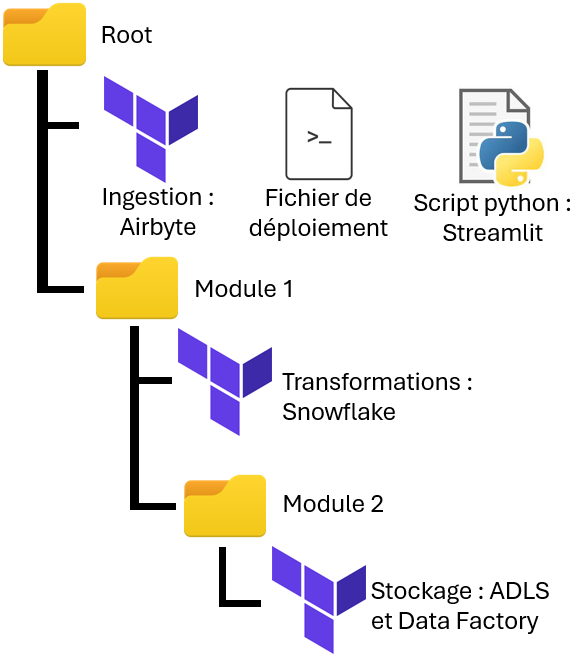
Les différents fichiers terraform du projet sont organisés en modules. Ceci permet de transmettre les variables entre chaque fichier.

Pour lancer le déploiement automatique, nous avons un fichier batch qui vérifie que les outils nécessaires sont présents sur la machine, puis exécute les commandes « terraform init » et « terraform apply ».

Un fichier terraform peut inclure les ressources d’un module s’il est défini en module. Cela permet d’appliquer plusieurs fichiers avec une seule commande, et les modules partagent automatiquement leurs outputs avec les fichiers qui les appellent. Dans notre cas, le Data Lake est le module le plus bas dans la hiérarchie car ses informations doivent être partagées avec la partie Transformations et la partie Ingestion.  
Le module intermédiaire doit aussi définir les outputs à nouveau pour les passer au module suivant.Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

La décision de mettre la partie Ingestion en racine plutôt que la partie Transformation est arbitraire. Cependant, mettre les deux fichiers au même niveau ne fonctionne pas, car s’ils appellent tous les deux le même module, Terraform essaiera de le déployer deux fois en même temps, ce qui entraîne des erreurs et empêche le déploiement. Cette structure en « cascade » est donc nécessaire.



Avec cette organisation, lancer un « terraform apply » au niveau root commencera en réalité par déployer le module 2 (Stockage), puis le module 1 (Transformation) et enfin le module principal (ingestion).