

Ajout d'un service dans systemd

Mettre un bot Python dans systemd

Ce guide explique comment exécuter un bot Python comme service **systemd** sur Linux, avec un environnement virtuel et un fichier de variables d'environnement. Le point clé est d'utiliser le binaire Python du virtualenv directement dans `ExecStart`, puis d'activer le service pour qu'il démarre automatiquement au boot.

Prérequis

Le projet doit idéalement contenir :

- `bot.py`
- un virtualenv, ici nommé `env`
- un fichier `.env` pour les variables sensibles comme le token Discord

Exemple d'arborescence :

```
/home/toto/discord-bot-application/  
├─ bot.py  
├─ .env  
└─ env/
```

Le virtualenv peut avoir n'importe quel nom, mais il faut ensuite pointer vers le bon chemin dans `ExecStart`.

Vérifier les chemins

Avant de créer le service, il faut vérifier que les chemins existent réellement :

```
ls -la /home/toto/discord-bot-application  
ls -la /home/toto/discord-bot-application/env/bin/python
```

Si le second chemin existe, systemd pourra lancer le script avec le Python du virtualenv.

Créer le fichier .env

Le fichier `.env` sert à stocker les variables d'environnement du bot, comme le token Discord ou les IDs du serveur et du salon. Avec systemd, ce fichier peut être chargé via `EnvironmentFile=`.

Exemple :

```
DISCORD_BOT_TOKEN=TON_TOKEN_ICI
DISCORD_GUILD_ID=TON_SERVER_ID
DISCORD_CHANNEL_ID=TON_CHANNEL_ID
```

Le format attendu est `CLE=VALEUR`, sans `export`.

Pour limiter l'accès au fichier :

```
chmod 600 /home/toto/discord-bot-application/.env
```

Tester le bot manuellement

Avant de passer par systemd, il est préférable de confirmer que le script démarre correctement à la main depuis le virtualenv.

```
cd /home/toto/discord-bot-application
source env/bin/activate
python bot.py
```

Si le bot affiche un message comme `Bot prêt`, la partie Python est probablement correcte.

Créer le service systemd

Créer le fichier du service :

```
sudo nano /etc/systemd/system/discord-job-bot.service
```

Puis coller ce contenu :

```
[Unit]
Description=Discord Job Bot
After=network.target

[Service]
Type=simple
User=toto
```

```
Group=toto
WorkingDirectory=/home/toto/discord-bot-application
EnvironmentFile=/home/toto/discord-bot-application/.env
Environment=PYTHONUNBUFFERED=1
ExecStart=/home/toto/discord-bot-application/env/bin/python /home/toto/discord-bot-application/bot.py
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target
```

Explication des lignes importantes

- `User` et `Group` : définissent l'utilisateur Linux qui exécutera le bot.
- `WorkingDirectory` : fixe le dossier courant du projet.
- `EnvironmentFile` : charge les variables du fichier `.env`.
- `Environment=PYTHONUNBUFFERED=1` : force l'affichage immédiat des logs Python dans `journalctl`.
- `ExecStart` : lance directement le script avec le Python du virtualenv `env`.
- `Restart=always` : redémarre automatiquement le bot s'il s'arrête.

Recharger systemd et démarrer le service

Après création ou modification du fichier `.service`, il faut recharger la configuration systemd, activer le service au démarrage, puis le lancer.

```
sudo systemctl daemon-reload
sudo systemctl enable discord-job-bot.service
sudo systemctl start discord-job-bot.service
```

Vérifier l'état du service

Pour vérifier que le service est bien actif :

```
sudo systemctl status discord-job-bot.service
```

Pour suivre les logs en direct :

```
journalctl -u discord-job-bot.service -f
```

`journalctl` est l'outil standard pour consulter les logs systemd et diagnostiquer les problèmes de démarrage.

Commandes utiles

Redémarrer le service après modification du code :

```
sudo systemctl restart discord-job-bot.service
```

Arrêter le service :

```
sudo systemctl stop discord-job-bot.service
```

Désactiver le lancement automatique au démarrage :

```
sudo systemctl disable discord-job-bot.service
```

Erreurs fréquentes

Les problèmes les plus fréquents sont :

- mauvais chemin dans `ExecStart`
- mauvais utilisateur dans `User`
- fichier `.env` absent ou mal formaté
- dépendances Python non installées dans le virtualenv
- permissions insuffisantes sur les fichiers du projet

Pour afficher les 50 dernières lignes de logs sans pagination :

```
journalctl -u discord-job-bot.service -n 50 --no-pager
```

Exemple final prêt à copier

Fichier service

```
[Unit]
Description=Discord Job Bot
After=network.target

[Service]
Type=simple
User=toto
Group=toto
WorkingDirectory=/home/toto/discord-bot-application
EnvironmentFile=/home/toto/discord-bot-application/.env
```

```
Environment=PYTHONUNBUFFERED=1
```

```
ExecStart=/home/toto/discord-bot-application/env/bin/python /home/toto/discord-bot-application/bot.py
```

```
Restart=always
```

```
RestartSec=5
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Commandes d'activation

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable discord-job-bot.service
```

```
sudo systemctl start discord-job-bot.service
```

```
sudo systemctl status discord-job-bot.service
```

```
journalctl -u discord-job-bot.service -f
```

Revision #3

Created 10 May 2026 00:57:16 by Bibi

Updated 10 May 2026 01:03:19 by Bibi