MISE EN PLACE DE SERVEUR RESOLVEUR RECURSIF SUR UBUNTU 24 LIVE-SERVER AVEC UNBOUND SUR PROXMOX VE

Réalisé par : **SOULAIMAN Rayane**

Administrateur système, réseau et sécurité

Table des matières

1-Introduction

1.1 Contexte du projet

Pour les besoins de résoudre rapidement et de manière sécurisée les noms de domaine pour un réseau, qu'il s'agisse d'une entreprise ou d'un particulier, la mise en place d'un serveur DNS récursif est justifiée.

Un serveur DNS récursif est déployé principalement dans les environnements tels que : les réseaux internes d'entreprise pour donner aux employés un accès DNS rapide, fiable et sous contrôle et pour éviter de dépendre des DNS publics comme google ou cloudflare ; les fournisseurs d'accès internet qui déploient ces serveurs DNS pour fournir la résolution DNS à leurs abonnés ; les data centers / clouds privés pour garantir la résolution interne et résoudre les domaines publics depuis un resolveur maîtrisé ; les environnements sécurisés ou sensibles comme les institutions publiques, les banques et entreprises qui veulent contrôler le trafic DNS et permet la mise en place de politiques (DNS filtering, logs, blocage de malware) ; les réseaux isolés / DMZ etc.

Ainsi ce projet vise à implémenter deux serveurs DNS résolveur récursif au sein de machine virtuelle sur proxmox afin de fournir aux employés d'une société un accès DNS rapide et fiable et dont a le contrôle, fournir aux abonnés d'un FAI des serveurs de résolution DNS et aussi ne plus dépendre des serveurs DNS publics.

1.2 Objectifs et avantages

Les principaux objectifs de ce projet sont :

- Donner à une société un accès DNS rapide fiable et sous contrôle
- Contrôler le trafic DNS
- Eviter de dépendre des DNS publics
- Fournir aux abonnés d'un FAI la résolution DNS

Ce projet a aussi des objectifs pédagogiques tels que comprendre le rôle d'un serveur résolveur récursif dans le processus de résolution de nom, installer et configurer un serveur DNS résolveur, installer un serveur live Ubuntu, Installer une machine virtuelle sur Proxmox

L'implémentation de ce serveur permet une performance accrue avec des temps de réponses beaucoup plus rapide, la réduction du trafic externe vers les serveurs autoritaires, le filtrage DNS, la privatisation, le contrôle et la gouvernance de ce qui se passe sur le réseau, la résilience et la redondance, la confidentialité, etc.

2-Présentation du matériel et des outils

2.1 Configuration matérielle

Pour ce projet, nous avons utilisé les ressources suivantes :

Ressources	Configuration minimales	Configuration utilisés
Processeur	1 (1 sockets, 1 core)	1 (1 sockets, 1 core)
Mémoire RAM	2Go	8Go
Disque	8 Go	32 Go
Pare-feu	1	1
Interface réseau	Brigde=vmbr0	Brigde=vmbr0

2.2 Outils utilisés

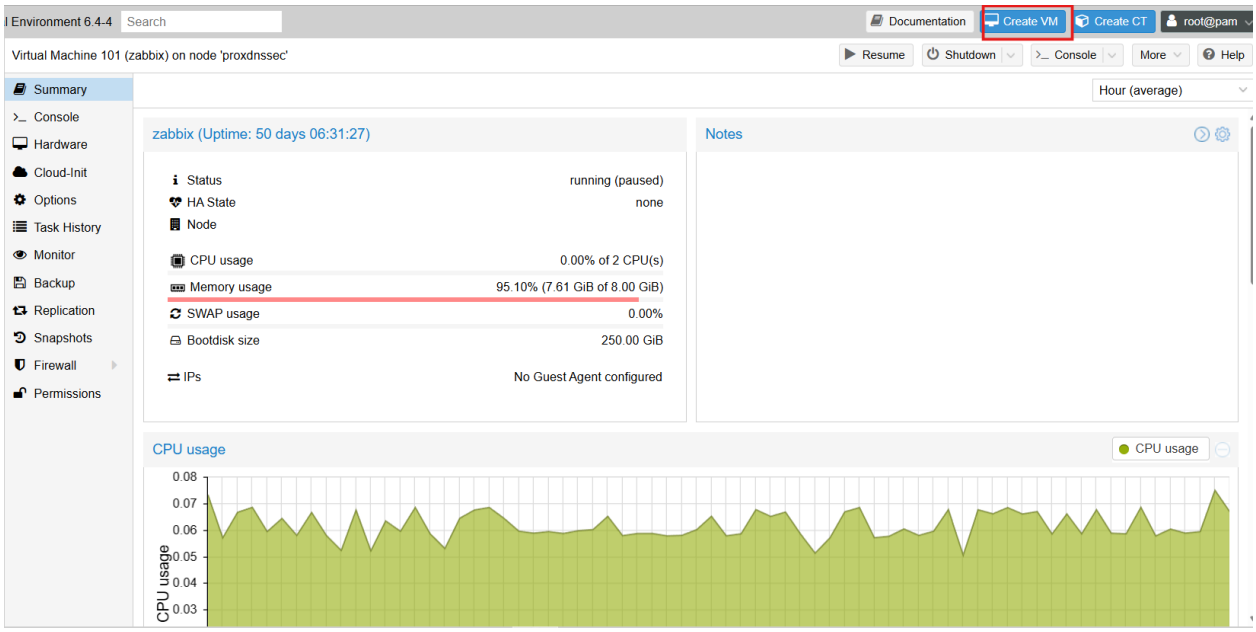
Nous verrons plus en détails lors des installations plus bas les ressources utilisées pendant la mise en place des machines virtuelles

Outils	Version	Ressources
PROXMOX VE	8.4.14 et 6.4.4	Disque 2T, RAM 62Go, 16 CPU
UBUNTU live-server	24.04	Disque 32 Go ; RAM 8Go; 1 CPU
Unbound	1.19.2	

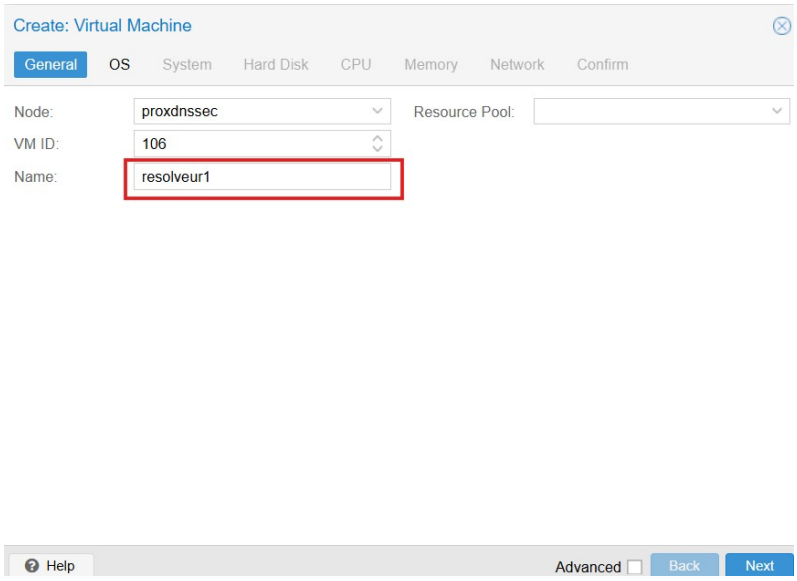
3-Création des machines

3.1 Création d'une nouvelle machine virtuelle sur proxmox

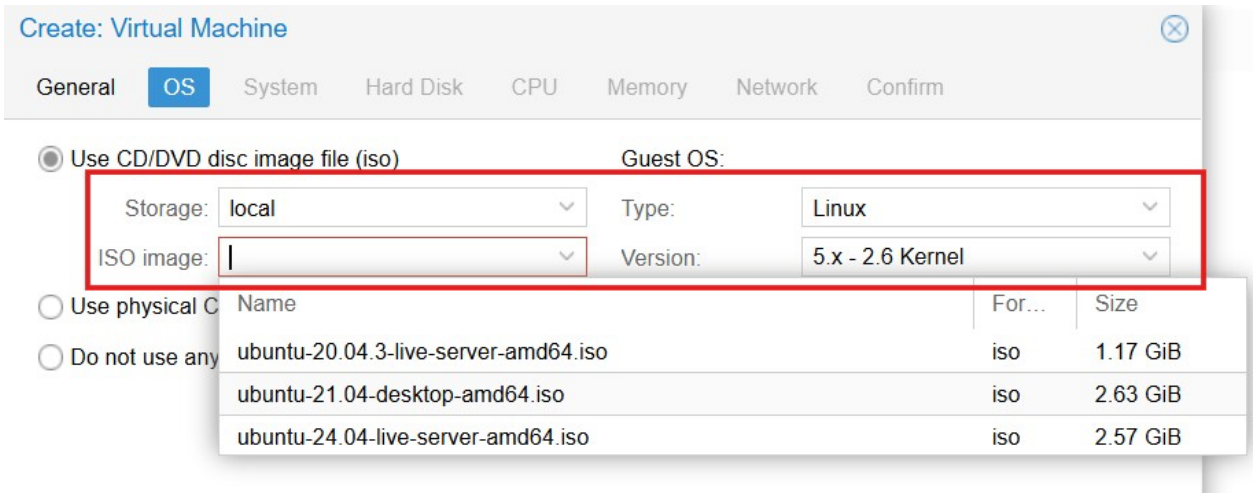
Sur le 1er serveur Proxmox on clique sur le bouton en haut à droite create VM pour créer une nouvelle machine virtuelle

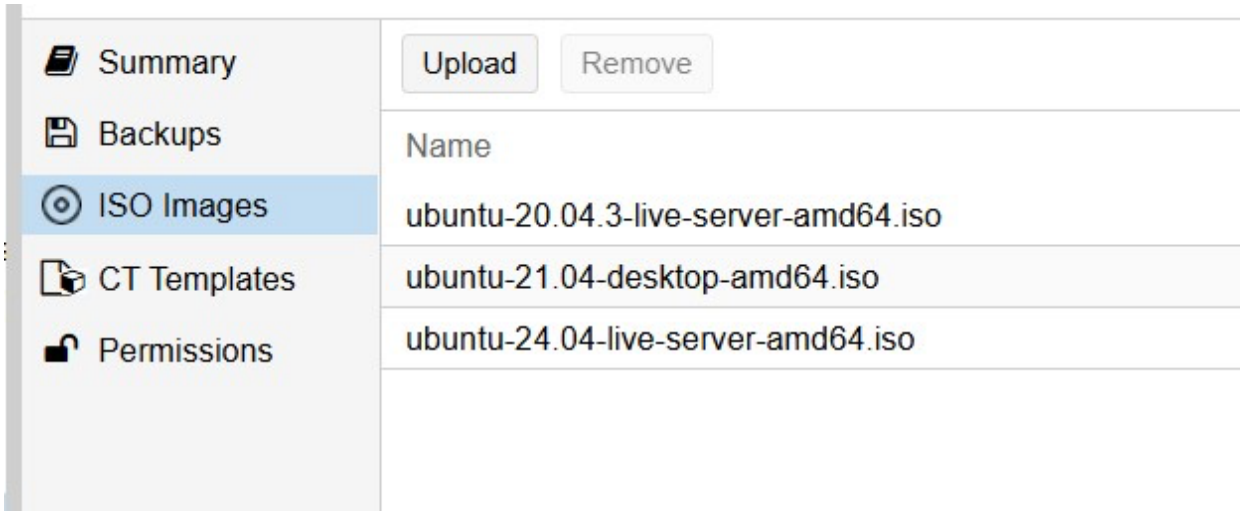


Ensuite donner le nom de votre choix à la nouvelle machine sur le point d'être installé dans le label Name. Vous pouvez aussi choisir l'ID de la machine par la même occasion

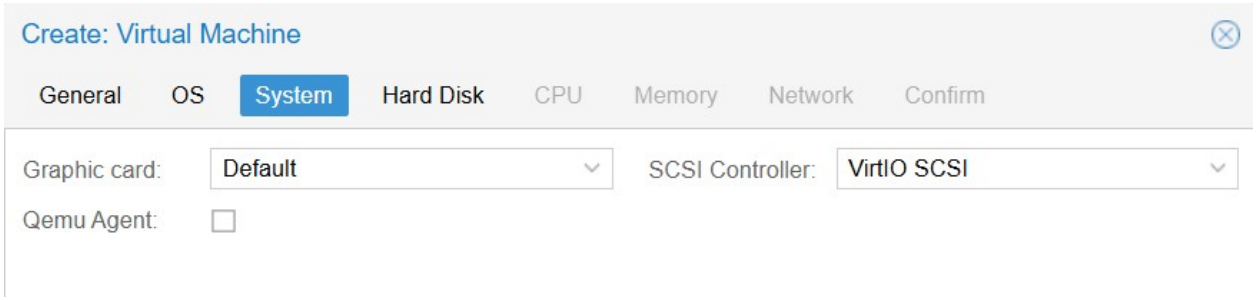


Puis choisissez l'ISO que vous souhaitez utiliser, ici ubuntu 24.04 live server qu'on a bien évidemment uploadé avant sur le serveur proxmox. Cette opération peut se faire par ligne de commande en copiant (scp) juste l'iso de notre machine locale vers le répertoire **/var/lib/vz/template/iso** sur notre serveur proxmox ou en cliquant sur upload dans le menu local par méthode graphique

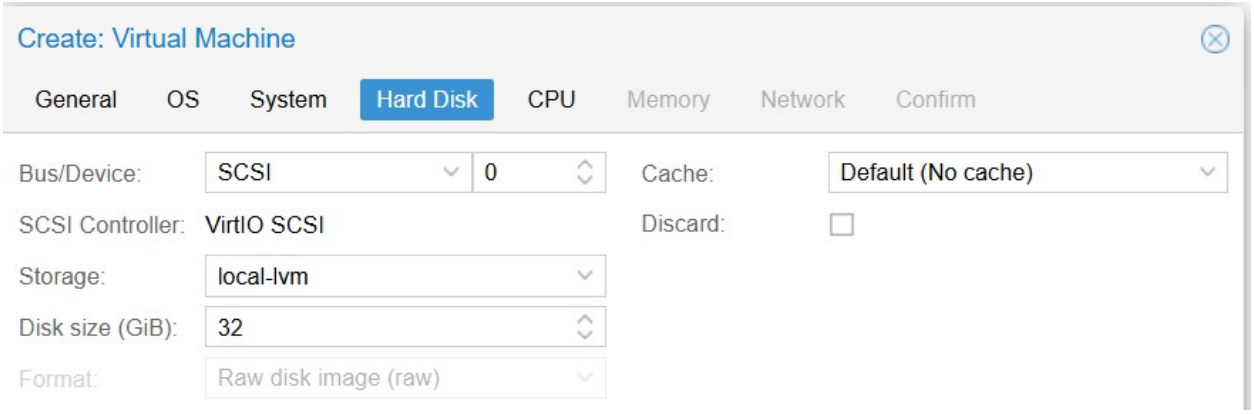




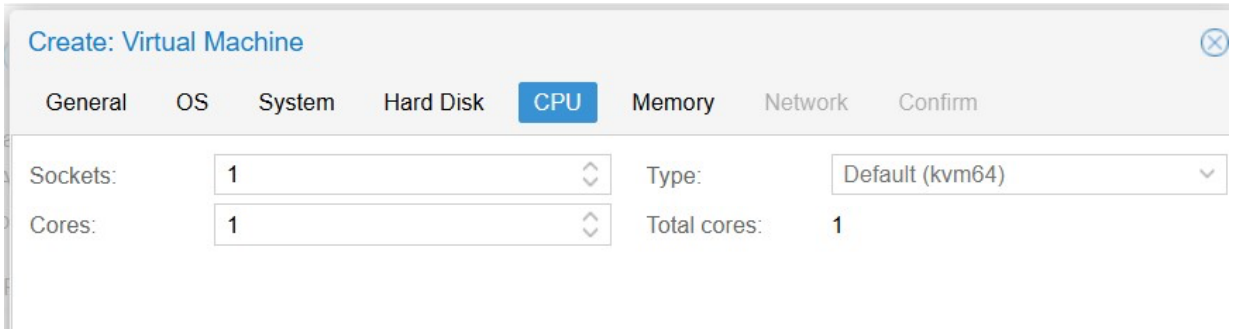
Laissez les configs par défaut dans le menu système



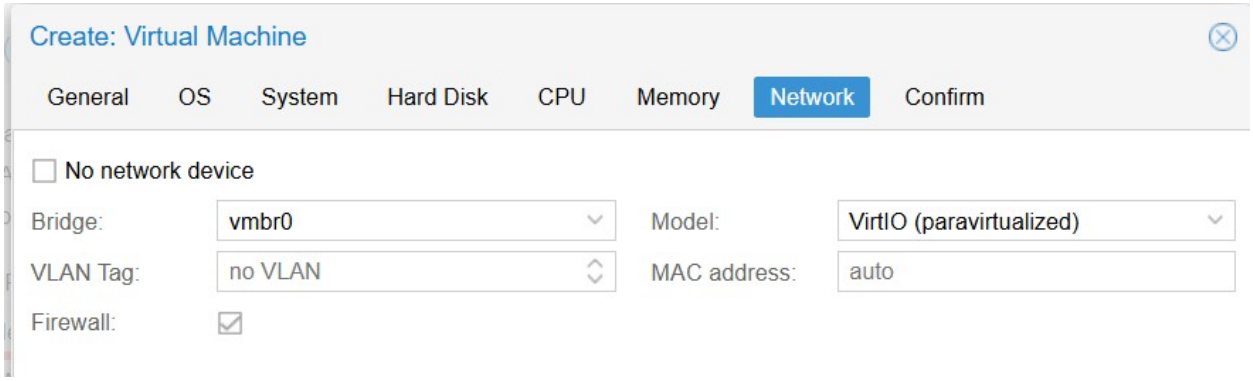
Dans le menu hard disk, choisissez les configurations de disque dur que vous souhaitez. Dans notre exemple nous avons pris les configurations précisées dans le chapitre précédent.



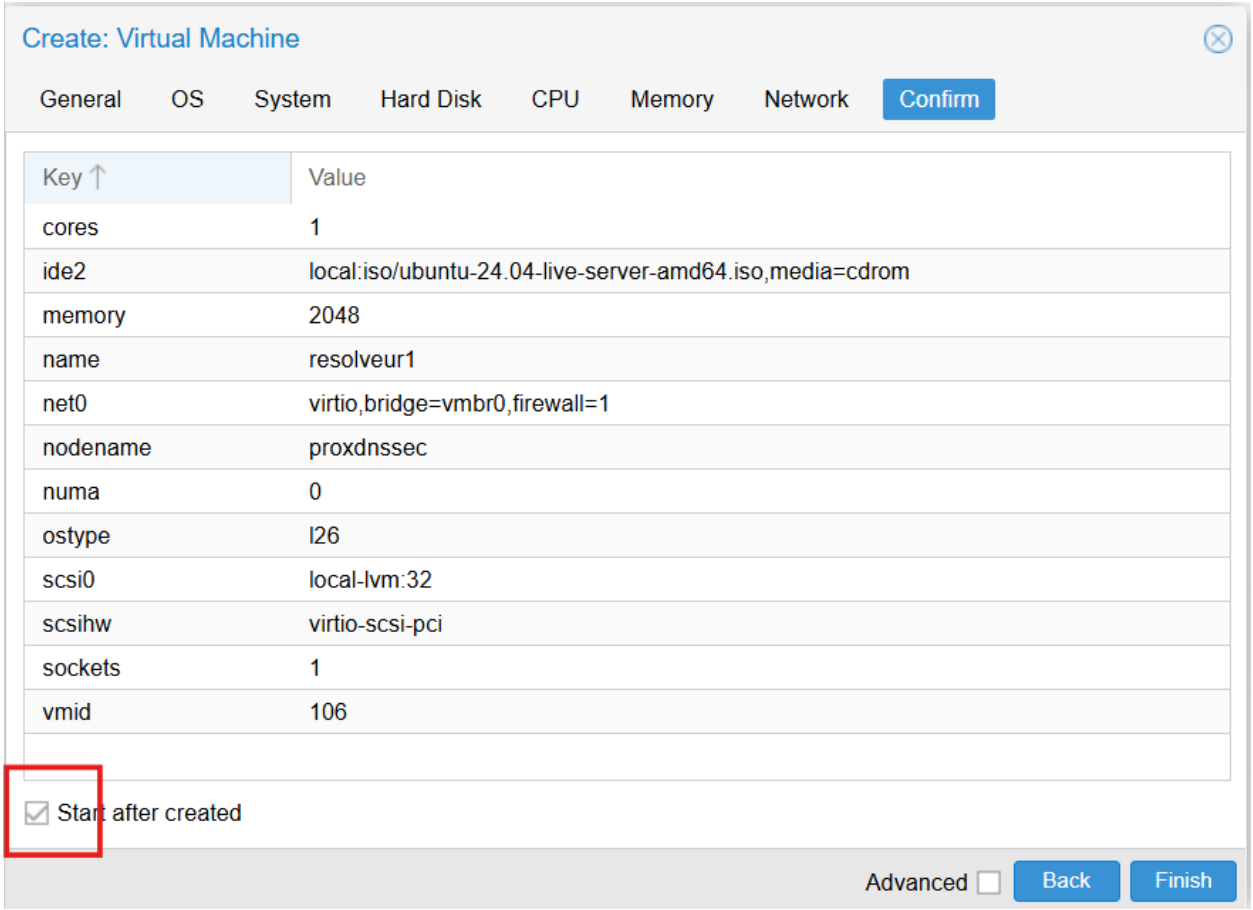
Pareil pour le CPU, la RAM



Dans le menu Network choisissez l'interface bridge à laquelle appartient l'adresse IP que vous souhaitez donner à votre machine

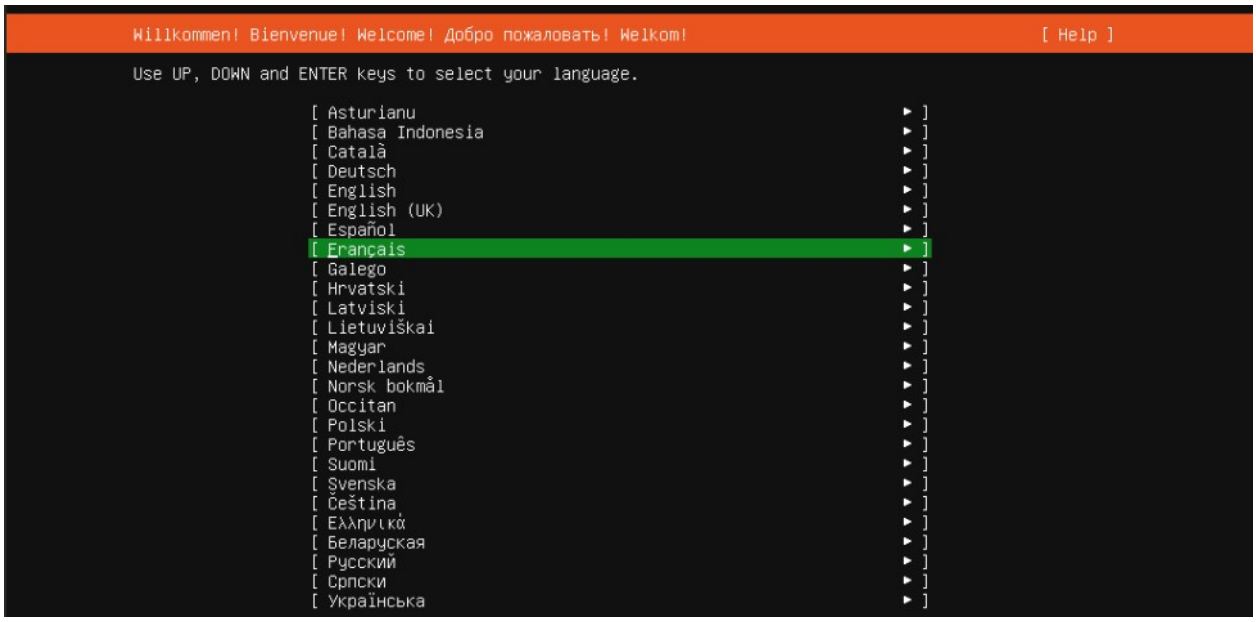


Lorsque toutes les configurations sont finies vous pouvez les revoir dans le menu **confirm** et repartir les corriger si nécessaire sinon, cliquer sur l'icône en bas à gauche **Start after created** pour lancer la machine après création.

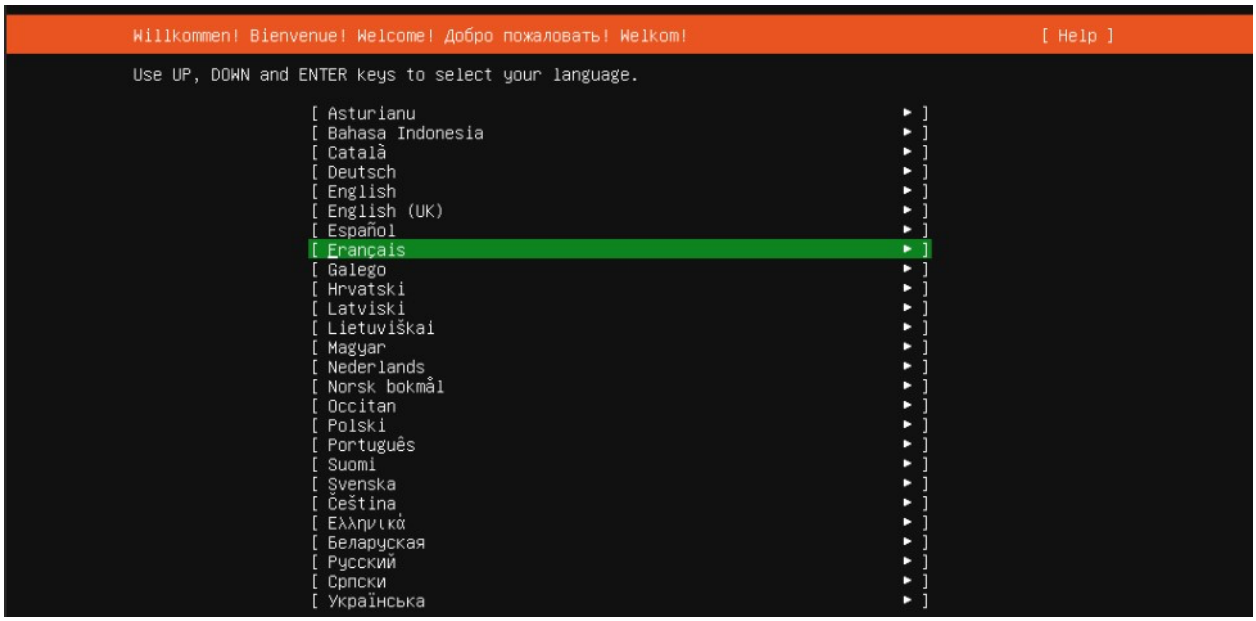


3.2 Installation de Ubuntu 24.04 LTS

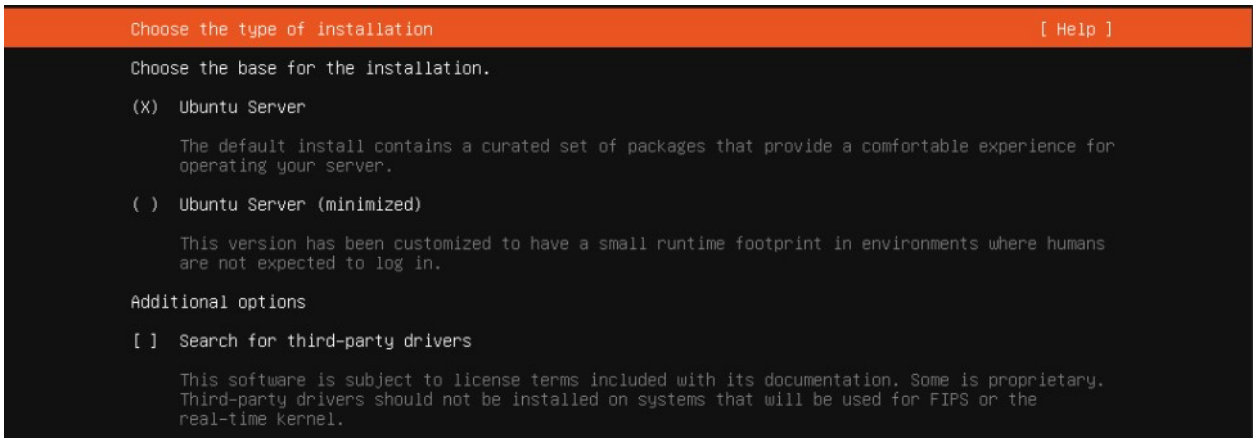
Choisir d'abord la langue,



Identifier le clavier



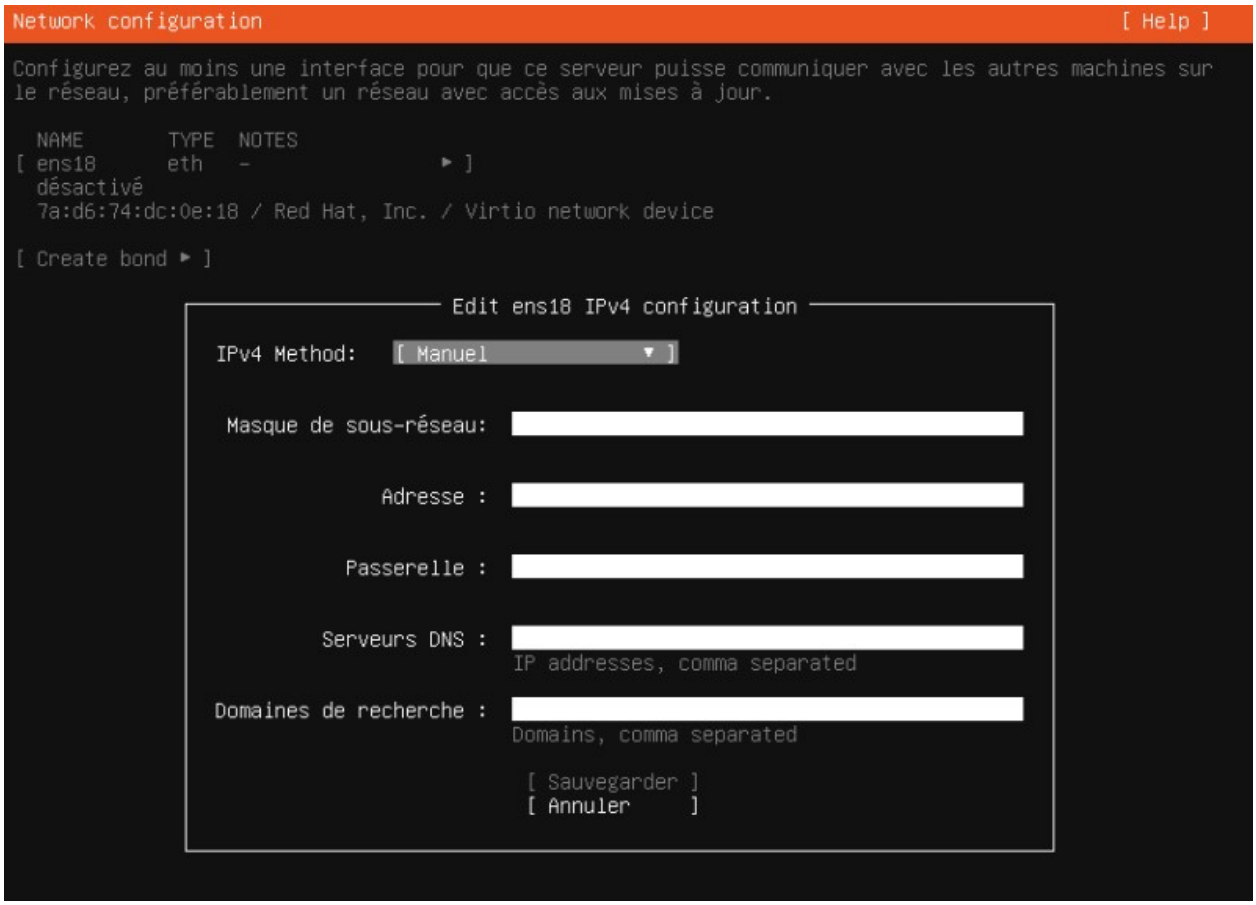
Choisir le type de serveur qu’on veut installer, dans notre cas, Ubuntu server, l’option par défaut



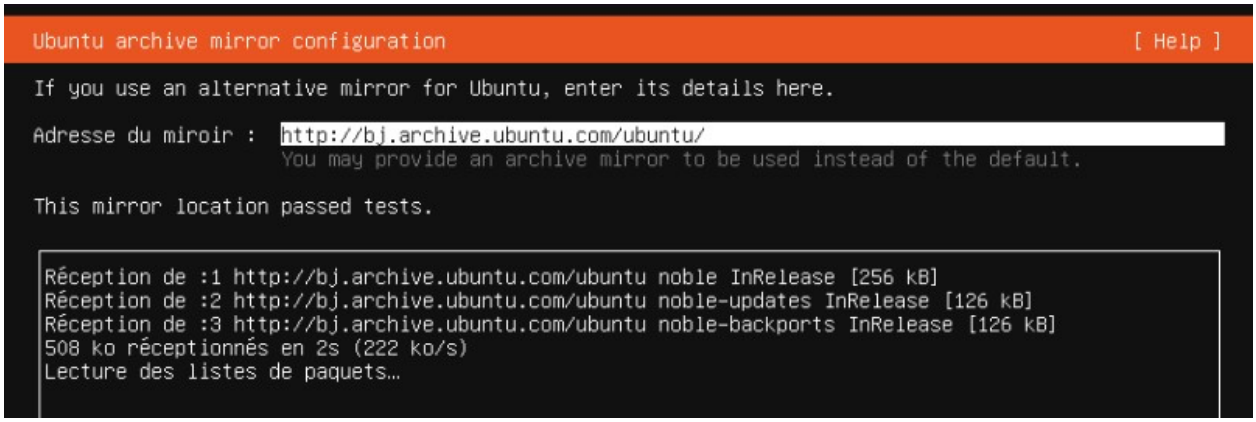
Configurer l'interface réseau pour faire les premières installations et installer unbound sur le server plus tard

Choisir Edit IPV4

Dans masque de sous-réseau, mettre l'adresse réseau sous format CIDR, 192.168.134.0/24 par exemple. Puis dans Adresse, mettre l'adresse IP qu'on souhaite donner à la machine et dans la passerelle, la passerelle du réseau. Dans serveur DNS, on va mettre 8.8.8.8 le serveur public de google. Notez bien que c'est des configurations qui seront modifiés plus tard. Puis Sauvegarder et évoluer.

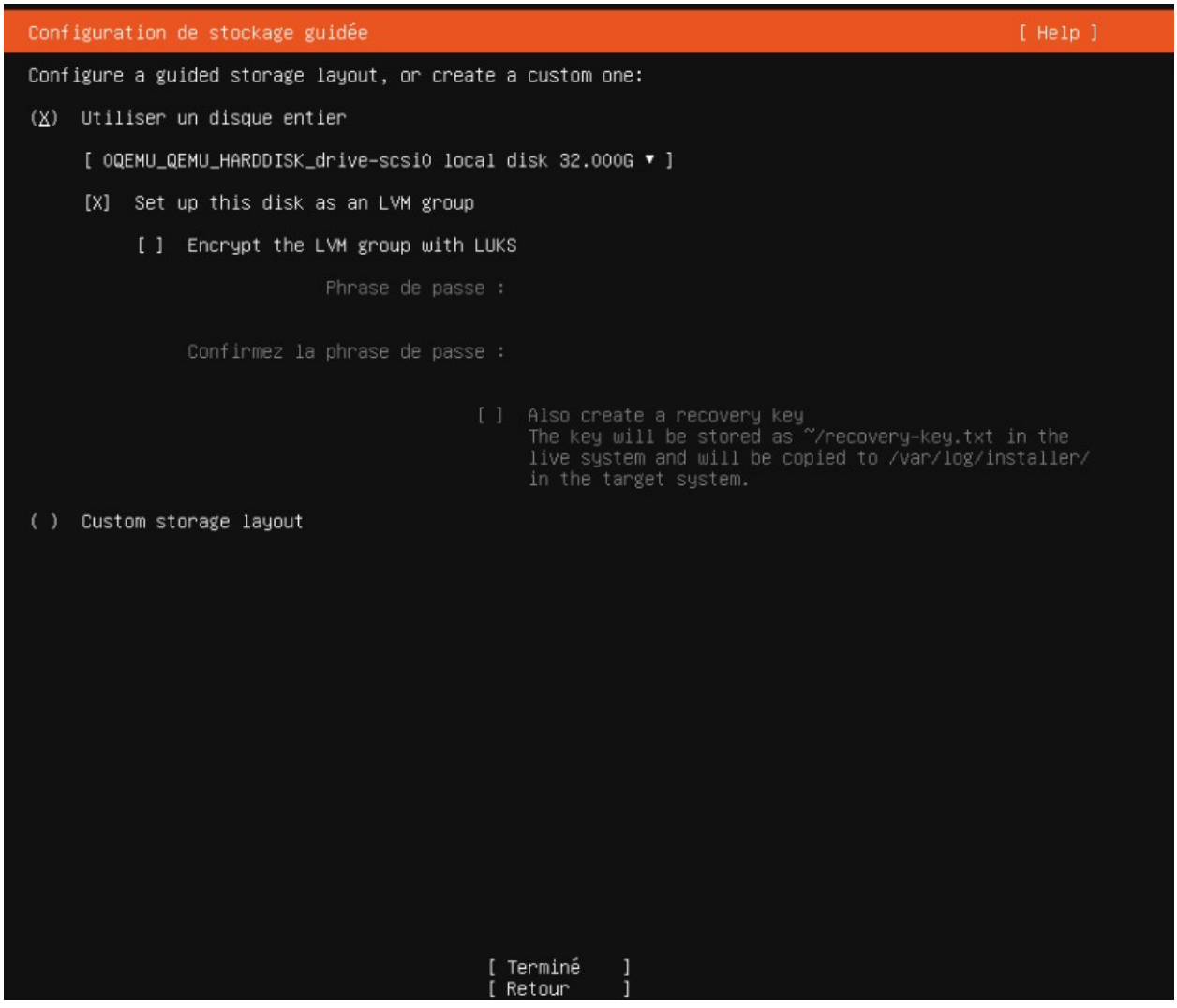


Attendre que les tests mirroir soient concluents, si non, les paramètres réseaux précédemment définis sont peut-être erronés.

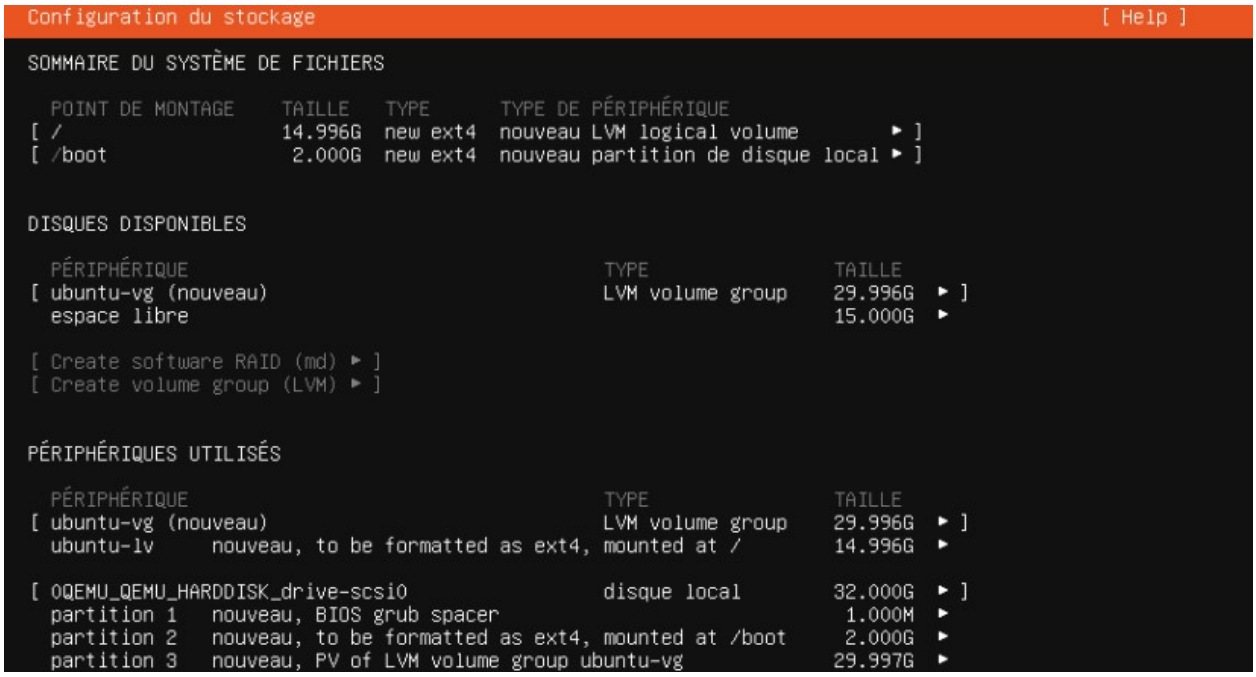


Si les tests mirroir sont effectives, choisissez continuer sans mis a jour, ici on fera la mis a jour plus tard mais elle peut aussi être faite durant l'installation

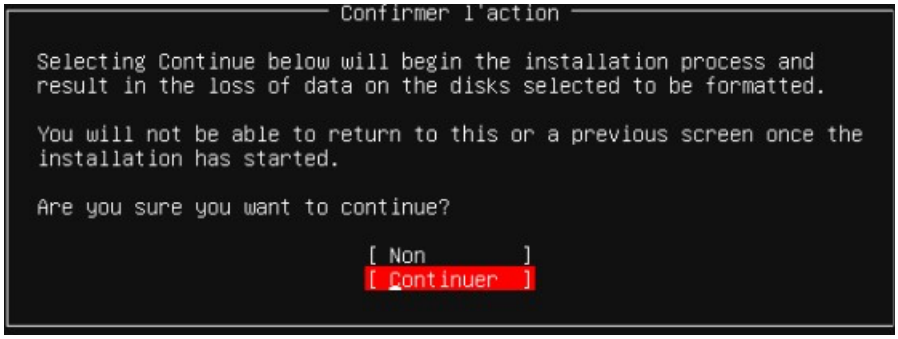
Choisir, utiliser le disque entier



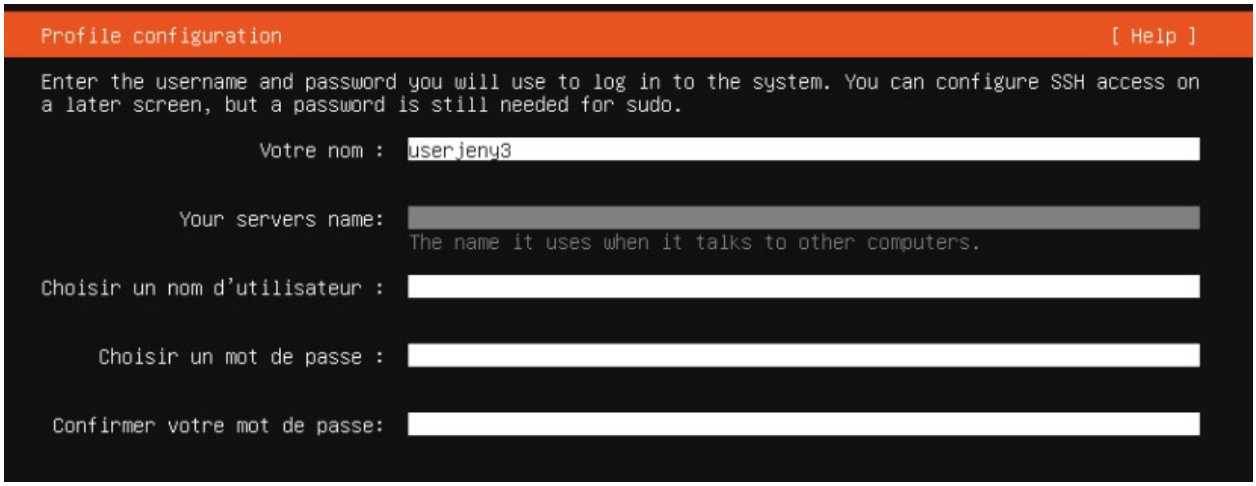
Valider la configuration du stockage



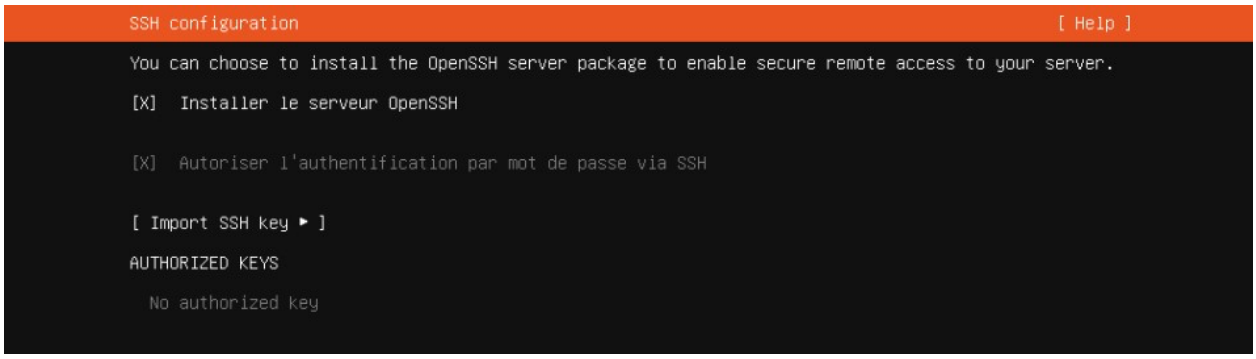
Confirmer l'action



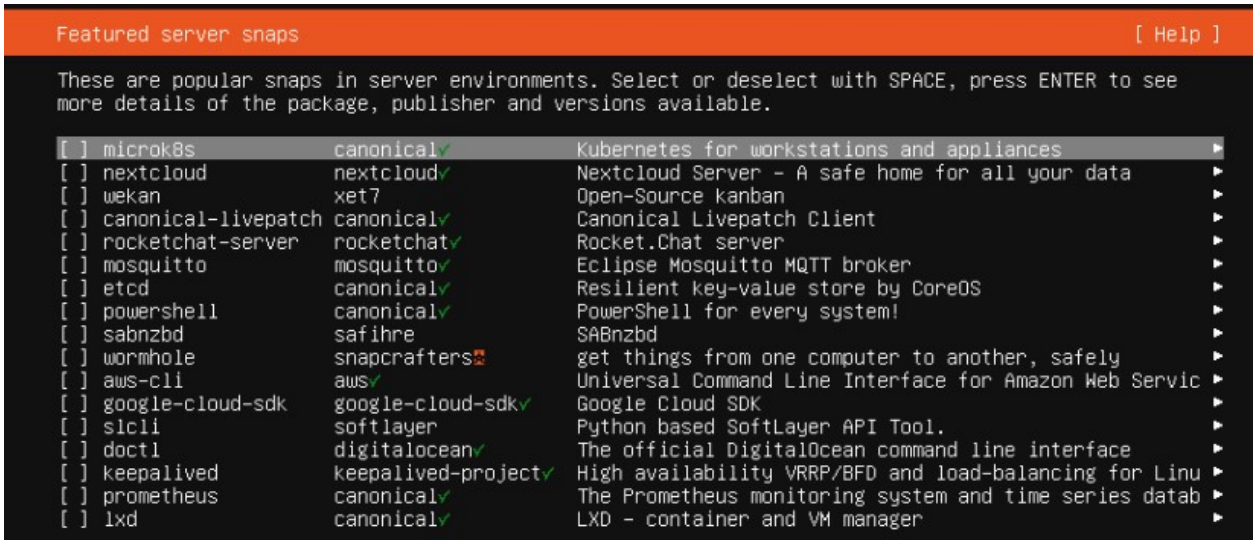
Ensuite configurer le profil utilisateur en ajouter d'abord votre nom, le nom que vous souhaitez donner au serveur, votre nom d'utilisateur et votre mot de passe



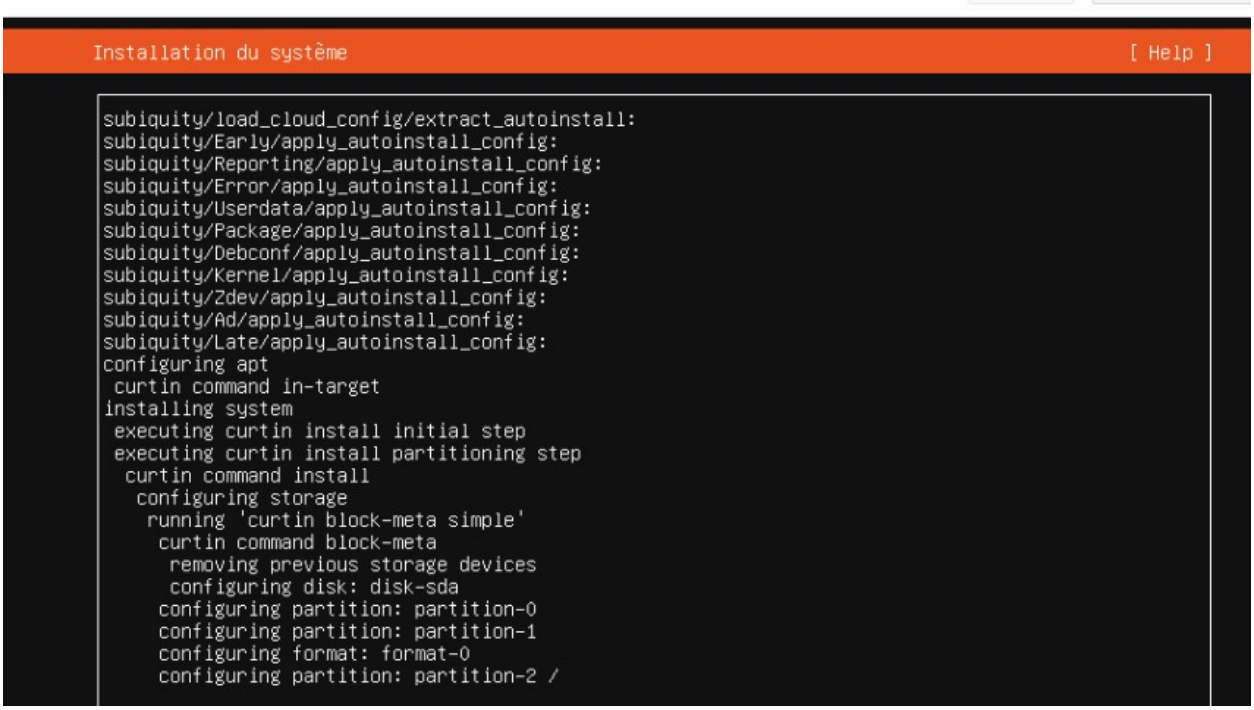
Cocher en appuyant sur espace, l'installation de SSH qui peut aussi être faite plus tard après installation, mais ici nous allons la faire en même temps.



Choisissez les services additionnels que vous souhaitez en les cochant avec espace. Ici nous n'en voulons aucun donc on n'en sélectionne aucun et on évolue.



Ensuite patientez pendant l'installation du système et le système démarrera à la fin.



4-Installation d'Unbound

Tout d'abord comme précisé lors de l'installation nous allons d'abord lancer la mise à jour de la base de données locale des paquets disponible avec la commande :

Sudo apt update

Ensuite installer Unbound

Sudo apt install unbound

Ubuntu 24 utilise systemd-resolved, qui écoute sur 127.0.0.53:53.Il doit être désactivé pour qu'Unbound puisse utiliser le port 53. Pour ce faire, on utilise donc les commandes suivantes :

sudo systemctl stop systemd-resolved : pour stopper le service **sudo**

systemctl disable systemd-resolved : pour désactiver le service

Vous pouvez ensuite lancer un **systemctl status systemd-resolved** afin de vous assurer que le service est bien arrêté

Modifier le fichier **resolv.conf** comme suit :

Mettre en commentaire les configurationset ajoutés les lignes suivantes :

nameserver 127.0.0.1

search localdomain

nameserver 127.0.0.1 pour préciser que le DNS est utilisé par la machine elle-même et search localdomaine qui définit le suffixe DNS qui sera automatiquement ajouté aux requêtes non qualifiées par exemple :

si **ping nano** est lancé ,le système fait, **ping nano.localdomain** Puis

modifier le fichier de configuration Unbound avec la commande :

sudo nano /etc/unbound/unbound.conf

et ajouter les lignes suivantes :

```
server:
    # Écoute sur toutes les interfaces
    interface: 0.0.0.0
    interface: ::0

    # Réseaux autorisés
    access-control: 127.0.0.1/8 allow
    access-control: 81.91.234.240/28 allow
    access-control: 0.0.0.0/0 allow
    access-control: ::0/0 allow

    # Sécurité
    hide-identity: yes
    hide-version: yes
    qname-minimisation: yes
    harden-glue: yes
    #harden-dnssec-stripped: yes

    # DNSSEC
    # auto-trust-anchor-file: "/var/lib/unbound/root.key"

    # Performance
    num-threads: 2
    prefetch: yes
    prefetch-key: yes

    # TTL cache
    cache-min-ttl: 60
    cache-max-ttl: 86400

    # Logging (désactivé par défaut car chroot)
    # logfile: /var/log/unbound/unbound.log
    verbosity: 1

remote-control:
    control-enable: yes
```

- Server : Cela indique à Unbound d'écouter sur **toutes les adresses IPv4** et **IPv6** disponibles.

0.0.0.0 = toutes les interfaces IPv4

::0 = toutes les interfaces IPv6

- Access control : les réseaux autorisés, y mettre les adresses des réseaux dont on veut autoriser les requêtes. 0.0.0.0/0 allow permet d'accepter toutes les requêtes.

- Sécurité

  Hide-identity : Empêche Unbound d'afficher son nom dans une requête CHAOS, Protection contre reconnaissance/scan.

  Hide-version : Empêche d'afficher la version d'Unbound, Rend les attaques ciblées plus difficiles.

  qname-minimisation : Important pour la vie privée, Unbound envoie le minimum d'informations nécessaires aux serveurs DNS.
  Exemple : pour www.google.com, il n'enverra pas "www.google.com" au serveur .com,

  mais seulement "google.com". harden-glue : Empêche les attaques par **Fake Glue**

  Records (empoisonnement DNS). Il vérifie que les NS/glue IP données par un TLD

  correspondent bien à la zone correcte.

  Harden-dnssec-stripped : Empêche d'accepter des réponses DNS qui ont eu leurs signatures DNSSEC supprimées. A décommenter si vous utilisez DNSSEC

- DNSSEC
  auto-trust-anchor-file: "/var/lib/unbound/root.key" Si
  vous activez :
  Unbound valide toutes les réponses DNS
  Utilise la clé racine (root key) pour vérifier les signatures
  Augmente la sécurité mais peut augmenter les erreurs si la configuration n'est pas propre
- Performance

Num-threads : 2

Nombre de threads de traitement des requêtes. En

général : nombre de CPU / 2 est optimal.

Prefetch : yes

Quand une entrée du cache est presque expirée, Unbound la met à jour **en background**. Améliore la vitesse perçue par les clients.

Prefetch-key : yes

Précharge aussi les **clefs DNSSEC** avant expiration. Optimise les performances avec DNSSEC si DNSSEC est utilisé. Dans notre cas-ci, non

- TTL du cache

Cache-min-ttl : 60 , Unbound garde les réponses au moins **60 secondes**, même si le TTL original est plus bas.

Cache-max-ttl : 86400 , Durée maximale du cache = **1 jour**. Bon compromis entre performance et fraîcheur.

- Logging

Verbosity : 1 , logs minimum • Remote-control control-enable: yes, permet de

piloter Unbound avec la commande unbound-control

Les configurations terminées, on peut vérifier que tout est en ordre avec la commande

Sudo unbound-checkconf

Puis redémarrer le service unbound avec les commandes suivantes afin de permettre à unbound de lire le contenu du nouveau fichier de configuration, l'activer et vérifier l'état du service :

sudo systemctl restart unbound

sudo systemctl enable unbound

Vérifier :

sudo systemctl status unbound

Vérifiez après que le serveur écoute sur le port 53 avec la commande : sudo

ss -tulpn | grep :53

Vous devez voir :

unbound LISTEN 0 ... 0.0.0.0:53

Nous pouvons ainsi faire des tests :

Depuis la machine elle-même : **dig**

google.com @127.0.0.1

Depuis une autre machine du réseau :

dig google.com @192.168.134.136

192.168.134.136 étant l'adresse du serveur résolveur

Si une réponse DNS apparaît alors le serveur DNS récursif fonctionne.

Pour mettre en place un serveur redondant / deuxième résolveur récursif, il suffit de créer une deuxième machine avec les mêmes configurations mais sur un réseau différent afin d'assurer une meilleure disponibilité.

5-Conclusion

Ainsi, un serveur DNS récursif affirme son utilité pour toute entreprise souhaitant avoir un meilleur contrôle sur son trafic DNS. Il se révèle encore plus utile pour les fournisseurs d'accès à internet qui ont besoin d'assurer un bon trafic et une résilience à leurs abonnés.

La configuration étudiée met en place un **serveur DNS récursif performant, sécurisé et adapté à un réseau local** grâce à Unbound. Les directives principales définissent où le service écoute, quels clients sont autorisés, et renforcent la sécurité via des mécanismes comme la minimisation des requêtes, la protection contre les attaques DNS et l'occultation des informations sensibles. Les paramètres de performance, comme le prefetch et la gestion des threads, optimisent la rapidité tandis que le contrôle des TTL assure un cache efficace. Enfin, l'activation du module de remote-control facilite l'administration quotidienne et le suivi du resolver.

En résumé, cette configuration Unbound offre une **infrastructure DNS fiable, rapide, résiliente et hautement sécurisée**, adaptée aussi bien aux environnements professionnels que personnels, tout en permettant une gestion simple et centralisée du service.