HEH.be
Sciences
et technologies

Avenue V. Maistriau 8a
B-7000 Mons
+32 (0)65 33 81 54
scitech-mons@heh.be

WWW.HEH.BE

Projet Linux 2024-2025

Rapport technique
tomananas.lan

Bachelier en Informatique
Orientation réseaux & télécommunications avec option sécurité

Tom Deneyer
Anastasiia Kozlenko

WALLONIE-BRUXELLES ENSEIGNEMENT
pôle hainuyer

Objectif du document

L'objectif de ce document est de présenter de manière structurée le travail réalisé par l'équipe tomananas.lan durant la semaine de Projet Linux à la HEH.

Contact

Kozlenko Anastasiia, Deneyer Tom
E-mail : anastasiia.kozlenko@std.heh.be, tom.deneyer@std.heh.be
Haute Ecole en Hainaut,
Département des Sciences et Technologies
8A, Avenue Maistriau,
7000 Mons

Confidentialité

Les informations contenues dans ce document sont réservées à un usage interne à la HEH.

Termes et conditions

Informations sur le document

Nom du document : tomananas_Rapport_Linux2025.pdf
Version : Version 1
Niveau de confidentialité : Utilisation interne à la HEH uniquement
Créateur du document : Anastasiia Kozlenko
Date de création : 17 mai 2025
Révisé par : L'équipe des professeurs du projet
Date de dernière révision : /
Approuvé par : /

Suivi des modifications

Nom Prénom	Date	Version	Mises à jour
Kozlenko Anastasiia	18-05-2025	1.0	1) Création du document, mise en page 2) Introduction, DNS, Samba, Chrony
Deneyer Tom	19-05-2025	1.1	1) Reformulation introduction 2) FTP, Web, Backend, CA, Monitoring, Base commune 3) Ameliorations possibles
Kozlenko Anastasiia	19-05-2025	1.1	1) Ajout des instructions des connexions Samba 2) NFS, scripts de déploiement
Kozlenko Anastasiia	21-05-2025	1.2	1) Gestion des utilisateurs

			2) Gestion des backups, plan de sauvegarde 3) Problèmes rencontrés, améliorations 4) Bibliographie
Deneyer Tom	21-05-2025	1.2	1) DRP 2) Problèmes rencontrés 3) Améliorations 4) Sécurisation serveur
Anastasiia Kozlenko	22-05-2025	1.3	1) Relecture du document 2) Mise en page des images 3) Correction des annexes
Deneyer Tom	22-05-2025	1.3	1) Relecture du document 2) Ajout d'améliorations possibles et conclusion 3) Export et imports d'annexes

Table des matières

1. Introduction

Ce document présente le travail réalisé par l'équipe de Tom et Anastasiia dans le cadre du projet Linux. Il décrit la mise en place d'une infrastructure complète d'hébergement destinée à accueillir des sites web pour des clients, avec la gestion des services tels que les bases de données, les serveurs web, le partage de fichiers, et les outils de supervision.

2. Présentation générale du projet

Le projet a pour but de mettre en place un environnement d'hébergement web complet. Les services ont été installés sur plusieurs serveurs Linux virtuels dans une architecture cloud AWS. Chaque serveur assure un ou plusieurs rôles précis : DNS, FTP, base de données, serveur web, etc. L'objectif est d'offrir un hébergement fonctionnel à des clients : site web, base de données, tout ça avec un accès sécurisé et personnel à leurs données.

Pour assurer une organisation efficace, nous avons utilisé la plateforme Notion pour gérer les tâches, répartir le travail au sein de l'équipe et documenter l'ensemble des procédures réalisées tout au long de la semaine ainsi que nos notes pour la configuration des services et informations principales.

3. Gestion d'infrastructure dans le cloud

Les professeurs ont mis à notre disposition une infrastructure de base dans AWS sur un VPC (Virtual-Private-Cloud) déployé le premier jour de la semaine de projet, comprenant les spécifications suivantes :

- Une instance VPN avec une IP publique fixe (Elastic IP) pour pouvoir connecter notre PC directement dans le réseau privé via OpenVPN et une configuration toute faite fournie.
- Deux instances EC2 sous Amazon Linux vierges donc chacune possédait deux volumes de 2 Go non configuré (pas de LVM, partitions, rien du tout)
- Un modèle d'instance préconfiguré avec les groupes de sécurité nécessaires
- Les groupes de sécurités (sg), subnets, routing, etc ont été configurés lors du déploiement automatisé des professeurs.

Pour organiser notre infrastructure de façon claire et sécurisée, nous avons décidé de séparer les services sur plusieurs machines virtuelles, cette séparation permet principalement la division des responsabilités en cas de pannes ou attaques, par exemple le backend est séparé du web.

Instance	Services	IP privée	IP publique
VPN	Accès VPN (IP fixe)	10.42.0.100	34.203.121.179
web-ftp-01	Serveur Web, FTP, Samba/NFS	10.42.0.87	—
dns-ntp-01	DNS, Serveur de temps (NTP)	10.42.0.37	—
admin-backup-01	Sauvegarde, Déploiement	10.42.0.113	—
mysql-01	MySQL, Interface PHPMyAdmin	10.42.0.170	—

4. Plan de partionnement

Toutes nos instances profitent des trois disques physiques installés, le disque principal par défaut pour le système qui est instancié avec l'instance EC2, ainsi que les deux disques supplémentaires ajoutés en configuration. Ces deux disques nous permettent de séparer les données utilisateurs ou bien ajouter un espace supplémentaire. Voici ci-dessous un tableau récapitulatif de nos instances et utilisations des volumes configurés en LVM via des groupes de volumes.

Les disques sont décrits comme :

*VG_NAME : **MOUNT_POINT***
UTILISATION

Les créations lv/vg, montages et configurations sont disponibles dans les archives dans la première partie des scripts de setup_*x*.sh.

Instance	Services	Disque supp.1 (2go)	Disque supp.2 (2go)
VPN	Accès VPN (IP fixe)	/	/
web-ftp-01	Serveur Web, FTP, Samba/NFS	srv_vg : **/srv/www/** Données des sites clients	share_vg : **/srv/share/** Dossier de partage de fichier publique
dns-ntp-01	DNS, Serveur de temps (NTP)	dns_vg : **/srv/dns-zones/** Fichiers de zones forward+reverse dns	
admin-backup-01	Sauvegarde, Déploiement	backup_vg : **/home/backup/backups** Enregistrements des archives et backups services et utilisateurs	
mysql-01	MySQL, Interface PHPMyAdmin	mysql_vg : **/var/lib/mysql** Données backends clients	

5. Services et leur configuration

5.1. Configuration de base commune

Pour chaque serveur déployé, une procédure est appliquée. En premier de fichier d'environnement contenant toutes les variables nécessaires est importé, ainsi que le script de déploiement de base commun à tous les serveurs et enfin le script de déploiement du service spécifique.

Par exemple si besoin de configurer pour remplacer une instance cassée de DNS, voici les fichiers :

- setup_env.sh
- setup_server_base.sh
- setup_dns-ftp.sh

Le script commun, setup_server_base.sh**,** permet la configuration basique et minimal de nos serveurs via des demandes en console lors du fonctionnement du script avec comme manipulations :

- Vérification de la présence du fichier d'environnement.
- Définition du hostname de la machine (pas de l'instance).
- Définition nouveau mot de passe root.
- Suppression du /etc/resolve.conf pour le DNS automatique problématique et attribution du DNS statique avec notre serveur interne.
- Configuration client du NTP (chrony) avec notre serveur NTP interne.
- Configuration du fuseau horaire Europe/Brussels.
- Installation et configuration fail2ban.
- Configuration pare-feu (Firewalld) avec SSH active seulement.
- Installation et configuration Antivirus (ClamAV + Linux Malware Detect LMD) avec vérification des répertoires sélectionnés pour un scan deux fois par jour automatiques crond
- Installation et ajout du node au serveur Netdata de monitoring en cloud.

5.2. Serveur DNS

Notre serveur DNS est hébergé sur la machine `dns-ntp-01`. Il utilise BIND avec une configuration de type maître pour les zones directe et inverse du domaine local tomananas.lan .

- Fichier de configuration principal : `/etc/named.conf`
- Répertoires de zones : `/srv/dns-zones/`
  - `forward.tomananas.lan` : zone directe, permet la résolution de noms d'hôtes vers IP (enregistrements A)
  - `reverse.tomananas.lan` : zone inverse, la résolution inverse (IP vers nom d'hôte, PTR)

Nous avons activé **DNSSEC** avec signature automatique (`inline-signing yes`) pour renforcer la sécurité des résolutions DNS. Le principe de DNSSEC est de signer cryptographiquement les enregistrements DNS. Cela empêche des attaques de type "DNS spoofing".

Dans la zone directe, nous avons déclaré un enregistrement A pour chaque machine du réseau ce qui permet :

- L'utilisation de noms d'hôtes dans les scripts, par exemple ssh web-ftp-01 au lieu de retenir une IP
- Chaque utilisateur dispose d'un enregistrement A pointant vers le serveur web.

Ces enregistrements sont ajoutés ou supprimés automatiquement par les scripts de gestion d'utilisateurs. À chaque modification de zone, le serial est incrémenté et le service named est redémarré pour appliquer les changements.

Exemple de résolution d'adresse :

```
nslookup MrRoland.tomananas.lan

Server:         10.42.0.37
Address:        10.42.0.37#53

Name:   MrRoland.tomananas.lan
Address: 10.42.0.87
```

5.3. Serveur FTP

Le service FTP a été mis en place à l'aide de vsftpd (Very Secure FTP Daemon). Le protocole FTPS explicite (FTP over SSL/TLS) a été activé pour chiffrer les connexions et garantir la confidentialité des données transmises entre les clients et notre serveur. La gestion (création et suppression) des accès utilisateur au ftp est réalisé via les scripts de gestion utilisateur (`utils_add_user.sh` + `utils_delete_user.sh`).

Grace à cette connexion l'utilisateur peut ainsi accéder à son dossier web et y déposer son `index.html` ou `index.php`, etc.

Notre configuration permet notamment :

- D'interdire les connexions anonymes.
- D'autoriser uniquement les utilisateurs locaux du système.
- D'isoler chaque utilisateur dans son propre répertoire.
- De forcer l'utilisation de TLS pour toutes les communications.

Chaque utilisateur lors de sa connexion ftp est isolé (chrooté) dans son dossier personnel dans `/srv/www/<USER>.`

Ce répertoire est utilisé comme racine FTP grâce à la directive `locale_root`, ce qui empêche tout accès aux autres fichiers du système ou autres clients.

L'option `chroot_local_user=YES` permet de confiner les utilisateurs dans leur répertoire personnel. Pour qu'ils puissent quand même y écrire, l'option `allow_writeable_chroot=YES` est activée.

Un certificat SSL auto-signé est utilisé pour activer la couche de chiffrement TLS. La configuration impose l'usage de TLS pour les connexions de commande et de transfert de données. Le serveur est configuré en mode passif, ce qui facilite les transferts derrière des pare-feu et une plage de ports est définie pour les connexions passives (40000-40100). Ces ports sont donc autorisés au niveau du pare-feu.

Voici la configuration `/etc/vsftpd/vsftpd.conf`

```
listen=YES
listen_ipv6=NO
anonymous_enable=NO
local_enable=YES
write_enable=YES
chroot_local_user=YES
allow_writeable_chroot=YES
user_sub_token=$USER
local_root=/srv/www/$USER
xferlog_enable=YES
xferlog_file=/var/log/vsftpd.log
log_ftp_protocol=YES
ssl_enable=YES
rsa_cert_file=/etc/pki/tls/certs/vsftpd.pem
rsa_private_key_file=/etc/pki/tls/private/vsftpd.key
force_local_logins_ssl=YES
force_local_data_ssl=YES
pasv_enable=YES
pasv_min_port=40000
pasv_max_port=40100
pasv_address=10.42.0.87
```

```
pam_service_name=vsftpd
userlist_enable=NO
```

Le service FTP est ainsi sécurisé par chiffrement TLS, restreint aux utilisateurs locaux, cloisonné par utilisateur, et configuré pour être conforme aux bonnes pratiques de sécurité réseau (ports restreints, chroot, chiffrement, journalisation). Ce service permet un transfert de fichiers sécurisé et contrôlé au sein de l'infrastructure. Toutes ces configurations sont déployées automatiquement dans le script `setup_web-ftp.sh` !

5.4. Serveur de Temps

Un serveur de temps permet de synchroniser l'horloge système avec une source de temps précise (un serveur NTP – Network Time Protocol). Cela garantit que l'heure est correcte, ce qui est essentiel pour :

- les logs système et réseau (pour diagnostiquer des erreurs ou corréler des événements)
- les certificats SSL/TLS (dates d'expiration, validation)
- les bases de données (timestamps cohérents)

Le service de synchronisation horaire a été installé sur la même machine que le serveur DNS (`dns-ntp-01`), à l'aide de Chrony. Nous avons comparé Chrony avec d'autres outils de synchronisation sur Linux, et nous l'avons choisi pour sa précision et son pouvoir de faire client/serveur :

Critère	Chrony	NTP (ntpd)	systemd-timesyncd
Type	Client/Serveur	Client/Serveur	Client uniquement
Précision	µs	ms	secondes
Environnement recommandé	Serveurs, machines virtuelles	Réseaux anciens/stables	Desktops, systèmes légers
Temps de convergence	Très rapide	Lent	Moyen
Serveur NTP intégré	Oui	Oui	Non
Paquet	chrony	ntp	Inclus avec systemd

Le fichier de configuration de Chrony `/etc/chrony.conf` a été modifié comme suit :

- **Sources de temps externes** :
  Un serveur NTP interne AWS (169.254.169.123) accessible sans sortir du VPC.
  Deux serveurs publics du pool NTP européen pour la redondance (`0.europe.pool.ntp.org` et `1.europe.pool.ntp.org`).
- Uniquement le sous-réseau local 10.42.0.0/24 est autorisé à se synchroniser avec notre serveur.
- Si aucune source externe n'est disponible, le serveur passe en stratum 8 pour fournir une horloge approximative aux clients internes.

- **Autres paramètres** :
  `makestep 1.0 3` : permet une correction immédiate de l'horloge si l'écart dépasse 1 seconde lors des trois premiers ajustements.
  `rtcsync` : synchronise l'horloge matérielle avec l'horloge système.

Tous les autres serveurs de notre infrastructure sont configurés en tant que client pour se synchroniser avec notre serveur Chrony.
La commande `chronyc sources` nous permet de voir avec quel serveur la machine est synchronisée.
Nous pouvons voir que AWS est la source avec le moins de latence, donc elle est choisie :

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 169.254.169.123               3   8   377   235   +174us[ +175us] +/-
704us
^- leontp2.office.panq.nl        1  10   377   708  +1531us[+1485us] +/-
43ms
^- 86-52-51-137.norlyscusto>     2  10   377   699   -334us[ -381us] +/-
108ms
```

Client NTP `web-ftp-01` :

```
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 10.42.0.37                    4   8   377    38   -111us[ -111us] +/-
581us
```

5.1. Serveur de Fichiers (Samba)

Le serveur `web-ftp-01` héberge également un service Samba, permettant aux utilisateurs de tomananas.lan d'accéder à deux types de partages depuis un poste Windows ou Linux : privé et public.
Samba permet de partager des dossiers Linux sur le réseau, en les rendant accessibles comme des dossiers partagés Windows (protocole SMB/CIFS).

Partages configurés :

- ✓ **Dossier personnel**
  - Chemin : `/srv/www/%U`
  - Seul l'utilisateur peut y accéder, lecture/écriture uniquement pour le propriétaire
  - Non visible dans l'explorateur Windows (`browseable = no`)
- ✓ **Partage public**
  - Chemin : `/srv/share`
  - Accessible par tous les utilisateurs (même en guest), lecture et écriture autorisées
  - Visibilité activée dans le réseau Windows (`browseable = yes`)

Accès depuis Windows :

- Exécuter \\10.42.0.87\public ou \\10.42.0.87\www
- Identifiants :
  guest - aucun mot de passe (possible pour \public)

utilisateur – login et mot de passe saisi lors de la création de compte (possible pour \public et \www)

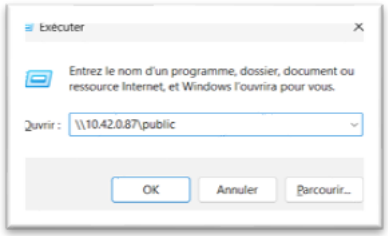


Figure 1. Exécution de chemin Win+R

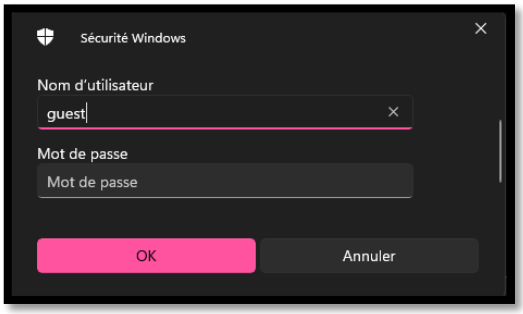


Figure 2. Saisie de mot de passe

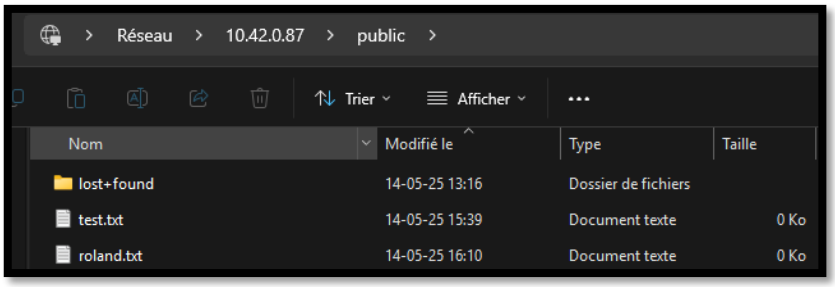


Figure 3. Explorateur ouvre le dossier

Accès depuis Linux via smbclient:

```
smbclient //10.42.0.87/public -U guest
Password for [SAMBA\guest]:
smb: \> ls
  test.txt                            A        0  Wed May 14 13:39:23 2025
  roland.txt                          A        0  Wed May 14 14:10:08 2025
```

Comptes utilisateurs

Les utilisateurs sont créés via nos scripts, qui :

- Génèrent un dossier `/srv/www/USERNAME`
- Ajoutent un compte Samba correspondant
- Appliquent les bons droits

5.2. Partage de Fichiers en Réseau (NFS)

Dans notre infrastructure, nous avons mis en place un service NFS (Network File System) aussi sur le serveur `web-ftp-01`, afin de permettre aux autres machines également sous Linux d'accéder au dossier partagé. Le dossier `/srv/share` est exporté à l'ensemble du sous-réseau privé `10.42.0.0/24`, avec des droits de lecture et d'écriture.

Sur les machines clientes, le dossier `/srv/share` est accessible automatiquement grâce à Autofs, un système de montage à la demande. Autofs permet :

- un montage automatique au moment de l'accès
- un démontage automatique après un délai d'inactivité (60 secondes dans notre cas)
- réduire utilisation des ressources de la machine si le dossier n'est pas monté

Nous avons également créé un script de configuration Autofs client, qui configure automatiquement le point de montage (/mnt/nfs/public) et ajoute les bonnes règles dans les fichiers du système. Une fois le script de configuration exécuté sur la machine cliente, le dossier partagé devient accessible à l'emplacement suivant : `/mnt/nfs/public`

Le système de fichiers est monté automatiquement au premier accès, par exemple avec une simple commande :

```
cd /mnt/nfs/share
ls
roland.txt  test.txt
```

Ce montage est ensuite démonté automatiquement après un certain temps d'inactivité (60 secondes dans notre configuration).

Ce fonctionnement est totalement transparent pour l'utilisateur : aucune action manuelle n'est nécessaire après le déploiement du script Autofs. Les utilisateurs et les scripts système peuvent utiliser le dossier comme s'il faisait partie du système de fichiers local.

5.3. Serveur Web (HTTP/HTTPS)

Le service web repose sur Apache HTTP Server (httpd), configuré pour servir des sites en HTTP et HTTPS. Chaque utilisateur/client dispose de son propre espace web personnel situé dans le dossier `/srv/www/%USER%`, qui est également utilisé comme racine FTP pour garantir une cohérence d'accès entre les services.

Notre configuration permet notamment :

- D'héberger des sites web personnels pour chaque utilisateur.
- De sécuriser l'accès HTTPS à l'aide d'un certificat SSL signé par notre propre CA.
- D'activer les virtual hosts pour permettre une configuration multi-sites.
- De désactiver le vhost SSL par défaut pour éviter les conflits sur le port 443.
- D'isoler la configuration des sites dans un répertoire dédié /etc/httpd/sites-available.

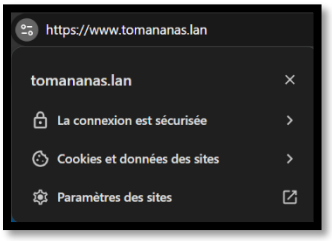


Figure 4. Visualisation de connexion https dans le navigateur

L'arborescence des sites web est centralisée dans le dossier `/srv/www/`, monté sur un volume logique dédié (LVM), permettant une meilleure gestion de l'espace disque.
Deux *virtual hosts* de fallback (ou « catch-all ») sont mis en place pour éviter les erreurs de sites accessibles avec des noms de domaine en cache DNS client même après la suppression de l'utilisateur etc… :

- HTTP sur le port 80.
- HTTPS sur le port 443.

Ils pointent vers un site par défaut situé dans `/srv/www/default`, qui affiche une page d'accueil informative lorsque la requête ne correspond à aucun site spécifique. Les fichiers de configuration de ces hôtes virtuels par défaut sont placés dans :

- `/etc/httpd/sites-available/000-default.conf` (HTTP)
- `/etc/httpd/sites-available/000-default-ssl.conf` (HTTPS)

Exemple de configuration : `/etc/httpd/sites-available/000-default-ssl.conf`

```
<VirtualHost *:443>
    ServerName fallback.
    DocumentRoot /srv/www/default
    SSLEngine on
    SSLCertificateFile    /etc/pki/tls/certs/wildcard.crt.pem
    SSLCertificateKeyFile /etc/pki/tls/private/wildcard.key.pem
    <Directory "/srv/www/default">
        Options -Indexes
        AllowOverride None
        Require all granted
    </Directory>
    ErrorDocument 404 /index.html
</VirtualHost>
```

Voici le fallback :

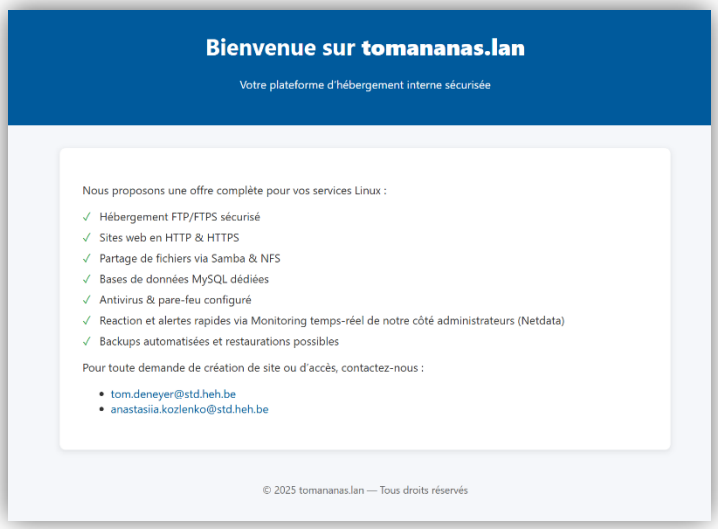


Figure 5. Page fallback ou www.tomananas.lan

Un certificat SSL est utilisé pour chiffrer les communications HTTPS. Le module mod_ssl est activé, et l'écoute sur le port 443 est explicitement configurée dans le fichier principal `/etc/httpd/conf/httpd.conf`. Le certificat est signé avec notre serveur de certificat interne, celui-ci nous permet, une fois ajouté aux clients windows par exemple, d'avoir un site https sans message rouge barré en comparaison avec l'url d'un site autosigné.

L'accès aux répertoires web est restreint :

- Chaque utilisateur n'a accès qu'à son propre site.
- L'utilisateur backup dispose d'un accès en lecture sur tous les sites pour les sauvegardes, via des ACL (setfacl).

Le service Apache est activé au démarrage (systemctl enable --now httpd) et redémarré à chaque modification de configuration. Le service web ainsi configuré permet d'héberger plusieurs sites, avec une gestion simple, modulaire et sécurisée. Toutes ces configurations sont automatiquement déployées par le script `setup_web-ftp.sh`.

5.4. Serveur de Base de Données (MySQL + PhpMyAdmin)

Le serveur de base de données utilisé dans cette infrastructure est MySQL Community Server 8.0, installé depuis le dépôt officiel de MySQL. L'objectif est de fournir un service SQL accessible à distance, tout en garantissant la sécurité et l'isolement des données. Les fichiers de données MySQL sont stockés dans `/var/lib/mysql` monté sur un groupe lvm séparé pouvant facilement être agrandit.

Une fois MySQL installé et démarré, les actions suivantes ont été réalisées pour sécuriser l'installation et préparer l'environnement :

- Changement du mot de passe root à une valeur définie et non disponible à distance après récupération du mot de passe temporaire généré automatiquement.
- Suppression des comptes anonymes.
- Suppression de la base de test par défaut.
- Nettoyage des privilèges.
- L'authentification par mot de passe est utilisée, et le module de vérification de complexité (component_validate_password) a été désactivé pour simplifier les mots de passe dans le cadre de ce projet.
- Création d'un compte administrateur principal

L'administrateur distant a été créé avec les privilèges complets sur toutes les bases :

- Nom d'utilisateur : admin
- Mot de passe : AdminStrongPwd!2025
- Hôte autorisé : % (toutes adresses IP)

Le fichier de configuration /etc/my.cnf.d/server.cnf a été modifié pour autoriser les connexions réseau en écoutant sur toutes les interfaces (bind-address = 0.0.0.0) et le port 3306 (port MySQL par défaut) a été autorisé dans le pare-feu via firewalld.

Pour faciliter la gestion graphique des bases de données, phpMyAdmin a été installé dans /var/www/html/phpmyadmin.

Les étapes de configuration incluent :

- Téléchargement de la version 5.2.1 depuis le site officiel.
- Génération d'un blowfish secret pour la sécurité des sessions.
- Attribution des droits à l'utilisateur Apache.

Enfin, un fichier `/root/.my.cnf` a été généré automatiquement pour permettre l'accès direct à MySQL depuis le terminal sans ressaisie de mot de passe, mais accessible uniquement par root (600) utilisé pour les backups.
Toute cette configuration est entièrement déployée via le script `setup_backend.sh`.

5.5. Serveur de certificat

Afin de sécuriser les connexions HTTPS au sein de l'infrastructure, une autorité de certification interne (CA**)** a été mise en place. Cette CA permet de générer et de signer des certificats SSL/TLS de manière autonome, évitant ainsi le recours à des tiers externes.
La CA est hébergée localement sur la machine de gestion et repose sur l'utilisation de OpenSSL. Elle est composée des éléments suivants :

- Une clé privée de la CA : `ca.key.pem`
- Un certificat auto-signé de la CA : `ca.crt.pem`
- Un répertoire structuré pour la gestion des certificats (`/root/ca`) avec les sous-dossiers : `private/`, `certs/`, `csr/`, etc.

Le certificat de la CA est valide pendant 10 ans (3650 jours) et doit être importé manuellement sur les machines clientes (Linux ou Windows) afin que les certificats émis soient reconnus comme fiables.
Ce certificat est :

- Signé par la CA locale,
- Valable pendant 825 jours,
- Accompagné de sa propre clé privée.

Le fichier de configuration utilisé pour la CSR (Certificate Signing Request) inclut les extensions SAN (Subject Alternative Name) nécessaires pour la reconnaissance des sous-domaines.

Une fois générés, les fichiers suivants sont transférés en toute sécurité sur le serveur web :

- wildcard.crt.pem (certificat signé)
- wildcard.key.pem (clé privée)
- ca.crt.pem (certificat de la CA)

Ces certificats sont ensuite installés dans les répertoires standards :

- /etc/pki/tls/certs/wildcard.crt.pem
- /etc/pki/tls/private/wildcard.key.pem
- /etc/pki/ca-trust/source/anchors/ca.crt.pem

Le certificat de la CA est intégré au système de confiance du serveur web via la commande `update-ca-trust`.
Le choix du placement du CA au niveau de l'instance `admin-backup-01` est stratégique. Cette instance possède les droits de se connecter en ssh aux autres instances et donc l'import du certificat et de la clef se fait

sans aucun soucis grâce à notre script setup_CA.sh qui réalise toutes ces configurations automatiquement (ainsi que l'imports des fichiers…)

5.6. Supervision et Monitoring (Netdata)

Pour assurer une surveillance continue de l'état des serveurs, le système de supervision Netdata a été mis en place sur chaque machine de l'infrastructure (chaque instance EC2 du VPC). Ce service permet de visualiser en temps réel l'activité système, les ressources utilisées, le trafic réseau, l'état des services, et bien plus encore. Netdata est un choix « *idéal* » dans le cadre de ce projet précis et avec nos use cases. Contrairement à des solutions plus lourdes comme Zabbix ou Prometheus, Netdata permet un déploiement en quelques secondes sur chaque machine, sans avoir à mettre en place un serveur central ni une base de données complexe. De plus le monitoring est disponible sans besoin d'un vpn car il est hébergé sur un cloud séparé. Les alertes sont automatiques par courriels sans besoin d'un service snmp etc.

L'objectif est d'avoir :

- Une vue centralisée et unifiée de tous les serveurs
- Des alertes automatiques en cas d'anomalies
- Un accès web sécurisé à l'interface de chaque nœud ou à un tableau de bord global via Netdata Cloud.

Nous avons créé un site web au sein de notre réseau (https://monitoring.tomananas.lan) qui centralise les liens vers les Dashboard de monitoring pour nos instances séparées. Mais tous les nœuds sont accessible depuis l'interface : https://app.netdata.cloud.

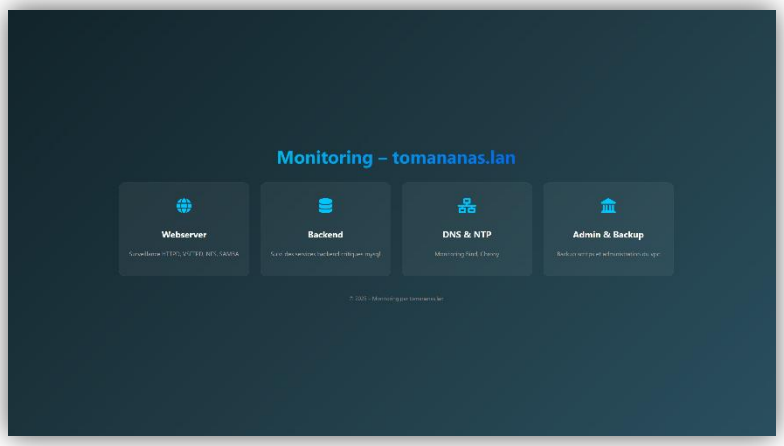


Figure 6. Page de monitoring.tomananas.lan

Netdata a été installé à l'aide du script officiel fourni par les développeurs :

```
curl https://get.netdata.cloud/kickstart.sh > /tmp/netdata-kickstart.sh &&
sh /tmp/netdata-kickstart.sh --stable-channel --claim-token
qLJ6N91iNhe9fyPhWzVe0AjWnM3895IwOJam2iApYmHgyiDW1qwvk3CT46VZKGPxdHZpppc4EZ6
u2ve0Br0zYwZ5sETWCxStRXXbiI1P-eVvsmDzVlAm9VPFYK6Je89BM96SYxY --claim-rooms
3db515f7-2058-48a3-b3f8-0c0ef773d79f --claim-url https://app.netdata.cloud
```

Quatre dashboards ont été créés pour couvrir le monitoring simple de nos ressources de bases des instances ainsi que des services principaux et critiques utilisés. (web serveur ci-dessous, le reste est en annexe)

Figure 7. Dashboard de serveur web

En plus de ces tableaux de bords personnalisés nous pouvons compter aussi sur les données récoltées par netdata, comme les logs des services de toutes les instances, les métriques supplémentaires, erreurs (selon criticités) avec des alertes de base bien pensées !
Ces alertes sont envoyées par email aux adresses des administrateurs ajoutés sur la plateforme cloud (anastasiia.kozlenko@std.heh.be et tom.deneyer@std.heh.be).
Ci-dessous quelques exemples d'alertes reçues :

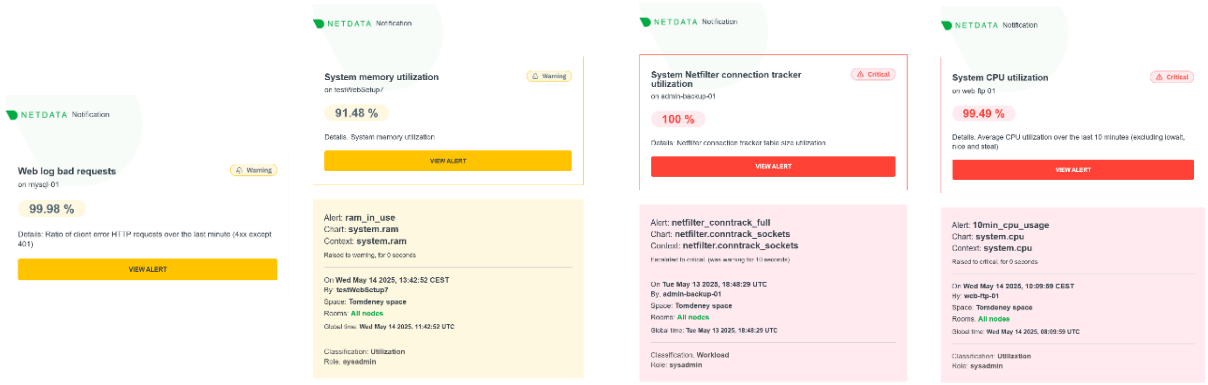


Figure 8. Exemples des alertes reçus par e-mail

6. Explication des scripts

6.1. Déploiement des serveurs

Tout le déploiement de nos machines a été automatisé à l'aide de scripts Bash.
Lorsqu'un nouveau serveur doit être déployé :

1. On exécute d'abord le script de base, commun à toutes les machines.
2. Ensuite, on exécute le script spécifique au rôle du serveur (web, DNS, base de données, etc.).

Chaque script suit une structure similaire:

- Charge un fichier de configuration (`setup_env.sh`) contenant les variables nécessaires
- Vérifie que ce fichier est bien présent et que toutes les variables requises sont définies
- Contrôle que l'exécution se fait avec le droit root
- Configure des services système (systemd, firewall, comptes, etc.) ;
- Utilise `set -euo pipefail` pour garantir que tout échec interrompt immédiatement le script (par ex. pour empêcher l'utilisation des variables vides ou des dossiers non créés)
- Affiche des messages clairs et structurés à l'utilisateur.

L'utilisation de fichiers de configuration centralisés permet d'adapter facilement le déploiement à un nouveau contexte (autres IP, autres domaines, volumes, etc.) sans modifier les scripts eux-mêmes.

6.2. Liste des scripts de déploiement et leurs étapes

Nom de Script	Etapes
setup_server_base.sh	• Configuration du hostname et du mot de passe root. • Configuration DNS (fichier `/etc/resolv.conf` verrouillé). • Synchronisation NTP via Chrony (client). • Sécurité de base : fail2ban, pare-feu firewalld (SSH autorisé), antivirus (ClamAV + Linux Malware Detect), scan automatique 2 fois par jour. • Création de l'utilisateur backup avec accès restreint.
setup_dns-ntp.sh	• Création d'un volume LVM pour stocker les fichiers de zones. • Configuration des zones DNS directe et inverse pour le domaine tomananas.lan, avec DNSSEC activé. Signature automatique des zones (inline-signing) et politique DNSSEC par défaut. • Définition des enregistrements A et PTR pour tous les serveurs. • Mise en place du service NTP interne, avec autorisation pour le sous-réseau local. • Configuration du pare-feu (ports DNS et NTP).
setup_web.sh	• Création de volumes LVM pour `/srv/www` (sites personnels) et `/srv/share` (dossier partagé). • Configuration des quotas sur `/srv/share`. • Installation de :

	- Apache + PHP - Serveur FTPS sécurisé (vsftpd + certificat auto-signé) - Samba : partage personnel par utilisateur (`/srv/www/%U`) et dossier public - NFS : export de `/srv/share` pour le réseau local. • Mise en place d'un site fallback HTTP/HTTPS avec une page 404 personnalisée. • Ouverture des ports dans le pare-feu (21, 40000–40100, 80, 443, Samba, NFS).
setup_backend.sh	• Création de volumes LVM pour `/var/lib/mysql`. • Installation du dépôt officiel MySQL et du serveur. • Initialisation, sécurisation et configuration du mot de passe root. • Création d'un utilisateur distant administrateur avec tous les droits. • Déploiement de phpMyAdmin en HTTP. • Configuration du firewall (3306, HTTP).
setup_backup.sh	• Volume LVM dédié pour `/home/backup/backups`. • Création de l'utilisateur backup, scripts et répertoires. • Planification d'une tâche cron pour lancer le script backup.sh tous les jours à 3h. • Gestion centralisée des sauvegardes des services Web, DNS, base de données, etc. • Application des permissions et sécurité sur les dossiers de sauvegarde.
setup_CA.sh	• Génération d'un certificat root auto-signé. • Génération d'un certificat wildcard pour *.tomananas.lan. • Signature par la CA interne. • Copie et installation des certificats sur le serveur Web. • Instructions pour importer le certificat sur les clients (Windows/Linux).

6.3. Gestion des utilisateurs

6.2.1 Ajouter un utilisateur

Nous avons développé un script complet pour automatiser l'ajout d'un nouveau client hébergé dans notre infrastructure. Ce script permet de configurer, en une seule commande, tous les éléments nécessaires pour qu'un utilisateur dispose d'un site web personnel, d'un accès FTP sécurisé, d'un compte Samba, d'une base de données et d'un enregistrement DNS.

La commande prend en argument le nom de nouvel utilisateur :

```
bash /home/ec2-user/utils_add_user.sh NewUser
```

Étapes réalisées par le script :

1. **Chargement des variables de setup_env.sh**
   Le script s'appuie sur un fichier `setup_env.sh` contenant les adresses IP internes des serveurs, les limites de quotas, et d'autres paramètres nécessaires.
2. **Saisie des mots de passe (+confirmation)**
   - Le mot de passe FTP/Samba (Linux)
   - Le mot de passe SQL (base de données)
3. **Connexion au serveur Web (web-ftp-01)**
   Toutes les connexions entre les serveurs de notre infrastructure se font via SSH avec l'option `-i`, en utilisant une clé privée associée à l'utilisateur ec2-user, qui dispose des droits sudo.
   Pour qu'un serveur accepte une connexion, la clé publique correspondante doit être présente dans le fichier `~/.ssh/authorized_keys` de l'utilisateur ec2-user. Nous avons choisi cette méthode car les connexions SSH par clé sont plus sûres que celles par mot de passe. Ça nous libère aussi de besoin de mettre le mot de passe dans une variable et le transmettre en clair. Exemple :

```
ssh -i "$SSH_KEY" ec2-user@"$WEB_PRIVATE_IP" bash -s <<EOF
echo "[+] Création de l'utilisateur Linux $USERNAME"
sudo useradd -m "$USERNAME"
```

   - Création de l'utilisateur Linux
   - Application des quotas disque sur les dossiers partagés
     Nous avons choisi 50Mb et 60 Mb pour le limite doux et limite dur, car chaque volume de stockage associé à ces dossiers est limité seulement à 2 Go. Limiter la place accordée à chaque utilisateur permet de prévenir les abus ou erreurs (upload massif, logs trop volumineux, etc.).

```
sudo setquota -u "$USERNAME" $SOFT_LIMIT $HARD_LIMIT 0 0 /srv/www
sudo setquota -u "$USERNAME" $SOFT_LIMIT $HARD_LIMIT 0 0 /srv/share
```

   - Définition du mot de passe FTP
   - Création du répertoire personnel `/srv/www/USERNAME` avec une page d'accueil personnalisée (voir ci-dessous)

Figure 9. Page MrRoland.tomananas.lan

- Configuration automatique du virtual host Apache HTTPS (*:443) avec certificat wildcard
- Ajout du compte Samba

4. **Connexion au serveur SQL (mysql-01)**
   - Création d'une base de données dédiée au client
   - Création d'un utilisateur SQL avec droits complets sur cette base
5. **Connexion au serveur DNS (dns-ntp-01)**
   - Ajout d'un enregistrement A dans la zone tomananas.lan
   - Incrément du serial DNS
   - Redémarrage du service DNS (named)
6. **Affichage du résumé final**
   Toutes les informations utiles au client sont récapitulées à la fin du script :
   - Adresse de connexion FTP
   - Informations SQL pour phpMyAdmin
   - Lien HTTP/HTTPS du site
   - Nom de domaine

6.2.2 Supprimer un utilisateur

Lorsque nous savons créer un client, il est tout aussi important de pouvoir le supprimer proprement, en archivant ses données, en libérant les ressources, et en nettoyant les services associés. Pour cela, nous avons développé un script automatisé qui réalise toutes les étapes critiques de la suppression d'un client :

1. **Chargement des variables de setup_env.sh**
2. **Création d'un dossier d'archives**
   Un dossier `/home/backup/archives/NOMUTILISATEUR` est créé sur le serveur de sauvegarde pour y stocker les fichiers et bases de données avant suppression.
3. **Suppression sur le serveur Web**
   Le script se connecte au serveur web-ftp-01 et effectue :
   - La sauvegarde du site web personnel dans une archive .tar.gz
   - La suppression du compte utilisateur Linux et de son dossier `/srv/www/USERNAME`
   - La suppression de ses vhosts Apache HTTP/HTTPS
   - La suppression de son compte Samba
   - Le rechargement d'Apache

   L'archive est ensuite transférée vers le serveur de sauvegarde et supprimée du serveur web.
   Tous les transferts de fichiers dans notre infrastructure sont réalisés à l'aide de la commande `scp`, qui repose sur SSH.

   ```
   scp -i "$SSH_KEY" ec2-user@$WEB_PRIVATE_IP:/tmp/$ARCHIVE_NAME
   $REMOTE_USER@$BACKUP_HOSTNAME:$ARCHIVE_DIR/
   ```

   Ceci permet une copie sécurisée, chiffrée de bout en bout, sans mot de passe, grâce à l'utilisation d'une clé privée SSH associée à l'utilisateur ec2-user.
4. **Suppression sur le serveur SQL**
   - Se connecter et tenter un dump SQL de la base de données du client (username_db) si elle existe

- Supprimer la base de données et l'utilisateur SQL
- Transférer le dump SQL vers le serveur de sauvegarde
- Supprimer le fichier temporaire

5. **Suppression DNS**
   - Se connecte et supprime l'enregistrement A correspondant à username.tomananas.lan
   - Redémarre le service named pour appliquer les modifications

À la fin du script, toutes les données sont archivées dans `/home/backup/archives/USERNAME` :

```
ll /home/backup/archives/
drwxr-xr-x. 2 backup   backup   74 May 14 16:14 MrRoland
drwxr-xr-x. 2 backup   backup   66 May 14 15:53 demo
drwxr-xr-x. 2 backup   backup   70 May 14 14:41 pryvit
```

6.2.3 Restaurer un utilisateur

Maintenant que nous savons ajouter et supprimer proprement des utilisateurs, il était logique d'ajouter la possibilité de les restaurer si un client revenait plus tard et souhaitait récupérer son site, ses fichiers ou sa base de données. Pour cela, nous avons développé un script de restauration complet qui utilise les archives stockées sur le serveur de sauvegarde lors de la suppression du client.

Étapes réalisées par le script :

1. **Chargement des variables de setup_env.sh**
2. **Recherche automatique des dernières archives**

   Le script détecte automatiquement la dernière version disponible grâce à la commande `ls -t`, qui trie les fichiers par date de modification (du plus récent au plus ancien).

   ```
   LATEST_TAR=$(ls -t "$ARCHIVE_DIR/$USERNAME"*.tar.gz | head -1)
   LATEST_SQL=$(ls -t "$ARCHIVE_DIR/$USERNAME"*.sql.gz | head -1)
   ```

   Si l'une des deux est absente, le script s'arrête pour éviter une restauration incomplète.
3. **Saisie sécurisée des mots de passe**

   Le mot de passe FTP/Samba de l'utilisateur et celui de sa base SQL sont saisis manuellement, avec confirmation.
4. **Transfert temporaire des archives**

   Les deux fichiers sont copiés temporairement dans `/tmp` puis envoyés via `scp` sur les serveurs Web et SQL.
5. **Restauration sur le serveur Web**
   - La création de l'utilisateur Linux (s'il n'existe pas)
   - L'extraction de l'archive du site dans `/srv/www/username`
   - La configuration des permissions
   - La recréation des vhosts Apache
   - La recréation du compte Samba avec le mot de passe saisi
6. **Restauration de la base de données**
   - Recrée l'utilisateur SQL avec son mot de passe
   - Crée la base de données si nécessaire
   - Restaure le contenu depuis le fichier .sql.gz via gunzip | mysql.

7. **Réinsertion DNS**
8. **Nettoyage final**
    - Les fichiers temporaires sont supprimés du répertoire /tmp sur le serveur de backup.

Résultat

En quelques secondes, le client récupère :

- Son site Web dans l'état où il l'avait laissé
- Sa base de données avec tous ses contenus
- Un accès FTP et Samba fonctionnel
- Un domaine username.tomananas.lan de nouveau actif.

6.4. Gestion des sauvegardes

6.3.1 Backup des fichiers de services

Au-delà des fichiers utilisateurs, notre infrastructure contient des fichiers critiques liés aux services (configuration web, DNS, bases de données) qui changent régulièrement.
Il est essentiel de les sauvegarder quotidiennement pour permettre une restauration rapide en cas d'incident.
Nous avons identifié les éléments suivants :

✓ **Serveur Web (web-ftp-01)**
- /etc/httpd/sites-available/ — fichiers de configuration des VirtualHosts Apache (un par utilisateur)
- /etc/httpd/sites-enabled/ — liens symboliques des VirtualHosts activés
- /srv/www/ — dossiers personnels des utilisateurs (contenu web)
- /srv/share/ — dossier partagé en lecture/écriture via Samba et NFS

✓ **Serveur DNS (dns-ntp-01)**
- /etc/named.conf — fichier principal de configuration BIND
- /var/named/ — répertoire contenant les fichiers de zone (directe et inverse)

✓ **Serveur SQL (mysql-01)**
- Dump de toutes les bases de données via mysqldump --all-databases

Fonctionnement du script utils_backup.sh :

1. **Chargement des variables de setup_env.sh**
2. **Création du dossier de sauvegarde** `/home/backup/backups/AAAA-MM-JJ/`
3.
4. **Sauvegarde distante des services Web et DNS**
   Le script utilise une fonction backup_host qui :
    - Se connecte à distance via `ssh -i`
    - Archive les fichiers définis
    - Transfère l'archive compressée en local
5. **Dump complet de MySQL**

Un dump compressé de toutes les bases (mysqldump --all-databases | gzip) est généré et stocké localement.

6. **Nettoyage automatique**

Les sauvegardes de plus de 30 jours sont supprimées pour éviter de saturer le disque.

```
find "$BACKUP_HOME/backups/" -type f -mtime +30 -exec rm -f {} \;
```

À la fin du script, toutes les données sont archivées dans `/home/backup/backups/DATE` :

```
ll /home/backup/backups/
drwxrwxr-x. 2 backup backup  4096 May 12 21:32 2025-05-12
drwxrwxr-x. 2 backup backup  4096 May 13 08:45 2025-05-13
drwxr-xr-x. 2 backup backup  4096 May 14 15:20 2025-05-14
```

Et à l'intérieur du dossier :

```
ll /home/backup/backups/2025-05-14
-rw-r--r--. 1 root root   1519 May 14 15:20 dns-ntp-01-2025-05-14.tar.gz
-rw-r--r--. 1 root root 282221 May 14 15:20 mysql-01-mysql-2025-05-
14.sql.gz
-rw-r--r--. 1 root root   4263 May 14 15:20 web-ftp-01-2025-05-14.tar.gz
```

6.3.2 Restauration de backup

Faire des sauvegardes régulières est essentiel… mais elles n'ont de valeur que si l'on est capable de les restaurer rapidement et efficacement. C'est pourquoi nous avons également conçu un script de restauration des services, capable de rétablir tout ou partie de l'infrastructure à une date donnée.

Fonctionnement :

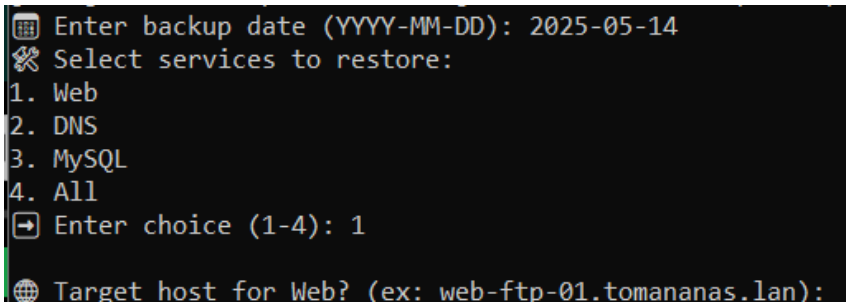
```
[bash /home/backup/scripts/utils_restore_backup.sh
```

1. **Saisie de la date de sauvegarde**

L'utilisateur entre la date (format YYYY-MM-DD) de la sauvegarde souhaitée.

Le script vérifie que le dossier /home/backup/backups/YYYY-MM-DD/ existe.

2. **Choix des services à restaurer**



3. **Saisie des machines cibles**

Pour chaque service sélectionné, le script demande à l'utilisateur le nom de la machine cible (par exemple `web-ftp-01.tomananas.lan`) sur laquelle déployer la restauration.

Pour chaque service sélectionné, le script effectue :

Web

- Envoie l'archive `.tar.gz` via ssh

- Extraction sur / de la machine cible (inclut `/etc/httpd/` et `/srv/www/`, `/srv/share/`, `/etc/httpd/sites-available/`, `/etc/httpd/sites-enabled/`)

DNS

- Envoie l'archive contenant `/etc/named.conf` et `/var/named/`
- Extraction automatique sur le serveur DNS

MySQL

- Envoie le dump compressé `.sql.gz`
- Exécution du dump via mysql en ligne de commande à distance

Toutes les opérations sont sécurisées via SSH (`ssh -i clé.pem`) et vérifiées étape par étape.

7. Plan de sauvegarde

Dans le cadre de ce projet, nous avons mis en place un plan de sauvegarde automatique répondant aux besoins de notre architecture.

7.1. Analyse des besoins

Avant de déployer notre stratégie de sauvegarde, nous avons répondu aux questions fondamentales:

Que faut-il sauvegarder ?

- ✓ Les données des clients :
  - Sites web des utilisateurs (`/srv/www/USERNAME`)
  - Bases de données personnelles MySQL
- ✓ Les configurations critiques de service :
  - Apache (VirtualHosts)
  - Samba (fichier smb.conf)
  - NFS (/etc/exports)
  - DNS (named.conf, zones DNS)
  - NTP (chrony.conf)
  - FTP (vsftpd.conf)
- ✓ Les données utilisateurs supprimés (pour une éventuelle restauration)

Où stocker les sauvegardes ?

- ✓ Sur un serveur de sauvegarde dédié (`admin-backup-01`) dans notre VPC
- ✓ Arborescence structurée par date :
  `/home/backup/backups/YYYY-MM-DD/`
  `/home/backup/archives/<utilisateur>/`

À quelle fréquence ?

- ✓ Sauvegarde complète quotidienne, lancée automatiquement à 1h du matin.
- ✓ Nettoyage automatique des sauvegardes vieilles de plus de 30 jours.

Quelles menaces couvrir ?

- ✓ Suppression ou modification accidentelle de fichiers
- ✓ Défaillance logicielle (ex. : erreur Apache, corruption DNS)

- ✓ Besoin de restauration manuelle (recréer un utilisateur, ou tout un service)
- ✓ Erreur humaine

7.2. Actions mises en place

✓ **Sauvegarde automatique quotidienne**

Un script dédié `utils_backup.sh` est exécuté chaque nuit à 1h via crontab.

```
0 1 * * * /home/backup/scripts/utils_backup.sh >> /var/log/restore.log 2>&1
```

✓ **Restauration à la demande**

- Script de restauration utilisateur : réintègre un ancien utilisateur complet (site, bd, DNS etc.)
- Script de restauration système : restaure un ou plusieurs services (Web, DNS, MySQL) à une date donnée à partir des archives

7.3. Améliorations possibles

Bien que notre projet soit limité à une durée d'une semaine, voici des pistes pour le rendre prêts pour aller en production:

- Activation des snapshots EBS hebdomadaires pour chaque volume attaché aux serveurs (via AWS Backup). Ça permettrait de restaurer l'ensemble du serveur dans l'état exact à une date donnée, même en cas de crash total.
- Export automatique des archives sur un bucket S3, un NAS distant ou une machine hors site.

8. DRP

Le Disaster-Recovery-plan ci-dessous reprend l'ensemble des points à suivre en cas de différents use case scénarios. Les manipulations à appliquer sont donc des procédures claires pouvant être suivies facilement. Les problèmes d'innascibilité, panne, suppression involontaire, arrêt brusque sont abordés.

8.1. Matériel actif

Se référer au point 3 de ce document (Gestion d'infrastructure dans le cloud) reprenant le tableau de toutes les instances et serveurs actifs sur notre réseau cloud privé. Chaque instance doit être en activité et opérationnel, en cas de panne veuillez réparer ou remplacer au plus vite l'instance problématique.

8.2. Problèmes courants

Voici une liste non exhaustive des problèmes les plus courants à résoudre, si aucune solution n'est trouvée, alors passer à un diagnostic plus poussé plus loin dans ce point DRP.

ID	Description du problème	Description des solutions courantes	Manipulations
1	VPN non fonctionnel	• Carte réseaux client mal configurée • Instance VPN AWS Eteinte	• Vérifier état cloud

		• Clef de VPC appropriée (ip pub) • Vérifier IP Elastique AWS ←→ IP publique de la clef • Security Group AWS modifié	• Vérifier configuration réseau client
2	Connexion SSH impossible mais ping valide	• Clef SSH non valide • Firewall Serveur modifié • SSH Bloqué côté client • Security Group AWS modifié	• Vérifier IP publique VPN utilisée • Vérifier état cloud • Connexion cloud instance
3	Aucun ping sur instance (VPN connecté)	• Instance éteinte • Firewall Serveur modifié • ICMP Bloqué côté client	• Vérifier état cloud • Connexion cloud instance
4	Service Web ou Service fichiers indisponible	• Instance web-ftp éteinte • VPN non connecté • Firewall modifié • Security group AWS modifié	• Vérifier état cloud • Connexion cloud instance
5	Backend non fonctionnel	• Instance backend éteinte • VPN non connecté • Firewall modifié • Security group AWS modifié	• Vérifier état cloud • Connexion cloud instance • Vérifier connexion réseau
6	Accès client en read-only sur leur dossier	• Droits d'accès /srv/www/%USER% mal défini sur instance web-ftp • Mauvais user utilisé (improbable mais au cas où…)	• Modifier droit d'accès sur dossier chrooté client 755

8.3. Diagnostique cloud

Dans le cas où une des instances du cloud privé tombe en panne complètement où devient indisponible pour cause inconnue, voici la procédure à suivre pour reconnaitre le problème :

1. Connexion VPN au VPC via OpenSSH client.
2. Ping 10.42.0.100 (VPN) pour vérifier la connexion
3. Si connexion établie continuer la manipulation, sinon vérifier état du VPN et AWS
4. Ping de l'instance actuellement problématique.
5. Si Ping abouti mais impossible de ssh alors vérification de l'état de la machine via connexion cloud et non ssh sur AWS EC2 > instance > Se connecter > Connect using a Public IP

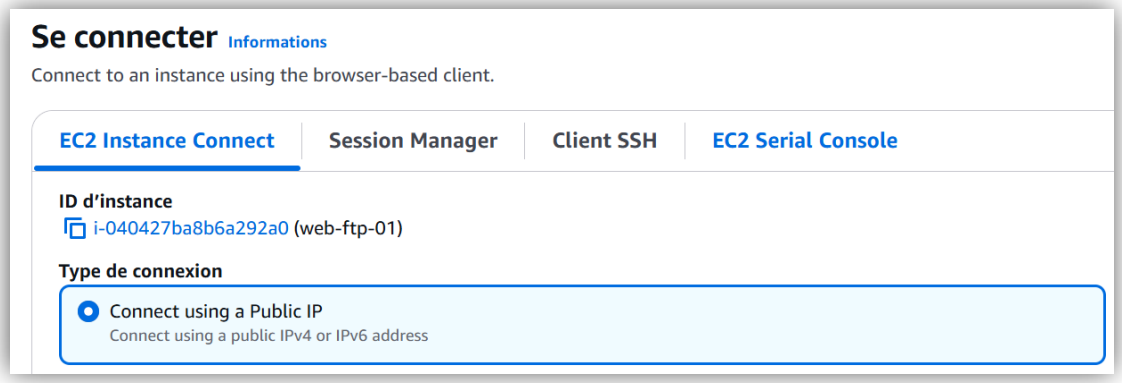


Figure 10. Interface AWS pour se connecter à une instance

6. Si Connexion impossible à ce niveau là, redémarrage de l'instance ac2 via interface cloud aws.

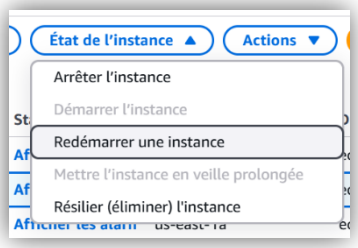


Figure 11. Interface AWS pour redémarrer une instance

7. Si après redémarrage, connexion toujours impossible en ssh ou adresse publique, alors suppression de l'instance et déploiement d'une nouvelle instance de service équivalent et utilisation de l'IP fixe déterminée dans le fichier de d'environnement setup_env.sh

8.4. Redéploiement d'urgence

En cas de suppression volontaire ou involontaire d'une ou plusieurs instances, ou encore d'un besoin de redéploiement d'un nouveau réseau complet sur un nouveau cloud privé, pas de panique, les scripts de déploiement (setup_*.sh) permettent un déploiement complet du VPC en moins de 30minutes pour le pire des cas avec l'ensemble des instances.

1. Créations d'un ou plusieurs instances concernées avec le même nom que leurs prédécesseurs, en utilisant la création par modèle EC2

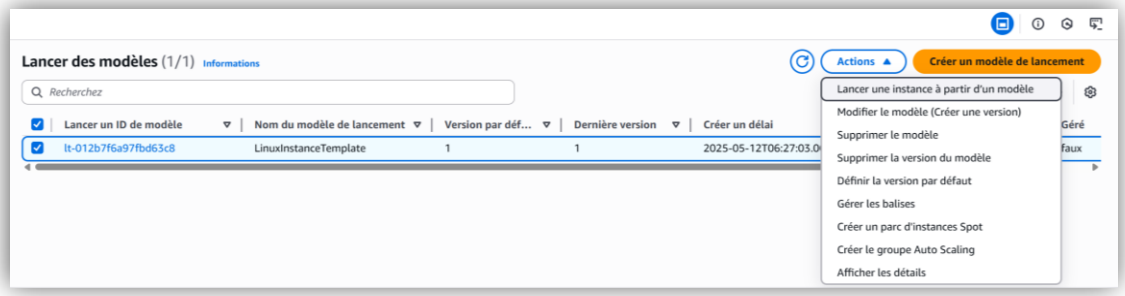


Figure 12. Interface AWS pour lance une instance à partir d'un modèle

2. Connexion SSH à l'instance après connexion au VPN. IPV4 privée visible dans les détails de l'instance fraichement crée.
3. `scp` du fichier de configuration des variables d'environnement (`setup_env.sh`)
4. Déplacer le `setup_env.sh` dans /root -> `/root/setup_env.sh`
5. `scp` du fichier de configuration de base des serveur (`setup_server_base.sh`)
6. Exécution en root : `bash setup_server_base.sh`, ensuite répondre aux questions posées
7. `scp` du fichier de configuration du service spécifique comme `setup_web.sh`, `setup_dns-ntp.sh`, `setup_CA.sh`, `setup_backup.sh`, `setup_backend.sh`.
8. Exécution en root du fichier de configuration avec bash.
9. Suppression des fichiers de setup utilisés
10. Modification AWS avec ajout d'une carte réseau comprenant l'ipv4 spécifique de l'instance remplacée pour ne pas modifier toutes les autres configurations fonctionnelles.

9. Sécurisation des serveurs

9.1. Antivirus ClamAV +LMD

L'antivirus sur le serveur a été mis en place en combinant deux outils complémentaires : ClamAV et Linux Malware Detect (LMD). ClamAV est utilisé comme moteur principal d'analyse, avec des mises à jour automatiques des signatures virales grâce au service freshclam, activé dès l'installation. LMD, quant à lui, est installé manuellement depuis les sources officielles et configuré pour utiliser ClamAV en backend. Cela permet de profiter à la fois de la rapidité de ClamAV et de la détection ciblée de LMD, spécialisée dans les menaces propres aux environnements Linux.

Un script a été créé pour lancer une analyse complète à la demande ou de manière automatique. Ce script met d'abord à jour les signatures de ClamAV, puis lance un scan en profondeur avec clamdscan en mode multithread, suivi d'un scan avec LMD. Enfin, une tâche cron exécute ce script deux fois par jour (à 10h et 18h), avec les résultats enregistrés dans un fichier de log. L'administrateur peut ainsi s'assurer que les chemins sensibles du système sont régulièrement vérifiés sans intervention manuelle.

Il est important de noter que les heures ne sont pas optimales pour un serveur de production réel, mais dans notre cas ce sont des heures auxquelles le laboratoire AWS academy sont disponibles et allumées.

9.2. Fail2ban

Nous avons installé et activé Fail2ban sur l'ensemble de nos machines afin de limiter les tentatives de connexion malveillantes, via la configuration commune du script `setup_server_base.sh`.
Cependant, sa configuration s'est limitée à la protection du service SSH, à travers un fichier `jail.local` très basique. Cette configuration par défaut permet de bloquer automatiquement une adresse IP après plusieurs échecs d'authentification SSH, ce qui renforce un minimum la sécurité d'accès distant.
Cela dit, nous reconnaissons que cette mise en place est incomplète. D'autres services exposés comme vsftpd, Apache ou encore Samba auraient également dû être protégés individuellement, en ajoutant des jails spécifiques adaptées à leurs journaux respectifs. Une configuration plus poussée aurait permis d'augmenter significativement la résilience du système face aux attaques ciblées sur ces services.

10. Problèmes rencontrés

10.1. Configuration FTP

Nous avons eu quelques problèmes lors de la configuration des services de fichiers pour l'accès des clients. Nous avions une mauvaise compréhension du choix de quel type de ftp sélectionner et mettre en place.
En premier, nous avions mis en place un sftp sous sshd qui ne fonctionnait pas bien pour cause d'une mauvaise configuration, ensuite lors de la résolution des problèmes le choix a été remis en question, car l'envoie d'une clef pour la connexion au client semblait embêtant et surdimensionné, donc nous sommes ensuite passé sur un ftps (ftp over tls) qui a été notre configuration finale sous vsftpd.

10.2. Adapter la configuration LVM

Les premiers scripts de configuration que nous avions préparés ne reprenaient aucune configuration LVM car les instances de tests crées avant le projet n'avait pas de disques supplémentaires. Donc une partie du travail d'adaptation a été de réécrire la partie de configuration pour les volumes AWS, avec l'ajout des VG, LV, points de montage supplémentaire pour les services spécifiques comme dossier share, web, backend.

10.3. Mauvaise configuration LVM web

Nous avions configuré un seul volume-group sur le serveur web de production en pensant que notre configuration était bonne. Mais c'était avant de penser au dossier share. Donc évidemment il était primordial de créer un volume group supplémentaire pour séparer le dossier de partage publique du dossier web. Le problème était que les deux volumes physiques étaient déjà utilisés dans le volume group du dossier web. Donc nous avons dû, sur le serveur de production directement, détacher le volume physique du vg et pv, créer un nouveau groupe de volume et le monter sur le dossier share !
Cette manipulation était très importante, donc nous avons recrée un environnement similaire sur une autre instance pour scripter la manipulation à faire et l'appliquer sur l'instance de production. Ceci a parfaitement fonctionné (après plusieurs essais).

10.4. Affichage des sites des utilisateurs supprimés & catch-all

Le problème majeur qui a été dépanné pour la configuration web durant cette semaine de projet dans notre infrastructure a été l'apparition absolument incompréhensible de sites qui restaient dans un cache quelque part après la suppression d'un utilisateur possédant un site auparavant. Nous avions pourtant vidé les caches dns, http, les lignes A liées supprimées du DNS, flush dns client, suppression des liens des sites-enables et sites-available, suppression du dossier web client, tout était complètement nettoyés et pourtant parfois un site apparaissait de nulle part.

Nous avons même réalisé un tcp dump et analyse wireshark du trafic lors de la récupération du site en http/https sur nos clients et serveur web…

Après l'impossibilité de la résolution de ce problème directement, une solution nous a sauvé ! Un site catch-all avec une wildcard récupère en priorité les sites non résolus. En réalité le premier fichier trouvé dans les vhost est considéré comme catch-all par apache, donc le site `000-default-ssl.conf` est en premier récupéré, avec un dossier web pointant vers `/srv/www/default`. Donc comme ça plus de problème de site en cache résolus incorrectement.

10.5. Automatisation des scripts

Certains problèmes bloquaient nos scripts `setup_*.sh` et `utils_*.sh` nous ne permettaient pas d'avoir un résultat optimisé de déploiement. Par exemple dans le cas où le script avait déjà été run des conditions `|| True` devaient être ajoutées, ou bien des `-y` manquants, en réalité beaucoup de petits problèmes qui coupaient les scripts au milieu. Comme cette ligne ci, sans `|| True` ajouté script s'arrêtait après la création de variable :

```
DIR=$(tar tzf maldetect-current.tar.gz | head -1 | cut -d'/' -f1) || true
```

Donc à chaque fois les tests ont dû être réalisés sur plusieurs instances différentes. Finalement nous sommes arrivés à un total de 17 instances ajoutées rien que pour les tests.

10.6. Configuration DNS écrasé par DHCP AWS

Nous avons assez vite remarqué que les `/etc/resolve.conf` des instances étaient écrasés par le DHCP AWS appliquant les options DHCP désignées. Par exemple le search ou encore serveur DNS principal passait de notre serveur 10.42.0.37 à une autre configuration incorrecte. Pour résoudre ce problème finalement nous avons mis en place cette solution simple :

```
sudo rm -f /etc/resolv.conf
echo -e "nameserver 10.42.0.37\nnameserver 8.8.8.8" | sudo tee
/etc/resolv.conf
sudo chattr +i /etc/resolv.conf
```

Une solution plus radicale aurait été de modifier les options DHCP du network utilisé pour les instances serveurs du VPC.

10.7. Accès guest Samba Windows

Nous avons configuré un partage Samba public accessible en guest, sans authentification. Ce partage fonctionne parfaitement sous Linux, mais Windows bloque par défaut les connexions invitées non sécurisées, pour des raisons de sécurité. Nous avons contourné ce blocage en modifiant la base de registre de Windows pour autoriser les connexions en mode invité.

Étapes à suivre**:**

1. Ouvrir l'éditeur de registre : Win + R → taper regedit → Enter
2. Naviguer jusqu'à la clé suivante :
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters
3. Créer une nouvelle valeur DWORD 32 bits :
   - Clic droit → Nouveau → Valeur DWORD (32 bits)
   - Nom : AllowInsecureGuestAuth
   - Valeur : 1
4. Redémarrer le service de réseau Windows :
   Ouvrir un terminal (cmd) en administrateur et exécuter :

```
net stop workstation
net start workstation
```

5. Connexion réussie !

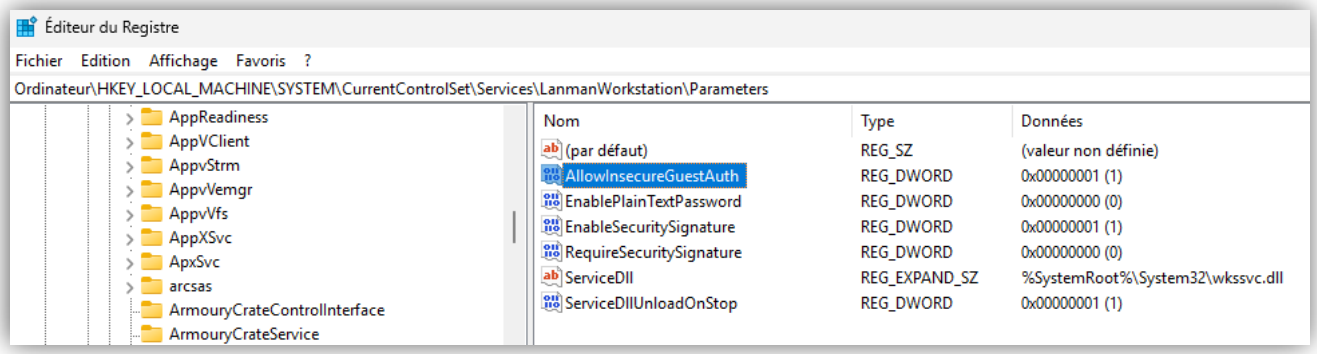


Figure 13. Visualisation de la nouvelle règle

11. Améliorations

11.1. Options de montage

Nous devrions ajouter des options de montages spécifiques pour les dossiers accessibles aux clients, comme sur l'instance web-ftp et ses répertoires `/srv/www` et `/srv/share` où il aurait été plus qu'important d'ajouter un noexec.

11.2. Alertes spécifiques sur NetData

Tous les services principaux des serveurs sont déjà monitorés, cependant il serait intéressant d'ajouter des alertes personnalisées sur ces services pour prévenir quand un d'entre eux est stoppé. Par exemple prévenir quand le dns ne fonctionne plus, sans devoir s'en apercevoir manuellement.

11.3. Tests de restauration des backup automatiques

Il serait intéressant, même si ce n'était pas dans nos priorités durant la semaine, d'ajouter un script automatique exécuté toutes les semaines, le samedi soir par exemple, qui restaure sur un serveur de test les derniers backups réalisées dans le but de vérifier leur intégrité. Cette vérification permet donc de ne pas se retrouver avec des sauvegardes et archives inutilisables au moment dans le besoin.

11.4. Sécurisation phpmyadmin

Nous avons configuré phpMyAdmin en http(80), mais ce n'est vraiment pas une bonne idée, il faut impérativement passer ce site en https(443) pour sécuriser l'accès des clients. Le certificat peut être signé avec notre CA interne qui plus est…

11.5. Vérifications supplémentaires dans les scripts

Nos scripts actuels effectuent déjà des vérifications importantes (présence des fichiers requis, exécution en tant que root, etc.), mais il est toujours possible d'aller plus loin pour renforcer la robustesse et la sécurité.
Par exemple, il faudrait mettre des regex pour vérifier la complexité des mots de passe, longueur/ caractères autorisés des noms des utilisateurs et des machines. Ou encore empêcher la création d'un client root .

11.6. FTPS signé par le CA

Nous avions configuré un FTP over TLS avec un certificat autosigné via openSSL, cependant pour ce projet avec notre configuration nous aurions pu signer le certificat avec notre CA en interne utilisé pour le serveur web ! Cette option n'a pas été pensée en premier lieu, même si après coup cela aurait été intéressant.

11.7. E-mails automatiques avec création d'un utilisateur

Nous avions prévu de prévenir les clients via un email (pour le moment manuel) pour lui donner les mots de passes d'accès à son répertoire web. Cependant il serait possible et même très intéressant d'automatiser la tâche via un envoie automatique de courriel par la suite avec notre propre serveur SMTP interne. Ce n'était cependant pas dans nos priorités les plus hautes durant ce projet.

12. Conclusion

Le projet Linux 2024-2025 a été une expérience particulièrement enrichissante, notamment sur les aspects liés au cloud et au déploiement de services web. Cette mise en pratique nous a permis d'acquérir une compréhension plus approfondie des services configurés. En effet, rien ne vaut l'expérimentation concrète pour consolider les connaissances théoriques.

L'un des points les plus positifs a été l'opportunité de manipuler des outils de configuration cloud de manière plus avancée, ce qui représente une réelle plus-value pour notre formation. À l'inverse, nous avons rencontré certaines difficultés liées à un cahier des charges manquant parfois de clarté. De plus, le fait que la note maximale ne soit pas attribuée à la complétion totale du cahier de charges nous semble quelque peu démotivant.
Nous souhaitons également souligner l'importance du choix du binôme, qui, selon nous, doit impérativement rester une liberté accordée dans ce type de projet, tant l'efficacité du travail en dépend.
De manière générale, notre ressenti est très positif. Nous avons conscience que certaines configurations et scripts auraient pu être davantage développés, mais nous avons fait le choix stratégique de prioriser les éléments que nous jugions essentiels. Les pistes d'amélioration sont listées plus haut sur ce document et témoignent de notre volonté et connaissances quant aux possibilités de pousser ce projet encore plus loin.

13. Bibliographie

- **Syllabus :**
  Antoine Malaise, *Administration Linux - théorie*, Haute Ecole en Hainaut, Année académique 2024-2025

- **Sources électroniques :**
  Documentation AWS VPC. Disponible à l'adresse :

  https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html

  Documentation vsftpd Red Hat. Disponible à l'adresse :
  https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/deployment_guide/s2-ftp-servers-vsftpd

  Documentation Bind. Disponible à l'adresse :
  https://bind9.readthedocs.io/en/latest/chapter3.html

  Documentation Chrony. Disponible à l'adresse :
  https://chrony-project.org/documentation.html

  DNSSEC configuration. Disponible à l'adresse :
  https://www.zenarmor.com/docs/linux-tutorials/how-to-install-dnssec-on-ubuntu-linux

  OpenSSL CA configuration. Disponible à l'adresse :
  https://networklessons.com/miscellaneous/openssl-certification-authority-ca-ubuntu-server

14. Annexes

14.1. Journal de bord

11/05/2025

- 8h30 : Découverte du cahier de charge et environnement AWS EC2
- 9h00 : Création instance backup/déploiement/admin/… (@Tom Deneyer)
- 9h10 : Mise à jour script en fonction du nouveaux volumes disques disponibles (@Tom Deneyer)
- 9h : Installation et configuration en script de NFS et Samba + configuration Windows (@KOZLENKO Anastasiia)
- 11h : Test des scripts de déploiements du dns, ftp et web, ntp, etc… avec lvm
- 13h : Fix des scripts pour utilisation des variables d'environnements pour les déploiements (@Tom Deneyer)
- 14h : Déploiement DNS complet final (@Tom Deneyer)
- 14h : Configuration et debug windows + procédure samba (@KOZLENKO Anastasiia)
- 15h15 : Configuration de base déployés sur serveur admin-backup-01
- 16h : Samba déploiement script validé et testé +nfs, autofs (@KOZLENKO Anastasiia)

En conclusion le lundi fin de journée nous avons scripté le déploiement complet de plusieurs serveurs avec services dédiés:

- web-ftp-01 : httpd, php, ftp…
- dns-ntp-01 : dns named, ntp chronny
- admin-backup : backup server pour déployer les autres scripts etc
- mysql-01 : serveur backend mysql

Problèmes rencontrés:

- Adapter nos scripts avec des lvm
- Corrections des sources des variables d'environement
- Samba sous windows

12/05/2025

- 08h00 : Déploiement de tous les services sur environnement final avec scripts préparés (@Tom Deneyer)
- 08h10 : Scripting de backup (@KOZLENKO Anastasiia)
- 09h00 : Début des recherches monitoring (@Tom Deneyer)
- 10h00 : Mise en place d'un solution Pour fuseaux horaire incorrect sur setup de base (@KOZLENKO Anastasiia)
- 10h00 : Mise en place du monitoring netdata sur instances +script (@Tom Deneyer)
- 11h00 : Solution pour garder le dns dans le /etc/resolve.conf (@KOZLENKO Anastasiia)
- 11h00 : Https sans CA sur le webserver (@Tom Deneyer)
- 12h00 : Scripting Delete User (@KOZLENKO Anastasiia)
- 14h00: Mise en place Https sécurisé avec script pour CA sur admin-backup (@Tom Deneyer)

- 14h00: Scripting Restore User (@KOZLENKO Anastasiia)

En conclusion le mardi fin de journée nous avons la création de backup, ajout des utilisateurs, suppression et archivage complètement automatisés ! Nous avons également réglé des petites choses embêtantes. Par exemple: après d'avoir remarqué que l'heure des machines est en retard de 2h, nous avons précisé le fuseau horaire, et également pour le resolve.conf qui se met à jour chaque 2min.
Problèmes rencontrés :

- Configuration https prend beaucoup de temps
- Script restore_user est compliqué à cause des problèmes des droits, en plus tests de suppression → restauration → suppression → restauration foirent si on fait la suppression d'user qui a été déjà restauré etc..
- On tombe sur la page web d'un autre utilisateur quand on met l'url d'un utilisateur supprimé.

13/05/2025

- 8h: Création de catch-all (@Tom Deneyer)
- 8h: Création du fichier my.cnf pour retirer le mdp en claire lors des connexions du serveur de admin-backup-01 dans les scripts (@KOZLENKO Anastasiia)
- 9h: Scripting des quotas (@KOZLENKO Anastasiia)
- 9h: Mise en place de serveur CA (@Tom Deneyer)
- 9h30: Debug script de restauration (@KOZLENKO Anastasiia)
- 10h: Ajout de partition pour /www/share (@Tom Deneyer)
- 10h30: Debug script setup de base, car nous avons rajouté des choses au fur et au mesure, mais pas testé (@KOZLENKO Anastasiia)
- 11h: Réparation de lvm dans le script setup_web (@Tom Deneyer)
- 14h: Préparation de démonstration (@Tom Deneyer, @KOZLENKO Anastasiia)
- 14h30h: Client NFS machine de test(@Tom Deneyer)
- 15h: Présentation

Mercredi c'était le dernier jour pour nous. Nous avons finalisé l'infrastructure avant 14h et nous avons présenté le fonctionnement automatisé de création d'utilisateurs, leurs accès aux ressources et au site web, ainsi que la suppression et la restauration complète.
Problèmes rencontrés :

- Script de setup web fonctionnait plus, il faillait passer du temps pour débuguer
- Pendant la démonstration, l'accès au dossier privé Samba a échoué, pourtant 10 min avant avec un autre utilisateur ça s'est bien passé.

14.2. Notes générales

Voir fichiers de l'archive.

14.3. Exemples de déploiements

Voir fichiers de l'archive.

14.4. Scripts complets

Voir fichiers de l'archive.

14.5. Tâches Notion

Voir fichiers de l'archive.

14.6. Fichiers de configurations services

Tous les fichiers sont disponibles dans l'archive, trié par services

14.7. Firewalls

Un dossier Firewall contient un .txt par serveur avec le résultat du « firewall-cmd --list-all »

14.8. Dashboard de monitoring

Un printscreen des quatre dashboard sont disponibles dans l'archive.

14.9. Courriels alertes monitoring netdata

Différents courriels d'alertes disponibles dans l'archive.