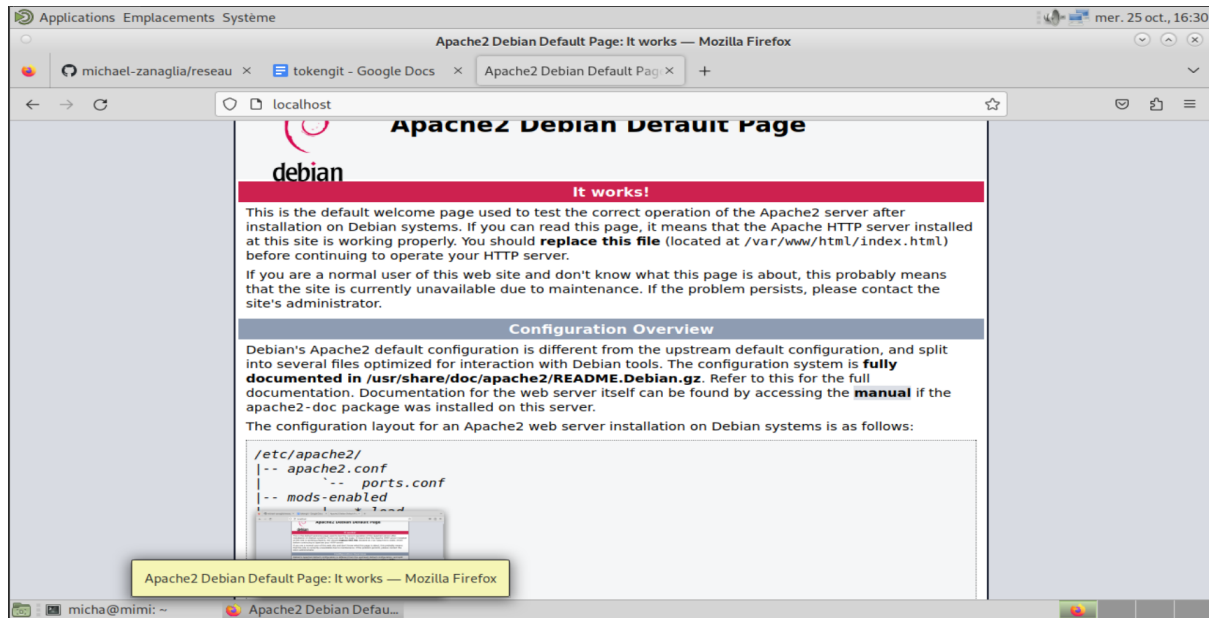


DDWS

Job 2



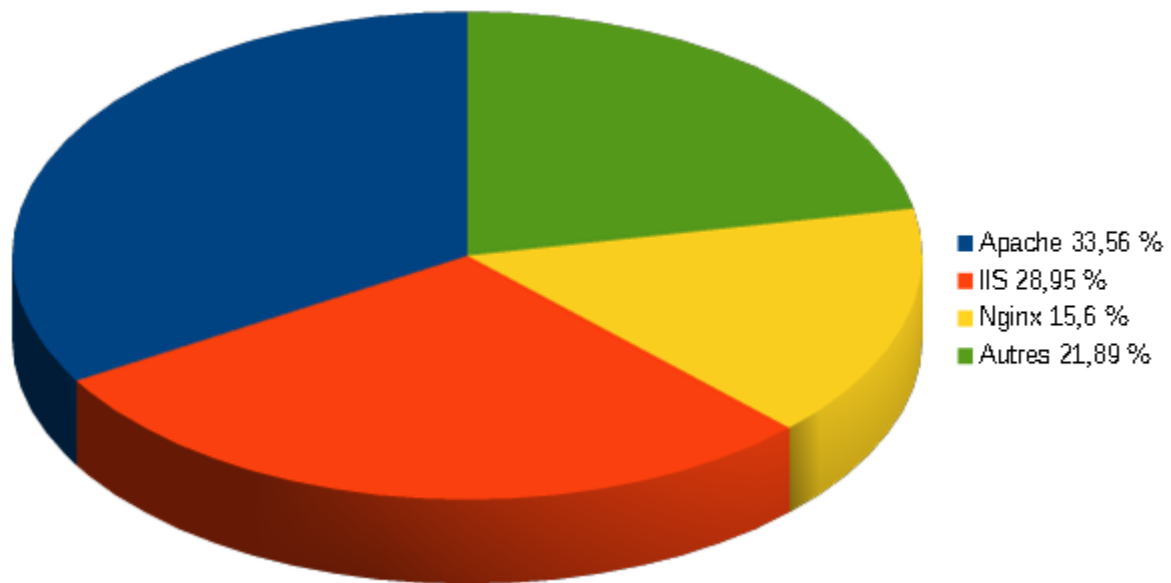
Job 3

Le tout premier serveur a été le CERN httpd, logiciel de serveur web sous protocole HTTP. Mais en raison de l'évolution constante du protocole il est devenu obsolète.

Il existe divers serveur web mais voici une liste non exhaustive :

- Apache : sûrement l'un des plus populaires. Open source, stable mais surtout compatible avec beaucoup d' OS. Il prend en charge de nombreux modules. En revanche configurations complexes et failles de sécurité souvent découvertes.
- Nginx : créé en 2002 pour gérer un site russe à trafic élevé, il a été classé selon Netcraft comme deuxième serveur le plus utilisé. Système asynchrone, il gère la multi-connexion en fonction des changements d'état. Cependant les modules tiers peuvent être très vite limités.
- Microsoft Internet Information Services (IIS) : serveur destiné au grand public, point fort, c'est qu'il est directement intégrée au système windows et point faible, il est moins présent sur les système non windows.
- LiteSpeed : serveur web léger, peu gourmand mais surtout une alternative à Apache. Cependant reste très limité.
- Unicorn : serveur web disponible pour Unix. Conçu pour des applications Python, simple d'utilisation mais en revanche nécessité de gérer en amont les requêtes HTTP
- Google Web Server (GWS) : version modifiée d'Apache, conçue pour gérer de grandes charges de travail ainsi qu'à une expansion envisageable. Inconvénient, il n'est pas open source et est utilisé spécifiquement pour google.

Ci dessous, une représentation de l'utilisation des serveurs HTTP en 2016.



Job 4

1. Attribuer un nom, une adresse IP et installer les paquets bind (pour cela entrez les commande : `sudo apt install bind9 bind9utils dnsutils`).
2. Je récupère l'adresse IP dont j'ai besoin avec la commande `/usr/sbin/ifconfig`

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.245.128 netmask 255.255.255.0 broadcast 192.168.245.255
    inet6 fe80::20c:29ff:fecc:b664 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:cc:b6:64 txqueuelen 1000 (Ethernet)
    RX packets 71629 bytes 35792838 (34.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 52031 bytes 4090789 (3.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 15621 bytes 1321853 (1.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15621 bytes 1321853 (1.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ici je recupere 192.168.245.128

3. Je me déplace ensuite sur le répertoire de bind : `cd /etc/bind`.
4. J'ouvre ensuite mon fichier `sudo nano named.conf.local` afin d'ajouter une zone.

```
GNU nano 5.4                                named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
// include "/etc/bind/zones.rfc1918";

zone "dnsproject.prepa.com" {
    type master;
    file "/etc/bind/db.dnsproject.prepa.com";
};
```

Je rentre mon bloc d'instruction. Je défini mon nom de domaine et j'enregistre le fichier dans mon dossier zones en format db.XXX.
Pour la création des fichiers db.XXX il va falloir copier le contenu du fichier db.local dans le fichier dnsproject.prepa.com.zone : cp db.local dnsproject.prepa.zone. Puis j'ouvre ce fichier :

```

;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA     localhost. root.localhost. (
                        2           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )    ; Negative Cache TTL
;
@         IN      NS      localhost.
@         IN      A       127.0.0.1
@         IN      AAAA    ::1
~
~
```

J'édite le fichier afin d'apporter les modifications suivante :

```
GNU nano 5.4                                db.dnsproject.prepa.com
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA     dnsproject.prepa.com. root.dnsproject.prepa.com. (
                        2           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )    ; Negative Cache TTL
;
IN      NS      dnsproject.prepa.com.
dnsproject.prepa.com IN A       192.168.245.128
@       IN      NS      dnsproject.prepa.com.
@       IN      A       192.128.245.128
dnsproject IN A       192.168.245.128
```

7. Nous allons éditer le fichier resolv.conf pour définir l'adresse IP. Pour se faire, entrez la commande `sudo nano /etc/resolv.conf` :

```
# Generated by NetworkManager
domain localdomain
search localdomain
nameserver 192.168.245.128

#nameserver 8.8.8.8
```

8. Redémarrer le bind9 : `systemctl restart bind9`

```
micha@mimi:/etc/bind$ nslookup dnsproject.prepa.com
Server:         192.168.245.128
Address:        192.168.245.128#53

Name:   dnsproject.prepa.com
Address: 192.128.245.128
```

Job 5

Avant tout il faut savoir qu'un nom de domaine est le substitut d'une adresse IP qui est traduit par l'intermédiaire du DNS.



La première chose à faire est de choisir un bureau d'enregistrement accrédité par l'ICANN est une organisation à but non-lucratif chargée de coordonner tous les espaces numériques sur Internet. C'est par ce bureau que l'on va choisir et acheter son nom de domaine. Pour en citer il y a par exemple Hostinger et PlanetHoster. Ensuite on fait une recherche pour vérifier si notre nom de domaine n'existe pas et une fois choisi il faut prendre en compte l'extension. Il en existe plein mais les plus connus sont .com, .net ou encore .fr ou .org. Le

choix va dépendre de plusieurs facteurs comme le coût, le public visé etc.. Une chose est sûre, .com est sans doute le plus utilisé.

Job 6

Job 7