Sommaire

Vue d'ensemble des modèles cloud

1. **Infrastructure as a Service (IaaS):** IaaS fournit des ressources informatiques virtualisées sur Internet. Dans ce modèle, les fournisseurs de cloud comme Microsoft Azure offrent des réseaux, des serveurs, du stockage et des systèmes d'exploitation. ADF peut être utilisé pour orchestrer et automatiser des transferts de données entre différentes ressources IaaS.

2. **Platform as a Service (PaaS):** PaaS fournit une plateforme et un environnement permettant aux développeurs de construire des applications et des services sur Internet. Les services PaaS incluent des bases de données, des environnements d'exécution de middleware, des outils de développement, etc.
   ADF est lui-même un exemple de PaaS, car il fournit une plateforme qui permet aux utilisateurs de créer, de gérer et d'orchestrer des flux de données sans avoir à gérer l'infrastructure sous-jacente. Les utilisateurs peuvent développer des pipelines de données qui intègrent divers services Azure comme Azure SQL Database, Blob Storage, et Table Storage, tout en bénéficiant de la scalabilité et de la gestion du PaaS.

3. **Software as a Service (SaaS):** SaaS offre aux utilisateurs un logiciel complet et fonctionnel fonctionnant sur une infrastructure distante. Les utilisateurs peuvent accéder à ce logiciel via un navigateur web sans se soucier de la maintenance ou de la gestion de l'infrastructure sous-jacente.
   Bien qu'ADF ne soit pas un SaaS, il peut intégrer des applications SaaS dans ses workflows. Par exemple, ADF peut être utilisé pour extraire des données de services SaaS tels que Salesforce ou Office 365, les transformer et les charger dans des systèmes pour une analyse plus poussée.

Vue d'ensemble des modèles cloud

4. **Function as a Service (FaaS) / Serverless:** FaaS, ou computing sans serveur, permet aux développeurs d'exécuter du code en réponse à des événements sans avoir à gérer explicitement les serveurs. Le fournisseur de cloud gère la capacité de calcul, le stockage, et le réseau.
   ADF peut intégrer des services serverless tels que Azure Functions pour exécuter des fonctions personnalisées qui répondent à des événements spécifiques dans un pipeline de données. Cela permet des transformations de données flexibles et dynamiques basées sur des événements.

Avantages des technologies cloud

1. **Scalabilité et Élasticité**: Les technologies cloud permettent aux organisations de dimensionner leurs ressources informatiques en fonction de la demande. Azure, par exemple, permet aux utilisateurs d'ajouter ou de réduire des capacités de calcul et de stockage en temps réel sans interruption de service, offrant ainsi une grande flexibilité.
2. **Coût-Efficacité**: Avec le modèle de paiement à l'usage, les entreprises ne paient que pour les ressources qu'elles utilisent. Cela élimine le besoin d'investissements initiaux lourds en infrastructure informatique et réduit les coûts de maintenance et d'exploitation.
3. **Accessibilité et Collaboration**: Le cloud permet l'accès aux données et aux applications de n'importe où, à tout moment, à condition d'avoir une connexion Internet. Cela facilite la collaboration entre équipes dispersées géographiquement et soutient le travail à distance.
4. **Récupération après Sinistre et Continuité des Affaires**: Les services cloud offrent des solutions robustes de sauvegarde et de récupération des données, minimisant les risques de perte de données en cas de panne matérielle ou de désastre naturel.

Avantages des technologies cloud

5. **Sécurité**: Les fournisseurs de cloud comme Microsoft investissent massivement dans des protocoles de sécurité avancés, des centres de données sécurisés, et des certifications de conformité, offrant un niveau de sécurité souvent supérieur à ce que les entreprises peuvent atteindre par elles-mêmes.
6. **Mises à Jour et Innovation Constantes**: Le cloud facilite l'adoption rapide des dernières technologies sans nécessiter des mises à jour matérielles coûteuses. Les fournisseurs de cloud déploient régulièrement des améliorations et des nouvelles fonctionnalités sans interruption de service pour les clients.

Services principaux d'Azure

1. **Compute**
   - **Azure Virtual Machines** : pour déployer des serveurs virtuels configurables.
   - **Azure Kubernetes Service (AKS)** : pour la gestion des applications conteneurisées.
   - **Azure Functions** : pour exécuter des morceaux de code sans provisionner ou gérer des serveurs (serverless).
2. **Stockage**
   - **Azure Blob Storage** : pour le stockage d'objets à grande échelle, idéal pour les données non structurées.
   - **Azure File Storage** : pour le partage de fichiers entre applications utilisant le standard SMB.
   - **Azure Queue Storage** : pour le stockage de messages utilisé pour la communication entre les composants de l'application.
3. **Bases de données**
   - **Azure SQL Database** : une base de données relationnelle en tant que service.
   - **Cosmos DB** : une base de données NoSQL pour le stockage et la gestion de données à l'échelle mondiale.
   - **Azure Database for MySQL et PostgreSQL** : services de base de données gérés pour les systèmes de gestion de bases de données open source populaires.

Services principaux d'Azure

4. **AI et Machine Learning**
   - **Azure Machine Learning Service** : une plateforme cloud pour le développement, le déploiement et la gestion de solutions ML.
   - **Azure Cognitive Services** : une collection de services prêts à l'emploi qui permettent aux applications de voir, entendre, parler, comprendre et interpréter les besoins des utilisateurs.
5. **Réseau**
   - **Azure Virtual Network** : pour la création de réseaux privés dans le cloud.
   - **Azure ExpressRoute** : pour des connexions privées entre les datacenters Azure et l'infrastructure locale.
6. **DevOps**
   - **Azure DevOps** : une suite de services pour soutenir une culture DevOps, incluant CI/CD, planification agile, et collaboration.
7. **Sécurité**
   - **Azure Security Center** : offre une visibilité unifiée de la sécurité de tous les services Azure, renforçant la posture de sécurité.

Services principaux d'Azure

Azure Data Factory (ADF) est un service d'intégration de données en cloud, proposé par Microsoft dans sa plateforme Azure, qui permet aux organisations de coordonner et d'automatiser le mouvement et la transformation des données à grande échelle. ADF est conçu pour faciliter la création, la programmation et l'orchestration de flux de données complexes où les données sont collectées à partir de diverses sources, transformées et chargées dans des data stores pour analyse et visualisation.

1. **Intégration de Données** : ADF permet de collecter des données de multiples sources, y compris des bases de données sur site, des systèmes cloud, et des sources de données non structurées. Il peut intégrer, par exemple, des données de systèmes comme SQL Server, Oracle, des fichiers CSV, des bases de données NoSQL comme Cosmos DB, et des services cloud comme Amazon S3 et Google BigQuery.
2. **Transformation de Données** : Avec Azure Data Factory, les utilisateurs peuvent effectuer des transformations de données à l'aide de services comme Azure HDInsight (Apache Hadoop) pour des traitements de big data, Azure Databricks pour des traitements avancés utilisant Spark, ou des services de transformation intégrés appelés Mapping Data Flows, qui permettent de construire des transformations de données visuelles sans coder.
3. **Orchestration et Automatisation** : ADF fournit des outils pour orchestrer et automatiser des workflows de données, permettant de planifier et de gérer le mouvement de données et les workflows de transformation.

Architecture d'Azure Data Factory

1. **Pipeline** : Le pipeline est le composant principal d'ADF, qui définit un workflow de données. Un pipeline peut contenir une ou plusieurs activités, qui sont des tâches spécifiques à effectuer sur les données.
2. **Activités** : Les activités sont les tâches que les pipelines exécutent, telles que la copie de données, l'exécution de procédures stockées, l'exécution de scripts Hive, Pig, ou Spark, ou des activités personnalisées.
3. **Datasets** : Les datasets représentent les données à utiliser dans les activités du pipeline, définissant les structures de données d'entrée ou de sortie.
4. **Linked Services** : Les services liés (linked services) fonctionnent comme des connexions à des sources de données, définissant les informations de connexion nécessaires pour que ADF puisse accéder aux données.
5. **Triggers** : Les déclencheurs contrôlent quand un pipeline doit s'exécuter, pouvant être définis sur une base horaire, quotidienne ou en réponse à un événement.

Cas d'usage typiques

1. **ETL et ELT Modernes** : ADF est souvent utilisé pour moderniser les solutions ETL (Extract, Transform, Load) et ELT (Extract, Load, Transform), profitant de la puissance du cloud pour gérer de grandes quantités de données plus efficacement.
2. **Migration de Données** : Les entreprises utilisent ADF pour migrer des données d'une variété de sources vers le cloud, aidant à centraliser les données pour des analyses avancées et des décisions basées sur les données.
3. **Intégration de Données Entreprise** : ADF peut intégrer des données provenant de différents départements ou sources de données pour fournir une vue unifiée, supportant les initiatives de business intelligence et d'analyse de données.
4. **Automatisation de Processus de Données** : Avec ADF, les organisations peuvent automatiser les processus de collecte, de transformation, et de distribution des données, réduisant les coûts opérationnels et augmentant l'efficacité.

Différences entre ETL et ELT

1. **ETL (Extract, Transform, Load)**

- **Extraction** : Les données sont extraites de la ou des sources originelles, qui peuvent inclure des bases de données, des systèmes ERP, des fichiers plats, etc.
- **Transformation** : Cette étape est réalisée avant de charger les données dans la destination finale. Elle est généralement effectuée dans un serveur séparé où les données sont nettoyées, reformatées, agrégées, et enrichies. Cela peut inclure des opérations telles que le filtrage, la validation, la jonction, la transposition de colonnes en lignes, etc.
- **Chargement** : Les données transformées sont ensuite chargées dans un système de stockage ou un entrepôt de données.

Différences entre ETL et ELT

2. **ELT (Extract, Load, Transform)**

- **Extraction** : Comme pour ETL, les données sont extraites de la source.
- **Chargement** : Les données sont chargées directement dans la destination finale, souvent un entrepôt de données moderne qui est capable de stocker de grandes quantités de données brutes.
- **Transformation** : Les transformations sont effectuées après le chargement, directement dans la base de données ou l'entrepôt de données. Cette approche tire parti de la puissance de calcul de la destination finale pour effectuer des transformations.

Différences entre ETL et ELT Avantages

1. **ETL**

- **Contrôle et Qualité des Données** : Puisque la transformation se fait avant le chargement, il y a un contrôle strict sur la qualité des données entrant dans l'entrepôt.
- **Performance** : Moins de charge sur l'entrepôt de données car les données sont déjà transformées et prêtes à être utilisées.
- **Sécurité** : Moins de données sensibles stockées ou traitées dans l'entrepôt de données car les transformations et le filtrage des données sont réalisés en amont.

2. **ELT**

- **Scalabilité** : Profite de la haute performance et de la scalabilité des entrepôts de données modernes pour traiter de grandes quantités de données.
- **Flexibilité** : Les transformations peuvent être modifiées ou réexécutées facilement puisque les données brutes sont déjà chargées dans l'entrepôt.
- **Efficacité Coût-Temps** : Moins de temps passé à transformer les données en dehors de l'entrepôt, ce qui réduit les coûts d'infrastructure et accélère le processus de chargement.

Différences entre ETL et ELT Inconvénients

1. **ETL**

- **Coût d'Infrastructure** : Nécessite souvent des serveurs supplémentaires pour la transformation des données, ce qui augmente les coûts.
- **Flexibilité Réduite** : Modifier les transformations peut être laborieux puisque cela nécessite souvent de retransformer les données et de les recharger.
- **Délais** : Le processus peut être plus lent, car les données doivent être complètement transformées avant le chargement.

2. **ELT**

- **Dépendance sur l'Entrepôt de Données** : Nécessite un système puissant capable de gérer de lourdes charges de transformation.
- **Sécurité et Gouvernance** : Les données non transformées peuvent inclure des informations sensibles qui sont chargées dans l'entrepôt, posant potentiellement des défis de conformité et de sécurité.

Comparaison ADF et SSIS

SQL Server Integration Services (SSIS):

- Lancé en 2005 comme un composant de Microsoft SQL Server.
- Conçu pour des solutions ETL (Extract, Transform, Load) pour les systèmes de bases de données, principalement on-premise.
- Orienté vers des scénarios où les données sont centralisées sur des serveurs de bases de données SQL Server.

Azure Data Factory (ADF):

- Introduit en 2015 comme un service cloud dans Microsoft Azure.
- Conçu pour orchestrer et automatiser le mouvement et la transformation de données dans des environnements cloud et hybrides.
- Adapté à des scénarios de big data et intégration de données multi-cloud et multi-source.

Comparaison ADF et SSIS

SSIS:

- **Transformation de Données**: Fortes capacités de transformation de données avec une large gamme de composants transformateurs disponibles.
- **Déploiement**: Déployé localement ou dans le cloud en tant que machine virtuelle mais conçu originellement pour des déploiements sur site.
- **Connectivité**: Excellente connectivité avec des systèmes Microsoft et des bases de données via des connecteurs dédiés.
- **Interface Utilisateur**: Fournit une interface graphique riche (SQL Server Data Tools) pour le design de packages ETL.

ADF:

- **Orchestration de Données**: Excellente pour orchestrer des workflows de données complexes, pas seulement des tâches ETL/ELT.
- **Intégration Cloud**: Native du cloud, optimisée pour intégrer facilement des données de diverses sources cloud et on-premise.
- **Scalabilité et Performance**: Gère dynamiquement la scalabilité et est capable de traiter des volumes de données beaucoup plus importants.
- **Serverless**: Ne nécessite pas de gestion de l'infrastructure, les ressources sont gérées automatiquement.

Comparaison ADF et SSIS Avantages et Inconvénients

Avantages de SSIS:

- **Familiarité**: Bien intégré dans l'écosystème Microsoft SQL Server, ce qui est avantageux pour les équipes IT existantes.
- **Riche en fonctionnalités de transformation**: Large ensemble de transformations prêtes à l'emploi.
- **Coût**: Pas de coût additionnel si vous avez déjà une licence SQL Server Enterprise.

Inconvénients de SSIS:

- **Scalabilité**: Peut nécessiter une gestion manuelle de la scalabilité.
- **Dépendance de l'Infrastructure**: Nécessite une maintenance et une gestion de l'infrastructure sous-jacente.

Comparaison ADF et SSIS Avantages et Inconvénients

Avantages de ADF:

- **Flexibilité et Scalabilité**: Capacités de scalabilité automatique pour traiter des volumes de données élevés.
- **Intégration avec Azure**: Intégration native avec d'autres services Azure, comme Azure Databricks, Azure SQL Database, et Azure Blob Storage.
- **Support pour le Big Data**: Conçu pour des solutions de big data utilisant des frameworks comme Hadoop ou Spark.

Inconvénients de ADF:

- **Complexité**: Peut être perçu comme plus complexe à configurer pour des utilisateurs non familiers avec le cloud.
- **Coût**: Peut être coûteux en fonction du volume des données traitées et de la fréquence des opérations.

Comparaison ADF et SSIS Cas d'utilisation

SSIS est préférable quand:

- Vous avez des solutions de données principalement localisées ou centrées sur SQL Server.
- Vous nécessitez une intégration étroite avec des applications Microsoft.
- Votre architecture IT est principalement on-premise.

ADF est préférable quand:

- Vous êtes engagé dans une stratégie cloud-first ou cloud-native.
- Vous traitez des données provenant de multiples sources cloud et on-premise.
- Vous avez besoin de traiter des volumes importants de données ou des scénarios de big data.

Implémentation et Utilisation d'Azure Data Factory

1. Configuration de Base

Création d'une instance d'Azure Data Factory :

- Connectez-vous au portail Azure.
- Sélectionnez "Créer une ressource" et recherchez "Data Factory".
- Remplissez les détails nécessaires comme le nom, la région, la version (V1 ou V2) et le groupe de ressources.
- Une fois créé, vous pouvez accéder à l'interface utilisateur d'ADF en sélectionnant "Author & Monitor" dans votre instance de Data Factory.

Implémentation et Utilisation d'Azure Data Factory

2. Configuration des Linked Services

Les Linked Services dans ADF agissent comme des connexions à vos sources de données et destinations.

Étapes pour configurer un Linked Service :

- Dans l'interface d'ADF, allez à la section "Manage" et cliquez sur "Linked services".
- Cliquez sur "New" pour créer un nouveau linked service.
- Sélectionnez le type de service ou de stockage que vous souhaitez connecter (par exemple, Azure SQL Database, Azure Blob Storage).
- Configurez les paramètres de connexion nécessaires et testez la connexion avant de sauvegarder.

Implémentation et Utilisation d'Azure Data Factory

3. Création de Datasets

Les Datasets représentent les structures de données spécifiques que vous utiliserez dans vos pipelines.

Création d'un Dataset :

- Dans l'interface "Author", cliquez sur "Datasets" et ensuite sur "New dataset".
- Sélectionnez le type de dataset correspondant à votre source ou destination de données.
- Associez le dataset à un linked service existant.
- Configurez les paramètres spécifiques du dataset, comme la table, le schéma, ou le chemin du fichier.

Implémentation et Utilisation d'Azure Data Factory

4. Création de Pipelines

Les pipelines sont au cœur d'ADF, orchestrant le flux de données entre différentes activités et services.

Étapes pour créer un pipeline :

- Cliquez sur "+", puis sur "Pipeline" dans l'interface "Author".
- Glissez et déposez des activités dans l'espace de conception, telles que "Copy Data", "Data Flow", "For Each", ou des activités personnalisées.
- Configurez chaque activité, en spécifiant les datasets d'entrée et de sortie, les paramètres de l'activité, et les options de débogage.
- Validez le pipeline et déclenchez-le manuellement ou configurez un déclencheur pour l'exécuter automatiquement.

Implémentation et Utilisation d'Azure Data Factory

5. Monitoring et Gestion des Pipelines

La surveillance des exécutions de pipeline est cruciale pour maintenir la fiabilité et l'efficacité de vos flux de données.

Monitoring des pipelines :

- Utilisez l'onglet "Monitor" dans l'interface d'ADF pour voir les exécutions de pipeline, les performances, et les éventuelles erreurs.
- Configurez des alertes pour être notifié en cas de problèmes avec vos pipelines.

Implémentation et Utilisation d'Azure Data Factory

6. Sécurité et Conformité

Assurer la sécurité des données est essentiel dans tout projet d'intégration de données.

Sécurité dans ADF :

- Utilisez Azure Active Directory pour gérer les authentifications et les autorisations.
- Configurez des politiques de sécurité, comme le contrôle d'accès basé sur les rôles (RBAC), pour contrôler l'accès aux ressources d'ADF.
- Assurez-vous que toutes les connexions de données sont sécurisées et cryptées.

Exercice

Objectif: Construire un pipeline dans Azure Data Factory qui extrait des données de ventes quotidiennes à partir d'une source de données, les agrège pour calculer le total des ventes par région, et charge les résultats dans une base de données cible pour le reporting.

La société XYZ souhaite consolider ses données de ventes quotidiennes provenant d'une base de données SQL Azure pour créer un rapport quotidien des ventes par région. Les données doivent être extraites, agrégées, et les résultats chargés dans une autre table SQL pour consultation par le département des ventes.

Exercice

L'entreprise fictive XYZ souhaite analyser les tendances des ventes pour optimiser ses opérations de stock. Elle dispose de données de ventes stockées dans plusieurs fichiers CSV dans Azure Blob Storage et souhaite utiliser Azure Databricks pour les transformer avant de les charger dans Azure SQL Database pour le reporting.

1. **Extraire des données** : Charger des fichiers CSV de données de vente depuis Azure Blob Storage.
2. **Transformer les données** : Utiliser Azure Databricks pour nettoyer les données et calculer des agrégats de ventes par produit.
3. **Charger les données** : Stocker les données transformées dans Azure blob storage pour le reporting.

```
Date,SaleID,ProductID,Quantity,UnitPrice
2024-01-01,001,1001,2,20.00
2024-01-01,002,1002,1,30.00
2024-01-02,003,1003,1,15.00
2024-01-02,004,1001,3,20.00
2024-01-03,005,1005,2,25.00
2024-01-03,006,1002,2,30.00
```