■ Solution d'hébergement Intranet — Tranche 1

Procédure de déploiement · Projet fil rouge BTS SIO SISR · Christophe Beaudenon · E2SE Digital IT Caen

Contexte et objectifs

Dans le cadre du projet fil rouge BTS SIO option SISR, la Tranche 1 consiste à déployer le cœur fonctionnel d'une solution d'hébergement d'applications web en intranet. L'environnement de développement repose sur VirtualBox installé sur la machine hôte Windows.

À l'issue de cette procédure, les services suivants sont opérationnels :

- Pile LAMP : Apache2, PHP 8.2, MariaDB
- Sécurisation : pare-feu UFW, fail2ban, durcissement MariaDB, SSH
- Service DNS : Bind9 résolvant la zone intranet.local
- Portail WordPress accessible sur http://intranet.local
- Service cloud NextCloud accessible sur http://cloud.intranet.local

Plan d'adressage

Machine	Interface	Adresse IP	Rôle
VM Hébergement	enp0s3 — NAT	DHCP (10.0.2.15)	Accès Internet (apt, téléchargements)
VM Hébergement	enp0s8 — Host-Only	192.168.56.200/24	LAN intranet — web, DNS, SSH
Machine hôte	VirtualBox Host-Only	192.168.56.1/24	Client — navigateur, DNS, SSH

Étape 1 — Installation de Debian 13 et configuration initiale

■ **Pourquoi ?** Debian 13 en installation minimale (sans interface graphique) garantit un système léger et sécurisé, avec uniquement les composants nécessaires. Moins de services = moins de surface d'attaque.

Installation Debian 13 Trixie

- Télécharger l'ISO netinst depuis debian.org
- Créer la VM VirtualBox : 2 Go RAM, 20 Go disque, Adaptateur 1 = Host-Only, Adaptateur 2 = NAT
- Installation minimale : aucun environnement graphique, utilitaires standard uniquement
- Partitionnement guidé simple — partition unique ext4

Configuration réseau (interface Host-Only)

■ **Pourquoi ?** L'interface Host-Only (enp0s8) nécessite une IP statique fixe — c'est l'adresse que la machine hôte utilisera pour accéder aux services web et DNS. L'interface NAT (enp0s3) reste en DHCP pour l'accès Internet.

Créer le fichier /etc/network/interfaces.d/enp0s8 :

```
auto enp0s8
iface enp0s8 inet static
address 192.168.56.200/24
# pas de gateway sur cette interface
```

```
systemctl restart networking
ip addr show enp0s8 # vérification : 192.168.56.200/24
```

Utilitaires de base et SSH

```
apt update && apt upgrade -y
apt install -y sudo htop curl wget git nano openssh-server net-tools unzip
```

Durcissement SSH dans /etc/ssh/sshd_config :

```
PermitRootLogin no

systemctl reload sshd
```

Étape 2 — Déploiement de la pile LAMP

> ■ **Pourquoi ?** La pile LAMP (Linux, Apache2, MariaDB, PHP) est la combinaison standard pour héberger des applications web PHP comme WordPress et NextCloud. Chaque composant a un rôle précis : Apache sert les pages, PHP les exécute, MariaDB stocke les données.

Apache2

```
apt install -y apache2
systemctl enable apache2 && systemctl start apache2
a2enmod rewrite headers ssl
systemctl restart apache2
```

PHP et extensions

> ■ **Pourquoi ?** WordPress et NextCloud nécessitent des extensions PHP spécifiques. Il vaut mieux les installer toutes dès maintenant plutôt que de déboguer des erreurs applicatives plus tard.

```
apt install -y php libapache2-mod-php
apt install -y php-json php-mysqli php-curl php-dom php-exif php-fileinfo \
php-igbinary php-imagick php-intl php-mbstring php-xml php-zip php-opcache

php -v # vérification version installée
```

MariaDB

```
apt install -y mariadb-server mariadb-client
systemctl enable mariadb && systemctl start mariadb

# Sécurisation initiale (répondre Oui à toutes les questions)
mariadb-secure-installation
```

Création des bases de données applicatives :

```
mysql -u root -p

-- Base WordPress
CREATE DATABASE IF NOT EXISTS wp_intranet_local CHARACTER SET
utf8 COLLATE utf8mb3_general_ci;
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'MotDePasseWP_Fort!';
GRANT ALL PRIVILEGES ON wp_intranet_local.* TO 'wp_user'@'localhost';

-- Base NextCloud
CREATE DATABASE IF NOT EXISTS nextcloud_db
CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'nc_user'@'localhost' IDENTIFIED BY 'MotDePasseNC_Fort!';
GRANT ALL PRIVILEGES ON nextcloud_db.* TO 'nc_user'@'localhost';

FLUSH PRIVILEGES;
EXIT;
```

Virtual Hosts Apache

■ **Pourquoi ?** Deux Virtual Hosts permettent à Apache de servir deux applications différentes sur le même serveur selon le nom de domaine demandé : intranet.local → WordPress, cloud.intranet.local → NextCloud.

Fichier /etc/apache2/sites-available/intranet.local.conf :

```
<VirtualHost *:80>
ServerName intranet.local
DocumentRoot /var/www/intranet.local
DirectoryIndex index.php index.html
<Directory /var/www/intranet.local>
AllowOverride All
Require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/intranet_error.log
CustomLog ${APACHE_LOG_DIR}/intranet_access.log combined
</VirtualHost>
```

Fichier /etc/apache2/sites-available/cloud.intranet.local.conf :

```
<VirtualHost *:80>
ServerName cloud.intranet.local
DocumentRoot /var/www/cloud.intranet.local
<Directory /var/www/cloud.intranet.local>
AllowOverride All
Require all granted
Options FollowSymLinks MultiViews
</Directory>
ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log
CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log combined
</VirtualHost>
```

```
a2ensite intranet.local.conf cloud.intranet.local.conf
a2dissite 000-default.conf
systemctl reload apache2
```

Étape 3 — Sécurisation de la machine

Pare-feu UFW

■ **Pourquoi ?** UFW (Uncomplicated Firewall) applique une politique de refus par défaut en entrée. Seuls les ports strictement nécessaires sont ouverts : SSH pour l'administration, HTTP pour les applications web, DNS pour le service Bind9.

```
apt install -y ufw
ufw default deny incoming
ufw default allow outgoing
ufw allow ssh # port 2222
ufw allow http # port 80
ufw allow https # port 443
ufw allow dns # port 53 TCP+UDP
ufw enable
ufw status verbose
```

■■ 'ufw allow dns' ouvre automatiquement le port 53 en TCP et UDP. Bind9 utilise UDP pour les requêtes standard et TCP pour les transferts de zone et les réponses dépassant 512 octets.

Fail2ban

■ **Pourquoi ?** Fail2ban analyse les logs et bannit automatiquement les IPs qui multiplient les tentatives d'authentification échouées. C'est la protection de base contre les attaques brute-force SSH et les scans Apache.

Créer /etc/fail2ban/jail.local (ne jamais modifier jail.conf directement) :

```
apt install -y fail2ban

# Contenu de /etc/fail2ban/jail.local
[DEFAULT]
bantime = 3600
findtime = 600
maxretry = 5

[sshd]
enabled = true

[apache-auth]
enabled = true

[apache-noscript]
enabled = true

systemctl enable fail2ban && systemctl restart fail2ban
fail2ban-client status
```

■■ Si tu travailles depuis 192.168.56.1 (machine hôte), ajouter ignoreip = 192.168.56.1 dans [DEFAULT] pour éviter de te faire bannir pendant les tests.

Étape 4 — Service DNS Bind9 — zone intranet.local

■ **Pourquoi ?** Bind9 est configuré comme serveur DNS autoritaire pour la zone intranet.local. Il résout les noms intranet.local et cloud.intranet.local vers 192.168.56.200, et relaie les autres requêtes vers des DNS publics (Cloudflare, Google). Sans DNS, il faudrait modifier le fichier hosts sur chaque machine cliente.

Installation

```
apt install -y bind9 bind9utils bind9-doc bind9-dnsutils bind9-host
systemctl enable bind9 && systemctl start bind9
```

Configuration /etc/bind/named.conf.options

```
acl intranet {
192.168.56.0/24;
127.0.0.0/8;
};

options {
directory "/var/cache/bind";
recursion yes;
allow-recursion { intranet; };
allow-query { intranet; };
listen-on { any; };
forwarders {
1.1.1.1; 1.0.0.1;
8.8.8.8; 8.8.4.4;
};
dnssec-validation auto;
};
```

Déclaration de zone /etc/bind/named.conf.local

```
zone "intranet.local" IN {
type master;
file "/etc/bind/db.intranet.local";
};
```

Fichier de zone /etc/bind/db.intranet.local

```
$TTL 86400
@ IN SOA ns.intranet.local. root.intranet.local.
( 2026051501 ; Serial
12h ; Refresh
15m ; Retry
3w ; Expire
2h ) ; Negative Cache TTL

@ IN NS ns.intranet.local.
@ IN A 192.168.56.200
ns IN A 192.168.56.200
root IN A 192.168.56.200
cloud IN CNAME root.intranet.local.
```

■■ cloud.intranet.local est défini en CNAME pointant vers root.intranet.local (qui porte l'IP 192.168.56.200). Les entrées APEX (@) et NS ne peuvent pas utiliser de CNAME.

Vérification et rechargement

```
named-checkconf
named-checkzone intranet.local /etc/bind/db.intranet.local
systemctl reload bind9
```

DNS local — résolveur de la VM elle-même

■ **Pourquoi ?** La VM doit utiliser son propre Bind9 comme résolveur pour que WordPress et NextCloud fonctionnent avec les noms intranet.local. En interne, on utilise 127.0.0.1 (loopback) et non 192.168.56.200.

Créer /etc/resolv.conf.head :

```
nameserver 127.0.0.1
```

Configurer la machine hôte Windows :

- Propriétés TCP/IPv4 de l'adaptateur VirtualBox Host-Only Ethernet Adapter
- Serveur DNS préféré : 192.168.56.200

Validation depuis la machine hôte :

```
nslookup intranet.local 192.168.56.200 # → 192.168.56.200
nslookup cloud.intranet.local 192.168.56.200 # → 192.168.56.200
```

Étape 5 — Déploiement de WordPress

■ **Pourquoi ?** WordPress est téléchargé depuis wordpress.org (source officielle, toujours à jour). Il est déposé dans le répertoire défini dans le VirtualHost Apache. Les droits www-data permettent à Apache d'accéder aux fichiers.

```
cd /tmp
wget https://wordpress.org/latest.zip
unzip latest.zip
mkdir -p /var/www/intranet.local
cp -R wordpress/* /var/www/intranet.local/
chown -R www-data:www-data /var/www/intranet.local
```

Configuration wp-config.php

```
cp /var/www/intranet.local/wp-config-sample.php /var/www/intranet.local/wp-config.php
nano /var/www/intranet.local/wp-config.php
```

Paramètres essentiels à renseigner :

```
define( 'DB_NAME', 'wordpress_intranet_local' );
define( 'DB_USER', 'wp_user' );
define( 'DB_PASSWORD', 'MotDePasseWP_Fort!' );
define( 'DB_HOST', 'localhost' );
define( 'DB_CHARSET', 'utf8mb4' );
```

■■ Générer les clés de sécurité WordPress depuis https://api.wordpress.org/secret-key/1.1/salt/ et les coller dans wp-config.php. Bonne pratique : changer le préfixe des tables (ex: 'intra_' au lieu de 'wp_').

```
a2enmod rewrite
systemctl restart apache2
```

Finaliser via le navigateur : http://intranet.local → assistant d'installation WordPress.

Étape 6 — Déploiement de NextCloud

■ **Pourquoi ?** NextCloud est une solution open-source de cloud d'entreprise. Le répertoire de données est placé hors de la racine web (/var/data/nextcloud) pour éviter qu'il soit accessible via le navigateur — c'est une exigence de sécurité de NextCloud.

```
cd /tmp
wget https://download.nextcloud.com/server/releases/latest.tar.bz2
tar -xjf latest.tar.bz2
mkdir -p /var/www/cloud.intranet.local
cp -R nextcloud/* /var/www/cloud.intranet.local/
chown -R www-data:www-data /var/www/cloud.intranet.local

# Répertoire de données hors racine web
mkdir -p /var/data/nextcloud
chown -R www-data:www-data /var/data/nextcloud
chmod 750 /var/data/nextcloud
```

Configuration PHP pour NextCloud

Créer /etc/php/8.x/apache2/conf.d/99-nextcloud.ini :

```
memory_limit = 512M
upload_max_filesize = 1024M
post_max_size = 1024M
max_execution_time = 300
date.timezone = Europe/Paris
opcache.enable = 1
opcache.memory_consumption = 128

systemctl restart apache2
```

Finaliser via le navigateur : http://cloud.intranet.local → assistant NextCloud.

Renseigner : compte admin, répertoire de données /var/data/nextcloud, base nextcloud_db / nc_user / localhost.

Tableau de validation

Test	Commande / Méthode	Résultat attendu	Statut
Résolution DNS intranet.local	`nslookup intranet.local 192.168.56.200`	192.168.56.200	✓ OK
Résolution DNS cloud.intranet.local	`nslookup cloud.intranet.local 192.168.56.200`	192.168.56.200	✓ OK

DNS depuis la VM (loopback)	`dig intranet.local @127.0.0.1`	192.168.56.200	✓ **OK**
Accès WordPress	`http://intranet.local` depuis navigateur hôte	Page d'accueil WordPress	✓ **OK**
Accès NextCloud	`http://cloud.intranet.local` depuis navigateur hôte	Page de connexion NextCloud	✓ **OK**
SSH fonctionnel	`ssh user@192.168.56.200`	Invite Debian	✓ **OK**
UFW actif	`ufw status verbose`	22, 53, 80, 443 ouverts	✓ **OK**
Fail2ban actif	`fail2ban-client status`	sshd, apache-auth actifs	✓ **OK**
MariaDB — accès local	`mysql -u root -p`	Connexion réussie	✓ **OK**
MariaDB — accès root distant refusé	`mysql -u root -h 192.168.56.200`	Accès refusé	✓ **OK**
Apache Virtual Hosts	`apachectl -S`	2 VirtualHosts configurés	✓ **OK**