

```
Terminal Shell Édition Présentation Fenêtre Abz
root@ip-172-31-37-222:~/airflow

for repo in "${repos[@]"; do
  echo "==== Traitement du dépôt : $repo ====="
  cd "$repo" || continue
  echo "--- Vérification des modifs locales ---"
  if [ -n "$(git status --porcelain)" ]; then
    echo "Modifications locales détectées, stash en cours..."
    git stash
  fi
  echo "--- Pull des dernières modifications ---"
  git pull origin main
  echo "--- Retour au répertoire parent ---"
  cd ..
done
echo "🟢 Tous les dépôts ont été mis à jour."

root@ip-172-31-37-222:~/airflow# ./update_repos.sh
==== Traitement du dépôt : airflow =====
--- Vérification des modifs locales ---
--- Pull des dernières modifications ---
Already up to date.
--- Retour au répertoire parent ---
==== Traitement du dépôt : airflow-data =====
--- Vérification des modifs locales ---
--- Pull des dernières modifications ---
Already up to date.
--- Retour au répertoire parent ---
🟢 Tous les dépôts ont été mis à jour.
root@ip-172-31-37-222:~/airflow#
```

TUTORIEL PAS À PAS

Automatisez votre script à chaque ouverture de session Mac

Mettez en place un LaunchAgent fiable, réversible et adapté à votre compte utilisateur. Suivez cette fiche pour préparer le script, créer sa configuration et vérifier qu'il s'exécute réellement.

Au sommaire

- 1 Choisir le bon déclenchement
- 2 Préparer le script avant toute automatisation
- 3 Créer le LaunchAgent dans votre session
- 4 Les repères du fichier plist
- 5 Vérifier et dépanner sans tout recommencer

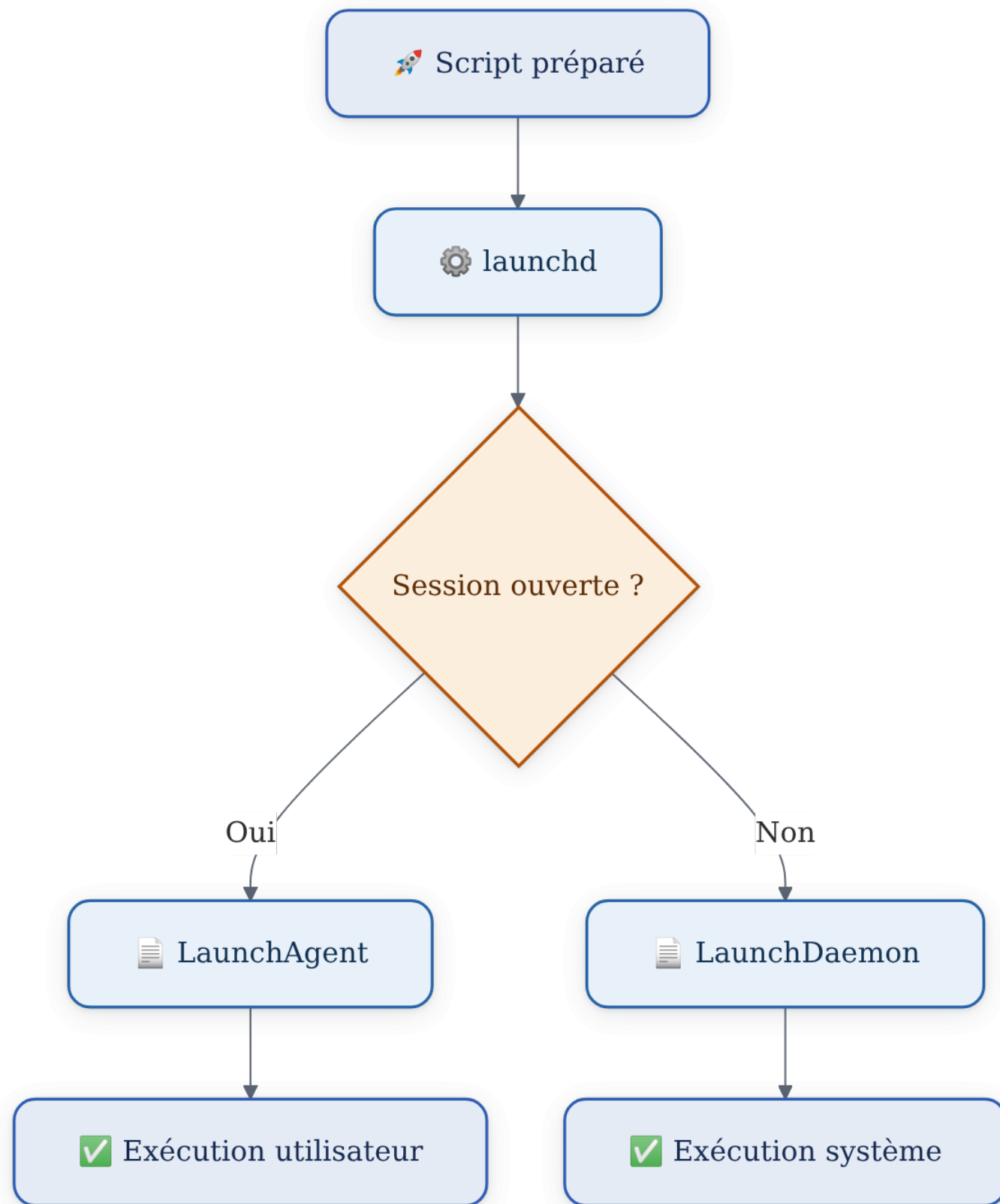
Choisir le bon déclenchement

Choisissez un LaunchAgent pour lancer un script personnel après l'ouverture de votre session macOS.

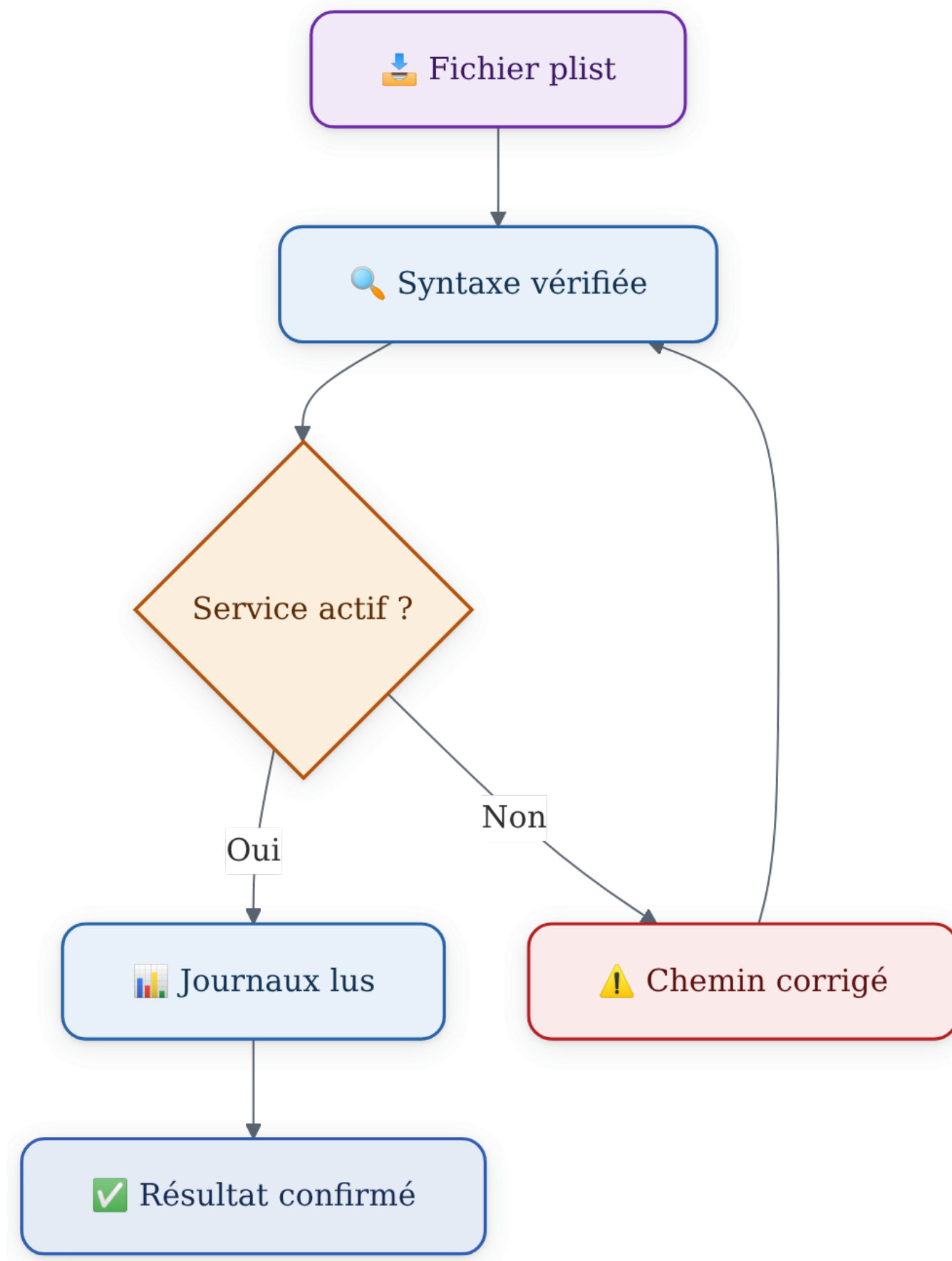
Pour une application visible après connexion, les éléments d'ouverture des Réglages Système peuvent être plus appropriés.

Utilisez un LaunchDaemon uniquement si le service doit fonctionner indépendamment d'un utilisateur ou avant la connexion : sa configuration et ses droits sont plus sensibles.

Pour une tâche récurrente, configurez launchd avec un intervalle ou un calendrier plutôt qu'un simple lancement à la connexion.



INFOGRAPHIE Schéma du fonctionnement d'un LaunchAgent et d'un LaunchDaemon pour lancer un script au démarrage du Mac



Préparer le script avant toute automatisation

Placez le script dans un emplacement personnel stable.

Évitez Téléchargements, les volumes externes et les emplacements temporaires.

Déclarez l'interpréteur en première ligne du script.

Par exemple : `#!/bin/zsh` ou le chemin réel de Python installé sur votre Mac.

Utilisez des chemins absolus.

Ne comptez pas sur le dossier courant, le PATH ou des variables définies dans Terminal.

Rendez le script exécutable uniquement pour votre compte.

La commande `chmod u+x` limite les permissions au nécessaire.

Exécutez le script manuellement avec le même interpréteur.

Corrigez tout échec avant de créer le LaunchAgent.

Prévoyez un journal de sortie et un journal d'erreurs.

Un script en arrière-plan ne fournit pas de retour fiable dans une fenêtre Terminal.

Supprimez les demandes interactives.

Prévoyez une configuration explicite ou une erreur lisible à la place.

Créer le LaunchAgent dans votre session

1. Créez les dossiers nécessaires

Préparez `~/Library/LaunchAgents` pour la configuration et `~/Library/Logs` pour les journaux. Cette opération se fait dans votre dossier utilisateur, sans `sudo`.

2. Créez un fichier plist

Enregistrez-le dans `~/Library/LaunchAgents` avec un nom unique, par exemple `com.exemple.script.plist`. Le fichier plist décrit le programme à exécuter et son comportement.

3. Définissez une étiquette unique

La clé `Label` identifie le service. Utilisez un nom stable pour pouvoir le contrôler et l'arrêter facilement par la suite.

4. Indiquez l'interpréteur et le script

Dans `ProgramArguments`, renseignez l'interpréteur puis le chemin complet du script. Cette méthode évite les ambiguïtés liées à l'environnement réduit de `launchd`.

5. Activez le lancement à la connexion

Ajoutez `RunAtLoad` pour demander l'exécution au chargement du service, donc à l'ouverture de votre session.

6. Redirigez les sorties

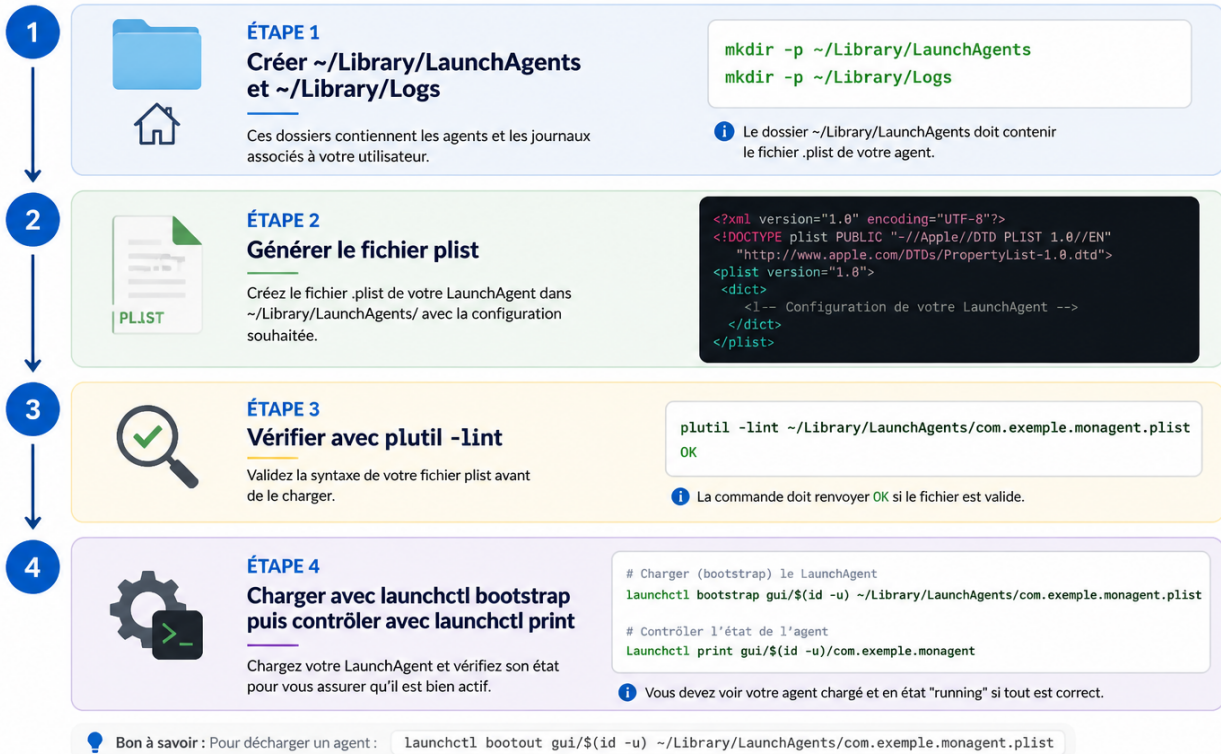
Utilisez `StandardOutPath` et `StandardErrorPath` vers des fichiers de `~/Library/Logs` afin de distinguer les messages normaux des erreurs.

7. Validez puis chargez le service

Vérifiez d'abord la syntaxe XML du plist, puis chargez-le avec `launchctl bootstrap` dans le domaine graphique de votre utilisateur.

LA CRÉATION D'UN LAUNCHAGENT macOS

Guide pas à pas pour créer, vérifier, charger et contrôler un LaunchAgent utilisateur.

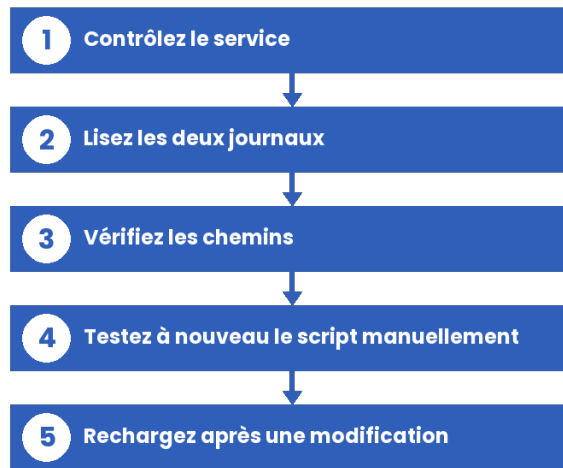


SCHEMA Les étapes essentielles : script prêt, fichier plist, chargement du LaunchAgent et contrôle des journaux.

Les repères du fichier plist

Élément	Rôle	Point de vigilance
Label	Identifie le service launchd.	Choisissez un identifiant unique et stable.
ProgramArguments	Indique le programme ou l'interpréteur, puis le script et ses arguments.	Employez des chemins absolus.
RunAtLoad	Demande l'exécution lors du chargement du service.	Correspond au besoin de lancement à l'ouverture de session.
StandardOutPath	Écrit la sortie normale dans un fichier journal.	Choisissez un chemin accessible et durable.
StandardErrorPath	Écrit les erreurs dans un fichier séparé.	Consultez-le en priorité si le script ne semble pas démarrer.

Vérifier et dépanner sans tout recommencer



1 **Contrôlez le service**

Utilisez `launchctl print` pour vérifier que le LaunchAgent est bien chargé dans le domaine graphique de votre utilisateur.

2 **Lisez les deux journaux**

Consultez la sortie standard et le journal d'erreurs : ils permettent de séparer un problème de script d'un problème de configuration.

3 **Vérifiez les chemins**

Confirmez le chemin de l'interpréteur, celui du script et ceux des fichiers manipulés. Les chemins relatifs et un PATH supposé sont des causes fréquentes d'échec.

4 **Testez à nouveau le script manuellement**

Lancez-le avec le même interpréteur que celui déclaré dans `ProgramArguments` pour reproduire les conditions les plus proches possible.

5 **Rechargez après une modification**

Après avoir corrigé le plist ou le script, déchargez proprement le service puis rechargez la configuration avant de refaire le contrôle.

! Garder une automatisation sûre et réversible

N'utilisez pas un `LaunchDaemon` ni des droits administrateur pour contourner les protections de macOS. Un `LaunchAgent` dans votre dossier utilisateur est généralement le choix le plus prudent pour une tâche personnelle. Conservez les journaux : ils facilitent le diagnostic et permettent de confirmer ce qui a été exécuté.

Cette fiche concerne l'automatisation d'un script personnel via `launchd`. Avant toute exécution automatique, testez le script manuellement et vérifiez les journaux.