



Documentation
Technique



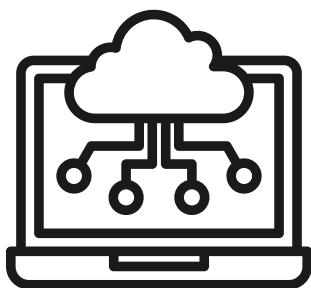
Utilisation de Google Big Query pour faire des rapports BI en temps réel



BDCC

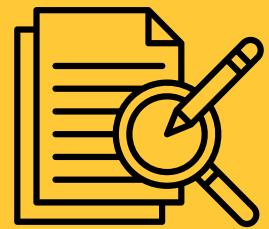
Rapport en temps réel sur le bitcoin cash

Dec 16, 2025



Google BigQuery permet d'analyser instantanément des flux de données massifs pour piloter l'entreprise avec des indicateurs en direct plutôt qu'en différé.

I- Contexte et Objectifs



L'urgence des cryptomonnaies

Le marché de la cryptomonnaie est un environnement caractérisé par sa vitesse et sa volatilité extrêmes. Les actifs comme le Bitcoin Cash (BCH) peuvent subir des variations de valeur majeures en quelques jours voire quelques minutes.

Pour les investisseurs qui cherchent à réaliser des profits, la clé du succès réside dans leur capacité à anticiper ces mouvements et à acheter ou vendre au moment précis. Cependant, se fier aux données de prix publiques traditionnelles rend souvent les décisions tardives et, par conséquent, moins rentables. Le temps de latence est l'ennemi de la performance.

ChainSight Solutions et la BI en temps réel

Qu'est ce que Chainsight ?

Une équipe d'analystes spécialisés accompagne les investisseurs en transformant des volumes massifs de données complexes en solutions stratégiques.

BI en temps réel

Nous proposons une plateforme de Business Intelligence permettant de suivre instantanément les évolutions du marché pour une réactivité maximale.

Analyse On-Chain

Notre approche se fonde sur l'étude rigoureuse des flux de transactions et de liquidités directement sur la blockchain, l'unique source de vérité sur les tendances du marché.

La solution et les bénéfices clients

0
Seconde de latence
100%
de transparence
24/7
de Monitoring



"En Dieu nous croyons ; pour tout le reste, apportez des données."

- W. Edwards Deming



Bénéfices clients

Maximisation du profit :

Savoir quand exactement acheter ou vendre du Bitcoin Cash en se basant sur des données fondamentales

Anticipation des tendances :

Identifier les signaux faibles et les mouvements de "Whales" (gros porteurs) avant que ces actions n'impactent publiquement les prix.

Prise de décision éclairée :

Disposer d'un rapport dynamique mis à jour en continu, permettant d'agir sans délai et de prendre des décisions éclairées.

Gestion du Risque :

Être alerté immédiatement (alarmes) lorsque des seuils critiques (comme une sortie massive de liquidité) sont atteints

Problématique du projet

La validation de notre solution passe par la réponse technique et métier à la question suivante

Comment l'analyse en temps réel des mouvements de liquidité et de l'activité transactionnelle sur la blockchain Bitcoin Cash peut-elle permettre d'anticiper les tendances de marché et d'optimiser les stratégies ?



Glossaire des termes clés

Cette section définit les concepts techniques fondamentaux qui soutiennent et structurent l'architecture de ce projet.

Terme clé	Définition précise et rôle dans le projet
Business Intelligence (BI)	Processus de collecte, d'intégration, d'analyse et de présentation des données brutes en informations exploitables (tableaux de bord) pour soutenir la décision stratégique.
Big Data	Domaine concernant le traitement d'ensembles de données massifs caractérisés par le Volume, la Vélocité (vitesse de création) et la Variété.
Analyse On-Chain	Méthode d'analyse qui étudie les métriques extraites directement de la blockchain (transactions, volumes, adresses) pour évaluer l'activité fondamentale du réseau.
Data Warehouse	Système d'entreposage de données optimisé pour les requêtes analytiques complexes sur de grands ensembles de données, tel que Google BigQuery.
Serverless	Modèle de Cloud Computing où la gestion de l'infrastructure serveur est entièrement déléguée au fournisseur cloud, permettant de se concentrer sur l'optimisation des requêtes et de la logique métier.
KPI (Key Performance Indicator)	Métrique quantifiable utilisée pour évaluer l'efficacité de l'activité du marché (ex: vitesse de circulation).
Pipeline	Le flux de travail séquentiel par lequel les données sont acheminées, transformées et chargées depuis leur source (Blockchain) jusqu'à leur destination (Dashboard Looker Studio).
Whale (Baleine)	Acteur du marché détenant une quantité significative de cryptomonnaie, capable d'influencer les tendances du marché par ses actions d'achat ou de vente.

La section suivante va détailler l'architecture technique choisie pour matérialiser cette solution, en commençant par la présentation exhaustive de la source de données brutes et l'utilisation de Google BigQuery comme pierre angulaire du traitement.

Architecture des données

Source des données



Pour réaliser une analyse On-Chain de haute fidélité, nous nous appuyons sur la puissance de calcul et la curation de données du Google Cloud Public Dataset officiel :

- Lien d'accès à la base Cloud sur Google Cloud Marketplace :
<https://console.cloud.google.com/marketplace/product/bitcoin-cash/crypto-bitcoin-cash?project=bigdata-cloud-bi-realtime> ;
- Nom du dataset : `bigquery-public-data.crypto_bitcoin_cash` ;
- Fréquence de mise à jour : temps réel (la base est synchronisée en permanence avec le réseau Bitcoin Cash) ;

- Modèle économique : ce dataset est hébergé sur Google BigQuery et est inclus dans l'offre gratuite de 1 To/mois de puissance de calcul, optimisant nos coûts de développement.

Aperçu du dataset

Le Bitcoin Cash est une cryptomonnaie qui permet d'inclure davantage d'octets dans chaque bloc que son ancêtre commun, le Bitcoin, ce qui est essentiel pour la vitesse. Cet ensemble de données contient l'intégralité des données de la blockchain, prétraitées pour être facilement compréhensibles et répondre aux besoins courants tels que l'audit, l'analyse et la recherche sur les propriétés économiques et financières du système.

Cet ensemble de données s'inscrit dans un effort plus vaste visant à rendre les données sur les cryptomonnaies disponibles dans BigQuery via le programme Google Cloud Public Datasets. L'interopérabilité des analyses a été pensée, permettant d'utiliser un schéma unifié pour toutes les cryptomonnaies de type Bitcoin, garantissant ainsi l'évolutivité de notre solution.

Justification du choix d'architecture des données

Le choix d'utiliser le dataset public BigQuery pour les données brutes est stratégique et offre plusieurs avantages décisifs pour la réalisation rapide et performante de notre solution de BI en temps réel.

Concentration sur l'analyse (Modèle ELT)

Grâce à l'intégration directe des données blockchain dans BigQuery, nous passons d'un ETL complexe à un modèle **ELT simplifié**. Cela élimine la gestion lourde de l'infrastructure pour nous concentrer immédiatement sur l'analyse et la création de valeur métier.

Performance et scalabilité serverless

BigQuery est un Data Warehouse Serverless, géré par Google, qui offre des performances exceptionnelles pour le scan et l'agrégation des données massives de la blockchain. Ceci garantit que nos requêtes (et futurs rapports en temps réel) s'exécutent en quelques secondes, même sur l'historique complet des transactions.

Schéma des composantes du dataset

Le Dataset Bitcoin Cash dans BigQuery est organisé autour de quatre composantes majeures qui nous permettent de reconstruire l'activité du réseau.

Table Transactions (L'activité)

Cette table est la plus importante, car elle détaille chaque transfert de valeur sur le réseau.

Champ	Type	Signification
hash	STRING	L'identifiant unique de la transaction (son numéro de série).
size	INTEGER	La taille de la transaction en octets.
virtual_size	INTEGER	La taille virtuelle utilisée pour calculer les frais.
version	INTEGER	La version du protocole de la transaction.
block_hash	INTEGER	L'identifiant du bloc qui a validé la transaction.
block_number	INTEGER	Le numéro séquentiel du bloc (sa position dans la chaîne).
block_timestamp	TIMESTAMP	L'heure exacte de validation de la transaction. Crucial pour notre analyse en temps réel.
input_count	INTEGER	Le nombre d'adresses d'envoi.
output_count	INTEGER	Le nombre d'adresses de réception.
input_value	NUMERIC	Valeur totale entrante.
output_value	NUMERIC	Valeur totale sortante. Utilisé pour le calcul de volume.
is_coinbase	BOOLEAN	Indique si c'est la création monétaire du réseau (la récompense des mineurs).
fee	NUMERIC	Les frais payés par l'utilisateur pour le minage.
inputs	RECORD	Données imbriquées détaillant les sources de fonds
outputs	RECORD	Données imbriquées détaillant les

Table Blocks (Les métadonnées du réseau)

Cette table fournit les informations de structure du réseau (la "boîte" qui contient les transactions).

Champ	Type	Signification
hash	STRING	L'identifiant unique de la transaction (son numéro de série).
size	INTEGER	La taille de la transaction en octets.
number	INTEGER	Le numéro du bloc (sa hauteur dans la chaîne).
version	INTEGER	La version du protocole de la transaction.
timestamp	TIMESTAMP	L'heure exacte de validation de la transaction. Crucial pour notre analyse en temps réel.
transactions_count	INTEGER	Le nombre total de transactions validées dans ce bloc.
difficulty	NUMERIC	La difficulté de calcul pour miner ce bloc.
merkle_root	STRING	L'empreinte numérique qui assure que toutes les transactions sont intactes.
fee	NUMERIC	Les frais payés par l'utilisateur pour le minage.

Table Inputs (Source des fonds)

Ce champ est une structure de données imbriquée (RECORD) dans la table `transactions`. Il nous permet de savoir d'où vient la monnaie consommée.

Champ	Type	Signification
index	INTEGER	L'ordre de l'entrée dans la transaction.
value	NUMERIC	La valeur en BCH qui est consommée de l'adresse source.
addresses	STRING	L'adresse crypto d'où provient la monnaie. Crucial pour l'analyse des adresses actives.
spent_transactions	STRING	L'identifiant de la transaction précédente qui a créé ces fonds (la traçabilité).

Table Outputs (Destination des fonds)

Ce champ est une structure de données imbriquée (RECORD) dans la table transactions. Il nous permet d'où vient de suivre d'où vient la monnaie.

Champ	Type	Signification
index	INTEGER	L'ordre de l'entrée dans la transaction.
value	NUMERIC	La valeur en BCH qui est consommée de l'adresse source.
addresses	STRING	L'adresse crypto d'où provient la monnaie. Crucial pour l'analyse des adresses actives.

Processus d'exploitation des données

Cette section décrit le guide pas à pas pour configurer l'environnement d'analyse sur Google Cloud Platform (GCP). Ce processus garantit la reproductibilité du projet et le respect du cadre Serverless.

Etape 1 : Initialisation de la console GCP

Accès à la Plateforme : rechercher Google Cloud Platform dans le navigateur et se connecter avec les identifiants du compte de service.

Accès à la Console : cliquer sur Accéder à la console (ou Go to console) pour afficher le tableau de bord principal de gestion des services Cloud.

Voici une illustration de la première étape d'initialisation de la console GCP

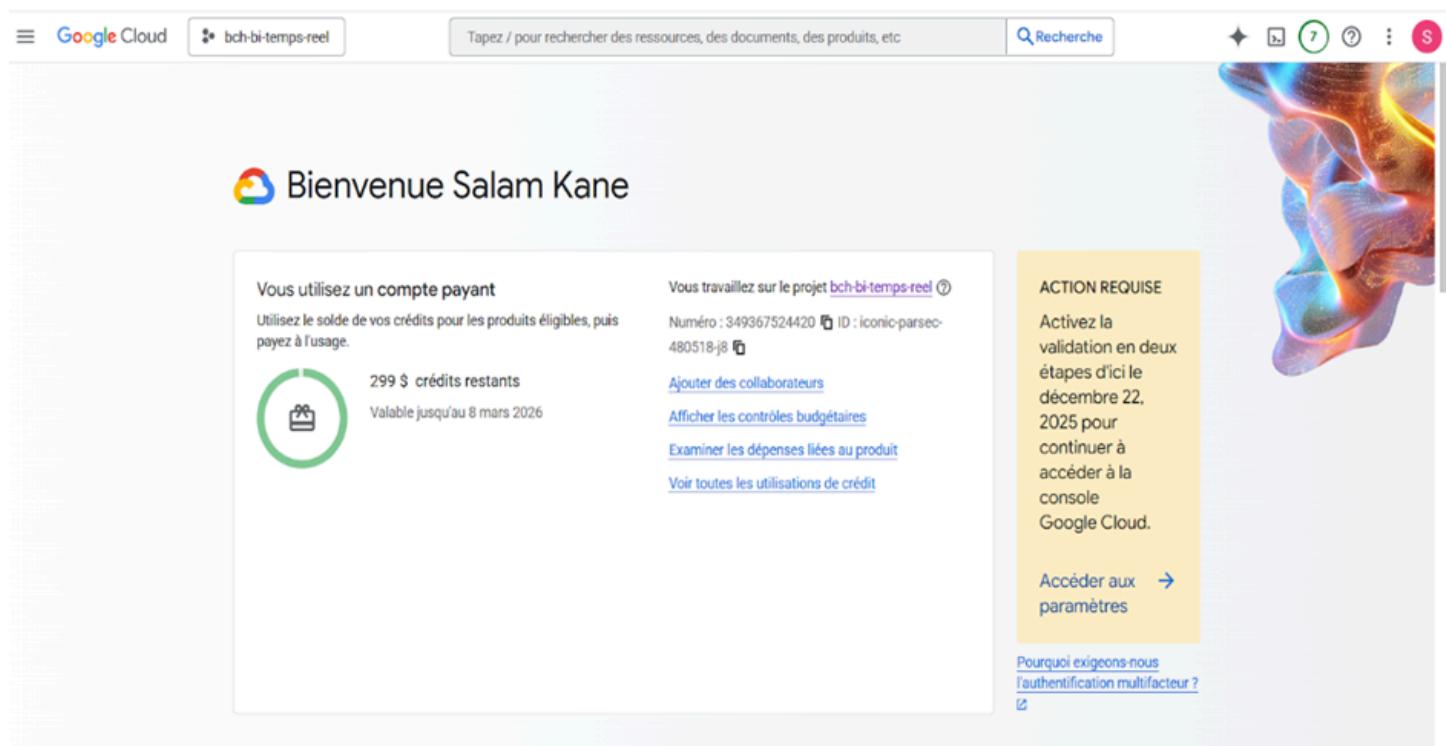


Illustration 1 : Console d'accueil Google Cloud Platform

Etape 2 : Crédit et gestion du projet

- Renommage du Projet : le projet par défaut créé par GCP est renommé (ex : projet-bdcc-2025-chainsight). Ce nommage est essentiel pour l'organisation, le suivi précis de l'utilisation des ressources et l'attribution des rôles IAM.

- Sélection de la Région : nous confirmons que les ressources seront localisées dans une région Cloud appropriée (ex : US ou EU) pour minimiser la latence de l'analyse.

The screenshot shows the Google Cloud IAM Admin interface. The left sidebar lists various services like Cloud Hub, Présentation Cloud, Solutions, and Consultations récentes. Under 'Produits préférés', 'API et services' is selected, and 'IAM et administrati...' is highlighted with a red box. A sub-menu for 'Paramètres' is also highlighted with a red box. The main content area displays information about the current project, including its number (349367524420), ID (iconic-parsec-480518-j8), and a call-to-action to complete a two-step validation by December 22, 2025. A yellow box on the right labeled 'ACTION REQUISE' provides instructions for enabling multi-factor authentication.

Illustration 2 : interface de renommage de projet et sélection de région GCP

This screenshot is identical to Illustration 2, showing the Google Cloud IAM Admin interface for the same project. The 'IAM et administrati...' menu item is highlighted with a red box, and a sub-menu for 'Paramètres' is also highlighted with a red box. The main content area displays project details and a call-to-action for multi-factor authentication.

Illustration 3 : Sélection de “IAM Administration” dans la barre de navigation puis de “paramètres”

The screenshot shows the Google Cloud IAM and Adminstration interface. On the left, a sidebar lists various options like IAM, PAM, Insights sur la sécurité, etc. The 'Paramètres' option is selected. In the main area, there's a form to 'Effectuer la migration'. A red box highlights the 'Nom du projet' input field, which contains 'bch-bi-temps-reel'. Below it, the 'ID du projet' is shown as 'iconic-parsec-480518-j8' and the 'Numéro du projet' as '349367524420'. There are sections for 'Compte de facturation' and 'Access Transparency'.

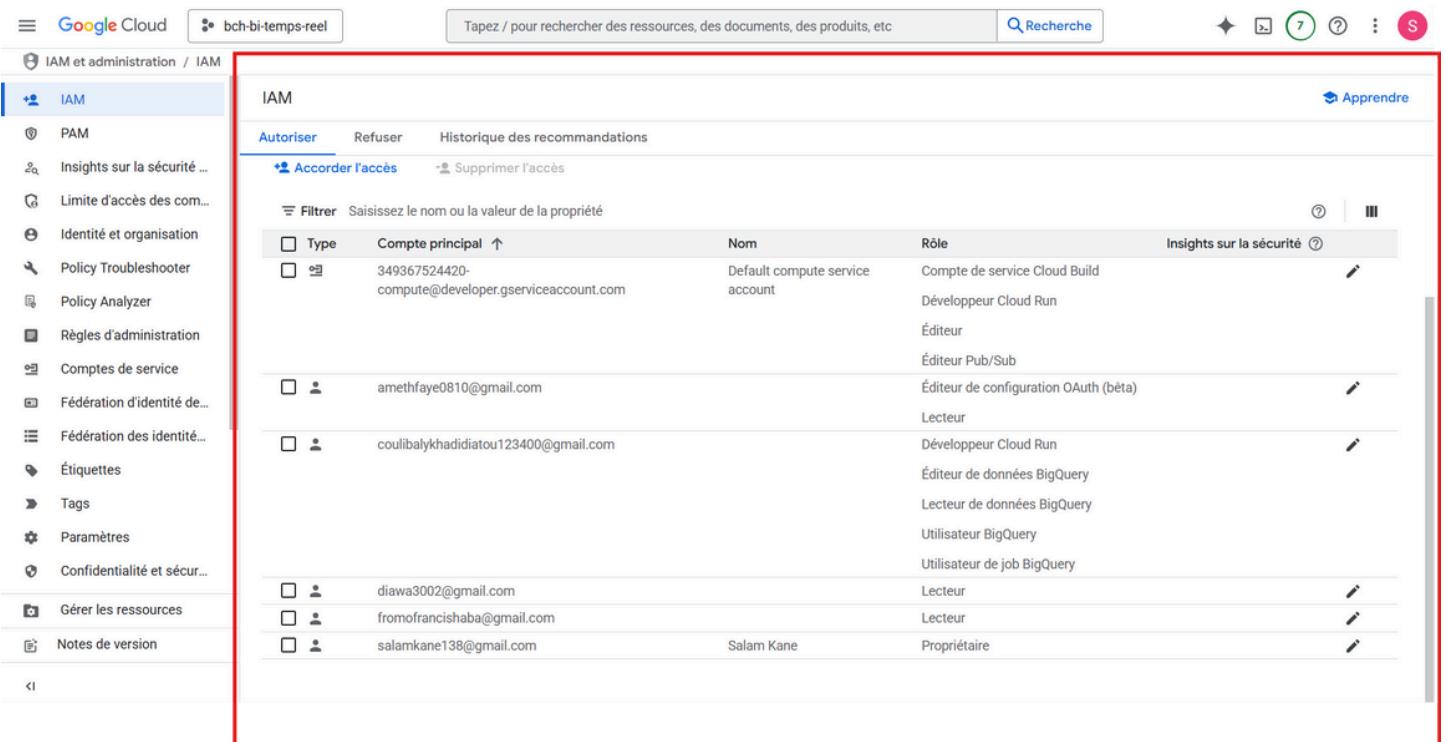
Illustration 4 : Interface de renommage de projet et sélection de région GCP

Etape 3 : Configuration du Contrôle d'Accès (IAM et Principe du Moindre Privilège)

La sécurité et la collaboration sont assurées par l'outil IAM (Identity and Access Management), en respectant le principe du moindre privilège.

1. **Navigation IAM** : Accéder à la barre de navigation, sélectionner IAM et administration puis IAM.
2. **Définition des Rôles** : Pour chaque membre de l'équipe (ou service automatisé), les rôles sont strictement définis. Ces rôles garantissent que l'équipe Data Analyst peut exécuter des Jobs BigQuery sans pouvoir, par exemple, modifier les factures.

Rôle GCP Défini	Détail Fonctionnel dans le Projet
Propriétaire	Rôle attribué au créateur du projet. Gère l'infrastructure, la facturation et l'attribution des rôles.
Lecteur de données BigQuery	Permet d'accéder aux données brutes et aux tables de KPIs. Nécessaire pour le BI avec Looker Studio.
Utilisateur de Job BigQuery	Autorise l'exécution des requêtes (Jobs) consommant de la puissance de calcul.
Éditeur Pub/Sub & Cloud Run Éditeur	Rôles Serverless prévus pour l'étape future de mise en temps réel et d'intégration de microservices.



The screenshot shows the Google Cloud IAM console interface. On the left, there's a sidebar with navigation links like PAM, Insights sur la sécurité, Limite d'accès des com..., Identité et organisation, Policy Troubleshooter, Policy Analyzer, Règles d'administration, Comptes de service, Fédération d'identité de..., Fédération des identités, Étiquettes, Tags, Paramètres, Confidentialité et sécur..., Gérer les ressources, and Notes de version. The main area is titled 'IAM' and shows a table of users and their roles. A red box highlights the table area. The table columns are Type, Compte principal, Nom, Rôle, and Insights sur la sécurité. The data includes:

Type	Compte principal	Nom	Rôle	Insights sur la sécurité
Compte principal	349367524420-compute@developer.gserviceaccount.com	Default compute service account	Compte de service Cloud Build Développeur Cloud Run Éditeur Éditeur Pub/Sub	
Utilisateur	amethfaye0810@gmail.com		Éditeur de configuration OAuth (bêta) Lecteur	
Utilisateur	coulibalykhadiatou123400@gmail.com		Développeur Cloud Run Éditeur de données BigQuery Lecteur de données BigQuery Utilisateur BigQuery Utilisateur de job BigQuery	
Utilisateur	diawa3002@gmail.com		Lecteur	
Utilisateur	fromofranchishaba@gmail.com		Lecteur	
Utilisateur	salamkane138@gmail.com	Salam Kane	Propriétaire	

Illustration 5 : Console IAM affichant les membres et leurs rôles précis dans le cloud

Etape 4 : Accès à BigQuery Studio et Localisation du Dataset Public

C'est l'étape de connexion à notre Data Warehouse Serverless.

- 1. Accès au Studio** : Dans la barre de navigation, sélectionner BigQuery puis Studio.
- 2. Recherche du Dataset** : Dans l'explorateur (panneau de gauche), au lieu de chercher uniquement dans notre projet, il est crucial de cocher l'option "Rechercher sur tous les projets".

Localisation : Entrer le nom exact du dataset : crypto_bitcoin_cash. Ce processus est la démonstration d'un accès Serverless à une ressource publique.

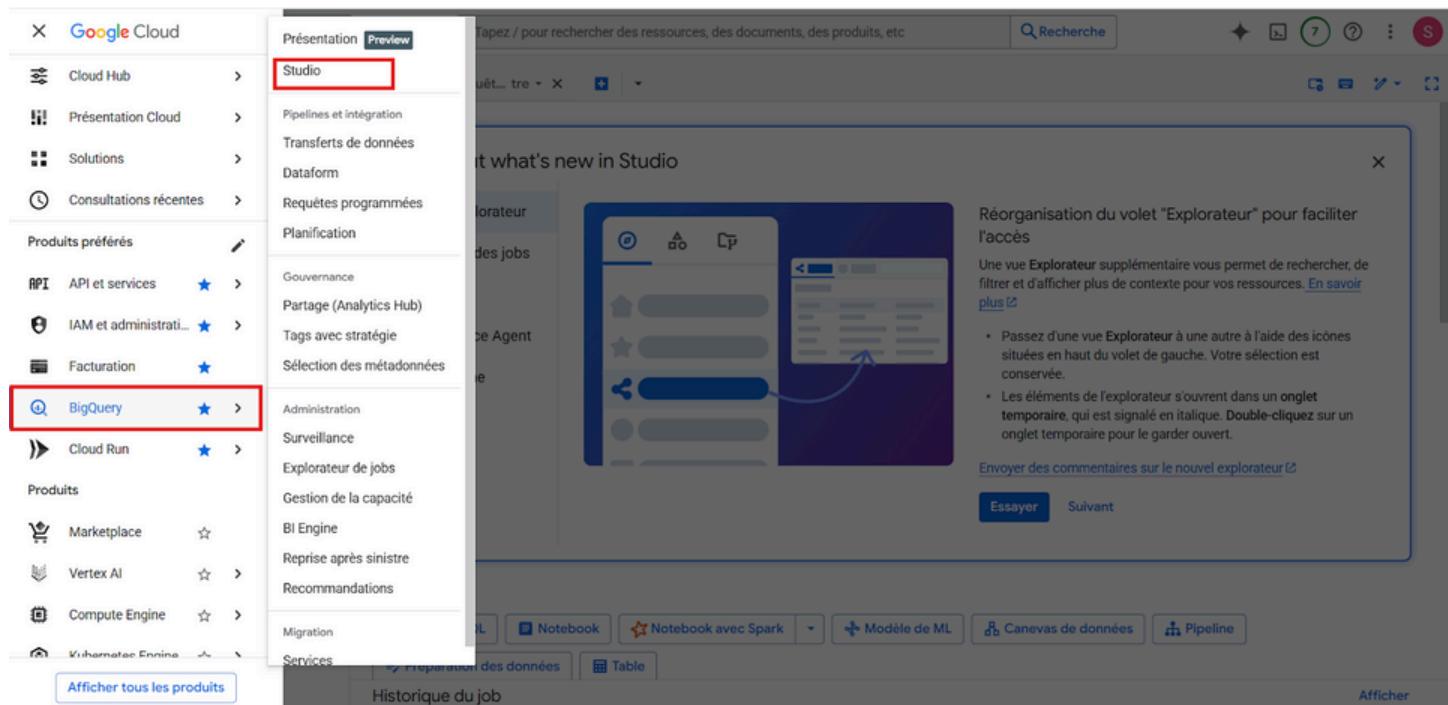


Illustration 6 : Accès à BigQuery Studio depuis la barre de navigation

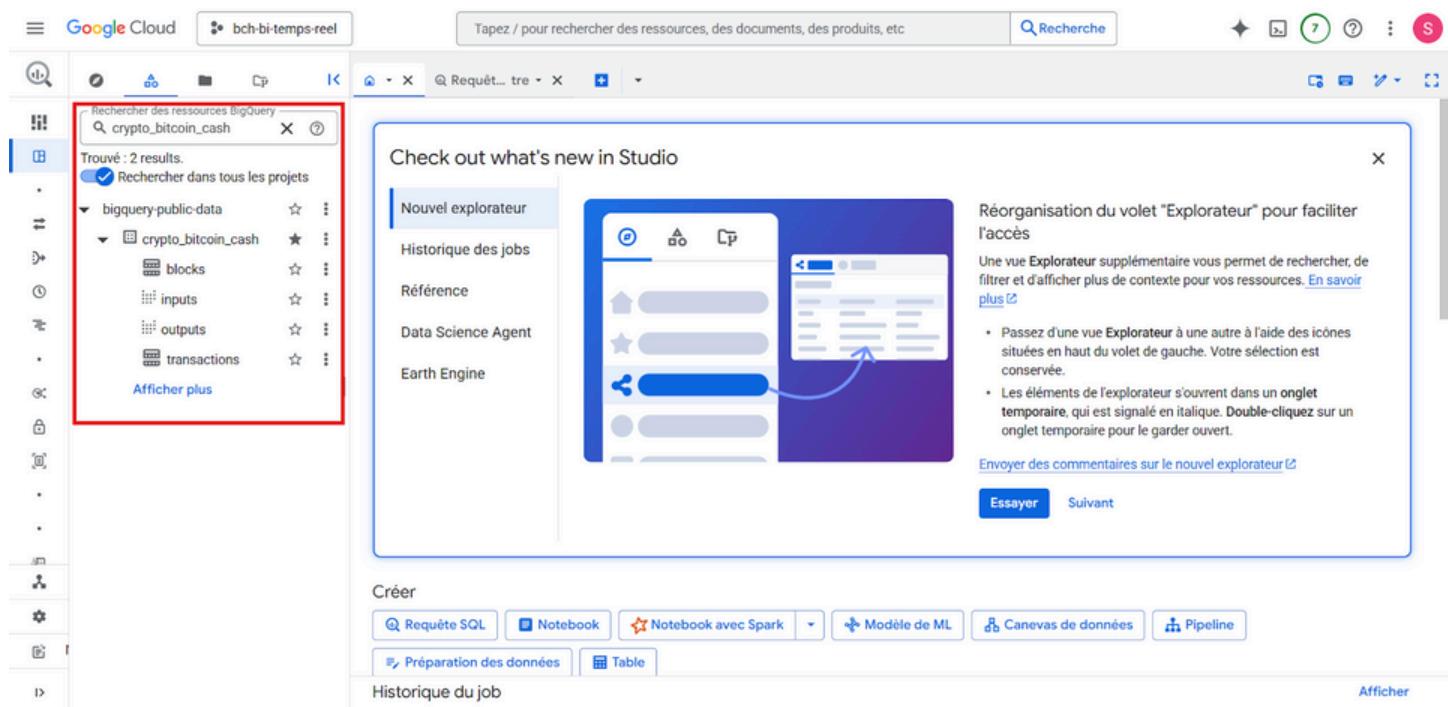


Illustration 7 : BigQuery Studio montrant la recherche "crypto_bitcoin_cash" avec l'option "Rechercher dans tous les projets"

Etape 5 : Préparation de la Zone de Transformation (ELT)

Pour que notre code SQL fonctionne, nous devons créer le schéma de destination et confirmer l'accès à la source.

- Épinglage** : Le dataset public est épinglé pour une référence rapide en appuyant sur l'étoile pour l'ajouter en favori comme on peut le constater dans l'image précédente pour faciliter la navigation.
- Vérification** : Nous cliquons sur les tables (transactions, blocks) pour vérifier leur schéma et leur structure (cf. Section 2.3).

Création du Dataset de Destination : Création du dataset crypto_analytics (iconic-parsec-480518-j8.crypto_analytics) qui hébergera les tables de KPIs finales (kpi_transactions_journalieres, etc.). Ce dataset doit être localisé dans la même région que le projet pour optimiser la performance et réduire les frais.

The screenshot shows the Google Cloud BigQuery Studio interface. The left sidebar shows various project and dataset options. The main area displays a list of datasets under the project 'iconic-parsec-480518-j8'. A new dataset named 'crypto_analytics' has been created and is listed in the table. The table has columns: ID de l'ensemble de données, Type, Zone, Date et heure de création, and Étiquette. The 'crypto_analytics' row is highlighted with a red box. The 'Zone' column shows 'US' and the 'Type' column shows 'Default'.

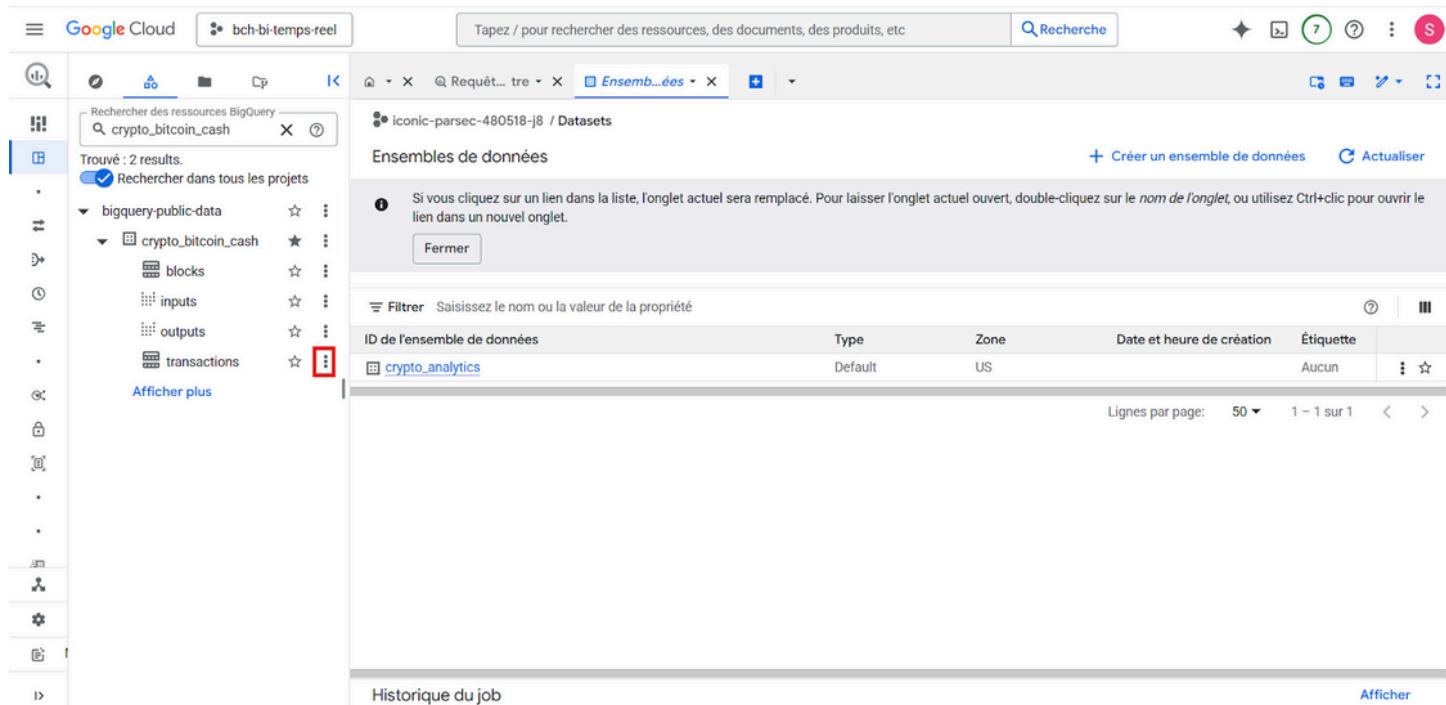
ID de l'ensemble de données	Type	Zone	Date et heure de création	Étiquette
crypto_analytics	Default	US		Aucun

Illustration 8 : Crédit à la création du Dataset de destination crypto_analytics dans BigQuery

Etape 6 : Lancement du Moteur de Requêtes et Déploiement du Code SQL

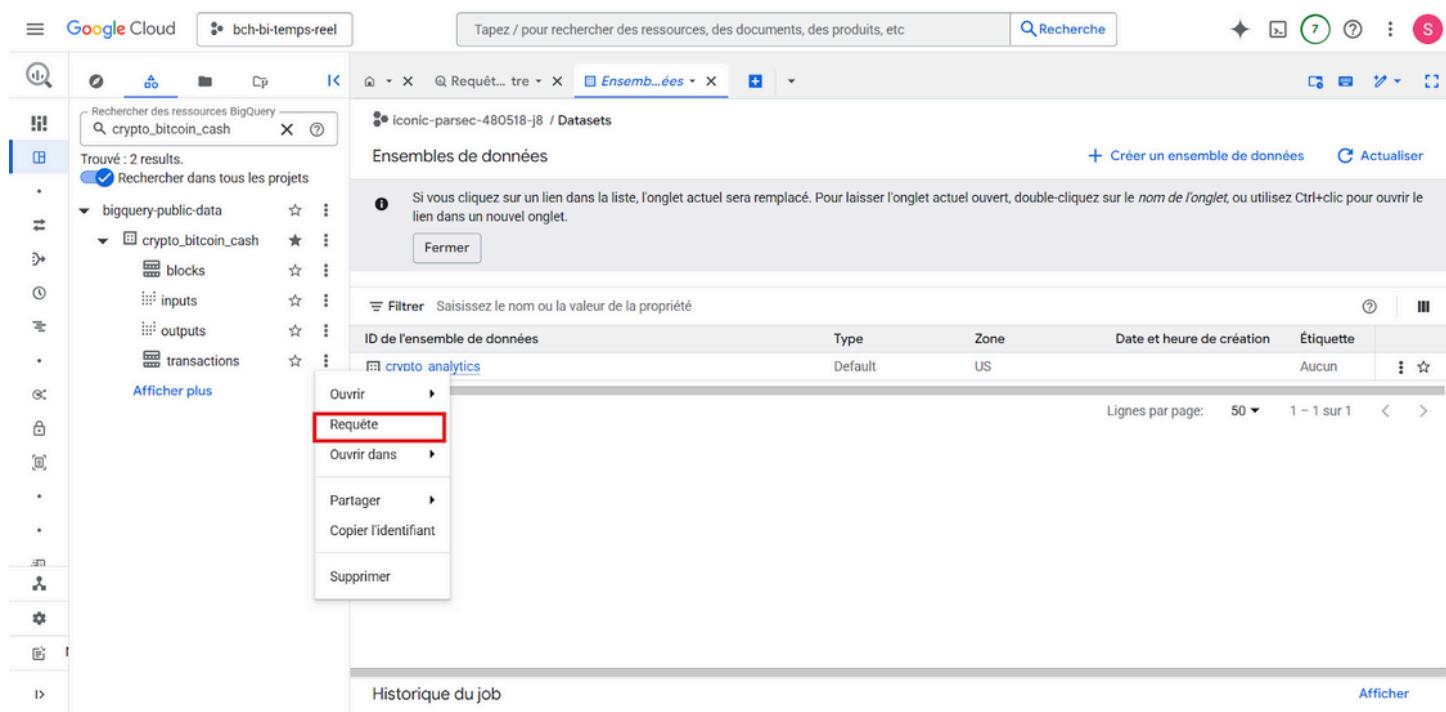
Cette dernière étape prépare l'exécution du pipeline de transformation ELT.

- Ouverture de l'Éditeur : Dans l'explorateur, nous cliquons sur les trois points (...) à côté de la table de travail ou du dataset et sélectionnons Requête.



The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar displays a project named 'bch-bi-temps-reel'. Under the 'Datasets' section, there is a dataset named 'crypto_bitcoin_cash' which contains three tables: 'blocks', 'inputs', and 'transactions'. The 'transactions' table is highlighted with a red box. In the main pane, the 'Ensembles de données' (Data Sets) page is open, showing a table with one row for 'crypto_analytics'. A context menu is open over this row, with the 'Requête' option highlighted by a red box. The top navigation bar includes the Google Cloud logo, a search bar, and various navigation icons.

Illustration 9 : Sélection de la table transaction puis du menu avec les 3 points.



This screenshot is identical to the previous one, showing the Google Cloud BigQuery interface. The 'transactions' table is selected, and a context menu is open over the 'crypto_analytics' data set row. The 'Requête' option in the menu is highlighted with a red box. The rest of the interface, including the sidebar and the main data table, remains the same.

Illustration 10 : Sélection de l'option Requête dans le menu déroulant des 3 points.

- **Exécution du Code :** L'interface de l'éditeur SQL est prête à recevoir et exécuter les scripts du dépôt Git (fichiers 1_initialisation.sql, 2_kpi_journaliers.sql, etc.), transformant les données brutes en indicateurs décisionnels prêts pour la BI.

The screenshot shows the Google Cloud BigQuery interface. On the left, there's a sidebar with various icons and a search bar for 'Rechercher des ressources BigQuery'. The main area has a search bar at the top with 'Requête sans titre' and a 'Tapez / pour rechercher des ressources, des documents, des produits, etc.' placeholder. Below it is a toolbar with 'Exécuter', 'Enregistrer', and 'Télécharger' buttons. A query editor window contains the following code:

```
1 SELECT FROM `bigquery-public-data.crypto_bitcoin_cash.transactions` WHERE  
block_timestamp_month = "2025-12-15" LIMIT 1000
```

A red error icon is present next to the first line. A tooltip below the line says 'Syntax error: SELECT list must not be empty at [1:9]'. To the right of the editor, a 'Documentation de référence' panel is open, showing the schema for the 'transactions' table:

Nom du champ	Type
hash	STRING
size	INTEGER
virtual_size	INTEGER
version	INTEGER
lock_time	INTEGER
block_hash	STRING
block_number	INTEGER
block_timestamp	TIMESTAMP
block_timestamp_month	DATE
input_count	INTEGER

Illustration 11 : Interface pour les requêtes SQL

Logique d'implémentation SQL

Cette section décrit le pipeline de transformation (la phase **Transform** du modèle ELT) exécuté dans **Google BigQuery**. L'objectif est de convertir les données brutes de la blockchain en indicateurs décisionnels optimisés pour la Business Intelligence. Cette partie de la documentation technique explique de manière succincte la méthodologie adoptée pour la conception des différentes requêtes SQL, mais pour plus de détails concernant la description de la méthodologie d'implementation SQL veuillez consultez le lien suivant [Documentation et Scripts SQL du projet Bitcoin Cash](#)

Initialisation et Optimisation de la Base (1_initialisation.sql)

Ce premier script assure la fondation technique et l'optimisation des performances Big Data :

- Création du Schéma : Mise en place du Dataset crypto_analytics (**iconic-parsec-480518-j8.crypto_analytics**) en zone US.
- Nettoyage et Enrichissement : Conversion des satoshis en Bitcoin Cash (BCH) via **SAFE\DIVIDE(t.output_value, 1e8)** et désimbrication des adresses (UNNEST).

Performance : Partitionnement quotidien par horodatage_bloc et clustering par hachage_transaction pour minimiser les coûts de scan.

Calcul des KPIs Journaliers et Globaux (2_kpi_journaliers.sql & 3_kpi_globaux.sql)

Nous utilisons des Common Table Expressions (CTE) pour générer des indicateurs de santé du réseau parmi lesquels nous avons :

KPI	Définition et Intérêt
Volume & Transactions	Mesure l'adoption et la liquidité totale (BCH) transférée chaque jour.
Whale Ratio	Proportion du volume issu de transactions ≥ 100 BCH, indicateur de l'activité des gros portefeuilles.
Vitesse de Circulation	Ratio Volume Sortant / Volume Entrant, indiquant la dynamique des échanges.
Valeur Médiane	Calculée globalement via APPROX_QUANTILES pour offrir une vue stable.

Stratégie de Simulation du "Temps Réel" via Fenêtre Glissante

Le cœur de ce projet repose sur une approche innovante de **simulation temporelle**. Cette méthode a été adoptée pour pallier deux contraintes majeures :

- Limitation des Données** : Les jeux de données publics gratuits pour Bitcoin Cash sur BigQuery ne sont plus mis à jour en temps réel depuis fin 2024.
- Contraintes Budgétaires** : L'accès à des flux de données blockchain en temps réel via des API propriétaires implique des coûts extrêmement élevés, dépassant les capacités de financement du projet.

Mécanique de la Simulation :

Pour démontrer la viabilité de notre pipeline, nous créons artificiellement notre propre "temps réel". Nous procédons par une translation de données : le pipeline extrait les transactions d'une semaine de données historiques, les agrège, et les injecte dans une fenêtre de 30 jours en supprimant les 7 jours les plus anciens.

Orchestration et Automatisation via Cloud Run

Ce processus de mise à jour est automatisé à un intervalle régulier de **5 minutes** via une requête programmée exécutée par **Google Cloud Run**.

- **Objectif :** Montrer comment le système se comporterait si nous disposions de données à jour.
- **Résultat :** Le tableau de bord affiche en permanence des indicateurs qui évoluent dynamiquement toutes les 5 minutes, simulant parfaitement un environnement de production réel.

Note technique : Cette partie a été amplement détaillée dans le repository GitHub du projet, plus précisément dans le sous-dossier SQL :

 [Accéder au dossier SQL sur GitHub](#)

Vous trouverez tous les détails concernant la mise à jour en temps réel, la logique de translation et la configuration des scripts dans le **README du sous-dossier SQL**.

Seuils d'Alerte et Actions (Règles Métier)

Nos KPIs ont une vocation **prescriptive** et sont associés à des seuils critiques qui déclenchent des signaux d'alerte pour guider la décision des investisseurs, conformément à l'objectif de la BI en temps réel.

Il est essentiel de noter que ces seuils critiques sont définis à des fins purement **interprétatives et décisionnelles** dans le cadre du rapport de Business Intelligence. Ils ne sont pas implémentés directement comme des **conditions d'alerte ou des filtres dans le code SQL de BigQuery** fourni, mais servent de grille d'analyse au décideur utilisant le Dashboard.

KPI surveillé	Seuil Critique	Interprétation pour l'Investisseur	Action Recommandée
Whale ratio	> 10% du volume journalier	Mouvement massif de capitaux initié par de gros acteurs. Potentiel signal de pression de vente ou d'achat	Alerte Rouge : Réduire l'exposition court terme ou, inversement, placer des ordres d'achat tactiques.
Vitesse de Circulation	> 5.0 (Vélocité anormale)	Forte dynamique d'échange des fonds (haute spéculation ou panique).	Alerte Orange : Surveiller les frais de transaction qui augmentent souvent lors des pics de vélocité.
Adresses Actives	Chute de > 20% (vs J-1)	Perte soudaine d'intérêt et d'utilisation du réseau par les utilisateurs actifs.	Notification Info : Réévaluer l'attractivité de l'actif à moyen terme.

Connexion à Looker Studio pour la Visualisation en Temps Réel

La dernière étape du pipeline consiste à connecter les tables de KPIs optimisées au service de Business Intelligence **Looker Studio** pour la création du Dashboard dynamique.

Etape 1 : Activation de la Connexion au Service

Vérifier que les rôles IAM sont configurés et que le service Looker Studio est activé dans le projet GCP.

Etape 2 : Crédit du Rapport et Sélection du Connecteur

1. Accéder à Looker Studio via un navigateur.

2. Cliquer sur Créer puis Rapport.
3. Cliquer sur Ajouter des données.
4. Sélectionner le connecteur natif BigQuery pour établir la liaison la plus performante.

Etape 3 : Liaison Directe aux Tables de KPIs

Nous établissons une connexion Direct Query pour garantir l'aspect temps réel du rapport :

1. Dans l'interface BigQuery de Looker Studio, naviguer vers la source de données : Projets Partagés ou Mes Projets.
2. Sélectionner le Projet Cloud contenant les résultats : iconic-parsec-480518-j8.
3. Sélectionner le Dataset : crypto_analytics.
4. Sélectionner la Table : La table kpi_transactions_journalieres est choisie comme source principale pour les graphiques de tendances.
5. Cliquer sur Ajouter pour finaliser la connexion.

The screenshot shows the Looker Studio dashboard. On the left, there's a sidebar with 'Créer' (Create) highlighted with a red box. Below it are 'Éléments récents' (Recent items), 'Partagés avec moi' (Shared with me), 'À moi' (To me), and 'Corbeille' (Trash). The main area has tabs for 'Rapports' (Reports), 'Sources de données' (Data sources), and 'Explorateur'. Under 'Commencer avec un modèle', there are cards for 'Rapport vide' (Blank report), 'GA4 Report Google Analytics', 'Marketing d'Acme Google Analytics', and 'Search Console Report Search Console'. A 'Galerie de modèles' section is also visible. At the bottom, a table lists a report named 'tableau de bord projet big data' with details like 'Nom', 'Indifférent', 'Dernière ouverture par moi', 'Emplacement', and sharing information.

Illustration 12 : Accès à l'interface Looker Studio

This screenshot is similar to Illustration 12, showing the Looker Studio dashboard. The 'Créer' button is still highlighted in red. In the sidebar, the 'Rapport' button is also highlighted with a red box. The main interface remains the same, with the 'Rapports' tab selected and various report and data source cards displayed. The table at the bottom shows the same report details as in Illustration 12.

Illustration 13 : Crédit d'un rapport BI dans Looker Studio

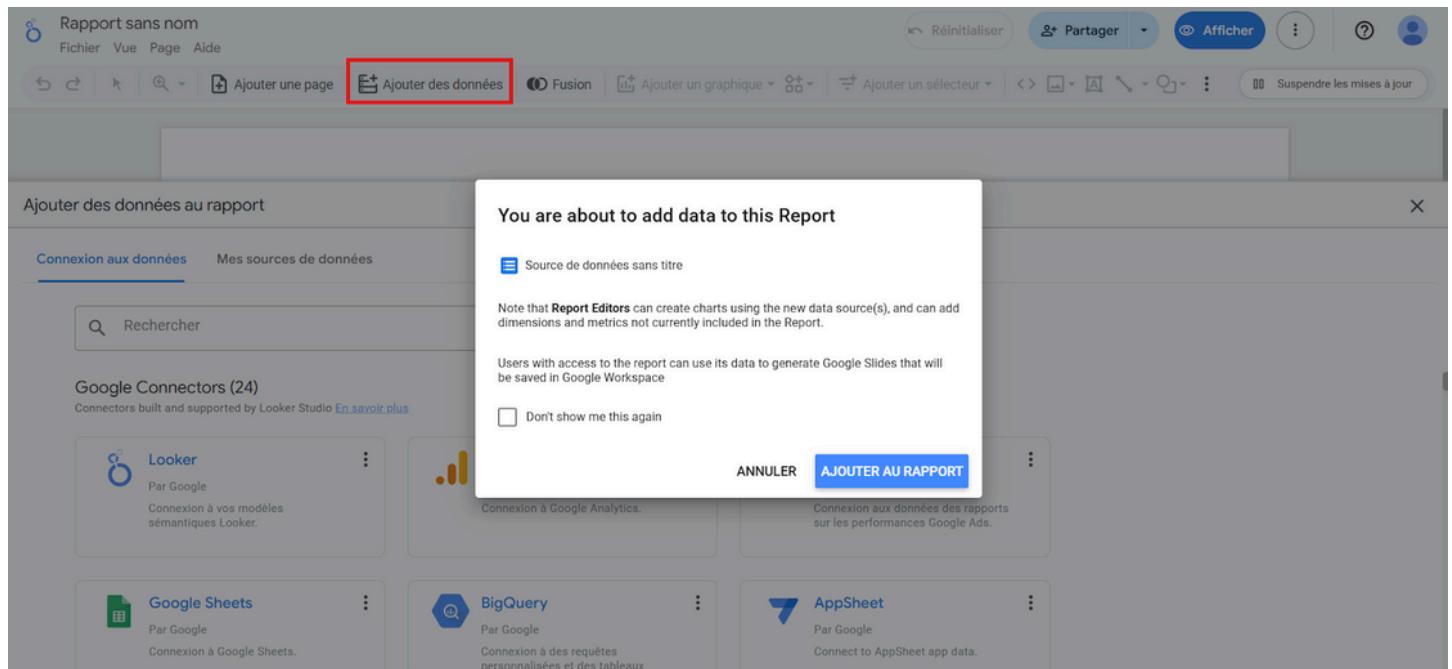


Illustration 14 : Connexion des données à notre Dashboard créé

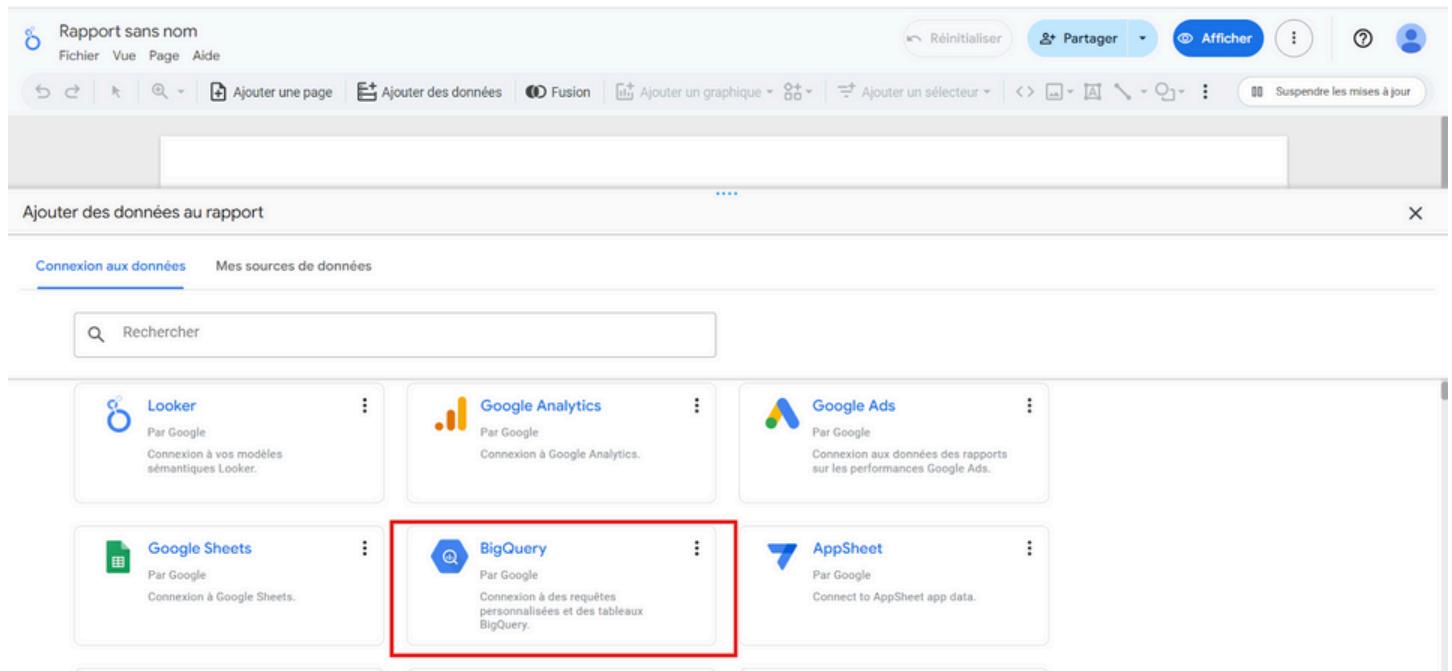


Illustration 15 : Connecter les données en provenance de BigQuery

The screenshot shows the Looker Studio interface for connecting to BigQuery. The 'Project' section lists 'bch-bi-temps-reel' as the selected project. In the 'Ensemble de données' section, 'crypto_analytics' is selected. Under 'Table', 'kpi_transactions_journalieres' is selected. A red box highlights the 'Ajouter' (Add) button at the bottom right of the configuration panel.

Illustration 16 : Interface de connexion Looker Studio à BigQuery, montrant la sélection du projet, du dataset `crypto_analytics` et de la table des KPIs

Le Dashboard Looker Studio est désormais connecté en mode Direct Query. Chaque visualisation exécute une requête optimisée sur BigQuery, garantissant que les données affichées aux investisseurs sont à jour et conservent leur nature **temps réel** lorsque le dashboard sera designé.

Etape 4 : Mise à jour des tables en temps réel via Cloud Functions

Dans cette étape intervient l'utilisation d'un service de Google Cloud Platform (GCP) appelé **Cloud Functions** pour mettre en place une fonction via un environnement Python qui va exécuter une boucle de code pour programmer la mise à jour des données sur la durée que l'on aura défini et ainsi simuler une impression de temps réel. C'est dans ce sens que pour réaliser cela, il faut procéder aux étapes suivantes :

- Rechercher **Cloud Functions** dans la barre de recherche Google Cloud Platform.
- Créer un service dans **Cloud Functions** qui va contenir la fonction exécutée dans l'environnement python de Google Cloud Platform.

Google Cloud bch-bi-temps-reel cloud functions Recherche Envoyer des commentaires

Filtrer par

- Produit ou page
- Documentation ou tutoriel
- Marketplace et API
- Organisation
- Dossier
- Projet
- Ressources

Filtres de ressources

Projet, dossier ou organisa bch-bi-temps-reel Type de ressource Tous

Résultats de recherche

Affichage de 30 résultats sur 64 pour "cloud functions".

Fonctions Cloud Run

Exécutez votre code sans aucune gestion de serveur grâce à une offre FaaS (Functions as a Service) évolutive et facturée à l'usage

Type : Produit ou page

Cloud Functions

Un environnement sans serveur pour créer et connecter des services cloud

Type : Produit Marketplace Producteur : Google

Cloud Run

Exécutez votre code, fonction ou conteneur sur l'infrastructure sans serveur de Google

Type : Produit ou page

Cloud Functions versus Cloud Run: quand et pourquoi ...

Cloud Functions optimise la productivité tout en sollicitant très peu les opérationnels : chaque « fonction » est un composant indépendant et ...

Type : Documentation

Recherchez plus vite grâce à ces astuces Appuyez sur / pour effectuer une recherche depuis n'importe quel emplacement dans la console Trouvez 111 types de ressources et plus encore grâce aux nouveaux filtres de cette page Recherchez une clé API en saisissant la chaîne de clé associée dans le champ de recherche Recherchez les meilleures offres AWS ou Azure, et obtenez des suggestions de produits Google Cloud similaires. [En savoir plus](#)

Illustration 17 : Accès à Cloud Functions via la barre de recherche

Google Cloud bch-bi-temps-reel cloud functions Recherche Regarder la vidéo Ignorer

Aperçu Actualiser

Ressources les plus utilisées

Nom	Région	Type	Dernière mise à jour
simulateur-sql-temps-reel	europe-west1	Service	il y a 56 minutes

Scaling Erreurs 7 jours

simulateur-sql-temps-reel : 0,57

Déployer un service Web

Déployez un site Web ou une API. [En savoir plus](#)

Connecter un dépôt Déployer le conteneur

Créer un job par lot ou un pool de noeuds de calcul d'arrière-plan

Exécutez des scripts, des jobs Cron ou des charges de travail de traitement de données chargées en parallèle. [En savoir plus](#)

Créer un job Créer un pool de noeuds de calcul

Illustration 18 : Accès à l'onglet “Services” dans Cloud Functions

Google Cloud

bch-bi-temps-reel

cloud functions

Recherche

Deployez et faites évoluer plus rapidement des modèles d'IA avec Cloud Run et Vertex AI. [En savoir plus](#)

Regarder la vidéo Ignorer

Services Déployer un conteneur Connecter un dépôt (-) Écrire une fonction Actualiser

Un service expose un point de terminaison unique et adapte automatiquement l'infrastructure sous-jacente pour gérer les requêtes entrantes. Deployez une image de conteneur, du code source ou une fonction pour créer un service.

Filtrer Filtrer les services

Nom ↑	Type de déploiement	Requêtes/s	Région	Authentification	Ingress	Dernier déploiement	Déployé par
<input type="checkbox"/> simulator-sql-temps-reel	(-) Fonction	0	europe-west1	Accès public	Toutes	il y a 58 minutes	salamkane138@gmail.com

Illustration 19 : Crédation d'une fonction dans le service de Cloud Functions

Deployez et faites évoluer plus rapidement des modèles d'IA avec Cloud Run et Vertex AI. [En savoir plus](#)

Regarder la vidéo Ignorer

Créer un service Afficher la ligne de commande

Un service expose un point de terminaison unique et adapte automatiquement l'infrastructure sous-jacente pour gérer les requêtes entrantes. Le nom et la région du service ne peuvent pas être modifiés.

Synthèse des tarifs

Tarifs de Cloud Run

Version gratuite

180 000 premiers vCPU-seconde par mois

360 000 premiers Go-seconde par mois

2 millions de requêtes par mois

→ Consulter les détails des niveaux payants

Ouvrir le simulateur de coût

Fonction

Utiliser un éditeur intégré pour créer une fonction

Artifact Registry Docker Hub

Déployer une révision à partir d'une image de conteneur existante

GitHub

Déployer en continu depuis un dépôt (source ou fonction)

Configurer

Nom de Service * exemple-de-fonction

Région * europe-west1 (Belgique)

Comment choisir une région ?

URL du point de terminaison

https://exemple-de-fonction-349367524420.europe-west1.run.app

Exécution * Node.js ??

Créer Annuler

Illustration 20 : Configuration d'une fonction dans le service de Cloud Functions (P1)

Illustration 21 : Configuration d'une fonction dans le service de Cloud Functions (P2)

Remarque : Pour les configurations dans Conteneur, réseau et sécurité, on laisse tout par défaut sauf le délai d'expiration de la requête que dans notre cas nous mettons à 540s pour les besoins du projet.

```

import time
from google.cloud import bigquery
import functions_framework

@functions_framework.http
def hello_http(request):
    client = bigquery.Client()

    # --- CONFIGURATION DE LA BOUCLE ---
    # 15 répétitions avec une pause très courte (5s) pour un effet "temps réel" rapide
    iterations = 10
    pause_secondes = 40

    for i in range(iterations):
        try:
            # -----
            # ÉTAPE 1 : Reconstruction de la table avec les bons ALIAS
            # -----
            query_update_table = """
            DECLARE cur_start, cur_end, new_start, new_end, m_start, m_end DATE;
            -- 1. Récupération des dates
            SET cur_start = (SELECT start_date FROM `iconic-parsec-480518-j8.crypto_analytics.rolling_state` LIMIT 1);
            SET cur_end = (SELECT end_date FROM `iconic-parsec-480518-j8.crypto_analytics.rolling_state` LIMIT 1);
            """
        
```

Illustration 22 : Ajout du script d'automatisation python dans l'environnement intégré sur le fichier **main.py** et des dépendances **google-big-query** et **functions_framework** dans requirements.txt qui sont importées dans le script python puis déploiement.

- Accéder au service **Cloud Scheduler** en recherchant son nom dans la barre de recherche de Google Cloud Platform puis en cliquant dessus.
- Planifier dans **Cloud Scheduler** une instruction automatique qui va exécuter la fonction chaque minute en connectant le sujet créé lors de la création du service lié à la fonction à l'API **Cloud Scheduler**.

Résultats de recherche

Affichage de 30 résultats sur 35 pour "cloud scheduler".

Cloud Scheduler

Planifiez tous les jobs, y compris les jobs par lots, les jobs de big data et les opérations d'infrastructure cloud

Type : Produit ou page

Cloud Scheduler API

A fully managed enterprise-grade cron job scheduler

Type : API
Producteur : Google Enterprise API

Ignite-GP

Easily launch, manage and report on virtual infrastructure anywhere in the world with confidence.

Type : Produit Marketplace
Producteur : Gunpowder

Tarifs | Cloud Scheduler

Une tâche est facturée 0,10 \$/tâche/31 jours, soit 0,003 \$ par jour pour chaque tâche facturable. L'utilisation facturée d'une tâche est calculée par jour. Par ...

Illustration 23 : Accès à Cloud Scheduler via la barre de recherche

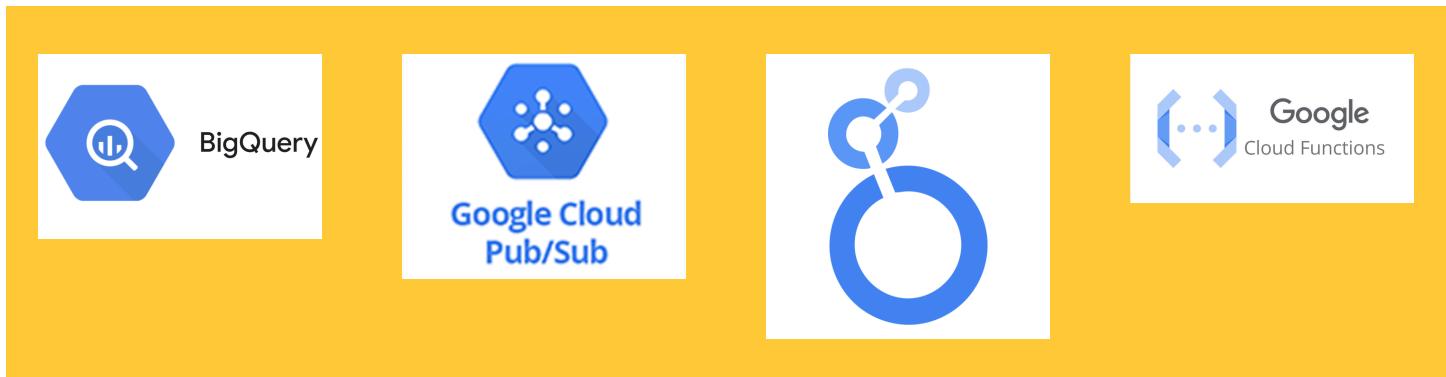
Nom	État de la dernière exécution	Région	État	Description	Fréquence	Cible	Dernière exécution	Prochaine exécution	Dernière mise à jour
auto-maj-bch	Opération réussie	us-central1	Activée	Mise à jour des données en temps réel	***** (Europe/London)	Sujet : projects/iconic-parsec-480518-j8/topics/trigger-bch-update	17 déc. 2025, 19:13:00	17 déc. 2025, 19:14:00	17 déc. 2025, 18:20:38
schedule-finance-update	Opération réussie	europe-west1	Activée		***** (America/Scoresbysund)	Sujet : projects/iconic-parsec-480518-j8/topics/cloud-workflows-events	17 déc. 2025, 19:13:04	17 déc. 2025, 19:14:04	8 déc. 2025, 11:55:27

Illustration 24 : Créer un job qui va automiser l'exécution du script python.

Dans le but de ne pas décourager le lecteur avec des images à n'en plus finir nous allons donner les étapes qui concernent la configuration du job d'automatisation.

1. **Initialisation du job dans Cloud Scheduler** : Recherchez le service **Cloud Scheduler** dans votre console et cliquez sur **Créer un job** pour définir le déclencheur principal de votre automatisation.
2. **Paramétrage du planning Cron** : Donnez un nom à votre tâche et saisissez la fréquence `* * * * *` dans le champ dédié pour programmer un déclenchement toutes les minutes.
3. **Sélection de la cible Pub/Sub** : Dans la section "Configurer l'exécution", choisissez **Pub/Sub** comme type de cible pour que le signal soit transmis à votre bus de messages.
4. **Liaison avec le sujet (Topic)** : Sélectionnez l'ID du sujet que vous avez créé précédemment, soit trigger-bch-update, afin de connecter le planificateur à votre fonction.
5. **Finalisation du message de déclenchement** : Saisissez une chaîne de caractères simple (comme le mot update) dans le corps du message et validez la création du job pour activer le cycle automatique.

Services de Cloud Platform utilisés pour la réalisation du projet



Analyse des Contraintes Techniques et Limites de la Solution

L'implémentation de ce pipeline analytique a mis en lumière certaines limites structurelles inhérentes aux outils sélectionnés. Bien que nous ayons dépassé les restrictions des *Scheduled Queries* classiques en adoptant une architecture plus agile basée sur **Cloud Functions** orchestrée par **Cloud Pub/Sub** (ou Cloud Scheduler), une latence incompressible persiste. En effet, même avec cette automatisation avancée, la granularité de déclenchement minimale reste techniquement fixée à une minute. Cette contrainte transforme notre flux en « **Near Real-Time** » (presque temps réel) plutôt qu'en un temps réel pur à la seconde, le traitement s'opérant par micro-lots fréquents.

Ce défi temporel se propage également à la couche de visualisation. **Looker Studio**, en plus de ses restrictions ergonomiques (absence de personnalisation CSS/JS due à sa nature "No-Code"), ne propose pas de rafraîchissement natif instantané (streaming). La mise à jour des données sur le tableau de bord n'est pas automatique en temps réel ; elle reste tributaire d'un cycle de rafraîchissement qui avoisine, dans le meilleur des cas, la minute. Nous sommes ainsi contraints de naviguer entre la simplicité opérationnelle de l'écosystème Google et les exigences de réactivité immédiate et de précision visuelle d'une plateforme BI sur mesure.

Perspectives et Évolutions Futures

Ce projet pilote a permis de valider la faisabilité d'un pipeline analytique sur Google Cloud Platform, tout en soulignant la nécessité d'évolutions techniques pour atteindre un véritable temps réel industriel. Pour franchir le mur du « **Near Real-Time** » imposé par l'orchestration actuelle via Cloud Functions, une transition vers une architecture de streaming événementiel native serait la prochaine étape logique. L'intégration de **Google Cloud Dataflow** (basé sur Apache Beam) couplée à un véritable flux **Pub/Sub**

permettrait de traiter les données transaction par transaction, éliminant ainsi les boucles de simulation artificielles.

Sur le plan de la visualisation, les limitations de rafraîchissement de Looker Studio suggèrent une migration vers des solutions plus dynamiques pour les besoins opérationnels critiques. Le développement d'une application web sur mesure (via **Streamlit** ou **React**) ou l'utilisation d'outils de monitoring comme **Grafana** offrirait un contrôle total sur le rafraîchissement des données (WebSocket) et l'expérience utilisateur, permettant une réactivité à la seconde près.

Enfin, l'enrichissement analytique constitue un axe de développement majeur. Au-delà de la simple visualisation descriptive des transactions passées, l'utilisation de **BigQuery ML** permettrait d'intégrer des modèles prédictifs directement dans l'entrepôt de données. Cela ouvrirait la voie à la détection d'anomalies en temps réel (fraude, mouvements de « Whales ») ou à la prévision des volumes de transactions à court terme, transformant ainsi ce tableau de bord de suivi en un véritable outil d'aide à la décision proactif.

Conclusion

Le projet « **Utilisation de BigQuery pour des rapports BI en temps réel** » marque une étape clé dans l'exploitation des données massives de la blockchain à des fins de Business Intelligence. À travers cette implémentation, nous avons démontré qu'il est possible de transformer un flux de données brut et complexe en un véritable outil d'aide à la décision dynamique.

Pour atteindre cet objectif, nous avons mis en place un pipeline ELT automatisé sur Google Cloud Platform, permettant de traduire l'activité brute du réseau en indicateurs stratégiques tels que le « Whale Ratio » ou la vitesse de circulation. Cette approche analytique a ensuite été directement connectée à Looker Studio en mode *Direct Query*, offrant ainsi aux investisseurs une visibilité immédiate et intuitive sur les mouvements de liquidités.

Toutefois, la réalisation de ce projet a dû faire face à des défis de taille, notamment l'obsolescence des données publiques de la base crypto_bitcoin_cash depuis fin 2024 et les coûts prohibitifs des flux privés. C'est ici que notre solution se distingue par son ingéniosité : en utilisant Google Cloud Run pour orchestrer une translation de données par fenêtre glissante, nous avons réussi à contourner ces obstacles techniques et financiers.

Cette architecture de simulation permet ainsi de maintenir un rafraîchissement toutes les cinq minutes, prouvant qu'une infrastructure BI peut être à la fois puissante, réactive et économiquement viable. Ce qu'il faut retenir de cette expérience, c'est que la valeur de la donnée réside moins dans sa disponibilité brute que dans la capacité à l'orchestrer intelligemment pour répondre à des besoins métiers concrets.

En conclusion, ce système constitue une base solide et évolutive, prête à être connectée à des flux réels dès que nécessaire, offrant ainsi aux décideurs une longueur d'avance stratégique dans l'écosystème volatil des cryptomonnaies. L'intégralité du code source et les configurations détaillées sont disponibles pour consultation sur notre repository GitHub.

 [Lien vers le repository GitHub du projet](#)

Références Bibliographiques

Dans le cadre de la réalisation technique du projet, les ressources webographiques suivantes ont été exploitées :

- [Intégration Google BigQuery et Looker](#) : Un guide détaillé sur l'architecture de connexion entre l'entrepôt de données et l'outil de visualisation pour une analyse fluide.
- [Optimisation des rapports BigQuery dans Looker Studio](#) : Études de cas et meilleures pratiques pour structurer ses données BigQuery afin d'obtenir des tableaux de bord performants.
- [Premiers pas avec Google Cloud Platform](#) : Documentation officielle pour configurer l'environnement cloud et comprendre les fondamentaux de l'infrastructure GCP.
- [Introduction à Google BigQuery](#) : Présentation complète du moteur de base de données serverless, de son fonctionnement et de ses capacités d'analyse de données massives.
- [Guide de démarrage rapide Looker Studio](#) : Manuel d'utilisation pour créer ses premiers rapports et explorer les fonctionnalités de design de l'outil BI.
- [Connecter Looker Studio à Google BigQuery](#) : Procédure technique étape par étape pour établir la liaison entre les tables SQL et l'interface de visualisation.

- [Exécution et optimisation des requêtes SQL](#) : Documentation sur la syntaxe et les méthodes pour lancer des requêtes performantes au sein de la console BigQuery.

Ce document a été rédigé par :

Alioune Abdou Salam KANE

Khadidiatou COULIBALY

Fromo Francis HABA

Ameth FAYE

Awa DIAW



**Documentation
technique**