

Compte Rendu Technique : Infrastructure B2Tech (Niveau 1)

1. Choix Technologiques & Justification

Pour répondre aux besoins de la DSI de B2Tech Solutions, nous avons opté pour le couple **Ubuntu Server + MySQL**.

- **Ubuntu Server** : Choisi pour son cycle de mise à jour prévisible (LTS) et sa gestion native des dépôts APT, ce qui facilite la maintenance à long terme.
 - **MySQL** : Référence de l'industrie pour la gestion de bases de données relationnelles, offrant une excellente intégration avec Apache.
-

2. Phase 1 : Déploiement du Système (Ubuntu Server)

2.1 Installation et Initialisation

L'installation a été réalisée via une image ISO chargée sur une machine virtuelle.

- **Partitionnement** : Utilisation de LVM (Logical Volume Manager) pour permettre une extension future du stockage si la base de données s'alourdit.
- **Mise à jour** :

Bash

```
sudo apt update && sudo apt upgrade -y
```

2.2 Configuration Réseau

Nous avons configuré une adresse IP statique (ou réservée via DHCP) pour garantir que le serveur soit toujours joignable à la même adresse par les collaborateurs.

```
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
      qlen 1000
      link/ether 00:0c:29:87:19:e2 brd ff:ff:ff:ff:ff:ff
      altname enp2s1
      inet 192.168.118.20/24 brd 192.168.118.255 scope global ens33
        valid_lft forever preferred_lft forever
      inet6 fe80::20c:29ff:fe87:19e2/64 scope link
        valid_lft forever preferred_lft forever
```

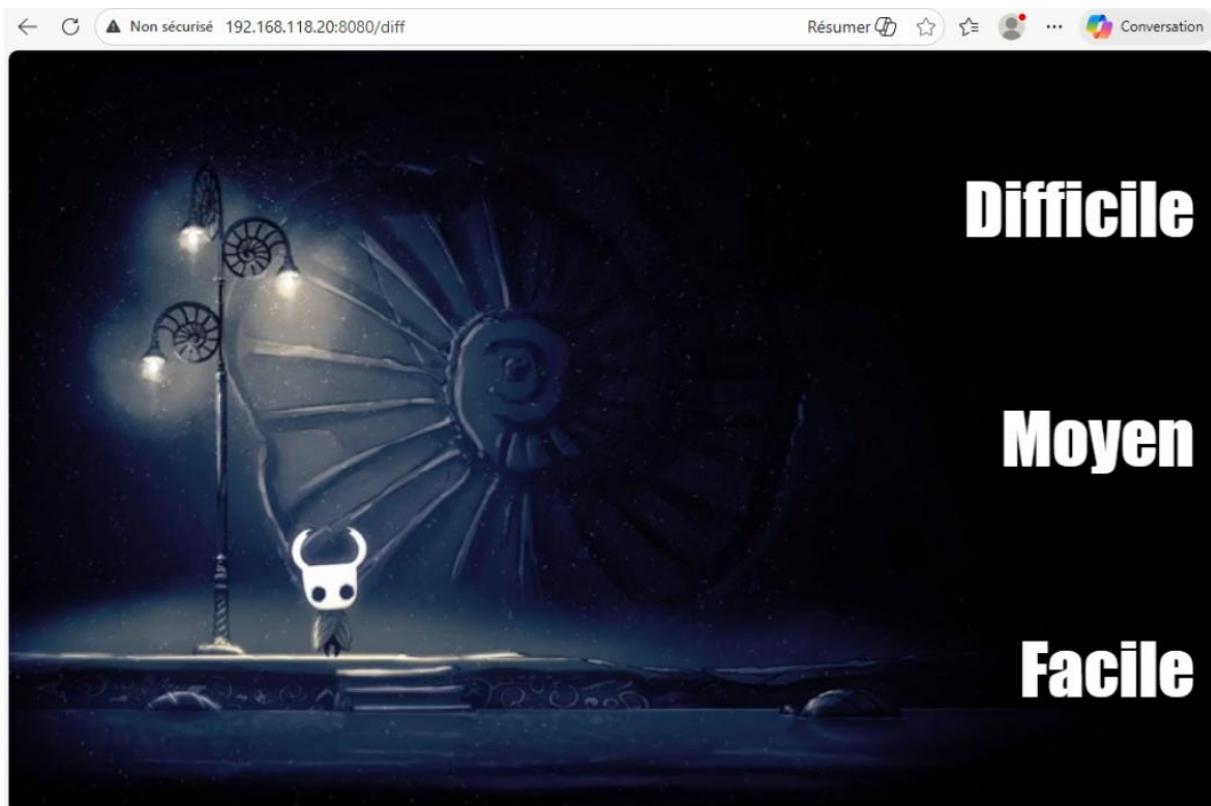
3. Phase 2 : Serveur Web Apache

3.1 Installation & Accessibilité

Installation du service via : sudo apt install apache2. Le pare-feu (UFW) a été configuré pour autoriser le trafic HTTP :

Bash

```
sudo ufw allow 'Apache'
```



3.2 Personnalisation

Création d'une page index.html de test dans /var/www/html/ pour valider la prise en compte des modifications.

4. Phase 3 : Système de Gestion de Base de Données (MySQL)

4.1 Installation et Durcissement (Hardening)

Installation avec sudo apt install mysql-server. Pour sécuriser l'instance, nous avons exécuté le script de sécurisation :

Bash

```
sudo mysql_secure_installation
```

- Désactivation des connexions root distantes.
- Suppression de la base de données test.

4.2 Création de la structure de données

Connexion à l'interface en ligne de commande :

Bash

```
sudo mysql -u root
```

Puis création de la base b2tech_admin et d'un utilisateur dédié pour éviter d'utiliser le compte root pour les applications.

Database
Unnamed
employees
information_schema
mysql
owncloud
performance_schema
sys
zabbix

2. Installation et Intégration d'ownCloud

2.1 Déploiement du Cloud Privé

Installation des dépendances PHP nécessaires (ownCloud en nécessite beaucoup pour fonctionner avec MySQL) :

Bash

```
sudo apt install php-mysql php-gd php-json php-curl php-mbstring php-intl php-xml  
php-zip
```

2.2 Configuration MySQL pour ownCloud

Création d'un utilisateur dédié pour plus de sécurité :

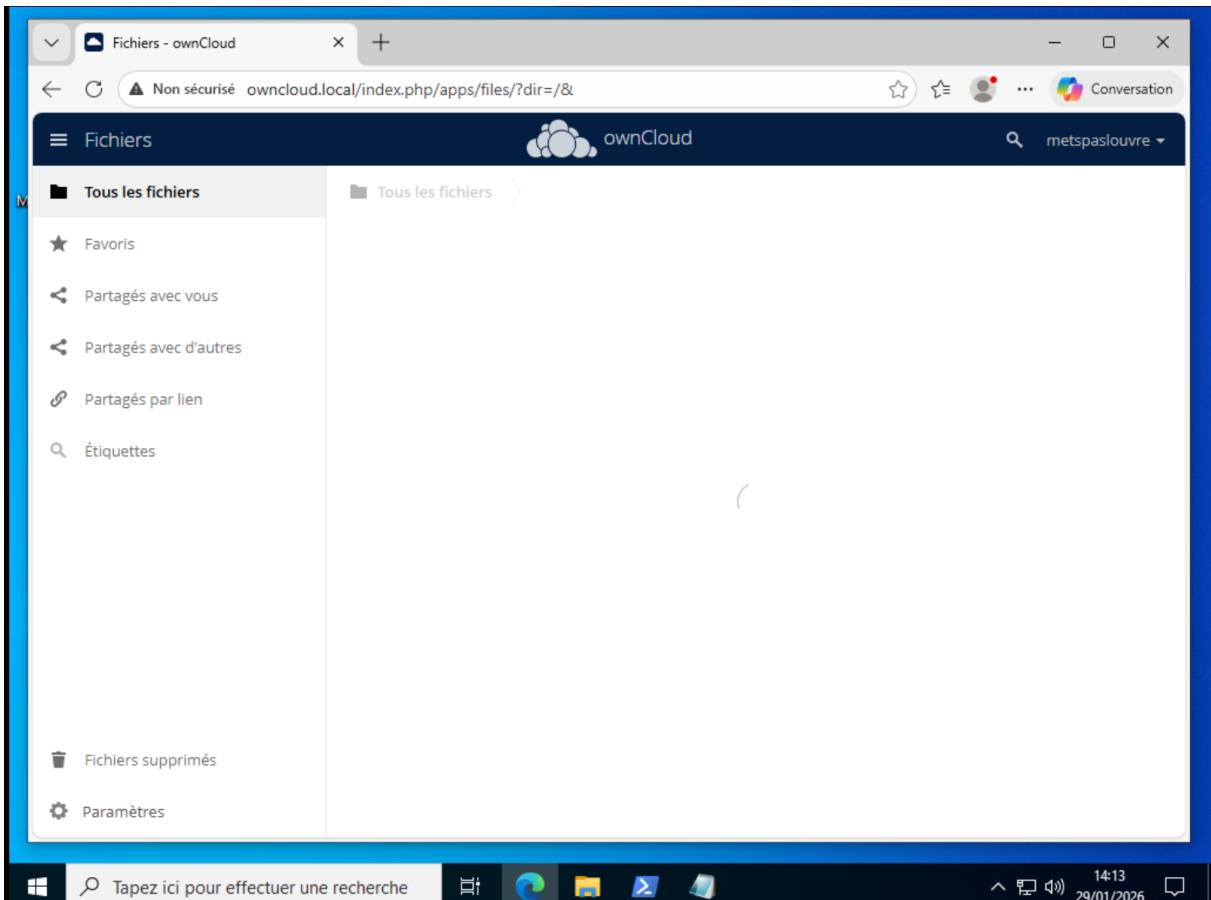
SQL

```
CREATE DATABASE owncloud;
```

```
CREATE USER 'sqlmaster' @'192.168.118.%' IDENTIFIED BY 'Password123!';
```

```
GRANT ALL PRIVILEGES ON owncloud.* TO 'sqlmaster'@'192.168.118.%';
```

FLUSH PRIVILEGES;



3. Services Réseau : DHCP et DNS

3.1 Serveur DHCP (ISC-DHCP-Server) — IP: 192.168.118.11

Configuration du fichier /etc/dhcp/dhcpd.conf :

Bash

```
subnet 192.168.118.0 netmask 255.255.255.0 {  
    option subnet-mask 255.255.255.0;  
    option broadcast-address 192.168.118.255;  
    option routers 192.168.118.254;  
    option domain-name-servers 192.168.118.12, 8.8.8.8;  
    option domain-name "google.com";  
    range 192.168.118.20 192.168.118.100;  
    default-lease-time 600;
```

```
max-lease-time 7200;  
}
```

👉 **PHOTO 2 :** Capture d'écran sur le **Windows Client** faisant un ipconfig /all montrant qu'il a bien reçu une IP dans la plage .100-.150 et qu'il pointe vers le DNS .12.

3.2 Serveur DNS (Bind9) — IP: 192.168.118.12

db.server.owncloud.local :

```
$TTL    604800  
@      IN      SOA     server.jeu.local. admin.server.jeu.local. (  
                      3           ; Serial (Incrémenté)  
                      604800      ; Refresh  
                      86400       ; Retry  
                     2419200     ; Expire  
                     604800 )    ; Negative Cache TTL  
;  
@      IN      NS      server.jeu.local.  
  
; Pointer le domaine racine (server.jeu.local) vers l'IP  
@      IN      A       192.168.118.20  
  
; Pointer l'hôte "server" vers l'IP  
server IN      A       192.168.118.20  
  
; Alias www  
www   IN      CNAME   server.jeu.local.
```

db.inverse.server.owncloud.local :

```
; BIND reverse data file for server.jeu.local  
;  
$TTL    604800  
@      IN      SOA     server.jeu.local. admin.server.jeu.local. (   
                      3           ; Serial (Incrémenté à 3)  
                      604800      ; Refresh  
                      86400       ; Retry  
                     2419200     ; Expire  
                     604800 )    ; Negative Cache TTL  
;  
; Enregistrements Name Server (NS)  
@      IN      NS      server.jeu.local.  
  
; Enregistrement PTR pour l'IP 192.168.118.20  
20    IN      PTR      server.jeu.local.
```

```
root@realdns:/home/anis# nslookup server.jeu.local 192.168.118.12
Server:      192.168.118.12
Address:     192.168.118.12#53

Name:   server.jeu.local
Address: 192.168.118.20
```

4. Sécurisation du Serveur (Hardening)

4.1 Configuration du Pare-feu (UFW)

Nous avons appliqué une politique de "refus par défaut" et autorisé uniquement le nécessaire :

Bash

```
sudo ufw default deny incoming
sudo ufw allow 80/tcp  # Apache
sudo ufw allow 53      # DNS (Bind9)
sudo ufw allow 67/udp  # DHCP
sudo ufw allow 22/tcp  # SSH (pour l'admin)
sudo ufw enable
```

4.2 Audit de sécurité

Utilisation de **Lynis** pour scanner les vulnérabilités :

Bash

```
sudo lynis audit system
```

Difficultés rencontrées & Solutions

- **Conflit de ports** : Vérifier que les machines n'utilisent pas systemd-resolved sur le port 53, ce qui bloquerait Bind9.
- **Permissions ownCloud** : Nécessité d'appliquer un chown -R www-data:www-data sur le dossier data pour permettre l'écriture des fichiers.

2. Automatisation avec Ansible (Serveur .13)

2.1 Pourquoi Ansible ?

Nous avons choisi Ansible face à Puppet et Chef car il est **agentless** (pas besoin d'installer de logiciel sur les cibles, tout passe par SSH), ce qui le rend léger et parfaitement adapté à l'infrastructure d'ADEF Solutions.

2.2 Déploiement par Playbooks

Nous avons conçu des fichiers YAML pour automatiser les tâches répétitives.

- **Playbook de sécurisation** : Création d'utilisateurs admin, déploiement des clés SSH et configuration de UFW.
- **Playbook applicatif** : Installation automatique des paquets LAMP sur un nouveau nœud.

Extrait de code Ansible (Exemple de création d'utilisateur) :

YAML

```
- name: Création de l'utilisateur admin B2Tech
  user:
    name: b2tech_admin
    groups: sudo
    shell: /bin/bash
```

```
root@zabbix:/home/anis/ansible# ansible-playbook deploy.yml
PLAY [Configuration de l'infrastructure Ubuntu] ****
TASK [Gathering Facts] ****
ok: [192.168.118.20]
ok: [192.168.118.12]
ok: [192.168.118.13]
ok: [192.168.118.11]

TASK [Créer l'utilisateur] ****
changed: [192.168.118.11]
changed: [192.168.118.13]
changed: [192.168.118.20]
changed: [192.168.118.12]
```

3. Supervision avec Zabbix (Serveur .13)

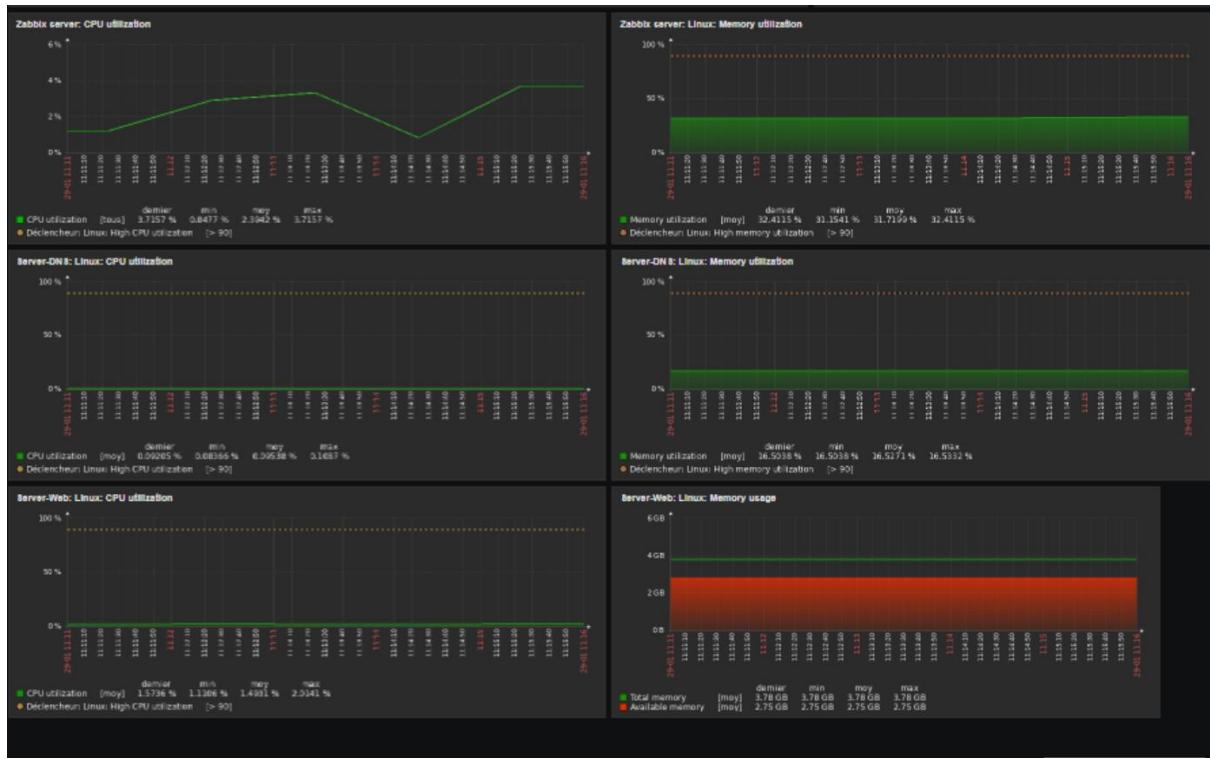
3.1 Monitoring en Temps Réel

Zabbix permet de surveiller la santé de nos serveurs .11, .12 et .20.

- **Indicateurs clés (Triggers)** : Charge CPU > 80%, Espace disque < 10%, Service Apache ou MySQL arrêté.

3.2 Simulation d'Incident

Nous avons coupé manuellement le service Apache (sudo systemctl stop apache2) pour vérifier la remontée d'alerte sur le tableau de bord Zabbix.



4. Maintenance & Sauvegardes

4.1 Plan de Maintenance

- Quotidien** : Vérification des logs système et des alertes Zabbix.
- Hebdomadaire** : Mise à jour des paquets (apt update) via Ansible.

4.2 Stratégie de Backup

Mise en place d'une tâche **Cron** pour sauvegarder la base de données MySQL et les fichiers de configuration DNS/DHCP.

Bash

```
# Sauvegarde MySQL
```

```
mysqldump -u owncloud_user -p owncloud_db > /backup/db_$(date +%F).sql
```

5. Bilan et Validation des Contraintes

- Disponibilité 24/7** : Validée par la supervision Zabbix.
- Sécurité** : Testée via un scan de vulnérabilité (Lynis/Nmap) montrant un score de durcissement élevé.
- Reproductibilité** : Grâce à Ansible, un nouveau serveur peut être configuré en moins de 5 minutes.

