

Sommaire

* [Introduction 3](#_Toc506183674)
* [Choix techniques 3](#_Toc506183675)
* [Serveur DNS 4](#_Toc506183676)
* [Serveur DHCP 9](#_Toc506183677)
* [Serveur HTTP 10](#_Toc506183678)
* [Scripts de supervision 15](#_Toc506183679)
* [Conclusion 16](#_Toc506183680)

# 

# Introduction

Ce projet « Services, protocoles et scripts » a pour but de nous apprendre comment bâtir un système d’information et manipuler les différents outils, serveurs et protocoles nous permettant de veiller à son bon fonctionnement.

# Choix techniques

1. **Introduction :** Pour ce projet nous avions dû faire les bons choix techniques et veiller à bien respecter le cahier des charges fourni par l’entreprise Carnofluxe.
2. **Configuration :**
   1. **Machine virtuelle :**
      1. **Choix du logiciel :** VMWare Workstation
      2. **Pourquoi VMWare Workstation :** Nous avions le choix entre VMWare Workstation et VirtualBox, malgré la gratuité de ce dernier il n’est pas aussi performant que VMWare WS qui bug très rarement et qui offre une compatibilité avec plusieurs OS et propose une meilleure navigation grâce à son interface bien optimisée.
   2. **Distribution Linux :**
      1. **Choix de distribution :** Ubuntu
      2. **Pourquoi Ubuntu :** Pour la facilité de l’installation et du partitionnement, cette distribution offre aussi une interface graphique facile à utiliser ce qui nous a aidé à gagner plus de temps pour l’installation.
   3. **RAM :**
      1. **Allocation de mémoire vive :**
         1. **Serveur DNS (maître) et DHCP : 4GiO**
         2. **Serveur HTTP : 4GiO**
         3. **Serveur DNS (esclave) : 2GiO**
         4. **Client Windows (1) : 1GiO**
         5. **Client Windows (2) : 1GiO**
      2. **Pourquoi :** Pour les serveurs DNS/DHCP et HTTP nous avions besoin de fluidité pour la navigation, nous en avions moins besoin pour le serveur DNS esclave et encore moins car ces machines là sont installées juste pour les tests.

# Serveur DNS

1. **Introduction :** Nous ne pouvions faire la résolution de nom de domaine sans serveur DNS, c’est pour cela qu’il est utilisé dans ce projet :
   1. **Qu’est-ce qu’un serveur DNS :**

DNS veut dire « Domain Name System » ou système de nom de domaine, DNS est un annuaire pour ordinateur. Lorsque vous voulez accéder à un ordinateur dans le réseau, votre ordinateur va interroger le serveur DNS pour récupérer l’adresse de l’ordinateur que vous voulez joindre. Une fois, que votre ordinateur aura récupéré l’adresse du destinataire, il pourra le joindre directement avec son adresse IP.

Le serveur DNS va permettre de faire la relation entre nom d’ordinateur et adresse IP.

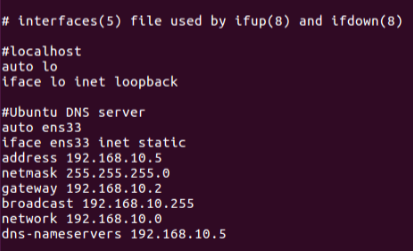
1. **Installation du serveur DNS :**
   1. **DNS maître :**

* **Etapes :**

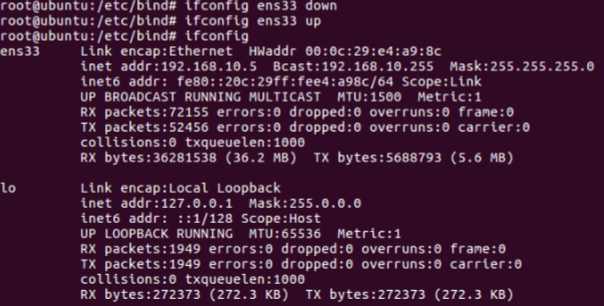
**apt-get install bind9**

**configuration de l’IP statique :**

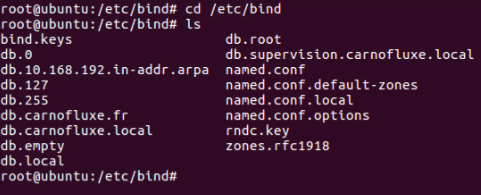
**nano /etc/network/interfaces**

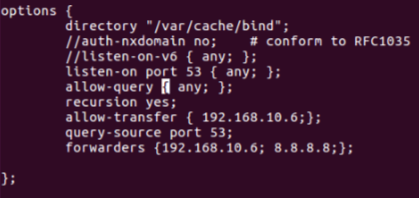


Redémarrage de l’interface :

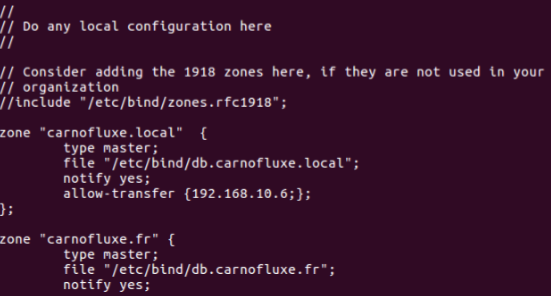


**Configuration du serveur DNS :**

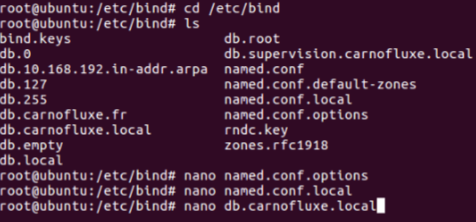


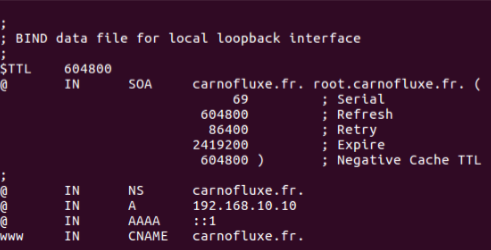


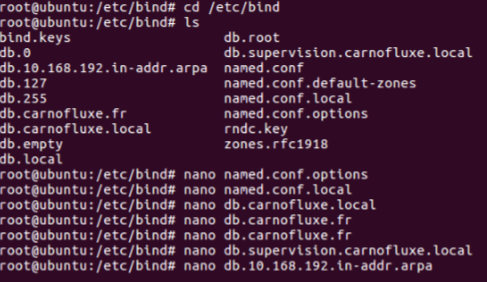
**Configuration des zones DNS :**

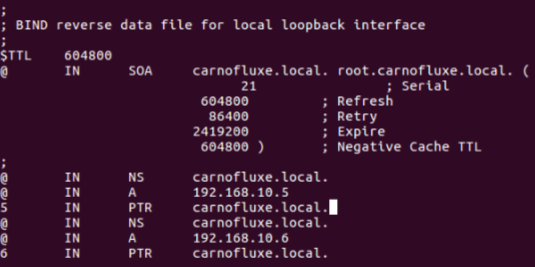


**Configuration de chaque zone DNS :**

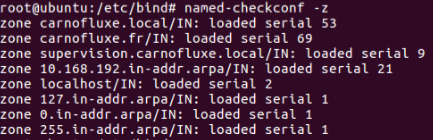




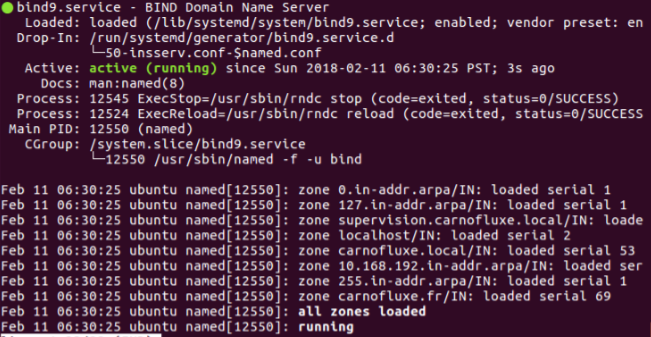




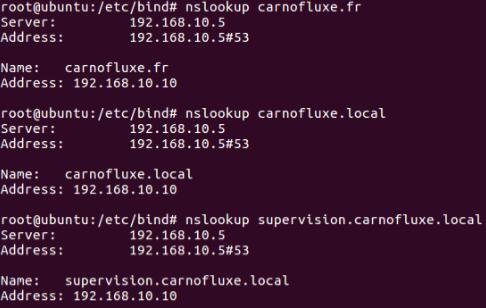
* service bind9 restart(Pour redémarrer les services DNS)



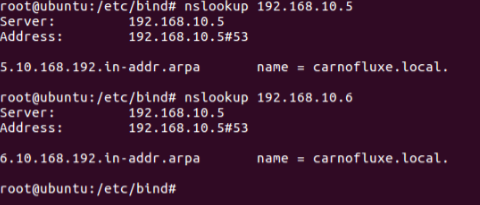
* service bind9 status(Pour connaître l’état du serveur DNS)



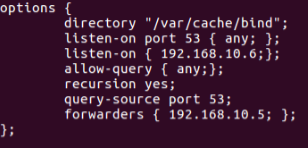
Résolution de noms de domaine :



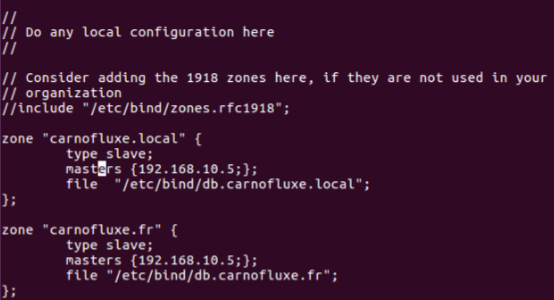
Résolution de l’adresse IP :



* 1. **Configuration DNS esclave :**
* nano named.conf.options

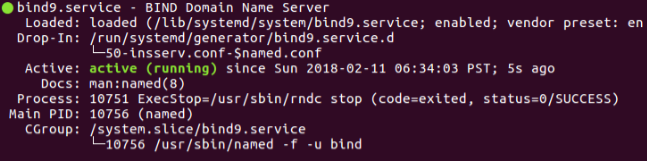


* nano named.conf.local





* service bind9 restart
* service bind9 status



# Serveur DHCP

1. **Introduction :**
2. **Qu’est-ce qu’un serveur DHCP :**

Un serveur DHCP (ou service DHCP) est un serveur (ou service) qui délivre des adresses IP aux ordinateurs qui se connectent sur le réseau de façon automatique.

1. **Fonctionnement d’un serveur DHCP :**

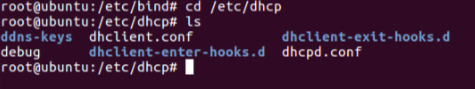
Les cartes réseaux des ordinateurs doivent être paramétrées pour recevoir automatiquement des adresses lorsque l’ordinateur démarre ou que l’on le connecte au réseau. Par défaut c’est le cas, car c’est la méthode la plus simple pour obtenir une adresse IP.

L’adresse IP doit être unique sur un réseau donc le serveur DHCP (ou service DHCP) va gérer les adresses et n’attribuer que des adresses non utilisées à tout nouvel ordinateur qui en fait la demande.

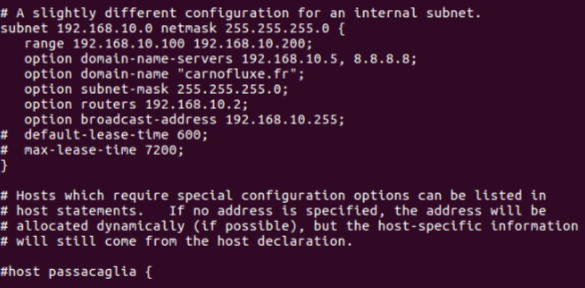
En fait, le serveur DHCP va délivrer un bail DHCP à l’ordinateur qui en fait la demande. (et uniquement à ceux qui en font la demande, et non pas à tous les ordinateurs qui se connectent sur le réseau).

1. **Installation et configuration :**

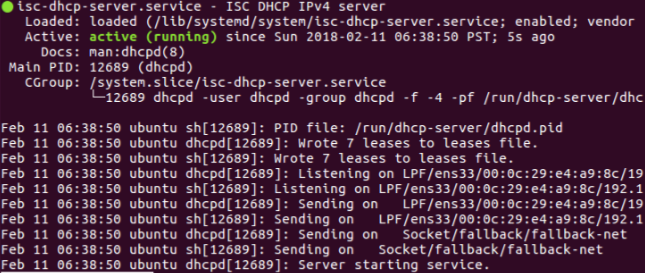
* apt-get install isc-dhcp-server



* nano dhcpd.conf (Configuration DHCP)



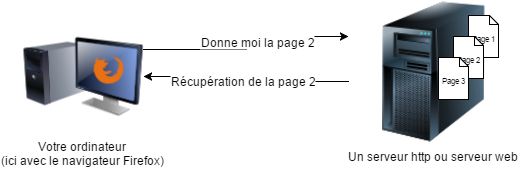
* service isc-dhcp-server restart
* service isc-dhcp-server status



# Serveur HTTP

1. **Introduction :** nous allons définir les différents termes que nous rencontrerons durant notre installation ainsi que nos choix techniques afin de les clarifier.
   1. **Qu’est-ce qu’un serveur web (serveur http) ?**

Pour aller sur une page web nous devons aller sur internet, et pour y accéder nous utilisons un navigateur internet, en fonction de nos clicks ou de nos saisies, ces pages sont contenues sur des serveurs, qui nous permettront de les afficher.



Le serveur attend les connexions et les demandes, il se contente de répondre à la demande en envoyant du contenu.

* 1. **Que signifie le terme HTTP ?**

Http est l’abréviation de HyperText Transfer Protocol. Pour faire simple c’est un protocole de transfert d’hypertext, un hypertexte est un texte contenant des liens qui permettent d’accéder directement à d’autres ressources (documents, images, …). Si on regarde bien, les pages qui sont affichées dans notre navigateur sont de l’hypertexte car elles contiennent des liens (vers des URL). Le port par défaut pour accéder à un serveur http est le port 80. Mais, les navigateurs sont capables de dialoguer avec un serveur http sur des ports différents (à condition bien sûr de leur dire et que le serveur attende de dialoguer sur un autre port).

* 1. **Apache 2 :**

Pourquoi avions-nous choisi Apache ? définissons d’abord Apache :

**Apache** *de son vrai nom HTTPD* est le plus populaire des serveurs HTTP. Il est produit par la « Apache Software Foundation »

En tant que projet open source Apache est un logiciel libre. Évidemment, le coût est souvent un facteur dans la décision sur un produit à utiliser. Cependant, de nombreux fans de Windows ferai remarquer que l'Internet Information Server (IIS) est gratuit (si vous avez déjà acheté Windows qui est). Vous devez considérer le coût total de possession, cependant. IIS n'est certainement pas libre si l'on considère la nécessité de posséder Windows afin de l'exécuter. Ajoutez à cela les spécifications matérielles supplémentaires requis pour exécuter Windows, IIS et les compétences supplémentaires nécessaires pour administrer IIS sur Windows et vous pouvez voir que IIS n'est guère libre. En ce sens, n'est pas non plus Apache. Cependant, Apache va fonctionner admirablement sur le matériel moindre, peut fonctionner sur des systèmes d'exploitation libres comme Linux ou Free BSD et a une configuration standard et simple. Cela représente, à mon avis, à un moindre coût de fonctionnement d'Apache.  
  
Apache gagne aussi sur les fonctionnalités. Les logs du serveur Apache sont très complètes et peut être facilement configuré pour se connecter adresses IP, URL de provenance et d'autres données qui sont importantes pour du marketing sur Internet. Apache supporte les modules chargeables, ce qui rend capable d'exécuter presque n'importe quel langage de script dynamique disponible, y compris Perl, Python, PHP et plus encore.  
  
Apache est assez stable et puissant. L'Apache Software Foundation (ASF), qui gère le développement open source du serveur web Apache, a créé des procédures et des techniques structurées pour des groupes de développeurs open source de collaboration. Ce processus cohérent ajoute à la stabilité Apache.  
  
La flexibilité permet également un gagnant Apache. Comme mentionné précédemment, le support d'Apache module chargeable, il est possible de configurer Apache dans une variété de façons. Apache peut être affiné pour servir du contenu statique ou pour servir de frontal pour un serveur d'application. Il a également la capacité de servir comme un serveur proxy. Le module de réécriture vous permet de manipuler l'espace URL avec le mandatement, le transfert et la redirection. Apache peut même être à l'écoute raisonnablement servir à la fois du contenu dynamique et statique. C'est l'une des nombreuses raisons pour lesquelles il est le serveur Web de choix pour les entreprises web de grande taille hébergement.   
Apache est également extrêmement portable. Actuellement, le serveur Web Apache sur un large éventail de plates-formes, y compris Mac, Windows, Solaris, Linux, Aix, et beaucoup plus. Le serveur Web Apache a été porté sur presque tous les matériels et logiciels de la plate-forme disponible. Vous ne serez jamais vous retrouver bloqué dans un système d'exploitation car vous avez besoin d'exécuter Apache.

* 1. **vHost (Virtual Host) :**

C’est une méthode que les [serveurs](https://fr.wikipedia.org/wiki/Serveur_informatique) tels que [serveurs Web](https://fr.wikipedia.org/wiki/Serveur_web) utilisent pour accueillir plus d'un [nom de domaine](https://fr.wikipedia.org/wiki/Nom_de_domaine) sur le même ordinateur, parfois sur la même [adresse IP](https://fr.wikipedia.org/wiki/Adresse_IP), tout en maintenant une gestion séparée de chacun de ces noms. Cela permet de partager les ressources du serveur, comme la mémoire et le processeur, sans nécessiter que tous les services fournis utilisent le même nom d'hôte. Le terme hébergement virtuel (virtual hosting) est utilisé habituellement en référence aux [serveurs Web](https://fr.wikipedia.org/wiki/Serveur_web).

* 1. **Page HTML :**

Que signifie **HTML ?** L’HyperText Markup Language, généralement abrégé HTML, est le [langage de balisage](https://fr.wikipedia.org/wiki/Langage_de_balisage) conçu pour représenter les [pages web](https://fr.wikipedia.org/wiki/Page_web). C’est un langage permettant d’écrire de l’[hypertexte](https://fr.wikipedia.org/wiki/Hypertexte), d’où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d’inclure des [ressources](https://fr.wikipedia.org/wiki/Ressource_du_World_Wide_Web) [multimédias](https://fr.wikipedia.org/wiki/Multim%C3%A9dia) dont des [images](https://fr.wikipedia.org/wiki/Image_num%C3%A9rique), des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents [interopérables](https://fr.wikipedia.org/wiki/Interop%C3%A9rabilit%C3%A9_en_informatique) avec des équipements très variés de manière conforme aux exigences de l’[accessibilité du web](https://fr.wikipedia.org/wiki/Accessibilit%C3%A9_du_web). Il est souvent utilisé conjointement avec le [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) [JavaScript](https://fr.wikipedia.org/wiki/JavaScript) et des [feuilles de style en cascade](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade) (CSS).

1. **Installation d’Apache 2 sur Linux (Ubuntu) :**

Nous allons commencer par installer les paquets d’Apache 2 :

* sudo apt-get update
* sudo apt-get install apache2

Nous allons ensuite passer en mode super-utilisateur pour pouvoir activer le pare-feu :

* sudo su
* ufw enable



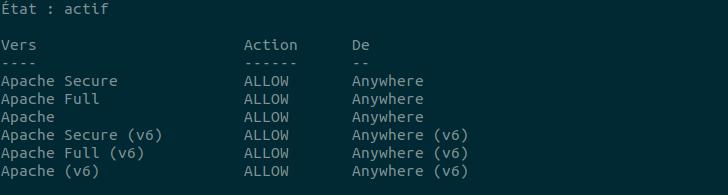
Le pare-feu est maintenant activé, nous pouvons commencer à configurer les ports, nous libérons donc le port 80 :

* sudo ufw allow 'Apache'

Vérifions maintenant :

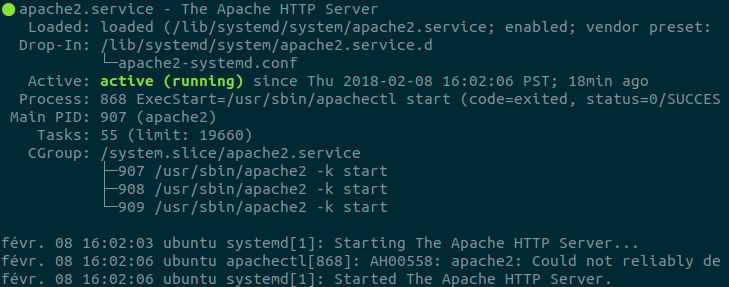
* sudo ufw status

Nous pouvons désormais utiliser le port 80.



Maintenant, vérifions l’état de notre serveur web :

* sudo systemctl status apache2

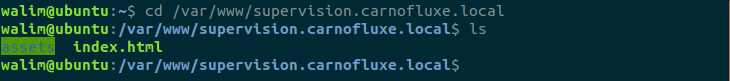


Maintenant que notre serveur web Apache 2 est installé et fonctionnel, nous allons pouvoir y mettre nos fichiers.

Pour cela il faudra se rendre dans le dossier « /var/www/ » et créer pour chaque site web un dossier avec nos noms de domaine :



Comme vous pouvez le constater nous avons deux dossiers « carnofluxe.fr » et « supervision.carnofluxe.local » contenant nos pages web.

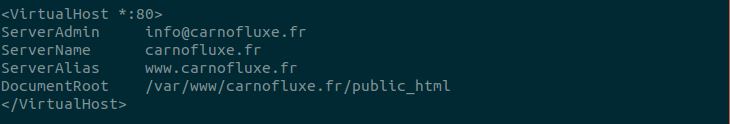


Le site « carnofluxe.fr » est accessible via l’IP mais pas « supervision.carnofluxe.local », mais pourquoi ? Et bien c’est parce que nous n’avons pas encore créé de fichier vHost qui nous permet d’associer plusieurs noms de domaines à une seule adresse IP, comment procéder ?



Nous allons nous rendre dans le dossier « sites-available » et créer un vHost pour chaque site :

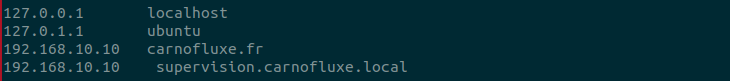




Nous allons entrer l’adresse de l’administrateur « ServerAdmin », le nom du serveur « ServerName », le nom de domaine du site web « ServerAlias » et enfin la localisation de notre page web « DocumentRoot ».

Pour nous assurer que notre dispositif était fonctionnel nous nous sommes permis d’ajouter nos noms de domaine sur le fichier « hosts »





Nous devrons ensuite activer nos sites web :



Nous allons maintenant redémarrer le serveur Apache afin que toutes les mises à jour soient appliquées :



Et voilà, notre serveur http est maintenant fonctionnel !



1. **Conclusion :**

L’installation du serveur web était simple et rapide, il fallait télécharger le serveur Apache, activer le pare-feu et le port 80, et créer les vHost.

**\*Concernant la modification du fichier /etc/Hosts :**

Cette étape a été appliquée uniquement pour les tests et ne fait aucunement l’assemblage final.

# Scripts de supervision

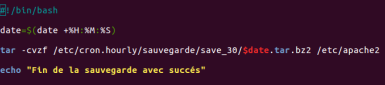
1. **Introduction :**
   1. **Qu’est-ce qu’un script Bash :**

Un langage de script est un [langage de programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation) qui permet de manipuler les fonctionnalités d'un système informatique configuré pour fournir à l'[interpréteur](https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique)) de ce langage un environnement et une [interface](https://fr.wikipedia.org/wiki/Interface) qui déterminent les possibilités de celui-ci. Le langage de script peut alors s'affranchir des contraintes de [bas niveau](https://fr.wikipedia.org/wiki/Bas_niveau) prises en charge par l'intermédiaire de l'interface.

1. **À quoi sert un script Bash :**

Un script Bash permet d’exécuter une multitude d’actions consécutives afin de faciliter une tâche.

1. **Explications des scripts :**
   1. **Script de sauvegarde sur 30 jours :**



1. **Exécution :** Serveur HTTP
2. **Fonction :** Ce script effectue une sauvegarde planifiée une fois par mois grâce à la commande crontab.
   1. **Script de sauvegarde sur 7 jours :**



1. **Exécution :** Serveur HTTP
2. **Fonction :** Ce script effectue une sauvegarde planifiée une fois par semaine grâce à la commande crontab.
   1. **Script de supervision :**



1. **Exécution :** Serveur DNS
2. **Fonction :** Ce script a pour fonction de récupérer une adresse IP en la stockant dans une variable, ensuite de la glisser sur le lien de l’API pour récupérer les informations concernant l’IP récupérée dans la variable.
   1. **Script de conversion :**



* + 1. **Exécution :** Serveur DNS
    2. **Fonction :** Ce script s’occupe de récupérer les informations du trafic

# Conclusion

En conclusion, ce projet nous a permis de maîtriser les différents outils qui composent un système d’information tout en veillant à son bon fonctionnement.

