

Travail effectué par Mathilde Cléret lors de la SAE 5B.01 "Évolution d'une infrastructure"

Présentation du projet et de l'équipe

Le projet consiste à installer l'infrastructure d'une entreprise fictive. Notre infrastructure possède un réseau public et un réseau privée qui contiennent chacun des postes physiques, des spécificités techniques ainsi que différents services.

Pour réaliser se projet, l'établissement ainsi que les professeurs concernés, mon mis à disposition une salle de travail ou se trouve des ordinateurs sur le réseau de l'IUT. De plus j'ai à disposition du matériel afin de maquetter l'infrastructure de l'entreprise, c'est à dire, une baie informatique, des postes physiques, plusieurs switch, un commutateur, des Raspberry, de nombreux câbles et du matériel supplémentaire est disponible sur demande.

Pour plus d'informations concernant le sujet du projet, veuillez-vous référer au README ainsi qu'aux différents fichiers du projet disponible sur le GIT-LAB de l'université de Lille.

Mon groupe est l'équipe B. Il est composée de 4 membres (mathilde.cleret.etu, cyriac.faubert.etu, guerric.lepretre.etu et mattieu.seneschael.etu). Mon entreprise est nommée SYSSKO, ce qui à permis de créer notre DNS: "syssko.iut". Ce DNS permettra aux deux autres groupes chargés de créer leur propre infrastructure, de communiqué sur notre réseau public. De plus, un groupe est chargé de mettre en place le FAI (*fournisseur d'accès à Internet*). Ce groupe nous a communiqué la tranche d'adresse IP que nous devons utiliser afin d'assurer une bonne communication entre l'infrastructure réseau de chaque groupe.

La mise en place et la réalisation de ce projet a était très difficile du à de nombreux problèmes techniques mais surtout social. Les problèmes d'entente entre étudiants n'ont évidemment pas leur place d contre-rendu. Mais il me semble tout de même important de les signaler car il explique la forme de ce contre rendu. Malgré la demande d'un contre-rendu daté répertoriant les différentes taches réalisées, je préfère écrire un contre-rendu plus général. Car mon travail à été parfois ignoré, modifié voir supprimé du projet. Le Git du projet aurais donc eu de nombreuses incohérences pénalisantes pour tout le monde.

Sous cette forme, j'espère que le ou les évaluateurs pourrons voir ce que j'ai compris à travers ce projet et ce que j'ai mis en place ou ce à quoi j'ai pu participer. Pour plus d'informations, n'hésitez pas à venir discuter avec moi ou avec d'autres étudiants.

Sur les nombreuses recommandations de M. Beauflis ainsi que sur nos propres expériences, nous avons voulu mettre en place un système permettant de sauvegarder et de réinstaller très rapidement les différents services et configurations de notre infrastructure. De cette façon notre travail ne seras pas impacté en cas de problèmes physique ou réseau.

Mon équipe à donc décidée de déployer l'intégralité des services sur des fichiers vagrant, dans l'idée de pouvoir déployer notre infrastructure rapidement et de façon illimitée.

Malheureusement, cette solution c'est vite avéré très limité. Mais suite à des désaccords d'équipe, aucune autre solution n'a été envisagée. **Il est donc indispensable pour les projets futurs ou pour des étudiants reprenant ce projet de trouver une solution alternative ou une solution complémentaire.**

Voici une liste des différents points bloquants que j'ai pu relever sur cette solution:

- Cette solution ne permettant pas une sauvegarde des configurations matériels, tel que les switch, le commutateur ou encore les différents câblages; ces configurations ont été "sauvegardées" sous forme de liste de commandes et/ou de photos. Il est donc difficile de se retrouver dans les nombreux documents et la restauration de l'infrastructure n'est plus rapide ni efficace.
- Les fichiers vagrant ne sont pas l'idéal pour la mise en place de certains services. En effet, certaines mise en place demande l'écriture de plusieurs fichiers de configurations, de choix en mode graphique ou encore de liens avec d'autres services. Ces problèmes peuvent être contournés mais l'écriture des fichiers vagrant devient chronophage et ralenti énormément le reste du travail. De plus certains fichiers vagrant ne fonctionnent, parfois, plus lors d'un nouveau déploiement ou d'utilisation sur un autre ordinateur. De ce fait, cette solution fait perdre beaucoup de temps à nouveaux.
- Les fichiers vagrant ne permettent pas d'utiliser les adresses IP de notre choix dû à des questions de configurations. Cela complique la mise en place du réseau de notre infrastructure.
- Suite aux choix de certaines personnes sur les noms des VM configurées sur les fichiers vagrant ou sur le nommage des fichiers, il est facile de se perdre sur l'utilité de chaque fichier. Cela ralentit de nouveau le déploiement de l'infrastructure qui est sensé être rapide et pratique.
- Lorsque chaque personne crée son propre fichier vagrant pour le service qu'il tente de mettre en place, peu voir pas de commentaire ou documentation est réalisé. De ce fait, il est quasiment impossible pour les autres membres de l'équipe de comprendre le travail réalisé ou d'y apporter des modifications. Il est donc nécessaire que le créateur du fichier soit présent et réexplique à chaque personne le fichier selon ce qu'il se souvient. Ce procédé n'est guère productif et n'aide pas au développement des connaissances des étudiants. *Ce qui est pour moi, l'essence même de la mise en place d'un projet: pouvoir développer ses compétences et connaissances d'une nouvelle façon.*

Réseau, routage et câblage

Services

DHCP

Le DHCP (*Dynamic Host Configuration Protocol*) est un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'une machine, notamment en lui attribuant automatiquement une adresse IP et un masque de sous-réseau.

LDAP

NFS

Le NFS (*Network File System*) est un protocole qui permet à un ordinateur d'accéder via un réseau à des fichiers distants.

Gestion des mails

Pour la gestion des mails, nous avons décidé d'utiliser **Postfix**. C'est un serveur de messagerie électronique libre. Il permet la livraison de courriels et il est connu comme une alternative plus rapide, plus facile à administrer et plus sécurisée que la plupart des autres serveurs de messageries.

Postfix présente plusieurs avantages. Il dispose de plusieurs "systèmes" qui permettent une gestion des pourriels efficace ainsi que la possibilité de déléguer la gestion des courriels à un processus externe (sous la forme d'un démon ou d'un programme externe lancé à chaque courriel reçu, qui exécute des règles définies dans `/etc/postfix/main.cf` et `/etc/postfix/master.cf`). De plus, Postfix reçoit régulièrement des mises à jour de sécurité.

Le fichier `/etc/postfix/main.cf` est la configuration principale du serveur de messagerie Postfix. Il contient de nombreuses options qui définissent le comportement global du serveur. Voici quelques configurations de base que nous avons du comprendre et/ou modifier dans ce fichier :

- **myhostname = mail.syssko.iut**

Définit le nom d'hôte du serveur de messagerie. Cela peut être le FQDN (*Fully Qualified Domain Name*) du serveur.

- **mydomain = syssko.iut**

Indique le domaine par défaut pour les adresses électroniques générées par Postfix.

- **myorigin = \$mydomain**

Détermine le domaine à utiliser comme origine pour les courriers sortants générés par le serveur.

- **inet_interfaces = all**

Spécifie les interfaces réseau sur lesquelles Postfix doit écouter les connexions entrantes. "all" signifie que le serveur écoute sur toutes les interfaces disponibles.

- **inet_protocols = ipv4**

Indique quels protocoles réseau est utiliser pour écouter les connexions. Ici, nous avons choisi "ipv4" car nous nous sommes uniquement concentré sur l'IPv4 pour le moment.

- **mydestination = \$myhostname, \$mydomain, localhost.\$mydomain, localhost**

Définit les domaines pour lesquels le serveur est responsable. Les courriers destinés à ces domaines seront traités localement. Ici, nous avons gardé la configuration par défaut.

- **mynetworks = 127.0.0.0/8, 192.168.60.42/24**

Spécifie les réseaux qui sont autorisés à relayer du courrier à travers le serveur. Dans notre configuration, les adresses IP 127.0.0.0/8 (localhost) et 192.168.60.42/24 (?) sont autorisées.

- **mailbox_size_limit = 0**

Fixe la taille maximale des boîtes aux lettres. La valeur "0" signifie illimité.

Firewall linux

Extensions

Les extensions n'ont malheureusement pas eu le temps de voir le jour. Cependant, je me suis renseignée sur la compréhension de ces éléments ainsi que sur les différents moyens de les mettre en place. De cette façon, j'ai pu approfondir mes connaissances et ces éléments qui pourront être déployés plus facilement et plus rapidement en cas d'une suite ou une reprise du projet par d'autres étudiants.

IPV4 et IPV6

- **DHCP**

Il existe un protocole qui permet une configuration automatique des paramètres IPV6. Ce protocole se nomme **DHCPv6** (*Dynamic Host Configuration Protocol version 6*).

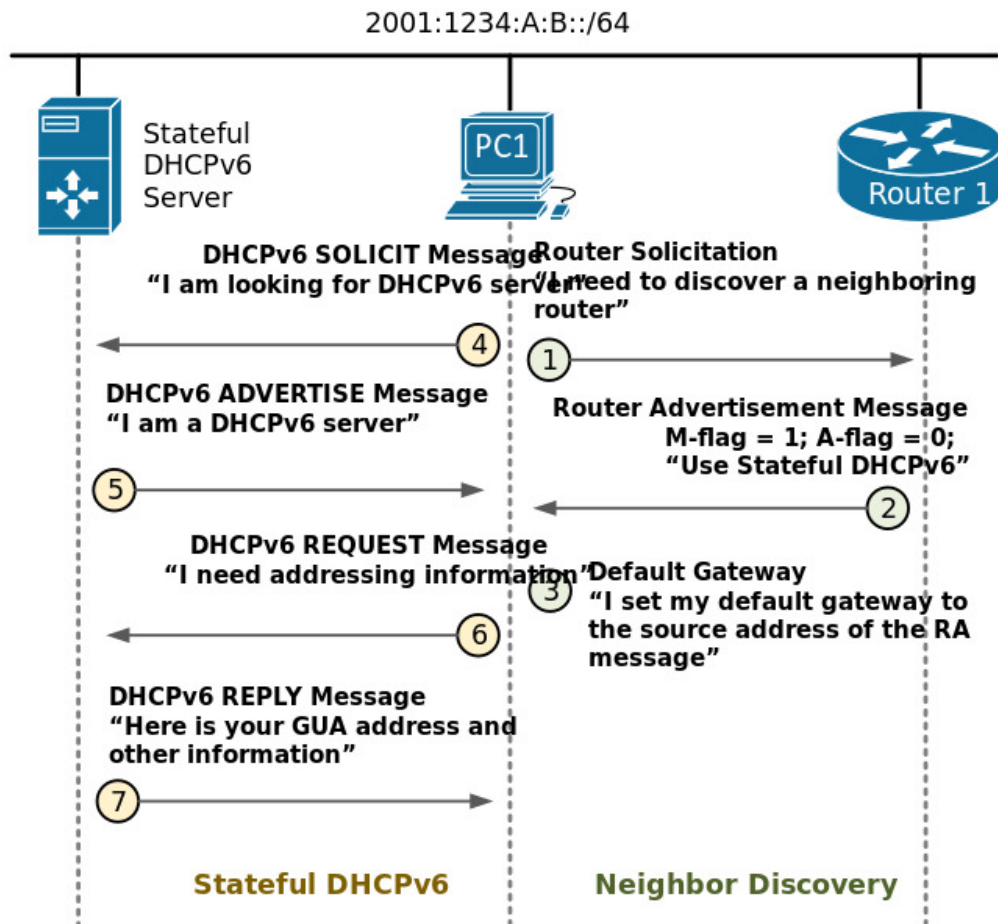
Il existe une autre méthode (plus simple) pour la configuration automatique des paramètres IPV6. Cette méthode se nomme **SLAAC** (*Stateless Automatic Auto Configuration*) et elle ne requiert pas de serveur DHCP.

Nous allons nous focaliser sur DHCPv6 afin de respecter les exigences du sujet.

Il existe deux versions de DHCPv6 : avec et sans états. Un protocole sans état (*stateless protocol*) est un protocole de communication qui n'enregistre pas l'état d'une session de communication entre deux requêtes successives. C'est à dire que chaque paire requête-réponse est traitée comme une transaction indépendante, sans lien avec les requêtes précédentes ou suivantes. A contrario, un protocole qui impose la conservation des informations sur l'état interne du serveur est appelé protocole avec état (*stateful protocol*).

Les deux versions de DHCPv6 présentent chacun leur avantages et inconvénients. Il me semble intéressant de s'orienter vers celui avec état. En effet, si une interruption se présente, le contexte et l'historique sont stockés et la reprise sera plus facile. De plus, la majorité des applications que nous utilisons au quotidien sont des applications "stateful", il peut donc être intéressant de mieux comprendre le fonctionnement de ce type de protocole.

Ci dessous, une représentation du fonctionnement de DHCPv6 lors de l'attribution d'une adresse IPV6.



La configuration de DHCPv6 est fortement similaire à la configuration de DHCP. Il s'agit donc de procéder en trois étapes. Pour commencer, il est recommandé d'établir sa table d'adressage (⚠ il faut indiquer les adresses des appareils en IPV6). Par la suite, il faut configurer le matériel nécessaire (commutateur / switch), pour cela les commandes requises sont semblable à celle pour DHCP. De nombreux exemples et documentations sont disponible sur internet. Une fois ces étapes réalisées, il ne reste plus qu'à vérifier le bon fonctionnement du service.

- **LDAP**

LDAP peut prendre en paramètres des adresses IPV6. Aucune installation ou mise en place particulière n'est requise. Il suffit donc de renseigner des adresses IPV6 lors de la configuration de LDAP.

- **NFS**

NFS est compatible avec IPV6 sur la plupart des systèmes. Il s'agit, simplement, de renseigner des adresses IPV6 lors de la configuration de NFS.

⚠ Veuillez vérifier la compatibilité de NFS en IPV6 sur le système utilisé. La liste des systèmes compatibles se trouve facilement sur internet. Pour le système utilisé lors de cette SAE (Debian), il n'y a aucun problème de compatibilité.

- **Gestion des mails**

Postfix supporte le protocole IPV6 depuis sa version 2.2 sur certains systèmes (dont *Linux* 2.4+). Il n'y a donc pas de problème de compatibilité sur les machines utilisées lors de cette SAE. La liste des systèmes compatibles se trouve facilement sur internet.

Cependant plusieurs configurations sont nécessaires afin d'assurer le bon fonctionnement de Postfix en IPv6. Il s'agit de deux nouveaux paramètres de configuration dans (*main.cf*) et une modification de la syntaxe de notation des adresses.

- Un des nouveaux paramètre est *inet_protocols*. Il indique quels protocoles Postfix doit utiliser pour établir ou accepter des connexions réseau, et également quels types de consultations DNS Postfix doit utiliser.

```
/etc/postfix/main.cf:
# Vous devez arrêter et redémarrer Postfix après avoir changer ce
paramètre
inet_protocols = ipv4          (DÉFAUT: n'active qu'IPv4)
inet_protocols = all          (active IPv4 et IPv6 s'il est supporté)
inet_protocols = ipv4, ipv6   (active IPv4 et IPv6)
inet_protocols = ipv6        (n'active qu'IPv6)
```

Par défaut, Postfix n'utilise qu'IPv4, car la plupart des systèmes ne sont pas raccordés à un réseau IPv6.

- L'autre paramètre est *smtp_bind_address6*. Il indique l'adresse d'interface locale utilisée pour les connexions IPv6 sortantes (comme *smtp_bind_address* pour IPv4).

```
/etc/postfix/main.cf:
smtp_bind_address6 = 2001:240:587:0:250:56ff:fe89:1
```

- ⚠ Il faut indiquer les adresses IPv6 entre crochet "[] " dans les valeurs des paramètres de *main.cf* et dans les autres fichiers de configuration afin d'éviter que Postfix confonde les adresses IPv6 avec d'autre type de valeurs à cause de la présence du caractère ":" dans les adresses IPv6.
- Plusieurs limitations de l'utilisation de Postfix en IPv6 sont connus et peuvent provoquer des erreurs ou des ralentissements. Voici quelques une de ces limitations:
 - L'ordre des tentatives de connexions sortantes IPv6/IPv4 n'est pas configurable. IPv6 est testé avant IPv4.
 - Postfix ne supporte pas les consultations DNSBL (*DNS Black Listing* - listes noires en temps réel) pour les adresses clientes IPv6.
 - Certains paramètres de configuration en IPv4 change "d'utilité" en IPv6 et peuvent compliquer l'utilisation de Postfix en IPv6.

- **Firewall linux**

VPN

Accès Wifi

Le mot de fin
