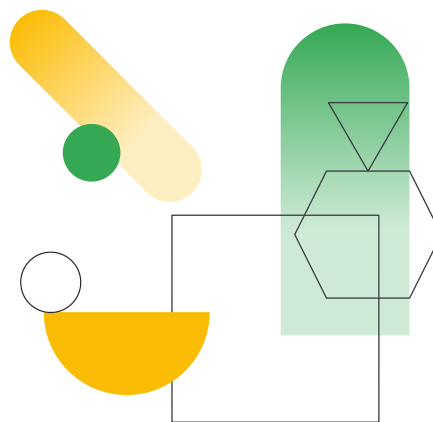




Google Cloud Architect – Livret d'exercices sur la conception et les processus



ClickTravel



En route pour le cloud !

1b. ClickTravel

Brève description

ClickTravel est une agence de voyages internationale qui souhaite créer une plate-forme d'e-commerce évolutive pour répondre à la demande d'une clientèle mondiale.

Indiquer quelques-unes des caractéristiques principales

- Les voyageurs peuvent rechercher et réserver des voyages (hôtels, vols, trains, voitures).
- La tarification sera individualisée en fonction des préférences du client et de la demande.
- La plate-forme doit être fortement intégrée aux réseaux sociaux pour permettre l'utilisation d'avis, de posts et de données analytiques.
- Les fournisseurs (compagnies aériennes, hôtels, etc.) peuvent importer leur inventaire.

Indiquer le rôle des utilisateurs types

- | | |
|------------|----------------------------|
| • Client | • Fournisseur d'inventaire |
| • Voyageur | • Responsable |

ClickTravel est une agence de voyages internationale qui souhaite créer une plate-forme d'e-commerce évolutive pour répondre à la demande d'une clientèle mondiale.

2a. Rédiger les personas utilisateur

Karen

Karen est une femme d'affaires très occupée qui aime s'offrir des week-ends de luxe, souvent réservés à la dernière minute. Elle réserve généralement un hôtel et un vol. Les recommandations jouent un rôle majeur dans le choix de Karen, tout comme les commentaires des clients. Karen aime réaliser toutes les opérations sur son téléphone.

Voici quelques exemples de personas pour notre portail de voyages en ligne.

Karen est une femme d'affaires très occupée qui aime s'offrir des week-ends de luxe, souvent réservés à la dernière minute. Elle réserve généralement un hôtel et un vol. Les recommandations jouent un rôle majeur dans le choix de Karen, tout comme les commentaires des clients. Karen aime réaliser toutes les opérations sur son téléphone.

2a. Rédiger les personas utilisateur

Andrew

Andrew est un étudiant qui aime rentrer chez lui rendre visite à ses parents. Il prend aussi des vacances deux fois par an. Il se préoccupe surtout du coût et il réservera toujours le voyage le moins cher, sans tenir compte de la commodité. Andrew n'est pas fidèle et passera par le marchand qui lui proposera la meilleure offre.

Andrew est un étudiant qui aime rentrer chez lui rendre visite à ses parents. Il prend aussi des vacances deux fois par an. Il se préoccupe surtout du coût et il réservera toujours le voyage le moins cher, sans tenir compte de la commodité. Andrew n'est pas fidèle et passera par le marchand qui lui proposera la meilleure offre.

2b. Rédiger les exigences utilisateurs

Rechercher un vol et un hôtel

En tant que voyageur, **je veux** rechercher une offre vol + hôtel pour une destination aux dates de mon choix **afin de** trouver le meilleur prix.

2b. Rédiger les exigences utilisateurs

Fournir l'inventaire hôtelier

En tant qu'hôtelier, je veux fournir mon inventaire hôtelier de manière groupée **afin que** ClickTravel se charge d'en faire la vente.

2b. Rédiger les exigences utilisateurs

Analyser les performances des ventes

En tant que responsable ClickTravel, **je veux** analyser les données de performances des ventes de tous nos fournisseurs **afin d'**identifier ceux qui ont de mauvais résultats et de les aider à s'améliorer.

3. Définir des SLI et des SLO

En tenant compte des exigences de votre étude de cas, indiquez des SLO et des SLI dans le tableau ci-dessous.

Récit utilisateur	SLO	SLI
Rechercher un vol et un hôtel	Disponibilité de 99,95 %	Part de réponses HTTP 200 vs 500 transmises par le point de terminaison de l'API (mesurée par mois)
Rechercher un vol et un hôtel	95 % des requêtes effectuées en moins de 200 ms	Temps écoulé jusqu'au dernier octet des demandes GET mesuré toutes les 15 secondes et agrégé par tranches de 5 minutes
Fournir l'inventaire hôtelier	Taux d'erreur < 0,00001 %	Pourcentage d'erreurs d'importation dans les importations groupées mesuré quotidiennement en fonction d'une métrique personnalisée
Fournir l'inventaire hôtelier	Disponibilité de 99,9 %	Part de réponses HTTP 200 vs 500 transmises par le point de terminaison de l'API (mesurée par mois)
Analyser les performances des ventes	95 % des requêtes effectuées en moins de 10 s	Temps écoulé jusqu'au dernier octet des demandes GET mesuré toutes les 60 secondes et agrégé par tranches de 10 minutes

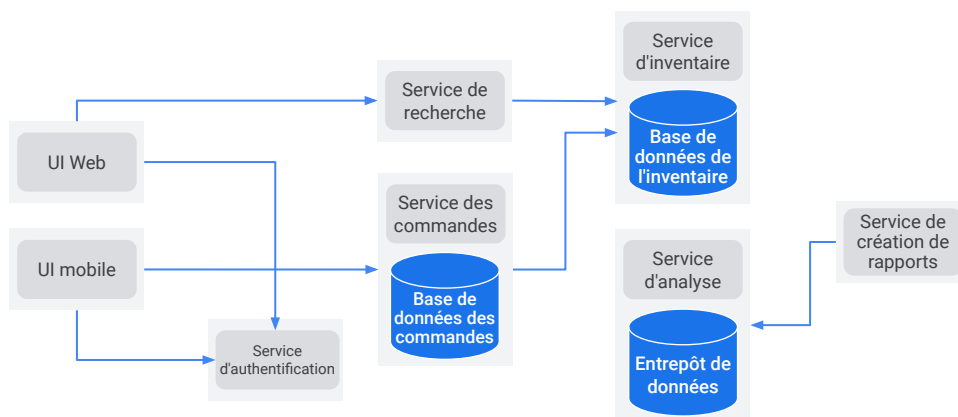
Google Cloud

Voici quelques exemples de SLO et de SLI pour notre application de voyages. Vous remarquerez que le SLI décrit ce que nous allons mesurer et de quelle manière : par exemple, la "part de réponses HTTP 200 vs 500 transmises par le point de terminaison de l'API (mesurée par mois)". Cet exemple est une façon de mesurer la disponibilité.

Le SLO représente l'objectif que nous essayons d'atteindre pour un SLI donné. Par exemple, "disponible 99,95 % du temps".

4. Concevoir des microservices pour votre application

Dessinez un diagramme représentant les microservices de votre application et leurs connexions.



Voici un exemple de diagramme représentant les microservices de notre portail de voyages en ligne. Cela n'est qu'un exemple de représentation. Il y a plusieurs façons de concevoir une application.

Vous remarquerez que nous avons des services séparés pour nos UI Web et mobile. Le service d'authentification est partagé et nous disposons de microservices pour la recherche, les commandes, l'inventaire, l'analyse et la création de rapports. Rappelez-vous que chacun de ces services sera déployé comme une application distincte. Nous souhaitons avoir des services sans état lorsque c'est possible, mais les services des commandes et de l'inventaire auront besoin de bases de données, et le service d'analyse fournira un entrepôt de données.

Cette conception constitue un bon point de départ, que nous pourrions ajuster en fonction des besoins quand nous implémenterons l'application.

5. Concevoir des API REST

Complétez le tableau en indiquant vos services, leurs ressources et leurs opérations.

Nom du service	Collections	Méthodes
Recherche	Voyages (offre vol + hôtel)	find save
Inventaire	Éléments (vols + hôtels)	add search get remove
Analyse	Ventes	analyze get list
Traitement des commandes	Commandes	add get list update

Google Cloud

Voici un exemple pour notre portail de voyages en ligne. Bien entendu, notre API ne se limiterait pas à cela, mais d'une certaine manière, les API reprennent toutes le même schéma. Chaque service gère et met à disposition une collection de données. Les données de chaque collection de données font l'objet d'un certain nombre d'opérations de base.

Ce mode de fonctionnement est semblable à celui des API Google Cloud. Par exemple, dans Google Cloud, le service Compute Engine sert à créer et gérer des machines virtuelles, des réseaux, etc. L'API Compute Engine est dotée de collections telles que les instances, les objets InstanceGroup, les réseaux, les sous-réseaux, etc. Pour chaque collection, différentes méthodes sont employées pour gérer les données.

6. Définir les caractéristiques de stockage








Indiquez les caractéristiques de stockage requises.

Service	Structuré ou non structuré	SQL ou NoSQL	Cohérence forte ou cohérence à terme	Quantité de données (Mo, Go, To, Po, Eo)	Lecture seule ou lecture/écriture
Inventaire	Structuré	NoSQL	Forte	Go	Lecture/écriture
Importations d'inventaire	Non structuré	Non applicable	Non applicable	Go	Lecture seule
Commandes	Structuré	SQL	Forte	To	Lecture/écriture
Analyse	Structuré	SQL	À terme	To	Lecture seule

Voici un exemple pour notre portail de voyages en ligne ClickTravel. Nous nous sommes concentrés sur les services d'inventaire, des importations d'inventaire, des commandes et d'analyse. Comme vous pouvez le constater, chacun de ces services présente des exigences singulières, qui pourraient nous amener à choisir des services Google Cloud différents.

7. Choisir les services de stockage et de données Google Cloud

Choisissez les produits de stockage Google Cloud pour chaque service.

Service	 Persistent Disk	 Cloud Storage	 Cloud SQL	 Firestore	 Cloud Bigtable	 Cloud Spanner	 BigQuery
Inventaire				X			
Importations d'inventaire		X					
Commandes			X				
Analyse							X

Google Cloud

Voici un exemple pour notre portail de voyages en ligne ClickTravel.

Pour le service d'inventaire, nous utiliserons Cloud Storage pour les importations d'inventaire brut. Les fournisseurs importeront leur inventaire sous forme de données JSON stockées dans des fichiers texte. Cet inventaire sera ensuite importé dans une base de données Firestore.

Le service des commandes stockera ses données dans une base de données relationnelle s'exécutant dans Cloud SQL.

Le service d'analyse agrégera les données de différentes sources dans un entrepôt de données. Nous utiliserons BigQuery.

8a. Définir les caractéristiques réseau de vos services

Indiquez les caractéristiques réseau requises.




Service	Web ou interne uniquement	HTTP	TCP	UDP	Multirégional ?
Recherche	Web	X			Oui
Inventaire	Interne		X		Non
Analyse	Web	X			Non
UI Web	Web	X			Oui
Commandes	Interne		X		Non

Voici un exemple pour notre portail de voyages en ligne ClickTravel.

Les services d'inventaire et des commandes sont des services internes et régionaux utilisant le protocole TCP. Les autres services doivent être des services Web utilisant le protocole HTTP. Nous avons fait le choix de les déployer dans plusieurs régions afin de réduire la latence, d'améliorer les performances et d'assurer une haute disponibilité à nos utilisateurs basés dans de nombreux pays du monde entier.

8b. Sélectionner les équilibreurs de charge de vos services

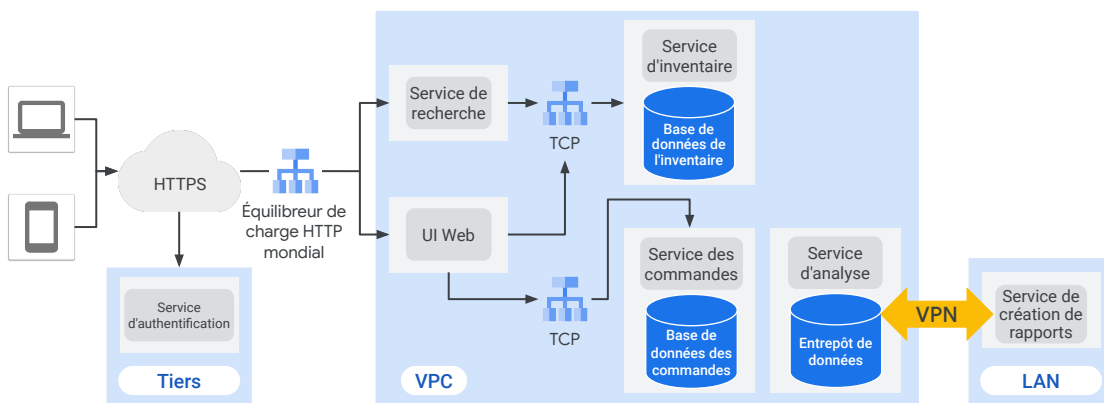
Choisissez le(s) produit(s) d'équilibrage de charge Google Cloud pour chaque service.

Service	 HTTP	 TCP	 UDP
Recherche	X		
Inventaire		X	
Analyse	X		
UI Web	X		
Commandes		X	

Compte tenu des caractéristiques réseau définies, nous avons choisi l'équilibreur de charge HTTP mondial pour nos services publics et l'équilibreur de charge TCP interne pour nos services internes.

9. Créer un diagramme de votre réseau

Dessinez un diagramme montrant comment vos services communiqueront sur le réseau. Incluez les régions, zones, équilibreurs de charge, CDN et DNS le cas échéant.



Google Cloud

Voici un exemple pour notre portail de voyages en ligne ClickTravel.

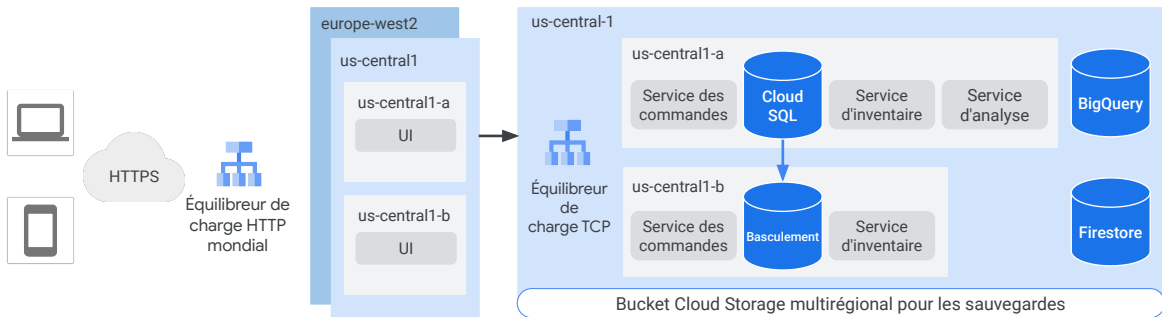
Le trafic utilisateur mobile et Web sera d'abord authentifié à l'aide d'un service tiers, puis un équilibreur de charge HTTP mondial dirigera le trafic vers nos services publics de recherche et de l'UI Web.

Ensuite, des équilibreurs de charge TCP régionaux dirigeront le trafic vers les services internes d'inventaire et des commandes.

Le service d'analyse pourrait utiliser BigQuery comme entrepôt de données. Le service de création de rapports sur site accédera au service d'analyse via un VPN. Cette configuration devrait suffire dans un premier temps, et nous pourrions l'affiner une fois l'implémentation commencée.

10. Concevoir des applications fiables et évolutives

Nous souhaitons que l'interface Web de notre application soit disponible pratiquement tout le temps, même si un service ne fonctionne pas. Nous souhaitons également que le site Web soit rapide avec une latence très faible pour les utilisateurs du monde entier. Dessinez un diagramme montrant comment y parvenir à l'aide de services Google Cloud.



Google Cloud

Dans le cas de notre application de voyages en ligne ClickTravel, je suppose que bien qu'il s'agisse d'une entreprise américaine, un grand nombre de clients se trouvent en Europe.

Je souhaite que l'UI ait une disponibilité élevée. Je l'ai donc placée dans us-central1 et europe-west2, derrière un équilibreur de charge HTTP mondial. Cet équilibreur de charge enverra les requêtes de l'utilisateur dans la région la plus proche de lui, sauf si celle-ci ne peut pas gérer le trafic.

Je pourrais également déployer les backends dans le monde entier. Mais si je cherche à optimiser les coûts, je peux me contenter de les déployer dans us-central1 pour commencer. Cela créera une latence pour les utilisateurs européens, mais je pourrai toujours revenir sur ce choix plus tard et déployer un backend similaire dans europe-west2.

Pour garantir une haute disponibilité, j'ai décidé de déployer les services des commandes et d'inventaire dans plusieurs zones. Étant donné que le service d'analyse n'est pas public, je peux réaliser des économies en ne le déployant que dans une seule zone. J'ai prévu une base de données Cloud SQL de basculement, et je n'ai pas à me préoccuper d'une solution de basculement pour la base de données Firestore et l'entrepôt de données BigQuery qui sont multirégionaux.

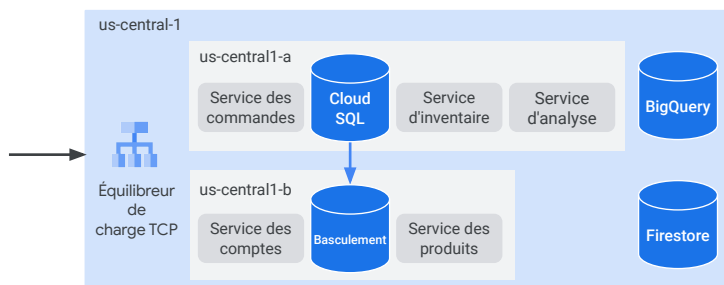
Une solution de basculement est nécessaire pour la base de données Cloud SQL afin d'assurer une haute disponibilité. Comme BigQuery et Firestore sont des services

gérés, nous n'avons pas à nous soucier de ce point.

En cas de sinistre, je conserverai des sauvegardes dans un bucket Cloud Storage multirégional. Ainsi, en cas d'indisponibilité régionale, je pourrai restaurer l'application dans une autre région.

11a. Scénario de reprise après sinistre

Vous avez déployé votre application de façon à garantir une haute disponibilité en dupliquant les ressources dans plusieurs zones. Cependant, pour vous conformer aux exigences réglementaires, vous devez disposer d'un plan de reprise en cas de sinistre qui rendrait indisponible la région entière. Élaborez un plan pour restaurer votre application dans une autre région si votre région principale est indisponible.



11b. Scénarios de reprise après sinistre pour les services

Dressez une liste détaillée des scénarios possibles.

Service	Scénario	Objectif de point de récupération	Objectif de temps de récupération	Priorité
Base de données Firestore de l'inventaire	Le programmeur a supprimé accidentellement tout l'inventaire	1 heure	1 heure	Élevée
Base de données Cloud SQL des commandes	La base de données des commandes a planté	0 minute	5 minutes	Élevée
Base de données BigQuery pour l'analyse	L'utilisateur supprime une table	0 minute	24 heures	Moyenne

Google Cloud

Voici un exemple de scénarios de reprise après sinistre pour notre portail de voyages en ligne ClickTravel.

Chacun de nos services utilise des services de base de données différents, et a des objectifs et des priorités uniques. Tout cela affecte la manière dont nous concevons nos plans de reprise après sinistre.

11c. Plans de reprise après sinistre pour les ressources

Pour chaque scénario, complétez le tableau.

Ressource	Stratégie de sauvegarde	Emplacement de la sauvegarde	Procédure de récupération
Base de données Firestore de l'inventaire	Sauvegardes automatiques quotidiennes	Bucket Cloud Storage multirégional	Cloud Functions et Cloud Scheduler
Base de données Cloud SQL des commandes	Sauvegardes et journalisation binaire Instance dupliquée de basculement dans une autre zone	Non applicable	Basculement automatique Exécution d'un script de sauvegarde si nécessaire
Base de données BigQuery pour l'analyse	Aucune sauvegarde spécifique nécessaire	Non applicable	Réimportation des données pour recréer les tables d'analyse

Google Cloud

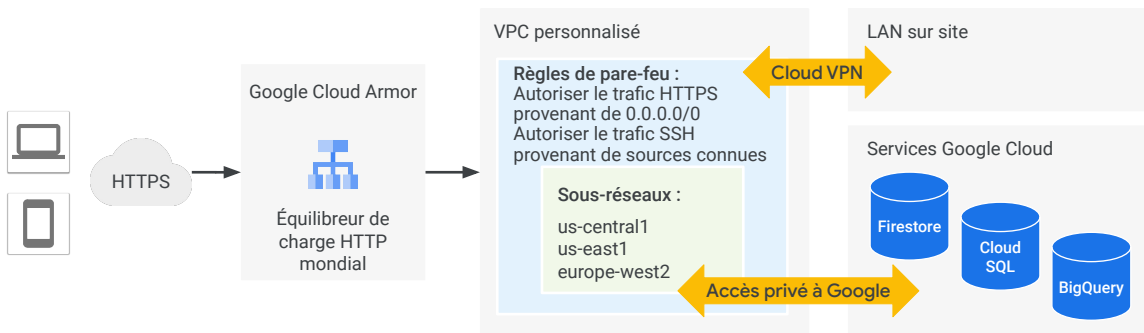
Notre service d'analyse dans BigQuery avait la priorité la plus basse. Par conséquent, nous devrions être en mesure de réimporter les données pour recréer les tables d'analyse si un utilisateur les supprime.

Le service des commandes ne peut tolérer aucune perte de données et doit être opérationnel presque immédiatement. Pour cela, nous avons besoin d'une instance dupliquée de basculement en plus de la journalisation binaire et des sauvegardes automatiques.

Notre service d'inventaire utilise la base de données Firestore pour laquelle nous pouvons implémenter des sauvegardes automatiques quotidiennes vers un bucket Cloud Storage multirégional. Cloud Functions et Cloud Scheduler peuvent simplifier la procédure de récupération.

12. Modéliser des services Google Cloud sécurisés

Dessinez un diagramme montrant comment vous allez sécuriser vos services. Incluez les pare-feu, rôles IAM, comptes de service et ressources réseau nécessaires.



Google Cloud

Pour commencer, j'ai configuré Google Cloud Armor sur un équilibreur de charge HTTP mondial afin de bloquer toutes les adresses IP refusées. Mon réseau VPC personnalisé a des sous-réseaux dans us-central1 pour mes clients américains, ainsi qu'un sous-réseau de secours dans us-east1 et un sous-réseau dans europe-west2 pour mes clients européens.

Mes règles de pare-feu n'autorisent que le trafic SSH provenant de sources connues. Bien que j'autorise le trafic HTTPS d'où qu'il provienne, je peux toujours refuser des adresses IP à l'aide de Google Cloud Armor en périphérie du réseau Google Cloud. J'ai également configuré des tunnels Cloud VPN afin de communiquer de façon sécurisée avec mon réseau sur site pour mon service de création de rapports.

Tandis que mon équilibreur de charge a besoin d'une adresse IP publique, je peux sécuriser mes services de backend en les créant sans adresses IP externes. Pour que ces instances communiquent avec les services de base de données Google Cloud, j'active l'accès privé à Google. Le trafic des services d'inventaire, des commandes et d'analyse reste ainsi privé, et je réduis mes coûts de mise en réseau.

13. Estimation et planification des coûts

Utilisez le [simulateur de coût](#) pour déterminer le coût de vos microservices et consignez-le.

Nom du service	Ressource Google Cloud	Coût
Commandes	Cloud SQL	1 264,44 \$
Inventaire	Firestore	215,41 \$
Inventaire	Cloud Storage	1 801 \$
Analyse	BigQuery	214,72 \$

Google Cloud

Voici une estimation approximative pour les applications de base de données de mon portail de voyages en ligne ClickTravel.

J'ai ajusté ma base de données des commandes afin d'inclure une instance dupliquée de basculement pour garantir une haute disponibilité et j'ai obtenu des estimations globales pour mes autres services. Mon service d'inventaire utilise Cloud Storage pour stocker des données JSON enregistrées dans des fichiers texte. Étant donné qu'il s'agit de mon service le plus cher, je peux envisager de changer de classe de stockage ou configurer la gestion du cycle de vie des objets.

Encore une fois, il ne s'agit que d'un exemple. Vos coûts dépendraient de votre étude de cas.

Cloud SQL : tarif basé sur 1 instance de PostgreSQL dans l'Iowa, 30 Go de stockage SSD, sauvegarde de 1 000 Go, 12 cœurs, 16 Go de RAM

Firestore : tarif basé sur un emplacement multirégional aux États-Unis, 200 000 lectures par jour, 10 000 écritures, 0 suppression et 1 000 Go stockés

Cloud Storage : tarif basé sur un hébergement dans l'Iowa (us-central), 50 To de stockage, 200 000 opérations de classe A et 200 000 opérations de classe B

BigQuery : tarif basé sur un emplacement multirégional aux États-Unis, 10 To de stockage, 100 Go de données par flux chaque mois, 2 To de requêtes

