Pour une formation en personne, le cas échéant, se présenter et demander aux participants de faire de même.

Programme

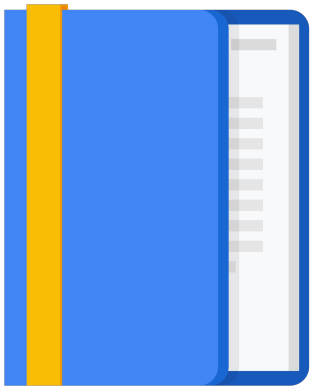
Traiter des flux de données

Cloud Pub/Sub

Fonctionnalités de traitement par flux Cloud Dataflow

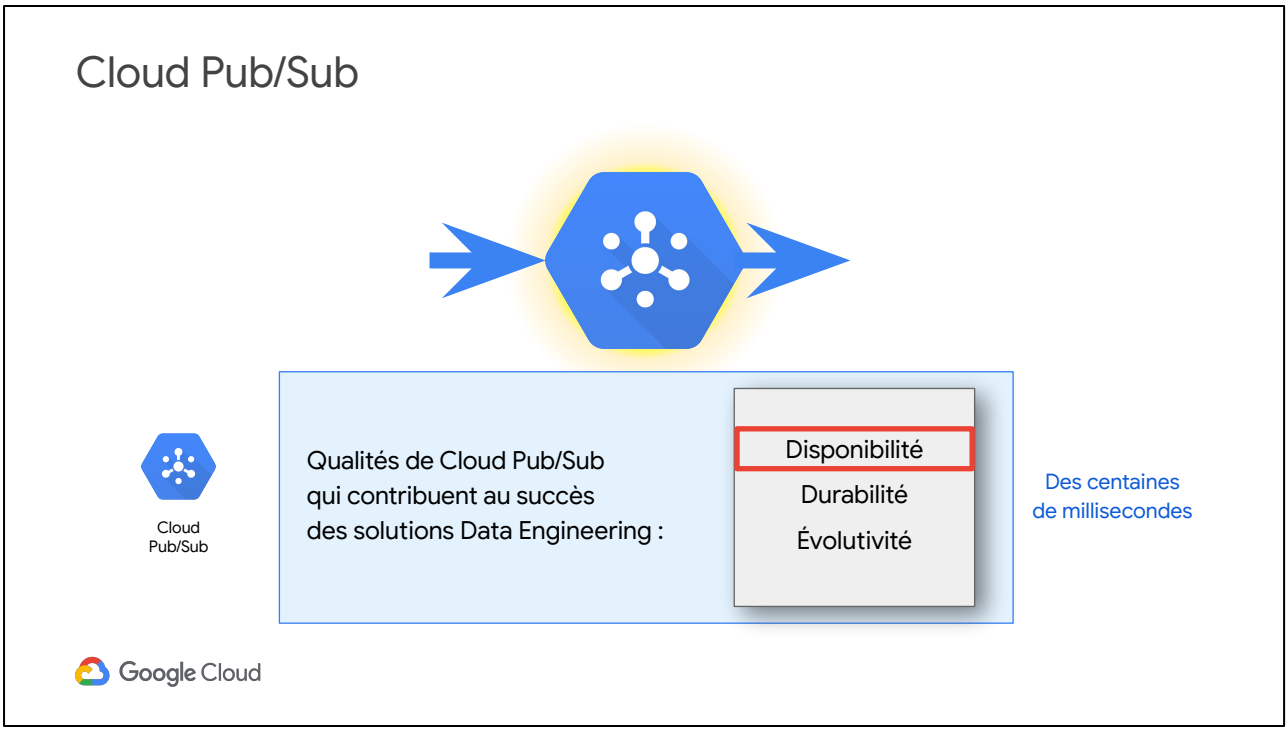
Fonctionnalités de traitement par flux BigQuery

Cloud Bigtable



Google Cloud

Maintenant que vous maîtrisez les flux de données, intéressons-nous au fonctionnement de Cloud Pub/Sub. Pour commencer, je vais vous demander de faire preuve d'ouverture d'esprit face à de nouvelles méthodes de travail. Cloud Pub/Sub gère les flux de données différemment de ce à quoi vous êtes habitué. Vous n'avez sans doute encore jamais rencontré ce modèle.

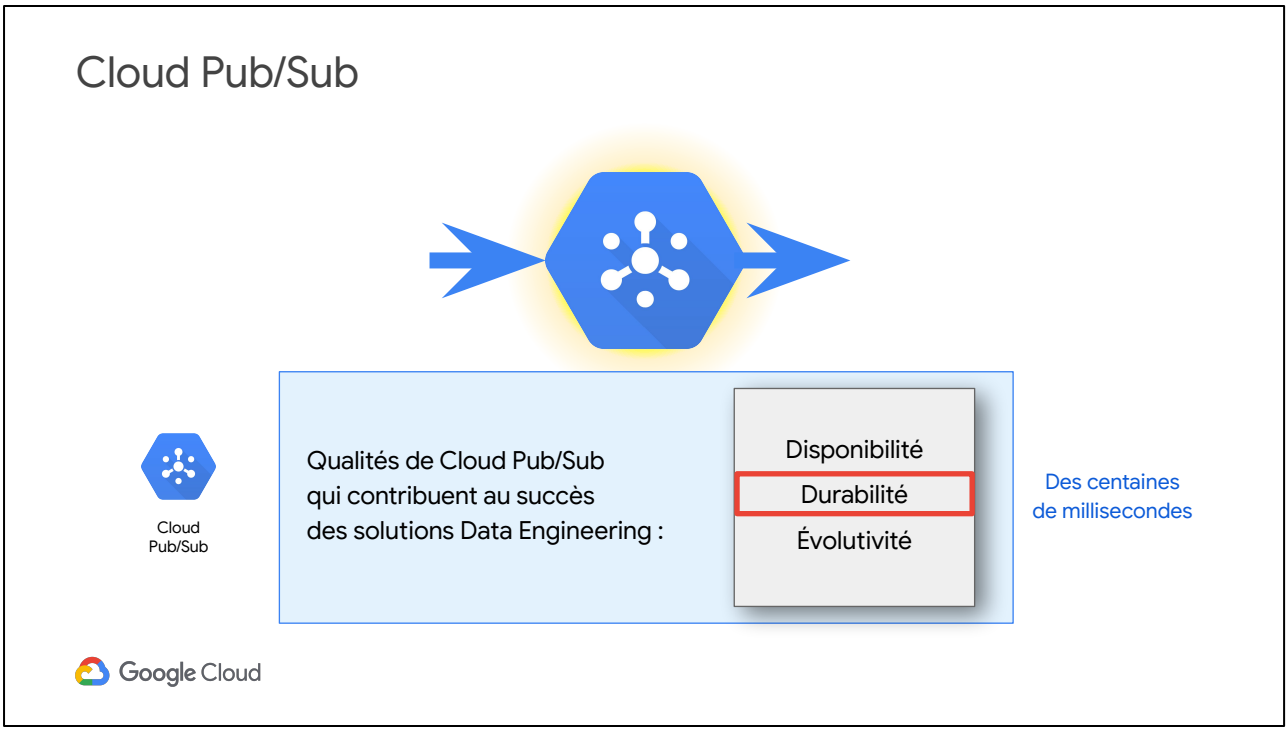


Cloud Pub/Sub est un système de distribution des données entièrement géré. Il peut remplir de nombreuses fonctions, mais sert le plus souvent à relier plusieurs éléments d'un système de manière souple. Vous pouvez utiliser Cloud Pub/Sub pour connecter des applications dans Google Cloud, et pour créer des solutions Data Engineering hybrides à partir d'applications hébergées sur site et dans d'autres clouds. Les applications n'ont pas besoin d'être en ligne et disponibles en permanence. Quant aux éléments à relier, ils n'ont pas besoin de savoir comment communiquer entre eux, mais seulement avec Cloud Pub/Sub, ce qui simplifie la conception du système.

Premièrement, Pub/Sub est un service, pas un logiciel. Aussi, comme les autres services sans serveur que nous avons abordés, vous pouvez l'utiliser sans avoir à installer quoi que ce soit. Il s'agit d'un pur service.

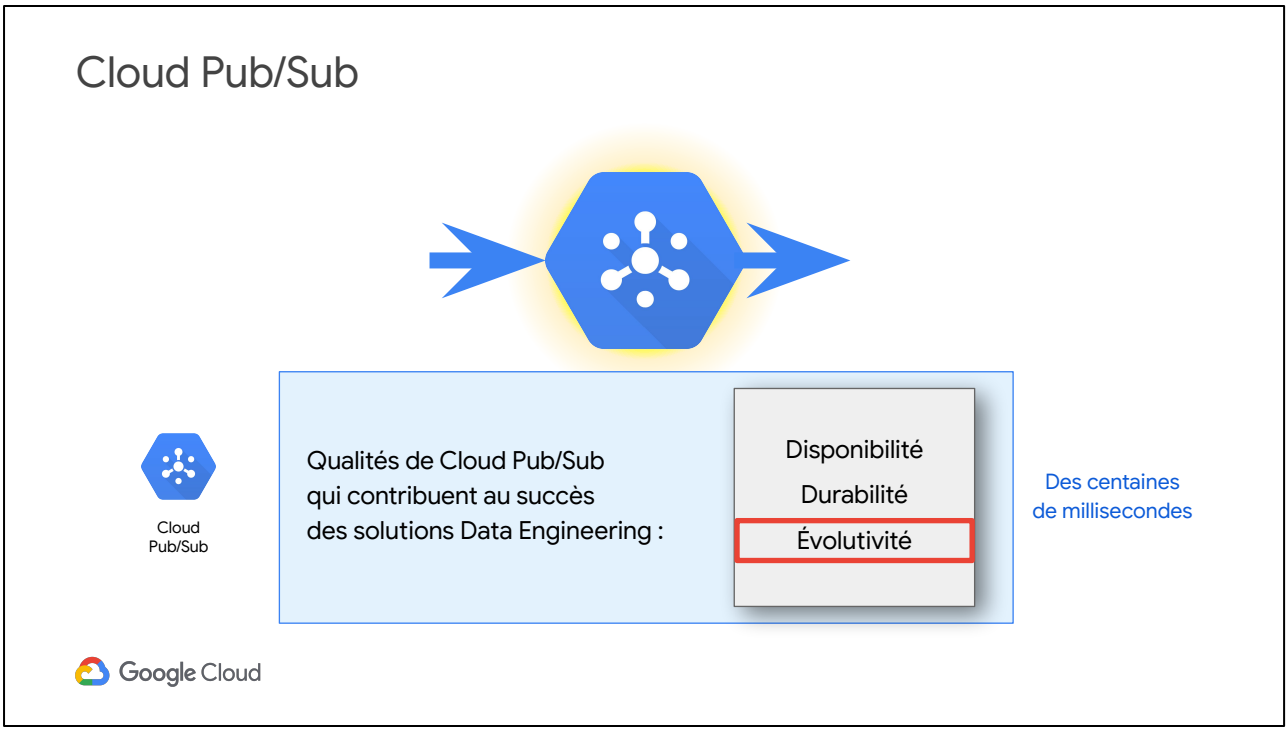
Les bibliothèques clientes Cloud Pub/Sub sont disponibles en C#, GO, Java, Node.js, Python et Ruby. Elles encapsulent plusieurs appels d'API REST, quel que soit leur langage.

Cloud Pub/Sub offre une disponibilité élevée.



Cloud Pub/Sub offre une messagerie durable. Par défaut, vos messages sont enregistrés pendant sept jours, au cas où vos systèmes s'arrêteraient et seraient dans l'incapacité de les traiter.

https://pixabay.com/fr/illustrations/chess-black-and-white-pieces-3413429/



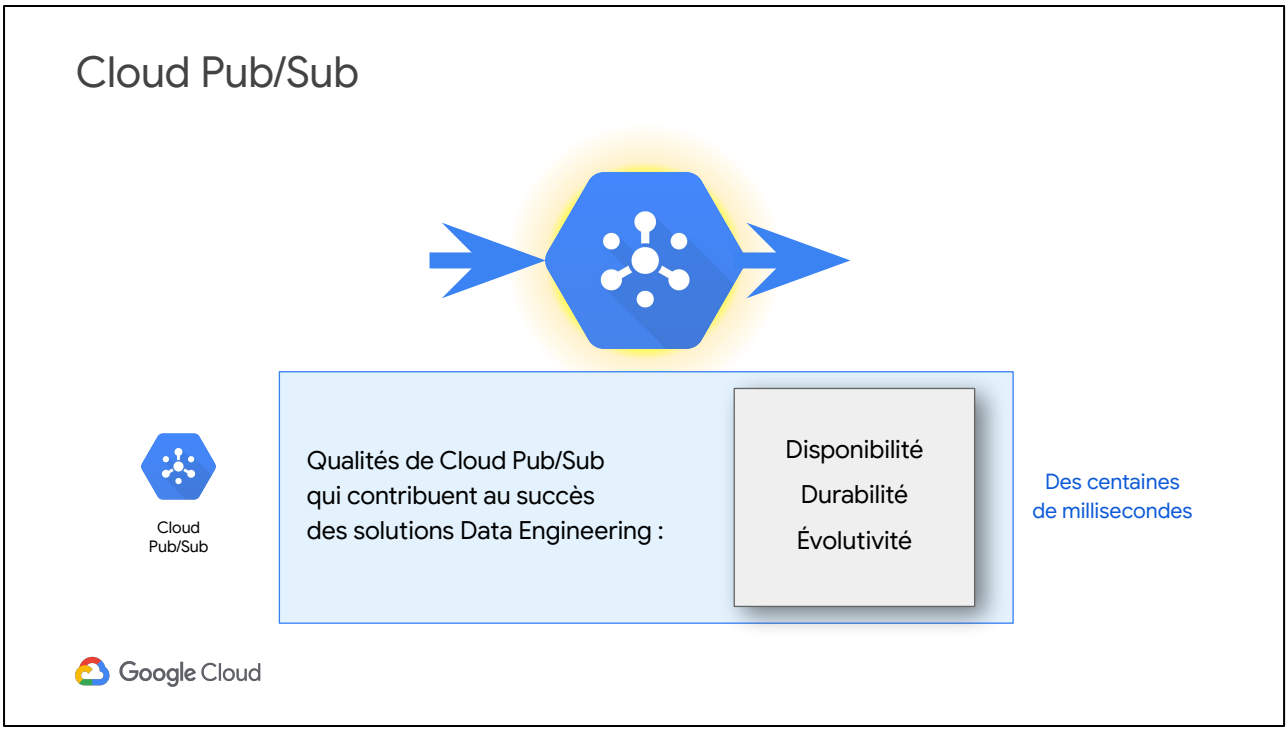
Enfin, Cloud Pub/Sub offre une haute évolutivité. Google traite en interne environ 100 millions de messages par seconde sur l'ensemble de l'infrastructure. C'est l'une des raisons pour lesquelles Pub/Sub a fait son apparition assez tôt chez Google, pour pouvoir proposer le moteur de recherche et l'index partout dans le monde étant donné que nous conservons les copies locales des recherches à l'échelle mondiale pour, vous l'imaginez bien, pouvoir renvoyer les résultats avec le minimum de latence.

Comment articuler tout cela ? Comme nous explorons l'ensemble de la toile mondiale, nous devons l'envoyer dans son intégralité, et aux quatre coins du globe. Il nous faut donc plusieurs robots d'exploration à différents endroits de la planète, mais cela pose des problèmes de cohérence au niveau des données, car chaque robot recevrait un index différent.

Pub/Sub est LA solution de distribution que nous privilégions. Le robot d'exploration récupère chaque page du Web, laquelle est ensuite envoyée sous forme de message sur Pub/Sub et sélectionnée par toutes les copies locales de l'index de recherche en vue de son indexation.

Actuellement, notre fréquence d'indexation peut aller de deux semaines (la plus forte latence) à plusieurs fois par heure, pour les sites d'actualités très populaires. En moyenne, Google indexe donc le Web trois fois par jour. Autrement dit, chaque jour, nous envoyons trois fois l'intégralité du Web sur Pub/Sub.

Voilà la preuve de l'évolutivité de Pub/Sub.

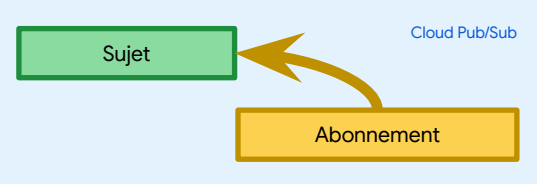


Cloud Pub/Sub est un service conforme à la loi HIPAA, offrant un contrôle ultraprécis des accès et un chiffrement de bout en bout. Les messages sont chiffrés en transit et au repos. Ils sont stockés à plusieurs endroits dans un souci de durabilité et d'évolutivité.

Vous contrôlez les qualités de votre solution Cloud Pub/Sub en fonction du nombre d'éditeurs, d'abonnés et de messages, et de la taille des messages. Ces facteurs permettent d'allier évolutivité élevée, faible latence et haut débit.

https://pixabay.com/fr/illustrations/chess-black-and-white-pieces-3413429/

Exemple d'application Cloud Pub/Sub



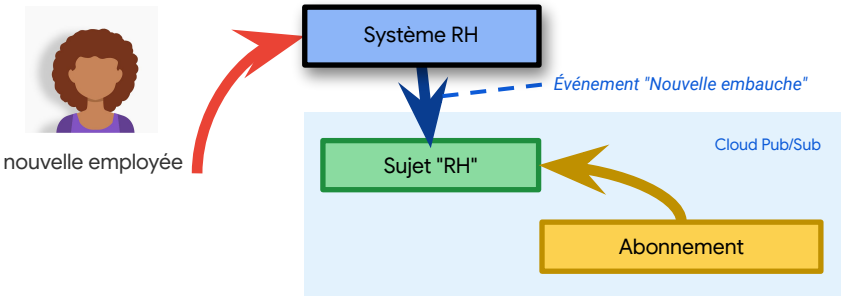
Google Cloud

Comment Pub/Sub fonctionne-t-il ? Le modèle, très simple, s'articule autour de deux structures de données : un sujet et un abonnement. Les deux coexistent dans le framework Pub/Sub indépendamment des nœuds de calcul, des abonnés et des autres éléments. Le client Cloud Pub/Sub qui crée le sujet est l'éditeur, celui qui crée l'abonnement est l'abonné.

Dans cet exemple, l'abonnement est abonné au sujet.

<FORMATEUR>
Pour recevoir des messages publiés dans un sujet, vous devez créer un abonnement associé à ce sujet. Seuls les messages publiés dans le sujet après la création de l'abonnement sont disponibles pour les applications d'abonnés. L'abonnement connecte le sujet à une application d'abonné, qui reçoit et traite les messages publiés dans le sujet. Un sujet peut être associé à plusieurs abonnements, mais un abonnement ne peut être associé qu'à un seul sujet.
</FORMATEUR>

L'arrivée d'une nouvelle employée déclenche un événement "Nouvelle embauche"



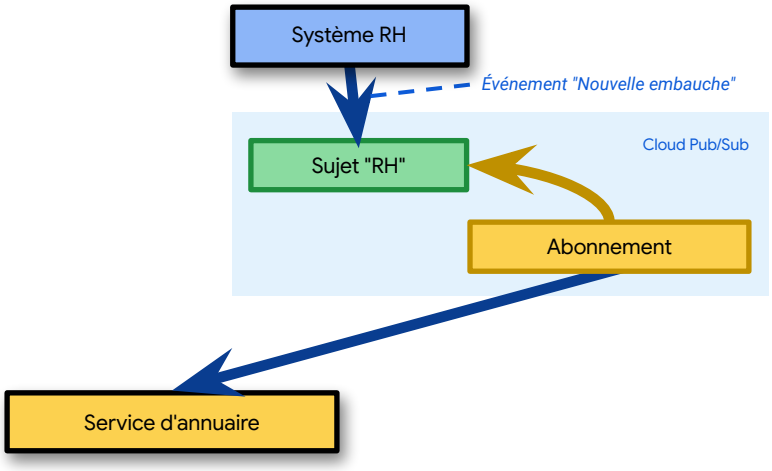
Google Cloud

Cet exemple correspond à un service de messagerie d'entreprise. Quel est son fonctionnement ?

Dans le cas présent, un sujet "RH" est associé à l'événement "Nouvelle embauche". Lorsque votre entreprise accueille un nouveau collaborateur, par exemple, cette notification permet aux autres applications qui doivent être informées des nouvelles embauches de s'abonner à ce sujet et d'obtenir ce message. Quelles applications peuvent vous informer de l'arrivée d'un nouveau membre du personnel ?

https://pixabay.com/fr/illustrations/avatar-clients-customers-icons-2155431/

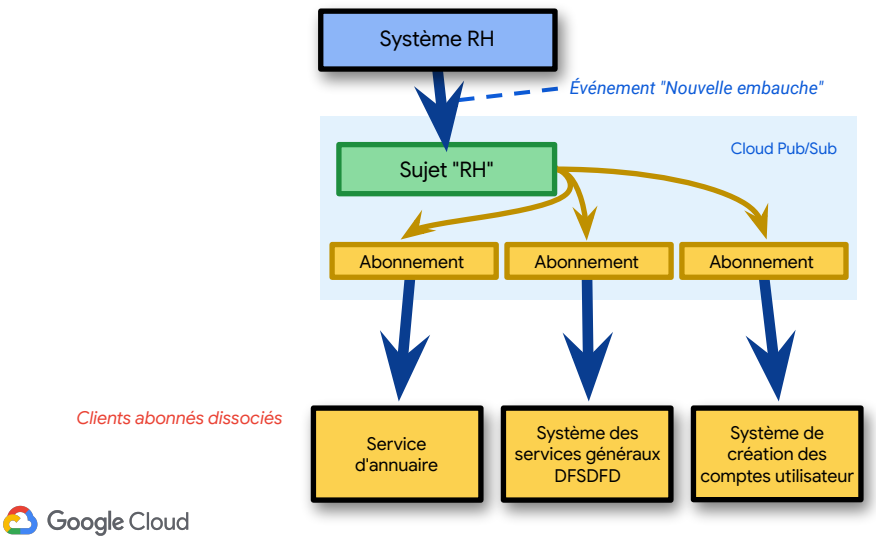
Le message est envoyé du sujet à l'abonnement



Google Cloud

Le service d'annuaire de l'entreprise en est un exemple. Il s'agit d'un client de l'abonnement, c'est-à-dire un abonné. Toutefois, vous n'êtes pas limité à un abonné ni à un abonnement dans Cloud Pub/Sub.

Il peut y avoir plusieurs abonnements par sujet



Google Cloud

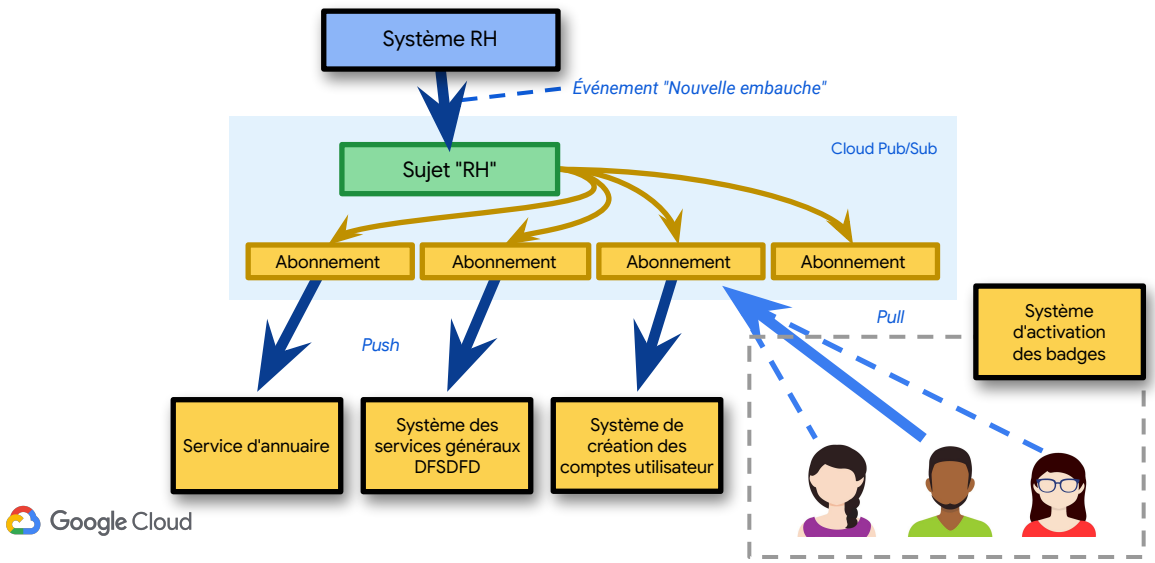
Cet exemple montre plusieurs abonnements et plusieurs abonnés. La nouvelle embauche doit sans doute être signalée au système des services généraux (pour la création du badge) et au système de création des comptes utilisateur (pour la paie).

Chaque abonnement garantit que le message sera transmis au service.

Les clients abonnés sont dissociés les uns des autres et isolés de l'éditeur. Nous verrons plus tard que si le système RH se déconnecte après avoir envoyé le message au sujet "RH", le message sera tout de même transmis aux abonnés.

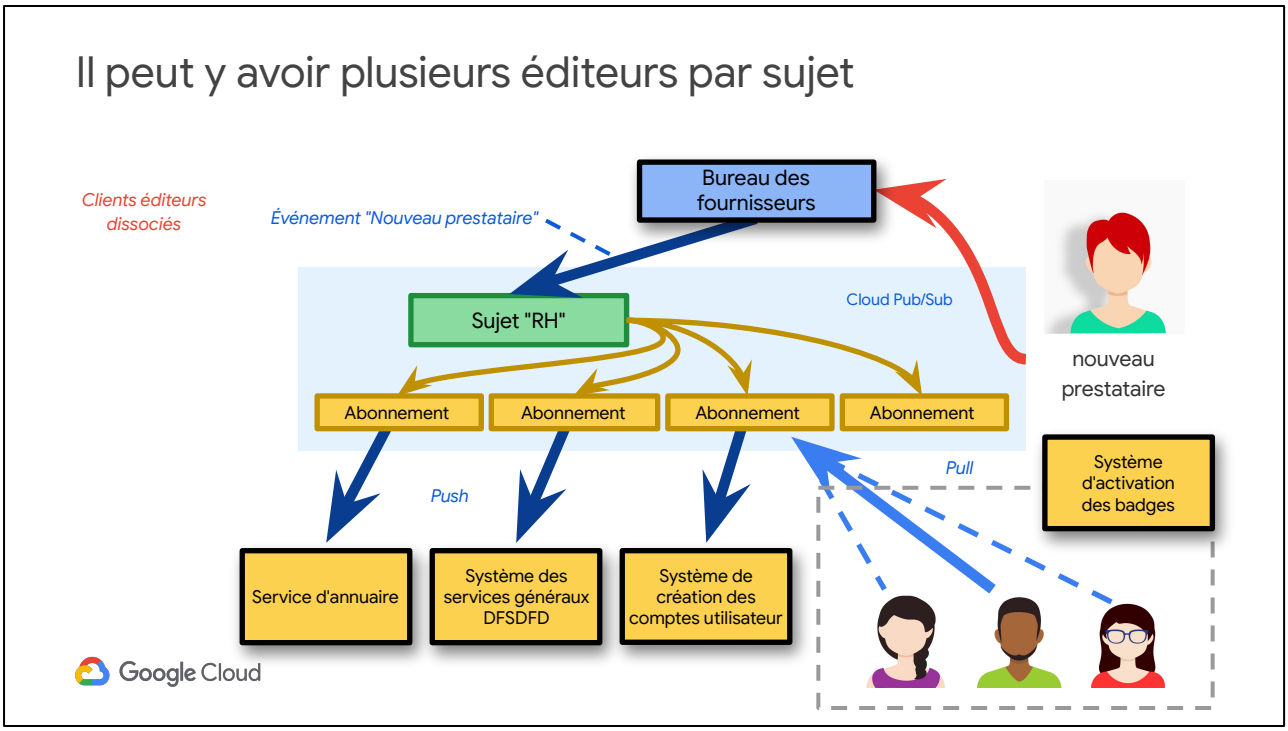
Dans cet exemple, chaque abonnement est associé à un abonné, mais il est possible d'avoir plusieurs abonnés par abonnement.

Il peut y avoir plusieurs abonnés par abonnement



Dans cet exemple, le système d'activation des badges nécessite l'intervention d'un employé pour activer le badge. L'entreprise compte plusieurs employés, mais ils ne sont pas tous constamment disponibles.
Cloud Pub/Sub transmet le message à l'ensemble des employés, en sachant qu'une seule personne devra le récupérer et le traiter. C'est ce qu'on appelle un abonnement pull, en opposition à l'abonnement push.

https://pixabay.com/fr/illustrations/avatar-clients-customers-icons-2191918/



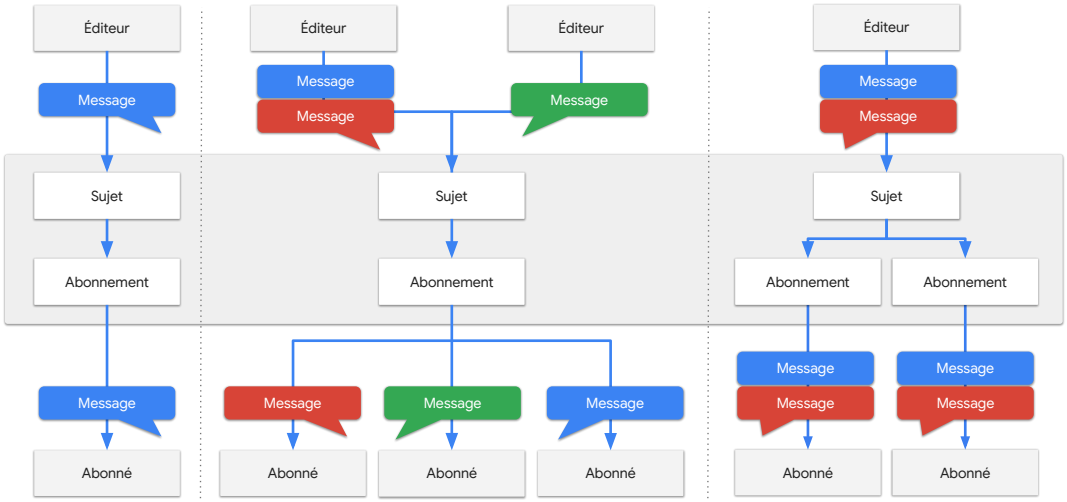
À présent, l'entreprise fait appel à un nouveau prestataire. Il passe par le bureau des fournisseurs, et non par le système RH. La marche à suivre est la même. Cette personne doit être enregistrée dans l'annuaire de l'entreprise. Le système des services généraux doit lui affecter un bureau. Le système de création des comptes utilisateur doit configurer son identité d'entreprise et ses comptes, et le système d'activation des badges doit imprimer et activer son badge de prestataire. Le bureau des fournisseurs peut publier un message dans le sujet "RH". Le bureau des fournisseurs et le système RH sont entièrement dissociés, mais peuvent faire appel aux mêmes services dans l'entreprise.

Cette diapositive montre l'importance du processus Pub/Sub. C'est pourquoi il a la priorité la plus élevée.

Vous savez maintenant à quoi sert Cloud Pub/Sub. Découvrez à présent son fonctionnement et ses nombreuses fonctionnalités avancées.

https://pixabay.com/fr/illustrations/avatar-clients-customers-icons-2191918/

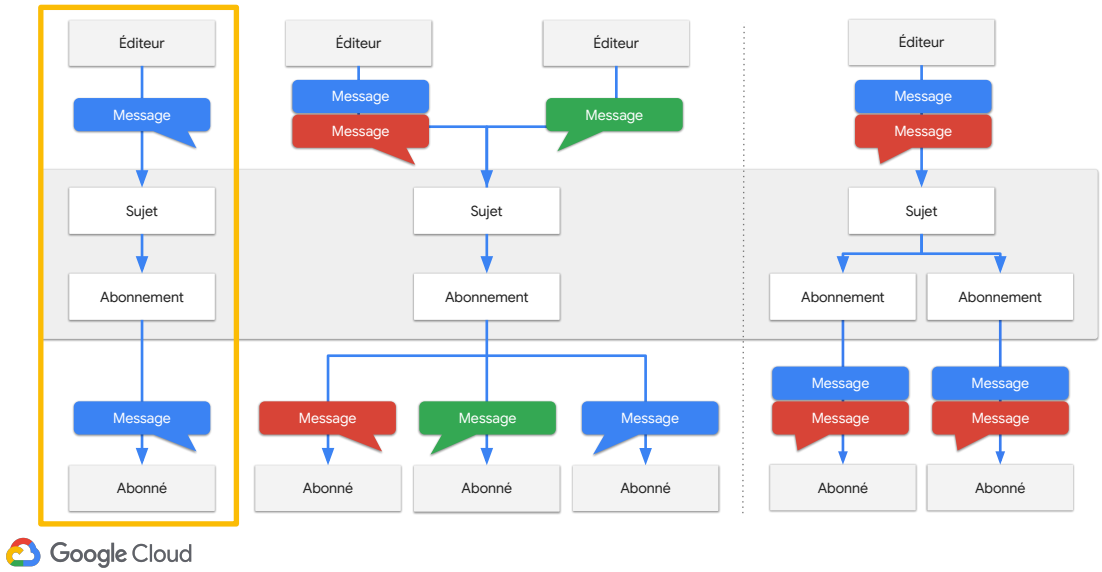
Modèles Pub/Sub



Google Cloud

Les modèles de publication et d'abonnement présentés ici permettent de distribuer les messages selon un schéma unique ou ramifié. Les couleurs représentent des messages différents.

Modèles Pub/Sub



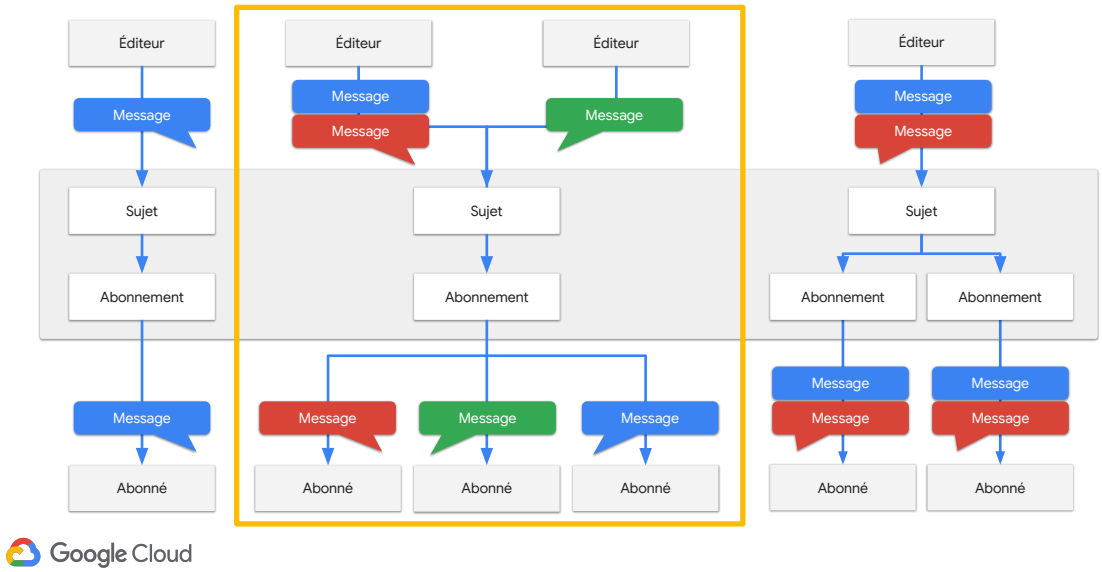
Google Cloud

Ceci est un modèle de base avec distribution directe, correspondant à une file d'attente.

(Distribution unique/équilibrage de charge) Plusieurs éditeurs publient un message dans le même sujet.
Ou plusieurs abonnés extraient des messages associés aux mêmes abonnements, traitement en parallèle de base. Dataflow,
un abonnement avec plusieurs clients, chaque abonné recevant un sous-ensemble des messages dans le cadre de l'abonnement.

(Distribution ramifiée) Plusieurs abonnés, avec plusieurs cas d'utilisation pour les mêmes données et où l'ensemble des données est envoyé à des abonnés différents.

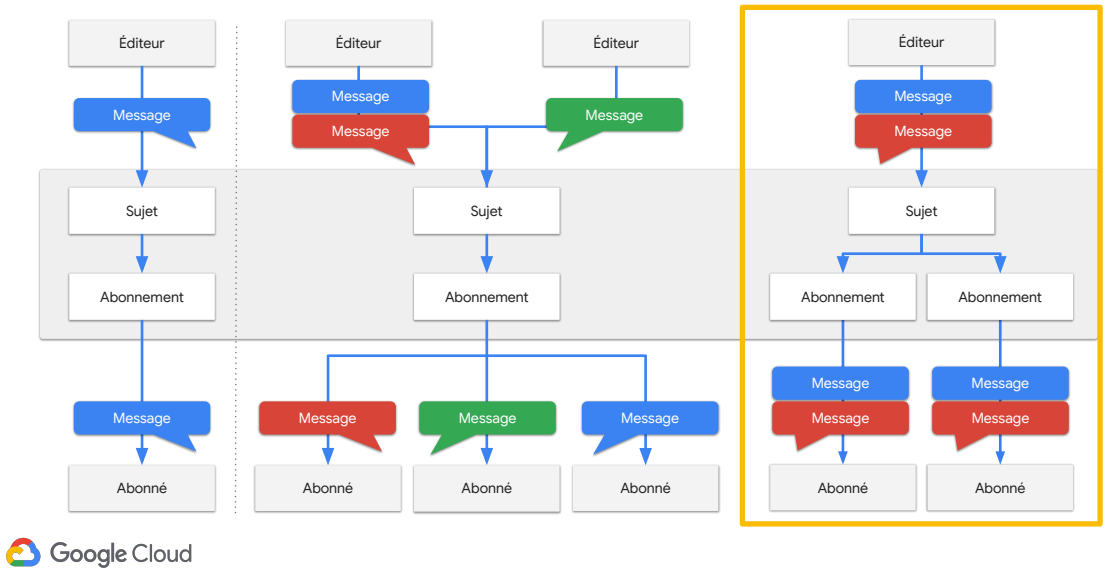
Modèles Pub/Sub



Google Cloud

Le deuxième exemple montre trois messages différents envoyés par deux éditeurs distincts concernant le même sujet. L'abonnement recevra donc les trois messages.

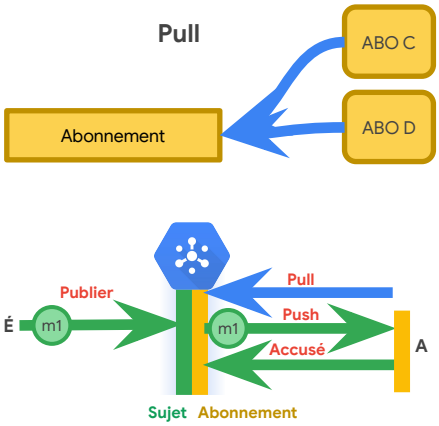
Modèles Pub/Sub



Google Cloud

Dans l'exemple de droite, nous avons deux abonnements, qui recevront chacun le message rouge et le message bleu.

Cloud Pub/Sub distribue les messages en mode push ou pull



Google Cloud

Cloud Pub/Sub distribue les messages en mode push ou pull. Dans le modèle pull, les clients sont des abonnés et appellent périodiquement des messages. Pub/Sub ne transmet que les messages reçus depuis le dernier appel.

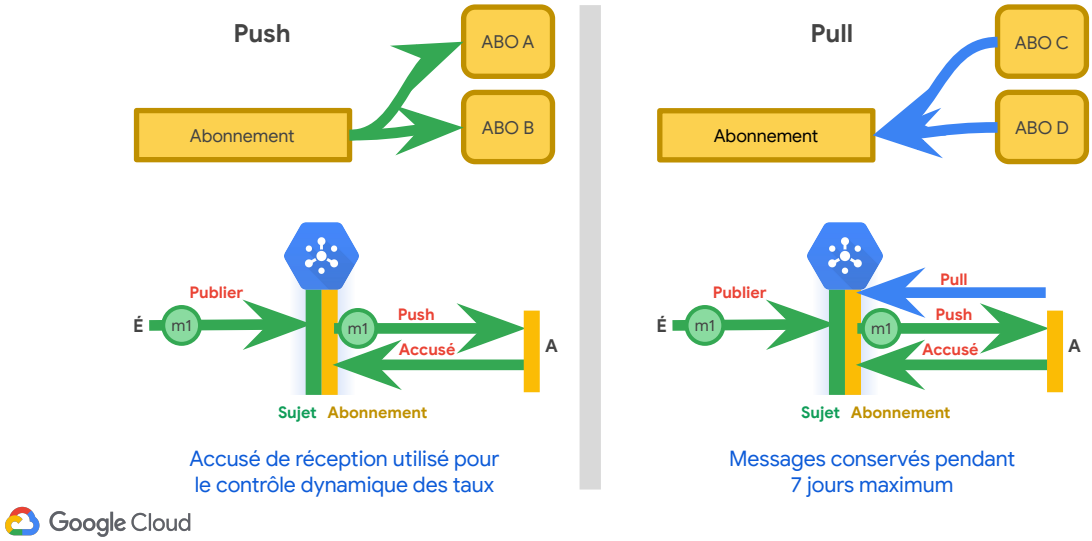
Dans le modèle pull, vous devez accuser réception du message séparément. Comme vous pouvez le voir, après avoir été appelé, l'abonné extrait les messages, puis en renvoie un, dont il accuse réception lors d'une étape distincte.

Dans le modèle pull, les files d'attente de retrait permettent surtout de mettre en place un système de file d'attente pour les tâches à effectuer. Vous ne voulez pas accuser réception du message tant que vous ne l'avez pas effectivement reçu et traité, au risque de le perdre en cas de défaillance du système.

C'est pourquoi il est conseillé d'attendre d'avoir reçu le message pour en accuser réception.

Dans le modèle pull, les messages sont conservés pendant sept jours maximum.

Cloud Pub/Sub distribue les messages en mode push ou pull



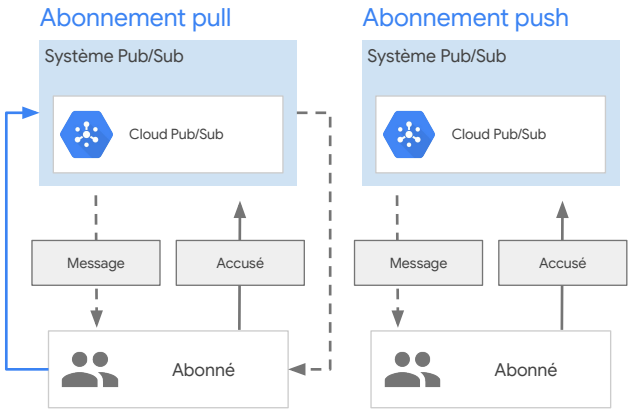
Google Cloud

Le modèle push s'appuie sur un point de terminaison HTTP. Vous enregistrez un webhook correspondant à votre abonnement et l'infrastructure Pub/Sub vous appelle pour obtenir les messages les plus récents. Vous répondez simplement par "status 200 ok" à l'appel HTTP et indiquez ainsi à Pub/Sub que le message a bien été distribué.

Pub/Sub s'autolimite en se basant sur le taux de réponses positives pour ne pas surcharger le nœud de calcul.

Garantie de distribution "au moins une fois"

- Un abonné accuse réception de chaque message pour chaque abonnement
- Un message est renvoyé si l'abonné répond au-delà du `délai` imparti
- Les messages sont conservés pendant 7 jours maximum
- Un abonné peut augmenter le délai de chaque message



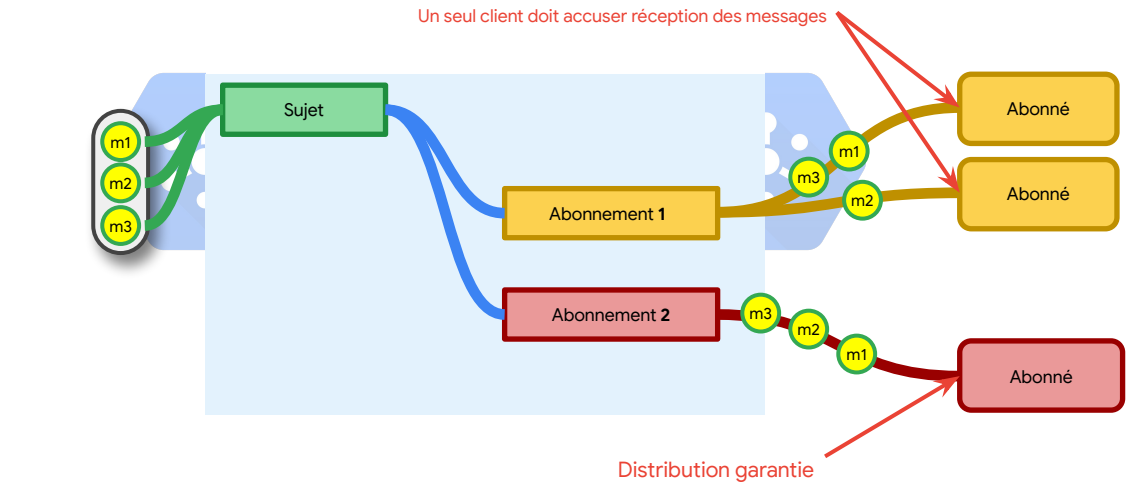
Google Cloud

L'accusé de réception sert à s'assurer que chaque message est distribué au moins une fois. Il est enregistré pour un abonnement précis. Avec deux abonnements, par exemple, l'abonnement pour lequel vous avez enregistré l'accusé de réception continue de recevoir les messages. Pub/Sub tente de transmettre le message pendant sept jours jusqu'à ce qu'il ait été confirmé.

Grâce au mécanisme de relecture, vous pouvez également remonter le temps et relire les messages, et ce jusqu'à sept jours.

Vous pouvez également définir le délai de confirmation pour chaque abonnement. Ainsi, si vous savez que vous avez besoin de 15 secondes en moyenne pour traiter un message dans la file d'attente de tâches, définissez le paramètre "ackDeadline" sur 20 secondes. Pub/Sub n'essaiera pas de retransmettre les messages avant que ce délai expire.

Les abonnés fonctionnent individuellement ou en équipe



Google Cloud

Les abonnés peuvent fonctionner de façon individuelle ou en groupe. Un abonné unique reçoit chaque message transmis via l'abonnement. Il est néanmoins possible de configurer des pools de nœuds de calcul en partageant un même abonnement avec plusieurs abonnés.

Dans ce cas, les messages 1 et 3 peuvent être distribués à l'abonnement 1, et le message 2 à l'abonnement 2. Cette distribution se fait de façon aléatoire, en fonction du moment de la journée où les messages sont extraits.

Le système d'abonnement push repose sur un seul point d'entrée Web et donc sur un seul abonné, en général. Cet abonné peut être une application standard App Engine ou une image de conteneur Cloud Run, qui effectue un autoscaling. Vous avez donc un point de terminaison Web, avec des nœuds de calcul d'autoscaling en arrière-plan. Ce modèle s'avère très efficace.

Publier un message avec Cloud Pub/Sub

```
gcloud pubsub topics create sandiego
```

Créer le sujet

Google Cloud

À présent, intéressons-nous un peu au code. Cet exemple s'appuie sur la bibliothèque client Pub/Sub. Pour publier le message, il faut avant tout créer le sujet.

Publier un message avec Cloud Pub/Sub

```
gcloud pubsub topics create sandiego
```

Créer le sujet

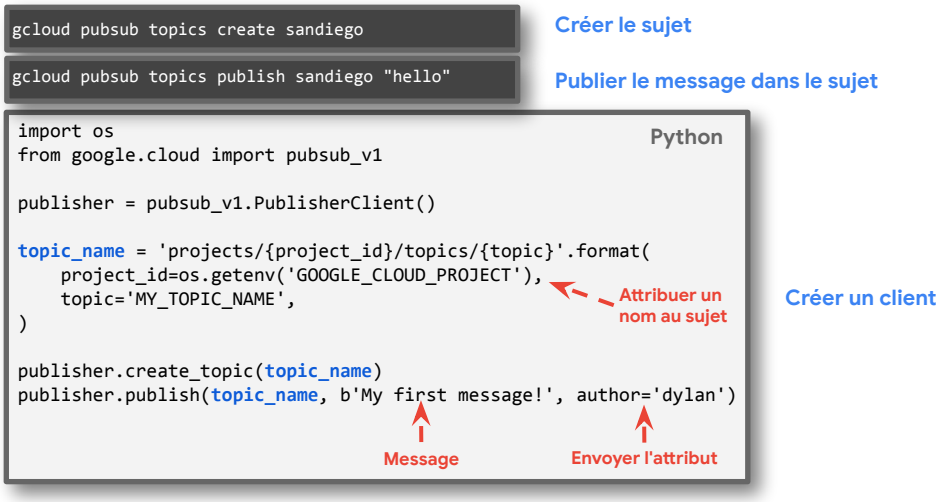
```
gcloud pubsub topics publish sandiego "hello"
```

Publier le message dans le sujet

Google Cloud

L'étape suivante consiste à publier le sujet sur la ligne de commande.

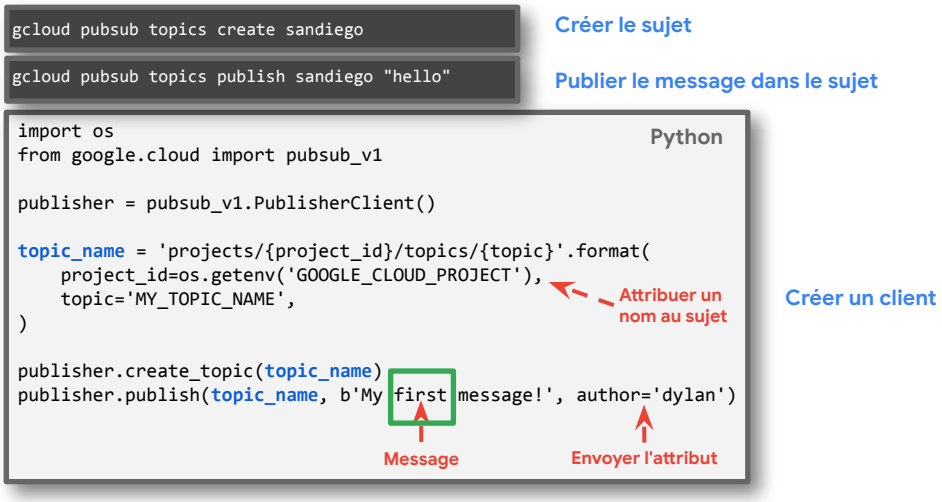
Publier un message avec Cloud Pub/Sub



Google Cloud

La procédure est généralement codée. Dans cet exemple, nous obtenons un PublisherClient, créons le sujet et publions le message.

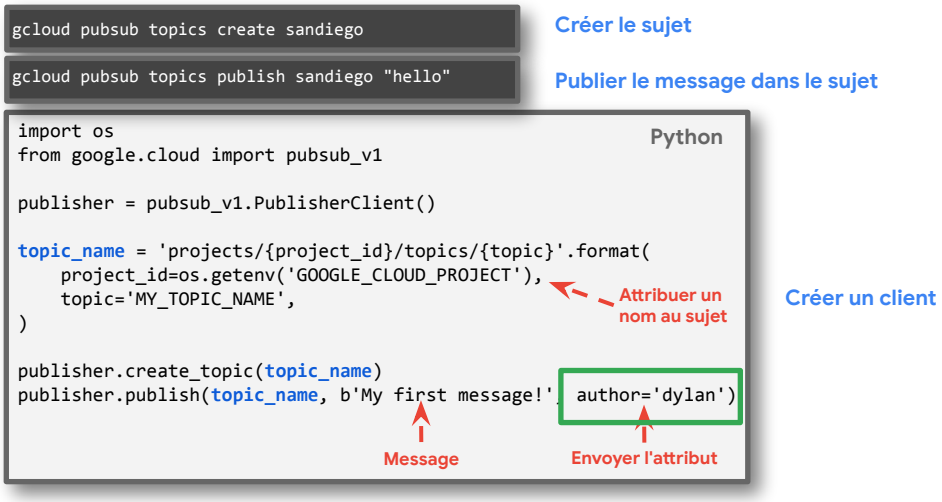
Publier un message avec Cloud Pub/Sub



Google Cloud

Notez la lettre "b" (pour "bytes", ou "octets" en français) juste devant "My first message!". Elle indique que Pub/Sub envoie uniquement des octets bruts. Autrement dit, vous n'êtes pas tenu d'utiliser du texte, vous pouvez aussi envoyer d'autres données, comme des images. La limite est fixée à 10 Mo.

Publier un message avec Cloud Pub/Sub

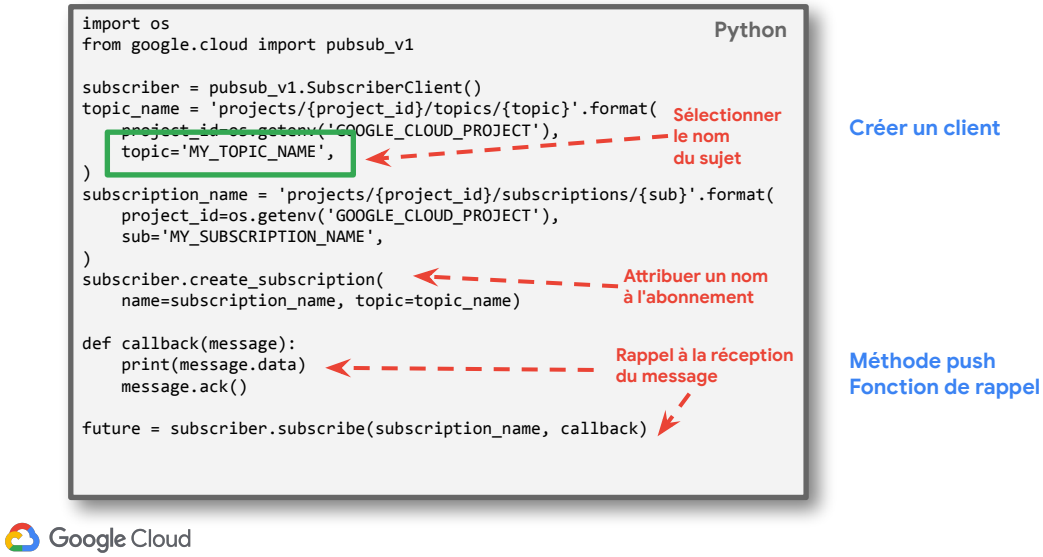


Google Cloud

Il est également possible d'ajouter d'autres attributs dans les messages. Ici, nous avons précisé l'auteur du message, Dylan. Pub/Sub garde une trace de ces attributs, ce qui permet à vos systèmes en aval de récupérer les métadonnées sur vos messages sans que vous ayez à les y insérer et à les extraire. Il garde juste une trace de ces paires clé-valeur, vous évitant ainsi les opérations de sérialisation et désérialisation.

Certains attributs ont une signification particulière. Nous y reviendrons dans quelques instants.

Abonnement pull asynchrone avec Cloud Pub/Sub



Google Cloud

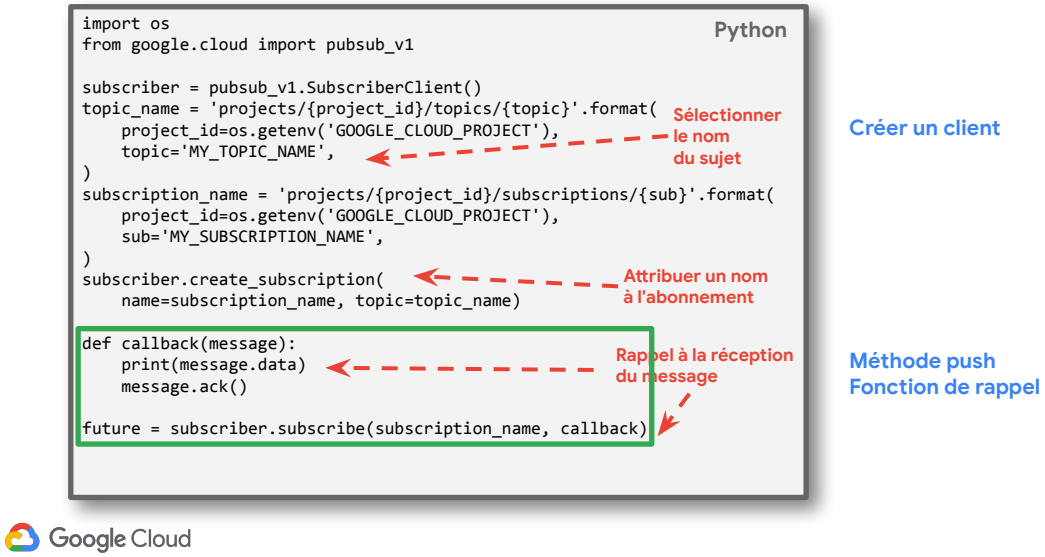
Le code pour s'abonner avec Pub/Sub à l'aide de la méthode push est identique. Vous devez sélectionner le sujet.

Abonnement pull asynchrone avec Cloud Pub/Sub

Google Cloud

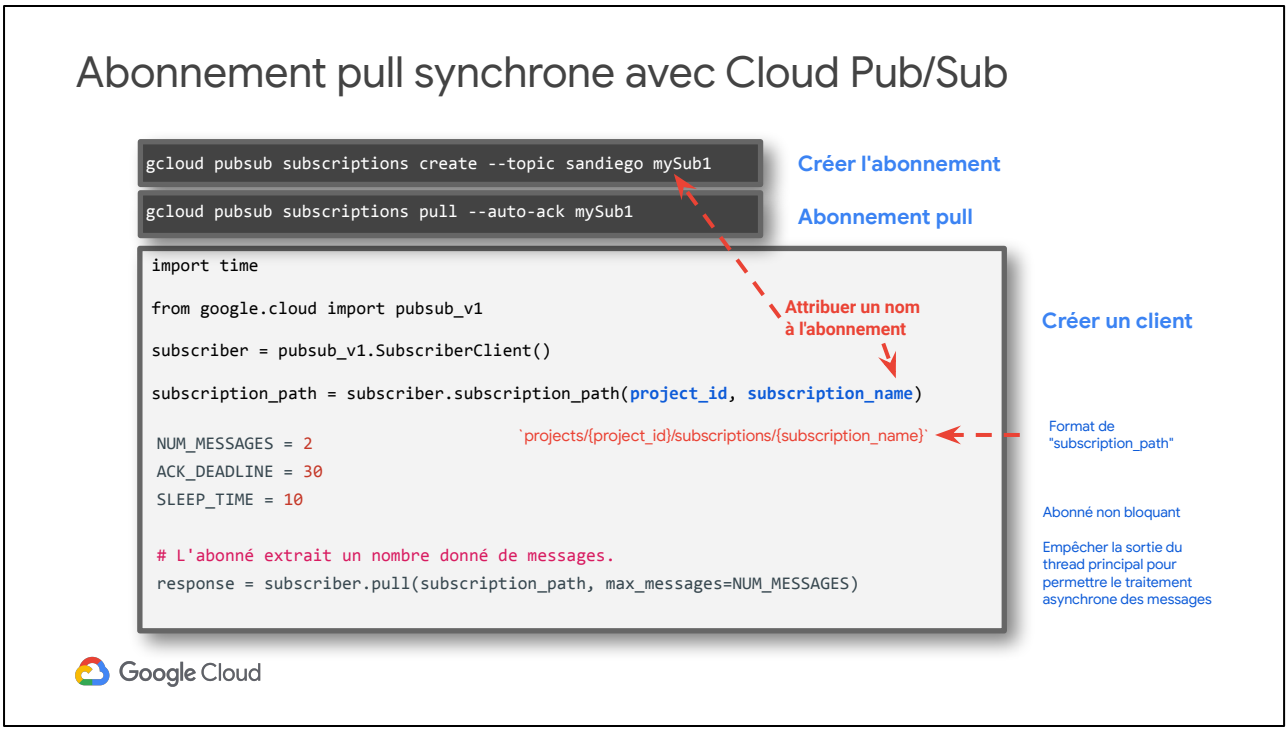
Et attribuer un nom à l'abonnement.

Abonnement pull asynchrone avec Cloud Pub/Sub



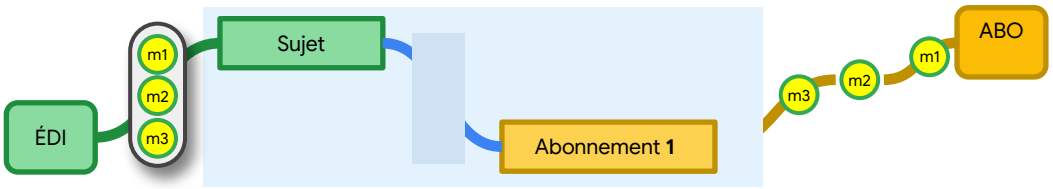
Google Cloud

Comme il s'agit d'un abonnement push, nous définissons un rappel.



Voici un exemple de code pour un abonnement pull. Il est possible d'extraire les messages depuis la ligne de commande. C'est ce que vous allez voir dans l'atelier. Par défaut, seul le message le plus récent est perdu, mais vous pouvez définir une limite maximale pour récupérer 10 messages à la fois, par exemple. Vous pourrez tenter l'expérience lors de l'atelier.

Par défaut, l'éditeur publie les messages par lot. Désactivez ce paramètre si vous souhaitez réduire la latence.



Publier des messages par lot : débit ou latence

Google Cloud

Les messages peuvent aussi être publiés par lot. Cette méthode évite d'appeler chaque message côté éditeur. Après une période d'attente, l'éditeur peut ainsi envoyer simultanément 10, voire 50 messages. Bien que cette stratégie soit efficace, le fait d'attendre 50 messages génère de la latence pour le premier. L'idée est donc de trouver un compromis dans votre système. Que souhaitez-vous optimiser ? Quoi qu'il en soit, même dans le cadre de la publication par lot, les messages sont toujours distribués un par un aux abonnés.

Nous nous entraînerons à cette technique dans l'atelier.

Modifier le paramètre de publication par lot dans Cloud Pub/Sub

Python

```
from google.cloud import pubsub
from google.cloud.pubsub import types

client = pubsub.PublisherClient(
    batch_settings=BatchSettings(max_messages=500),
)
```

Modifier le paramètre de publication par lot

Google Cloud

Voici comment définir la publication par lot dans Python.

Pub/Sub : latence, distribution dans le désordre et duplication

- Latence : pas de garantie
- Distribution des messages dans n'importe quel ordre, surtout avec un long temps d'attente
- Duplication éventuelle des messages

Google Cloud

Il est certes possible de distribuer les données dans le désordre et dans plusieurs instances, mais vous pouvez tout à fait procéder autrement.

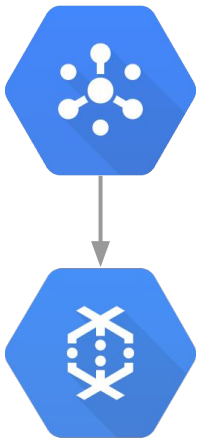
Vous pouvez, par exemple, écrire une application qui gère les messages répliqués qui sont distribués sans respecter l'ordre chronologique. Cette application est différente d'un véritable système de mise en file d'attente, où les messages sont souvent distribués dans l'ordre chronologique. Avec Pub/Sub, c'est impossible à cause du compromis qu'il a fallu faire pour une meilleure évolutivité, compte tenu de la dimension mondiale de ce service. Le réseau est maillé, ce qui explique qu'un message prenne parfois une autre route, plus lente, et arrive après d'autres envoyés a posteriori.

Cette configuration ne conviendrait pas pour une application de chat, car il serait gênant que les messages arrivent dans le désordre. Nous utiliserons donc d'autres techniques pour gérer l'ordre des messages.

Dernier point : nous devons nous préparer au fait que les messages puissent être dupliqués.

Cloud Pub/Sub avec Dataflow : traitement unique et ordonné

- Cloud Pub/Sub effectue au moins une distribution
- Cloud Dataflow : déduplication, ordre et fenêtrage
- Séparation des problèmes → évolutivité



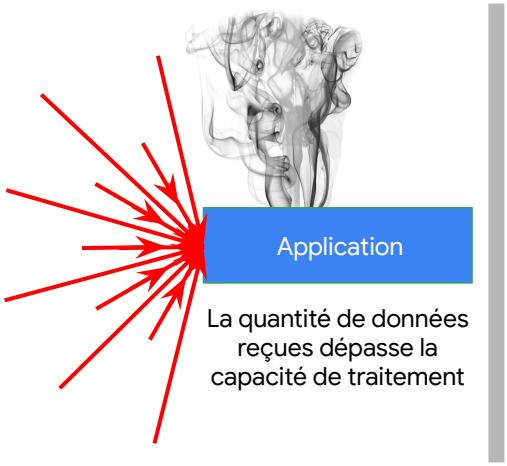
Google Cloud

Pour résoudre ces problèmes, nous utiliserons Dataflow en plus de Pub/Sub. Dans Pub/Sub, un message distribué deux fois conserve le même ID. Dataflow va donc dédupliquer les messages d'après leur identifiant. BigQuery offre également des fonctionnalités de déduplication, qui sont toutefois limitées.

Dataflow ne peut pas distribuer les messages selon l'ordre exact dans lequel ils ont été publiés, mais il facilite la gestion des données tardives.

La combinaison Pub/Sub et Dataflow apporte un niveau d'évolutivité inégalé.

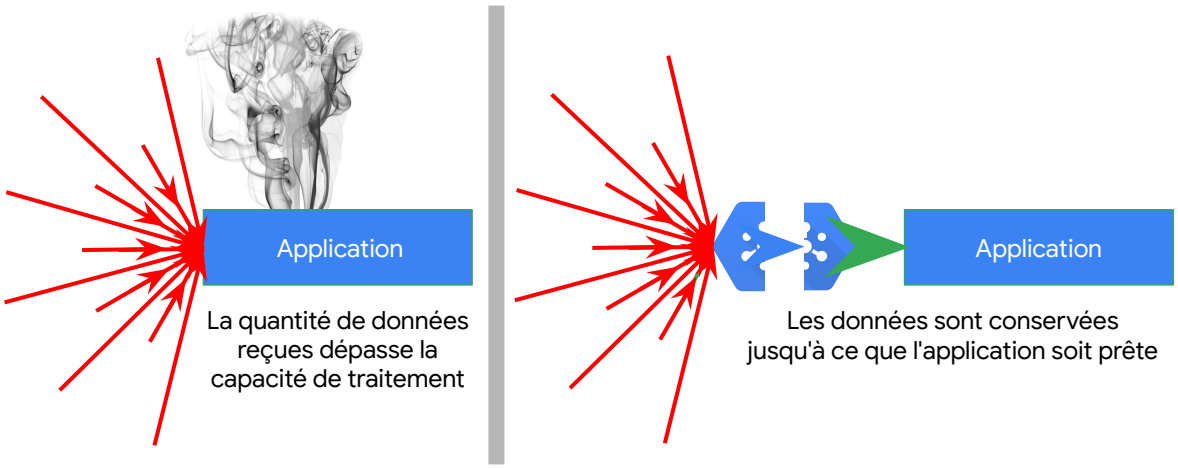
Résilience en flux continu avec Cloud Pub/Sub



Google Cloud

Pub/Sub est également un allié sur le plan de la résilience en flux continu, ou mise en mémoire tampon. Que se passe-t-il lorsque vos systèmes sont surchargés par des volumes de transactions élevés, comme à l'occasion du Black Friday ? Ce qu'il vous faut, c'est un mécanisme de tampon ou de mise en attente pour distribuer les messages au fur et à mesure que vos systèmes sont en mesure de les traiter. Pub/Sub intègre justement cette fonctionnalité.

Résilience en flux continu avec Cloud Pub/Sub



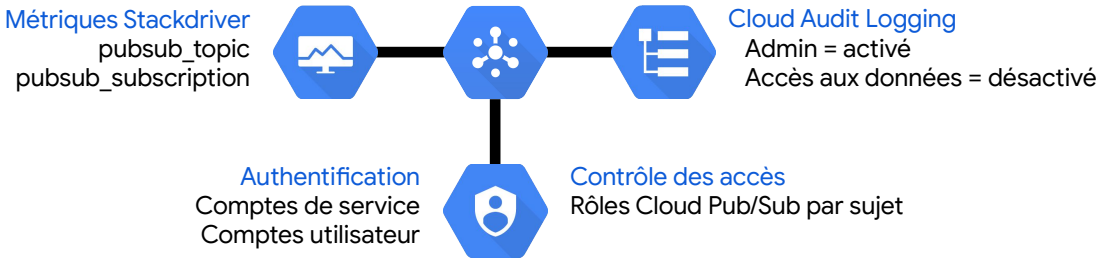
Google Cloud

Récapitulons les choses. Dans l'exemple de gauche sur cette diapositive, la surcharge de données entrantes génère un pic de trafic. La fumée sert à illustrer la surmultiplication des ressources de l'application. Une solution consiste à dimensionner l'application de sorte qu'elle puisse gérer le pic de trafic maximum et offrir dans le même temps de la capacité supplémentaire, un filet de sécurité en quelque sorte. Cela représente autant d'espace perdu pour des ressources qui doivent rester au maximum de leur capacité même lorsqu'elles sont inutilisées, et offre une parade contre les attaques par déni de service distribué en fixant une limite supérieure au-delà de laquelle l'application cessera de fonctionner normalement et se comportera de manière non déterministe.

Dans la solution de droite, Cloud Pub/Sub joue le rôle d'un intermédiaire qui reçoit et conserve les données jusqu'à ce que l'application dispose des ressources nécessaires pour les gérer, que ce soit en traitant les tâches en attente ou en effectuant un autoscaling pour satisfaire la demande.

https://pixabay.com/fr/photos/smoke-background-artwork-swirl-69124/

Sécurité, surveillance et journalisation pour Cloud Pub/Sub

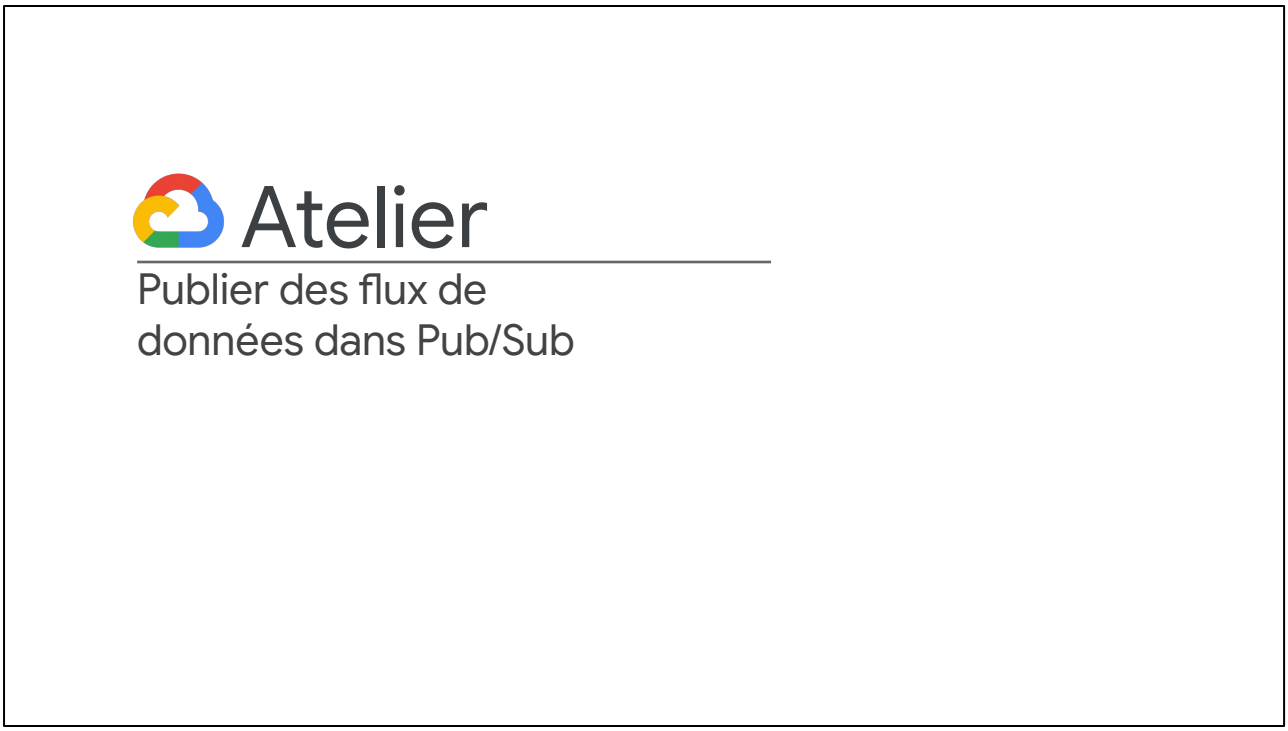


Google Cloud

Cloud Audit Logging propose trois journaux d'audit pour chaque projet, dossier et organisation Google Cloud Platform : **Activités d'administration**, **Accès aux données** et **Événements système**.
Les entrées des journaux d'audit des activités d'administration concernent des appels d'API ou toute autre opération d'administration modifiant la configuration ou les métadonnées des ressources.
Les journaux d'audit relatifs à l'accès aux données contiennent des appels d'API qui lisent la configuration ou les métadonnées des ressources, ainsi que des appels d'API pilotés par l'utilisateur qui créent, modifient ou lisent des données de ressources fournies par l'utilisateur.
Les journaux d'audit des activités d'administration sont toujours renseignés. Vous ne pouvez pas les configurer ni les désactiver.Les journaux d'audit relatifs à l'accès aux données sont désactivés par défaut, car ils peuvent être volumineux. Ils doivent être activés explicitement pour être renseignés.

Cloud Pub/Sub fournit des métriques à Stackdriver, dont "pubsub_topic" et "pubsub_subscription", qui vous permettent de surveiller l'utilisation du quota de services depuis un tableau de bord ou de définir des notifications et des alertes.

Les comptes de service servent de mécanisme d'authentification. Vous pouvez également authentifier directement les utilisateurs grâce à leur compte utilisateur (leur identité apparaît dans les journaux d'audit). Cette configuration est toutefois déconseillée. Cloud IAM assure le contrôle des accès.



À présent, entraînons-nous à publier des flux de données dans Pub/Sub.

Objectifs de l'atelier

Créer un sujet et un abonnement Pub/Sub

Simuler les données de capteurs de trafic dans Pub/Sub

Google Cloud

Dans cet atelier, nous allons créer un sujet et un abonnement Pub/Sub, et simuler les données de trafic de San Francisco dans Pub/Sub. Voilà pour la première partie de cet atelier, qui en compte quatre au total.