GCP

m2iformation.fr

Sommaire

- Rappel Provider Cloud
- Présentation de GCP
- Intéraction de GCP
- Commencer avec GCP
- Service de Calcule GCP
- GCP Compute Engine
- GCP Cloud Function et Cloud RUN

Sommaire

Sommaire

- Service de traitement de données
  - Cloud dataflow
  - Cloud dataprep
  - Cloud Pub/Sub
  - Cloud Data Fusion
  - Cloud Dataproc
  - Cloud Composer

Rappel Provider Cloud

1. Définition du Cloud Computing

- Le Cloud Computing est la livraison de services informatiques (serveurs, stockage, bases de données, mise en réseau, logiciels, etc.) via Internet (le cloud), permettant une innovation rapide, des ressources flexibles et des économies d'échelle.

2. Pourquoi le Cloud?

- **Économies d'échelle**: Payez uniquement pour les ressources que vous utilisez.
- **Flexibilité et Évolutivité**: Ajustez facilement les ressources en fonction de vos besoins.
- **Accès Mondial**: Accédez à vos services de n'importe où dans le monde.

3. Principaux Fournisseurs de Services Cloud

- **Amazon Web Services (AWS)**: Leader du marché offrant une large gamme de services de calcul, de stockage et de bases de données.
- **Microsoft Azure**: Offre une plateforme cloud intégrée pour les entreprises pour déployer, gérer et développer des applications.
- **Google Cloud Platform (GCP)**: Spécialisé dans les analyses de données, l'apprentissage automatique et des infrastructures évolutives.

Rappel Provider Cloud

4. Types de Services Cloud

- **Infrastructure as a Service (IaaS)**: Location de l'infrastructure IT.
- **Platform as a Service (PaaS)**: Plateforme de développement et déploiement d'applications.
- **Software as a Service (SaaS)**: Accès aux applications logicielles via le cloud.

5. Choix du Fournisseur

- **Spécificités du Projet**: Choisissez en fonction des besoins spécifiques de votre projet.
- **Compétences Techniques**: Préférez les plateformes avec lesquelles votre équipe est la plus familière.
- **Coût**: Comparez les coûts en fonction de l'utilisation prévue.

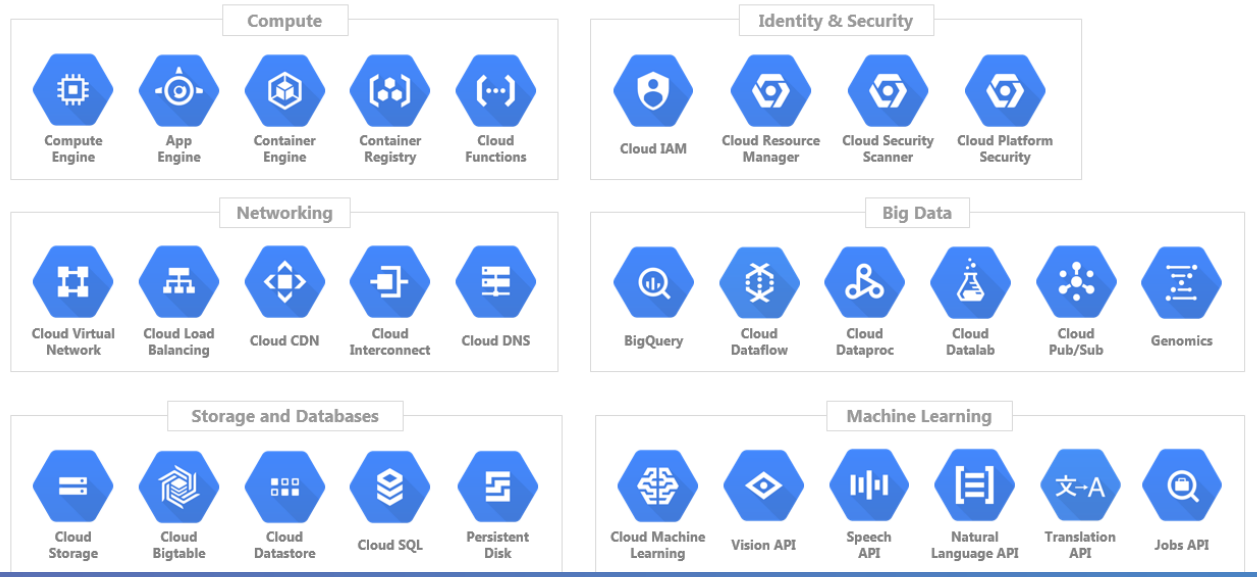
Présentation de GCP

Google Cloud Platform (GCP) est une suite de services de cloud computing proposée par Google qui s'appuie sur la même infrastructure interne que celle utilisée par Google pour ses produits destinés aux utilisateurs finaux, comme Google Search, Gmail, Google Drive et YouTube. GCP offre une large gamme de services dans plusieurs domaines, notamment le calcul, le stockage, la mise en réseau, le Big Data, l'apprentissage automatique (Machine Learning), l'Internet des Objets (IoT), la sécurité et le cloud management.

1. **Compute Engine** : Un service IaaS (Infrastructure as a Service) qui permet aux utilisateurs de lancer des machines virtuelles (VM) sur la infrastructure de Google.
2. **App Engine** : Une plateforme PaaS (Platform as a Service) pour le développement et l'hébergement d'applications web dans les centres de données gérés par Google.
3. **Google Kubernetes Engine (GKE)** : Un service de gestion de conteneurs pour l'exécution et l'orchestration de systèmes d'applications conteneurisées à l'aide de Kubernetes.
4. **Cloud Storage** : Un service de stockage d'objets puissant et simple pour stocker et accéder à des données depuis n'importe où sur le web.
5. **BigQuery** : Un entrepôt de données d'entreprise pour l'analyse de Big Data qui est entièrement géré et sans serveur, permettant des analyses rapides SQL sur de grands ensembles de données.

Présentation de GCP

6. **Cloud Pub/Sub** : Un service de messagerie asynchrone qui permet d'intégrer des systèmes distribués et des applications asynchrones.
7. **Cloud Machine Learning Engine** : Un service géré pour la construction de modèles d'apprentissage automatique à grande échelle, basé sur les frameworks populaires comme TensorFlow.
8. **Cloud IoT Core** : Un service entièrement géré pour connecter, gérer et ingérer les données des appareils IoT (Internet des Objets).



Intéraction de GCP

Il existe plusieurs approches, chacune adaptée à différents besoins et niveaux de compétence technique :

1. Console Google Cloud

La Console Google Cloud est l'interface utilisateur graphique web pour gérer vos services et ressources Google Cloud. C'est le point de départ le plus accessible pour les nouveaux utilisateurs de GCP. Vous pouvez créer et gérer des projets, configurer des réseaux, lancer des machines virtuelles, configurer des bases de données.

2. Google Cloud SDK (gcloud)

Le Google Cloud SDK est un ensemble d'outils en ligne de commande qui vous permettent de gérer vos ressources Google Cloud depuis votre terminal. Le `gcloud` est l'outil principal pour créer et gérer les ressources GCP. Il est particulièrement utile pour l'automatisation des tâches via des scripts.

3. Google Cloud Shell

Google Cloud Shell est un environnement de ligne de commande géré par Google, accessible directement via la Console Google Cloud. Il vient préinstallé avec le Google Cloud SDK et d'autres outils utiles. C'est une option pratique pour exécuter rapidement des commandes `gcloud` sans avoir à installer le SDK sur votre machine locale.

4. Client Libraries

Pour les développeurs qui souhaitent intégrer des services GCP dans leurs applications, Google fournit des bibliothèques clientes dans plusieurs langages de programmation tels que Python, Java, Node.js, Go, .NET, et d'autres. Ces bibliothèques facilitent l'interaction avec les API de Google Cloud à partir de votre code.

Intéraction de GCP

5. API REST

Tous les services GCP sont accessibles via des API REST. Si vous avez besoin d'une flexibilité maximale ou si vous utilisez un langage de programmation pour lequel il n'existe pas de bibliothèque cliente officielle, vous pouvez interagir directement avec ces API REST pour gérer les services et ressources GCP.

6. IaC

- **Terraform**: Terraform est un outil open-source d'IaC qui permet de définir l'infrastructure sous forme de code (utilisant la syntaxe HCL - HashiCorp Configuration Language). Il peut être utilisé pour provisionner et gérer des ressources sur GCP ainsi que sur d'autres plateformes cloud. Terraform fonctionne selon le principe du plan d'exécution, où il génère un plan d'actions à partir de votre code d'infrastructure, que vous pouvez ensuite appliquer pour atteindre l'état désiré.
- **Google Cloud Deployment Manager**: Google Cloud Deployment Manager est un service d'automatisation de l'infrastructure natif de GCP qui vous permet de créer et de gérer des ressources cloud à l'aide de templates. Ces templates peuvent être écrits en YAML, Python, ou Jinja2. Deployment Manager permet de déployer une infrastructure complète à partir d'un ensemble de fichiers de configuration, facilitant ainsi la gestion des ressources à grande échelle.

Commencer avec GCP

Pour commencer avec GCP

1. **Créez un compte Google Cloud** : Rendez-vous sur le site de Google Cloud Platform et inscrivez-vous ou connectez-vous avec votre compte Google.
2. **Créez un projet** : Un projet GCP sert de conteneur pour vos ressources GCP. Toutes les ressources que vous créez sont associées à un projet.
3. **Explorez et activez des services** : Dans la Console Google Cloud, explorez les différents services disponibles et activez ceux que vous souhaitez utiliser.
4. **Configurez l'authentification** : Pour utiliser le SDK ou les API, vous aurez besoin de configurer l'authentification, généralement via un compte de service et des clés d'API.
5. **Commencez à construire** : Que ce soit via la console, le SDK, ou directement via les API, vous pouvez maintenant commencer à construire et déployer vos applications sur GCP.

Installation Gcloud cli

- Lien d'installation `https://cloud.google.com/sdk/docs/install`
- Connexion avec gcloud `gcloud auth login`
- Pour voir la liste des crédentials `gcloud auth list`

Service de Calcule GCP

1. Google Compute Engine (GCE)

Compute Engine offre des machines virtuelles (VM) hautement personnalisables qui s'exécutent sur l'infrastructure avancée de Google. Vous pouvez choisir parmi différentes configurations de CPU, de mémoire, de disque et de GPU pour répondre à vos besoins spécifiques. C'est idéal pour les tâches de calcul général, comme l'hébergement de sites web, d'applications d'entreprise, et de bases de données.

2. Google Kubernetes Engine (GKE)

Kubernetes Engine est un service de gestion de conteneurs orchestré par Kubernetes qui permet d'exécuter des applications conteneurisées sur Google Cloud. GKE automatise la gestion, la mise à l'échelle et le déploiement des applications conteneurisées, ce qui le rend idéal pour les architectures de microservices et les applications évolutives.

3. Google App Engine (GAE)

App Engine est une plateforme en tant que service (PaaS) qui permet aux développeurs de construire et d'héberger des applications web et mobiles sur l'infrastructure de Google. GAE gère automatiquement l'équilibrage de charge, le monitoring et la gestion des serveurs, permettant aux développeurs de se concentrer sur le code de leur application. Il prend en charge plusieurs langages de programmation, tels que Java, Python, PHP, et Node.js.

Service de Calcule GCP

4. Google Cloud Functions (GCF)

Cloud Functions est un service de calcul sans serveur qui exécute du code en réponse à des événements. Il est idéal pour créer des applications orientées événements, des pipelines de données, et des intégrations d'API, sans avoir à gérer l'infrastructure sous-jacente. Les fonctions peuvent être écrites dans plusieurs langages, tels que Node.js, Python, Go, et Java.

5. Google Cloud Run

Cloud Run est un service de calcul sans serveur qui permet d'exécuter des conteneurs sans état, entièrement gérés ou dans un environnement Kubernetes. Cloud Run est optimal pour les applications qui ont besoin de démarrer rapidement et de s'adapter automatiquement à la demande, tout en payant uniquement pour les ressources utilisées.

6. Google Cloud GPUs et TPUs

GCP offre également la possibilité d'ajouter des **unités de traitement graphique (GPUs)** et des **unités de traitement tensoriel (TPUs)** à vos VMs pour des tâches de calcul intensif telles que le machine learning, l'analyse de données, et le rendu graphique.

Service de Calcule GCP Compute Engine

- **Performance et Flexibilité**: Compute Engine offre une large gamme de types de machines (VM) qui peuvent être personnalisées pour répondre à divers besoins de performance. Cela inclut des configurations optimisées pour le calcul, la mémoire, ou l'E/S, ainsi que la possibilité d'utiliser des GPU pour l'accélération de calcul. Cette flexibilité est cruciale pour les tâches de data science, qui peuvent varier de l'analyse de données légère à des entraînements de modèles de machine learning intensifs en ressources.
- **Scalabilité**: Avec Compute Engine, vous pouvez facilement mettre à l'échelle vos ressources verticalement (en améliorant les capacités d'une seule VM) ou horizontalement (en augmentant le nombre de VMs) pour gérer des charges de travail variables. Cette capacité est essentielle pour les projets de données qui nécessitent des ressources supplémentaires pour traiter des volumes de données plus importants ou pour réduire le temps d'exécution des tâches de calcul intensives.

Service de Calcule GCP Compute Engine

- **Intégration avec les services de données GCP**: Compute Engine s'intègre étroitement avec d'autres services GCP, offrant une solution complète pour les pipelines de données.
  - **Cloud Storage** pour le stockage d'objets à haute durabilité et accessibilité.
  - **BigQuery** pour l'analyse de grands ensembles de données.
  - **Cloud Dataproc** pour le traitement de données big data via des clusters Hadoop et Spark.
  - **Cloud Dataflow** pour le traitement de flux et de lots de données en temps réel.
  - **AI Platform** pour la formation et l'inference de modèles de machine learning.
- **Sécurité et Fiabilité** Compute Engine offre une sécurité robuste, avec des réseaux virtuels isolés, des options de chiffrement pour les données au repos et en transit, et la gestion des identités et des accès. La fiabilité est assurée par des SLAs compétitifs, des redémarrages automatiques de VM en cas de défaillance, et la possibilité de créer des architectures hautement disponibles à travers des zones et des régions multiples.
- **Tarification Flexible**: La tarification de Compute Engine est compétitive et flexible, avec des options de paiement à l'utilisation, des engagements à long terme pour des réductions de coûts (engagements d'utilisation et instances réservées), et des tarifs préemptifs pour les VMs qui peuvent être interrompues pour des réductions de coûts supplémentaires. Cette flexibilité tarifaire permet aux équipes de data de gérer efficacement leurs coûts en fonction de leurs besoins de calcul.

Service de Calcule GCP Compute Engine - Atelier - 1

- Création d'une machine virtuelle sur gcp.
- La connexion à la machine virtuelle.
- Mise à jour et vérfication d'installation de Python par exemple.
- Copie d'un projet python vers la machine virtuelle.
- Execution du projet sur la machine virtuelle.

TP : Analyse de Données sur Google Compute Engine

Objectif du TP: Familiarisez-vous avec Google Compute Engine (GCE) en configurant une machine virtuelle (VM) pour analyser un jeu de données spécifique. Vous préparerez l'environnement python, chargerez le jeu de données, réaliserez une analyse exploratoire des données (EDA) et utiliserez des outils de visualisation pour explorer les relations dans les données avec python.

- **Étapes du TP**

1. **Introduction à Google Cloud Platform et Compute Engine**
   - Découvrez les principes de base et les cas d'utilisation de Compute Engine.
2. **Création d'une VM sur Compute Engine**
   - Créez une nouvelle VM avec une configuration appropriée via la console GCP.
3. **Configuration de l'Environnement de Data Science sur la VM**
   - Configurez votre VM pour l'analyse de données.

TP : Analyse de Données sur Google Compute Engine

4. **Chargement et Exploration d'un Jeu de Données**
   - Chargez le jeu de données Iris dans votre VM.
   - Explorez ses caractéristiques principales.
5. **Analyse Exploratoire des Données (EDA)**
   - Réalisez une analyse exploratoire pour comprendre les données.
6. **Visualisation des Données**
   - Utilisez des outils de visualisation pour examiner les relations dans les données.
7. **Conclusion et Nettoyage**
   - Apprenez comment arrêter et nettoyer les ressources utilisées.

Pour cet atelier, vous utiliserez le jeu de données Iris, disponible via le lien suivant :

- **Iris Dataset**: Kaggle Dataset Link

GCP Cloud function

Google Cloud Functions est un service de calcul sans serveur qui vous permet d'exécuter votre code en réponse à des événements sans nécessiter de provisionnement préalable ou de gestion d'infrastructure. Ce modèle de calcul est idéal pour des cas d'usage où vous avez besoin de traiter des événements en temps réel, comme l'analyse de logs, le traitement de données en streaming, ou l'intégration et l'orchestration de services cloud.

- **Sans Serveur** : Cloud Functions supprime la nécessité de gérer des serveurs. Google gère l'infrastructure, y compris la gestion des ressources, la maintenance, et la mise à l'échelle automatique.
- **Événementiel** : Il s'exécute en réponse à des événements provenant de votre infrastructure cloud ou de services tiers. Les événements peuvent être déclenchés par des modifications dans Google Cloud Storage, des messages sur Google Pub/Sub, des requêtes HTTP via Google Firebase, et plus encore.
- **Langages de Programmation** : Supporte plusieurs langages de programmation, notamment Node.js, Python, Go, Java, et .NET, ce qui vous permet d'utiliser le langage avec lequel vous êtes le plus à l'aise.
- **Intégration avec GCP** : Intègre étroitement avec d'autres services Google Cloud, permettant des workflows complexes où les fonctions peuvent interagir avec des bases de données, des services d'analyse, et d'autres outils cloud.
- **Facturation à l'Utilisation** : Vous payez uniquement pour le temps d'exécution de votre fonction, mesuré en gigahertz-secondes, ce qui rend Cloud Functions économiquement attractif pour les tâches intermittentes et à volume variable.

GCP Cloud function Cas d'Usage

- **Traitement de Données** : Exécuter des tâches de traitement de données en réponse à des changements dans Google Cloud Storage ou à des messages reçus via Google Pub/Sub.
- **Intégrations d'API** : Créer des microservices qui répondent aux requêtes HTTP, permettant de construire des API RESTful sans gérer d'infrastructure.
- **Automatisation d'Infrastructure** : Utiliser Cloud Functions pour automatiser des réponses aux événements d'infrastructure, comme la mise à jour de configurations ou la gestion de ressources.
- **Traitement d'Images ou de Vidéos** : Exécuter des fonctions pour traiter ou analyser des images et des vidéos dès qu'elles sont téléchargées dans un bucket Cloud Storage.
- **Machine Learning et IA** : Intégrer avec AI Platform pour exécuter des modèles de machine learning en réponse à des événements, permettant des applications d'IA dynamiques.

GCP Cloud function Avantages

- **Développement Agile** : Permet un développement rapide et agile, avec la possibilité de déployer des morceaux de logique métier indépendamment.
- **Scalabilité Automatique** : Gère automatiquement la mise à l'échelle des fonctions pour répondre aux demandes, sans intervention manuelle.
- **Pas de Gestion d'Infrastructure** : Élimine la nécessité de gérer des serveurs ou des environnements d'exécution, vous permettant de vous concentrer sur le code et la logique métier.

TP : Création d'une Google Cloud Function avec Python pour une Analyse de Données Simple

- **Objectif du TP**: L'objectif de ce TP est de créer une Google Cloud Function en Python qui exécute une analyse de données simple en réponse à une requête HTTP. Ce TP est conçu pour les participants d'une formation professionnelle en data sans prérequis de connaissances sur les services de stockage et de base de données de GCP.
- **Description du TP**: Dans ce TP, vous développerez une fonction qui, lorsqu'elle est déclenchée par une requête HTTP, génère des statistiques descriptives de base à partir d'un jeu de données intégré dans le code. Vous utiliserez Python et des bibliothèques de manipulation de données telles que Pandas pour effectuer l'analyse.
- **Étapes du TP**

1. **Création de la Cloud Function**
   - Accédez à la console Google Cloud Platform.
   - Naviguez jusqu'à la section Cloud Functions et créez une nouvelle fonction.
   - Définissez un nom pour votre fonction et choisissez l'environnement d'exécution Python.
   - Dans le champ du déclencheur, sélectionnez HTTP pour que la fonction puisse être invoquée via des requêtes HTTP.

TP : Création d'une Google Cloud Function avec Python pour une Analyse de Données Simple

2. **Développement de la Fonction en Python**
   - Utilisez l'éditeur en ligne ou votre IDE local pour développer le code de la fonction.
   - Écrivez une fonction qui charge un petit jeu de données codé en dur (par exemple, un DataFrame Pandas avec des données d'exemple).
   - Implémentez la logique pour calculer des statistiques descriptives de base sur le jeu de données (moyenne, médiane, écart-type).
   - Configurez la fonction pour qu'elle retourne les résultats des statistiques calculées en réponse à la requête HTTP.
3. **Déploiement de la Fonction**
   - Déployez votre fonction Cloud à partir de la console Google Cloud ou en utilisant le Google Cloud SDK.
   - Une fois le déploiement terminé, un point de terminaison HTTP sera généré.
4. **Test de la Fonction**
   - Testez votre Cloud Function en envoyant une requête HTTP au point de terminaison généré.
   - Vérifiez que la fonction retourne les statistiques descriptives attendues du jeu de données.

GCP - Service de stockage

Les services de stockage de Google Cloud Platform (GCP) offrent une gamme variée d'options pour stocker vos données de manière sécurisée et accessible. Ces services sont conçus pour répondre à différents besoins, allant du stockage d'objets et de fichiers au stockage de blocs

- Google Cloud Storage (GCS)
- Persistent Disk
- Filestore

GCP - Google Cloud Storage (GCS)

C'est le service de stockage d'objets de GCP, conçu pour stocker de grandes quantités de données non structurées. Il est hautement scalable, durable et sécurisé. GCS est idéal pour stocker des images, des vidéos, des fichiers de sauvegarde, et des données de sites web. Il offre plusieurs classes de stockage pour optimiser les coûts et les performances, selon la fréquence d'accès aux données.

- **Scalabilité :** GCS peut stocker des données allant de quelques octets à plusieurs exaoctets, ce qui le rend adapté à presque toutes les applications de stockage de données.
- **Durabilité et Disponibilité :** Les données stockées dans GCS sont automatiquement répliquées dans plusieurs emplacements et offrent une garantie de durabilité de 99.999999999% (11 neuf) pour assurer la protection contre la perte de données.
- **Sécurité :** GCS fournit de robustes contrôles d'accès et des capacités de chiffrement pour sécuriser vos données. Il supporte le chiffrement des données au repos et en transit.
- **Performance :** Le service offre une performance élevée pour le téléchargement et l'accès aux données grâce à une infrastructure globale optimisée.

GCP - Google Cloud Storage (GCS)

Fonctionnalités

- **Classes de Stockage :** GCS propose plusieurs classes de stockage pour répondre à différents besoins en termes de coût et d'accès aux données, incluant Standard, Nearline, Coldline, et Archive.
- **Gestion du Cycle de Vie :** Permet de configurer des règles pour réduire automatiquement les coûts en déplaçant ou supprimant des données en fonction de l'âge, de la classe de stockage, ou d'autres conditions.
- **Uniformité des Données Forte :** GCS garantit une cohérence forte, ce qui signifie que toute lecture après une écriture réussie renvoie immédiatement les données écrites.
- **Intégration avec d'autres services GCP :** GCS s'intègre étroitement avec d'autres services Google Cloud, tels que BigQuery, pour l'analyse de données, et Cloud Machine Learning Engine pour l'entraînement de modèles d'apprentissage automatique.
- **Contrôles d'accès fins :** Supporte la gestion fine des permissions à l'aide des rôles IAM et des politiques de bucket, permettant un contrôle précis sur qui peut accéder à vos données.
- **Transfert de Données :** Offre des outils et services pour faciliter l'importation de grandes quantités de données vers GCS, y compris le service de transfert de données et l'outil de ligne de commande `gsutil`.

GCP - Google Cloud Storage (GCS)

Utilisation

GCS est largement utilisé pour une variété d'applications, telles que :

- **Hébergement de sites web statiques** : Stocker et servir des contenus web statiques comme des images, des feuilles de style CSS et des fichiers JavaScript.
- **Archives et sauvegardes** : Archiver des données à long terme et effectuer des sauvegardes régulières des systèmes et bases de données.
- **Contenus multimédia** : Stocker, traiter et diffuser des contenus multimédia, comme des vidéos et des images, à grande échelle.
- **Analyse de données** : Servir de référentiel pour les données analysées par des services d'analyse de données comme Google BigQuery.

GCP - Google Cloud Storage (GCS) - Atelier

Objectif : Apprendre à interagir avec Google Cloud Storage (GCS) pour le stockage et la manipulation de données en utilisant Python, en se concentrant sur les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer) et sur la gestion des permissions d'accès à partir d'une instance de VM sur Google Cloud Platform (GCP).

Étapes de l'Atelier :

1. **Configuration de l'Environnement :**
   - Configurer une instance de VM sur GCP avec le rôle IAM approprié pour accéder à GCS.
   - Installer Python et les bibliothèques nécessaires sur la VM.
2. **Introduction à GCS :**
   - Présentation de Google Cloud Storage, des classes de stockage, et des cas d'usage.
   - Création et configuration d'un bucket GCS pour l'atelier.

GCP - Google Cloud Storage (GCS) - Atelier

3. **Scripts Python pour GCS :**
   - **Script 1 : Téléchargement et Upload de Fichiers**
     - Écrire un script Python pour télécharger un fichier depuis le bucket GCS vers la VM, et pour uploader un fichier de la VM vers le bucket GCS.
   - **Script 2 : Listing des Fichiers**
     - Écrire un script Python qui liste tous les fichiers présents dans un bucket GCS spécifique.
   - **Script 3 : Suppression de Fichiers**
     - Écrire un script Python pour supprimer un fichier spécifique dans un bucket GCS.
4. **Gestion des Permissions :**
   - Apprendre à attribuer des rôles IAM à des comptes de service pour contrôler l'accès à GCS.
   - Modifier le rôle d'une VM pour limiter ou étendre ses capacités d'accès à GCS.

GCP - Google Cloud Storage (GCS) - TP

Objectif du TP

Développer une Google Cloud Function qui :

- Se déclenche à chaque fois qu'un nouveau fichier est ajouté à un bucket GCS spécifié.
- Lit le contenu du nouveau fichier.
- Agrège ce contenu à un fichier existant (ou en crée un nouveau s'il n'existe pas) dans le même bucket.

GCP - Persistent Disk

Persistent Disk est le service de stockage en bloc offert par Google Cloud Platform (GCP) qui permet de créer et d'utiliser des disques virtuels avec les instances Compute Engine et Google Kubernetes Engine. Les disques persistants sont conçus pour offrir une performance élevée, une durabilité exceptionnelle, et une intégration transparente avec les instances virtuelles. Voici une vue d'ensemble plus détaillée de Persistent Disk, ainsi que des conseils sur son utilisation pour les projets de données.

Caractéristiques de Persistent Disk

- **Performance:** Les disques persistants offrent une latence faible et un haut débit, avec des options pour des performances SSD (Solid-State Drive) ou HDD (Hard Disk Drive) selon les besoins en performances et en coût.
- **Durabilité:** Les données sur un disque persistant sont répliquées automatiquement pour assurer la durabilité et la disponibilité. Même en cas de défaillance d'une instance, les données restent accessibles et intègres.
- **Flexibilité:** Les tailles de disque peuvent être ajustées à la volée, et les performances peuvent être configurées indépendamment de la capacité de stockage, permettant une personnalisation selon les besoins spécifiques du projet.
- **Chiffrement:** Toutes les données stockées sur des disques persistants sont chiffrées par défaut, assurant la sécurité et la confidentialité des données.

GCP - Persistent Disk

Utilisation de Persistent Disk pour les Projets de Données

1. **Stockage de Bases de Données:** Les disques persistants sont idéaux pour stocker des bases de données relationnelles ou non relationnelles, offrant à la fois la performance requise pour des opérations d'E/S (entrée/sortie) intensives et la durabilité nécessaire pour la conservation à long terme des données.
2. **Systèmes de Fichiers Partagés:** Pour les applications nécessitant un accès simultané à des fichiers par plusieurs instances, un disque persistant peut être attaché en mode lecture seule à plusieurs instances, ou vous pouvez utiliser Filestore, le service de stockage de fichiers entièrement géré de GCP, pour un système de fichiers partagés compatible POSIX.
3. **Disques de Démarrage:** Les disques persistants peuvent servir de disques de démarrage pour les instances Compute Engine, stockant le système d'exploitation et les applications. Cela permet de séparer les données applicatives du stockage de l'OS pour une meilleure gestion et sécurité.
4. **Sauvegardes et Snapshots:** GCP permet de créer facilement des snapshots de disques persistants, facilitant la mise en œuvre de stratégies de sauvegarde et de reprise après sinistre. Les snapshots peuvent être utilisés pour restaurer des données ou pour créer de nouveaux disques.

GCP - Persistent Disk

Exemple d'Utilisation

Imaginons que vous avez une application de traitement de données qui nécessite une base de données PostgreSQL hautement disponible et performante. Vous pouvez déployer cette base de données sur une instance Compute Engine en utilisant un disque persistant SSD pour stocker les fichiers de données de la base de données. Grâce aux capacités de redimensionnement et aux performances ajustables du disque persistant, vous pouvez facilement adapter les ressources à la demande de votre application, garantissant ainsi que les opérations de lecture et d'écriture de la base de données sont rapides et efficaces.

GCP - Persistent Disk - Atelier

Pour utiliser un Persistent Disk dans votre projet de données :

1. **Créez une instance Compute Engine** via Google Cloud Console, gcloud CLI ou l'API Compute Engine.
2. **Attachez un disque persistant** à l'instance lors de sa création ou ajoutez-le à une instance existante. Vous pouvez spécifier la taille et le type de disque (SSD ou HDD) selon vos besoins.
3. **Formatez et montez le disque** dans votre instance pour l'utiliser comme stockage de données ou comme disque de démarrage.

GCP - Filestore

Cloud Filestore est un service de stockage de fichiers entièrement géré pour les applications qui nécessitent un système de fichiers partagés. Il est compatible POSIX, ce qui le rend adapté pour des applications et des charges de travail exigeant un accès standard aux fichiers.

Caractéristiques Clés

- **Performance :** Filestore offre des performances élevées en termes d'IOPS et de débit, avec des options pour des performances standard ou élevées pour répondre aux besoins spécifiques de votre application.
- **Compatibilité POSIX :** Comme il est basé sur un système de fichiers standard, il est facilement compatible avec les applications existantes qui s'attendent à un système de fichiers POSIX.
- **Scalabilité :** Bien que pas aussi dynamiquement scalable que d'autres services de stockage sur le cloud, il offre tout de même des capacités de montée en charge prévisibles et la possibilité d'ajuster les ressources en fonction des besoins.
- **Intégration avec Google Cloud :** Filestore s'intègre avec d'autres services GCP, comme Google Kubernetes Engine (GKE) et Compute Engine, permettant un partage de fichiers efficace entre différentes instances et applications.

GCP - Filestore

Utilisations pour les Données

- **Applications Héritées :** Idéal pour migrer des applications héritées vers le cloud sans modifier le système de fichiers attendu par ces applications.
- **Stockage et Partage de Fichiers :** Filestore sert de solution pour les scénarios nécessitant le partage de fichiers entre plusieurs instances ou services, comme dans les environnements de développement logiciel collaboratif.
- **Applications Multimédia :** Pour les applications nécessitant un accès en lecture/écriture rapide à de grands fichiers multimédia, Filestore fournit la performance nécessaire.
- **Données de Simulation et de Modélisation :** Pour les charges de travail scientifiques ou d'ingénierie nécessitant un stockage partagé pour les données de simulation volumineuses.

GCP - Cloud SQL

Cloud SQL est un service de base de données entièrement géré qui facilite la création, la maintenance, la gestion et l'administration des bases de données relationnelles sur le cloud.

1. **Gestion entièrement gérée** : Cloud SQL offre une gestion de base de données simplifiée, en automatisant les tâches courantes telles que la mise en place, la maintenance, les sauvegardes, et la reprise après sinistre. Cela permet aux utilisateurs de se concentrer davantage sur le développement d'applications plutôt que sur la gestion de l'infrastructure de base de données.
2. **Performance et évolutivité** : Le service fournit des capacités de performance et d'évolutivité qui permettent aux utilisateurs d'ajuster les ressources (CPU, mémoire, stockage) selon les besoins de leur application, offrant ainsi la flexibilité pour gérer la croissance et les pics de demande.
3. **Sécurité** : Cloud SQL intègre plusieurs fonctionnalités de sécurité, y compris le chiffrement des données en repos et en transit, ainsi que des options pour la configuration du réseau privé et le contrôle d'accès basé sur les rôles, pour aider à protéger les données sensibles.

GCP - Cloud SQL

4. **Compatibilité** : Le service prend en charge plusieurs systèmes de gestion de bases de données relationnelles populaires, notamment MySQL, PostgreSQL et SQL Server. Cela permet une intégration facile avec des applications existantes et facilite la migration des bases de données vers le cloud.
5. **Intégration avec GCP** : Cloud SQL s'intègre étroitement avec d'autres services de Google Cloud, permettant aux développeurs de créer des solutions complètes qui tirent parti des capacités du cloud, telles que l'analyse de données, le machine learning, et l'informatique sans serveur.
6. **Accessibilité** : En tant que service cloud, Cloud SQL est accessible de partout, permettant aux équipes distribuées de travailler ensemble efficacement, et aux applications d'accéder aux bases de données depuis n'importe quel endroit, sous réserve des politiques de sécurité configurées.
7. **Modèle de facturation à l'utilisation** : Comme beaucoup d'autres services cloud, Cloud SQL adopte un modèle de facturation basé sur l'utilisation, ce qui signifie que les coûts sont associés aux ressources consommées, offrant ainsi une certaine flexibilité pour gérer les coûts en fonction des besoins réels.

GCP - Cloud SQL - Atelier

Objectifs de l'Atelier

- Comprendre les principes fondamentaux de Cloud SQL et son intégration dans l'écosystème cloud.
- Apprendre à provisionner et configurer une instance Cloud SQL.
- Découvrir comment connecter une base de données Cloud SQL à une application Python.
- S'exercer à effectuer des opérations CRUD (Create, Read, Update, Delete) et à exécuter des requêtes SQL depuis Python.
- Explorer les meilleures pratiques pour la gestion des données et la sécurité.

Etape 1 : Configuration de Cloud SQL

- Création d'une instance Cloud SQL (MySQL/PostgreSQL).
- Configuration des paramètres de l'instance (région, version du SGBD, etc.).
- Sécurité et accès réseau : définir les règles de pare-feu et les configurations d'accès.

Etape 2 : Connexion à Cloud SQL depuis Python

- Installation des bibliothèques nécessaires (`pymysql` pour MySQL, `psycopg2` pour PostgreSQL, etc.).
- Établissement d'une connexion sécurisée à l'instance Cloud SQL.
- Discussion sur l'importance de gérer les informations d'identification de manière sécurisée.

Etape 3 : Manipulation de Données avec Python

GCP - Cloud SQL - TP

Sujet de TP : Automatisation de l'Insertion de Données dans Cloud SQL via Cloud Functions et Google Cloud Storage

Objectif : Le but de ce TP est de développer une solution automatisée pour l'insertion de données dans une table MySQL hébergée sur Cloud SQL chaque fois qu'un nouveau fichier est ajouté à un bucket spécifique dans Google Cloud Storage (GCS). Cette solution implique l'utilisation de Google Cloud Functions pour déclencher un processus d'insertion de données à partir des informations contenues dans le fichier nouvellement ajouté.

Contexte

Dans de nombreux scénarios de traitement de données, les organisations reçoivent régulièrement des fichiers de données (par exemple, des fichiers CSV) qui doivent être importés dans des bases de données relationnelles pour un traitement ultérieur, une analyse ou une visualisation. L'automatisation de ce processus réduit les erreurs manuelles et améliore l'efficacité opérationnelle.

GCP - Cloud SQL - TP

Instructions

1. **Préparation de l'Environnement :**
   - Créez un bucket Google Cloud Storage pour stocker les fichiers de données.
   - Configurez une instance Cloud SQL avec une base de données MySQL. Assurez-vous que la base de données contient une table cible pour les données à insérer.
   - Activez les API Cloud Functions et Cloud SQL Admin si elles ne sont pas déjà activées.
2. **Création de la Table Cloud SQL :**
   - Définissez une table dans votre base de données MySQL sur Cloud SQL. Par exemple, une table `ventes` pour stocker des informations de transactions de vente.
   - Assurez-vous que la structure de la table correspond aux données attendues dans les fichiers CSV.

GCP - Cloud SQL - TP

3. **Développement de la Cloud Function :**
   - Écrivez une fonction Python qui se déclenche sur l'événement `google.storage.object.finalize` indiquant l'ajout d'un nouveau fichier dans le bucket GCS.
   - La fonction doit lire le contenu du fichier CSV, parser les données, et les insérer dans la table Cloud SQL précédemment créée.
   - Gérez les erreurs et les exceptions pour assurer une insertion réussie des données.
4. **Déploiement de la Cloud Function :**
   - Utilisez le CLI de Google Cloud ou la console pour déployer votre Cloud Function. Configurez-le pour qu'il écoute les événements de création de fichiers dans le bucket GCS spécifié.
5. **Test de la Solution :**
   - Téléchargez un fichier CSV de test dans le bucket GCS et vérifiez si les données sont correctement insérées dans la table Cloud SQL.
   - Vérifiez les logs de la Cloud Function pour tout débogage nécessaire.

GCP - Cloud Spanner

Cloud Spanner est un service de base de données entièrement géré offert par Google Cloud Platform (GCP) qui combine les avantages des bases de données relationnelles SQL traditionnelles avec ceux des systèmes de bases de données NoSQL, pour offrir une solution évolutive, robuste et globale. Cela signifie que Cloud Spanner peut gérer de très grands volumes de données tout en fournissant une forte cohérence des transactions et une haute disponibilité à travers le monde.

1. **Haute disponibilité et durabilité** : Cloud Spanner assure une disponibilité de 99,999% dans les configurations multi-régionales, ce qui le rend extrêmement fiable pour les applications critiques. Les données sont automatiquement répliquées dans plusieurs zones géographiques pour garantir la durabilité et la résilience face aux défaillances régionales.
2. **Echelle globale** : Il est conçu pour l'échelle globale, permettant aux applications de grandir sans les contraintes de capacité typiques des systèmes de gestion de bases de données traditionnels. Cela est particulièrement utile pour les entreprises qui ont besoin de gérer des bases de données volumineuses réparties à travers le monde.
3. **Cohérence forte** : Contrairement à beaucoup d'autres bases de données distribuées qui offrent une cohérence éventuelle, Cloud Spanner maintient une cohérence forte des transactions à travers toutes ses réplications, garantissant que toutes les lectures reçoivent la version la plus récente des données après une transaction commitée.

GCP - Cloud Spanner

4. **Compatibilité SQL** : Cloud Spanner supporte le langage SQL pour les requêtes, rendant facile pour les développeurs familiers avec les bases de données relationnelles de migrer leurs applications ou d'utiliser leurs compétences existantes.
5. **Gestion entièrement managée** : En tant que service entièrement géré, Google s'occupe de la maintenance, des mises à jour et de la gestion de l'infrastructure, permettant aux développeurs de se concentrer sur la création et l'optimisation de leurs applications plutôt que sur la gestion de la base de données.

GCP - FireStore

Firestore est une base de données NoSQL flexible et évolutive pour le développement d'applications mobiles, web et serveur, offerte par Google Cloud Platform (GCP). Elle est conçue pour stocker et synchroniser des données entre vos utilisateurs en temps réel. Firestore est la nouvelle version de Firebase Database et offre un modèle de données plus riche, des requêtes plus puissantes, et une structure de données basée sur des documents et des collections.

1. **Modèle de Données Basé sur des Documents** : Les données sont stockées dans des documents, qui sont ensuite organisés en collections. Les documents peuvent contenir plusieurs types de données (comme des strings, des nombres, et des objets complexes) et peuvent également contenir des sous-collections.
2. **Synchronisation en Temps Réel** : Firestore permet de synchroniser les données entre les appareils des utilisateurs en temps réel, facilitant la création d'expériences utilisateur interactives et collaboratives.
3. **Requêtes Puissantes** : Firestore fournit des capacités de requêtes avancées, vous permettant de récupérer des documents basés sur plusieurs critères, d'effectuer des triages et des filtrages complexes.
4. **Sécurité au Niveau des Données** : Avec Firestore, vous pouvez définir des règles de sécurité pour contrôler l'accès aux données au niveau du document ou de la collection.
5. **Intégration avec Google Cloud** : Firestore est intégré à d'autres services Google Cloud, permettant un développement de back-end complet et flexible.

GCP - FireStore - Atelier

Prérequis

- Avoir un compte Google Cloud Platform (GCP).
- Installer Python et `pip` sur votre machine.
- Installer la bibliothèque `google-cloud-firestore` en exécutant `pip install google-cloud-firestore`.

1. **Configurer Firestore**
    1. **Créez un projet sur Google Cloud Platform** si vous n'en avez pas déjà un.
    2. **Allez dans Firestore** dans la console GCP et créez une nouvelle base de données Firestore en mode de verrouillage ou en mode natif, selon vos préférences.

GCP - FireStore - Atelier

2. **Créer un Document**

Ce code Python crée un nouveau contact dans la collection `contacts`.

```python
from google.cloud import firestore

# Initialiser le client Firestore
db = firestore.Client()

def add_contact(name, email, phone):
    contacts_ref = db.collection('contacts')
    contacts_ref.add({
        'name': name,
        'email': email,
        'phone': phone
    })
    print('Contact ajouté avec succès.')
```

GCP - FireStore - Atelier

3. Lire des Documents

Cette fonction récupère tous les contacts de la collection `contacts` et les affiche.

```
def list_contacts():
    contacts_ref = db.collection('contacts')
    for contact in contacts_ref.stream():
        print(f"{contact.id} => {contact.to_dict()}")
```

4. Mettre à Jour un Document

Mettez à jour un contact spécifique en modifiant son numéro de téléphone, par exemple.

```
def update_contact(contact_id, new_phone):
    contact_ref = db.collection('contacts').document(contact_id)
    contact_ref.update({'phone': new_phone})
    print('Contact mis à jour.')
```

5. Supprimer un Document

Supprimez un contact spécifique.

```
def delete_contact(contact_id):
    db.collection('contacts').document(contact_id).delete()
```

GCP - FireStore - Atelier

6. **Recherche dans les Documents**

Recherchez des contacts par nom.

```
def search_contacts_by_name(search_name):
    contacts_ref = db.collection('contacts').where('name', '==', search_name)
    for contact in contacts_ref.stream():
        print(f"{contact.id} => {contact.to_dict()}")
```

GCP - FireStore - vs DataStore

- **Google Cloud Datastore**

1. **Modèle de Données**: Datastore est également une base de données NoSQL, mais elle est basée sur un modèle d'entité-attribut. Les données sont organisées en entités, qui sont des ensembles d'attributs (ou de paires clé-valeur).
2. **Consistance des Données**: Datastore offre des options de consistance des données plus flexibles, y compris la consistance forte pour les lectures et requêtes dans un même groupe d'entités.
3. **Indexation Automatique**: Datastore indexe automatiquement toutes les propriétés des entités, ce qui facilite la mise en place de requêtes sans avoir à gérer manuellement les index.
4. **Scalabilité**: Tout comme Firestore, Datastore est conçu pour l'échelle automatique et peut gérer des applications à fort trafic, mais il est principalement optimisé pour les applications nécessitant des transactions complexes et une consistance forte à l'échelle mondiale.
5. **Migration vers Firestore**: Google recommande Firestore comme la base de données de choix pour les nouveaux projets en raison de sa flexibilité, de ses capacités en temps réel et de son intégration avec Firebase. Datastore mode est maintenant considéré comme un mode de compatibilité dans Firestore, permettant aux utilisateurs existants de Datastore de bénéficier des nouvelles fonctionnalités de Firestore sans avoir à migrer leurs données manuellement.

Cloud dataflow

Cloud Dataflow est un service entièrement géré offert par Google Cloud Platform qui est conçu pour simplifier le processus de développement, de déploiement et de gestion d'applications de traitement de données en temps réel et par lots. Basé sur le modèle de programmation Apache Beam, Cloud Dataflow permet aux développeurs de créer des pipelines de données complexes qui peuvent traiter des volumes massifs de données de manière efficace et scalable. Voici une explication détaillée et quelques exemples d'utilisation.

- **Apache Beam**: Un modèle de programmation open-source qui fournit des abstractions pour définir et exécuter des pipelines de traitement de données. Cloud Dataflow utilise Apache Beam comme base, ce qui permet une portabilité des pipelines entre différentes plateformes de traitement de données.
- **Pipelines de Données**: Les applications développées avec Cloud Dataflow sont organisées en pipelines de données, qui sont composés de séries de transformations de données. Ces pipelines peuvent ingérer des données de diverses sources, les transformer et les envoyer vers différentes destinations (bases de données, systèmes de fichiers, services de messagerie, etc.).
- **Traitement en Temps Réel et par Lots**: Cloud Dataflow peut traiter des données en temps réel (streaming) ainsi que des ensembles de données stockées (batch), offrant une flexibilité pour répondre à divers cas d'utilisation.

Cloud dataflow - Exemples d'Utilisation

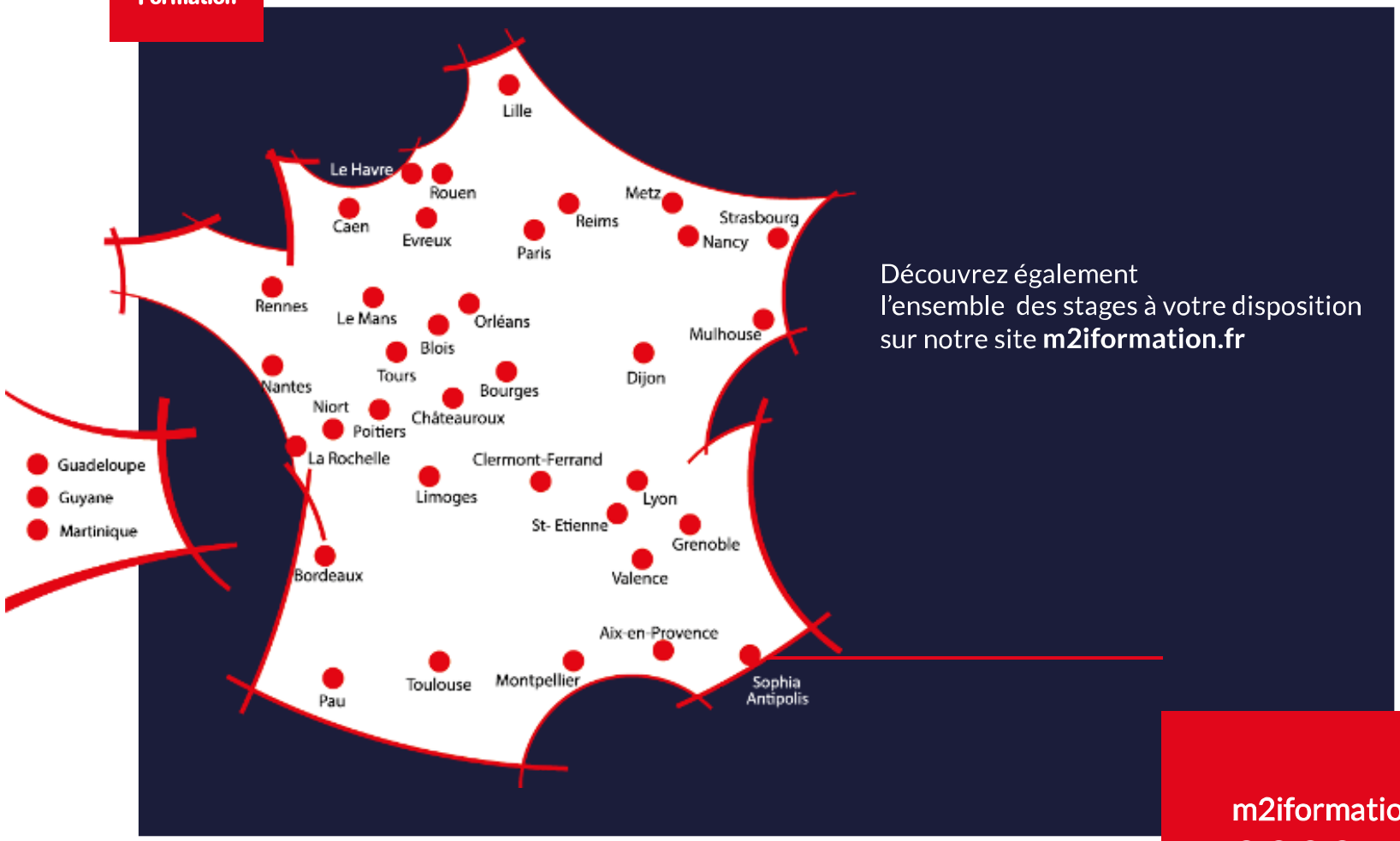
1. **Analyse en Temps Réel des Logs**: Imaginez une application web à fort trafic où il est crucial de surveiller et d'analyser les logs en temps réel pour détecter rapidement les problèmes de performance ou de sécurité. Avec Cloud Dataflow, vous pouvez construire un pipeline qui ingère les logs en continu, les analyse pour identifier les tendances ou les anomalies, et déclenche des alertes ou des actions automatiques en fonction des résultats.
2. **Traitement par Lots de Données Historiques**: Une entreprise pourrait avoir des téraoctets de données historiques stockées dans un lac de données. Pour générer des insights à partir de ces données, un pipeline Cloud Dataflow peut être configuré pour les lire, les nettoyer, les transformer (par exemple, agrégation, filtrage, enrichissement avec d'autres sources de données) et les stocker dans un format optimisé pour l'analyse et le reporting.
3. **Intégration de Données en Temps Réel**: Dans un scénario où différentes applications et systèmes doivent synchroniser leurs données en temps réel (par exemple, inventaires de commerce électronique, bases de données de clients), Cloud Dataflow peut faciliter l'intégration en temps réel en établissant des pipelines qui extraient, transforment et chargent les données de manière fluide et efficace entre les systèmes.
4. **Machine Learning sur des Flux de Données**: Pour les applications nécessitant des modèles de machine learning qui s'adaptent continuellement à partir de données en streaming (comme la recommandation de produits ou la détection de fraude), Cloud Dataflow peut être utilisé pour prétraiter les données en streaming (nettoyage, normalisation, extraction de caractéristiques) avant de les envoyer à des modèles de machine learning pour l'inférence en temps réel.

Cloud Dataproc

- **Objectif principal**: Service de traitement de données basé sur Hadoop et Spark, optimisé pour le cloud. Il est conçu pour exécuter des tâches de traitement de données par lots, des requêtes interactives et du traitement de données en streaming.
- **Modèle de programmation**: Supporte les écosystèmes Hadoop et Spark, offrant une transition facile pour les applications existantes basées sur ces technologies vers le cloud.
- **Gestion de l'infrastructure**: Bien qu'il soit un service géré, les utilisateurs ont plus de contrôle sur la configuration de l'environnement de cluster par rapport à Dataflow. Dataproc permet de créer, redimensionner et supprimer des clusters Hadoop/Spark rapidement.
- **Cas d'utilisation**: Convient aux entreprises qui utilisent déjà Hadoop/Spark et qui cherchent à migrer leurs workloads dans le cloud avec une gestion simplifiée et une scalabilité. Idéal pour le traitement par lots, l'analyse de données interactive et le traitement de données en streaming utilisant l'écosystème Hadoop/Spark.

Cloud Composer

- **Objectif principal**: Un service d'orchestration de workflows entièrement géré, basé sur Apache Airflow. Il permet de créer, planifier, surveiller et gérer des workflows complexes qui traversent plusieurs services dans Google Cloud et en dehors.
- **Modèle de programmation**: Utilise Apache Airflow, fournissant une interface riche pour la création de workflows à l'aide de Python, permettant une intégration facile avec de nombreux services et API.
- **Gestion de l'infrastructure**: En tant que service géré, Cloud Composer s'occupe de la configuration, de la maintenance et de l'évolutivité de l'environnement Airflow, tout en offrant une personnalisation et un contrôle sur les workflows.
- **Cas d'utilisation**: Idéal pour l'orchestration de workflows de traitement de données complexes, l'intégration de systèmes disparates, et l'automatisation de pipelines ETL (Extract, Transform, Load) à travers plusieurs services cloud et on-premise.



Découvrez également
l'ensemble des stages à votre disposition
sur notre site **m2iformation.fr**

m2iformation.fr