Azure Cosmos DB, a globally distributed database (microsoft.com)

336 points by andysinclair 15 days ago | hide | past | web | 113 comments | favorite

ahelwer 15 days ago [-]

Designed with TLA+! :D Small interview with Leslie Lamport:

https://techcrunch.com/2017/05/10/with-cosmos-db-microsoft-w...

Hope Cosmos team releases a whitepaper on their experiences with the language. I'd heard snatches of gossip here and there that TLA+ was used inside Cosmos, but no concrete details.

edit: apparently there's also a video of Lamport talking about this https://www.youtube.com/watch?v=L_PPKyAsR3w

> dmix 15 days ago [-]
>
> The effort to formalize the database was apparently led by:
>
> > Leslie Lamport the Turing Award winner whose work underpins many of these concepts (and who also wrote the LaTeX document preparation system) and who joined Microsoft Research in 2001
>
> Quote about TLA+:
>
> > As Shukla noted, though, his idea here was to build a database system that could last decades. To do this, he also brought in Lamport to teach the team TLA+. Lamport has long had a special interest in how developers spec out their applications. TLA+ is essentially a formal language for doing just that. "When we started out in 2010, we wanted to build a system — a lasting system. This was the database of the future for Microsoft," Shukla told me. "We try to apply as much rigor to our engineering as we possibly can. […] TLA+ has been wonderful in getting that level of rigor in a team of engineers to set the bar high for quality." TLA+, Lamport noted, allows you to do the high-level design of a system in a completely formal way — and because it's done formally, it can be checked for correctness, too (and to be fair, AWS and others also use TLA+ to spec out their distributed systems). "I don't want to give the impression that TLA+ is great and I'm brilliant. It's great because it's almost entirely based on mathematics," Lamport added.

> dharmashuklaMS 13 days ago [-]
>
> Cosmos DB codebase consists of multi-million lines of C++ code. It is fully asynchronous and like all large scale (stateful) distributed systems, it is also extremely complex. The database engine of Cosmos DB (including the b-tree and the log structured storage engine) is fully latch free; the engine, resource governance subsystem, global distribution infrastructure, partition management etc -- are all deeply integrated. Each of these subsystems have extremely complex state machines which are hard to describe with the required degree of *precision* using the English language. Any correctness bug that gets introduced because of lack of precision, can potentially lead to data loss, corruption, partial or complete failures of the entire service. This is where TLA+ comes in.
>
> Background: Dr. Leslie Lamport's work has been a constant source of inspiration for the Cosmos DB team. A few of the engineers on the team had learnt TLA+ initially on their own and started seeing its benefits. Subsequently, other members of the team started applying it as well. Leslie had also taught a fantastic class on TLA+ (across Microsoft), which engineers on the Cosmos DB team attended. It was a wonderful, once in a lifetime opportunity for the team to learn TLA+ from Leslie.
>
> To be clear, Leslie personally didn't write any of the TLA+ specs for Cosmos DB. It was Cosmos DB engineers who wrote the TLA+ specs to specify & verify the design (incl. consistency models). The net result is that TLA+ made Cosmos DB a more robust system, which is crucial to offer strict and comprehensive SLAs encompassing availability, consistency, throughput and latency at the 99th percentile. Further, writing TLA+ specs for the five consistency models which we have exposed (as well as many that we have experimented, internally) enabled us to precisely define the semantics for each of the consistency models. This in-turn enables developers building apps on top of Cosmos DB, to rely on the well-defined semantics.

Hope this is helpful.

reply

ahelwer 12 days ago [-]

Thanks, Dharma! I probably actually TA'd the TLA+ class which your engineers attended ;)

Still hope you write a whitepaper. The AWS paper is super valuable when making the case for TLA+ use in industry.

reply

Quiark 15 days ago [-]

Wow, that's cool. TLA+ is the same formalism that Amazon uses to verify (some of) their cloud services.

OriginalJGC 15 days ago [-]

Here is a longer version with Dr. Lamport with more details
https://www.youtube.com/watch?v=L_PPKyAsR3w&feature=youtu.be

dharmashuklaMS 15 days ago [-]

Hi, This is Dharma from Azure Cosmos DB team. We are super excited to make the service available today.We published the first of the series of technical blog posts here -> https://azure.microsoft.com/en-us/blog/a-technical-overview-... . Would love to answer any Cosmos DB questions.

daxfohl 15 days ago [-]

Are graph ops/queries atomic? i.e. if you run a tree query on a tree-graph, at the same time you're re-parenting a treenode, is there a chance that the node could end up in the result tree twice or zero times?

Also, if they're atomic, are they optimistic or pessimistic transactions? Also if they're atomic, does that mean queries done on the write server? (My understanding is that readonly transactions on read-replicas are not supported, or at least they weren't under DocDB).

Any lay-programmer insight into what's going on under the hood, or at least performance/atomicity implications, without giving up too much proprietary info, would be appreciated.

dharmashuklaMS 13 days ago [-]

Read only transactions existed in DocumentDB too. DocumentDB was a strict subset of the capabilities that Cosmos DB provides/existed underneath. Hence, read only transactions exist in Cosmos DB too. The first technical overview blog post attempted to provide you a high level overview. We are hoping to cover specific areas in either future blog posts, conference publications.

reply

tosh 15 days ago [-]

Can you give an overview on how Cosmos differs from Spanner and CockroachDB?

dharmashuklaMS 15 days ago [-]

There are many significant differences in capabilities, and design approaches between other systems and Cosmos DB. At a very high level. differences are at two levels - the design of the database engine and the larger distributed system. The database engine design is inspired on LLAMA http://db.disi.unitn.eu/pages/VLDBProgram/pdf/research/p853-..., Bwtree - > https://pdfs.semanticscholar.org/7655/9c6cc259c6ab5baf7bd19d... and schema-agnostic indexing techniques -> http://www.vldb.org/pvldb/vol8/p1668-shukla.pdf. Please note that these papers are significantly behind the current state of the implementation. The most crucial aspect that these papers dont cover is the integration of the database engine with the larger distributed system components of Cosmos DB including the resource governance, partition management, and the implementation of replication protocol /consistency models etc. Our goal is to publish all of the design specifications including TLA+ specs over time.

jasondc 15 days ago [-]

> Latency: 99.99% of <10 ms latencies at the 99th percentile

Impressive SLA to guarantee, I'm curious if this will hold up in all random customer workloads that are coming, e.g. updating a lot of fields in a large document (or just a very large insert).

yazaddaruvala 15 days ago [-]

"For a typical 1-KB item, Cosmos DB guarantees end-to-end latency of reads under 10 ms and indexed writes under 15 ms at the 99th percentile, within the same Azure region."

i.e. 1 KB item; Same Azure region;

This now seems more plausible.

One thing I'm curious about is if they tested load on a single partition, or if they only tested latencies for random access.

ChuckMcM 15 days ago [-]

It is interesting that given their scale it essentially says that you can implement twitter on one of these and be done with a chunk of the infrastructure.

aravindkr1 15 days ago [-]

Cosmos DB guarantees both low latency and that you can achieve your provisioned throughput with SLAs. Latency is guaranteed at p99 regardless of storage size or number of partitions.

yazaddaruvala 15 days ago [-]

Sorry, I'm not talking about "number of partitions" but if I "forcibly"[0] hit the same partition, will the SLA hold?

[0] i.e. If I somehow pick keys which are on the same partition.

bigtones 15 days ago [-]

Agreed - seems like a rebranding in response to Google Cloud Spanner DB announcement a couple of months back. Microsoft's marketing seems to be pushing some of the same outcomes in the ad copy.

aliostad 15 days ago [-]

We just did a benchmarking for a PoC on DocumentDB side-by-side Cassandra. It does the job, I have not yet seen anything revolutionary. Cassandra benchmarks seemed better.

aravindkr1 15 days ago [-]

One key difference is the cost difference between running Cosmos DB and Cassandra. We have a TCO paper (https://aka.ms/documentdb-tco-paper) that shows that for a 1M operations/second workload, Cosmos DB is significantly 3x-10x cheaper than other systems.

brianwawok 14 days ago [-]

And white papers are full of it. Has random things like s Cassandra cluster needs 1 full time engineer per 100 nodes. Where do you come up with this stuff?

throwanem 14 days ago [-]

In my experience of Cassandra (working on an application client, not managing it), one FTE per hundred nodes is *extremely* generous.

bshanks 14 days ago [-]

Do you mean that you typically need more engineers, or less?

throwanem 14 days ago [-]

The former. One FTE per dozen nodes is more in line with what I've seen in practice.

rattray 15 days ago [-]

Given that this makes bolder claims that Cassandra, near-parity is pretty impressive, no?

(I don't know a ton about either technology, please correct me if I'm wrong)

Dimi9909 15 days ago [-]

this is a little bit hard to believe. Inter region Ping can even take longer than 15ms... I guess this SLA is for eventual consistency model not strong consistency model

aravindkr1 15 days ago [-]

The latency SLAs are within the same Azure region. You can distribute your data across any of the 30+ regions that Azure is available in. Your apps always read from the local/closest region with the homing APIs.

xapata 14 days ago [-]

That didn't answer the question: will all regions be always consistent?

GovindMS 14 days ago [-]

We would invite you to explore the consistency levels available in Cosmos DB. As a developer you can choose what makes most sense in distributed scenario. https://docs.microsoft.com/en-us/azure/documentdb/documentdb... Eventual is one end of the consistency. But then consistent prefix, session (if you can control) and bounded staleness all can play a role.

xapata 14 days ago [-]

Tunable consistency. Very cool.

judah 15 days ago [-]

This isn't a new database. This is a rebranding of the generically-named Azure DocumentDB, plus some new features.

sealord 15 days ago [-]

Not exactly. DocumentDB was primarily a document store. Cosmos allows you to store graphs and KV pairs as well.

jchrisa 15 days ago [-]

I was unable to glean from a quick read of the consistency documentation, does CosmosDB support uniqueness and foreign key constraints?

sealord 14 days ago [-]

Afraid not - there's no real concept of constraints with CosmosDB.

aliostad 15 days ago [-]

DocumentDB is KV.

rattray 15 days ago [-]

DocumentDB looks like a JSON document store?

hoschicz 15 days ago [-]

It is.

henriksen 15 days ago [-]

Not really. It's a new database and is a superset of DocumentDB.

dharmashuklaMS 15 days ago [-]

Azure Cosmos DB has been many years in the making. Azure Cosmos DB started as "Project Florence" in late 2010 to address developer the pain-points faced by large scale applications inside Microsoft. Observing that the challenges of building globally distributed apps are not a problem unique to Microsoft, in 2015 we made the first generation of this technology available to Azure developers in the form of DocumentDB. Since that time, we've been steadily adding new capabilities both in the database engine as well as, larger distributed system components. Azure Cosmos DB is the result. It is the next big leap in globally distributed, at scale, cloud databases. As a part of this release of Azure Cosmos DB, DocumentDB customers, with their data, are automatically Azure Cosmos DB customers. They now have access to the new system and capabilities offered by Azure Cosmos DB today as well as, as we keep evolving the service.

curiousDog 15 days ago [-]

Does Cosmos DB support distributed transactions like Spanner? Would be nice to see a deeper dive into Transactions like how isolation works etc.

brandur 15 days ago [-]

I'd also be interested in an answer. I'm guessing the answer is "no" based on my reading of the docs, but it'd be nice to confirm that.

vyrotek 15 days ago [-]

Is GROUP BY support a possibility in the future? I was a little disappointed to see aggregates implemented without them after waiting for so long...

Here's a link for anyone who would like to vote for this feature. https://feedback.azure.com/forums/263030-documentdb/suggesti...

GovindMS 14 days ago [-]

Thanks - this and other features are planned. Please stay tuned.

joshuatalb 15 days ago [-]

This feels like MS' version of Google's Cloud Spanner that's GA in a few days. Same kind of marketing too.

strmpnk 15 days ago [-]

Having evaluated both, I'd say Cloud Spanner is quite a bit behind in some regards. It's not that Spanner itself can't do something but that Cloud Spanner hasn't productized some important features like multi-datacenter failovers. It's certainly coming but CosmosDB (aka. DocumentDB) does a lot of this today (and has been for awhile as this is not entirely new).

azurezyq 15 days ago [-]

But from Cosmos DB's doc, cross-dc strong consistency seems not even supported.

https://docs.microsoft.com/en-us/azure/documentdb/documentdb...

"Azure Cosmos DB accounts that are configured to use strong consistency cannot associate more than one Azure region with their Azure Cosmos DB account."

strmpnk 15 days ago [-]

I didn't claim cross-dc consistency. I said failover, which is shockingly hard to make many competitors do. The key here is that only one DC can take writes but the failover works transparently with your client (also with clear SLAs).

lern_too_spel 15 days ago [-]

Why do you need failover if you have global multimaster like Spanner?

aliuy 15 days ago [-]

An interesting thing to point out is the current beta for Cloud Spanner does not have multi-region deployments... and instead, allows you to do single-region deployment in your choice of 3 (not >30) regions:

https://cloud.google.com/spanner/docs/instance-configuration

voellm 15 days ago [-]

One of the best parts of the perf SLA is we did it with all Data Encrypted at Rest. I'm biased. I lead security for CosmosDB.

sargun 14 days ago [-]

Why is encryption at rest difficult? I presume it's all AES (hardware accelerated), with some key derived at system boot time?

Are y'all doing encryption in the D/C, or just on the WAN? If you're doing it on the WAN, any luck using MACSEC?

henriksen 15 days ago [-]

Talk about the foundations of Cosmos DB: https://www.youtube.com/watch?v=Yfmw7swCtZs

vikestep 15 days ago [-]

I can't seem to find the old DocumentDB prices any more, but it seems like it's a lot cheaper now? Also, is User-Defined Performance something new? since last time I looked into DocumentDB you had to pay the monthly RU fee per 10GB disk.

One more thing, as someone who went from DocumentDb to Azure Storage (Tables) back in April 2016 because of the higher price, slower queries, and scalability problems, is there anything that may make Cosmos DB a better option?

yunong 15 days ago [-]

What's the CAP tradeoffs of Cosmos? It's not clear to me looking at the SLA docs.

aliuy 15 days ago [-]

CAP only talks about the uncommon unhappy path (given a network partition, what is tradeoff between availability and consistency). PACELC theorem builds on CAP to describe that even in the absence of a network partition, there is a trade-off between latency and consistency (in other words, describing the tradeoffs associated with BOTH the common happy path and uncommon unhappy path).

Azure Cosmos DB offers 5 well-defined consistency models for you to choose from, so that you can choose the right tradeoffs for a given application or scenario. This way, you aren't stuck choosing between the hard extremes of Strong and Eventual consistency.

See: https://docs.microsoft.com/en-us/azure/documentdb/documentdb...

dharmashuklaMS 15 days ago [-]

As the Cosmos DB SLA doc describes, for any of the 5 consistency models the service supports, the service guarantees all the other three guarantees (latency at the 99th percentile, availability and throughput) at 99.99%. One of the benefits of well-defined consistency models is that for a given model, developers can clearly make the tradeoffs between (1) latency vs. consistency, (2) availability vs. consistency and (3) throughput vs. consistency. The service documentation covers some of these tradeoffs.

See (1) https://docs.microsoft.com/en-us/azure/documentdb/documentdb... and (2) https://docs.microsoft.com/en-us/azure/documentdb/documentdb... (also check out the references at the end of the page)

lobster_johnson 15 days ago [-]

Is Cosmos related to the work on Corfu/CorfuDB [2] [1] in any way?

[1] https://www.microsoft.com/en-us/research/publication/corfu-a...

[2] https://github.com/CorfuDB/CorfuDB

GovindMS 14 days ago [-]

AS Dharma mentioned - Azure Cosmos DB has been many years in the making. Azure Cosmos DB started as "Project Florence" in late 2010 to address developer the pain-points faced by large scale applications inside Microsoft. Observing that the challenges of building globally distributed apps are not a problem unique to Microsoft, in 2015 we made the first generation of this technology available to Azure developers in the form of DocumentDB.

The database engine design is inspired on LLAMA http://db.disi.unitn.eu/pages/VLDBProgram/pdf/research/p853-..., Bwtree - > https://pdfs.semanticscholar.org/7655/9c6cc259c6ab5baf7bd19d.... and schema-agnostic indexing techniques -> http://www.vldb.org/pvldb/vol8/p1668-shukla.pdf. Please note that these papers are significantly behind the current state of the implementation. The most crucial aspect that these papers dont cover is the integration of the database engine with the larger distributed system components of Cosmos DB including the resource governance, partition management, and the implementation of replication protocol /consistency models etc. Our goal is to publish all of the design specifications including TLA+ specs over time.

dharmashuklaMS 14 days ago [-]

Azure Cosmos DB is not related to the work on Corfu/CorfuDB.

willchen 15 days ago [-]

Very interesting DB service. If I'm reading the docs right it sounds like you can't do JOINs across documents?

https://docs.microsoft.com/en-us/azure/documentdb/documentdb...

extesy 15 days ago [-]

Not using DocumentDB API but you can kinda do joins using graph traversal Gremlin API.

rectalogic 14 days ago [-]

I don't see CosmosDB on the HIPAA compliance list, anyone know if there are plans to add it? https://www.microsoft.com/en-us/trustcenter/compliance/hipaa

GovindMS 14 days ago [-]

https://azure.microsoft.com/en-us/blog/azure-compliance-docu...

tracker1 15 days ago [-]

From the intro page[1]... Many of the descriptions comparing to NoSQL are wrong. There are plenty of NoSQL options that have similar features, though it isn't universal, it can and often is there. Cassandra, for example, probably does just as well in multi-zone/dc concurrency. Consistency options are also similarly tunable. Cockroach 1.0 was announced earlier as well.

It's not that I don't appreciate the option. This seems far closer to what DocumentDB should have been earlier on. Though tbh, I think Storage Tables are already pretty useful.

[1] https://docs.microsoft.com/en-us/azure/cosmos-db/introductio...

smithkl42 15 days ago [-]

Azure Table Storage? Uggh. Nasty. I've tried half a dozen times to use them, and every time I've given up. Great for write-only data that you never want to see again. Horrible for real-world querying.

dharmashuklaMS 15 days ago [-]

Cosmos DB has native extensibility to support various APIs Azure Table Storage "APIs" is one of them. If you are a Azure Table Storage customers, by virtue of

accessing Cosmos DB using the Table Storage API, you can now get all of the capabilities of Cosmos DB incl. automatic indexing, global distribution etc.

Your Table Storage queries should be really fast with Cosmos DB since, Cosmos DB supports efficient indexing and query.

tracker1 14 days ago [-]

Most of my usage has been mostly predictable single-key lookups, and some limited querying... The amount of data I've needed to put in it has always been fairly limited as well (generally under a million records per collection/table). It's a decent and really inexpensive option depending on your needs.

VikingCoder 15 days ago [-]

I'd like to see an in-depth comparison with Google Cloud Spanner.

afeezaziz 15 days ago [-]

Is there a real life case study of people/startup using Cloud Spanner? I am wondering what are the use cases of Spanner that cannot be fulfilled by other Google Cloud Platform products.

random3 15 days ago [-]

relational model globally distributed ACID transactions 99.999 availability

to name a few

arosenbaum 15 days ago [-]

Please post link where they talk about transactions - I can't find it in the docs anywhere. Availability SLA is clearly not 5 9's - https://azure.microsoft.com/en-us/support/legal/sla/cosmos-d...

nindalf 15 days ago [-]

random3 is replying to someone asking about Google's Spanner, which does have those transactions.

daxfohl 15 days ago [-]

They're very different things. It's like wanting an in-depth comparison between SQL Server and Redis.

0xFFC 15 days ago [-]

Exactly, although cloud is not my best suite, I am so exited. This is serious fight.

rattray 15 days ago [-]

I find this product slightly befuddling.

It seems like a "just throw all your data in this" kind of database, probably intended for everything but core application relational data (so, good for analytics, messaging, etc).

It sounds like the atom-record-sequence model at the heart of it is pretty key, but there's not a lot in the article about what that is and how it works. Is this a well-understood data structure used elsewhere?

The project seems very ambitious, and I could see it being used pretty heavily at a lot of companies. Thoughts?

datasage 15 days ago [-]

Its not uncommon to in sass stacks to have multiple data stores, each specialized for different use cases. I do see it appealing to have one system that can support each type of use case and able to tune that as needed.

hoodoof 15 days ago [-]

Does it provide search?

It's a strange thing, but almost all new database technologies seem to leave search as an afterthought for some later day instead of starting on day one with the assumption that "it's

all about search".

A database system that doesn't support rich search capabilities is restricted to very limited types of applications.

Often search is left unimplemented for years, or perhaps never implemented.

liamca 14 days ago [-]

[Full disclosure, I work on the Azure Search team]

Cosmos DB has tight integration with Azure Search, to the point that you can choose to extend your Cosmos DB to Azure Search with a few clicks right from the portal to allow for full text search over this content.

You can learn more about how to do this at these two links:
https://azure.microsoft.com/en-us/blog/adding-search-to-docu...

https://docs.microsoft.com/en-us/azure/search/search-howto-i...

Liam

harigov 15 days ago [-]

It does support search through SQL like queries. You can use existing functions or implement new ones in Javascript. If you want free text search, you would be disappointed, but otherwise it is pretty decent.

hoodoof 15 days ago [-]

So half marks on search for Azure Cosmos DB.

If free text search is such a hard problem then you'd think that would be even more reason to start with solving that toughest of all problems. If it's a really hard problem then it will be even harder to retrofit later into some system that is already architected and built.

A light spanking for the architect. No search would have been a thorough spanking.

harigov 14 days ago [-]

The good news is that you can configure all the data to be copied over to Azure Search on a regular basis and then use search capabilities provided by Azure Search. This may sound like a complicated thing to do but the functionality provided by Azure makes it ridiculously easy to configure something like this.

hoodoof 14 days ago [-]

Search should be a first class feature of a database not an addon. Every.Single. Time. I have ever needed to "just add on search" it has resulted in deep pain.

Nothing you say will convince me that it's just a super easy dance in the daisies to "just do X" or "just do Y" and suddenly its database search heaven.

Database application architects need to wise up that the world revolves around search - built in, not add on.

edward_rolf 14 days ago [-]

Search is not hard. Microsoft are being lazy. Which is good for me. I'm building what I thought was a Elasticsearch killer. Seems it's also a DocumentDB killer. Search is first-class with Resin:
https://github.com/kreeben/resin

sargun 14 days ago [-]

Does CosmosDB have any relationship to Microsoft Cosmos (http://web.stanford.edu/class/ee380/Abstracts/111026a-Hellan...)? Or is this another case of Dynamo / DynamoDB?

GovindMS 14 days ago [-]

Hi ! Sargun, If you see Dharma's detailed reply which consists of papers and history from 2010. This is a different effort with a different focus.

Azure Cosmos DB has been many years in the making. Azure Cosmos DB started as "Project Florence" in late 2010 to address developer the pain-points faced by large scale applications inside Microsoft. Observing that the challenges of building globally distributed apps are not a problem unique to Microsoft, in 2015 we made the first generation of this technology available to Azure developers in the form of DocumentDB. Since that time, we've been steadily adding new capabilities both in the database engine as well as, larger distributed system components. Azure Cosmos DB is the result. It is the next big leap in globally distributed, at scale, cloud databases. As a part of this release of Azure Cosmos DB, DocumentDB customers, with their data, are automatically Azure Cosmos DB customers. They now have access to the new system and capabilities offered by Azure Cosmos DB today as well as, as we keep evolving the service.

The database engine design is inspired on LLAMA http://db.disi.unitn.eu/pages/VLDBProgram/pdf/research/p853-..., Bwtree - > https://pdfs.semanticscholar.org/7655/9c6cc259c6ab5baf7bd19d.... and schema-agnostic indexing techniques -> http://www.vldb.org/pvldb/vol8/p1668-shukla.pdf. Please note that these papers are significantly behind the current state of the implementation. The most crucial aspect that these papers dont cover is the integration of the database engine with the larger distributed system components of Cosmos DB including the resource governance, partition management, and the implementation of replication protocol /consistency models etc. Our goal is to publish all of the design specifications including TLA+ specs over time.

dagi3d 15 days ago [-]

If I understood it correctly, they mentioned they offer horizontal scalibility for their databases and I wonder how does it work for the graph data model

aravindkr1 15 days ago [-]

This is covered in https://docs.microsoft.com/azure/cosmos-db/gremlin-support. You can specify a partition key for your graphs for scale out, and access vertices and edges using the partition key + item key ("id") like g.V(['USA', 'Seattle']).

hoodoof 15 days ago [-]

Databases is one area that Amazon is way behind Google and Microsoft. DynamoDB is thoroughly awful so good to see some competition.

jupp0r 14 days ago [-]

> something no other database service can offer.

Needs to be updated since Spanner was released

dmarlow 15 days ago [-]

I want a competitor to Google Cloud SQL in Azure.

curiousDog 15 days ago [-]

It's called Azure SQL DB (unless you want Mysql, then you'd want IaaS SQL DB)

dmarlow 15 days ago [-]

With the exception that Azure SQL doesn't horizontally scale across regions/datacenters like Google Cloud SQL does.

curiousDog 15 days ago [-]

I don't think Google Cloud SQL does either in a fully managed way. Spanner OTOH does that but it's NewSQL. Traditional sharding has been supported for a while: https://docs.microsoft.com/en-us/azure/sql-database/sql-data.... you can create databases where ever you want.

martinknafve 15 days ago [-]

No backup/restore?

dharmashuklaMS 15 days ago [-]

Cosmos DB does local persistence and replication both within a region and across any number of regions.All data is durable and made highly available via replication. You dont need to take backups or restore them for ensuring durability or availability. See (1) https://docs.microsoft.com/en-us/azure/cosmos-db/introductio... and (2) https://docs.microsoft.com/en-us/azure/cosmos-db/introductio.... That said, if you need backup/restore for the cases where you accidentally delete your data and want to resurrect it, Cosmos DB automatically takes backups for you periodically.

> martinknafve 14 days ago [-]
>
> Actually, I wish Microsoft would stop referring to replication when asked about backup. It's the modern way of saying "You probably don't need backup, because you have RAID". There's a reason Azure SQL Database has self-service point-in-time-recovery despite also having replication.
>
> Do you see many use cases where there is no need for backup to protect against accidental deletion, overwriting, deletion by application vulnerabilities and so on?
>
> As far as I understood the backup docs, I should contact Microsoft Support within 8 hours if any of those things happens. Is that still correct? What if we don't notice the issue until 7 days later?
>
> > Ecio78 14 days ago [-]
> >
> > Totally agree with you. Even if you have delayed copies, you still need proper backup in place. They recently implemented long term retention for Azure SQL backups in Azure: https://azure.microsoft.com/en-us/blog/azure-sql-database-no... as it was only allowing 'til 35 days. I would expect that something similar is provided also for this type of DBs
> >
> > > martinknafve 14 days ago [-]
> > >
> > > One can hope. Blob and table storage still have no form of backup ~8 years after introduction.
> > >
> > > When I've asked they have been referring to replication and to call them if we accidentally lose data. But we need to contact them within two hours otherwise it's too late. And of course Azure Support never responds that quickly when I submit a case to them.

oedenfield 15 days ago [-]

Backup is automatic and restore is available within a time frame. https://docs.microsoft.com/en-us/azure/documentdb/documentdb...

> martinknafve 15 days ago [-]
>
> > If the data is accidently dropped or corrupted, please contact Azure support within 8 hours
>
> Thats sounds less then optimal. :)
>
> > dharmashuklaMS 13 days ago [-]
> >
> > For supporting the "oops I accidentally deleted my table/collection/graph - how do I get it back?" scenario, Cosmos DB automatically takes periodic, full backups of all your data. Currently, we restore your backups on demand and its free - you just send a mail to us. In a few months, we will expose an API using which you can restore the data you accidentally deleted yourself.
> >
> > reply

bpicolo 15 days ago [-]

They guarantee durability. I imagine that's a part of it? I guess no backup/restore is a problem if you manage to delete all your data, so we'd have to see some docs, heh

> martinknafve 15 days ago [-]

At some point someone will accidentally overwrite or delete data and discover that some days later...

chaostheory 15 days ago [-]

[flagged]

cheeze 15 days ago [-]

> Is it open source?

Probably not

> Why do I care?

You might not.

> The same reason I don't use AWS or Google Cloud only datastores: I want to avoid vendor lockin.

Sounds like this isn't the product for you then.

Is that a surprise?

chaostheory 15 days ago [-]

My questions were rhetorical. One reason I post my thoughts here is because companies like MS are on HN. It's useful to know why people don't want to use your product or are hesitant to do so. Besides, MS has been doing a lot of surprising things lately post-Ballmer.

obmelvin 15 days ago [-]

This is meant for people who need services and guarantees that can only be provided by a huge infrastructure provider

chaostheory 15 days ago [-]

You can say the same for pretty much every other managed datastore service

oedenfield 15 days ago [-]

See the news about Azure Database for MySQL and Postgresql
https://azure.microsoft.com/en-us/services/postgresql/
https://azure.microsoft.com/en-us/services/mysql

Thaxll 15 days ago [-]

Vendor lock in is a bad example, it's not like you're going to move your infrastructure from AWS to Google every 6months. And if you do you will have many more complicated problems than vendor lock in.

chaostheory 15 days ago [-]

Just having an easier option to move gives you a better advantage for pricing and it gives the service provider more of an incentive to do more to keep you on their platform.

polskibus 15 days ago [-]

You could if you just used managed postgresql or iaas.

TotallyHuman 14 days ago [-]

Except Cosmos DB in no way compares to those.

Dimi9909 15 days ago [-]

is this similar to DynamoDB in AWS?

anand_MSFT 15 days ago [-]

See https://www.youtube.com/watch?v=Yfmw7swCtZs by Turing Award Winner, Dr. Leslie Lamport, as he talks about Azure Cosmos DB