DOCUMENTATION TECHNIQUE : DÉPLOIEMENT DE NETBOX (IPAM / DCIM)

1. PRÉSENTATION ET OBJECTIFS

Netbox est une application open source de modélisation d'infrastructure (OT et IT).
Elle sert de "Source de Vérité" (Source of Truth) pour l'ensemble du réseau.
Elle combine deux fonctionnalités majeures :
 * IPAM (IP Address Management) : Gestion centralisée des adresses IP, préfixes, VRF et VLANs.
 * DCIM (Data Center Infrastructure Management) : Gestion des sites, des racks, des équipements et du câblage physique.
Objectif : Centraliser la documentation technique du réseau et faciliter l'automatisation.

2. PRÉREQUIS ET ARCHITECTURE

2.1. Environnement technique

- Système d'exploitation : Debian 13.
- 4 cœur / 4 go de ram / 50 go de stockage
- Accès : Privilèges root ou sudo.

2.2. Architecture logicielle

Netbox est une application Web basée sur le framework Django (Python). Elle s'appuie sur la pile technique suivante :
 * PostgreSQL : Base de données relationnelle (stockage des objets).
 * Redis : Gestion du cache et des files d'attente (Webhooks).
 * Gunicorn : Serveur d'application WSGI HTTP.
 * Nginx : Serveur Web utilisé comme Reverse Proxy pour gérer le HTTPS.

3. INSTALLATION DES COMPOSANTS DE BASE

3.1. Base de données PostgreSQL

Netbox nécessite une base de données dédiée. Après l'installation du paquet postgresql, il est nécessaire de créer une base et un utilisateur avec les droits appropriés.
Commandes de configuration de la base :
CREATE DATABASE netbox;
CREATE USER netbox WITH PASSWORD ‘netboxkillian';
GRANT ALL PRIVILEGES ON DATABASE netbox TO netbox;

```
import os
import ldap
from django_auth_ldap.config import LDAPSearch, NestedGroupOfNamesType

# ------------------------------
# Base & sécurité
# ------------------------------
SECRET_KEY = 'DjjK85t$pnpq-sOT+U7%5^Ga9d0mTv0IcnYgD=p&y2R50v-Jz_'
DEBUG = False
ALLOWED_HOSTS = [
    'netbox.chu.fr',
    'lmc03m.chu.fr',
    '127.0.0.1',
    'localhost'
]
TIME_ZONE = 'UTC'
USE_X_FORWARDED_HOST = True
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO', 'https')

# ------------------------------
# PostgreSQL
# ------------------------------
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'netbox',
        'USER': 'netbox',
        'PASSWORD': 'netboxkillian',
        'HOST': 'localhost',
        'PORT': '',
        'CONN_MAX_AGE': 300,
    }
}
```

4. INSTALLATION ET CONFIGURATION DE NETBOX

4.1. Récupération des sources

L'application est installée dans le répertoire /opt/netbox via le dépôt Git officiel.
Un utilisateur système netbox est créé pour exécuter le service sans droits root.

4.2. Configuration principale (configuration.py)

Le fichier /opt/netbox/netbox/netbox/configuration.py contient les paramètres vitaux. Il faut y renseigner les hôtes autorisés et les identifiants de base de données.
Exemple de configuration :
# Autoriser l'IP et le nom de domaine
ALLOWED_HOSTS = ['lmc-netbox.chu.fr', '192.168.159.19']

# Connexion Base de Données
DATABASE = {
'NAME': 'netbox',
'USER': 'netbox',
'PASSWORD': ’netboxkillian’,
'HOST': 'localhost',
'PORT': '',
}

```
import os
import ldap
from django_auth_ldap.config import LDAPSearch, NestedGroupOfNamesType

# ------------------------------
# Base & sécurité
# ------------------------------
SECRET_KEY = 'DjjK85t$pnpq-sOT+U7%5^Ga9d0mTv0IcnYgD=p&y2R50v-Jz_'
DEBUG = False
ALLOWED_HOSTS = [
    'netbox.chu.fr',
    'lmc03m.chu.fr',
    '127.0.0.1',
    'localhost'
]
TIME_ZONE = 'UTC'
USE_X_FORWARDED_HOST = True
SECURE_PROXY_SSL_HEADER = ('HTTP_X_FORWARDED_PROTO', 'https')

# ------------------------------
# PostgreSQL
# ------------------------------
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'netbox',
        'USER': 'netbox',
        'PASSWORD': 'netboxkillian',
        'HOST': 'localhost',
        'PORT': '',
        'CONN_MAX_AGE': 300,
    }
}
```

4.3. Script d'installation

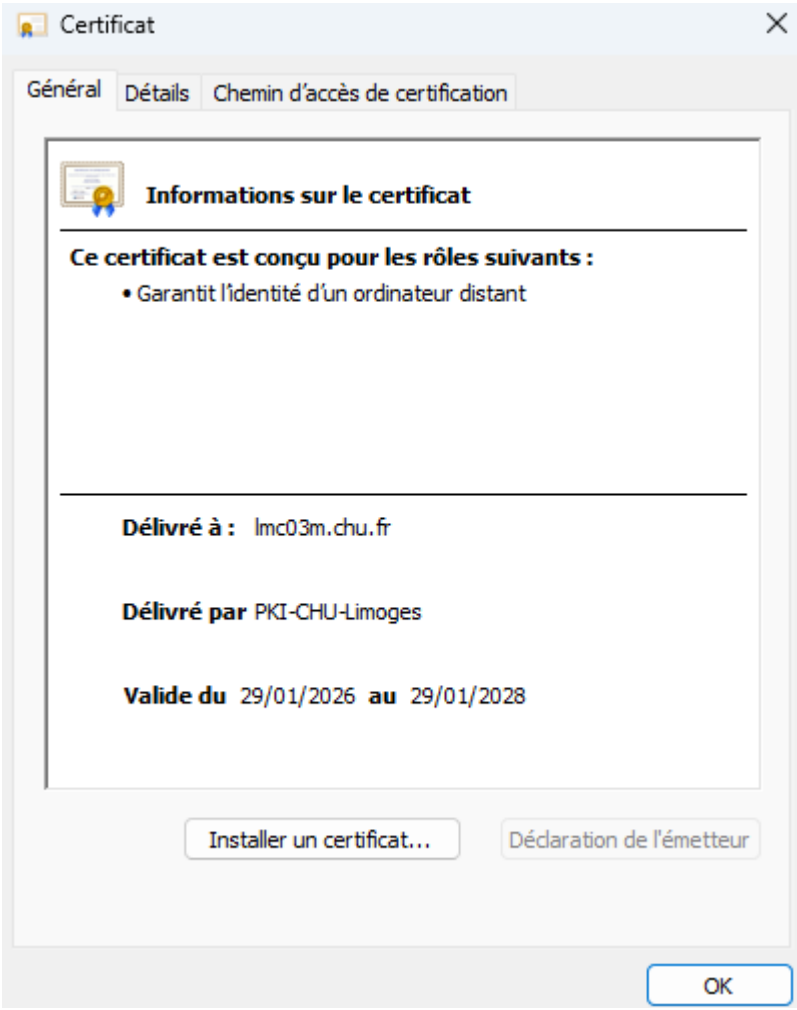
Le script upgrade.sh fourni par Netbox permet d'installer les dépendances Python, d'effectuer les migrations de base de données et de collecter les fichiers statiques.

5. SÉCURISATION HTTPS (REVERSE PROXY NGINX)

Comme pour la supervision Grafana, l'accès direct au serveur d'application (Gunicorn, port 8001) n'est pas sécurisé. Nginx est utilisé en frontal pour chiffrer les communications (HTTPS) sur le port 443.

5.1. Gestion des Certificats SSL

La procédure est identique à celle de Grafana : génération d'une clé privée (.key) et d'une demande de signature (.csr) pour obtenir un certificat signé par la PKI de l'organisation.
Fichiers utilisés :
 * Clé : /etc/ssl/netbox/netbox.key
 * Certificat : /etc/ssl/netbox/netbox_2026.cer



5.2. Configuration du VirtualHost Nginx

Fichier : /etc/nginx/sites-available/netbox

```
# HTTP → HTTPS
server {
    listen 80;
    server_name netbox.entreprise.local;
    return 301 https://$host$request_uri;
}

# HTTPS
server {
    listen 443 ssl;
    server_name netbox.chu.fr;

    ssl_certificate     /etc/ssl/netbox/netbox_2026.cer;
    ssl_certificate_key /etc/ssl/netbox/netbox.key;

    client_max_body_size 25m;

    location /static/ {
        alias /opt/netbox/netbox/static/;
    }

    location / {
        proxy_pass http://127.0.0.1:8001;

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
    }
}
```

Une fois la configuration appliquée, on redémarre Nginx :

sudo systemctl restart nginx

6. AUTHENTIFICATION CENTRALISÉE (LDAP)

Pour éviter la gestion de comptes locaux, Netbox est connecté à l'Active Directory via le protocole LDAPS (Port 636).

6.1. Configuration LDAP

Le fichier ldap_config.py définit les paramètres de connexion et le mapping des groupes.

* Serveur : ldaps.chu.fr
* Bind DN : CN=svc_netbox,OU=Comptes de service,DC=chu,DC=fr
* Filtre de recherche : Permet de trouver les utilisateurs via leur sAMAccountName.

```
# LDAP / Active Directory
# ------------------------------
AUTH_LDAP_SERVER_URI = "ldaps://ldaps.chu.fr:636"
AUTH_LDAP_CONNECTION_OPTIONS = {ldap.OPT_REFERRALS: 0}
AUTH_LDAP_BIND_DN = "CN=svc_netbox,OU=Comptes de service,DC=chu,DC=fr"
AUTH_LDAP_BIND_PASSWORD = [redacted]
LDAP_IGNORE_CERT_ERRORS = False
LDAP_CA_CERT_DIR = '/etc/ssl/certs'
LDAP_CA_CERT_FILE = '/path/to/example-CA.crt'

AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "ou=User,dc=chu,dc=fr",
    ldap.SCOPE_SUBTREE,
    "(|(userPrincipalName=%(user)s)(sAMAccountName=%(user)s))"
)
AUTH_LDAP_USER_DN_TEMPLATE = None
AUTH_LDAP_USER_ATTR_MAP = {
    "username": "sAMAccountName",
    "email": "mail",
    "first_name": "givenName",
    "last_name": "sn",
}
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
    "dc=chu,dc=fr",
    ldap.SCOPE_SUBTREE,
    "(objectClass=group)"
)
AUTH_LDAP_GROUP_TYPE = NestedGroupOfNamesType()
AUTH_LDAP_REQUIRE_GROUP = "CN=Utilisateurs_Netbox,OU=Groupes Ressources,DC=chu,DC=fr"
AUTH_LDAP_MIRROR_GROUPS = True
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    "is_active": "CN=Utilisateurs_Netbox,OU=Groupes Ressources,DC=chu,DC=fr",
    "is_superuser": "CN=Utilisateurs_Netbox,OU=Groupes Ressources,DC=chu,DC=fr"
}
AUTH_LDAP_FIND_GROUP_PERMS = True
AUTH_LDAP_CACHE_TIMEOUT = 3600
AUTH_LDAP_ALWAYS_UPDATE_USER = True

# ------------------------------
# Logging LDAP / NetBox
# ------------------------------
LOG_DIR = os.path.join(os.path.dirname(__file__), "logs")
os.makedirs(LOG_DIR, exist_ok=True)

LOGGING = {
    "version": 1,
    "disable_existing_loggers": False,
    "formatters": {
        "verbose": {
            "format": "[%(asctime)s] %(levelname)s [%(name)s:%(lineno)s] %(message)s",
            "datefmt": "%Y-%m-%d %H:%M:%S",
        },
        "simple": {"format": "%(levelname)s %(message)s"},
    },
    "handlers": {
        "console": {"class": "logging.StreamHandler", "formatter": "verbose"},
        "netbox_auth_log": {
            "class": "logging.FileHandler",
            "filename": os.path.join(LOG_DIR, "ldap.log"),
            "formatter": "verbose",
            "level": "DEBUG",
        },
        "netbox_file": {
            "class": "logging.FileHandler",
            "filename": os.path.join(LOG_DIR, "netbox.log"),
            "formatter": "verbose",
            "level": "INFO",
        },
    },
    "loggers": {
        "django": {"handlers": ["console", "netbox_file"], "level": "INFO", "propagate": True},
        "django_auth_ldap": {"handlers": ["console", "netbox_auth_log"], "level": "DEBUG", "propagate": False},
        "netbox": {"handlers": ["console", "netbox_file"], "level": "INFO", "propagate": True},
    },
}
```

6.2. Mapping des permissions

Les groupes AD sont mappés vers des groupes de permissions Netbox.

* Le groupe AD CN=Utilisateurs_Netbox donne les droits d'édition ou d'administration selon la configuration.

7. EXPORT DES MÉTRIQUES (SUPERVISION)

Afin de surveiller l'état de santé du serveur Netbox dans Grafana, un agent Node Exporter est installé sur le serveur.

* Rôle : Expose les métriques système (CPU, RAM, Disque).

* Port d'écoute : 9100.
* Intégration : L'adresse lmc03m est ajoutée aux cibles de Prometheus.