

Documentation d'architecture

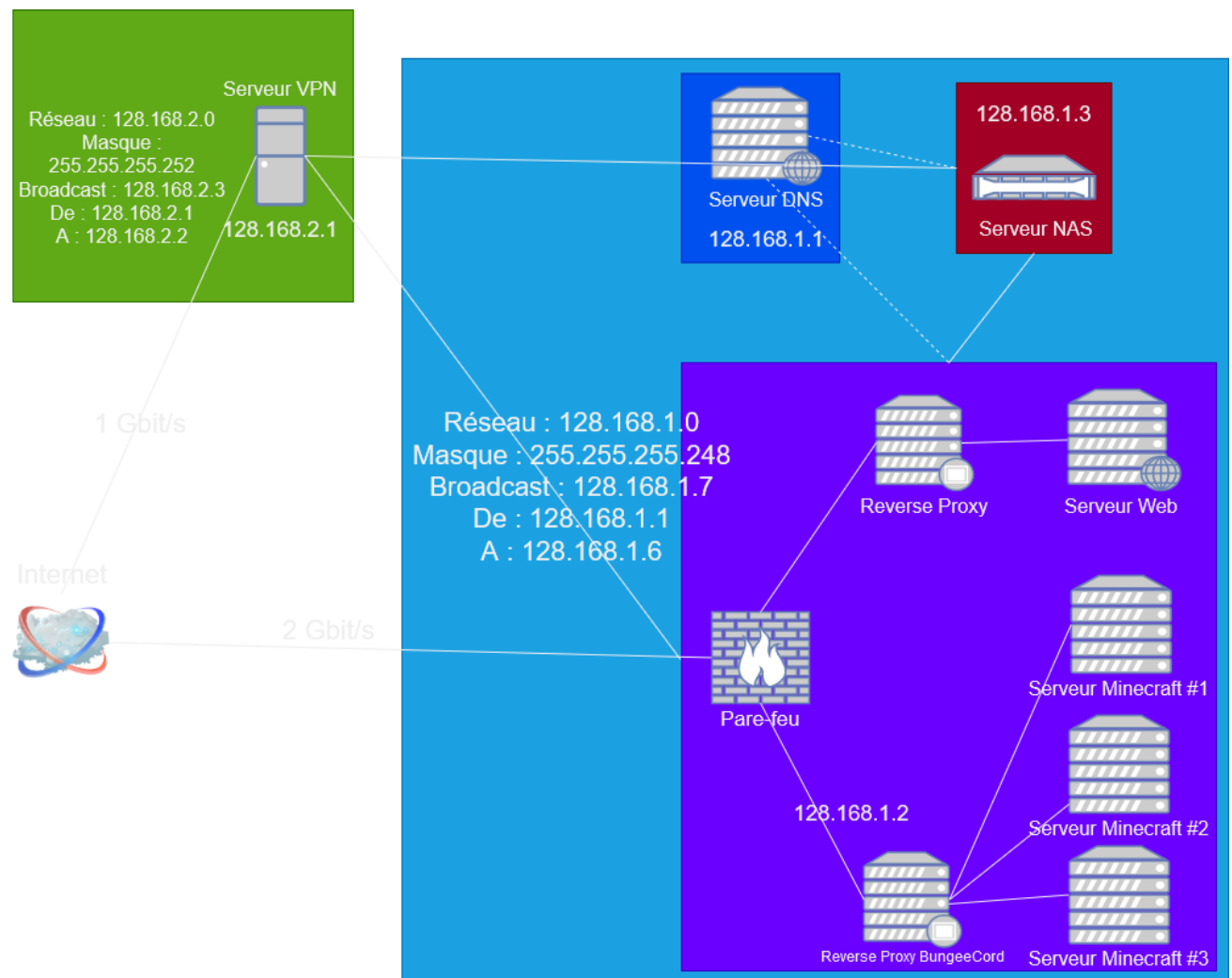
Sommaire :

1. Infrastructure et réseaux
2. Les services et fonctionnalités utilisés
3. Configurations réalisées sur les systèmes et services
4. Les bonnes pratiques réalisées

1. Infrastructure et réseaux :

Pour rappel, nous devons mettre en place un environnement type que l'on pourrait trouver dans des entreprises gérant des serveurs Minecraft.

Nous avons donc théorisé l'infrastructure suivante :



Comportant donc 2 réseaux avec 4 machines au total. Nous avons donc choisi deux réseaux avec peu d'adresse IP disponible simulant une petite infrastructure.

Notre premier réseau est le réseau 128.168.1.0 qui est contient la majeure partie de notre infrastructure à savoir 3 machines. Ce réseau à un masque, permettant d'avoir au maximum 6 appareils sur le réseau ce qui permet de restreindre et donc de participer à la sécurité du réseau mais laisse une marge si jamais l'infrastructure serait amenée à grandir. L'adresse de broadcast est la suivante : 128.168.1.7.

Le serveur DNS sera hébergé seul sur la machine avec l'adresse IP 128.168.1.1 avec la configuration suivante 2 cœur de CPU et 2GB de RAM afin que le système et le serveur DNS puisse subvenir à leurs besoins.

Notre second serveur est le serveur comportant le pare-feu, les deux reverse proxy menant respectivement au serveur web et aux différents serveurs Minecraft. Ce serveur a pour adresse la suivante : 128.168.1.2. Il est celui avec plus de capacités à savoir 8 cœurs et 64 GB de RAM ainsi qu'une bande passante de 2Gbit/s minimum. Les serveurs de jeux étant très gourmand.

La dernière adresse est notre serveur NAS qui a pour adresse : 128.168.1.3.

Le second réseau est le réseau 128.168.2.0 qui lui est dédié au serveur VPN qui permet une sécurité supplémentaire de connexion pour les administrateurs réseaux et administrateurs systèmes. Ce réseau n'ayant qu'un seul serveur le réseau est fait pour accueillir un maximum de deux appareils. Le serveur VPN se trouvera donc sur l'adresse 128.168.2.1.

2. Les services et fonctionnalités utilisés :

- 3 serveurs Minecraft Spigot
- 1 VPN (OpenVPN)
- 2 reverse proxy (Apache2 & Bungeecord)
- 1 serveur DNS (Dnsmasq)
- 1 serveur NAS
- 1 pare-feu (UFW)
- 1 serveur web (Apache2)
- Certificat SSL auto-signé

Vous pouvez trouver ci-dessus la liste des services et fonctionnalités dont nous avons besoin pour notre infrastructure.

Services de jeu :

Nous avons fait le choix de prendre des serveurs Minecraft sous Spigot car c'est une version modifiée de la version originale des serveurs du jeu qui est plus optimisée et qui permet d'apporter des extensions en jeu.

Avoir une version plus optimisée permet avec les mêmes capacités ressources d'avoir des serveurs plus performants. Les 3 serveurs Minecraft seront donc soumis au reverse proxy nommé BungeeCord, c'est une reverse proxy développée par md_5.

Ce reverse proxy est spécifique aux serveurs Minecraft Spigot et ne peut être utilisé qu'avec ce type de serveurs. Il permet de réguler le flux entrant et sortant des serveurs Minecraft et de rediriger les connexions vers un serveur Minecraft spécifique pour changer de serveur il faudra le faire en jeu.

Tout ceci est administré par des plugins et relève de la décision des créateurs des serveurs Minecraft.

Le reverse proxy se trouve sur le port 27565 et les serveurs Minecraft sur les ports suivants :

- 27566
- 27567
- 27568

Service web :

Nous avons fait le choix de s'orienter sur le serveur Apache2 car étant l'un des serveurs web les plus utilisés à travers et comportant un grand nombre de documentations ainsi que certaines fonctionnalités supplémentaires utiles à notre infrastructure tel que le reverse proxy qu'il comporte. Sur ce serveur se trouvera un site web en PHP permettant d'accéder aux informations des différents serveurs Minecraft tel que leurs statuts, le nombre de joueurs connectés, etc...

Ce site web se trouvera donc sur le port 443 qui est donc le port correspondant à l'HTTPS, à savoir que notre serveur web sera équipé d'un certificat SSL auto-signé permettant donc l'accès à un niveau de sécurité supplémentaires mais ne garantissant pas à l'utilisateur une sécurité totale car non reconnue pas une instance. Et enfin pour terminer avec la partie web, un reverse proxy Apache2 est déployé afin d'améliorer la sécurité une fois de plus de notre infrastructure du point de vue web.

Système de sauvegarde :

Afin d'assurer un rétablissement rapide des services en cas d'interruptions dû à un problème concernant les données sauvegardées sur les différentes machines. Nous avons mis en place un serveur NAS en configuration raid 1, afin d'avoir quoi qu'il arrive une double sauvegarde Il vient sauvegarder le serveur web toute les semaines et toute l'infrastructure Minecraft deux par jour.

Service DNS :

Afin de pouvoir pour les administrateurs réseaux et systèmes mieux si retrouver dans l'infrastructure, nous avons décidé de mettre en place un serveur DNS avec Dnsmasq. Celui-ci permettant donc d'attribuer aux serveurs se trouvant sur le réseau 128.168.1.0 un nom de domaine.

Service VPN :

Pour améliorer la sécurité de l'infrastructure il a été décidé de mettre en place un serveur VPN avec OpenVPN permettant aux différents administrateurs de pouvoir se connecter de manière centralisée à l'infrastructure limitant des problèmes de sécurité. Ce serveur ne sert de liaison seulement pour administrer l'infrastructure. Le grand public se connectant directement au service dont il a besoin.

Sécurité :

Enfin sur la machine ayant tous les services accessibles au grand public nous avons décidé de déployer un pare-feu avec UFW permettant de réguler l'entrée et la sortie sur les différents ports du serveur. En bloquant tout ceux non utilisés par nos services afin de limiter au maximum les entrées ou sorties indésirées.

3. Configurations réalisées sur les systèmes et services :

Si les adresses IP ne correspondent pas à ce qui a été énoncé plus haut c'est que les exemples donnés sont une simulation dans un environnement virtuel.

Système :

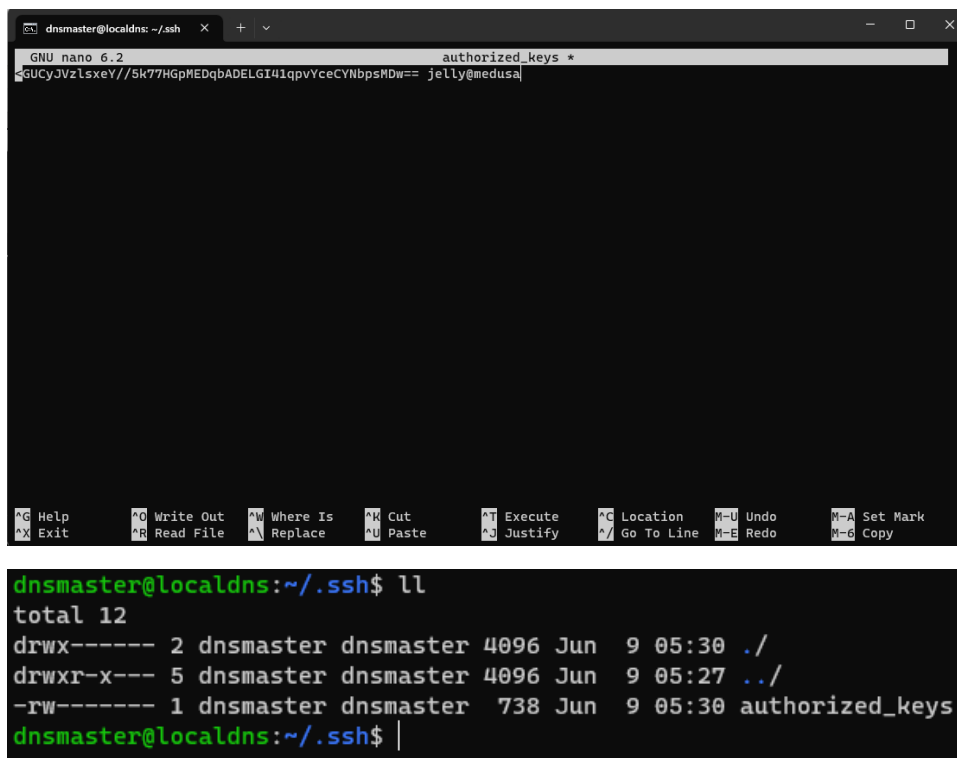
Nous avons fait le choix pour l'OS de nous munir de Ubuntu server 22.04 étant la version la plus récente et stable à ce jour. Il est recommandé pour les serveurs de jeux d'être sur un OS Linux basé sur Debian et nous avons fait le choix de Ubuntu pour son accessibilité ainsi que pour les nombreux packages installés.

Nous avons mis à jour le système dès son installation avec les commandes suivantes :

`sudo apt update; sudo apt upgrade; sudo apt dist-upgrade`

Clés SSH :

Ajouter la clé publique sur l'utilisateur voulu en allant modifier le fichier `authorized_keys` comme cela :



```
dnsmaster@localdns: ~/.ssh$ nano authorized_keys
GNU nano 6.2 authorized_keys *
gUCyJVzlsxeY//5k77HGpMEDqbADELGI41qpVYceCYNbpsMDw== jelly@medusa

dnsmaster@localdns: ~/.ssh$ ll
total 12
drwx----- 2 dnsmaster dnsmaster 4096 Jun  9 05:30 ./
drwxr-x--- 5 dnsmaster dnsmaster 4096 Jun  9 05:27 ../
-rw----- 1 dnsmaster dnsmaster 738 Jun  9 05:30 authorized_keys
dnsmaster@localdns: ~/.ssh$
```

Modifier la configuration SSH de la machine :

```
dnsmaster@localdns:~/.ssh$ sudo nano /etc/ssh/sshd_config
```

Changement du port SSH et interdire la connexion avec root en SSH :

```
Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

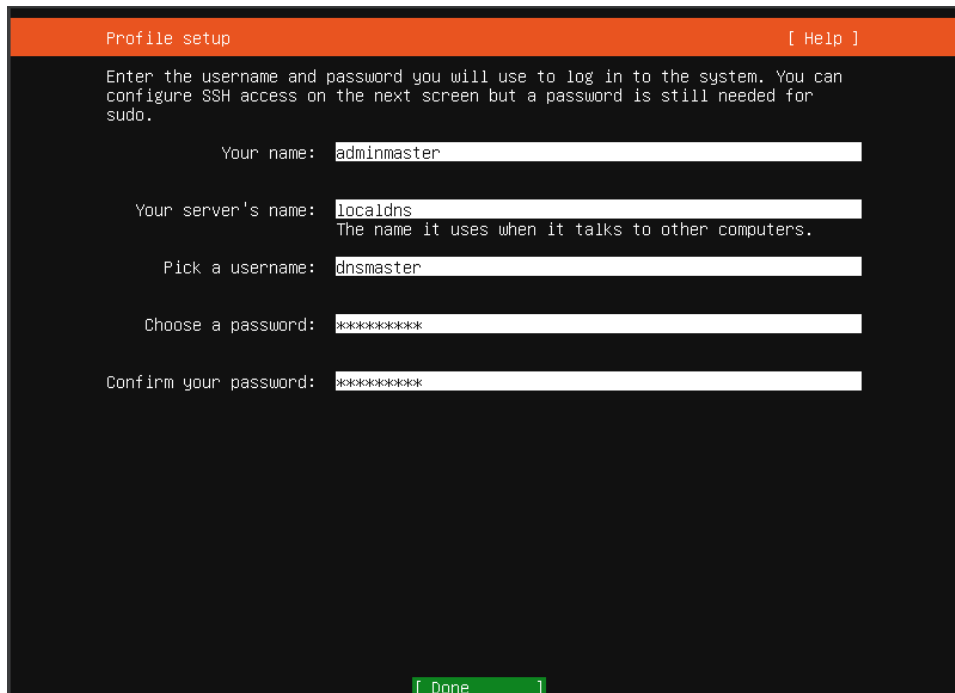
#LoginGraceTime 2m
PermitRootLogin no
```

Désactiver la connexion par mot de passe :

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
```

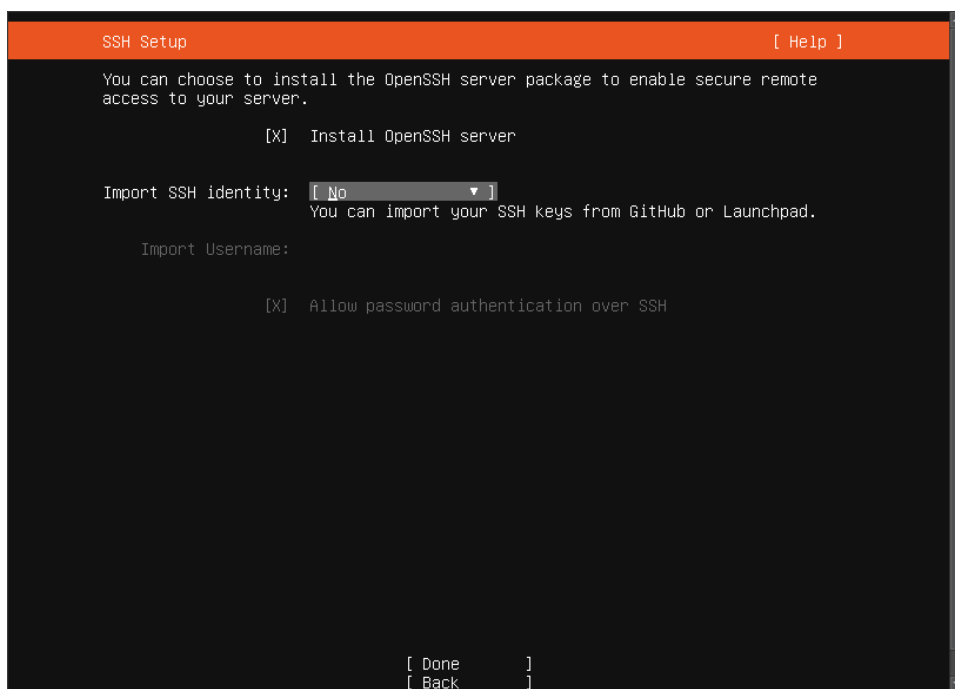
DNS (Dnsmasq) :

Information utilisateur du serveur DNS :



The screenshot shows a terminal window titled "Profile setup" with a "[Help]" link in the top right corner. The text inside the terminal reads: "Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo." Below this, there are five input fields: "Your name:" with the value "adminmaster", "Your server's name:" with the value "localdns" and a subtext "The name it uses when it talks to other computers.", "Pick a username:" with the value "dnsmaster", "Choose a password:" with the value "*****", and "Confirm your password:" with the value "*****". At the bottom center, there is a green button labeled "[Done]".

Installation par défaut d'un serveur SSH :



The screenshot shows a terminal window titled "SSH Setup" with a "[Help]" link in the top right corner. The text inside the terminal reads: "You can choose to install the OpenSSH server package to enable secure remote access to your server." Below this, there are three options: "[X] Install OpenSSH server", "Import SSH identity:" with a dropdown menu showing "No" and a subtext "You can import your SSH keys from GitHub or Launchpad.", and "Import Username:". At the bottom, there is an option "[X] Allow password authentication over SSH". At the bottom center, there are two buttons: "[Done]" and "[Back]".

Mise à jour de la machine comme expliqué dans Système.

Mise en place de la configuration SSH comme expliqué dans clés SSH.

Pour installer le DNS :

Sudo apt install dnsmasq

Pour modifier la configuration :

```
dnsmaster@localdns:~$ sudo nano /etc/dnsmasq.conf
```

Changement du port

Domain-needed : faire passer les noms de domaine en prioritaire en local

Bogus-priv : bloque l'adresse en dehors du réseau

```
# Configuration file for dnsmasq.
#
# Format is one option per line, legal options are the same
# as the long options legal on the command line. See
# "/usr/sbin/dnsmasq --help" or "man 8 dnsmasq" for details.
#
# Listen on this specific port instead of the standard DNS port
# (53). Setting this to zero completely disables DNS function,
# leaving only DHCP and/or TFTP.
port=5353
#
# The following two options make you a better netizen, since they
# tell dnsmasq to filter out queries which the public DNS cannot
# answer, and which load the servers (especially the root servers)
# unnecessarily. If you have a dial-on-demand link they also stop
# these requests from bringing up the link unnecessarily.
#
# Never forward plain names (without a dot or domain part)
domain-needed
# Never forward addresses in the non-routed address spaces.
bogus-priv
```

Ajout d'un dns primaire et d'un dns secondaire (Google).

Ajout d'une taille de cache pour fluidifier le réseau.

```
server=8.8.8.8
server=4.4.4.4
cache-size=1000|
```

Redémarrer le service et vérifier qu'il est bien démarré :

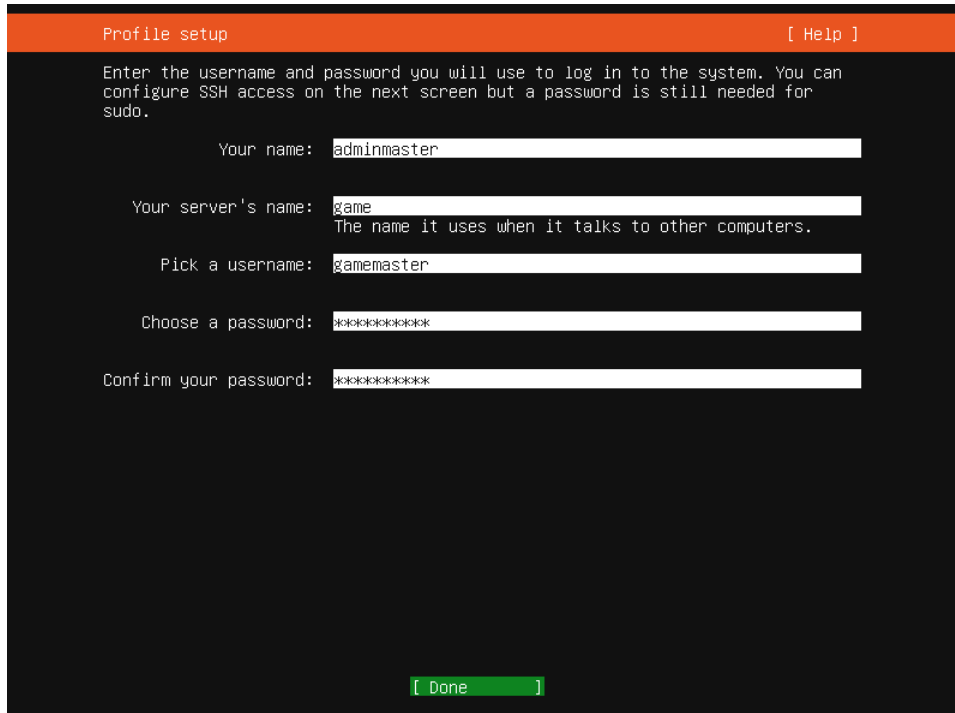
```
dnsmaster@localdns:~$ sudo systemctl restart dnsmasq
dnsmaster@localdns:~$ sudo systemctl status
● localdns
   State: running
   Jobs: 0 queued
  Failed: 0 units
   Since: Fri 2023-06-09 05:28:04 UTC; 19min ago
  CGroup: /
          └─user.slice
              └─user-1000.slice
                  └─user@1000.service ...
                      └─init.scope
                          └─957 /lib/systemd/systemd --user
                              └─958 (sd-pam)
                                  └─session-3.scope
                                      └─984 sshd: dnsmaster [priv]
                                          └─1030 sshd: dnsmaster@pts/0
                                              └─1031 -bash
                                                  └─1777 sudo systemctl status
                                                      └─1778 sudo systemctl status
                                                          └─1779 systemctl status
                                                              └─1780 less
                                                                  └─session-1.scope
                                                                      └─679 /bin/login -p --
                                                                          └─966 -bash
                                                                              └─init.scope
                                                                                  └─1 /sbin/init
```

Modifier le fichier hosts pour ajouter les noms de domaine à toutes les ip :

```
dnsmaster@localdns:/etc$ sudo nano hosts
192.168.1.51 dns.mc.net
```

Serveur web (Apache2) :

Création de la machine d'environnement de jeu :



Installation d'un serveur SSH

Mise à jour de la machine comme expliqué dans Système.

Mise en place de la configuration SSH comme expliqué dans clés SSH.

Ajouter le DNS en tant que DNS primaire :

```
gamemaster@game:/etc/netplan$ nano 00-installer-config.yaml
```

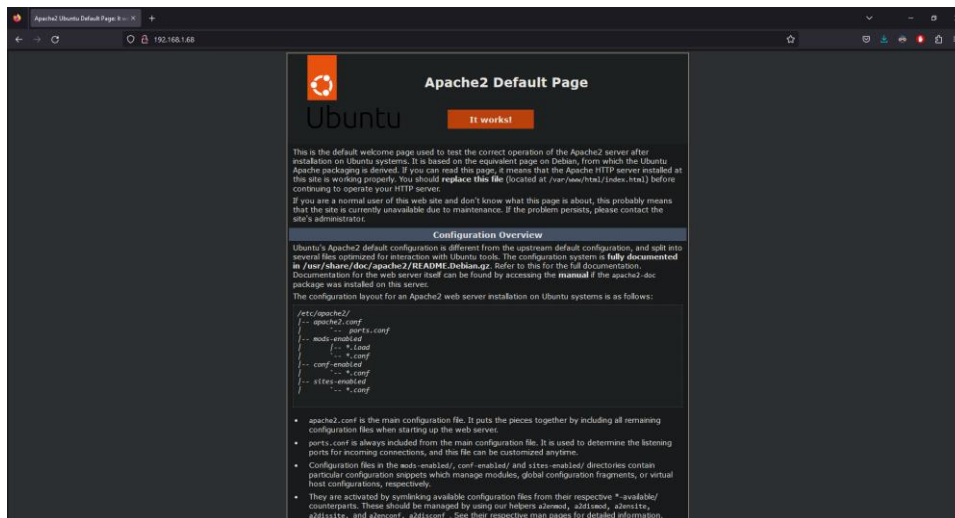
```
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.1.68/24
      gateway4: 192.168.100.1
      nameservers:
        addresses: [192.168.1.51, 8.8.8.8]
  version: 2
```

Installation du serveur web Apache2 :

Sudo apt install apache2

Vérification que le serveur web est démarré et fonctionnel :

Sudo systemctl status apache2



Créer la page web :

```
gamemaster@game:/var/www$ sudo chown -R $USER:$USER /var/www/game.mc.net
gamemaster@game:/var/www$ sudo chown -R 755 /var/www/game.mc.net
gamemaster@game:/var/www$ sudo nano game.mc.net/index.php
```

Création du virtual host :

```
gamemaster@game:/var/www$ sudo nano /etc/apache2/sites-available/game.mc.net.conf
```

Virtual sur le port 80 se dirigeant vers la page web créer précédemment :

```
<VirtualHost *:80>
    ServerName game.mc.net
    ServerAlias game.mc.net
    DocumentRoot /var/www/game.mc.net
</VirtualHost>
```

Désactivation du site par défaut :

```
sudo a2dissite 000-default.conf
```

Activation du nouveau site :

```
sudo a2ensite game.mc.net.conf
```

```
systemctl reload apache2
```

Reverse proxy (Apache2) :

Activer les modules nécessaires :

```
sudo a2enmod proxy && sudo a2enmod proxy_http && a2enmod ssl
```

```
sudo systemctl restart apache2
```

Modification du virtual host :

```
sudo nano /etc/apache2/sites-available/game.mc.net.conf
```

```
<VirtualHost *:80>
    ServerName game.mc.net
    ServerAlias game.mc.net
    DocumentRoot /var/www/game.mc.net
    ProxyRequests On
</VirtualHost>
```

Ajouter le certificat auto-signé :

```
sudo apt install certbot
```

```
sudo certbot --apache
```

Modification du virtual host :

Changement du port en 443

Ajout des lignes suivantes :

SSLEngine on

SSLCertificateFile /etc/letsencrypt/live/example.com/fullchain.pem

SSLCertificateKeyFile /etc/letsencrypt/live/example.com/privkey.pem

sudo systemctl reload apache2

Environnement Minecraft :

Installer Java afin de pouvoir démarrer les différents services :

```
sudo apt install openjdk-17-jdk
```

Création des dossiers :

```
gamemaster@game:~$ mkdir server1
gamemaster@game:~$ mkdir server2
gamemaster@game:~$ mkdir server3
```

Téléchargement d'un serveur Minecraft Spigot version 1.19.4 :

```
gamemaster@game:~$ wget https://download.getbukkit.org/spigot/spigot-1.19.4.jar
--2023-06-09 07:12:19-- https://download.getbukkit.org/spigot/spigot-1.19.4.jar
Resolving download.getbukkit.org (download.getbukkit.org)... 188.114.97.2, 188.114.96.2, 2a06:98c1:3121::2, ...
Connecting to download.getbukkit.org (download.getbukkit.org)[188.114.97.2]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 71219476 (68M) [application/java-archive]
Saving to: 'spigot-1.19.4.jar'

spigot-1.19.4.jar      100%[=====] 67.92M  61.6MB/s   in 1.1s
2023-06-09 07:12:20 (61.6 MB/s) - 'spigot-1.19.4.jar' saved [71219476/71219476]

gamemaster@game:~$ |
```

Copier le fichier dans les 3 dossiers.

```
gamemaster@game:~$ ls -R
.:
server1 server2 server3

./server1:
spigot.jar

./server2:
spigot.jar

./server3:
spigot.jar
gamemaster@game:~$ |
```

Création du script de lancement (run.sh) :

```
#!/bin/sh

java -Xms1G -Xmx2G -XX:+UseG1GC -jar spigot.jar nogui
```

Rendre le fichier run.sh exécutable :

```

gamemaster@game:~$ chmod u+x run.sh
gamemaster@game:~$ ll
total 52
drwxr-x--- 8 gamemaster gamemaster 4096 Jun  9 07:15 ./
drwxr-xr-x 3 root      root      4096 Jun  9 06:08 ../
-rw-r--r-- 1 gamemaster gamemaster 220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 gamemaster gamemaster 3771 Jan  6 2022 .bashrc
drwx----- 2 gamemaster gamemaster 4096 Jun  9 06:09 .cache/
drwxrwxr-x 3 gamemaster gamemaster 4096 Jun  9 07:14 .local/
-rw-r--r-- 1 gamemaster gamemaster 807 Jan  6 2022 .profile
-rwxrw-r-- 1 gamemaster gamemaster 65 Jun  9 07:15 run.sh*
drwxrwxr-x 2 gamemaster gamemaster 4096 Jun  9 07:13 server1/
drwxrwxr-x 2 gamemaster gamemaster 4096 Jun  9 07:13 server2/
drwxrwxr-x 2 gamemaster gamemaster 4096 Jun  9 07:13 server3/
drwx----- 2 gamemaster gamemaster 4096 Jun  9 06:13 .ssh/
-rw-r--r-- 1 gamemaster gamemaster  0 Jun  9 06:09 .sudo_as_admin_successful
-rw-rw-r-- 1 gamemaster gamemaster 170 Jun  9 07:12 .wget-hsts
gamemaster@game:~$ |

```

Lancer les serveurs via le script bash.

Accepter le fichier eula.txt

Nouveau lancement de serveur afin de correctement les créer.

Mise en place du reverse proxy Bungeecord :

```

gamemaster@game:~$ mkdir bungeecord
gamemaster@game:~$ cd bungeecord/
gamemaster@game:~/bungeecord$ wget https://ci.md-5.net/job/BungeeCord/lastSuccessfulBuild/artifact/bootstrap/target/BungeeCord.jar
--2023-06-09 07:24:11-- https://ci.md-5.net/job/BungeeCord/lastSuccessfulBuild/artifact/bootstrap/target/BungeeCord.jar
Resolving ci.md-5.net (ci.md-5.net)... 104.26.15.18, 104.26.14.18, 172.67.71.53, ...
Connecting to ci.md-5.net (ci.md-5.net)|104.26.15.18|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19233900 (18M) [application/java-archive]
Saving to: 'BungeeCord.jar'

BungeeCord.jar      100%[=====] 18.34M  52.6MB/s   in 0.3s

2023-06-09 07:24:11 (52.6 MB/s) - 'BungeeCord.jar' saved [19233900/19233900]
gamemaster@game:~/bungeecord$ |

```

Script de lancement (run.sh) :

```

#!/bin/sh
java -Xms512M -Xmx512M -jar BungeeCord.jar

```

```

gamemaster@game:~/bungeecord$ chmod u+x run.sh
gamemaster@game:~/bungeecord$ ll
total 18796
drwxrwxr-x 2 gamemaster gamemaster 4096 Jun  9 07:25 ./
drwxr-x--- 9 gamemaster gamemaster 4096 Jun  9 07:24 ../
-rw-rw-r-- 1 gamemaster gamemaster 19233900 Jun  7 16:12 BungeeCord.jar
-rwxrw-r-- 1 gamemaster gamemaster 53 Jun  9 07:25 run.sh*
gamemaster@game:~/bungeecord$ |

```

Lancer le reverse proxy afin de créer les fichiers.

Le stopper pour le configurer :

Ajout des serveurs dans la configuration :

```

servers:
  server1:
    motd: '&1Just another BungeeCord - Forced Host'
    address: localhost:25566
    restricted: false
  server2:
    motd: 'Server2'
    address: localhost:25567
    restricted: false
  server3:
    motd: 'Server3'
    address: localhost:25568
    restricted: false

```

Changement des ports d'écoute des différents servers dans le server.properties de chaque serveur.

Changement du port d'écoute du reverse proxy par le port défaut de Minecraft :

```

host: 0.0.0.0:25565

```

Pare-feu :

Activer le pare-feu :

Sudo ufw enable

Modifier la configuration pour désactiver l'IPV6 :

Nano /etc/default/ufw

```

# Set to yes to apply rules to support IPv6 (no means only IPv6 on loopback
# accepted). You will need to 'disable' and then 'enable' the firewall for
# the changes to take affect.
IPV6=no

```

Sudo ufw reload

Autoriser le trafic sortant et refuser le trafic entrant :

```

gamemaster@game:/etc/netplan$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
gamemaster@game:/etc/netplan$ sudo ufw default deny
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
gamemaster@game:/etc/netplan$ |

```

Configuration du serveur NAS :

Le serveur NAS est configuré en RAID 1. Pour les accès seul la machine comportant le serveur web et les jeux y a accès pour sauvegarder et redéployer les sauvegardes. Afin de faciliter l'administration un script bash peut être créé.

4. Les bonnes pratiques réalisées :

5.

Pour faciliter l'administration système, une crontab avec un script bash basique comportant les commandes suivantes :

```
Sudo apt update; sudo apt upgrade -y
```

Script s'exécutant toutes les 24h à 6 heures du matin.

Les serveurs ne sont accessibles que par clé ssh et non par mot de passe, mot de passe étant désactivé pour des raisons de sécurité.

Une amélioration possible aurait été docker, donc de mettre dans des conteneurs les différents services tel que le serveur DNS, le serveur WEB, les différents serveurs Minecraft et le reverse proxy Bungeecord.

Les mots de passe utilisateurs sont à 30 caractères.