

Documentation Complète - Projet ClassCord

Date de création : 2025-06-16 15:09:44

Auteur : delcoco95

Version : 1.0

État : Production

Table des matières

1. [Configuration de l'environnement virtuel](#)
2. [Installation du système](#)
3. [Configuration réseau et sécurité](#)
4. [Transfert et préparation des fichiers](#)
5. [Résolution des problèmes](#)
6. [Utilisation du serveur](#)
7. [Maintenance et dépannage](#)
8. [Sécurité](#)

1. Configuration de l'environnement virtuel

1.1 Prérequis

- Oracle VM VirtualBox (dernière version stable)
- Image ISO Debian 11 (debian-11.x.x-amd64-netinst.iso)
- Au moins 20 Go d'espace disque disponible
- Minimum 4 Go de RAM sur l'hôte
- Accès administrateur sur le système hôte

1.2 Création de la VM

1. **Lancer VirtualBox** et cliquer sur "Nouvelle"
2. **Configuration de base :**

```
Nom : ClassCord-Server  
Type : Linux  
Version : Debian (64-bit)  
Mémoire : 2048 MB
```

3. **Configuration du disque :**

```
Type : VDI (VirtualBox Disk Image)  
Allocation : Dynamiquement alloué  
Taille : 20 Go
```

1.3 Paramètres avancés de la VM

1. **Système :**

Processeur : 2 CPU
Ordre d'amorçage :
- Disque dur
- DVD
PAE/NX : Activé
Horloge UTC : Activé

2. **Affichage :**

Mémoire vidéo : 128 MB
Contrôleur : VBoxVGA
Accélération 3D : Désactivé

3. **Stockage :**

- Contrôleur IDE : ISO Debian
- Contrôleur SATA : Disque VDI

4. **Audio :**

- Désactivé (serveur sans interface graphique)

2. Installation du système

2.1 Installation initiale

1. **Démarrage et configuration de base :**

Language : French
Pays : France
Clavier : French
Nom de machine : classcord-server
Domaine : [laisser vide]

2. **Partitionnement :**

Méthode : Assisté - utiliser un disque entier
Disque : SCSI1 (0,0,0) (sda) - 20.0 GB VMware Virtual disk
Schéma : Tout dans une partition

3. **Configuration utilisateur :**

```
Root password : [mot_de_passe_sécurisé]
Compte utilisateur : classcord
Mot de passe : [mot_de_passe_sécurisé]
```

4. Sélection des logiciels :

- ☒ serveur SSH
- ☒ utilitaires usuels du système
- ☐ environnement de bureau
- ☐ serveur web
- ☒ utilitaires système standard

2.2 Configuration post-installation

1. Première connexion et mise à jour :

```
su -
apt update
apt upgrade -y
```

2. Installation des paquets essentiels :

```
apt install -y python3 python3-pip python3-venv
apt install -y git curl wget
apt install -y net-tools htop
apt install -y ufw
```

3. Configuration de l'environnement Python :

```
python3 -m pip install --upgrade pip
python3 -m pip install virtualenv
```

3. Configuration réseau et sécurité

3.1 Configuration réseau VirtualBox

1. Configuration de la carte réseau :

- Mode d'accès réseau : NAT
- Type de carte : Intel PRO/1000 MT Desktop

2. Redirection de ports :

Nom	Protocole	IP Hôte	Port Hôte	IP Invité	Port Invité
SSH	TCP	127.0.0.1	2222	10.0.2.15	22
Server	TCP	127.0.0.1	12345	10.0.2.15	12345

3.2 Configuration SSH

1. Édition du fichier de configuration :

```
nano /etc/ssh/sshd_config
```

Modifications :

```
PermitRootLogin no  
PasswordAuthentication yes  
MaxAuthTries 3
```

2. Redémarrage du service :

```
systemctl restart ssh
```

3.3 Configuration des pare-feu

3.3.1 Windows Defender

1. Ouvrir les paramètres avancés :

```
Windows + R  
wf.msc
```

2. Règles entrantes :

```
Protocole : TCP  
Ports locaux : 12345  
Action : Autoriser  
Profil : Tous  
Nom : ClassCord_Server_In
```

3. Règles sortantes :

```
Protocole : TCP
Ports distants : 12345
Action : Autoriser
Profil : Tous
Nom : ClassCord_Server_Out
```

3.3.2 UFW (Debian)

1. Configuration de base :

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

2. Ajout des règles :

```
sudo ufw allow 2222/tcp comment 'SSH Access'
sudo ufw allow 12345/tcp comment 'ClassCord Server'
```

3. Activation :

```
sudo ufw enable
sudo ufw status verbose
```

4. Transfert et préparation des fichiers

4.1 Installation et configuration de WinSCP

1. Téléchargement et installation :

- Télécharger WinSCP depuis le site officiel
- Installer avec les options par défaut

2. Configuration de la connexion :

```
Protocole : SFTP
Hôte : localhost
Port : 2222
Utilisateur : classcord
Mot de passe : [votre_mot_de_passe]
```

3. Enregistrement de la session :

- Nom : ClassCord-Server

- Cocher "Sauvegarder le mot de passe"

4.2 Structure des fichiers du projet

1. Création de l'arborescence :

```
mkdir -p ~/classcord-server  
cd ~/classcord-server  
mkdir logs  
mkdir data
```

2. Fichiers à transférer :

```
server_classcord.py    # Serveur principal  
requirements.txt       # Dépendances Python  
config.py             # Configuration  
test_client.py        # Client de test
```

3. Permissions des fichiers :

```
chmod 750 ~/classcord-server  
chmod 640 *.py  
chmod 660 data/  
chmod 750 server_classcord.py
```

4.3 Installation des dépendances

1. Création de l'environnement virtuel :

```
cd ~/classcord-server  
python3 -m venv venv  
source venv/bin/activate
```

2. Installation des packages :

```
pip install -r requirements.txt
```

5. Résolution des problèmes initiaux

5.1 Erreur "Address already in use"

1. Symptôme :

```
OSError: [Errno 98] Address already in use
```

2. Solution appliquée :

```
def main():  
    load_users()  
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
    server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1) #  
    Ajout  
    server_socket.bind((HOST, PORT))  
    server_socket.listen()
```

3. Vérification des ports utilisés :

```
netstat -tuln | grep 12345
```

5.2 Autres problèmes résolus

1. Permissions des fichiers :

```
# Vérification des droits  
ls -la  
# Correction si nécessaire  
chmod -R g+rw data/
```

2. Logs d'erreurs :

```
tail -f logs/server.log
```

6. Utilisation du serveur

6.1 Démarrage du serveur

1. Activation de l'environnement virtuel :

```
cd ~/classcord-server  
source venv/bin/activate
```

2. Lancement du serveur :

```
python3 server_classcord.py
```

3. Vérification du démarrage :

```
[INIT] Utilisateurs chargés: []  
[DEMARRAGE] Serveur en écoute sur 0.0.0.0:12345
```

6.2 Tests de fonctionnement

1. Test avec le client de test :

```
python3 test_client.py
```

2. Logs attendus :

```
[CONNEXION] Nouvelle connexion depuis ('127.0.0.1', 55776)  
[RECU] ('127.0.0.1', 55776) >> {"type": "register", "username": "testuser"}  
[SAVE] Utilisateurs sauvegardés
```

7. Maintenance et dépannage

7.1 Surveillance du serveur

1. Monitoring des ressources :

```
htop
```

2. Surveillance réseau :

```
# Connexions actives  
netstat -nat | grep 12345  
  
# Trafic réseau  
tcpdump -i any port 12345
```

7.2 Sauvegarde et restauration

1. Sauvegarde des données :


```
# Sauvegarde manuelle  
cp -r data/ backup_$(date +%Y%m%d)/
```

2. Restauration :

```
# Depuis une sauvegarde  
cp -r backup_20250616/* data/
```

7.3 Scripts utiles

1. Redémarrage du serveur :

```
#!/bin/bash  
cd ~/classcord-server  
source venv/bin/activate  
pkill -f server_classcord.py  
sleep 2  
python3 server_classcord.py &
```

8. Sécurité

8.1 Bonnes pratiques

1. Mise à jour régulière :

```
sudo apt update  
sudo apt upgrade
```

2. Vérification des logs :

```
# Logs système  
sudo tail -f /var/log/syslog  
  
# Logs d'application  
tail -f logs/server.log
```

3. Surveillance des connexions :

```
# Connexions SSH  
sudo tail -f /var/log/auth.log
```

```
# Tentatives bloquées par UFW  
sudo tail -f /var/log/ufw.log
```

8.2 Plan de reprise

1. En cas de crash :

- Vérifier les logs
- Redémarrer le service
- Vérifier l'intégrité des données

2. En cas d'intrusion :

- Déconnecter le serveur du réseau
- Analyser les logs
- Restaurer depuis une sauvegarde propre

Contact et Support

Support technique

- **Développeur principal** : delcoco95
- **Repository** : AstrowareConception/classcord-server
- **Documentation mise à jour le** : 2025-06-16 15:12:08

Liens utiles

- Documentation Python : python.org/doc
- Guide VirtualBox : virtualbox.org/manual
- Documentation Debian : debian.org/doc

Note : Cette documentation est maintenue régulièrement. Pour toute suggestion ou correction, contacter l'équipe de développement.