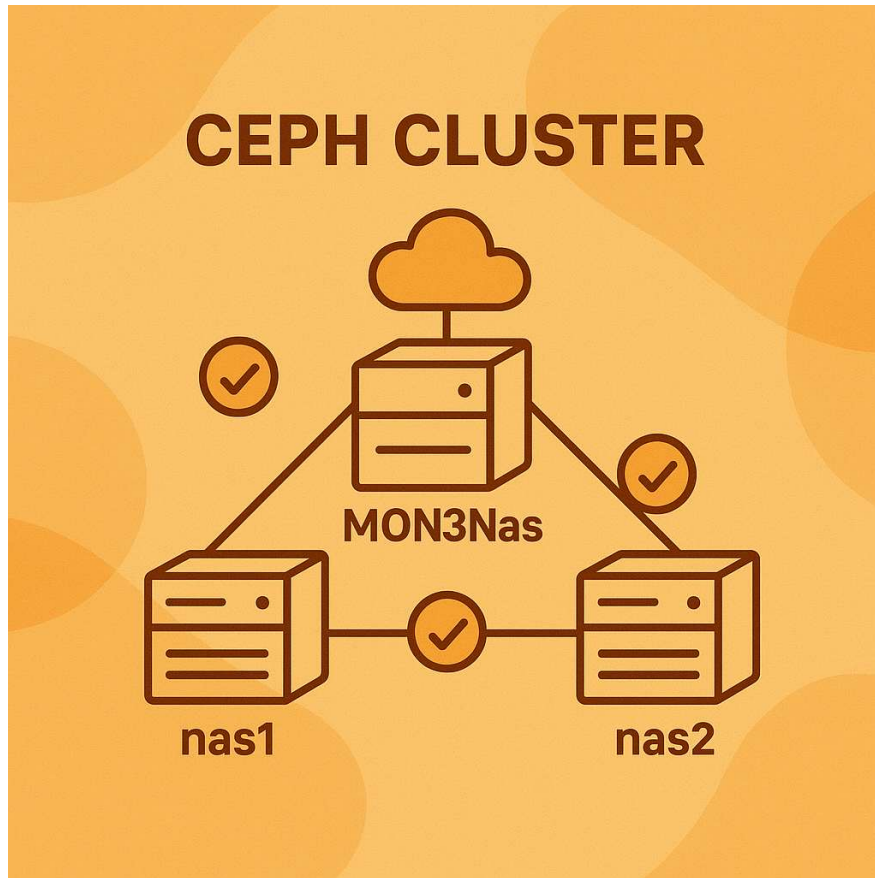

MISE EN PLACE D'UN CLUSTER CEPH AVEC 3 MACHINES – EPREUVE E6 BTS SIO SISR



CONTEXTE DU PROJET - MISE EN PLACE D'UNE INFRASTRUCTURE DE STOCKAGE HAUTE DISPONIBILITE

Dans le cadre du besoin du cabinet comptable Perlier d'assurer la continuité de service et la sécurité des données des clients, nous avons conçu et déployé une infrastructure de stockage redondante, basée sur **Ceph**, une solution distribuée, résiliente et évolutive.

L'environnement repose sur un **cluster haute disponibilité**, réparti sur trois nœuds :

- **NAS1 (192.168.30.13)** – Nœud principal avec 500 Go de stockage (OSD)
- **NAS2 (192.168.30.14)** – Nœud secondaire avec 500 Go de stockage (OSD)
- **MON3Nas (192.168.30.21)** – Nœud moniteur dédié, utilisé comme **arbitre** (tiebreaker) pour garantir le **quorum** même en cas de panne d'un des deux nœuds de stockage

L'objectif était de mettre à disposition une **solution de stockage centralisée, redondante et tolérante aux pannes**, capable d'assurer :

- Une **accessibilité continue des données**, même en cas de défaillance d'un nœud
- Une **gestion simplifiée** grâce à l'outil intégré **cephadm**
- Une **scalabilité** permettant de rajouter facilement des nœuds ou du stockage

Ce projet s'inscrit dans une démarche de professionnalisation orientée **qualité de service, résilience et modernisation de l'infrastructure**.

1. PRE-REQUIS ET PREPARATION

Les machines doivent être sur un réseau commun, synchronisées via NTP (ou configuration de l'horodatage manuelle), et disposées ainsi :

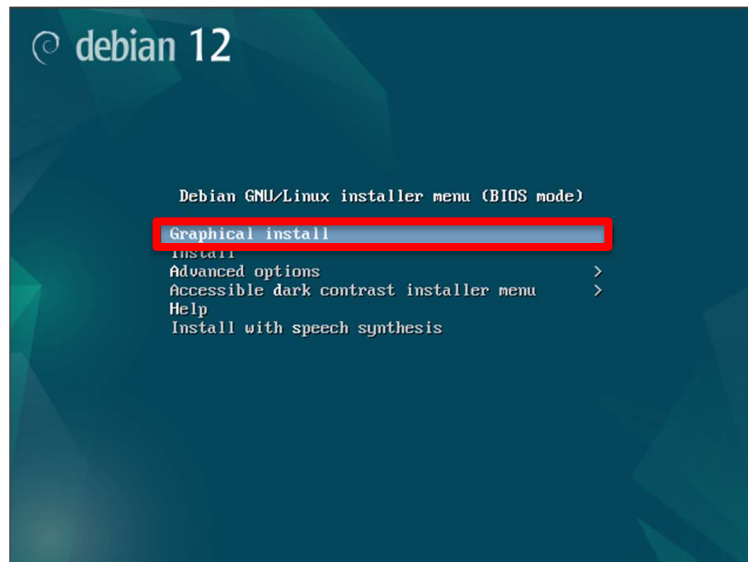
- Debian 12 installé sur chaque machine
- Paquets nécessaires : podman ou docker, curl, chrony, openssh-server, sudo

Installation de Debian 12 :

Télécharger le .iso afin d'installer Debian 12 sur les deux machines

→ <https://www.debian.org/index.fr.html>

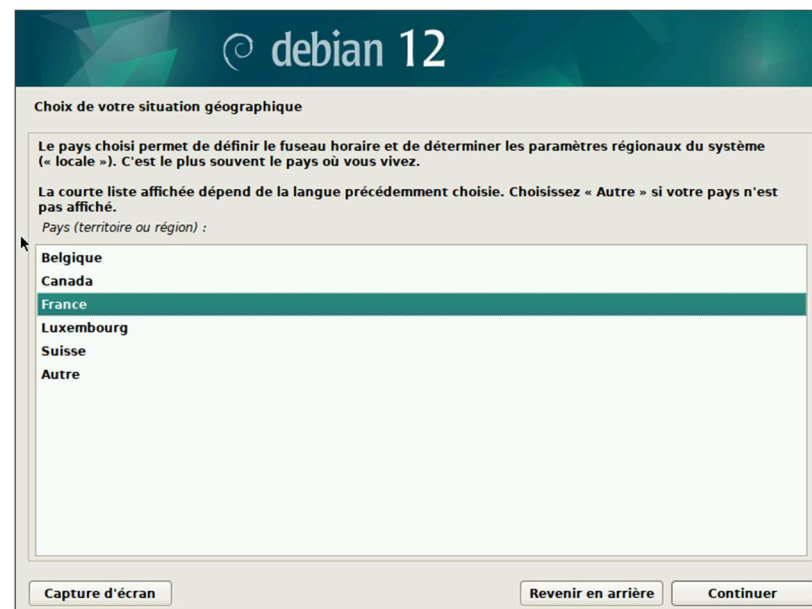
Lancer le .iso



Sélectionner « Graphical install »



Sélectionner la langue



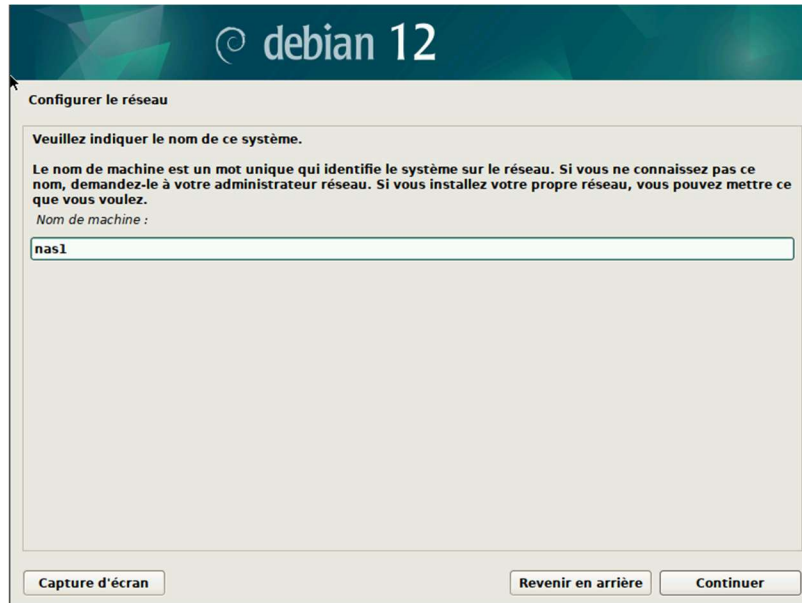
Sélectionner le pays



Sélectionner la disposition du clavier



Si un DHCP est configuré l'interface principale récupère une configuration IP



debian 12

Configurer le réseau

Veuillez indiquer le nom de ce système.

Le nom de machine est un mot unique qui identifie le système sur le réseau. Si vous ne connaissez pas ce nom, demandez-le à votre administrateur réseau. Si vous installez votre propre réseau, vous pouvez mettre ce que vous voulez.

Nom de machine :

nas1

Capture d'écran Revenir en arrière Continuer

Choisir le hostname



debian 12

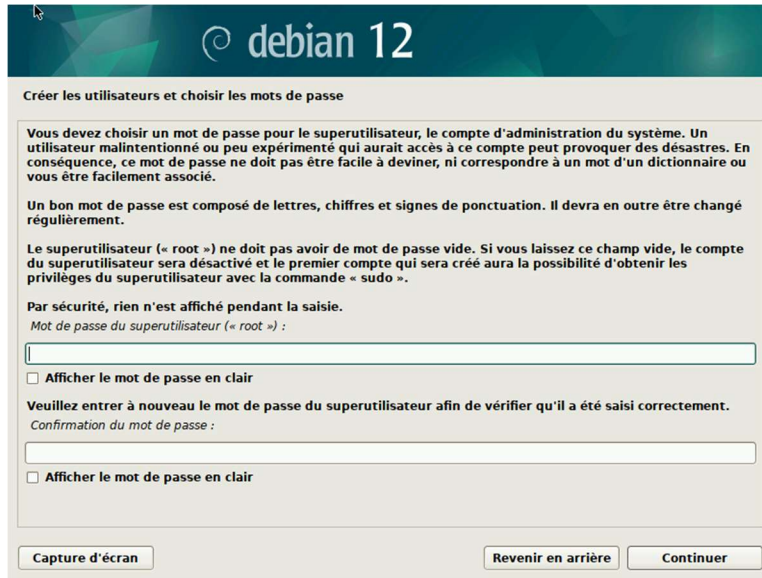
Configurer le réseau

Le domaine est la partie de l'adresse Internet qui est à la droite du nom de machine. Il se termine souvent par .com, .net, .edu, ou .org. Si vous paramétrez votre propre réseau, vous pouvez mettre ce que vous voulez mais assurez-vous d'employer le même nom sur toutes les machines.

Domaine :

Capture d'écran Revenir en arrière Continuer

Laisser le nom de domaine vide à cette étape



debian 12

Créer les utilisateurs et choisir les mots de passe

Vous devez choisir un mot de passe pour le superutilisateur, le compte d'administration du système. Un utilisateur malintentionné ou peu expérimenté qui aurait accès à ce compte peut provoquer des désastres. En conséquence, ce mot de passe ne doit pas être facile à deviner, ni correspondre à un mot d'un dictionnaire ou vous être facilement associé.

Un bon mot de passe est composé de lettres, chiffres et signes de ponctuation. Il devra en outre être changé régulièrement.

Le superutilisateur (« root ») ne doit pas avoir de mot de passe vide. Si vous laissez ce champ vide, le compte du superutilisateur sera désactivé et le premier compte qui sera créé aura la possibilité d'obtenir les privilèges du superutilisateur avec la commande « sudo ».

Par sécurité, rien n'est affiché pendant la saisie.

Mot de passe du superutilisateur (« root ») :

☐ Afficher le mot de passe en clair

Veuillez entrer à nouveau le mot de passe du superutilisateur afin de vérifier qu'il a été saisi correctement.

Confirmation du mot de passe :

☐ Afficher le mot de passe en clair

Capture d'écran Revenir en arrière Continuer

Entre un mot de passe pour le super-utilisateur **root**



debian 12

Créer les utilisateurs et choisir les mots de passe

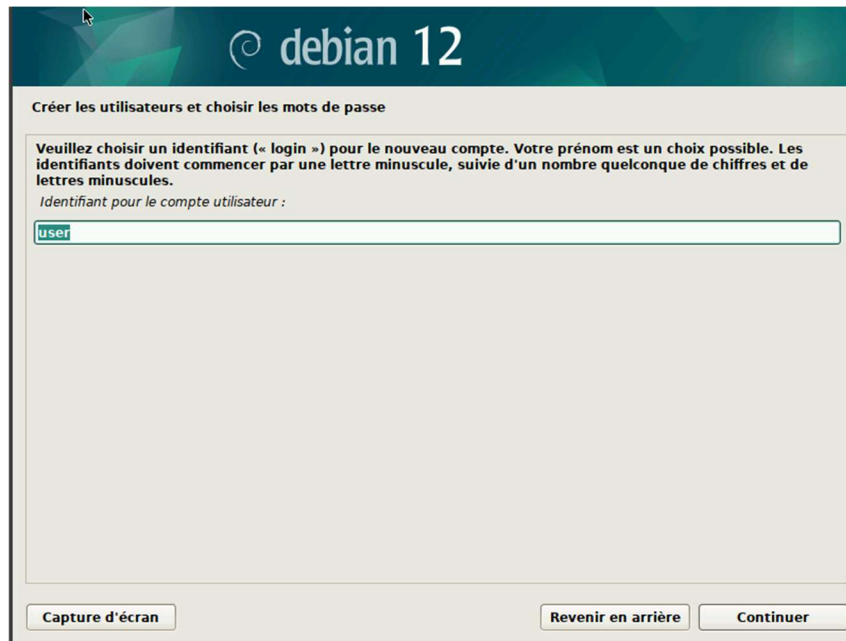
Un compte d'utilisateur va être créé afin que vous puissiez disposer d'un compte différent de celui du superutilisateur (« root »), pour l'utilisation courante du système.

Veuillez indiquer le nom complet du nouvel utilisateur. Cette information servira par exemple dans l'adresse d'origine des courriels émis ainsi que dans tout programme qui affiche ou se sert du nom complet. Votre propre nom est un bon choix.

Nom complet du nouvel utilisateur :

Capture d'écran Revenir en arrière Continuer

Créer un utilisateur (sudoer) → ex : user ou ceph



debian 12

Créer les utilisateurs et choisir les mots de passe

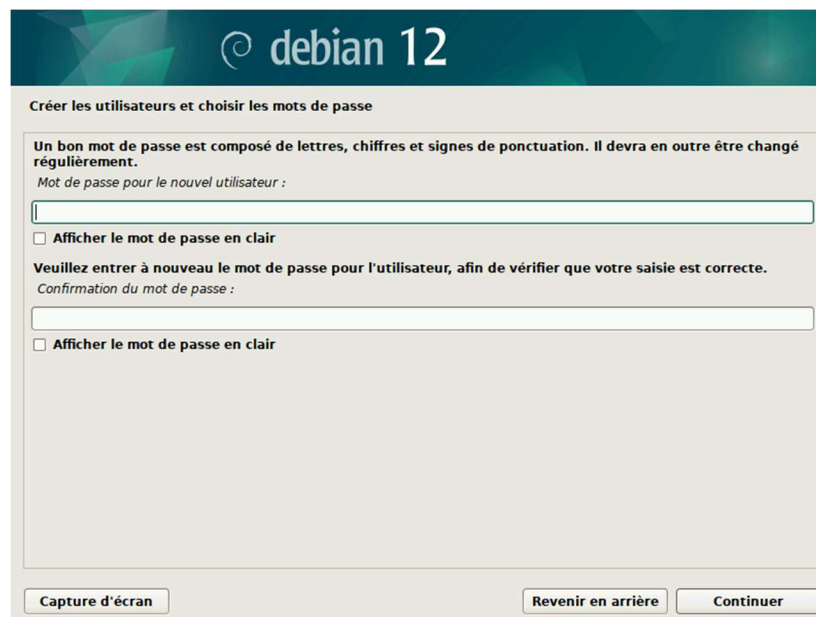
Veuillez choisir un identifiant (« login ») pour le nouveau compte. Votre prénom est un choix possible. Les identifiants doivent commencer par une lettre minuscule, suivie d'un nombre quelconque de chiffres et de lettres minuscules.

Identifiant pour le compte utilisateur :

user

Capture d'écran Revenir en arrière Continuer

Lui donner un nom d'utilisateur



debian 12

Créer les utilisateurs et choisir les mots de passe

Un bon mot de passe est composé de lettres, chiffres et signes de ponctuation. Il devra en outre être changé régulièrement.

Mot de passe pour le nouvel utilisateur :

☐ Afficher le mot de passe en clair

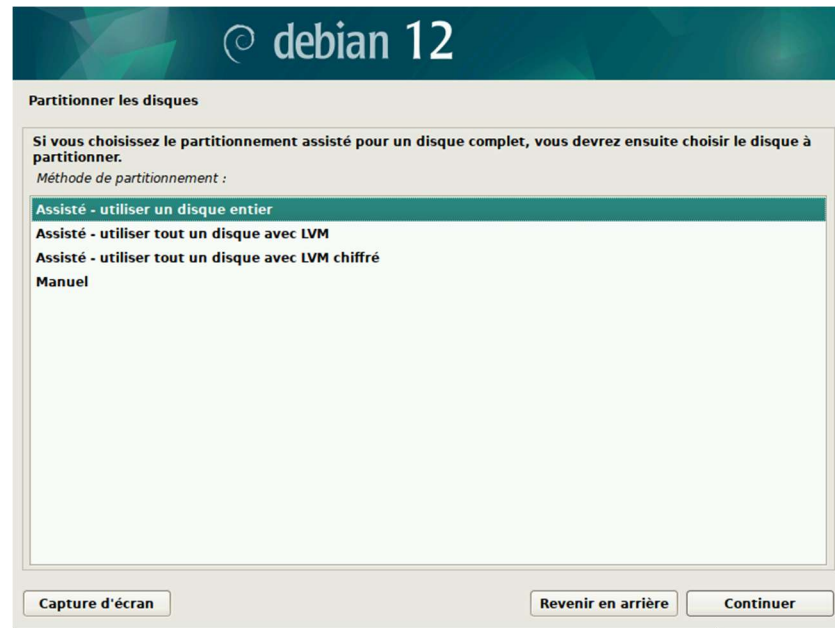
Veuillez entrer à nouveau le mot de passe pour l'utilisateur, afin de vérifier que votre saisie est correcte.

Confirmation du mot de passe :

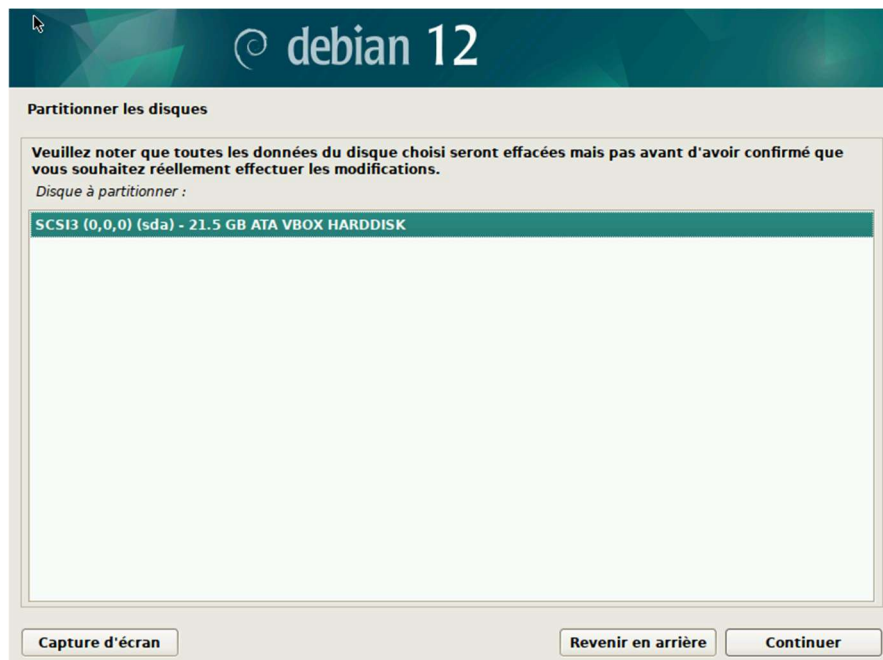
☐ Afficher le mot de passe en clair

Capture d'écran Revenir en arrière Continuer

Définir un mot de passe pour l'utilisateur



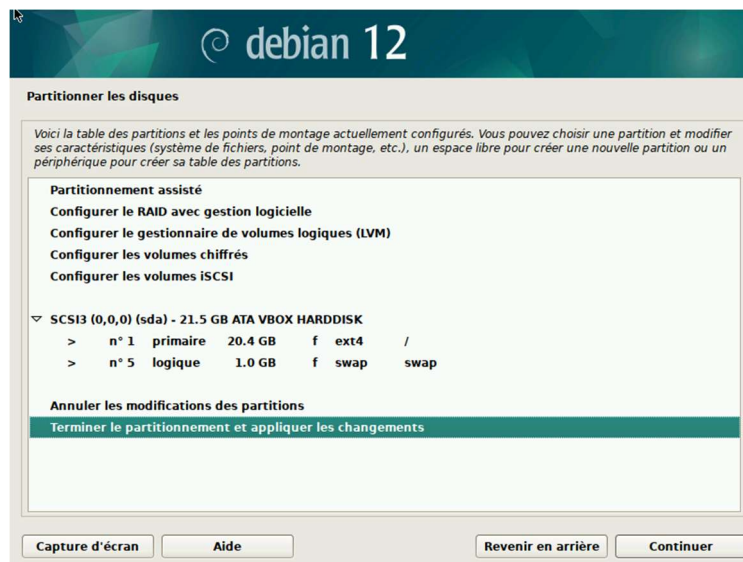
Sélectionner « Assisté – utiliser un disque entier »



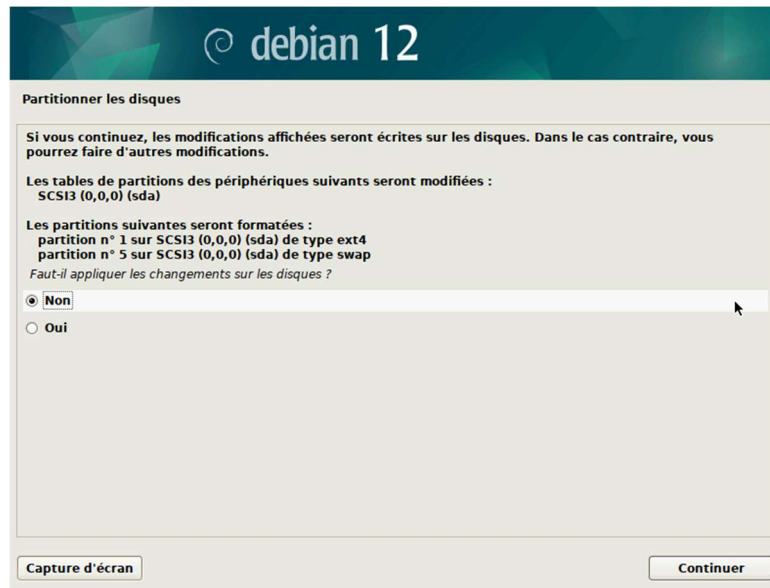
Sélectionner le disque pour l'installation de l'OS



Sélectionner « Tout dans une seule partition »



Terminer le partitionnement



Sélectionner « **Oui** »



Les partitions se formattent



Debian s'installe

Sur chaque machine :

Mettre à jour le système :

apt update && apt upgrade -y

Installer les dépendances nécessaires :

apt install -y curl gnupg2 lsb-release podman

```
root@nail:~# apt install -y curl gnupg2 lsb-release podman
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
lsb-release est déjà la version la plus récente (12.0-1).
lsb-release passe en « installé manuellement ».
Les paquets supplémentaires suivants seront installés :
  aardvark-dns buildah catatonit common containernetworking-plugins crun fuse-overlayfs
  golang-github-containers-common golang-github-containers-image iptables libipset2 libslirp libsubid0 netavark
  slirpnetns uidmap
Paquets supprimés :
  containers-storage firewalld docker-compose
Les NOUVEAUX paquets suivants seront installés :
  aardvark-dns buildah catatonit common containernetworking-plugins curl fuse-overlayfs gnupg2
  golang-github-containers-common golang-github-containers-image iptables libipset2 libslirp libsubid0 netavark
  podman slirpnetns uidmap
0 mis à jour, 19 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 27,7 Mo dans les archives.
Après cette opération, 117 Mo d'espace disque supplémentaires seront utilisés.
Réception de : 1 http://deb.debian.org/debian bookworm/main amd64 netavark amd64 1.4.0-3 [1 836 kB]
Réception de : 2 http://deb.debian.org/debian bookworm/main amd64 aardvark-dns amd64 1.4.0-3 [707 kB]
Réception de : 3 http://deb.debian.org/debian bookworm/main amd64 libsubid0 amd64 1:4.13+dfsg1-1+b1 [211 kB]
Réception de : 4 http://deb.debian.org/debian bookworm/main amd64 uidmap amd64 1:4.13+dfsg1-1+b1 [189 kB]
Réception de : 5 http://deb.debian.org/debian bookworm/main amd64 libipset2 amd64 1.8.9-2 [19,4 kB]
Réception de : 6 http://deb.debian.org/debian bookworm/main amd64 iptables amd64 1.8.9-2 [386 kB]
Réception de : 7 http://deb.debian.org/debian bookworm/main amd64 containernetworking-plugins amd64 1.1.1+ds1-3+b5 [6 769 kB]
Réception de : 8 http://deb.debian.org/debian bookworm/main amd64 golang-github-containers-image all 5.23.1-4 [11,7 kB]
Réception de : 9 http://deb.debian.org/debian bookworm/main amd64 golang-github-containers-common all 0.50.1+ds1-4 [36,2 kB]
Réception de : 10 http://deb.debian.org/debian bookworm/main amd64 buildah amd64 1.28.2+ds1-3+deb12u1 [6 104 kB]
Réception de : 11 http://deb.debian.org/debian bookworm/main amd64 catatonit amd64 0.11.7-1+b1 [264 kB]
Réception de : 12 http://deb.debian.org/debian bookworm/main amd64 common amd64 2.1.6+ds1-1 [37,9 kB]
Réception de : 13 http://deb.debian.org/debian bookworm/main amd64 crun amd64 1.8.1-1+deb12u1 [289 kB]
Réception de : 14 http://deb.debian.org/debian bookworm/main amd64 curl amd64 7.88.1-3+deb12u1 [115 kB]
Réception de : 15 http://deb.debian.org/debian bookworm/main amd64 fuse-overlayfs amd64 1.10-1 [43,5 kB]
Réception de : 16 http://deb.debian.org/debian bookworm/main amd64 gnupg2 all 2.2.40-1 [1 446 kB]
Réception de : 17 http://deb.debian.org/debian bookworm/main amd64 libslirp amd64 4.7.0-1 [63,0 kB]
Réception de : 18 http://deb.debian.org/debian bookworm/main amd64 podman amd64 4.3.1+ds1-3+deb12u1 [10,7 MB]
Réception de : 19 http://deb.debian.org/debian bookworm/main amd64 slirpnetns amd64 1.2.0-1 [37,5 kB]
27,7 Mo réceptionnés en 3s (10,0 Mo/s)
Sélection du paquet netavark précédemment désélectionné.
(Lecture de la base de données... 154910 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../00-netavark_1.4.0-3_amd64.deb ...
Dépaquetage de netavark (1.4.0-3) ...
Sélection du paquet aardvark-dns précédemment désélectionné.
Préparation du dépaquetage de .../01-aardvark-dns_1.4.0-3_amd64.deb ...
Dépaquetage de aardvark-dns (1.4.0-3) ...
Sélection du paquet libsubid0 amd64 précédemment désélectionné.
Préparation du dépaquetage de .../02-libsubid0_1:4.13+dfsg1-1+b1_amd64.deb ...
Dépaquetage de libsubid0:amd64 (1:4.13+dfsg1-1+b1) ...
Sélection du paquet uidmap précédemment désélectionné.
Préparation du dépaquetage de .../03-uidmap_1:4.13+dfsg1-1+b1_amd64.deb ...
Dépaquetage de uidmap (1:4.13+dfsg1-1+b1) ...
Sélection du paquet libipset2:amd64 précédemment désélectionné.
Préparation du dépaquetage de .../04-libipset2_1.8.9-2_amd64.deb ...
Dépaquetage de libipset2:amd64 (1.8.9-2) ...
Sélection du paquet iptables précédemment désélectionné.
Préparation du dépaquetage de .../05-iptables_1.8.9-2_amd64.deb ...
Dépaquetage de iptables (1.8.9-2) ...
Sélection du paquet containernetworking-plugins précédemment désélectionné.
Préparation du dépaquetage de .../06-containernetworking-plugins_1.1.1+ds1-3+b5_amd64.deb ...
Dépaquetage de containernetworking-plugins (1.1.1+ds1-3+b5) ...
Sélection du paquet golang-github-containers-image précédemment désélectionné.
Préparation du dépaquetage de .../07-golang-github-containers-image_5.23.1-4_all.deb ...
Dépaquetage de golang-github-containers-image (5.23.1-4) ...
Sélection du paquet golang-github-containers-common précédemment désélectionné.
Préparation du dépaquetage de .../08-golang-github-containers-common_0.50.1+ds1-4_all.deb ...
Dépaquetage de golang-github-containers-common (0.50.1+ds1-4) ...
```

2. INSTALLATION DE CEPHADM (SUR NAS1 UNIQUEMENT)

```
curl --silent --remote-name
```

```
https://download.ceph.com/keys/release.asc
```

```
root@nas1:~# curl --silent --remote-name https://download.ceph.com/keys/release.asc
```

```
gpg --no-default-keyring --keyring ./ceph-release.gpg --import  
release.asc
```

```
root@nas1:~# gpg --no-default-keyring --keyring ./ceph-release.gpg --import release.asc
gpg: le trousseau local « ./ceph-release.gpg » a été créé
gpg: répertoire « /root/.gnupg » créé
gpg: /root/.gnupg/trustdb.gpg : base de confiance créée
gpg: clef E84AC2C0460F3994 : clef publique « Ceph.com (release key) <security@ceph.com> » importée
gpg: Quantité totale traitée : 1
gpg: importées : 1
```

```
gpg --no-default-keyring --keyring ./ceph-release.gpg --export >  
/etc/apt/trusted.gpg.d/ceph-release.gpg
```

```
root@nas1:~# gpg --no-default-keyring --keyring ./ceph-release.gpg --export > /etc/apt/trusted.gpg.d/ceph-release.gpg
```

```
echo deb https://download.ceph.com/debian-reef/ $(lsb_release  
-sc) main > /etc/apt/sources.list.d/ceph.list
```

```
root@nas1:~# echo deb https://download.ceph.com/debian-reef/ $(lsb_release -sc) main > /etc/apt/sources.list.d/ceph.list
```

```
apt update && apt install -y cephadm
```

```
root@nas1:~# apt update && apt install -y cephadm
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Réception de :4 https://download.ceph.com/debian-reef bookworm InRelease [8 547 B]
Réception de :5 https://download.ceph.com/debian-reef bookworm/main amd64 Packages [17,6 kB]
26,2 ko réceptionnés en 1s (22,0 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
cephadm
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 330 ko dans les archives.
Après cette opération, 370 ko d'espace disque supplémentaires seront utilisés.
```

3. BOOTSTRAP DU CLUSTER SUR NAS1

```
cephadm bootstrap --mon-ip 192.168.30.13 --initial-dashboard-  
user admin --initial-dashboard-password admin
```

```

root@nas1:~# cephadm bootstrap --mon-ip 192.168.30.13 --initial-dashboard-user admin --initial-dashboard-password admin
Creating directory /etc/ceph for ceph.conf
Verifying podman[docker] is present...
Verifying lvm2 is present...
Verifying time synchronization is in place...
Unit systemd-timesyncd.service is enabled and running
Repeating the final host check...
podman (/usr/bin/podman) version 4.3.1 is present
systemctl is present
lvcreate is present
Unit systemd-timesyncd.service is enabled and running
Host looks OK
Cluster fsid: c79d6alc-0e0f-11f0-9d88-a85e45c6070d
Verifying IP 192.168.30.13 port 3300 ...
Verifying IP 192.168.30.13 port 6789 ...
Mon IP '192.168.30.13' is in CIDR network '192.168.30.0/24'
Mon IP '192.168.30.13' is in CIDR network '192.168.30.0/24'
Internal network (--cluster-network) has not been provided, OSD replication will default to the public_network
Pulling container image quay.io/ceph/ceph:v18...
Ceph version: ceph version 18.2.4 (e7ad5345525c7aa95470c26863873b581076945d) reef (stable)
Extracting ceph user uid/gid from container image...
Creating initial keys...
Creating initial monmap...
Creating mon...
Waiting for mon to start...
Waiting for mon...
mon is available
Assimilating anything we can from ceph.conf...
Generating new minimal ceph.conf...
Restarting the monitor...
Setting public_network to 192.168.30.0/24 in mon config section
Wrote config to /etc/ceph/ceph.conf
Wrote keyring to /etc/ceph/ceph.client.admin.keyring
Creating mgr...
Verifying port 0.0.0.0:9283 ...
Verifying port 0.0.0.0:8765 ...
Verifying port 0.0.0.0:8443 ...
Waiting for mgr to start...
Waiting for mgr...
mgr not available, waiting (1/15)...
mgr not available, waiting (2/15)...
mgr not available, waiting (3/15)...
mgr not available, waiting (4/15)...
mgr not available, waiting (5/15)...
mgr is available
Enabling cephadm module...
Waiting for the mgr to restart...
Waiting for mgr epoch 5...
mgr epoch 5 is available
Setting orchestrator backend to cephadm...
Generating ssh key...
Wrote public SSH key to /etc/ceph/ceph.pub
Adding key to root@localhost authorized_keys...
Adding host nas1...
Deploying mon service with default placement...
Deploying mgr service with default placement...
Deploying crash service with default placement...
Deploying ceph-exporter service with default placement...
Deploying prometheus service with default placement...
Deploying grafana service with default placement...
Deploying node-exporter service with default placement...
Deploying alertmanager service with default placement...
Enabling the dashboard module...
Waiting for the mgr to restart...
Waiting for mgr epoch 9...
mgr epoch 9 is available
Generating a dashboard self-signed certificate...
Creating initial admin user...
Fetching dashboard port number...
Ceph Dashboard is now available at:

    URL: https://nas1:8443/
    User: admin
    Password: admin

Enabling client.admin keyring and conf on hosts with "admin" label
Saving cluster configuration to /var/lib/ceph/c79d6alc-0e0f-11f0-9d88-a85e45c6070d/config directory
Enabling autotune for osd_memory_target
You can access the Ceph CLI as following in case of multi-cluster or non-default config:

```

4. AJOUT DES AUTRES NŒUDS AU CLUSTER

Copier la clé SSH générée par cephadm sur chaque hôte :


```
ssh-copy-id -f -i /etc/ceph/ceph.pub root@nas2
ssh-copy-id -f -i /etc/ceph/ceph.pub root@MON3Nas
```

```
ceph orch host add nas2 192.168.30.14
ceph orch host add MON3Nas 192.168.30.21
```

5. DEPLOIEMENT DES SERVICES CEPH

Déployer les moniteurs :
ceph orch apply mon 'nas1,nas2,MON3Nas'

Déployer les OSD :

1. Identifier les disques à utiliser comme OSD :

```
lsblk
```

2. Ajouter l'OSD sur nas1 (par exemple sur /dev/sdb) :

```
ceph orch daemon add osd nas1:/dev/sdb
```

3. Ajouter l'OSD sur nas2 (par exemple sur /dev/sdb) :

```
ceph orch daemon add osd nas2:/dev/sdb
```

Une fois ces commandes exécutées, Ceph configure automatiquement les disques, crée les partitions nécessaires et déploie les services OSD correspondants. Il est possible de vérifier leur bon fonctionnement avec la commande :

```
ceph osd status
```

6. DEPLOIEMENT DU MDS (METADATA SERVER)

Le service MDS (Metadata Server) est requis pour le système de fichiers CephFS. Il gère les métadonnées (noms, hiérarchie, permissions) tandis que les OSDs gèrent les blocs de données.

- Voici les étapes de déploiement :

1. Créer un volume CephFS :

```
ceph fs volume create cephfs
```

2. Vérifier la création :

```
ceph fs ls
```

3. Déployer un MDS sur le cluster :

```
ceph orch apply mds cephfs --placement='1 host'
```

- Pour assurer une haute disponibilité, ajouter un MDS en standby :

```
ceph orch apply mds cephfs --placement='2 hosts'  
ceph orch apply mds cephfs --placement='nas1,nas2'
```

5. Vérifier l'état des MDS :

```
ceph mds stat
```

7. ACCES A L'INTERFACE WEB DU CLUSTER

L'interface Web d'administration du cluster est accessible via : <https://192.168.30.13:8443> (ou <https://192.168.30.14:8443> si nas2 devient le manager du cluster)

8. VÉRIFICATION DU CLUSTER

```
ceph -s  
ceph osd status  
ceph health detail
```

9. CONCLUSION

Ce projet permet de mettre en œuvre un cluster de stockage tolérant aux pannes en utilisant les bonnes pratiques modernes de déploiement via cephadm. MON3Nas, bien qu'il ne dispose pas de stockage, permet de maintenir un quorum même si un des nœuds OSD tombe en panne.

10. ACCES CEPHFS DEPUIS UN CLIENT WINDOWS VIA SMB

Ceph ne fournit pas de service SMB (partage Windows) nativement. Pour permettre à un client Windows d'accéder au CephFS, il est nécessaire de passer par un serveur Samba intermédiaire.

1. Installer Samba sur une machine Linux (ex: nas1 ou une VM intermédiaire)

apt update

apt install samba ceph-common -y

2. Monter le CephFS localement sur cette machine

Créer un point de montage :

mkdir -p /mnt/cephfs

Monter le FS avec l'utilisateur admin :

*mount -t ceph 192.168.30.13:6789:/ /mnt/cephfs -o
name=admin,secretfile=/etc/ceph/admin.secret*

Le fichier /etc/ceph/admin.secret doit contenir la clé de l'utilisateur admin :

echo 'AQB...cle_base64...' > /etc/ceph/admin.secret

chmod 600 /etc/ceph/admin.secret

3. Partager le dossier CephFS avec Samba

Modifier */etc/samba/smb.conf* et ajouter à la fin :

[cephfs]

path = /mnt/cephfs

browsable = yes

read only = no

guest ok = yes

force user = root

Redémarrer Samba :

systemctl restart smbd

4. Depuis Windows

Dans l'explorateur de fichiers :

\\192.168.30.13\cephfs

Ce chemin donne accès au contenu de CephFS via SMB.

11. CONCLUSION DU PROJET

Ce projet m'a permis de concevoir une solution de stockage redondante, sécurisée et hautement disponible, adaptée à un environnement professionnel. Après plusieurs essais sur des solutions prêtes à l'emploi comme **TrueNAS** puis **Synology**, j'ai fait le choix de basculer vers une approche **entièrement maîtrisée** et **open source** en utilisant **Ceph** et **Cephadm**.

J'ai ainsi pu bâtir une infrastructure robuste reposant sur :

- Un **cluster Ceph haute disponibilité** avec quorum
- Une **bascule automatique** via Keepalived (IP flottante)
- Une **réplication MariaDB** en mode actif/passif
- Une interface **Nextcloud** synchronisée entre les deux nœuds
- Une **supervision SNMP** avec intégration dans **Observium**

Enrichissement personnel et technique

Ce projet m'a permis de :

- Me confronter à un **environnement de production réaliste**
 - Appréhender des notions avancées comme le **quorum**, le **failover**, la **résilience**, ou encore la **consistance des données**
 - Automatiser des tâches critiques pour garantir une **continuité de service**
-

Perspectives d'amélioration

Dans un cadre professionnel, plusieurs axes pourraient encore renforcer cette solution :

- Déploiement de **Prometheus/Grafana** pour des métriques temps réel
 - Mise en place d'un **outil de failover avancé** comme **Orchestrator**
 - Ajout d'un **reverse proxy** (HAProxy/NGINX) pour une gestion fine des accès
 - Intégration avec un **annuaire LDAP** pour une gestion centralisée des utilisateurs
-

Application professionnelle

Ce type d'infrastructure est parfaitement adapté pour :

Une PME ou un établissement souhaitant **garantir la disponibilité de ses données**, tout en conservant **la main sur sa solution** et en évitant des coûts de licence élevés.