vers sa destination en passant ensuite par une succession de fournisseurs locaux, de niveau 2 et 1, qui sont de plus en plus nombreux.

Figure 10: Architecture des opérateurs des différents niveaux



Dans le but de tout le monde doit être connecter physiquement, ces opérateurs de Backbone utilisent tous les protocoles de réseau TCP/IP, qu’est généralement se composé par deux protocoles : le protocole de contrôle de transmission ou bien TCP (Transmission Control Protocol) qu’est responsable de l'établissement et de la gestion des connexions réseau, et le protocole internet ou IP, qu’est un protocole de communication utilisé pour acheminer des paquets de données sur les réseaux informatiques tels qu'Internet et il fait partie de la couche réseau du modèle de référence OSI

* **Le fonctionnement d’un Backbone :**

Les dorsales Internet jouent un rôle crucial dans l'interconnexion des réseaux locaux (LAN) et des réseaux étendus (WAN). Les réseaux locaux sont des réseaux privés utilisés par les entreprises et les organisations pour connecter leurs équipements informatiques locaux, tels que des ordinateurs, des serveurs et des périphériques réseau. Les réseaux étendus, quant à eux, sont utilisés pour connecter des réseaux locaux entre eux, souvent sur de longues distances. Les dorsales Internet permettent aux différents réseaux, qu'ils soient locaux ou étendus, de communiquer entre eux en utilisant différents protocoles, tels que TCP/IP, BGP, OSPF, etc. Les réseaux locaux peuvent se connecter aux dorsales Internet via des liaisons de fibre optique ou des connexions haut débit, tandis que les réseaux étendus peuvent se connecter à travers des connexions de type MPLS ou VPN. Pour assurer la communication entre les différents réseaux, les dorsales Internet utilisent également des interconnexions d'égal à égal ou peering. Ce processus d'appairage permet aux réseaux de s'échanger mutuellement du trafic sans passer par des réseaux tiers, ce qui réduit les coûts de transit et améliore les performances.

# Généralité sur MPLS au sein du Backbone de SII

MPLS ou bien Multiple Protocol Label Switching est une technologie de réseau qui permet de transporter des paquets de données à travers un réseau en utilisant des étiquettes plutôt que des adresse IP pour le but de fournir un ensemble de services. Cette technologie est supportée par les divers équipement Cisco, Nokia etc. Ensuite, la technologie MPLS donne la possibilité de jouer sur plusieurs paramètres pour acheminer un trafic d’un client à travers un réseau MPLS qui dépend à nos besoins, comme les qualités de services et le TE.

Figure 11: L’architecture générale d’un réseau MPLS

Une image contenant texte, diagramme, carte

Description générée automatiquement

Traffic Engineering permettant de définir des chemins de routage explicites dans les réseaux IP (avec RSVP ou CR-LDP). L’ingénierie des flux est la faculté de pouvoir gérer les flux de données transportés au-dessus d’une infrastructure réseau. Aujourd’hui, cette ingénierie des flux est essentiellement faite à l’aide d’ATM, avec comme conséquence une grande complexité de gestion (en effet IP et ATM sont deux techniques réseaux totalement différents, avec parfois des contraintes non compatibles). Avec l’intégration de cette fonctionnalité, MPLS va permettre une simplification radicale des réseaux.

En plus, Les labels peuvent être attribués à différents éléments tels que le chemin, la destination, la source, l'application, le critère de qualité de service, ou une combinaison de ces éléments, offrant ainsi une grande flexibilité pour le routage IP.

En générale le MPLS permet d’améliorer les performances et baisser le coût d’une façon que ne traite pas le paquet IP (couche 3 de modèle OSI) dans chaque routeur, MPLS basé sur le principe de label qui peut avoir plusieurs sens, ainsi que le MPLS prendre en compte la QoS.

## Fonctionnement de MPLS

Pour comprendre le bon fonctionnement de MPLS il faut savoir que nous devons comprendre deux points essentiels, le Data Plane et Contrôle Plane, c’est deux éléments sans en relation entre eux s’il un élément n’existe pas nous n’aurons pas la technologie de MPLS dans notre infrastructure.

Figure 12 : Le modèle de fonctionnement de MPLS

Une image contenant texte, capture d’écran, Police, Rectangle

Description générée automatiquement

### Le Data Plane

Data Plan également appelé Forwarding Plane, est la composante de MPLS chargée de l'acheminement effectif des paquets de données à travers le réseau. Il est constitué des équipements de réseau tels que les routeurs MPLS, qui transfèrent les paquets en se basant sur les étiquettes MPLS associées à ces paquets. Lorsqu'un paquet arrive à un routeur MPLS, le Data Plane extrait l'étiquette MPLS pour déterminer le chemin optimal vers sa destination. Le routeur applique ensuite les transformations nécessaires sur l'en-tête du paquet et le transmet au prochain routeur MPLS sur le chemin. Le Data Plane de MPLS joue un rôle crucial dans l'acheminement rapide et efficace des paquets de données, ce qui est essentiel pour des applications nécessitant des performances réseau élevées et une faible latence, comme la voix sur IP, la vidéo en temps réel et les services de cloud computing.

### Le Contrôle Plane

Le Contrôle Plane est la partie de MPLS chargée de la gestion du réseau. Il configure et surveille le réseau MPLS, distribue les informations de routage et d'étiquetage aux équipements, et utilise des protocoles de routage tels que OSPF ou BGP pour échanger ces informations. Il établit des chemins de transport à travers le réseau en tenant compte de la topologie et des préférences de routage. Le Contrôle Plane distribue également les étiquettes MPLS aux équipements, qui sont utilisées par le Data Plane pour acheminer les paquets de données efficacement. En garantissant des informations de routage et d'étiquetage à jour, le Contrôle Plane joue un rôle essentiel dans la gestion fiable du réseau MPLS. Il assure une connectivité optimale pour les applications et les services du réseau.

### Commutation Labels

Une architecture standard qui permet d’adopter la technologie MPLS se compose par deux éléments essentiel E-LSR (Edge Label Switch Router), LSR qui sont les équipements intermédiaires de réseau MPLS. Dans le nuage MPLS, la permutation d'étiquette est effectuée au niveau d'un LSR (Label Switch Router) en analysant une étiquette entrante, qui est ensuite permutée avec l'étiquette sortante et envoyée au saut suivant.

Les étiquettes ne sont imposées qu'une seule fois sur les paquets en périphérie du réseau MPLS, au niveau du Ingress E-LSR, où un calcul est effectué pour attribuer un label spécifique au datagramme.

Ce calcul n'est effectué qu'une seule fois, lors de l'arrivée du datagramme d'un flux au Ingress E-LSR. Le label est supprimé à l'autre extrémité par le Egress E-LSR. Le mécanisme de fonctionnement est le suivant : Le Ingress LSR (E-LSR) reçoit les paquets IP, réalise une classification des paquets, y assigne un label et transmet les paquets labellisés au nuage MPLS. En se basant uniquement sur les labels, les LSR du nuage MPLS commutent les paquets labellisés jusqu’à l’Egress LSR qui supprime les labels et remet les paquets à leur destination finale. L’affectation des étiquettes aux paquets dépend des groupes ou des classes de flux FEC (Forwording Équivalence Classes) et en se basant aussi sur la table LFIB (Label Forwording Information Base) qui permet d’envoyer notre paquet en se basant sur les labels qui sont affecté a une destination précise comme il est présenté sur la figure suivante :

Figure 13 : Une architecture simplifiée des équipements de MPLS



Les paquets d'une même classe FEC (Forwarding Equivalence Class) sont traités de manière identique. Un chemin prédéfini, appelé LSP (Label Switched Path), est établi par MPLS pour un flux de paquets, en fonction des étiquettes qui leur sont attribuées. Chaque routeur MPLS, appelé LSR (Label Switch Router), utilise l'étiquette pour déterminer le prochain saut vers la destination du paquet. Chaque LSR dans le chemin LSP est responsable de la commutation des étiquettes et de l'acheminement du paquet vers le prochain LSR. Ainsi, tous les paquets d'un même flux partageant la même étiquette MPLS suivent le même chemin LSP établi.

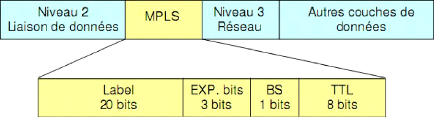


Figure 14 : La structure de l’entête MPLS

L'en-tête MPLS est composé de plusieurs champs. Le champ principal est le label de 20 bits, qui identifie le chemin LSP auquel le paquet appartient. Le champ EXP de 3 bits contient des informations sur la classe de service. Le champ S, d'un bit, indique s'il s'agit du dernier label empilé ou s'il y en a d'autres. Enfin, le champ TTL de 8 bits indique la durée de vie du paquet dans le réseau.

* + - **Label**

Les labels dans MPLS sont utilisés pour faciliter l'acheminement des paquets à travers le réseau. Ils sont manipulés à chaque bond le long du chemin LSP. Lorsqu'un paquet entre dans le réseau MPLS, un label est ajouté à son en-tête (Push) pour le diriger vers la prochaine étape du chemin. À chaque étape, le label est échangé (Swap) avec un nouveau label correspondant à la prochaine destination. Lorsque le paquet atteint sa destination finale, le dernier label est retiré (Pop) et le paquet est traité comme un paquet IP classique. Lorsque le dernier label est retiré, il peut être nécessaire de faire une recherche d'adresse IP pour acheminer correctement le paquet vers sa destination. Cependant, pour éviter d'envoyer le paquet à un autre équipement pour effectuer cette recherche, le concept de PHP est utilisé. Le PHP permet à l'équipement qui retire le dernier label MPLS d'effectuer immédiatement une recherche IP pour atteindre la destination finale, optimisant ainsi le traitement des paquets et économisant la bande passante réseau.

* **FEC (Forwarding Equivalent Class)**

C’est une représentation d’un groupe de paquet ayant le même besoin en termes de service. Elle est associée une fois pour toutes à un paquet IP lors de son entrée au réseau MPLS.

### Implicite Routing (LDP)

Dans le but d’établir les chemins LSP au sein du réseau MPLS, les LSR en utilisant le protocole LDP et distribuer les labels de manière implicite. Le protocole LDP permet aux LSR de s'informer mutuellement du mapping entre les labels et les flux de données.

Une image contenant texte, capture d’écran, diagramme, conception

Description générée automatiquement

Figure 15 : Le fonctionnement de protocole LDP

Lorsqu'un LSR reçoit un flux pour lequel il n'a pas de label, il utilise LDP pour obtenir les informations nécessaires des autres LSR afin de créer un label et établir un chemin LSP pour ce flux. Les LSR échangent des messages LDP pour se mettre d'accord sur les labels à utiliser et les chemins LSP à établir. Le protocole LDP utilise une méthode de découverte dynamique et peut nécessiter l'utilisation de LSR intermédiaires pour l'échange de messages lorsque les LSR ne sont pas directement connectés.

Une image contenant cercle, capture d’écran

Description générée automatiquement

Figure 16 : Le fonctionnement de LSP

Le protocole LDP permet une découverte bidirectionnelle des nœuds adjacents grâce à l'échange de messages Hello via UDP. Une fois que deux nœuds se sont découverts, ils établissent une session TCP pour assurer la fiabilité du transport des messages, tels que les messages d'établissement de session, les messages d'annonce de labels et les messages de notification.

### Explicit Routing

L’explicite Routing est la solution MPLS pour faire du trafic Engineering dont l’objectif d’utiliser efficacement des ressources du réseau et d’éviter les points de forte congestion en répartissant le trafic sur l’ensemble du réseau. En effet, le plus court chemin déterminé par le routage classique IP pour atteindre une destination peut ne pas être possible et certains chemins alternatifs peuvent être sous-utilisés alors que le plus court chemin est surutilisé. Le trafic Engineering a été réalisé grâce à des métriques de liens associées à des protocoles de routage internes (RIP, OSPF, IS-IS).

L’explicite routing permet à un opérateur de faire du trafic engineering en imposant au réseau des contraintes sur les flux, du point source jusqu’au point destination. Dans le but que l'opérateur peut spécifier un chemin alternatif pour des raisons de performance, de sécurité ou de politique. Le réseau détermine lui-même le chemin se basant sur :

* + - * + L’état du réseau : au niveau de la topologie, la bande de passante réelle d’un lien, la bande de passante utilisée, la bande passante restante.
        + Calcule le chemin : à l’aide des protocoles de routage chaque nœud puisse calculer le meilleur chemin pour in flux de données donné.
        + Établissement du ER-LSP : Une fois que la source a identifié le chemin complet du début à la fin, elle peut spécifier explicitement les LSR à travers les quels le flux doit passer. Dans cette étape il faut informer les LSR pour réserver les ressources nécessaires pour les chemins spécifier d’une façon que le moment que le flux va se traverser ces LSR nous devons assurer que les ressources existent. Il existe deux protocoles de signalisations qui sont couramment utilisées pour établir un ER- LSP:
* **RSVP (Ressource Reservation Protocol) :** ce protocole fonctionne par l’envoie des messages de contrôle de bout en bout pour négocier des réservations de ressources à travers les différents nœuds du chemin de communication.
* **CR-LDP (Constraint-Based Routing LDP)** : Le fonctionnement de CR-LDP est similaire à celui de LDP, mais avec des extensions pour prendre en compte les contraintes de qualité de service. Contrairement à RSVP, CR-LDP ne nécessite pas d'installation de route explicite sur les routeurs intermédiaires pour établir le chemin de bout en bout. Au lieu de cela, le chemin est spécifié à la source, puis les nœuds intermédiaires utilisent des informations de contraintes pour déterminer le prochain saut dans le chemin.

## Les Application de MPLS

Le MPLS présente plusieurs applications courantes dans le domaine des réseaux. Tout d'abord, l'ingénierie de trafic permet de diriger le trafic via un chemin spécifique, offrant une distribution optimale du trafic et améliorant l'utilisation globale du réseau. Ensuite, la bande passante garantie permet aux fournisseurs de services d'allouer des largeurs de bande passante et des canaux garantis, offrant ainsi une meilleure qualité de service pour le trafic prioritaire, tel que la voix et les données. Le routage rapide est une autre application importante du MPLS, car il permet une reprise très rapide après la défaillance d'une liaison ou d'un nœud. Cela évite toute interruption des applications utilisateur ainsi que toute perte de données.

Les réseaux privés virtuels MPLS simplifient également considérablement le déploiement des services par rapport aux VPN IP traditionnels. Ils peuvent facilement monter en charge tout en offrant le même niveau de confidentialité que les technologies de niveau 2, et ils peuvent également transporter des adresses IP non-uniques à travers un domaine public. La fonction Classe de service (CoS) MPLS assure que le trafic important est traité avec la priorité adéquate sur le réseau et que les exigences de latence sont respectées. Les mécanismes de qualité de service IP peuvent être mis en œuvre de façon transparente dans un environnement MPLS.

# Ingénierie de trafic avec MPLS (MPLS-TE)

L'ingénierie du trafic Internet est une discipline qui vise à optimiser la performance du réseau et à réduire la congestion du trafic. Traffic Engineering permettant de définir des chemins de routage explicites dans les réseaux IP (avec RSVP ou CR-LDP). L’ingénierie des flux est la faculté de pouvoir gérer les flux de données transportés au-dessus d’une infrastructure réseau, cette dernière a un objectif principal c’est de faciliter des opérations réseau efficaces et fiables tout en optimisant l'utilisation des ressources du réseau et les performances du trafic.

Et pour atteindre cet objectif, les ingénieurs du trafic Internet utilisent une variété de techniques telles que la gestion de la bande passante, la répartition de charge, le contrôle de congestion et la qualité de service (QoS). Sachant que les protocoles prennent en compte les caractéristiques du trafic et la disponibilité de la bande passante pour prendre des décisions de routage efficaces et prévenir la congestion.

## Le type de service (QoS)

MPLS TE utilise des tunnels MPLS pour acheminer le trafic de manière efficace et prévisible à travers le réseau. Ces tunnels sont configurés en fonction des contraintes de QoS, permettant ainsi de garantir une qualité de service appropriée pour les flux, lorsque le trafic traverse le réseau, il est routé sur des chemins qui garantissent que les exigences de QoS sont respectées.

La bande passante est allouée en fonction des exigences de QoS, en garantissant que chaque flux de trafic reçoit la bande passante minimale requise pour fonctionner correctement. Sachant que les priorités de traitement appropriées sont également assignées à chaque flux de trafic, garantissant ainsi que les flux de trafic à haute priorité reçoivent un traitement prioritaire dans le réseau.

## L’agrégation de flux

L'une des forces de MPLS est sa capacité d'agrégation de flux, qui permet de combiner le trafic entrant dans un routeur via plusieurs LSP en un seul LSP de sortie. Cette agrégation de flux réduit considérablement le nombre de connexions que les routeurs de cœur de réseau ont à gérer. MPLS introduit également un concept de hiérarchie au niveau des LSP et d'empilement des labels. Cela permet de construire des LSP encapsulés dans un autre LSP, qui lui-même peut être encapsulé dans un autre LSP, et ainsi de suite. en MPLS, le nombre de niveaux d'encapsulation n'est pas limité à deux. Ce concept d'encapsulation hiérarchique offre une flexibilité accrue dans la conception de la topologie de réseau et permet une meilleure gestion du trafic.

# Protocoles de routage

## Protocole de routage IGP

Afin de réaliser la communication au sein du système autonome, le protocole IGP doit être configuré. Les protocoles IGP utilisés pour les grands systèmes autonomes sont : ISIS et OSPF

### Le protocole ISIS

Le protocole de routage « Intermédiate System to Intermédiate System » (IS-IS) un « Interior Gateway Protocol » (IGP), qui fonctionne au sein d’un Système Autonome (AS).

Il est de type à état de liens :

• Les systèmes intermédiaires (routeurs) ont une connaissance exacte et complète de la topologie du réseau.

• Chaque système intermédiaire (IS) connaît l’existence de ses voisins adjacents.

• Les émissions des mises à jour sont déclenchées par des modifications topologiques.

• L’algorithme utilisé est « Shortest Path First » de Dijkstra.

IS-IS intégré est un protocole classless. Il est optimisé pour supporter le CIDR (agrégation de routes) et le VLSM (masque variable de sous réseau).

* **Principales caractéristiques d'IS-IS :**

Une image contenant capture d’écran, texte, cercle, diagramme

Description générée automatiquement

Figure 17 : Architecture générale en IS- IS

IS-IS est conçu pour prendre en charge de grands réseaux de taille et de complexité évolutives. En conséquence, les systèmes autonomes sont organisés hiérarchiquement en régions. Cela permet aux routeurs d’avoir une table de routage de taille réduite, ainsi qu’une convergence plus rapide. La majorité d’équipements à l’intérieur d’une zone est isolée de l’extérieur. En effet, on pourrait considérer une zone comme un mini système autonome. Un routeur IS-IS peut être identifié dans l’un des trois niveaux (level) suivants :

* Niveaux 1 pour le routage intra-zone.
* Niveaux 2 pour le routage interzone.
* Niveaux 1 /2 correspond aux routeurs capables d’opérer sur Niveaux 1 et 2.

### Le protocole OSPF

OSPF est un IGP utilisé à grande échelle (AS). C'est l'un des protocoles d'état de lien. Les routeurs OSPF échangent l'état, le coût et d'autres informations pertinentes avec les voisins. L'échange d'informations permet à tous les routeurs de construire une carte de la topologie du réseau. Chaque routeur applique un algorithme (SPF) pour calculer le chemin le plus court vers chaque destination du réseau. La table de transfert OSPF résultante est Soumise au RTM pour calculer la table de routage.

* **Caractéristiques** **de l’OSPF**

Une caractéristique clé d'OSPF est la prise en charge des très grands Internet, grâce à Regroupez les routeurs en entités logiques appelées "secteurs" ou zones. Cette La communication interrégionale ne permet de transmettre que le minimum d'informations nécessaires Gardez la zone connectée. En conséquence, tous les calculs d'itinéraire fonctionnent Ne se produit que dans la même zone. Les routeurs de la zone ne sont pas Un autre domaine de changement.

Les routeurs OSPF peuvent prendre en charge trois types d'opérations : opérations de zone, Les connexions interzones et les connexions entre systèmes autonomes (AS) nous distinguons donc quatre Types de routeurs OSPF :

* Routeur interne (IR) - IR exécute une fonction dans une seule zone. Sa fonction principale est de maintenir à jour sa base de données d'état des liens. Il renvoie toute information aux autres routeurs de sa zone, routage ou L'inondation d'autres zones nécessite l'intervention de l'Area Border Router (ABR).
* Routeur principal (BR) - L'une des règles de conception OSPF est que chaque zone est L'inter réseau doit être connecté à une seule zone, zone 0 ou backbone. Cette La plupart des BR ont une interface connectée à la zone dorsale et un ou plusieurs interfaces avec d'autres domaines.
* Routeur de frontière de zone (ABR) - Un ABR connecte deux zones ou plus. il a Il existe autant de bases de données d'état de liens que d'interfaces connectées à différentes zones. Chacune de ces bases de données contient toute la topologie de la zone liée. Ces informations peuvent être transmises à la zone dorsale distribuer.
* Autonomous System Border Router (ASBR), puisque OSPF est un IGP, ce dernier doit se connecter au reste d'Internet via Autre AS. Ce type de routeur agira comme une passerelle vers un ou Plusieurs AS.

Une image contenant texte, diagramme, cercle, ligne

Description générée automatiquement

Figure 18 : les différentes zones en OSPF

### Comparaison entre IS-IS et OSPF

* IS-IS et OSPF sont des protocoles à état de liaison, et les deux protocoles utilisent l'algorithme de Dijkstra pour trouver le chemin le plus court. Les deux protocoles prennent également en charge les masques de réseau de longueur variable, peuvent utiliser la multidiffusion pour trouver des routeurs à proximité à l'aide de paquets hello et peuvent prendre en charge l'authentification lors de la mise à jour des routes.
* Alors qu'OSPF est à l'origine conçu pour acheminer les adresses IP et fonctionne lui-même comme un protocole de couche 3 sur IP, IS-IS est à l'origine la couche réseau du modèle OSI. L'adoption généralisée des adresses IP a contribué à la popularité croissante d'OSPF. IS-IS n'utilise pas les adresses IP comme routes. Il est indépendant de l'adresse réseau, tandis que OSPF a été conçu pour IPv4. Cela facilite l'utilisation d'IS-IS pour IPv6. Pour prendre en charge les réseaux IPv6, OSPF a été réécrit dans OSPF v3.
* Les routeurs IS-IS construisent une représentation topologique du réseau. Cette carte montre les sous-réseaux que les routeurs IS-IS peuvent atteindre et les chemins les moins chers (les plus courts) pour transférer le trafic.
* La principale différence entre OSPF et IS-IS réside dans la manière dont ils divisent le système autonome en zones et dont le routage entre les zones est effectué. L'inconvénient d'IS-IS est que les routeurs de couche 2 ne peuvent communiquer qu'avec les mêmes routeurs, la couche 1 a la même situation. Les routeurs de couche 1-2 sont utilisés pour l'interaction entre les routeurs des 1er et 2ème niveau et les zones correspondantes. Dans OSPF, les zones sont délimitées par des interfaces sur le routeur, de sorte qu'un routeur de bordure de zone (ABR) peut se trouver dans plusieurs zones en même temps, créant ainsi des frontières en son sein. Alors que les limites de zone IS-IS se situent entre les routeurs de couche 2 ou de couche 1-2, ce qui fait qu'un routeur IS-IS ne fait partie que d'une seule zone. De plus, IS-IS ne prend pas en charge la zone nulle (zone dorsale, traversé par tout le trafic interrégional). L'explication logique à cela est que l'OSPF crée un réseau de topologie en étoile avec de nombreuses zones se croisant avec zéro, tandis que IS-IS crée une topologie logique avec une couche principale de routeurs de couche 2, des routeurs de branche - couche 1-2 et des zones séparées de la couche 1 routeurs.
* Le protocole OSPF a plus d'extensions et de fonctionnalités supplémentaires. Cependant, le protocole IS-IS envoie moins de surcharge et peut évoluer vers de grands réseaux. Avec la même quantité de ressources, IS-IS pourra prendre en charge plus de routeurs dans une zone qu'OSPF. Cela encourage IS-IS à être utilisé par les FAI.

## Protocole BGP

### Généralité

Est un protocole utilisé dans les grands réseaux ainsi que l’internet, qui parcourir au niveau de plusieurs autonomes systèmes, même si ça convergence est plus faible mais sa performance est très importante. BGP ou Border Gateway Protocol, c’est un protocole EGP Exterior gateway protocol. Ce protocole a plusieurs avantages tant qu’il est utilisé dans les grandes réseau il est capable de supporter un nombre important des trafics et des routes.

Une image contenant diagramme, texte, cercle, ligne

Description générée automatiquement

Figure 19 : Architecture de protocole BGP

Le BGP possible de l’utiliser en tant que protocole externe ou interne, pour le premier cas nous pouvons l’utiliser pour relier plusieurs autonomes systèmes différents pour se connecter entre eux, et nous pouvons aussi utiliser au niveau d’un seul AS. Sachant qu’un équipement qui relie deux AS différents est très performant il doit avoir le minimum 256 M.

### Autonomes System

AS est un groupe des routeurs qui partagent entre eux une policy de routage dans un seule domaine administratif, donc une AS se caractérise par un numéro identifiant unique qui serait entre 1 et 65535 sachants que le AS nulle 0 est réserver, et entre 1 et 64495 utiliser pour les AS public, et entre 64512 et 65535 et pour les AS privées. Il existe des types des AS. Nous pouvons réserver un numéro d’une AS lorsque nous voulons avoir une connexion de notre AS avec multi Home d’une façon que nous serons capables de diriger notre trafic vers une AS bien précisé qui baser sur des notions des attribues.

Une image contenant cercle, diagramme

Description générée automatiquement

Figure 20 : Le rôle de la notion AS

Pour établir une relation de voisinage entre deux routeurs BGP il faut distribuer les routes de chaque réseau et les partagés avec l’autre routeur.

### BGP Split Horizon

E-BGP utilise l'AS-PATH pour éviter les boucles, les voisins I-BGP n'ajoutent pas leur ASN à l'AS\_PATH lors de l'envoi de mises à jour. Alors, qu'est-ce que I-BGP utilise pour la prévention des boucles ? Split Horizon est parmi les principaux mécanismes pour éliminer les boucles BGP, ce mécanisme permet un routeur appartient à une AS et de pas partager les routes qu’il reçoit pour la raison que ne pas avoir une boucle.

Une image contenant texte, capture d’écran, diagramme, orange

Description générée automatiquement

Figure 21 : Fonctionnement de Split Horizon

### BGP peering full mesh

Pour qu’un routeur Edge peut partager les routes d’une autre AS c-à-d un routeur BGP Peer, il faut configurer une commande neighbor x.x.x.x masq next-hope self, cette commande peut partager avec les routeurs de la même AS qui sont en relation par I-BGP les routes des BGP de la AS connectent par E-BGP de notre routeur. Donc on peut résoudre le problème par une relation full mesh .

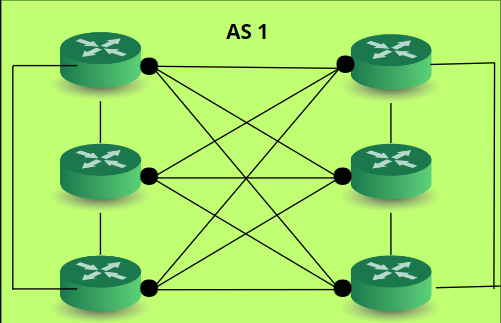


Figure 22 : Architecture full mesh

La notion de full mesh n’est pas réalisable car si nous somme dans le cas ou nous avons plusieurs BGP Peers cela va nous créer des problèmes sur tous que nous travaillons sur une architecture Backbone qui contiennent un nombre important des équipements. Sauf qu’un réflecteur de route n'est pas autorisé à modifier un attribut de route réfléchi.

### Router réflecteur principe

C’est une solution pour réduire le nombre de peering BGP au sein d’un AS est la réflexion de route. Au lieu que chaque système BGP doive s’apparier avec tous les autres systèmes BGP avec l’AS, chaque équipement BGP s’apparie à la place avec un réflecteur de routeur. Les annonces de routage envoyées au réflecteur de route sont ensuite renvoyées à tous les autres locuteurs BGP. Il existe en générale trois rôles principaux :

* + - * + Client
        + Non-Client
        + Une image contenant capture d’écran, diagramme, cercle, ligne

          Description générée automatiquementServeur RR

Figure 23 : Fonctionnement de Routeur Réflecteur

Les trois attributs BGP utilisés pour empêcher les boucles d'informations de routage lors de l'utilisationde réflecteurs de route.

# Les services MPLS (Les VPNs basés sur MPLS)

Les réseaux MPLS offrent aux fournisseurs de services de télécommunications une méthode efficace elle permet aux fournisseurs de créer des réseaux privés virtuels sur leur infrastructure partagée, qui va permettre aux entreprises de connecter leurs différents sites géographiques de manière transparente et sécurisée. Les offres de services VPN sur MPLS sont conçues pour répondre aux différents besoins des entreprises en matière de connectivité et de sécurité, il existe plusieurs types de services VPN basés sur MPLS, tels que le VPN MPLS Layer 3 (L3VPN), le VPN MPLS Layer 2 (L2VPN) et le VPLS (Virtual Private LAN Service), chacun offrant des avantages spécifiques en termes de flexibilité, de sécurité et de performances.

## VPN VPRN L3

Les VPN de couche 3 (Layer 3 VPN) permettent aux opérateurs de réseau de fournir des services de VPN basés sur le routage IP aux clients. Les préfixes IP annoncés par chaque site client sont transportés par le réseau de MPLS et sont séparés des autres préfixes IP de clients différents.

Les VPN de couche 3 offrent une solution de connectivité à longue distance, en permettant à des sites géographiquement dispersés de communiquer entre eux comme s'ils étaient sur le même réseau local. Les VPN de couche 3 sont utilisés dans les entreprises pour relier des sites distants.

Une image contenant diagramme, texte, ligne, capture d’écran

Description générée automatiquement

Figure 24 : Exemple de VPN L3

* **Fonctionnement :**

Le service VPRN permet aux clients de gérer leur propre adressage IP et de prendre en charge le routage à l'intérieur de leur réseau privé. Les informations de routage sont échangées entre les clients et les routeurs PE du fournisseur de services. Pour éviter les conflits d'adresses IP, des mécanismes tels que les tables VRF et les route distinguisher sont utilisés pour isoler les réseaux privés des clients. Les routes sont annoncées entre les PE et les CE, assurant une connectivité sécurisée entre les différents sites des clients. La valeur Route Target est utilisée pour identifier le routeur de destination du client.

Une image contenant texte, diagramme, capture d’écran

Description générée automatiquement

Figure 25 : Le fonctionnement de VPN L3 VPRN

Dans les services VPRN, deux plans sont utilisés pour l’opération : le plan de contrôle qui permet d'identifier les préfixes VPRN et le réseau client, et le plan de données qui permet le transport et la transmission des données en fonction de l'étiquette de service VPN. le VPN L3 vient pour donner la possibilité de liée des site distant des client par une connexion sécurisé.

## Epipe VPN L2

Les routeurs Alcatel-Lucent prennent en charge un service point à point de couche 2 appelé Virtual Private Wire Service (VPWS), il existe plusieurs services tel qu’Ethernet (Epipe), ATM (Apipe) et Frame Relay (Fpipe), chacun étant utilisé pour émuler un tunnel sur un réseau IP à commutation de paquets. Les Layer 2 VPNs sont souvent utilisés pour relier des sites géographiquement dispersés en étendant un réseau local (LAN) via un WAN.

Figure 26 : Exemple de VPN L2

Une image contenant texte, capture d’écran, diagramme, conception

Description générée automatiquement

Les Layer 2 VPNs sont souvent utilisés dans les environnements où il est nécessaire de conserver les informations de couche 2 des paquets, telles que les adresses MAC. Ils sont généralement déployés dans des réseaux MPLS qui fournissent des services VPN à leurs clients.

* **Fonctionnement :**

Avec le service EPIPE, les données Ethernet provenant d'un réseau d'extrémité (CE) sont encapsulées et transportées sur un réseau fournisseur de services (PE) via un tunnel dédié. Ce tunnel, établi sur un réseau IP ou MPLS, fonctionne comme un pseudowire qui émule une liaison Ethernet privée entre les sites connectés. EPIPE permet aussi de fournir des connexions Ethernet sécurisées et isolées entre différents sites d'une organisation. Il offre des fonctionnalités telles que la transparence Ethernet, le contrôle de la bande passante et la qualité de service (QoS) pour garantir la fiabilité et les performances nécessaires.

## VPLS VPN L2

Le réseau de fournisseur de services permet de connecter plusieurs sites clients au sein d’une zone géographique limité, comme une ville. Pour établir ces connexions, des commutateurs et des bridges Ethernet sont utilisés pour créer des arbres couvrants, permettant d’acheminer le trafic entre les différents sites clients.

Le service Virtual Private LAN Service (VPLS) est une classe de service de réseau privé virtuel qui permet de connecter plusieurs sites dans un seul domaine de pontage via un réseau IP/MPLS géré par le fournisseur.

* **Fonctionnement :**

Une image contenant diagramme, carte, texte, capture d’écran

Description générée automatiquement

Figure 27: Fonctionnement de service VPN L2 VPLS

VPLS est un type de VPN de couche 2 qui permet à tous les sites d'un client de se comporter comme s'ils étaient sur le même réseau local. Pour ce faire, VPLS utilise l'interface Ethernet du client et se compose de CE, PE et d'un cœur MPLS. Le CE est le routeur ou le commutateur situé dans les locaux du client, tandis que le PE gère les VPN et assure l'apprentissage des adresses MAC, la commutation et la diffusion. Le cœur MPLS sert uniquement à interconnecter les PE et à commuter le trafic grâce aux labels MPLS.

Un maillage complet de LSP bidirectionnels (appelés Pseudo-wires) est ensuite établi entre tous les PE d'une instance VPLS, permettant ainsi une connectivité transparente et bidirectionnelle entre tous les sites clients connectés à ces équipements. Le trafic Ethernet est encapsulé dans des étiquettes MPLS et acheminé entre les différents nœuds du réseau, émulant ainsi un circuit filaire point à point entre ces sites.

Un Pseudo-Wire est composé de deux LSPs qui sont configurés dans des directions opposées pour permettre à un PE de connaître les adresses MAC. Lorsqu'un PE reçoit une trame Ethernet avec une adresse MAC source inconnue, il sait sur quel PW elle est arrivée, de sorte que cette adresse devient accessible. Pour chaque instance VPLS, les PE implémentent un commutateur virtuel grâce à une table de correspondance entre les adresses MAC et les labels LSP (PW), appelée FLIB. Cette FLIB permet aux PE de remplir leur rôle de commutation et de diffusion pour chaque instance VPLS.

# Les VRFs (Virtual Routing and Forwarding)

Le VRF est une technologie incluse dans les routeurs de réseau IP qui permet à plusieurs instances d’une table de routage de coexister dans un routeur et de travailler simultanément. La fonctionnalité réseau est alors amélioré, car les chemins réseau peuvent être segmentés sans faire appel à plusieurs routeurs. Dans la mesure ou le trafic est automatiquement isolé, le VRF permet aussi d’accroître la sécurité du réseau et d’éviter le chiffrement et l’authentification. Les fournisseurs d’accès internet (FAI) s’appuient souvent sur le concept VRF pour créer des réseaux privés virtuels (VPN) distinctes pour chaque client. C’est pourquoi on fait également référence à cette technologie sous l’appellation Routage et transfert VPN.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure 28 : Le fonctionnement de VRF

Le VRF fonctionne comme un routeur logique, mais à la différence d'un routeur physique qui peut inclure plusieurs tables de routage, une instance VRF n'utilise qu'une seule table de routage, séparer par rapport aux autres tables de routages des autre VRF.

# Qualité de service

## Classification des applications et besoin de QoS

Généralement les réseaux des entreprises offrent différents services tels que la voie, vidéo, le transfert de données… il est évident que chaque service soit plus accessible que d’autres, donc en tant que des ingénieures des réseaux nous somme besoin de valorisé un flux qui porte le trafic d’un service voie/vidéo par rapport le trafic de transférer de données. La QoS (Qualité de Service) fonctionne en donnant la priorité à certains flux de données.

## Paramètres-clé de la QoS

Il existe plusieurs paramètres pour configurer la QoS des flux IP :

* **La latence :** qu’est le délai total pour traverser un réseau, leur métrique c’est le délai Round Trip Time (délai AR) ce paramètre concerne les applications interactives.
* **Gigue (jitter) :** qu’est la variation du délai, ce paramètre a deux métriques le délai max et le délai min, ce paramètre utilisant dans les applications temps réel (audio, vidéo).
* **Bande passante disponibles :** leurs métriques c’est le débit minimal, le débit moyen, le débit en pointe, ce paramètre concentre aussi sur les applications interactives audio et vidéo.
* **La perte de paquets :** qu’est basé sur le taux de perte et le taux d’erreur.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Figure 29: Les exemples de classes de service

## Techniques de traitement de la QoS

Nous pouvons traiter la QoS en classifiant les flux, et prévenir de la congestion conditionnement à l’entrée, d’une façon qu’en identifiant le type de trafic et puis en l’assignant un priorité traite les files d’attente on déclare chaque flux qu’il vient et on l’associer à la file d’attente qu’il convient comme ça chaque flue serait traiter en se basant a leur priorité.La première étape du QoS est la classification du trafic.

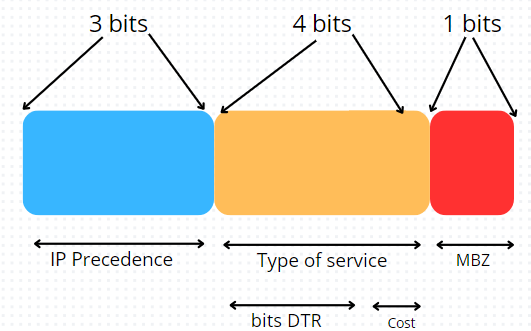


Figure 30 : Entête QoS

Le champ TOS se compose de 8 bits, les 3 premiers bits sont réservés pour les niveaux de priorité, puis ensuite 4 bits pour le mode de transport selon son type 3 bits pour le DTR, DELAI COURT, DEBIT ELEVE, GRANDE FIABILITE, 1 bit pour le coût et le dernier bit est inutilisé, il doit être mis à 0 (MBZ, Must Be Zero). Les bits des priorités présentent huit niveaux de priorité :

|  |  |
| --- | --- |
| **IP Precedence** | **Signification** |
| Priorité 1 | Routine |
| Priorité 2 | Prioritaire |
| Priorité 3 | Immédiate |
| Priorité 4 | Urgent |
| Priorité 5 | Très urgent |
| Priorité 6 | Critique |
| Priorité 7 | Supervision interconnexions |
| Priorité 8 | Supervision réseaux |

Tableau 1 : Les signification des priorités

Les priorités sont entre 1 et 8 comme le tableau montre le trafic va prendre une valeur d’une priorité, pour les priorités entre 1 et 5 qu’ont responsable de différencier le trafic et pour les priorités 6 et 7 sont recommandées pour la communication entre les différents équipements. La priorité 3 associer pour le protocole de signalisation de voix, et pour la priorité 4 associer aux flux vidéo, et pour la priorité 5 est utilisée pour la voix.

## Les Modèles de la QoS

Pour définir la qualité de service QoS pour les flux de données IP, il existe deux types d’architecture étudiées, la QoS intégrée (Integrated Services - IntServ) et la QoS différenciée (Differentiated Services - DiffServ). Le modèle IntServ propose une architecture qui permet de garantir la qualité de service (QoS) sans nécessiter de modifications du protocole IP. Pour ce faire, IntServ utilise un protocole de signalisation dédié appelé RSVP. Et le modèle DiffServ propose une approche de classification du trafic basée sur des classes, contrairement à IntServ qui se concentre sur une classification basée sur les flux individuels.

### IntServ

IntServ suppose que la QoS est basée sur la définition de classes de service et la réservation statique ou dynamique des ressources dans les différents éléments du réseau en utilisant le protocole de réservation RSVP. Ces ressources permettent d’assurer une certaine qualité de service pour les flots identifiés ayant requis cette qualité de service. L'utilisation d'IntServ peut être complexe, surtout dans les réseaux de grande taille, car elle nécessite un maintien de l'information de réservation de ressources pour chaque flux de données IP. Cela peut entraîner une surcharge du réseau et une complexité accrue pour les administrateurs réseau

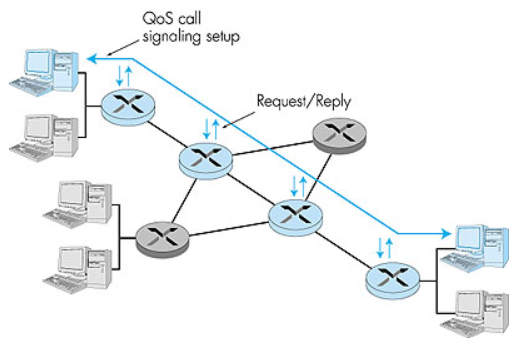


Figure 31: Le fonctionnement de IntServ

### DiffServ

Le modèle propose de regrouper les flux homogènes en les classifiant, les marquant de manière identique, les régulant en fonction du profil associé (SLA ; Service Level Agreements) et en les transportant entre deux nœuds adjacents. Contrairement à IntServ, ce transport ne nécessite pas de signalisation de réservation, mais un comportement par routeur ou PHB (Per Hop Behavior) basé sur la priorité par classe pour répondre à la demande de qualité de service. Les différents flux sont classifiés selon des règles prédéfinies, puis agrégés selon un nombre limité de classes de services. Cette approche permet de minimiser la signalisation et simplifie le traitement des flux de données. Chaque routeur gère la qualité de service pour chaque flux de données en fonction de la classification de la classe de service.

Une image contenant diagramme, cercle, ligne, capture d’écran

Description générée automatiquement

Figure 32: Le fonctionnement de DiffServ

* **Fonctionnement de DiffServ**

Le modèle DiffServ fonctionne grâce à des routeurs Edge et des routeurs de cœur de réseau, chacun ayant des fonctionnalités spécifiques. Les routeurs Edge sont chargés de vérifier la conformité du trafic par rapport au SLA. Si le trafic est conforme, les paquets sont marqués et remis en forme. Les routeurs de cœur de réseau gèrent quant à eux l'acheminement des paquets en fonction de la gestion des files d'attente et de l'ordonnancement. Les routeurs Edge permettent également de découvrir la classe de service attribuée à chaque paquet sur la base du SLA. Cette classe est déterminée en analysant l'en-tête DSCP qui contient des informations sur le type de service, les adresses sources et destinations, les protocoles et les ports. En fonction des mesures effectuées sur les débits moyens et crête et conformément au SLA, les routeurs Edge peuvent décider d'éliminer ou de conserver les informations des paquets. En cas de congestion dans le cœur de réseau, les paquets éliminables sont traités dans une classe moins prioritaire ou détruits. Les paquets sont également régulés en fonction de leur classe, classés ou détruits en cas de congestion. L’identification et la classification du trafic peuvent être faites par l’identification de la source du trafic et sa destination. Les LSR doivent appliquer le traitement de QoS demandé dans le champ DSCP des paquets IP entrants :

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Figure 33: Comparaison d’entête IP et l’entête MPLS

DSCP (differentiated services code point), est un champ dans l ‘entête IP. Il est utilisé pour différencier les services élaborés par le modèle DiffServ. Il est codé sur 6 bits : 3 bits sont consacrés pour les priorités (Drop Preference) et 3 bits définissent la classe choisie pour le paquet (Class Selector CodePoints). Le DSCP peut s’exprimer sous forme numérique ou à l’aide d’un nom appelé Per-Hop Behavio(PHB).

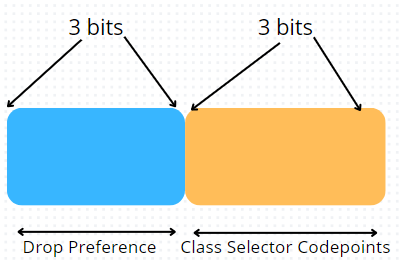


Figure 34: L’entête DSCP

Ce que nous pouvons remarquer en se basant sur la figure précédente c’est que le DSCP est dans l’entête IP donc il ne sera plus visible, le LSR Ingress marque le champ Exp (CoS) de l’entête MPLS, il ne peut y avoir que 8 PHB au plus (3 bits). Comme il est montré sur la **figure 30** un champ de l’entête MPLS est appelé EXP, il s’agit du champ expérimental qui permet d’utiliser la QoS. Ce champ est de 3 bits et il est lié à la valeur du champ DSCP. Ces 3 bits correspondent aux 3 premiers bits du champ DSCP.

# L'automatisation via Ansible

## L’architecture d’Ansible

Un logiciel d’automatisation du réseau trouve le moyen le plus efficace de mapper, configurer, provisionner et gérer un réseau. Alors, l’automation c’est l’interaction entre un système centralisé qui gère les différents scriptes afin d’être interagir avec les équipements réseau (configuration, déploiement, modification, suppression). C'est un ensemble des opérations enchaîné bien structuré consiste a utilisé un logicielle.

Ansible est un logiciel d’automatisation les plus connues, c’est un logiciel libre de gestion des configuration et open source, il nous permet de contrôler des nœuds distants

La force d'Ansible réside dans sa capacité à être utilisé non seulement pour le déploiement à un seul niveau, mais également pour les systèmes et infrastructures à plusieurs niveaux. Il est souvent décrit comme un outil "sans agent", ce qui signifie qu'il fonctionne en établissant des connexions entre les nœuds via le protocole SSH par défaut.

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

Figure 35: L‘architecture de fonctionnement Ansible

Ansible fonction comporte trois composant basiques : Nœud de contrôle, Nœud géré, Inventaire. Pour le deuxième élément, il fonction avec deux fichiers qui sont importants, le fichier Host ou bien inventorie qui porte les hôtes et les machines qui seront automatiser ces équipements doivent être déjà configuré par une configuration minimale, pour que Ansible peut communiquer avec eux, alors la configuration de SSH sur l’équipement est nécessaire. Et d’une autre part le fichier Playbook, celui qui vas porter notre configuration ou nos rôles que nous voulons configurer sur les équipements qui sont indiqué dans le fichier Host.

## Le fonctionnement en général

Premièrement, nous commençons par le système dur lequel nous installons Ansible qui s’appelle Nœud de contrôle. Ce dernier ne se connecte pas à chaque équipement via SSH et ne copie pas de code de configuration sur chaque équipement réseau. En fait, lorsqu'il fonctionne en mode local, Ansible se connecte en réalité à lui-même et exécute le script de configuration localement. Le script qui s'exécute localement peut toujours se connecter à l'équipement réseau via SSH qui sont les nœuds géré, une fois le script s’exécute et se prépare il l’enregistre dans un fichier indiqué puis il passe vers la prochaine étape c’est de pusher la configuration dans l’équipement.

## Les composantes d’Ansible

Ansible parcourt dans plusieurs fichiers et il utilise plusieurs packages et des scripts afin d’atteindre à l’étape de passer (push) une configuration, donc il est basé et structurer sur plusieurs composantes liées entre eux d’une façon organiser et bien déterminer si un élément de ces composant manque ou a une mâle configuration, l’exécution de playboy serait plus effectuée, parmi les composantes d’Ansible :

### Inventaire (Inventory)

Ansible fonctionne contre plusieurs systèmes de votre infrastructure en même temps. Il le fait en sélectionnant des parties des systèmes répertoriés dans l’inventaire d’Ansible, qui est par défaut enregistré dans le dossier : /etc/ansible/hosts. Si nous voulons spécifier un fichier d’inventaire différent en utilisant : -i < directoire>. Généralement nous pouvons représenter les équipements par leurs adresses IP ou leurs noms.



Figure 36: exemple de fichier inventaire

Le fichier d’inventaire est utilisé pour déclarer et informer Ansible les équipements qu’on veut le configurer.

### Les Groupes

Il y a deux emplacements géographiques principaux avec différents types d’appareils dans les régions AMERS et EMEA. Nous voudrons créer différents groupes afin que nous puissions automatiser facilement tous les appareils d’un certain type, pour un certain système d’exploitation, ou dans un emplacement donné. Nous pouvons organiser nos équipements en se basant sur plusieurs critères tels que les l’emplacement, l’utilité, le types de système d’exploitation, Nous commencerons par créer deux groupes pour les appareils dans la région AMERS appelés amers-cpe et amers-dc. Chaque groupe aura deux appareils. Le groupe cpe aura deux appareils Cisco CSR 1000v et le groupe dc aura deux commutateurs Nexus :

**[amers-cpe]**

**csr1**

**csr2**

**[amers-dc]**

**nxos-spine1**

**nxos-spine2**

L’inventaire Ansible peut se constituer des groupes qui comprennent les hôtes et des variables attachées à ces hôtes. Un groupe peut imbriquer d’autres groupes (children). Un hôte appartient toujours au moins à deux groupes, car deux groupes d’hôtes sont toujours présents par défaut : all qui contient tous les hôtes, et ungrouped qui contient tous les hôtes qui ne sont pas dans un autre groupe de all.

### Variables

Lors de la définition des variables de groupe, vous créez une nouvelle section dans le fichier d’inventaire avec le nom du groupe et ajoutez : **vars.**

On applique des variables comportementales qui décris des informations concernant le type de connexion des différents groupes :

**[all : vars]**

**ansible\_connection**

**ansible\_ssh\_user et ansible\_ssh\_pass**

**ansible\_become et ansible\_become\_method**

Les fichiers de variables d'hôtes et de groupes doivent respecter la syntaxe de langage YAML. Les extensions de fichier valides incluent ".yml", ".yaml", ".json" ou aucune extension de fichier.

### Playbook

Le Playbook est un fichier qui contient vos instructions d’automatisation. En d’autres termes, les Playbook contiennent les tâches individuelles et les flux de travail que vous souhaitez utiliser pour automatiser votre réseau. Le Playbook lui-même est écrit en YAML. Prenons un exemple de Playbook pour mieux comprendre sa structure et sa terminologie associée.

La figure suivante décrit différents paramètres utilisés dans Ansible pour automatiser les appareils réseau :

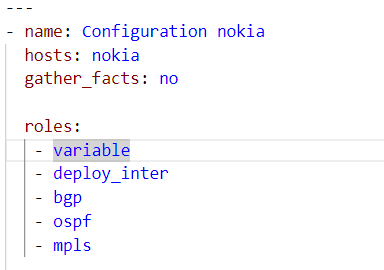


Figure 37:Exemple d’un playbook

**"name"** est un paramètre facultatif utilisé pour définir le but de la tâche d'automatisation.

**"hosts"** spécifie les appareils à automatiser.

**"gather\_facts"** est utilisé pour collecter des informations sur les appareils en cours d'automatisation.

### Les tache (Tasks)

Sont généralement la section principale dans Ansible sont les actions ou les opérations spécifiques effectuées sur les cibles par les modules Ansible. Les tâches sont les unités de travail principales dans un Playbook Ansible. Elles représentent les différentes étapes ou actions à exécuter sur les hôtes cibles.

### Rôle

Les rôles vous permettent de charger automatiquement des variables, des fichiers, des tâches, des gestionnaires et d'autres artefacts Ansible connexes en fonction d'une structure de fichiers connue. Après avoir regroupé votre contenu en rôles, vous pouvez facilement les réutiliser et les partager avec d'autres utilisateurs. Un rôle Ansible a une structure de répertoire définie avec huit répertoires

### Les modules

Sont parmi les éléments les plus important, ils sont sous forme des unités de code qu’ils sont exécutés sur les nœuds géré, nous pouvons dire que les modules sont comme des programmes qu’Ansible les utilisent afin d’atteindre au nœuds. Ces modules sont exécutés à l’aide de Playbook, ils permettent Ansible de définir les attribues des fichiers de configuration et de déploiement.

## Les structures de contrôle

### Facts

Dans le cadre d'Ansible, ils désignent les informations ou les données collectées auprès des hôtes gérés lors de l'exécution d'un Playbook. Ils fournissent un aperçu de l'état actuel des hôtes gérés et permettent d'obtenir des informations précieuses sur leur configuration, leur environnement et leurs caractéristiques.

### Les conditions (When)

Le mot-clé "when" d'Ansible est utilisé pour contrôler de manière conditionnelle l'exécution des tâches, Il vous permet de définir une condition ou un ensemble de conditions qui déterminent si une tâche spécifique doit être exécutée ou ignorée lors de l'exécution du Playbook, différents opérateurs tels que l'égalité, l'inégalité, les opérateurs logiques et les comparaisons de chaînes sont disponibles pour créer des conditions complexes.

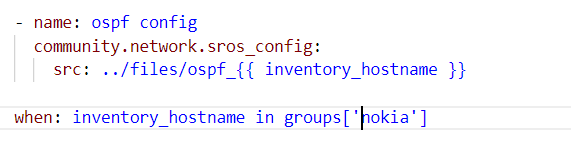


Figure 38 : Un exemple de condition When

La condition " when: inventory\_hostname in groups['nokia'] " est utilisée pour vérifier si l’hôte appartient au groupe 'nokia'. Si la condition est vraie, cela signifie que la cible fait partie du groupe 'nokia' et la tâche sera exécutée. Dans le cas contraire, si la condition est fausse, la tâche sera ignorée.

### Les filtres Jinja

Le filtre Jinja {{hostname | default('localhost')}} est utilisé pour afficher la valeur de la variable "hostname". Si la variable "hostname" est définie, elle sera affichée normalement.

Toutefois, si la variable "hostname" n'est pas définie ou est vide, le filtre "default" sera utilisé pour afficher la valeur par défaut 'localhost' à la place. Cela permet de s'assurer qu'une valeur de remplacement est utilisée si la variable principale n'est pas définie.

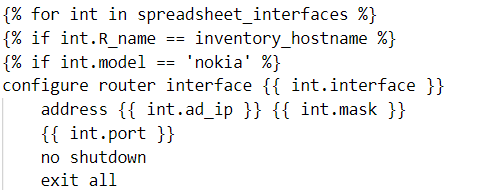


Figure 39: Un filtre en langage Jinja

## YAML

YAML est un langage de sérialisation des données qui est souvent utilisé pour coder des fichiers de configuration. Pour certains, YAML est l'acronyme de Yet Another 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 parfaitement lisible et compréhensible par l'humain. YAML est utilisé par l’outil d’automatisation Ansible pour créer des processus d'automatisation, sous la forme de Playbook Ansible

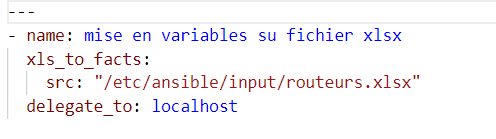


Figure 40: Une Playback écrit en YAML

YAML est un format de données flexible qui permet de structurer l'information en utilisant une indentation pour l'imbrication des éléments. Il est utilisé pour représenter des données de manière lisible et concise, en évitant les caractères de formatage spéciaux. Il est un format de données flexible qui intègre des fonctionnalités provenant de différents langages de programmation.

Chapitre 3 : Étude de l’existant

# Introduction

Ce chapitre se concentre sur l'étude de l'existant d'une topologie backbone MPLS/IP, afin de souligner les lacunes et les critiques de cette solution. Nous analyserons en détail les aspects de cette infrastructure et mettrons en évidence ses limitations, notamment l'absence de redondance et sa dépendance à une configuration manuelle pour plusieurs services tels que le routage, le MPLS et les VPN. Ce chapitre se propose donc de mettre en évidence les critiques de la solution existante de topologie Backbone MPLS/IP. L'objectif de ce chapitre est donc de mettre en évidence les critiques de la solution existante de la topologie Backbone MPLS/IP, afin de proposer plusieurs solutions pour remédier à ces points faibles et de sélectionner la meilleure option comme solution finale.

# Architecture existante

La société SII se démarque par son expertise technique avancée, sa capacité à relever des défis technologiques complexes et son engagement à fournir des solutions innovantes à ses clients. Elle entretient des partenariats stratégiques avec les principaux acteurs du marché des technologies pour offrir des services de haute qualité et se maintenir à la pointe des avancées technologiques. En tant qu'entreprise spécialisée dans les réseaux et les télécommunications, SII propose une gamme variée de services, allant des configurations de protocoles et des flux à la mise en place de services VPN. Ces services couvrent à la fois des aspects complexes et des aspects plus simples, répondant ainsi aux besoins diversifiés de ses clients.

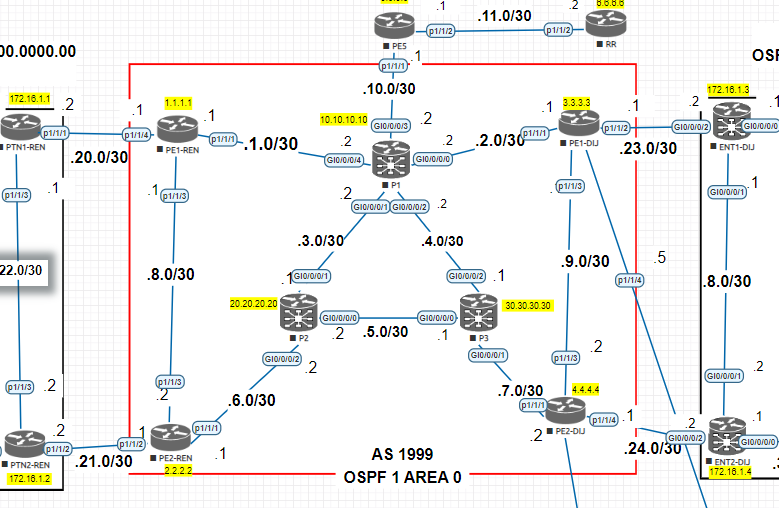


Figure 41 : L’architecture de la solution existante

Une architecture est proposée pour lier deux sites entre eux à travers un réseau Backbone qui contient plusieurs services les en situés par la suite les deux clients peuvent accéder à un centre de données, mais pour nous, nous restons focalisés sur le réseau de Backbone c’est notre champ de bataille. L’objectif maintenant c’est de configurer cette architecture par une configuration minimale qui permet que les deux sites puissent se communiquer entre eux à travers la Backbone cette opération nécessite la configuration des services suivante :

* + - Les interfaces
    - OSPF
    - I-BGP et E-BGP
    - MPLS
    - LDP
    - Les VRF

Pour la deuxième étape c’est l’implémentation des services VPN :

* + - VPRN
    - Epipe
    - VPLS

En fin, nous essayons d’augmenter les performances d’un flux échangé entre les deux clients, nous parlons de la qualité de services des flux qui travers notre Backbone.

Notre architecture existante contient 4 équipements PE et 3 équipements P et RR, c’est 7 équipements représente notre Backbone qui relié deux sites entre eux chaque site contient les services :

* + - Routage E-BGP.
    - Routage OSPF.
    - Le routage Inter-VLAN.
    - Les VLANs.

# Critique de l’architecture

La méthode traditionnelle de configuration manuelle des services et protocoles sur une architecture Backbone présente plusieurs inconvénients majeurs. Tout d'abord, elle nécessite un temps et un effort considérables, et est sujette aux erreurs. De plus, la maintenance continue peut devenir de plus en plus complexe à mesure que l'infrastructure évolue. Une architecture Backbone, par définition, comprend plusieurs équipements et protocoles, ce qui se traduit par un grand nombre de lignes de commande nécessaires. Par exemple, imaginons que notre architecture actuelle comporte 7 routeurs au niveau du Backbone et que nous devons configurer les services mentionnés précédemment. Rien que pour les interfaces, il faudrait saisir plus de 200 lignes de commande . Pour notre cas nous avons plusieurs services qui nécessite un ensemble de configuration ce qui demande de temps pour la configurer.



Figure 42: exemple d’une partie des configurations d’interface pour un routeur

Lorsque nous travaillons sur une architecture Backbone, il s'agit en réalité d'une topologie qui relie plusieurs sites entre eux. C'est un environnement complexe dans lequel une configuration simple peut engendrer des problématiques, notamment en ce qui concerne l'impact sur d'autres protocoles et services. Cela peut entraîner une véritable catastrophe. En outre, notre architecture actuelle n'est pas garantie en cas de panne de notre équipement P. Dans cette situation, il existe peu de chances de pouvoir récupérer la connexion, ce qui expose nos clients au risque de perdre leur communication entre eux.

Pour régler ce problème il faut ajouter des équipements qui vont offrir une redondance au niveau de notre architecture, la redondance est très importante car les problèmes de tomber une interface ou un équipement ShutDown est toujours possible, et si nous augmente le nombre des équipements nous augmentant le nombre des commandes de configuration et la possibilité d’une mal-configuration augmente aussi.

Les opérations et les demande qui peuvent donner à un ingénieur réseau n’est pas toujours une intégration ou bien une implémentation pour la première fois d’un service VPN, il se peut que le client veuille modifier ou bien supprimer carrément une configuration ou un service, seras parfois plus difficile de supprimer ou modifier une configuration. D’un autre côté tous les processus et les méthodologies de travail ne peuvent pas nous garantisse 100 %, le monde de réseau et très complexes il faut que toujours faire des opérations de vérifications à chaque fois et dans chaque étape c’est dernier peuvent prendre beaucoup de temps en termes de charge et de coût. Ainsi que parfois ils existent des opérations de configurations d’un nombre d’équipement qui atteint vers 60, 70 équipements ce qu’est très difficile pour un seul ingénieur de configurer, ça va prendre de temps.

# L'étude des solutions possible pour notre besoin : Chef, Puppet, Terraform, NSO .

## Terraform

C’est un utile d’infrastructure en tant que code qui nous permet de créer, modifier et de versionner l’infrastructure de manière sûre et efficace. Cela inclut des composants de bas niveau tels que les instances de calcul, le stockage et la mise en réseau, ainsi que des composant de haute niveau, Terraform peut gérer à la fois les fournisseurs de services existants et les solutions internes personnalisées. Il s’agit d’une plateforme open source conçue pour créer, modifier et versionner l’infrastructure de manière efficace et sécurisée.

Terraform est un outil polyvalent qui offre une prise en charge pour plusieurs fournisseurs de services cloud tels que AWS, Azure, Google Cloud, etc. Cela permet aux utilisateurs de gérer leur infrastructure de manière cohérente et portable, indépendamment du fournisseur de cloud choisi.

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Figure 43: L’architecture de fonctionnement de Terraform

Par contre, plusieurs Fonctions de collaboration et de sécurité disponibles uniquement dans des solutions entreprise onéreuses, c-à-d certaines fonctionnalités avancées de collaboration et de sécurité de Terraform peuvent être disponibles uniquement dans des solutions d'entreprise qui peuvent être coûteuses. Cela signifie que certaines équipes ou organisations peuvent ne pas avoir accès à toutes les fonctionnalités souhaitées, en particulier si elles utilisent la version open-source ou la version gratuite de Terraform.

## Puppet

Puppet est un outil d'automatisation d'infrastructure donc c’est un gestionnaire des configurations et d'automatisation comme Ansible qui peut être utilisé avec Linux et Windows. Puppet est réalisé en Ruby.

Avec Puppet, les tâches de configuration, de gestion et de provisionnement sont très simples à réaliser. Au lieu de décrire une suite d'actions à réaliser, comme avec les outils d'administration classiques, l'administrateur saisit l'état qu'il souhaite obtenir (permissions souhaitées, fichiers et logiciels à installer, configurations à appliquer). Puppet permet ainsi d'administrer un grand parc hétérogène de façon centralisée.

Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquement

Figure 44: L’architecture de fonctionnement de Puppet

Puppet peut générer des rapports sur ces packages, tant qu'un agent Puppet est installé sur le nœud hébergeant le package. Puppet Entreprise peuvent vous aider à suivre et à gérer les vulnérabilités des packages, même sur les nœuds où vous ne gérez pas le package vulnérable avec Puppet. Le serveur sur lequel Puppet Software est chargé s'appelle Puppet Master. Étant donné que Puppet utilise un modèle basé sur un agent, il existe également un agent Puppet sur le nœud géré.

## Chef

Chef est un outil d'automatisation d'infrastructure écrit en Ruby, dont le fonctionnement est analogue à Puppet. L’administrateur écrit des « recettes » qui décrivent l’état dans lequel doivent se trouver les nœuds administrés par Chef (déploiement de fichiers de configuration, installation de paquets, gestion de mots de passe, ...). Chef se charge alors d’appliquer les recettes sur les différents nœuds, permettant d’administrer de manière centralisée un parc hétérogène composé d’un grand nombre de machines.

Chef bénéficie d’une communauté d’utilisateurs plus jeune que Puppet mais semble toutefois très prometteur. De nombreuses idées introduites par Chef ont d’ailleurs été reprises par Puppet. Chef est distribué sous licence Apache. Chef Software comporte différents composants. Trois d'entre eux sont les principaux composants de Chef Software. Ces composants Chef sont :

* + - * Chef Serveur
      * Postes de travail
      * Nœuds

Une image contenant texte, capture d’écran, ordinateur portable, ordinateur

Description générée automatiquement

Figure 45 : L’architecture de fonctionnement de Chef

Chef Server est le centre des opérations de Chef. Il stocke, gère et fournit toutes les configurations requises aux composants Chef. Les postes de travail sont les appareils dont les codes, les configurations sont développées, créés, testés et modifiés. Il est configuré avec les outils Chef CLI. Nous pouvons tester des Recipe, des Cookbook avec des Workstations. Les nœuds sont les appareils gérés par Chef Software. Ces appareils peuvent être des appareils physiques, des appareils cloud, des appareils virtuels, etc. Sur ces appareils, Chef Client est installé. Chef Client est utilisé pour extraire les coockbokks du Chef Server.

## NSO

Un Orchestrateur de Services Réseau (Network Services Orchestrator, NSO) est un composant crucial dans la gestion et les opérations des réseaux modernes, en particulier dans des environnements réseau complexes, multi-fournisseurs et dynamiques. Les NSO automatisent le déploiement, la gestion et l'orchestration des services réseau, assurant qu'ils sont délivrés efficacement, de manière cohérente et avec une intervention manuelle minimale. Voici un aperçu détaillé des aspects clés d'un NSO :

Les avantages d'un Orchestrateur de Services Réseau (NSO) sont multiples et significatifs. Il améliore l'efficacité opérationnelle en automatisant les tâches répétitives, réduisant ainsi les efforts manuels et les erreurs humaines. Cela permet également une réponse plus rapide aux demandes de changement et aux incidents réseau. L'NSO accroît l'agilité et la flexibilité du réseau, permettant une adaptation rapide aux changements de la demande et facilitant l'intégration de nouvelles technologies. De plus, il offre une visibilité et un contrôle centralisés, améliorant la supervision et la gestion de la performance et de l'état des services réseau. En somme, un NSO optimise la gestion réseau, renforce l'efficacité opérationnelle et assure une haute qualité de service dans un environnement technologique en constante évolution.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 46 : Architecture NSO

# La solution

## Ansible est plus cohérent

Il existe de nombreux outils d'automatisation. Des outils propriétaires aux scripts python maison, il serait vain d'essayer de tous les lister. Puppet et Chef ont eu leur heure de gloire, mais ont l'inconvénient de nécessiter des agents sur les équipements managés (faute d'utiliser un protocole de configuration comme NETCONF). Depuis quelques années, Ansible tend à s'imposer comme outil Open Source d'automatisation. Sa capacité à pouvoir tourner sans nécessité d'agent, sa simplicité, sa richesse en « Network modules » supportés, et sa très large communauté d'utilisateurs font d'Ansible un outil particulièrement prisé. S'il est possible de pousser des configurations en CLI. Il existe différents cas d'utilisation du logiciel d'automatisation Ansible. Ces cas d'utilisation sont : Gestion de la configuration, la gestion du réseau, la sécurité, la livraison continue.... Alors travailler avec un outil comme Ansible, permet de structurer l’approche des spécialistes réseaux. L'architecte réseau doit définir ce qu'il attend d'Ansible comme actions de configuration. Cela se fait à partir de la création de « Playbooks », en utilisant un langage texte simple et lisible par un humain : YAML.

Ansible repose sur un nœud de contrôle (l'emplacement d'exécution) et plusieurs nœuds gérés (les points de terminaison à automatiser). Ces nœuds peuvent représenter des serveurs Linux, des serveurs Windows, des périphériques réseau, des environnements virtuels et des plateformes de cloud public et de conteneurs divers et variés. L'architecture sans agent permet à Ansible de communiquer avec de nombreux points de à terminaison, sans nécessiter l'installation d'une application ou d'un service sur le nœud géré. Il faut savoir que la plupart des commutateurs et routeurs réseau sont des systèmes fermés qui, bien souvent, ne peuvent pas héberger d'agents logiciels.

Ansible utilise une approche de programmation procédurale (ou impérative), qui tente de préserver la configuration d'une infrastructure informatique en définissant les étapes permettant d'atteindre l'état souhaité.

### Introduction générale

Pour remédier à ces problèmes, une solution d'automatisation basée sur Ansible et Python peut être mise en place. L'automatisation avec Ansible et Python offre plusieurs avantages. Tout d'abord, elle permet de réduire considérablement le temps et les efforts nécessaires pour configurer les services et protocoles sur le Backbone. Les tâches fastidieuses de configuration manuelle peuvent être automatisées, ce qui libère du temps pour se concentrer sur d'autres aspects importants du réseau.

Réduction des erreurs : Une configuration manuelle est sujette aux erreurs humaines, telles que des fautes de frappe, des oublis ou des configurations incorrectes. L'automatisation avec des outils tels qu'Ansible et Python permet de minimiser ces erreurs en utilisant des Playbooks ou des scripts bien définis. Ces outils peuvent également effectuer des vérifications et des validations automatiques pour s'assurer que les configurations sont cohérentes et correctes.

Facilité de maintenance : Une fois que les services et protocoles sont configurés automatiquement, la maintenance continue devient plus facile. Les modifications, les ajouts ou les suppressions de services peuvent être effectués de manière centralisée en modifiant simplement les Playbooks ou les scripts correspondants. Cela garantit une cohérence dans l'ensemble de l'infrastructure et réduit le risque d'erreurs lors de la mise à jour des configurations.

Gestion centralisée : L'automatisation permet également une gestion centralisée de l'infrastructure. Les Playbooks ou les scripts peuvent utiliser des variables pour définir des paramètres communs à plusieurs équipements, ce qui facilite la gestion cohérente de l'ensemble du Backbone. Les mises à jour et les modifications peuvent être effectuées à partir d'un emplacement central, assurant ainsi la cohérence et la facilité de gestion.

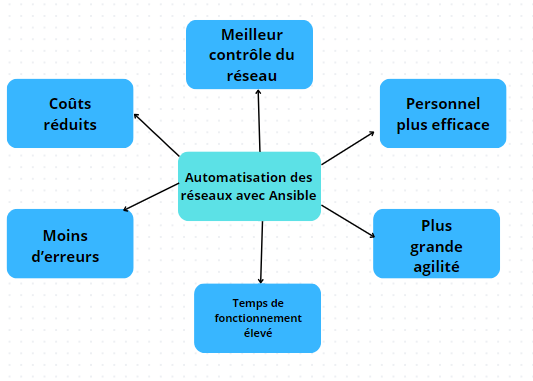


Figure 47: Les avantages de l’automatisation des réseaux

L'automatisation avec Ansible et Python offre une solution efficace pour remplacer la configuration manuelle des services et protocoles sur une architecture de Backbone. Elle permet de réduire les erreurs humaines, de simplifier la maintenance continue et de faciliter la gestion centralisée de l'infrastructure. En adoptant cette approche, vous pouvez économiser du temps et des efforts tout en assurant une configuration cohérente et précise de votre Backbone.

### Ansible est un bon outil pour l’automatisation

Tout d'abord, Ansible a une architecture Agentless. Ils n'ont besoin d'aucun agent sur le périphérique géré à distance. Avec cette architecture, de nombreux appareils Cisco prend en charge Ansible. Chef a une architecture basée sur les agents tandis que Puppet a les deux.

Ansible est gratuit et open source C'est un simple logiciel d'automatisation Ansible est écrit par Python. Ansible utilise YAML très simple Il modélise l'ensemble de votre infrastructure en décrivant l'interrelation de vos systèmes alors il utilise YAML comme extension. Il utilise SSL pour connecter les serveurs. Ansible est sans agent contrairement à Puppet et Chef. Vous n'avez rien à installer sur l'appareil distant.

Parmi les points les plus fort pour Ansible c’est qu’il utilise le mécanisme Push tandis que Chef et Puppet utilisent le mécanisme Pull, ce qui donne Ansible une exécution immédiate. En fin si en compare le processus d’installation et la configuration initiale pour les trois outils Ansible reste le meilleur du côté de la facilité. Ansible reste un outil plus performant sans besoins des agents et facile à installés, a une grande communauté qui peuvent de nous donner un coup de main si jamais nous avons des problèmes pour nous donne des propositions, d’une autre côte il utilise YAML simple et facile à comprendre, et plus d’autre avantages qu’ils ont favoriser cet outil de l’utiliser dans notre projet.

Chapitre 4 : La Réalisation

# Introduction

Ce chapitre explore les différentes étapes de réalisation de notre projet, de la configuration de l'architecture Backbone MPLS/IP dans EVE-NG à l'automatisation de cette configuration à l'aide d'Ansible. Nous mettrons également l'accent sur l'utilisation de Python pour la collecte des données des équipements et leur exploitation ultérieure. L'objectif final est de démontrer l'efficacité et les avantages de l'automatisation dans la gestion d'une infrastructure réseau complexe, tout en facilitant la maintenance et en améliorant la performance globale

# Logiciel et les équipements utilisé

## Eve- NG

Pour représenter une topologie plus réaliste et simuler les différents protocoles, ainsi que leur impact sur les autres, il est essentiel d'avoir recours à un simulateur performant. C'est pourquoi nous avons choisi d'utiliser EVE-NG (Emulated Virtual Environment - Next Generation), un logiciel de simulation réseau avancé largement utilisé par les professionnels des technologies de l'information et des réseaux. EVE-NG nous permettra de créer des environnements virtuels qui reproduiront fidèlement les configurations et les interactions réseau, offrant ainsi une plateforme idéale pour étudier et tester les protocoles.

Une image contenant Police, Graphique, logo, texte

Description générée automatiquement

Figure 48:EVE-NG

EVE-NG est un logiciel de simulation réseau avancé utilisé par les professionnels des technologies de l'information et des réseaux. Il permet de créer des topologies de réseau complexes en utilisant des machines virtuelles pour simuler différents équipements réseau tels que des routeurs, des commutateurs et des pares-feux.

## VMWare Workstation

Le logiciel EVE-NG est une machine virtuelle utilisée pour la simulation réseau. Pour exécuter EVE-NG, nous utiliserons le logiciel de virtualisation VMWare, qui nous permettra de créer et gérer des machines virtuelles sur notre ordinateur personnel. VMWare offre également des outils de mise en réseau pour connecter les machines virtuelles entre elles et avec le réseau de l'ordinateur hôte. En combinant EVE-NG et VMWare, nous pourrons créer un environnement de simulation réseau réaliste et complet.

Une image contenant Graphique, logo, symbole, Police

Description générée automatiquement

Figure 49: VM WARE

## Ansible

Ansible est un outil d'automatisation et de gestion de configuration largement utilisé dans le domaine de l'informatique, c’est un outil open source. Il permet de déployer, de configurer et de gérer des systèmes informatiques de manière efficace et reproductible. Nous allons utilisée la version [core 2.11.0], qui fonctionne très bien avec la version de Python 3.10.6 et la version Jinja 3.0.3.

Une image contenant Police, logo, symbole, Graphique

Description générée automatiquement

Figure 50: la solution Ansible

## Visual studio Code

Visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft, disponible sur plusieurs plateformes (Windows, MacOs et Linux). Il est largement utilisé par les développeurs pour écrire du code dans différents langages de programmation.

Une image contenant capture d’écran, Graphique, symbole, Bleu électrique

Description générée automatiquement

Figure 51:logiciel Vs code

## SecureCRT

SecureCRT est un client de terminal et un émulateur de terminal développé par VanDyke Software. Il est couramment utilisé pour l'administration des réseaux et la gestion des systèmes grâce à ses fonctionnalités avancées pour la connectivité sécurisée.

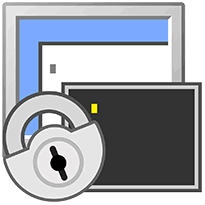


Figure 52 : secureCRT

# Les équipements utilisés

## NOKIA VSR 7750

Cette technologie révolutionnaire offre des performances, une évolutivité et une polyvalence pour toutes les applications IP, qu'elles soient périphériques ou centrales. Les fournisseurs de services de communication, ainsi que les opérateurs de réseaux cloud publics et privés du monde entier, font confiance à cette technologie novatrice pour alimenter leurs réseaux et services essentiels.

Une image contenant intérieur, conteneur, boîte

Description générée automatiquement

Figure 53: L’équipement NOKIA VSR 7750

## CISCO XRv

IOS XR est un système d'exploitation réseau développé par Cisco pour ses routeurs de grande envergure. Il offre une architecture modulaire et distribuée, assurant une haute disponibilité et une scalabilité optimale. IOS XR est connu pour sa stabilité, sa résilience et sa capacité à gérer des volumes de trafic importants. Il prend en charge des fonctionnalités avancées de routage, de commutation et de sécurité, ainsi que des protocoles tels que OSPF, BGP, IS-IS et MPLS. Grâce à son architecture modulaire, les mises à jour peuvent être effectuées sans interruption du service, garantissant une disponibilité continue du réseau. IOS XR offre également des fonctionnalités de gestion avancées, une virtualisation et une sécurité renforcée.

Une image contenant texte, conception

Description générée automatiquement

Figure 54: L’équipement CISCO XRv

# Mise en œuvre de la topologie réseau (la maquette du backbone)

## Création de la maquette

Une image contenant texte, capture d’écran, pixel

Description générée automatiquement

Figure 55: Architecture du projet

## Adressage de chaque routeur

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Router | Loopback/M | IP Adresse | Mask | Interface | Description |
| P1 | 1.1.1.1/32 | 10.1.2.1  10.1.1.2  10.1.10.1  10.1.6.2 | 255.255.255.252  255.255.255.252  255.255.255.252  255.255.255.252 | G0/0/0/0 G0/0/0/1 G0/0/0/2  G0/0/0/3 | TO P1-P2 TO P1-P3  TO P1-PE1REN  TO P1-PE1BDX |
| P2 | 2.2.2.2/32 | 10.1.2.2  10.1.7.2  10.1.5.2  10.1.3.1 | 255.255.255.252  255.255.255.252  255.255.255.252  255.255.255.252 | G0/0/0/0 G0/0/0/2 G0/0/0/3  G0/0/0/4 | TO P2-P1 TO P2-PE1DIJ TO P2-PE2BDX  TO P2-P3 |
| P3 | 3.3.3.3/32 | 10.1.1.1  10.1.3.2  10.1.8.2  10.1.9.2 | 255.255.255.252  255.255.255.252  255.255.255.252  255.255.255.252 | G0/0/0/0 G0/0/0/1 G0/0/0/2  G0/0/0/3 | TO P3-P1 TO P3-P2  TO P3-PE2DIJ  TO P3-PE2REN |
| PE1-REN | 9.9.9.9/32 | 10.1.10.2  10.1.23.2  10.1.22.1 | 255.255.255.252  255.255.255.252  255.255.255.252 | P1/1/1 P1/1/2  P1/1/3 | TO PE1REN-P1  TO PE1REN-PTN1REN TO PE1REN-PE2REN |
| PE2-REN | 8.8.8.8/32 | 10.1.9.1  10.1.24.1  10.1.22.2 | 255.255.255.252  255.255.255.252  255.255.255.252 | P1/1/1 P1/1/2  P1/1/3 | TO PE2REN-P3  TO PE2REN-PTN2REN TO PE2REN-PE1REN |
| PE1-DIJ | 6.6.6.6/32 | 10.1.17.1  10.1.16.2 | 255.255.255.252  255.255.255.252 | P1/1/1  P1/1/2 | TO PE1DIJ-ENT1DIJ  TO PE1DIJ-PE2DIJ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 10.1.7.1 | 255.255.255.252 | P1/1/3 | TO PE1DIJ-P2 |
| PE2-DIJ | 7.7.7.7/32 | 10.1.16.1  10.1.18.1  10.1.8.1 | 255.255.255.252  255.255.255.252  255.255.255.252 | P1/1/1 P1/1/2  P1/1/3 | TO PE2DIJ-PE1DIJ TO PE2DIJ-ENT2DIJ  TO PE2DIJ-P3 |
| RR | 10.10.10.10/32 | 10.1.14.1  10.1.15.2 | 255.255.255.252  255.255.255.252 | P1/1/1  P1/1/2 | TO RR-PE1BDX  TO RR-PE2BDX |
| PTN1-REN | 13.13.13.13/32 | 10.1.23.1  10.1.25.1  10.1.26.1  10.1. | 255.255.255.252  255.255.255.252  255.255.255.252  255.255.255.252 | P1/1/1 P1/1/2 P1/1/3  P1/1/4 | TO PTN1REN-PE1REN TO PTN1REN-PTN2REN TO PTN1REN-PTN3REN  TO PTN1REN-VPLS |
| PTN2-REN | 12.12.12.12/32 | 10.1.24.2  10.1.25.2  10.1.27.1  10.1 | 255.255.255.252  255.255.255.252  255.255.255.252  255.255.255.252 | P1/1/1 P1/1/2 P1/1/3  P1/1/4 | TO PTN2REN-PE2REN TO PTN2REN-PTN1REN TO PTN2REN-PTN3REN  TO PTN2REN-EPIPE |
| PTN3-REN | 14.14.14.14/32 | 10.1.26.2  10.1.27.2 | 255.255.255.252  255.255.255.252 | P1/1/1  P1/1/2 | TO PTN3REN-PTN1REN  TO PTN3REN-PTN2REN |
| ENT1-DIJ | 15.15.15.15/32 | 10.1.19.2  10.1.20.1  10.1.17.2  10.1 | 255.255.255.252  255.255.255.252  255.255.255.252  255.255.255.252 | G0/0/0/0 G0/0/0/1 G0/0/0/2  G0/0/0/3 | TO ENT1DIJ-ENT2DIJ TO ENT1DIJ-ENT3DIJ TO ENT1DIJ-PE1DIJ  TO ENT1DIJ-VPLS |
| ENT2-DIJ | 16.16.16.16/32 | 10.1.19.1  10.1.21.1  10.1.18.2  10.1 | 255.255.255.252  255.255.255.252  255.255.255.252  255.255.255.252 | G0/0/0/0 G0/0/0/1 G0/0/0/2  G0/0/0/3 | TO ENT2DIJ-ENT1DIJ TO ENT2DIJ-ENT3DIJ TO ENT2DIJ-PE2DIJ  TO ENT2DIJ-EPIPE |
| ENT3-DIJ | 17.17.17.17/32 | 10.1.20.2  10.1.21.2 | 255.255.255.252  255.255.255.252 | G0/0/0/0  G0/0/0/1 | TO ENT3DIJ-ENT1DIJ  TO ENT3DIJ-ENT2DIJ |

# Méthodologie d’approche

Une méthodologie d’approche pour implémenter notre architecture backbone en trois zones à base de MPLS a été développée en suivant les étapes :

➢ Réseaux Transport

➢ Réseaux Mobile

➢ Réseaux Entreprise

Alors on commence de faire la configuration qui se répétant sur les trois réseaux.

## Configuration des interfaces

Commençons tout de suite par configurer les interfaces et le réseau de l’opérateur, et en même temps nous allons configure une liaison virtuelle entre les PE et le CE, et puisque nous avons deux types de routeur (Nokia et Cisco) alors on va faire de configurer les liaison point to point sur les 2 type.

• Configuration des interfaces sur Cisco :

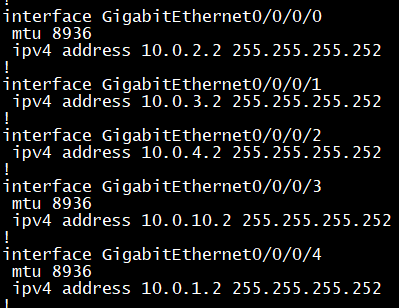


Figure 56 : configuration des interfaces sur Cisco

Configuration des interfaces sur Nokia :

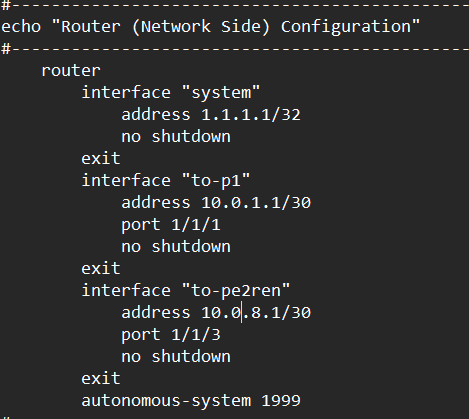


Figure 57: configuration des interfaces sur Nokia

Cette configuration sera appliquée séparément à chaque interface du routeur comme dans cette architecture, afin de respecter le choix de l'adressage et de pouvoir s'assurer que l'adressage a été fait correctement.

**Résultat :**

* Test des interfaces sur Cisco

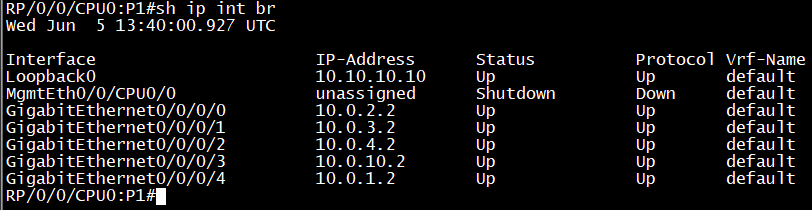


Figure 58 : test des interfaces sur Cisco

* Test des interfaces sur Nokia

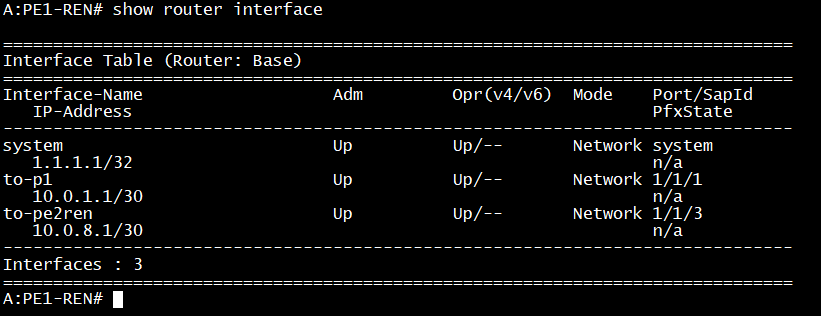


Figure 59 : test des interfaces sur Nokia

## Réseau de transport

### Mise en place du protocole OSPF

Dans cette section, nous activons l’OSPF pour garantir la communication des routeurs P et PE.

• Configuration de protocole de routage sur Cisco

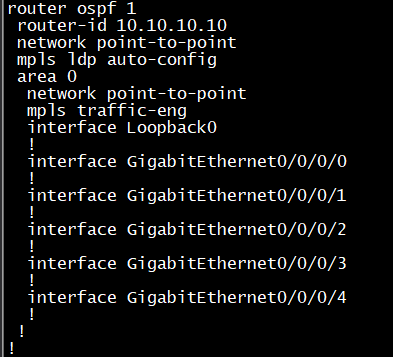


Figure 60 : configuration de OSPF sur Cisco

• Configuration de protocole de routage sur Nokia

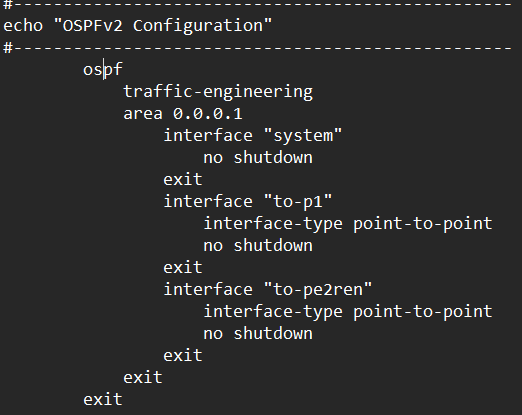


Figure 61 : configuration de OSPF sur Nokia

Donc on appliquera le protocole OSPF au sein des interfaces respectives se trouvant au Cœur du réseau, Le protocole de routage interne au backbone sera actif seulement sur le backbone de réseau Résultat :

* Test de protocole OSPF sur Cisco

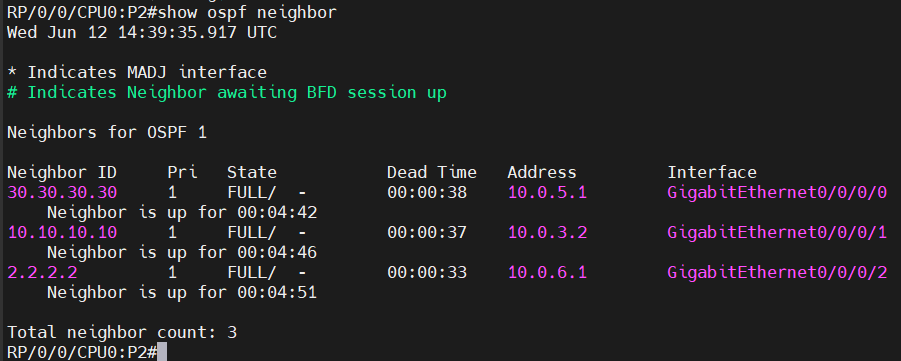


Figure 62 : test OSPF sur Cisco

* Test de protocole OSPF sur Nokia

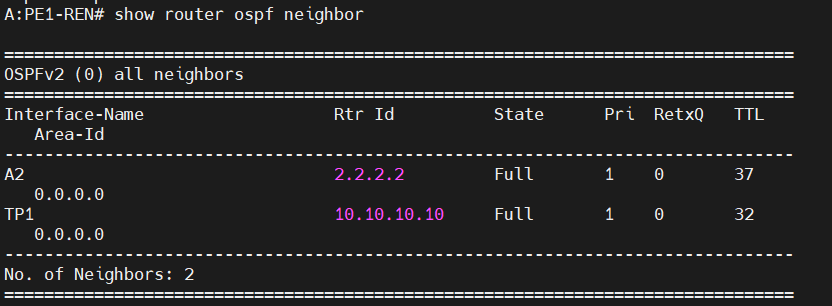


Figure 63 : test OSPF sur Nokia

### Activation du protocole MPLS et LDP et RSVP sur les routeurs du cœur de réseau :

L'implémentation du protocole MPLS sur les routeurs Cisco est très simple. La création de la table d'acheminement se fait dynamiquement grâce au protocole de distribution d'étiquettes. Protocole propriétaire Cisco TDP (Label Distribution Protocol) utilisé. Pour notre réseau, nous utilisons LDP (Label Distribution Protocol).

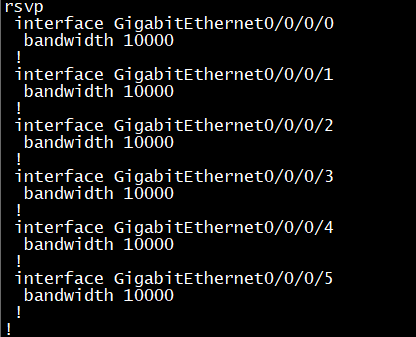
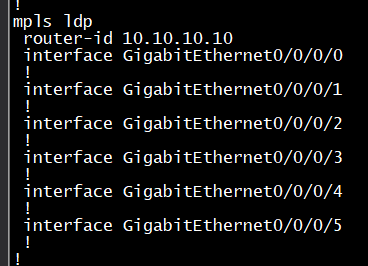
 • Configuration des trois protocoles sur Cisco

Figure 64 : configuration RSVP sur Cisco

Figure 65 : configuration LDP sur Cisco

Avant d'envoyer des paquets sur un réseau MPLS, l'interface doit encapsuler les paquets. Cette commande doit être ajoutée à la configuration de chaque interface cœur de réseau.

• Configuration des trois protocoles sur Nokia

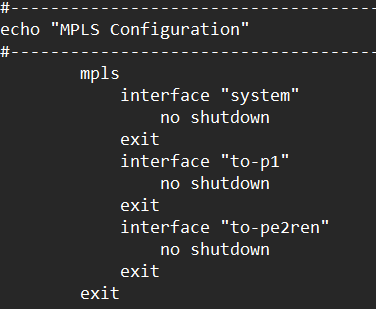


Figure 66 : configuration MPLS sur Nokia

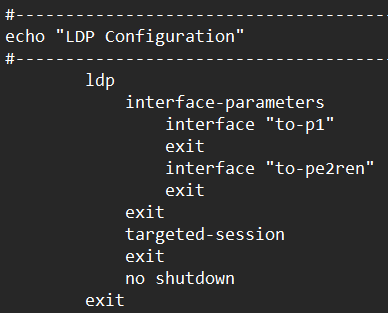


Figure 67 : configuration LDP sur Nokia

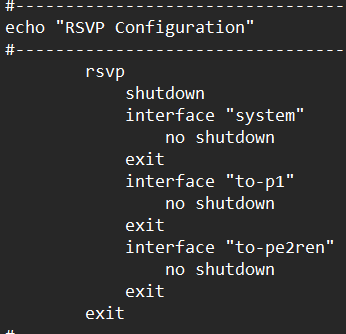


Figure 68: configuration RSVP sur Nokia

Sur le routeur Nokia en entre sur chaque interface de routeur et on active le MPLS et LDP et RSVP. Cette configuration n'est pas utilisée pour les interfaces des LER connectées au réseaux entreprise et réseaux mobile, car ces machines utilisent des paquets IP standards. Donc on applique ses commandes au sein de réseaux backbone seulement

**Résultat :**

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 69: test des interfaces de RSVP sur cisco

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 70 : test mpls par ping sur CISCO

### Mise en place du protocole MP-BGP à base de RR

Pour configurer la liaison vpnv4 entre les PE que l’on recherche à faire, il nous faut configurer sur les routeurs, comme on le ferait en BGP, une relation de voisinage en prenant comme référence les IPs de loopback. Alors on prendre adresse système de routeur réflecteur, La configuration il est répété sur tous les PE et les P de réseaux backbone :

• Exemple de configuration sur PE de Nokia

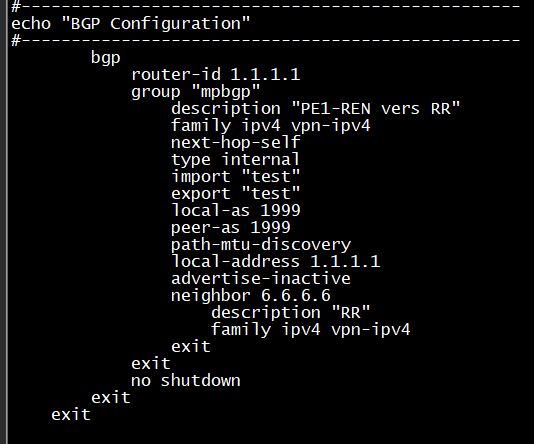


Figure 71 : configuration de IBGP ENTRE PE ET RR

* **Résultat :**

• Test de connectivité d’un PE et le routeur réflecteur RR

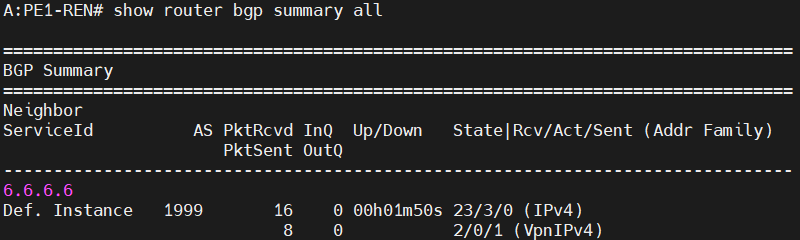


Figure 72: test de BGP SUMMARY entre PE ET RR

* Test de connectivité entre le routeur réflecteur RR et les équipements PE et P

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 73 : test de connectivité entre le RR ET les PE

## Réseau mobile

### Mise en place du protocole CE-PE

La configuration dédiée aux CE est très simple, du fait que cet équipement n’a aucune notion de MPLS, il va juste établir une adjacence avec le PE auquel il est relié et partager ses routes avec celui-ci, donc on va appliquer ISIS sur le réseau mobile base a des équipement Nokia

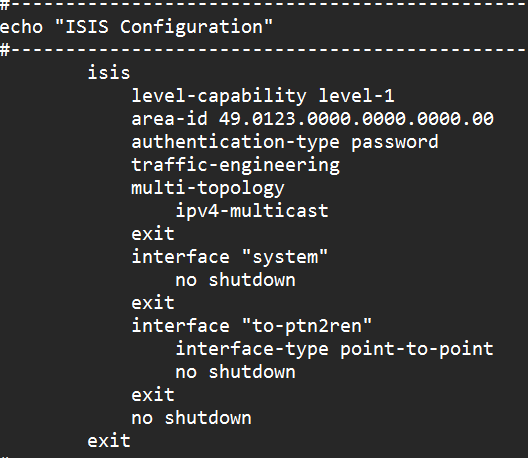


Figure 74 : configuration ISIS sur le site MOBILE

* **Résultat :**

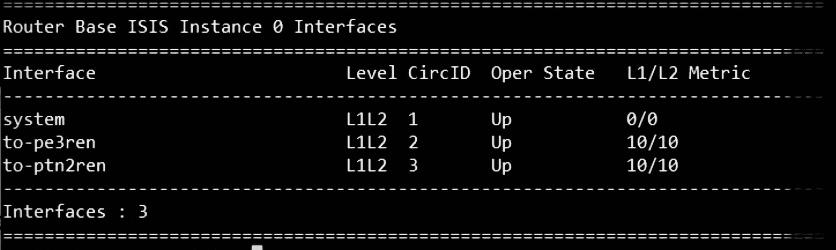


Figure 75 : test de protocole ISIS

### Configuration de INTERNAL BGP entre les PTNs

Alors là j’ai défini à l’intérieur de group IBGP qui comprend tous les routeurs de réseau mobile, et j’ai déclaré les adresse système de chaque routeur comme voisin

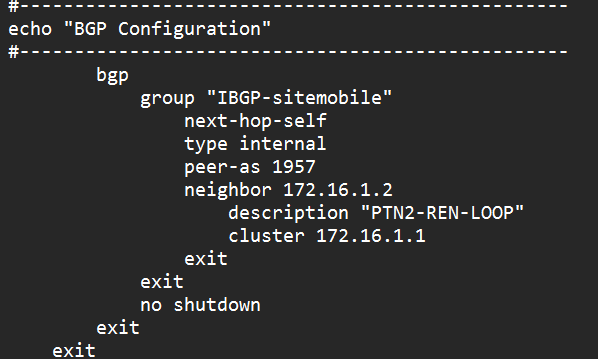


Figure 76 : configuration de IBGP sur le site mobile

### Gestion de la redistribution respective des préfixes sur le routeur PTN1-REN

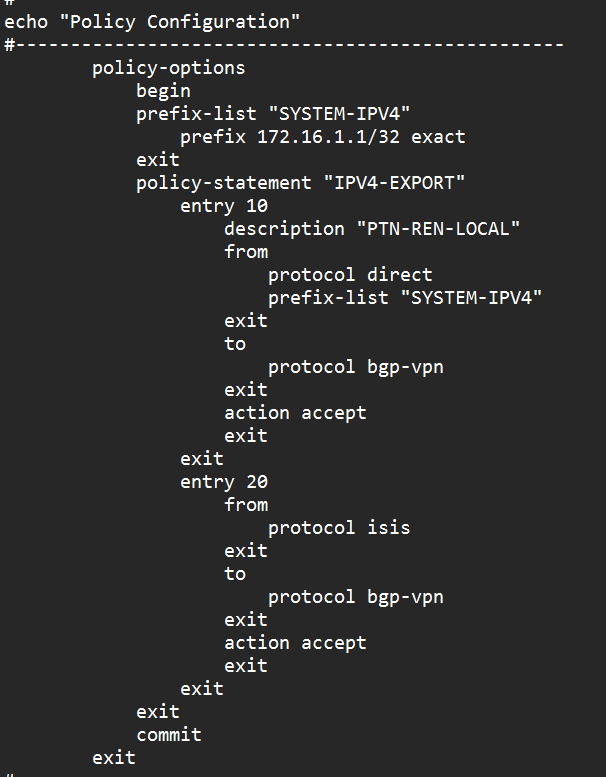


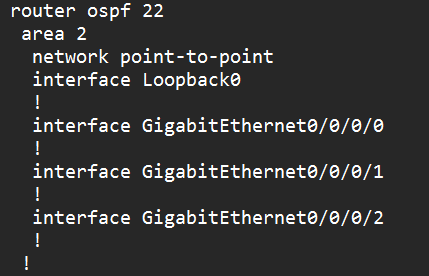
Figure 77 : configuration de POLICY option sure la zone mobile

## Réseaux entreprise

### Mise en place du protocole CE-PE

On va appliquer OSPF sur le réseau entreprise basé sur des équipement Cisco

Figure 78 : configuration de OSPF sur le site entreprise



### Configuration de IBGP entre les ENT de Réseaux entreprise

Alors là j’ai défini à l’intérieur de group IBGP qui comprend tous les routeurs de réseau entreprise, et j’ai déclaré les adresse système de chaque routeur comme voisin, et pour la redistribution de protocole utiliser à l’intérieur de réseaux entreprise tu fais seulement la commande redistribue connected

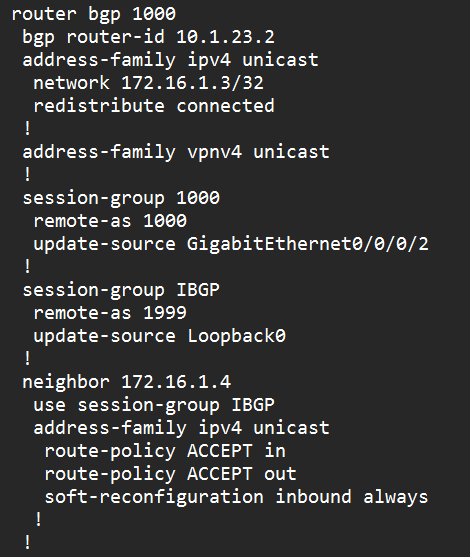


Figure 79 : configuration de IBGP sur le site entreprise

## Configuration des services : VPRN – EPIPE – VPLS (9)

Dans cette partie, nous nous concentrerons sur la configuration des services sur interface point à point des routeur PEs et Ces sur les deux sites d’opérateur. Alors on commence de configurer interface entre le PE et CE afin de pouvoir initialiser tous les services sur un seul interface.

### Configurer les interfaces entre PE et CE

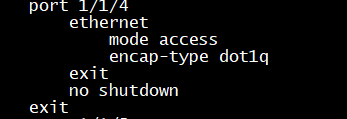


Figure 80 : configuration des interfaces entre le PE et le CE

Cette commande est pour chaque port du PE liée directement au CE.

### Service VPRN

Avant d’appliquer le service sur le routeur tu dois créer des clients pour affecter aux services, alors un client pour chaque service

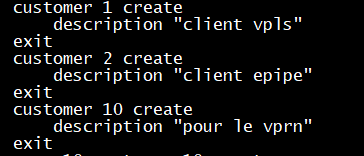


Figure 81 : création des clients

• Exemple sur le coté de réseaux mobile

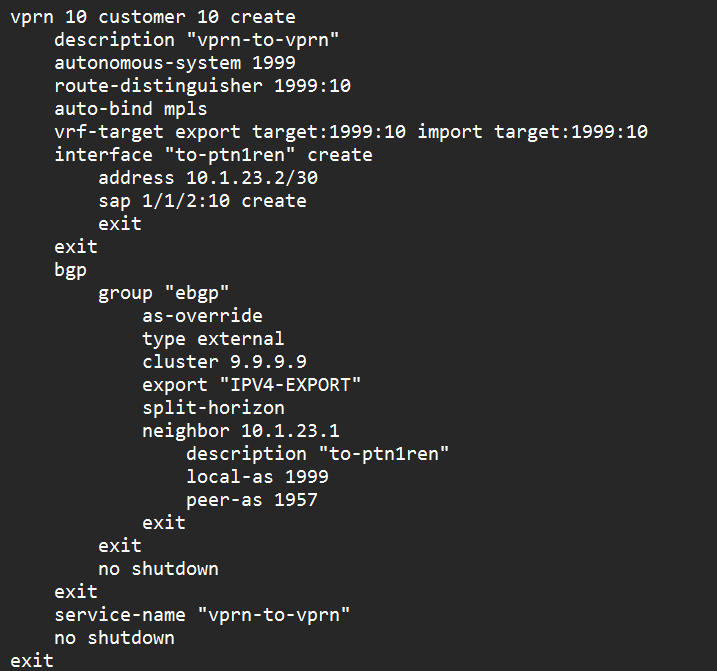


Figure 82 : configuration de service VPRN sur le site mobile

Alors à l’intérieur de ce service là j’ai donné une interface virtuelle, et j’ai affecter aussi le protocole BGP et à l’intérieur de ce protocole j’ai déclaré le voisin qui relie avec le réseau mobile.

• Exemple sur le coté de réseaux entreprise

Nous ferons ce que nous avons fait dans le réseau mobile sur le PE1-REN mais ce sera un simple changement sur le réseau entreprise parce qu’en ont un routeur Cisco qui dépend de VRF et dans un premier temps en va crier d’abord une interface virtuelle avant d’applique le BGP sur VRF

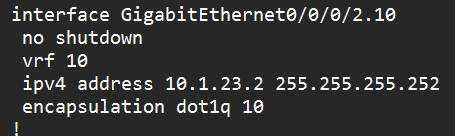


Figure 83 : création de sous interface virtuelle

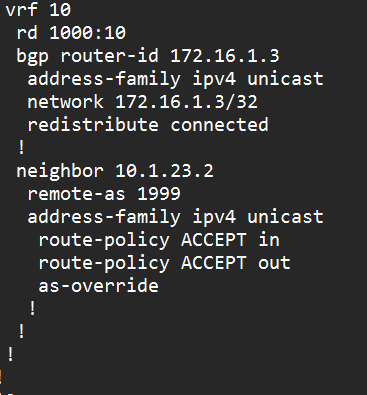


Figure 84 : configuration de VRF sur le site entreprise

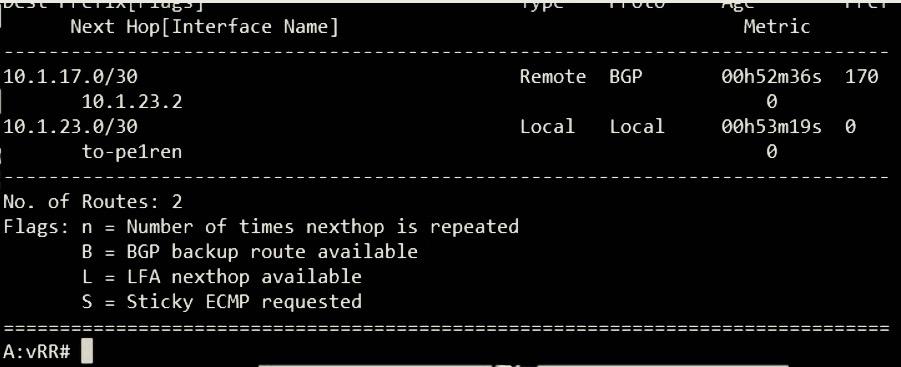


Figure 85 : le test de service VPRN

### Service VPLS

Avant appliquer le service VPWS il faut créer des clients comme nous l’avons fait dans l’exemple de service VPRN et puis nous avons créé un SDP entre le PE1-REN et PE1- DIJ et PE2-DIJ pour la négociation

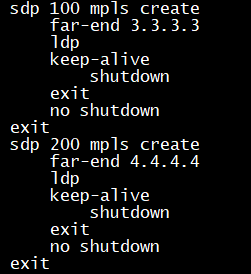


Figure 86 : configuration de SDP

Un SDP MESH est un type de SDP à prévention de boucle et est utilisé dans les services VPLS.

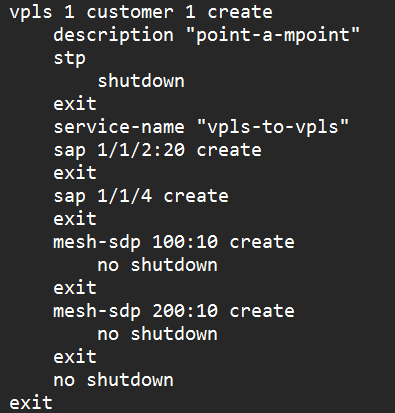


Figure 87 : configuration de service VPLS

**Résultat** :

Après la configuration de SDP et le VPLS nous aurons un lient entre le client des routeur PE qu’on a affecté le MESH SDP

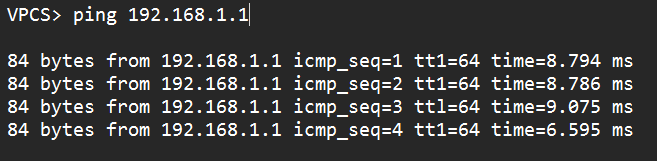


Figure 88 : test par ping de service VPLS

### Service EPIPE (VPWS)

La même chose pour le service EPIPE on va crier des SDP, mais la déférence avec le VPLS c’est on n’a pas de MESH SDP, donc on utilise le SPOKE-SDP est le seul que vous pouvez configurer sur un E-PIPE.

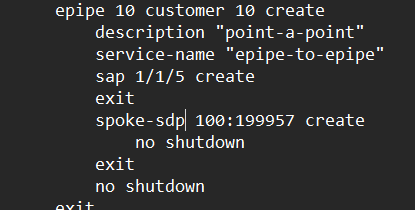


Figure 89 : configuration de service EPIPE

* **Résultat :**

Le service EPIPE il accepte juste la liaison point to point

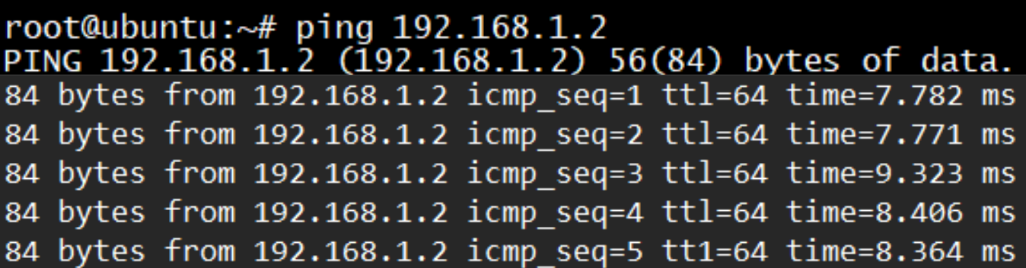


Figure 90 : test par ping de service EPIPE

# Automatisation de configuration et le test

## Configuration du fichier d’inventaire

Le fichier d’inventaire contient des informations sur les hôtes que vous gérerez avec Ansible. Vous pouvez inclure entre un et plusieurs centaines de serveurs dans votre fichier d’inventaire, et les hôtes peuvent être organisés en groupes et sous-groupes. Le fichier d’inventaire est également souvent utilisé pour définir des variables qui ne seront valables que pour des hôtes ou des groupes spécifiques, afin d’être utilisées dans les modèles de développement et les modèles. Certaines variables peuvent également affecter la façon dont un modèle de développement est exécuté, comme la variable ansible\_python\_interpreter que nous allons voir dans un instant.

Pour modifier le contenu de votre inventaire Ansible par défaut, ouvrez le fichier/etc/hosts en utilisant l’éditeur

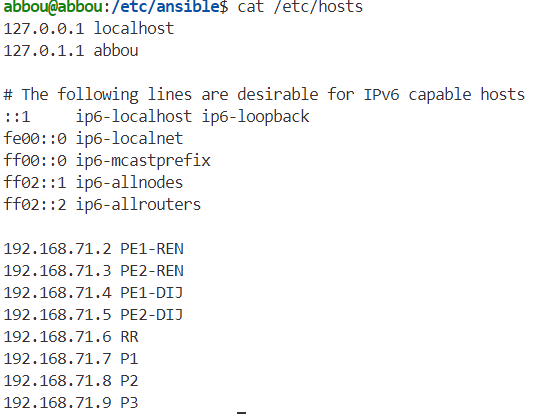


Figure 91 : le fichier /etc/hosts

## Test de connexion

Après avoir configuré le fichier d'inventaire pour inclure vos serveurs, il est temps de vérifier si Ansible peut se connecter à ces serveurs et exécuter des commandes via SSH.

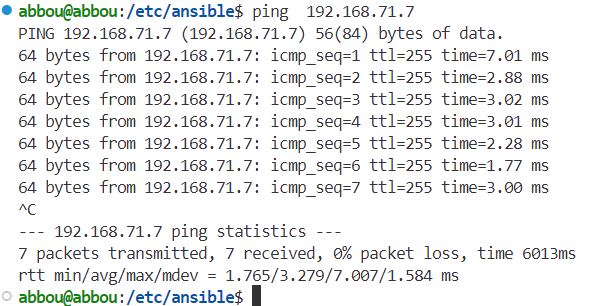


Figure 92 : test de connectivité entre ansible et un routeur

## Connectivite de serveur ansible avec le LAB via SSH

Alors pour accéder à la machine via SSH il faut faire des commandes pour l’autorisation de serveur de connecter à la machine

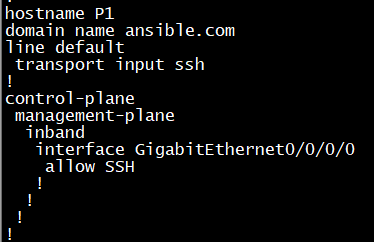


Figure 93: configuration ssh sur le routeur

Après avoir configuré le SSH sur équipement pour inclure vos serveurs, donc on test maintenant est ce que le serveur il est capable de connecter sur l’équipement

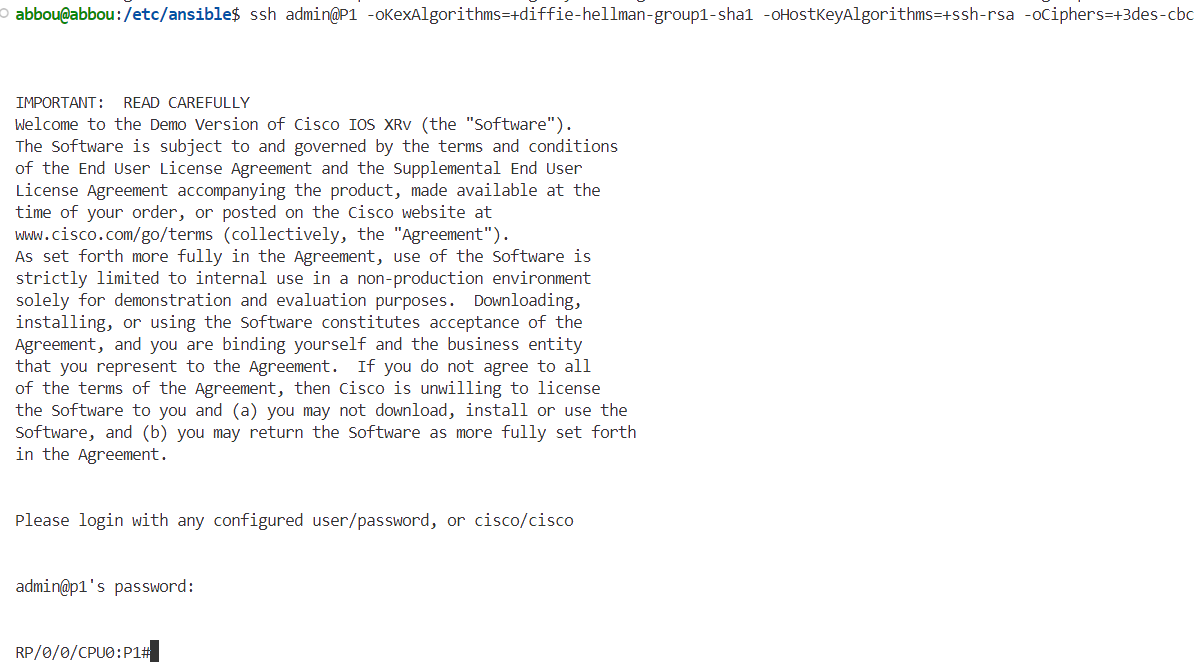


Figure 94 : test de connectivité par SSH

## Création des rôles

Ansible recherche les rôles dans un sous-répertoire nommé rôles dans le répertoire du projet. Cela peut être remplacé dans la configuration Ansible. Chaque personne a son propre répertoire. Pour faciliter la création de nouveaux rôles, l'outil ansible-galaxy init "non-roles" peut être utilisé.

La figure suivante présente quelque rôles que j’ai créé :

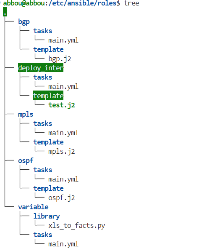


Figure 95 : création des rôles

## Création des taches

Après au répertoire roles/deploy-interface. Editez le fichier tasks/main.yml :

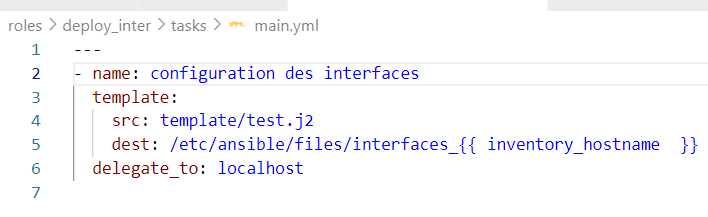


Figure 96 : contenu de fichier roles/deploy-interface /task

La dernière tâche que nous ajoutons utilise le module Template pour créer le fichier de configuration deploy-inetrafce à partir d’un modèle j2 ça sur la src et sur le dest j’ai défini un fichier pour stocker toute la config que j’ai définie sur le fichier j2

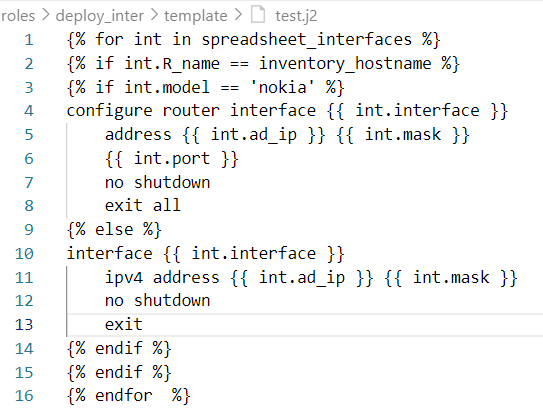


Figure 97 : contenu de fichier jinja2

Donc nous ferons de même pour les autres rôles

## Extraction des données

Le fichier xlsx qui contient tous les informations des routeurs, c’est données vont être extrait par un script python xlsx\_to\_fact.py sous forme d’un dictionnaire (clés :valeur).

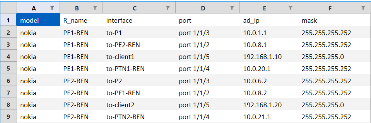


Figure 98 : contenu de fichier excel

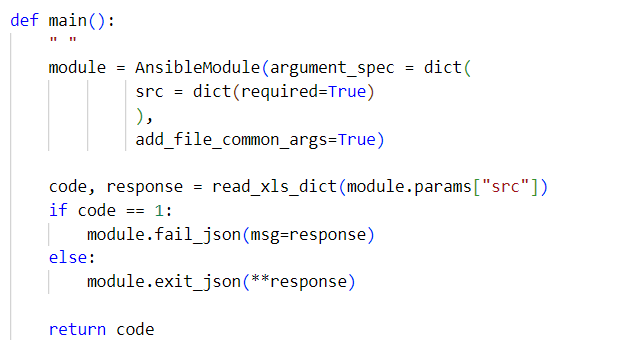


Figure 99 : Script xls\_TO\_FACT

## Création de PLAYBOOK :

Le Playbook va contenir le groupe d’host concerné et les roles qu’on va utiliser et les tasks qui vont faire le push des fichiers de configuration créé par les roles.

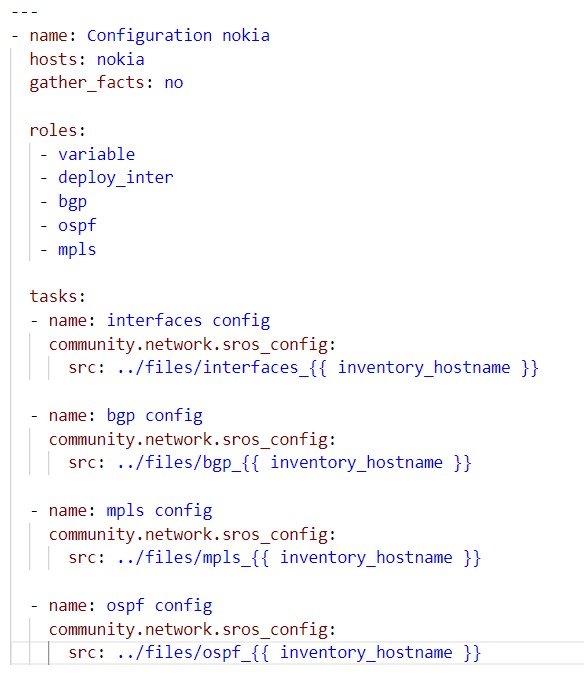


Figure 100 : création d'un PLAYBOOK

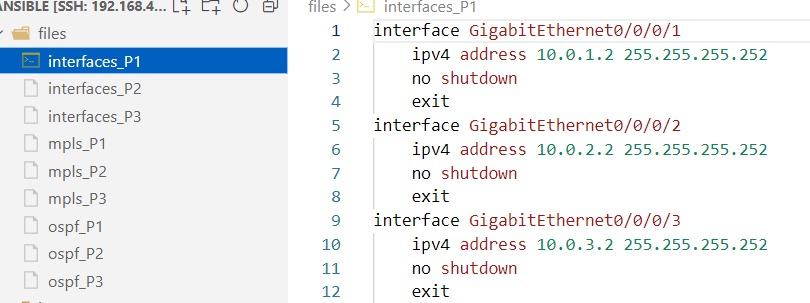


Figure 101: Fichier de configuration

Et pour lancer le PLAYBOOK il suffit d’exécuter une commande comme dans la figure suivante :

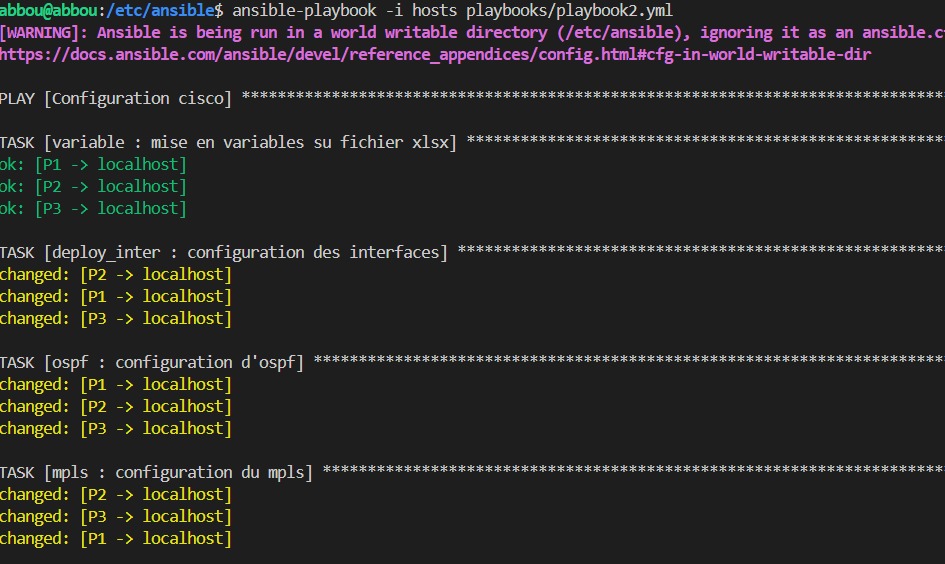


Figure 102 : lancement de PLAYBOOK

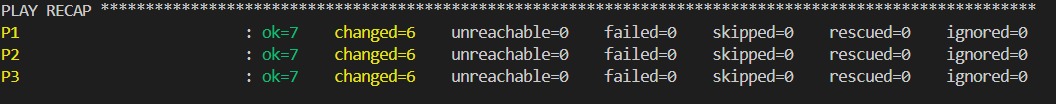


Figure 103:Réussite du push

## Automatisation de test

* **Création d’un rôle check conf**

Pour le faire le test on va utiliser un rôle qui va contenir des lignes de commandes qui vont être exécuté sur les équipement pour faire le test de configuration.

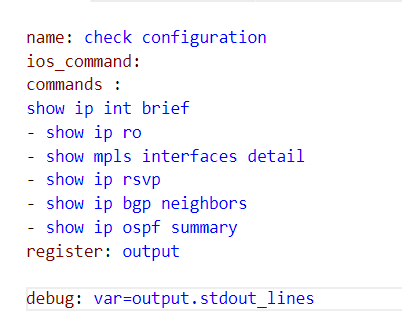


Figure 104 : rôle check-conf

**Conclusion générale**

En conclusion, mon projet se concentre sur l'automatisation des configurations et des services au sein de notre architecture Backbone MPLS/IP. Nous avons choisi Ansible comme outil pour automatiser les tâches de configuration, tandis que Python sera utilisé pour collecter les données relatives à la qualité de service (QoS). L'objectif principal de ce projet est de simplifier et d'accélérer les opérations de configuration tout en garantissant une collecte efficace des données de QoS. En automatisant les tâches de configuration, nous visons à réduire les erreurs humaines et à optimiser les performances du réseau.

En utilisant Ansible, nous serons en mesure de déployer rapidement et de manière cohérente les configurations requises pour notre infrastructure MPLS/IP. Cela nous permettra de gagner du temps et d'améliorer l'efficacité opérationnelle. Parallèlement, l'utilisation de Python pour collecter les données de QoS nous permettra d'obtenir une visibilité approfondie sur les performances du réseau. Nous pourrons ainsi analyser et optimiser les paramètres de QoS pour garantir une expérience utilisateur optimale.

En résumé, notre projet vise à automatiser les configurations et les services dans notre architecture Backbone MPLS/IP, tout en collectant les données de QoS à l'aide d'Ansible et de Python. Cette approche nous permettra d'améliorer l'efficacité opérationnelle, de réduire les erreurs et de garantir des performances réseau optimales.