

Today we are going to talk about the **Azure Cosmos DB** service.

Given the name, I guess that it's a database (DB) service?

Indeed.

To be more precise, Cosmos DB is a **NoSQL** type DB service, ie non-relational DB.

Unlike SQL DBs?

Absolutely.

Is a NoSQL-type DB completely managed by Microsoft which offers very low response times, as well as automatic and instantaneous scalability.



Which makes it a perfect solution when you have large-scale applications!

Yes, but that's not all!

One of the big advantages of Cosmos DB is that it offers several open source DB APIs.

The Core (SQL) API stores data in document format with support for querying data using SQL syntax.

The Mongo DB API stores data in a document structure through BSON format.

The Cassandra API stores data in a column-oriented schema and allows interacting with the data using CQL.

The Gremlin API stores data as a graph.

And finally, the **Azure table API**, stores data in key/value format.



So as I understood it supports different DB formats?!

Absolutely.

Cosmos DB offers **automatic scaling of storage**, but also of throughput, and offers different **data persistence options**.

But which API should be preferred?

There is no ideal API, you have to choose the one that best suits the use case of the application.

Indeed, that seems logical.

When you create a new DB, the **Core API** is defined **by default**.

It is interesting to choose another API, when you already have one of the DBs that we have seen previously.

Or when you don't want to rewrite the data access layer for a new application.

I thought I also saw that there were **different deployment modes**?

Yes, indeed, you have 2 choices of deployments.

Provisioned Throughput mode which is recommended for workloads with sustained traffic requiring predictable performance.

You will then provision **request units (RU)** per second to process requests.

And **Serverless mode**, recommended for workloads with intermittent or unpredictable traffic.

next

$$P=m \cdot V \quad T=2\pi\sqrt{\frac{l}{g}}$$





And I suppose that depending on the mode of deployment, the associated features are **different**?

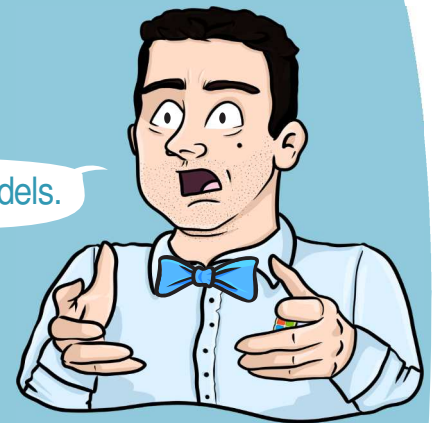
Yep!

Your deployment mode will influence, for example, the **performance** of your DB, the **storage space**, or the possibility or not of activating **geo-replication** which is not available in Serveless mode.

What about data backup?

Cosmos DB offers 2 backup models.

The 1st, the **periodic backup**, carried out at regular intervals, or the restoration is carried out by creating a ticket with Microsoft support.



It's a bit restrictive because we are not independent in terms of catering.

Yes, that's why Azure offers another model.

It is the **continuous backup** or **Point In Time Restore (PITR)** which allows you to restore whenever you want according to the configured retention.

I saw that it was possible, if desired, to **migrate** from **periodic backup** mode to **continuous backup** mode.



Exactly, but with some limitations.

Not all APIs are supporting continuous backup mode yet, so check the documentation before jumping in.

Thanks for the advice.

next





And in terms of security, what do we have on the menu?

A lot of things as often:

It is possible to **filter traffic** to limit access to Cosmos DB. Either to existing subnets or to certain public IP addresses.

You can also activate the **Private Link** option to consume Cosmos DB via a private address of a VNET.

Authentication to Cosmos DB is done through a **key** or a **connection string**.

What's nice is that the product team provides one **key for write access**, and another **key for read access**.

And in addition, if necessary, we can **renew the keys** in the event of **compromise**.



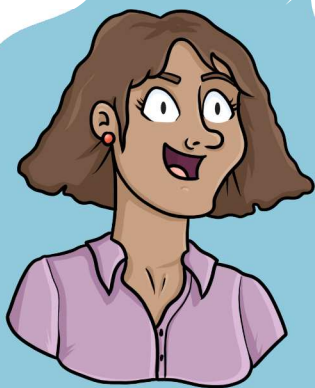
Very ingenious mechanism.



We also have the possibility of **encrypting the contents** of the DB **at rest**, either with an encryption key provided by Microsoft, or with an encryption key provided by the user.

And of course, we can store it in **Azure Key Vault**.

Another cool thing is that Cosmos DB is natively integrated with **Microsoft Defender for Cloud** to provide an additional layer of security.



Excellent.

Azure Cosmos DB is therefore a service that can be used for several use cases which makes it super cool.



Thank you!



If you want to continue **learning** in a fun way about the **Azure ecosystem**, and not miss any of our illustrations ...

... Feel free to subscribe at:



<https://aka.ms/grow-una>



<https://tinyurl.com/youtube-growuna>

If you like our work, please share it ;o)

See you soon!

