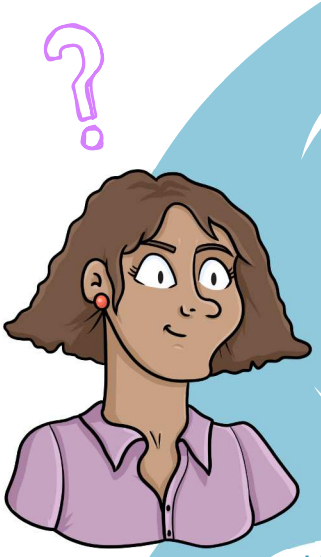
La Minute CLOUD de JULES & LÉA

# Azure Cosmos DB #1

Aujourd'hui, nous allons évoquer le service **Azure Cosmos DB**.



Vu le nom, je suppose que c'est un service de base de données (DB) ?

Effectivement.

Pour être plus précis, Cosmos DB est un service de DB de type **NoSQL**, c'est à dire de DB non relationnelles.

A l'inverse des DB SQL ?

Tout à fait.

Une DB de type NoSQL complètement managée par Microsoft qui propose des temps de réponse très faible, ainsi qu'une scalabilité automatique et instantanée.

Ce qui fait de lui une solution parfaite, lorsqu'on a des applications à grande échelle !

Oui, mais ce n'est pas tout !

L'un des gros avantages de Cosmos DB, est qu'il offre plusieurs API de DB open source.

L'**API Core** (**SQL**) stocke les données au format document en prenant en charge l'interrogation des données à l'aide de la syntaxe SQL.

L'**API Mongo** DB stocke les données dans une structure de document via le format BSON.

L'**API Cassandra** stocke les données dans un schéma orienté colonne et permet d'interagir avec les données à l'aide du langage CQL.

L'**API Gremlin** stocke les données sous forme de graph.

Et enfin, l'**API Azure Table**, stocke les données au format clé/valeur.

suivante

La Minute CLOUD de JULES & LÉA

Azure
Cosmos DB #2

Donc si je comprends bien, il prend en charge différents formats de DB ?!

Tout à fait.

Cosmos DB offre une **mise à l'échelle automatique** du stockage, mais aussi du débit, et propose différentes options de **persistence des données**.

Mais quelle API faut il privilégier ?

Il n'y a pas d'API idéale, il faut choisir celle qui convient le mieux au cas d'usage de l'application.

Effectivement cela parait logique.

Lorsque tu crées une nouvelle DB, c'est l'**API Core** qui est définie **par défaut**.

Il est intéressant de choisir une autre API, lorsque tu possèdes déjà une des DB qu'on à vu précédemment.

Ou lorsque tu ne souhaites pas réécrire la couche d'accès aux données pour une nouvelle application.

J'ai cru voir aussi qu'il existait **différents mode de déploiement** ?

Oui, effectivement, tu as 2 choix de déploiements.

Le **mode Débit approvisionné** qui est recommandé pour les charges de travail avec un trafic soutenu nécessitant des performances prévisibles.

Tu vas alors provisionner de l'**unité de requête** (**RU**) par seconde pour traiter les requêtes.

Et le **mode Serveless**, recommandé pour les charges de travail avec un trafic intermittent ou imprévisible.

suivante

La Minute CLOUD de JULES & LÉA

Azure
Cosmos DB #3

Et je suppose qu'en fonction du mode de déploiement, les **fonctionnalités** associées sont **différentes** ?

C'est vrai.

Ton mode de déploiement influencera par exemple **les performances** de ta DB, **l'espace de stockage**, ou bien la possibilité ou non, d'activer la **géo-réplication** qui n'est pas disponible dans le mode Serveless.

Et en terme de sauvegarde des données ?

Cosmos DB propose 2 modèles de sauvegardes.

Le 1er, la **sauvegarde périodique**, effectuée à intervalles réguliers, ou la restauration s'effectue en créant un ticket auprès du support Microsoft.

C'est un peu contraignant car nous ne sommes pas autonome en terme de restauration.

Oui, c'est pour cela que Azure propose un autre modèle.

C'est la **sauvegarde continue** ou **Point In Time Restore** (PITR) qui permet de restaurer quand tu le souhaites en fonction de la rétention configurée.

J'ai vu qu'il était possible, de **migrer** du mode de **sauvegarde périodique** au mode de **sauvegarde continue**.

Exactement, mais avec certaines limitations.

**Toutes les APIs ne sont pas encore supportées** dans le mode de sauvegarde continue, il faut donc vérifier la documentation avant de se lancer.

Merci pour le conseil.

suivante

La Minute CLOUD de JULES & LÉA

# Azure Cosmos DB #4

Et en terme de sécurité, qu'avons nous au menu ?

Pas mal de choses comme souvent :

Il est possible de **filtrer le trafic** pour limiter l'accès à Cosmos DB. Soit aux subnets existants, soit à certaines adresses IP publiques.

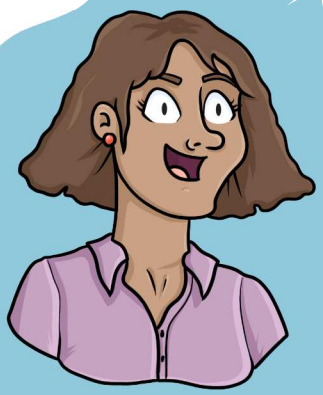
On peut aussi activer l'option **Private Link** pour consommer Cosmos DB via une adresse privée d'un VNET.

**L'authentification** à Cosmos DB, s'effectue au travers d'une **clé** ou d'une **chaine de connexion**.

Ce qui est sympa, c'est que l'équipe a pensé à une **clé pour les accès en écriture**, et une autre **clé pour les accès en lecture**.

Et en plus si besoin on peur **renouveler les clés** en cas de **compromission**.

Très ingénieux comme mécanisme.



Sans oublier la possibilité de **chiffrer le contenu** de la DB **au repos**, soit avec une clé de chiffrement fournie par Microsoft, ou avec une clé de chiffrement fournie par l'utilisateur.

Et évidemment, on peut la stocker dans **Azure Key Vault**.

Autre chose sympa, c'est que Cosmos DB est nativement intégré à **Microsoft Defender for Cloud** pour apporter une couche de sécurité supplémentaire.

Excellent.

Azure Cosmos DB est donc un service qui peut être utilisé dans plusieurs cas d'usage, et les différentes APIs supportées le rendent super cool.

La Minute CLOUD de JULES & LÉA

Merci à vous !

Si vous souhaitez continuer à **apprendre**, de façon ludique, sur **l'écosystème Azure**, et ne rater aucune de nos illustrations ...

... N'hésitez pas à vous abonner sur :

**https://aka.ms/grow-una**

**https://tinyurl.com/youtube-growuna**

Et si le contenu vous plaît, partagez-le ;o)

A très vite !

GROW UNA