**Azure Cosmos DB setup**

Today's applications are required to be highly responsive and always online. To achieve low latency and high availability, instances of these applications need to be deployed in datacenters that are close to their users. Applications need to respond in real time to large changes in usage at peak hours, store ever increasing volumes of data, and make this data available to users in milliseconds.

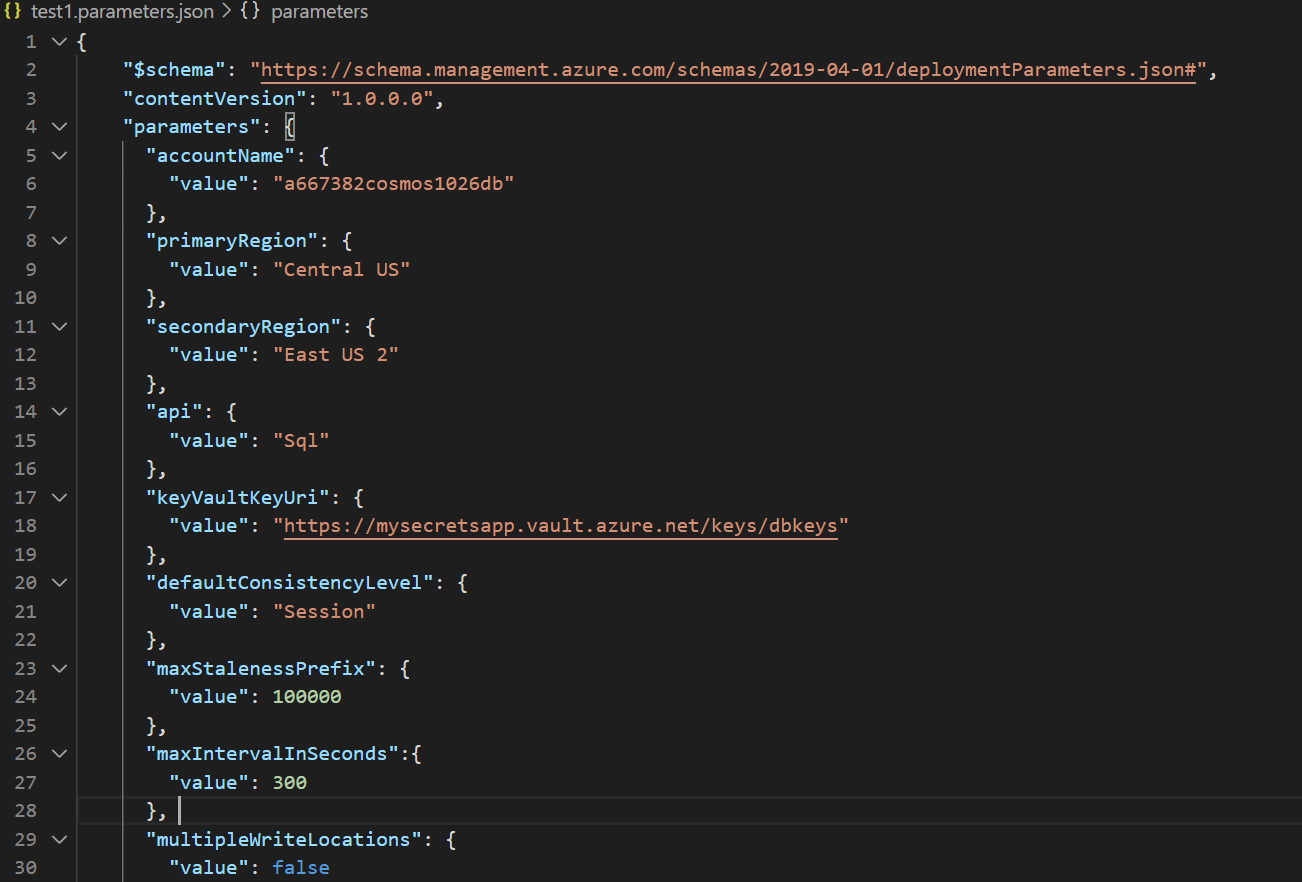
***Azure Cosmos DB is***

* A fully managed NoSQL database for modern app development.
* Single-digit millisecond response times, and automatic and instant scalability, guarantee speed at any scale.
* Business continuity is assured with SLA-backed availability and enterprise-grade security.
* App development is faster and more productive thanks to turnkey multi-master data distribution anywhere in the world, open source APIs and SDKs for popular languages.

As a fully managed service, Azure Cosmos DB takes database administration off your hands with automatic management, updates and patching. It also handles capacity management with cost-effective serverless and automatic scaling options that respond to application needs to match capacity with demand.

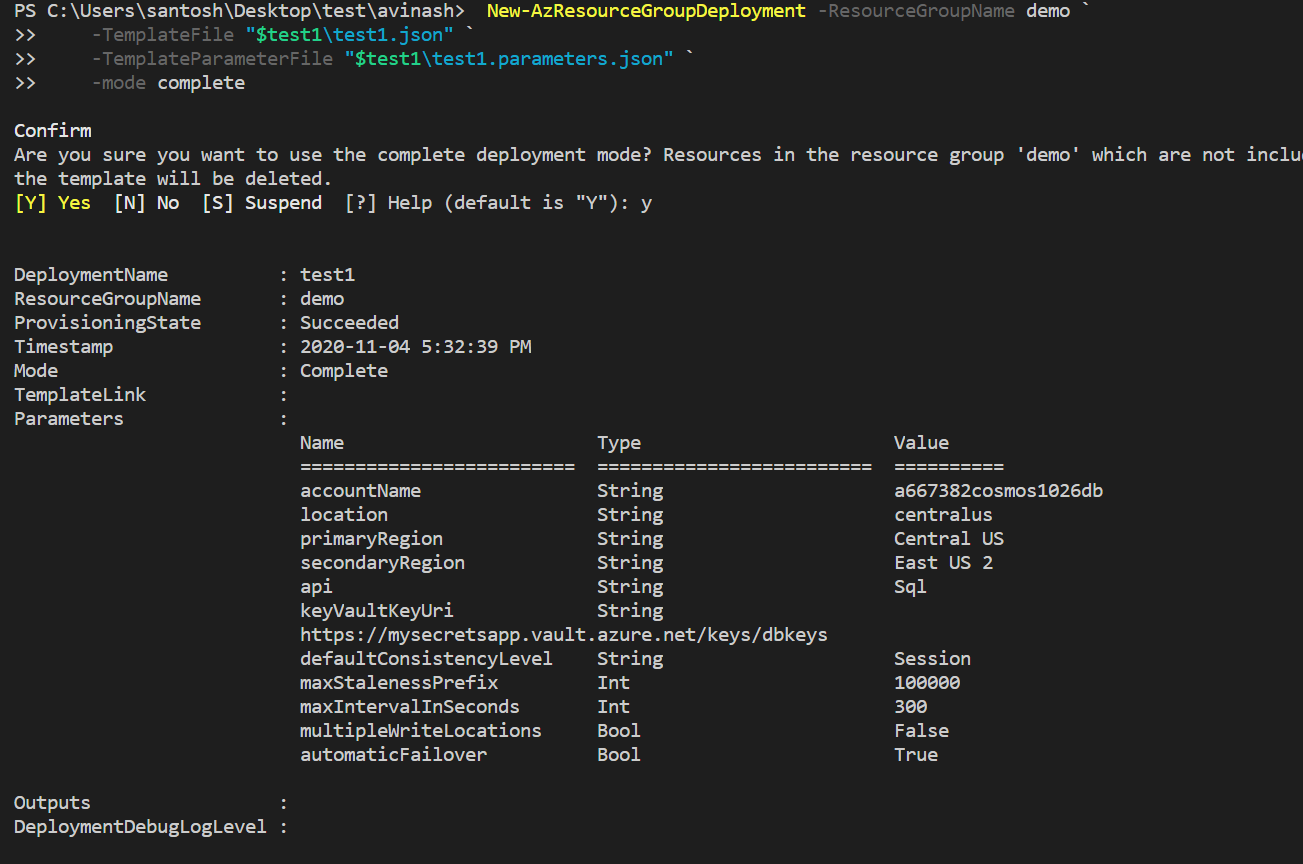
Arm template for Azure Cosmos DB setup:



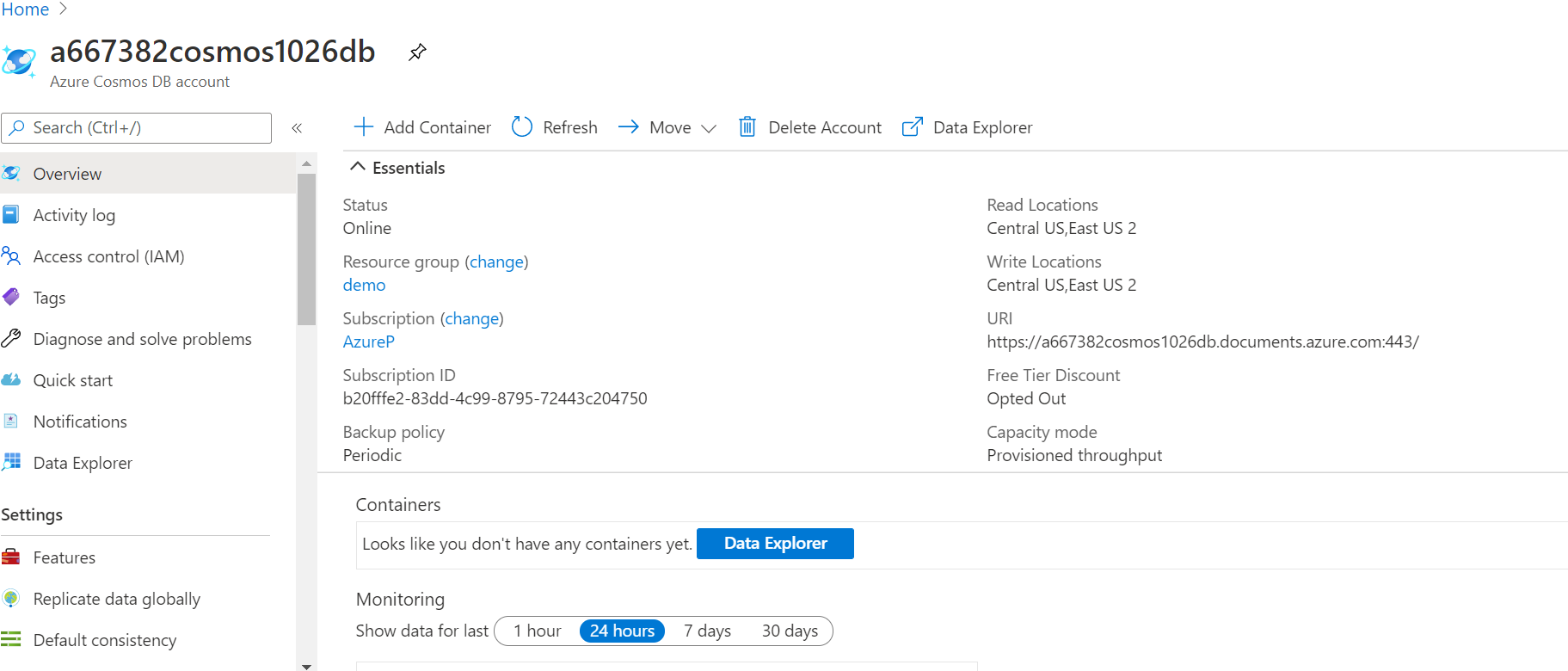


Template file and parameters files has been added to the user story and you can access from git repo

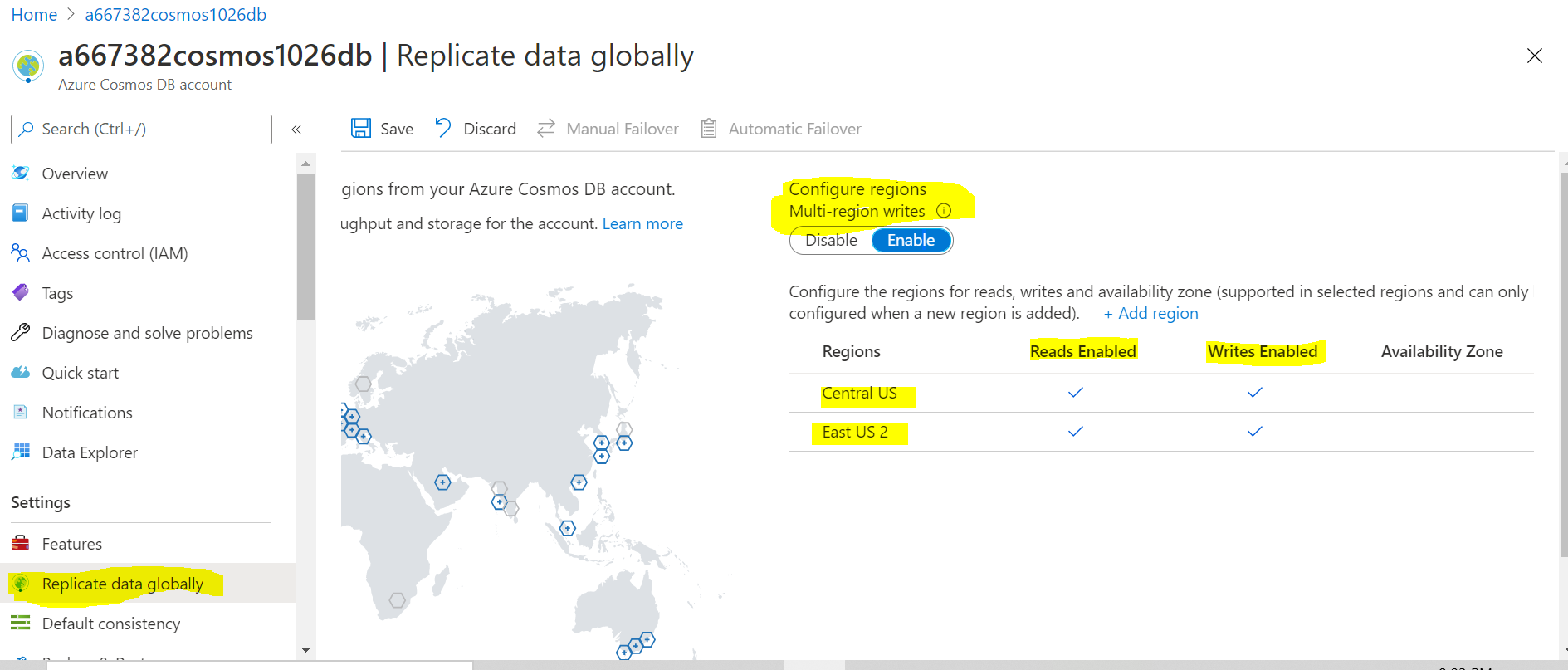
The Deployment was successful and the resources got deployed



In Azure portal, go to the azure cosmos db account and click on overview where you can see service was setup in two regions Central US and East US 2

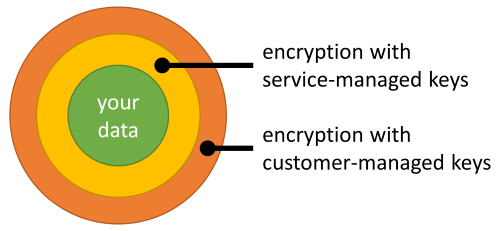


From Azure portal you can see the features such as replicate data globally, this cosmos db can read and write from multiple regions(Multi region writes are Enabled)



**Data encryption using CMK(customer managed keys)**

Data stored in your Azure Cosmos account is automatically and seamlessly encrypted with keys managed by Microsoft (service-managed keys). Optionally, you can choose to add a second layer of encryption with keys you manage (customer-managed keys).



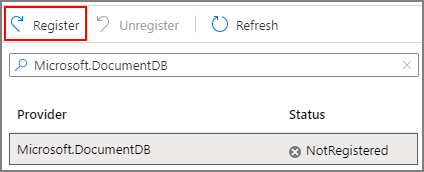
You must store customer-managed keys in Azure Key Vault and provide a key for each Azure Cosmos account that is enabled with customer-managed keys. This key is used to encrypt all the data stored in that account.

**Note**

Currently, customer-managed keys are available only for new Azure Cosmos accounts. You should configure them during account creation.

## **Register the Azure Cosmos DB resource provider for your Azure subscription**

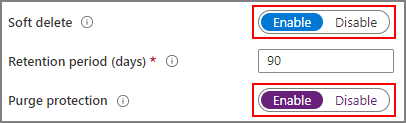
1. Sign in to the Azure portal, go to your Azure subscription, and select Resource providers under the Settings tab:
2. Search for the Microsoft.DocumentDB resource provider. Verify if the resource provider is already marked as registered. If not, choose the resource provider and select Register:



## **Configure your Azure Key Vault instance**

Using customer-managed keys with Azure Cosmos DB requires you to set two properties on the Azure Key Vault instance that you plan to use to host your encryption keys: Soft Delete and Purge Protection.

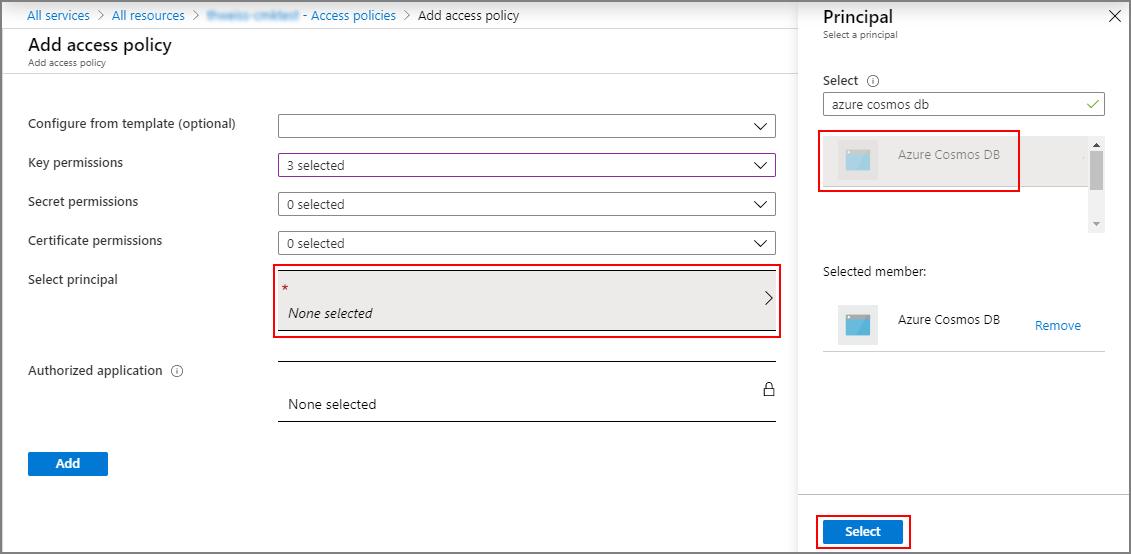
If you create a new Azure Key Vault instance, enable these properties during creation:



If you're using an existing Azure Key Vault instance, you can verify that these properties are enabled by looking at the Properties section on the Azure portal.

## **Add an access policy to your Azure Key Vault instance**

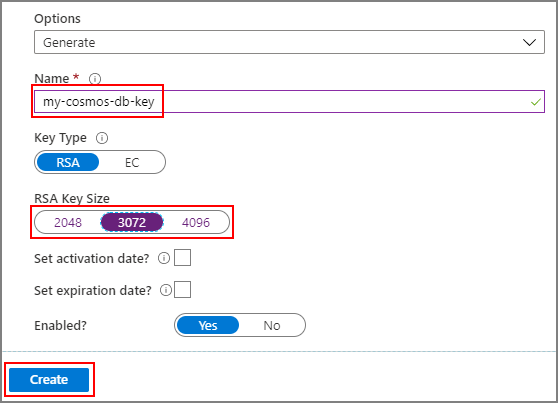
1. From the Azure portal, go to the Azure Key Vault instance that you plan to use to host your encryption keys. Select Access Policies from the left menu:
2. Select + Add Access Policy.
3. Under the Key permissions drop-down menu, select Get, Unwrap Key, and Wrap Key permissions:
4. Under Select principal, select None selected. Then, search for Azure Cosmos DB principal and select it (to make it easier to find, you can also search by principal ID: a232010e-820c-4083-83bb-3ace5fc29d0b for any Azure region except Azure Government regions where the principal ID is 57506a73-e302-42a9-b869-6f12d9ec29e9). Finally, choose Select at the bottom. If the Azure Cosmos DB principal isn't in the list, you might need to re-register the Microsoft.DocumentDB resource provider as described in the Register the resource provider section.



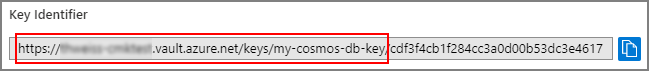
1. Select Add to add the new access policy.
2. Select Save on the Key Vault instance to save all changes.

## **Generate a key in Azure Key Vault**

1. From the Azure portal, go to the Azure Key Vault instance that you plan to use to host your encryption keys. Then, select Keys from the left menu:
2. Select Generate/Import, provide a name for the new key, and select an RSA key size. A minimum of 3072 is recommended for best security. Then select Create:



1. After the key is created, select the newly created key and then its current version.
2. Copy the key's Key Identifier, except the part after the last forward slash:



This key identifier is copied and provided in the arm template\

After the deployment you can check this feature in azure portal, go to azure cosmos db account and click on Data Encryption

