Pour intégrer votre code de web scraping et de nettoyage des données dans une data pipeline sous Apache NiFi qui s'exécute toutes les heures, suivez les étapes ci-dessous :

**1. Préparer le code pour le web scraping et le nettoyage**

**A. Créez un script Python unique pour le scraping et le nettoyage**

* Combinez votre code de scraping et de nettoyage dans un script Python unique (par exemple, crypto\_data\_pipeline.py) qui :
  + Récupère les données de CoinMarketCap toutes les heures.
  + Effectue le nettoyage et enregistre les données nettoyées dans un fichier CSV (par exemple, crypto\_data\_cleaned.csv).
* Assurez-vous que le script fonctionne correctement en dehors de NiFi.

**2. Charger le script dans Apache NiFi**

**B. Démarrez Apache NiFi et ouvrez l'interface utilisateur**

* Accédez à http://localhost:8080/nifi.

**C. Créer un nouveau groupe de processus (Process Group)**

* Dans l'interface, cliquez avec le bouton droit sur le *canvas*, sélectionnez **Add Process Group** et nommez-le (par exemple, "CryptoDataPipeline").

**3. Mettre en place les composants NiFi pour exécuter le script et gérer la pipeline**

**D. Ajoutez un ExecuteScript ou ExecuteProcess pour exécuter le script Python**

* Dans le groupe de processus, ajoutez un composant **ExecuteProcess** (ou **ExecuteScript** si le code peut être intégré directement).
* Configurez le processeur pour qu’il exécute le script Python :
  + Dans **Command**, entrez le chemin de l'exécutable Python (ex: /usr/bin/python ou python sur Windows).
  + Dans **Command Arguments**, entrez le chemin vers le script crypto\_data\_pipeline.py.
* Configurez les paramètres du processeur pour définir l'intervalle d'exécution à toutes les heures (voir étape 5).

**E. Ajouter des processeurs pour gérer le flux de données**

* **Input/Output** : si vous voulez lire directement depuis une API ou un fichier pour chaque exécution, ajoutez des processeurs comme **InvokeHTTP** (pour une API) ou **GetFile** (pour un fichier local).
* **PutFile** : utilisez ce processeur pour sauvegarder les résultats nettoyés du fichier CSV (crypto\_data\_cleaned.csv) dans un dossier spécifique pour faciliter le suivi des données traitées.

**F. Ajouter un UpdateAttribute pour gérer l’horodatage**

* Si vous avez besoin d'ajouter l'horodatage à chaque exécution, utilisez un processeur **UpdateAttribute** pour définir un attribut de temps pour chaque fichier ou lot de données.

**4. Configurer l’ordonnancement pour une exécution toutes les heures**

* Cliquez sur le processeur principal (par exemple, ExecuteProcess pour exécuter le script Python).
* Allez dans l’onglet **Scheduling** et configurez :
  + **Run Schedule** à 1 heure pour exécuter le processeur toutes les heures.
* Assurez-vous que tous les processeurs sont configurés pour s'exécuter à l'intervalle souhaité.

**5. Tester la pipeline**

* Démarrez le groupe de processus et vérifiez que les données sont bien scrappées, nettoyées, et stockées dans le dossier spécifié.
* Vous pouvez surveiller l’exécution de chaque processeur dans l'interface NiFi et vérifier les journaux pour déceler d’éventuelles erreurs.

**6. Superviser et ajuster les performances**

* Dans NiFi, utilisez le **Data Provenance** pour suivre l'exécution de chaque donnée.
* Si des erreurs surviennent, vérifiez les logs de NiFi pour diagnostiquer le problème.

**7. Automatiser la pipeline avec NiFi Flow Management**

* Si des améliorations sont nécessaires, vous pouvez ajuster la fréquence d'exécution, ajouter des étapes de validation des données, ou encore configurer des alertes en cas d’échec.

Avec cette configuration, votre pipeline NiFi s'exécutera automatiquement toutes les heures pour scrapper, nettoyer et sauvegarder les données des crypto-monnaies, consolidant ainsi votre pipeline complet dans un workflow NiFi.