DXC AZURE ANALYTICS

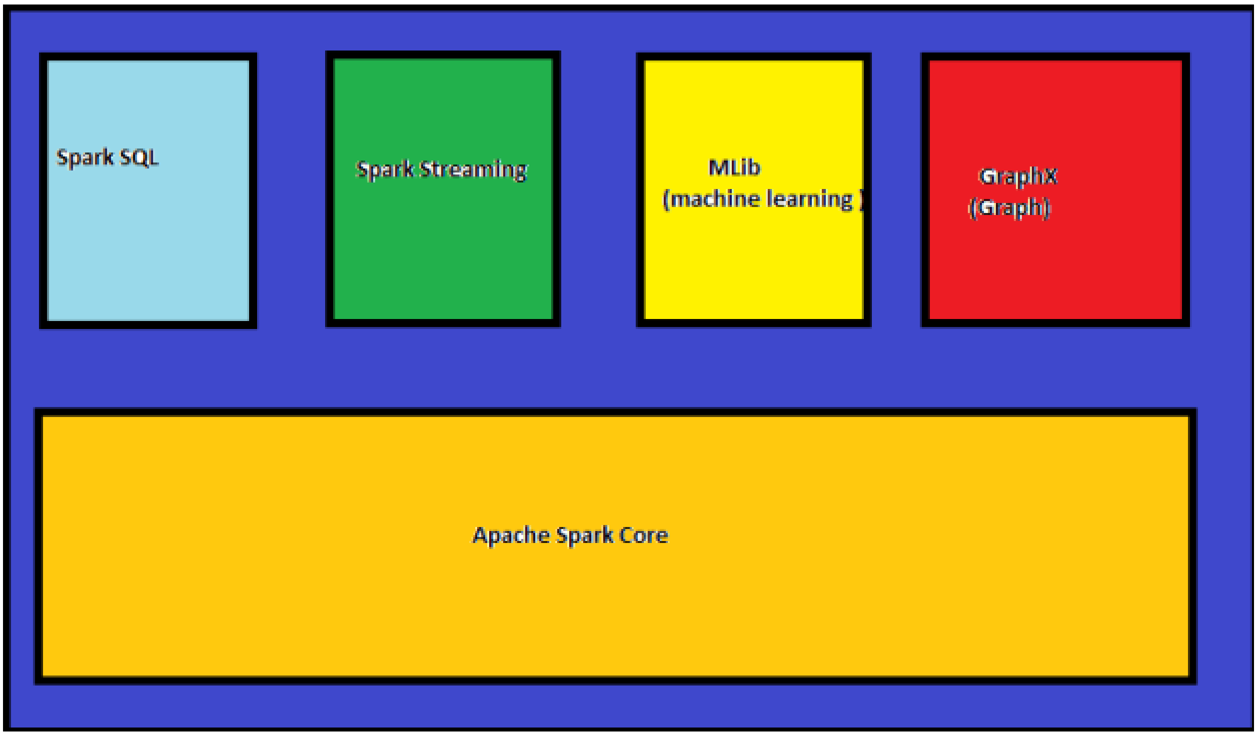
Assignment – 7

Name: Nandi Vomkara Aditya Mohan ID : DXCAB12003

Date of submission: 07-06-2022 Batch : DXC-Azure-analytics

1. Explain what are various components of SPARK with block diagram? explain functionality of every components?

Ans)



Apache Spark Core

Spark Core is the underlying general execution engine for spark platform that all other functionality is built upon. It provides In-Memory computing and referencing datasets in external storage systems.

Spark SQL

Spark SQL is a component on top of Spark Core that introduces a new data abstraction called SchemaRDD, which provides support for structured and semi-structured data.

Spark Streaming

Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD (Resilient Distributed Datasets) transformations on those mini batches of data.

MLlib (Machine Learning Library)

MLlib is a distributed machine learning framework above Spark because of the distributed memory-based Spark architecture. It is, according to benchmarks, done by the MLlib developers against the Alternating Least Squares (ALS) implementations. Spark MLlib is nine times as fast as the Hadoop disk-based version of Apache Mahout (before Mahout gained a Spark interface).

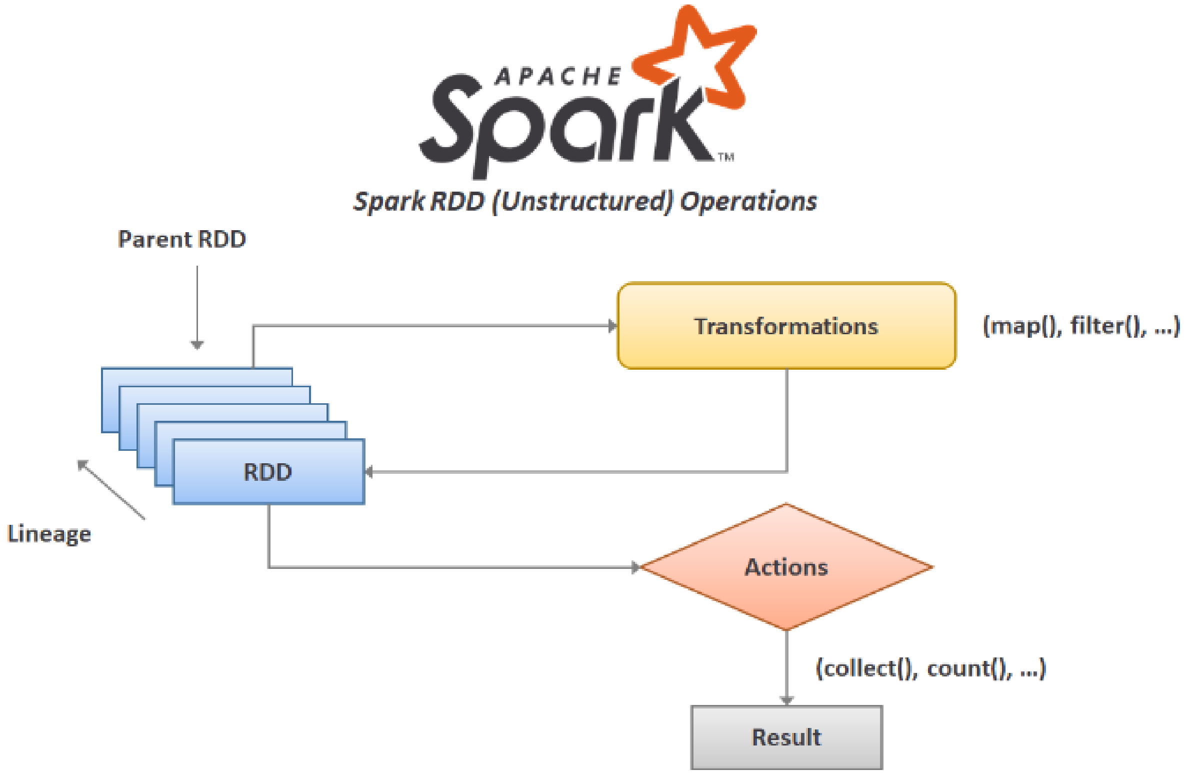
GraphX

GraphX is a distributed graph-processing framework on top of Spark. It provides an API for expressing graph computation that can model the user-defined graphs by using Pregel abstraction API. It also provides an optimized runtime for this abstraction.

2. Explain Spark core in details & how RDD is related to Spark core - explain with Spark program?

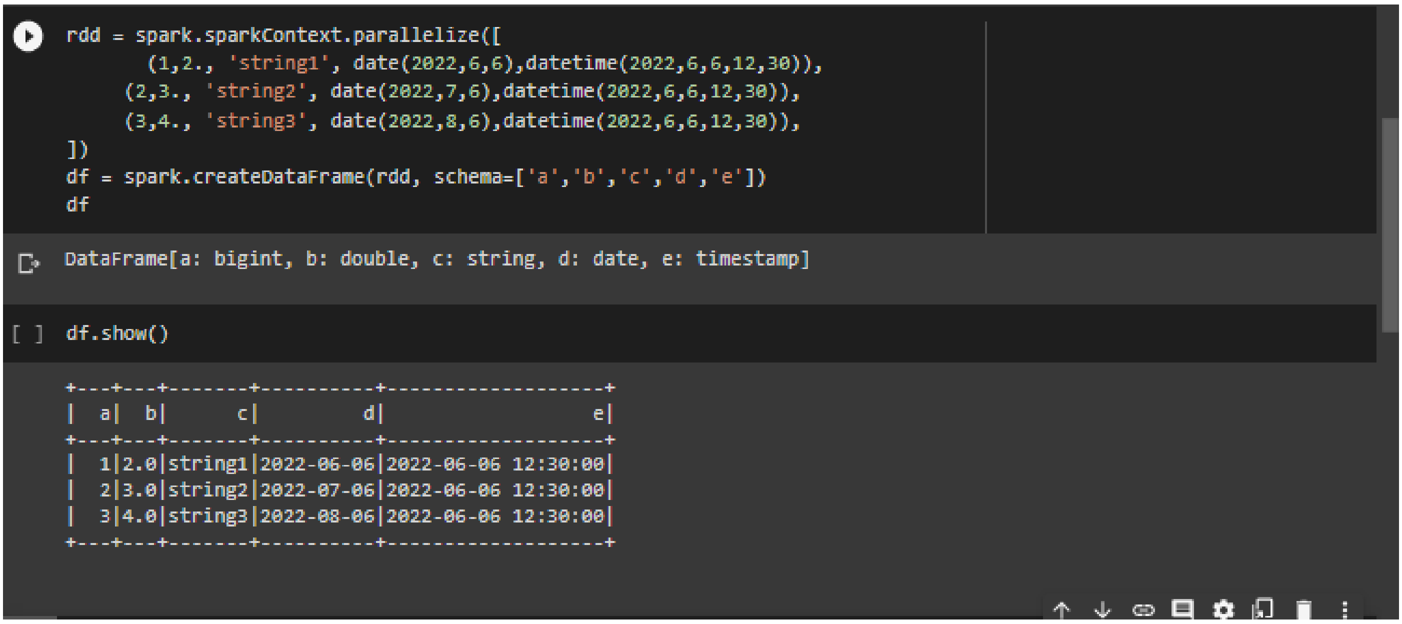
Ans) **Apache Spark Core**

Spark Core is the underlying general execution engine for spark platform that all other functionality is built upon. It provides In-Memory computing and referencing datasets in external storage systems.Spark is embedded with RDD(resilient distributed datasets) an immutable fault tolerant, distributed collection of objects that can be operatedon in parallel.



Spark RDD Operation Schema

And the following program describes the how rdd is related with spark



```
rdd = spark.sparkContext.parallelize([
       (1,2., 'string1', date(2022,6,6),datetime(2022,6,6,12,30)),
     (2,3., 'string2', date(2022,7,6),datetime(2022,6,6,12,30)),
     (3,4., 'string3', date(2022,8,6),datetime(2022,6,6,12,30)),
])
df = spark.createDataFrame(rdd, schema=['a','b','c','d','e'])
df
```

```
DataFrame[a: bigint, b: double, c: string, d: date, e: timestamp]
```

```
[ ] df.show()
```

```
+---+---+-------+----------+-------------------+
|  a|  b|      c|         d|                  e|
+---+---+-------+----------+-------------------+
|  1|2.0|string1|2022-06-06|2022-06-06 12:30:00|
|  2|3.0|string2|2022-07-06|2022-06-06 12:30:00|
|  3|4.0|string3|2022-08-06|2022-06-06 12:30:00|
+---+---+-------+----------+-------------------+
```

3. Explain various Mlib algorithms Spark is supporting?

Ans) Spark.ml is the primary Machine Learning API for Spark. The library Spark.ml offers a higher-level API built on top of Data Frames for constructing ML pipelines.

Spark MLlib tools are given below:

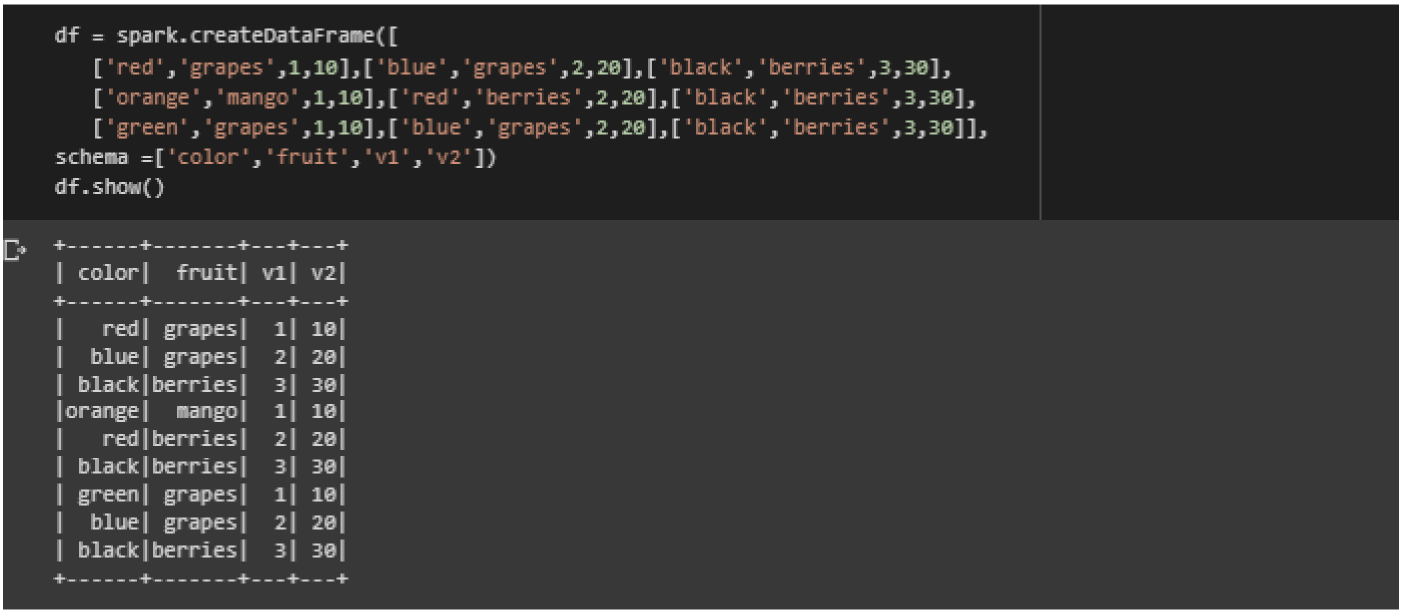
1. ML Algorithms
2. Featurization
3. Pipelines
4. Persistence
5. Utilities

4. Explain benefits Spark SQL & how relational data will be inserted into SPARK?

Ans) Spark SQL brings native support for SQL to Spark and streamlines the process of querying data stored both in RDDs (Spark’s distributed datasets) and in external sources. Spark SQL conveniently blurs the lines between RDDs and relational tables. Unifying these powerful abstractions makes it easy for developers to intermix SQL commands querying external data with complex analytics, all within in a single application. Concretely, Spark SQL will allow developers to:

- Import relational data from Parquet files and Hive tables
- Run SQL queries over imported data and existing RDDs
- Easily write RDDs out to Hive tables or Parquet files

Spark SQL also includes a cost-based optimizer, columnar storage, and code generation to make queries fast. At the same time, it scales to thousands of nodes and multi-hour queries using the Spark engine, which provides full mid-query fault tolerance, without having to worry about using a different engine for historical data.



```
df = spark.createDataFrame([
   ['red','grapes',1,10],['blue','grapes',2,20],['black','berries',3,30],
   ['orange','mango',1,10],['red','berries',2,20],['black','berries',3,30],
   ['green','grapes',1,10],['blue','grapes',2,20],['black','berries',3,30]],
schema =['color','fruit','v1','v2'])
df.show()
```

```
+------+-------+---+---+
| color|  fruit| v1| v2|
+------+-------+---+---+
|   red| grapes|  1| 10|
|  blue| grapes|  2| 20|
| black|berries|  3| 30|
|orange|  mango|  1| 10|
|   red|berries|  2| 20|
| black|berries|  3| 30|
| green| grapes|  1| 10|
|  blue| grapes|  2| 20|
| black|berries|  3| 30|
+------+-------+---+---+
```

5.Explain Spark streaming in detail ?

Ans) Apache Spark Streaming is a scalable fault-tolerant streaming processing system that natively supports both batch and streaming workloads. Spark Streaming is an extension of the core Spark API that allows data engineers and data scientists to process real-time data from various sources including (but not limited to) Kafka, Flume, and Amazon Kinesis. This processed data can be pushed out to file systems, databases, and live dashboards. Its key abstraction is a Discretized Stream or, in short, a DStream, which represents a stream of data divided into small batches. DStreams are built on RDDs, Spark's core data abstraction. This allows Spark Streaming to seamlessly integrate with any other Spark components like MLlib and Spark SQL. Spark Streaming is different from other systems that either have a processing engine designed only for streaming, or have similar batch and streaming APIs but compile internally to different engines. Spark's single execution engine and unified programming model for batch and streaming lead to some unique benefits over other traditional streaming systems.

Four Major Aspects of Spark Streaming

- Fast recovery from failures and stragglers
- Better load balancing and resource usage
- Combining of streaming data with static datasets and interactive queries
- Native integration with advanced processing libraries (SQL, machine learning, graph processing)

6. Explain SPARK architecture? what is Master - Slave architecture ?

Ans)

The Spark follows the master-slave architecture. Its cluster consists of a single master and multiple slaves.

The Spark architecture depends upon two abstractions:

- Resilient Distributed Dataset (RDD)
- Directed Acyclic Graph (DAG)

Resilient Distributed Datasets (RDD)

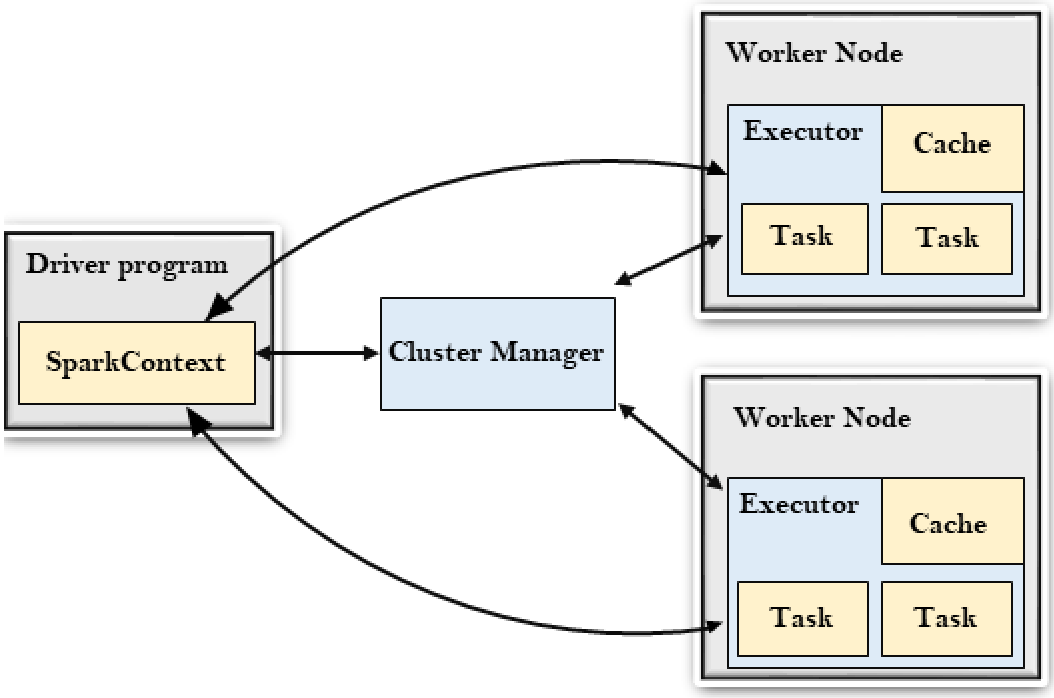
The Resilient Distributed Datasets are the group of data items that can be stored in-memory on worker nodes. Here,

- Resilient: Restore the data on failure.
- Distributed: Data is distributed among different nodes.
- Dataset: Group of data.

We will learn about RDD later in detail.

Directed Acyclic Graph (DAG)

Directed Acyclic Graph is a finite direct graph that performs a sequence of computations on data. Each node is an RDD partition, and the edge is a transformation on top of data. Here, the graph refers the navigation whereas directed and acyclic refers to how it is done.



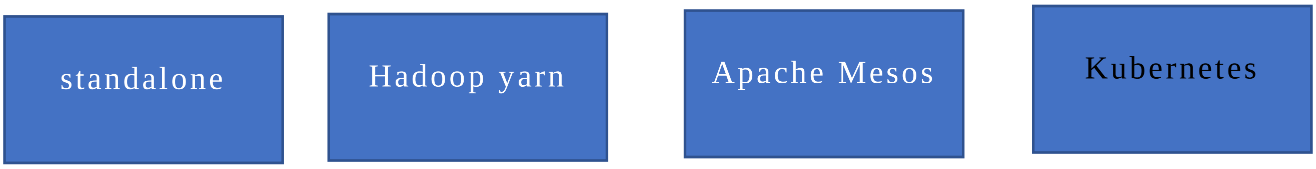
7. Explain various cluster managers in SPARK?

Ans) **Cluster manager** is a platform (cluster mode) where we can run Spark. Simply put, cluster manager provides resources to all worker nodes as per need, it operates all nodes accordingly.
We can say there are a master node and worker nodes available in a cluster. That master nodes provide an efficient working environment to worker nodes.

There are three types of Spark cluster manager. Spark supports these cluster manager:

1. Standalone cluster manager
2. Hadoop Yarn
3. Apache Mesos
4. Kubernetes

Apache Spark also supports pluggable cluster management. The main task of cluster manager is to provide resources to all applications. We can say it is an external service for acquiring required resources on the cluster.



standalone

Hadoop yarn

Apache Mesos

Kubernetes

8. Explain with sceenshots & steps how to create Cosmos DB ?

Ans) Cosmos data base is azures no sql data base. To create the cosmos db we have to allow the following steps

Step-1: we have to login to the Microsoft Azure account with our credentials

Step-2: After login to the azure portal search for COSMOS DB in the search bar. Follow the fig 8.1 to have a clear understanding about the process

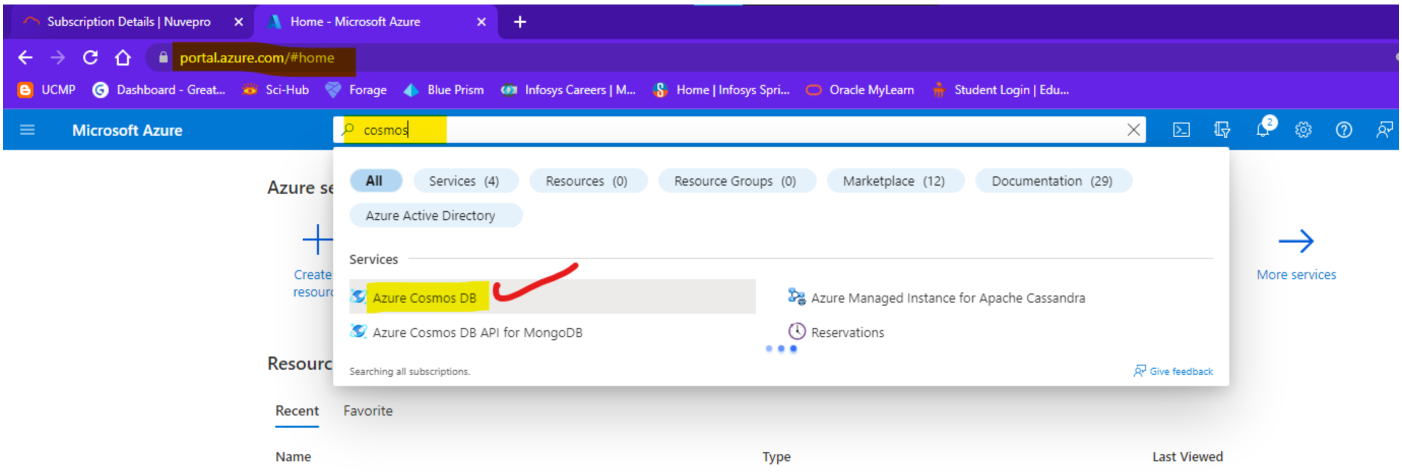


Fig-8.1 shows the searching & selection of cosmos DB

Step-3: click on COSMOS DB and click on Create button. Follow the fig 8.2 to have a clear understanding about the process

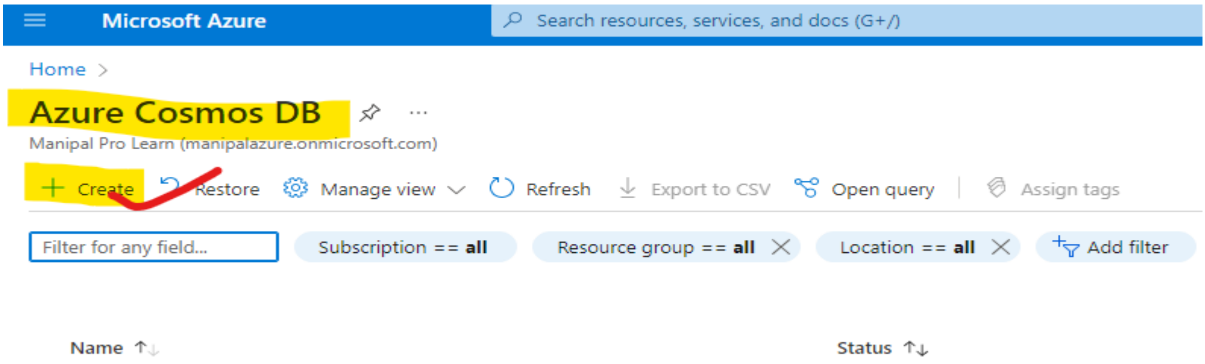
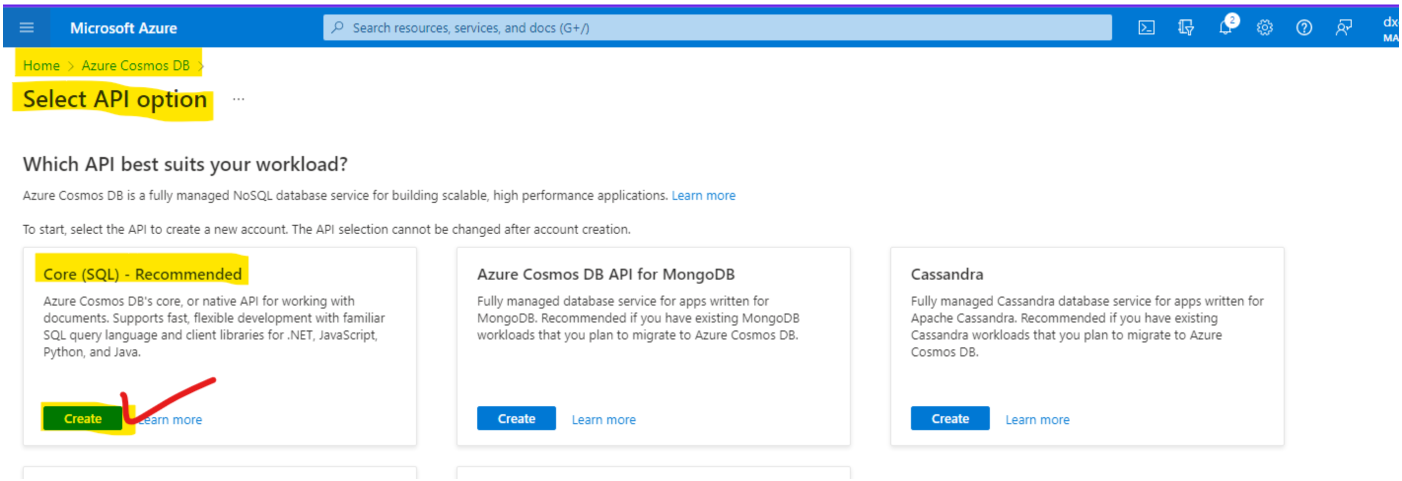


fig-8.2 shows the creation of COSMOS DB

Step-4: Select The API option and select the CORE(SQl) which is a recommended one. . Follow the fig 8.2 to have a clear understanding about the process



Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. Learn more

To start, select the API to create a new account. The API selection cannot be changed after account creation.

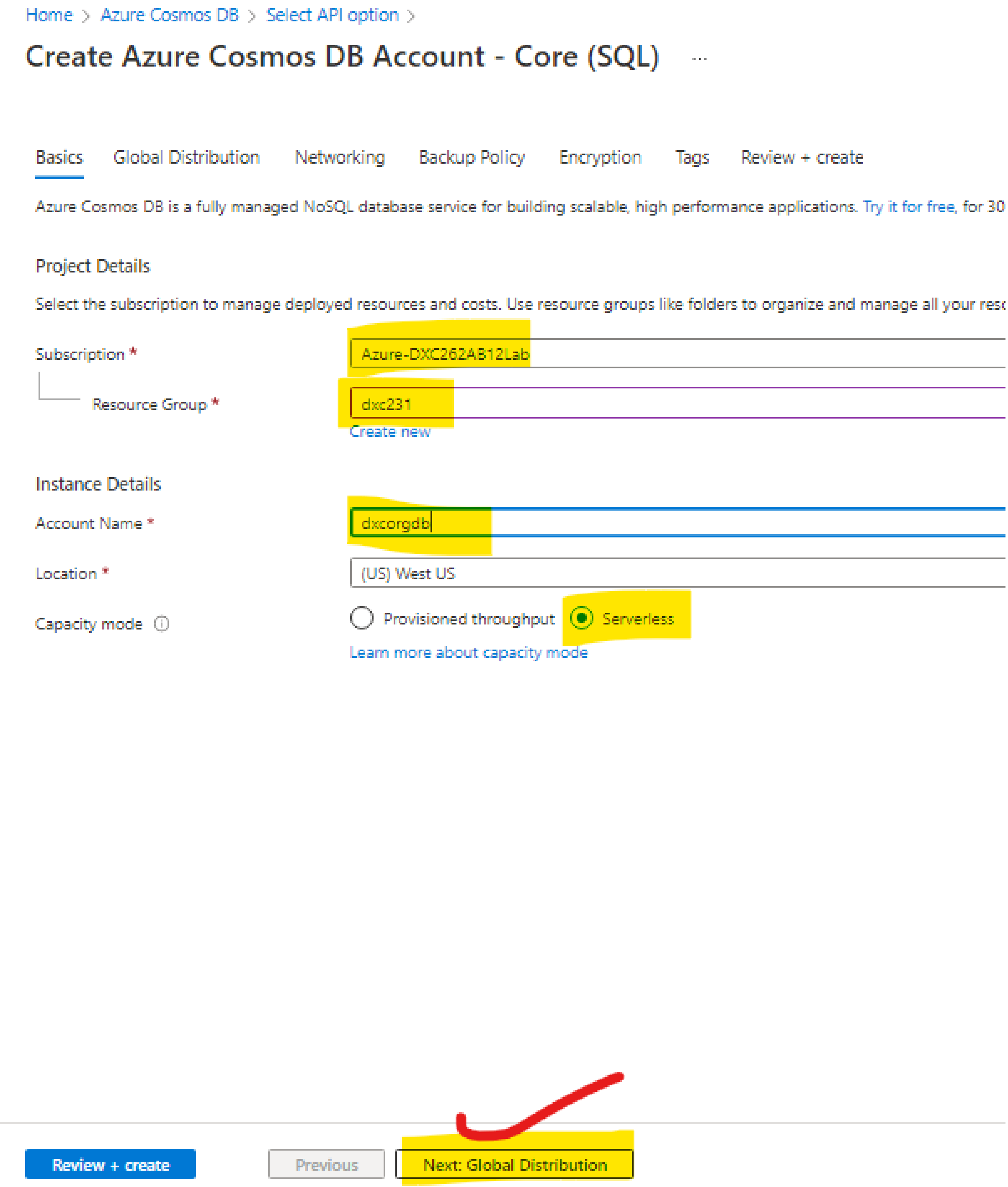
Core (SQL) - Recommended: Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.

Azure Cosmos DB API for MongoDB: Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.

Cassandra: Fully managed Cassandra database service for apps written for Apache Cassandra. Recommended if you have existing Cassandra workloads that you plan to migrate to Azure Cosmos DB.

Fig: 8.3 shows the selection of API for Cosmos DB

Step-5: Fill all the basic details required as shown in the figure below.



Create Azure Cosmos DB Account - Core (SQL) ...

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. Try it for free, for 30

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your reso

Subscription * Azure-DXC262AB12Lab

Resource Group * dxc231

Create new

Instance Details

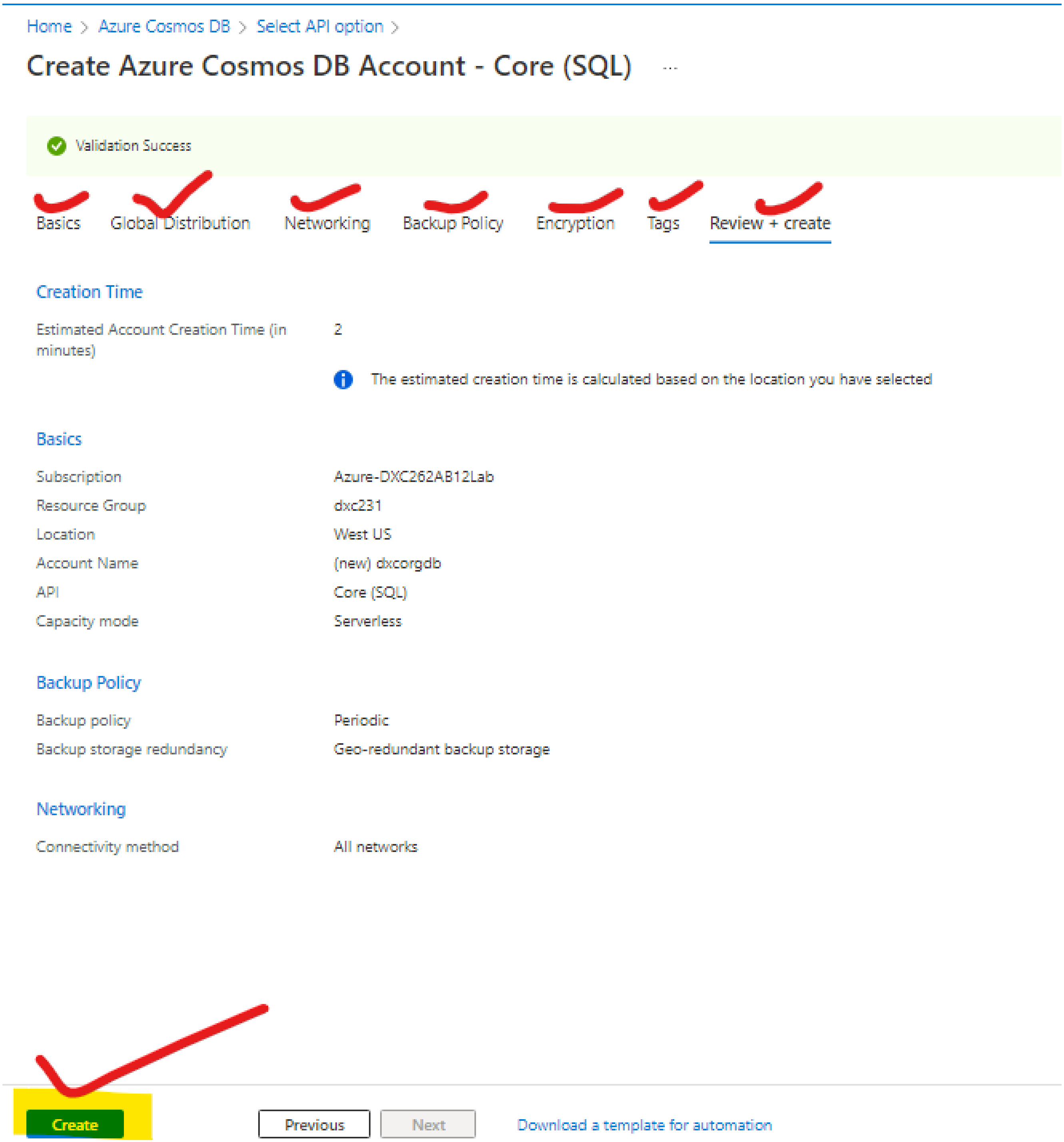
Account Name * dxcorgdb

Location * (US) West US

Capacity mode ⓘ ○ Provisioned throughput ● Serverless

Learn more about capacity mode

Step-6: after that ensure all the next windows to be as usual because there is no need of update requirement till now and follow the steps



Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL) ...

✓ Validation Success

Creation Time

Estimated Account Creation Time (in minutes)	2

ⓘ The estimated creation time is calculated based on the location you have selected

Basics

Subscription	Azure-DXC262AB12Lab
Resource Group	dxc231
Location	West US
Account Name	(new) dxcorgdb
API	Core (SQL)
Capacity mode	Serverless

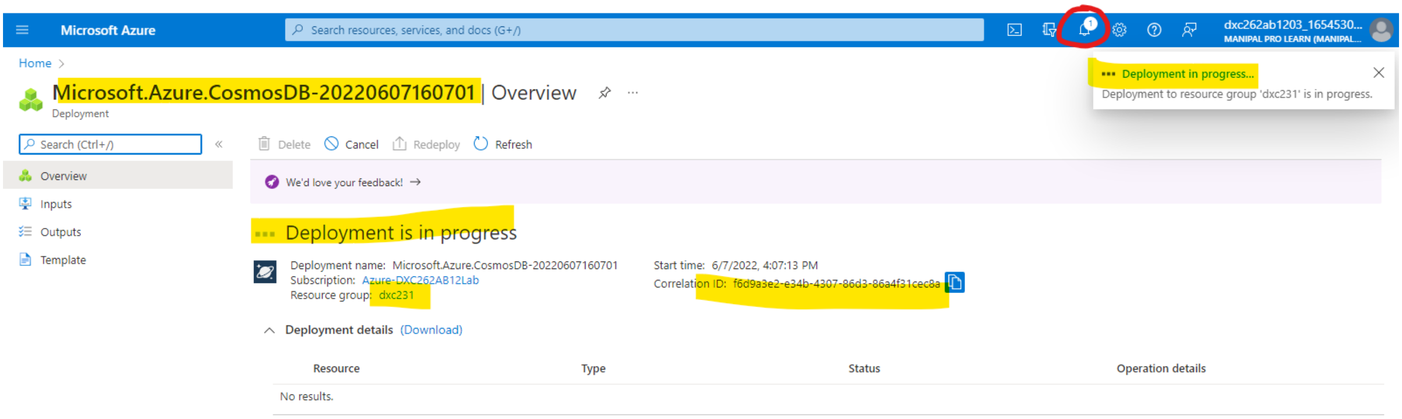
Backup Policy

Backup policy	Periodic
Backup storage redundancy	Geo-redundant backup storage

Networking

Connectivity method	All networks

Step-7: after that the deployment will be in progress . And it may takes some time. after that you can access the cosmos DB.



Microsoft.Azure.CosmosDB-20220607160701 | Overview

Deployment

Deployment in progress...

Deployment to resource group 'dxc231' is in progress.

Deployment is in progress

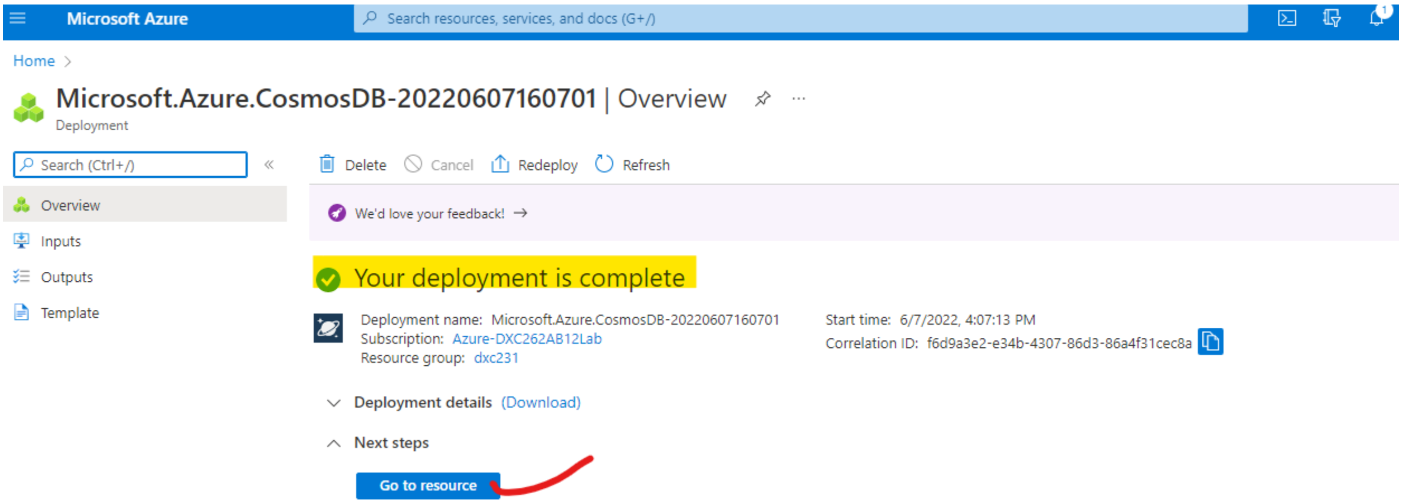
Deployment name: Microsoft.Azure.CosmosDB-20220607160701
Subscription: Azure-DXC262AB12Lab
Resource group: dxc231

Start time: 6/7/2022, 4:07:13 PM
Correlation ID: f6d9a3e2-e34b-4307-86d3-86a4f31cec8a

Deployment details (Download)

Resource	Type	Status	Operation details
No results.			

Step-8: once The deployment is completed you can manage the COMOS DB and Click on GO to resources



Microsoft.Azure.CosmosDB-20220607160701 | Overview

Deployment

Your deployment is complete

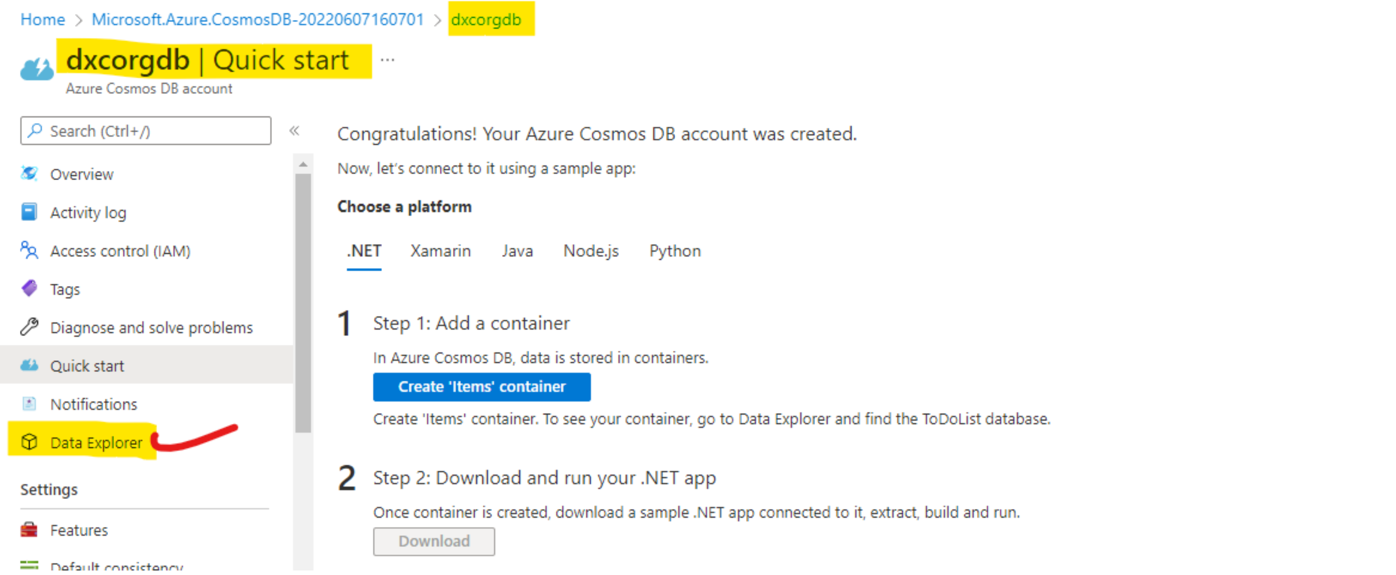
Deployment name: Microsoft.Azure.CosmosDB-20220607160701
Subscription: Azure-DXC262AB12Lab
Resource group: dxc231

Start time: 6/7/2022, 4:07:13 PM
Correlation ID: f6d9a3e2-e34b-4307-86d3-86a4f31cec8a

Deployment details (Download)

Next steps

Step-9: After Navigating to the Go to resource we find the data explorer as shown in figure below . click on that



dxcorgdb | Quick start

Azure Cosmos DB account

Congratulations! Your Azure Cosmos DB account was created.

Now, let's connect to it using a sample app:

Choose a platform

1 Step 1: Add a container

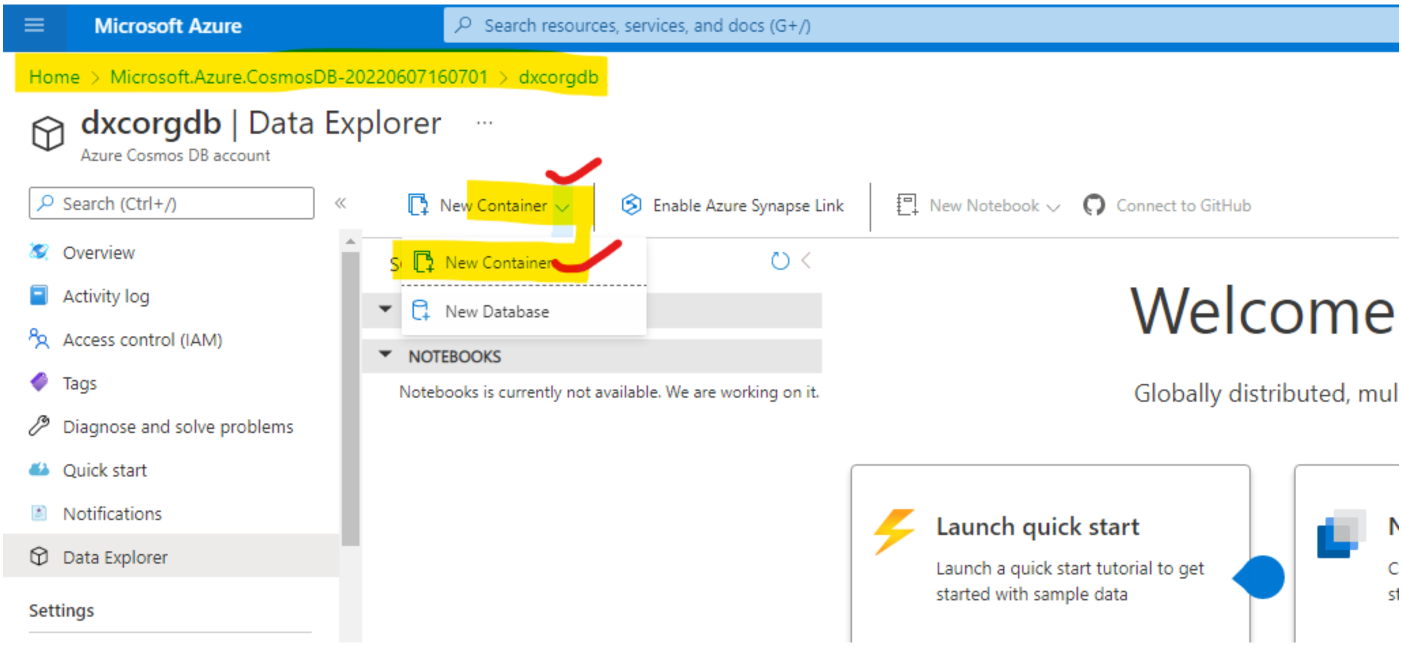
In Azure Cosmos DB, data is stored in containers.

Create 'Items' container. To see your container, go to Data Explorer and find the ToDoList database.

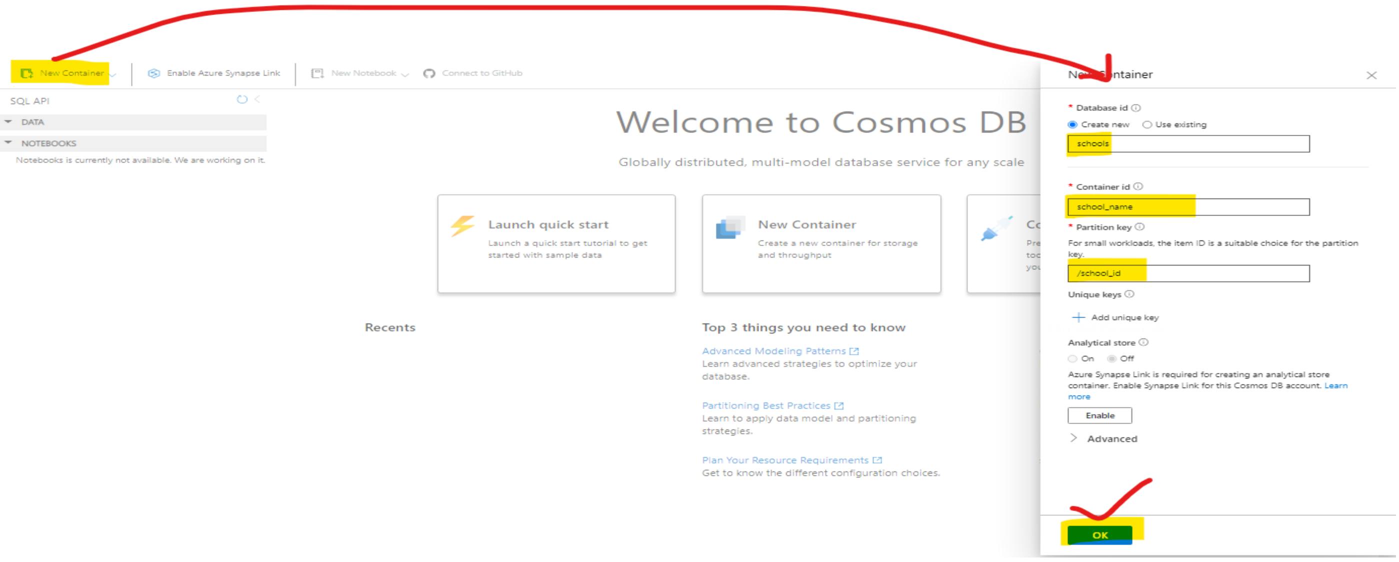
2 Step 2: Download and run your .NET app

Once container is created, download a sample .NET app connected to it, extract, build and run.

Step-10: click on new container and select new container from the dropdown menu. As shown in figure below.


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.Azure.CosmosDB-20220607160701 > dxcorgdb
dxcorgdb | Data Explorer
Azure Cosmos DB account
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Quick start
Notifications
Data Explorer
Settings
New Container
Enable Azure Synapse Link
New Notebook
Connect to GitHub
New Container
New Database
NOTEBOOKS
Notebooks is currently not available. We are working on it.
Welcome
Globally distributed, mul
Launch quick start
Launch a quick start tutorial to get started with sample data

Step-11: after that name the data base, container_id, primary key as shown in screen shot attached below.


New Container

* Database id

● Create new ○ Use existing

schools

* Container id

school_name

* Partition key

For small workloads, the item ID is a suitable choice for the partition key.

/school_id

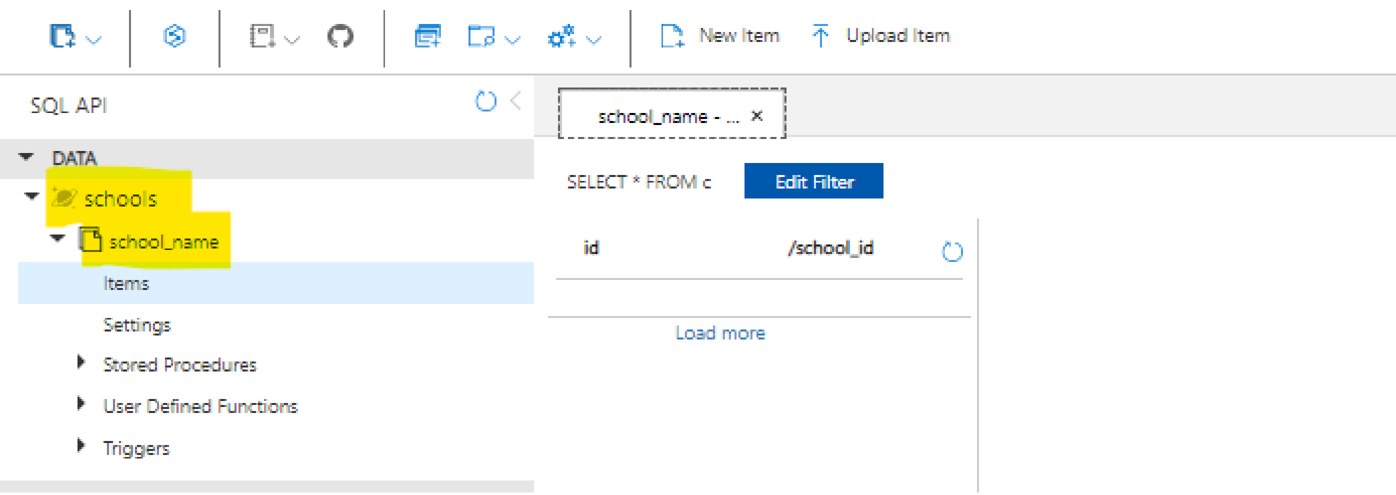
Analytical store

○ On ● Off

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. Learn more

New Container
Enable Azure Synapse Link
New Notebook
Connect to GitHub
SQL API
DATA
NOTEBOOKS
Notebooks is currently not available. We are working on it.
Welcome to Cosmos DB
Globally distributed, multi-model database service for any scale
Launch quick start
Launch a quick start tutorial to get started with sample data
New Container
Create a new container for storage and throughput
Recents
Top 3 things you need to know
Advanced Modeling Patterns
Learn advanced strategies to optimize your database.
Partitioning Best Practices
Learn to apply data model and partitioning strategies.
Plan Your Resource Requirements
Get to know the different configuration choices.
Unique keys
Add unique key
Enable
Advanced
OK

Step12: The cosmos Db setup has been setup successfully and the database is created as per the data given above as shown in fig 8.12



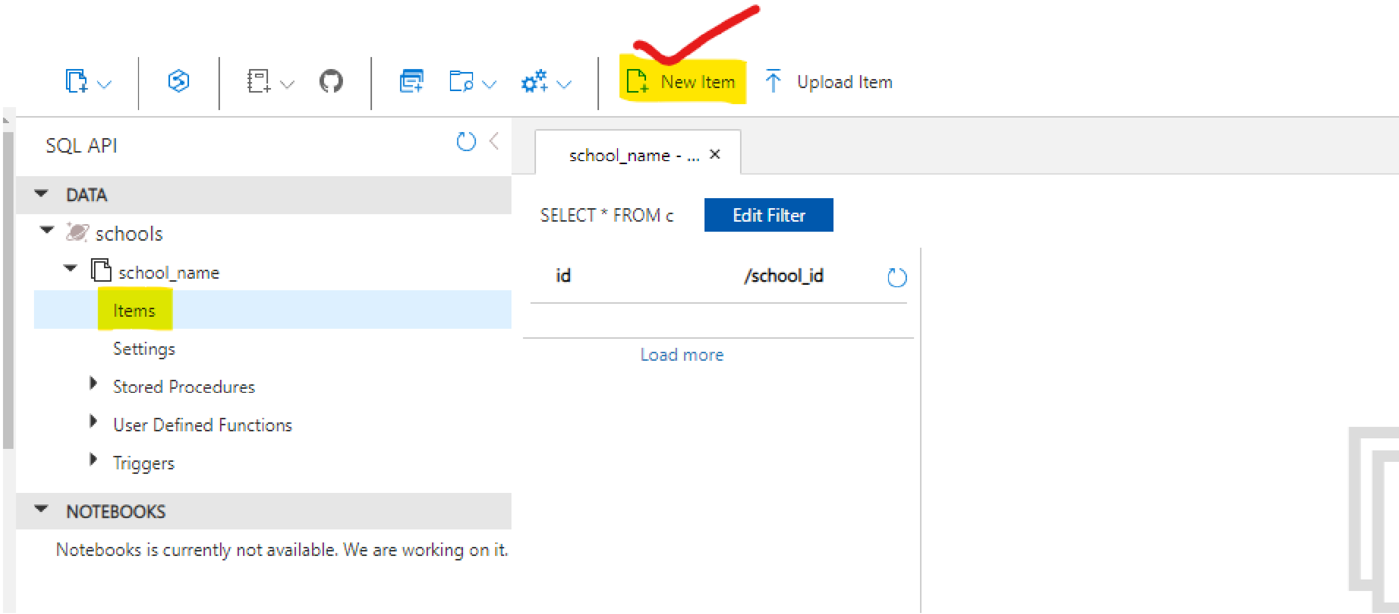
9. Explain with screenshots & step how to insert data into Cosmos DB?

Ans) The Cosmos DB creation is done as mentioned in the above. And the data can be inserted into the cosmos db in two ways

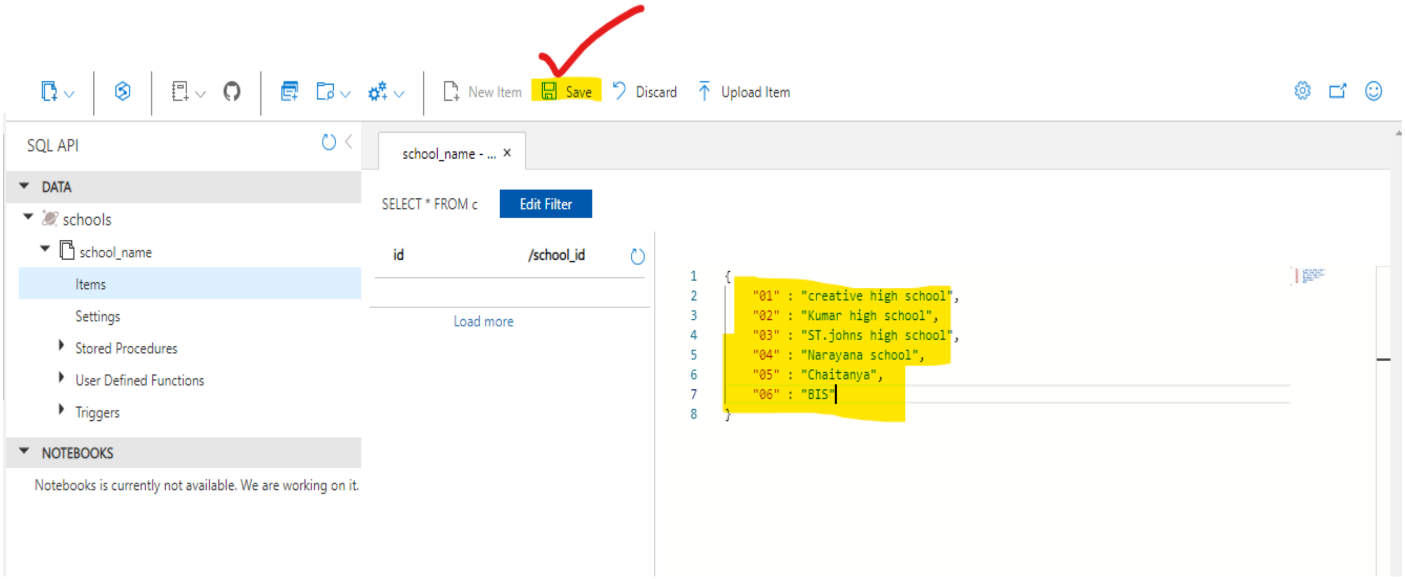
1. We can enter manually by key-value pairs
2. We can upload the json/csv file as well.

Steps to be followed to insert the data in case-1

Step-1 : navigate to the resource page and click on items and you can see add items at the ribbon as shown below



Step-2: after entering the data in the form of key value pairs as mentioned in the below format click on save button. As shown below

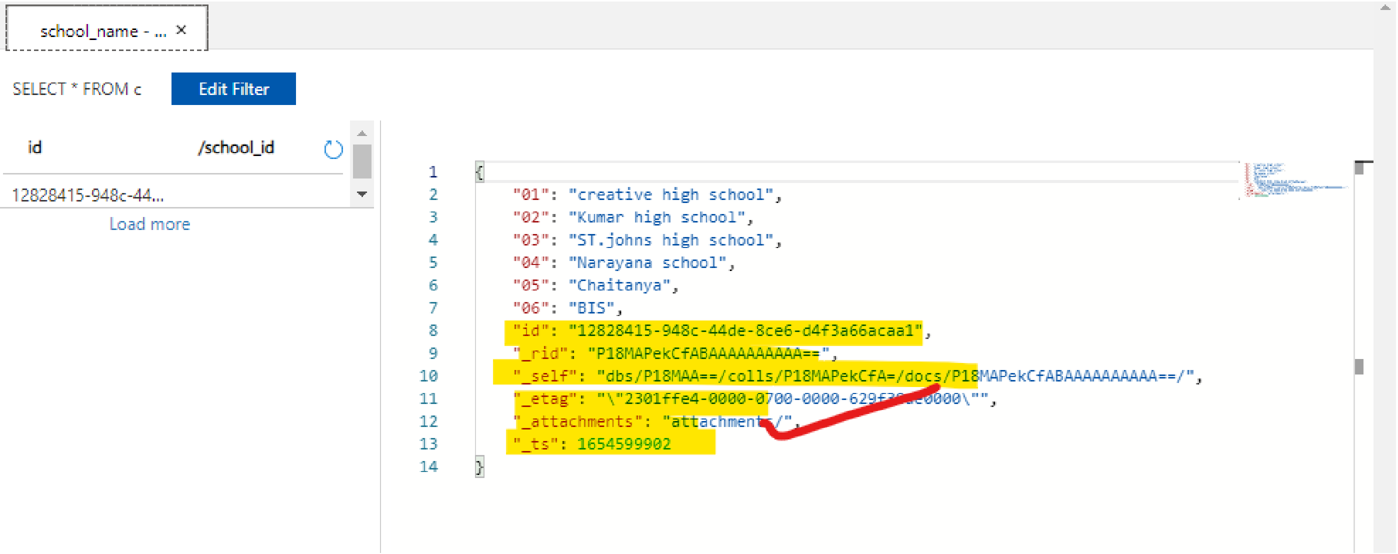


Notebooks is currently not available. We are working on it.

SELECT * FROM c

```
{
    "01" : "creative high school",
    "02" : "Kumar high school",
    "03" : "ST.johns high school",
    "04" : "Narayana school",
    "05" : "Chaitanya",
    "06" : "BIS"
}
```

Step-3: after clicking the save button the system will generates the unique ID for the data that you have inserted. Please follow the attachment to understand clearly.



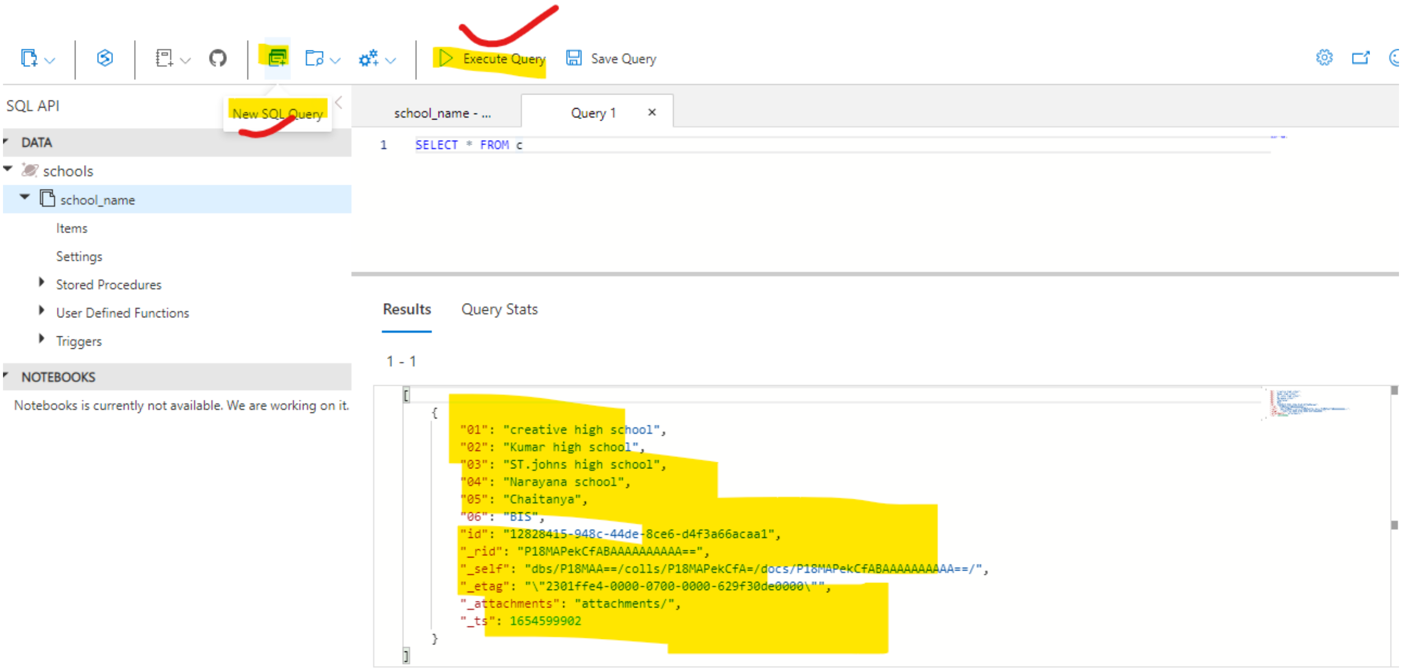
SELECT * FROM c

12828415-948c-44...

```
{
    "01": "creative high school",
    "02": "Kumar high school",
    "03": "ST.johns high school",
    "04": "Narayana school",
    "05": "Chaitanya",
    "06": "BIS",
    "id": "12828415-948c-44de-8ce6-d4f3a66acaa1",
    "_rid": "P18MAPekCfABAAAAAAAAAA==",
    "_self": "dbs/P18MAA==/colls/P18MAPekCfA=/docs/P18MAPekCfABAAAAAAAAAA==/",
    "_etag": "\"2301ffe4-0000-0700-0000-629f3[illegible]0000\"",
    "_attachments": "attachments/",
    "_ts": 1654599902
}
```

In this way the data can be inserted into the database of cosmos.

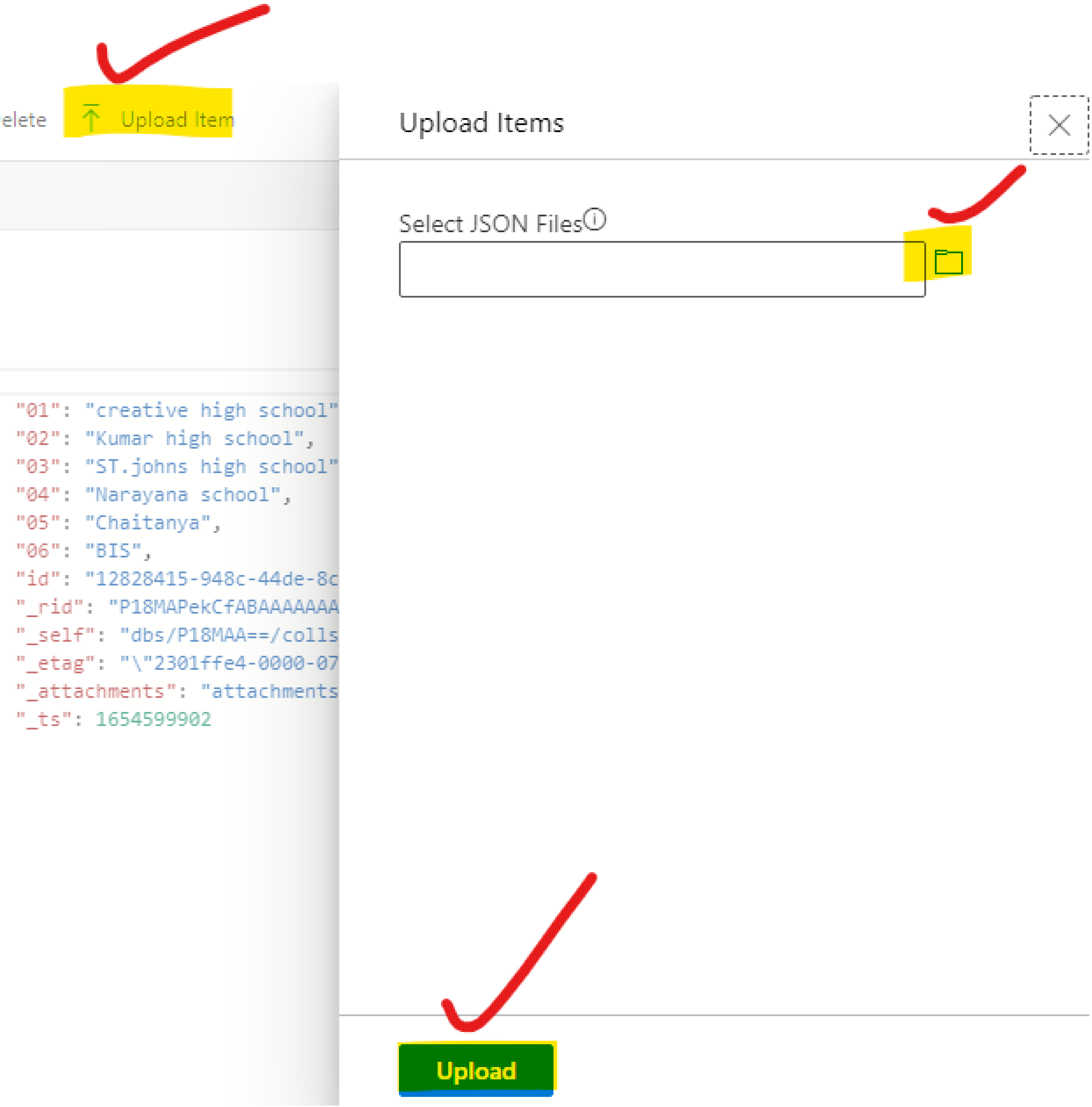
To know the data that you have inserted use the new SQl query option from the ribbon.



```
SELECT * FROM c
```

```
[
    {
        "01": "creative high school",
        "02": "Kumar high school",
        "03": "ST.johns high school",
        "04": "Narayana school",
        "05": "Chaitanya",
        "06": "BIS",
        "id": "12828415-948c-44de-8ce6-d4f3a66acaa1",
        "_rid": "P18MAPekCfABAAAAAAAAAA==",
        "_self": "dbs/P18MAA==/colls/P18MAPekCfA=/docs/P18MAPekCfABAAAAAAAAAA==/",
        "_etag": "\"2301ffe4-0000-0700-0000-629f30de0000\"",
        "_attachments": "attachments/",
        "_ts": 1654599902
    }
]
```

In another way we can upload the JSON file to the cosmos db and we can get the data.

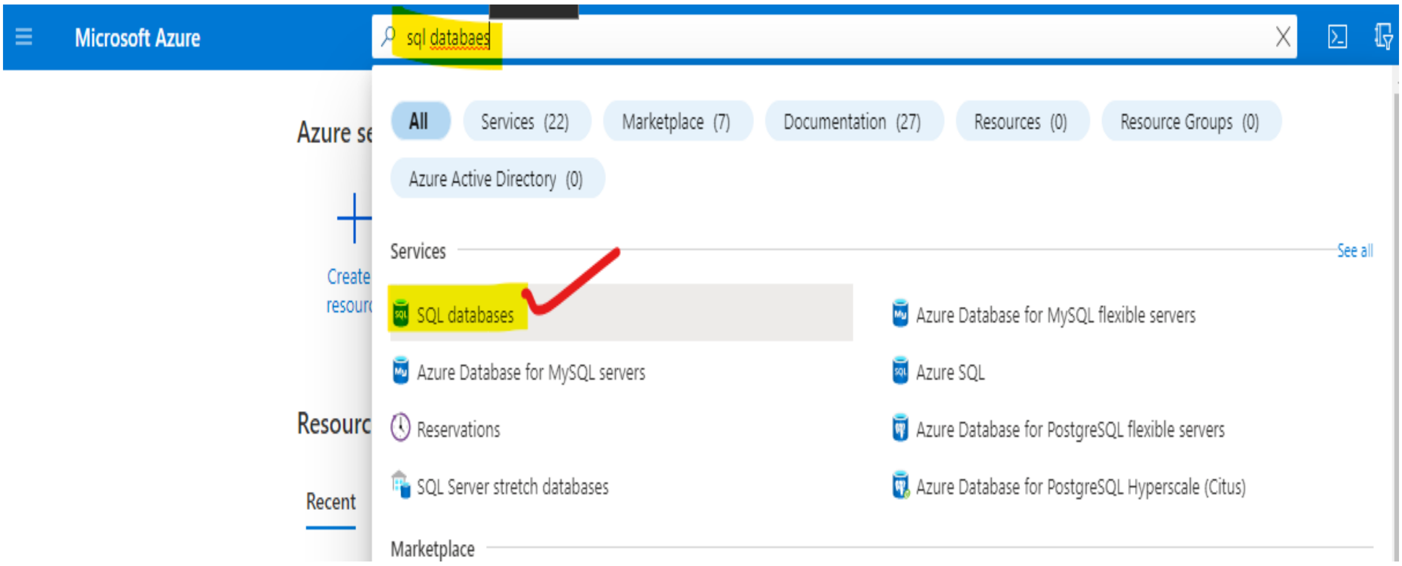


```
"01": "creative high school"
"02": "Kumar high school",
"03": "ST.johns high school"
"04": "Narayana school",
"05": "Chaitanya",
"06": "BIS",
"id": "12828415-948c-44de-8c
"_rid": "P18MAPekCfABAAAAAAA
"_self": "dbs/P18MAA==/colls
"_etag": "\"2301ffe4-0000-07
"_attachments": "attachments
"_ts": 1654599902
```

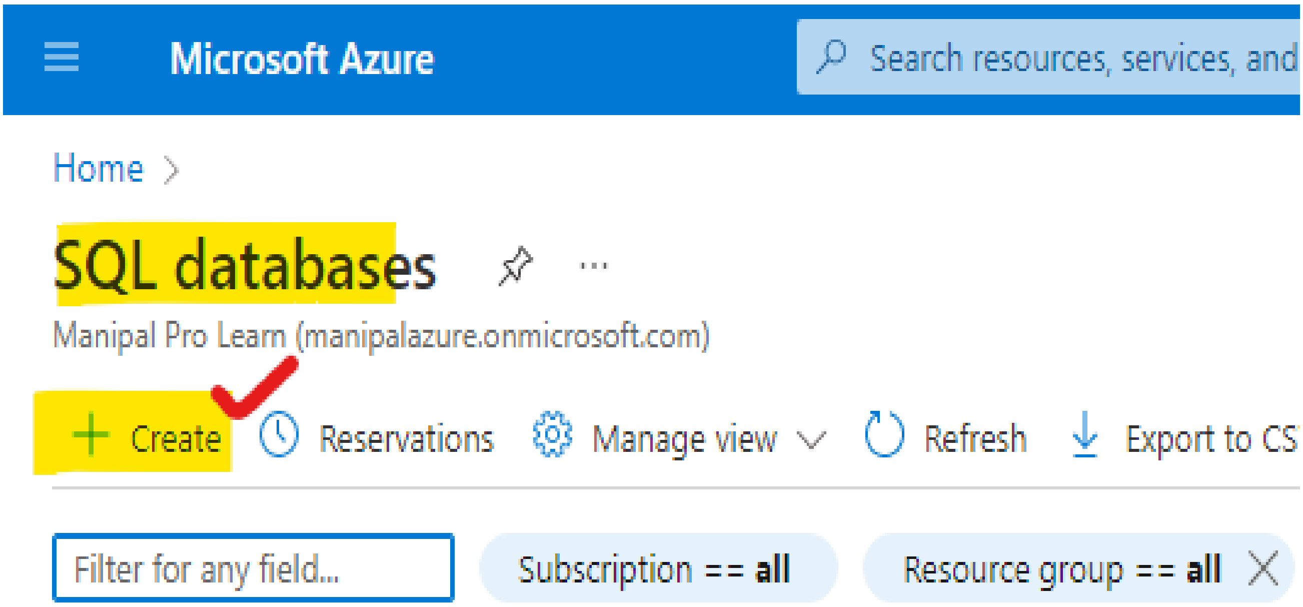
10. Explain with screenshots & step how to create Azure SQL Db & also explain how to insert data into Azure SQL DB?

Ans) To create AZURE sql DB. We have to login to the azure account

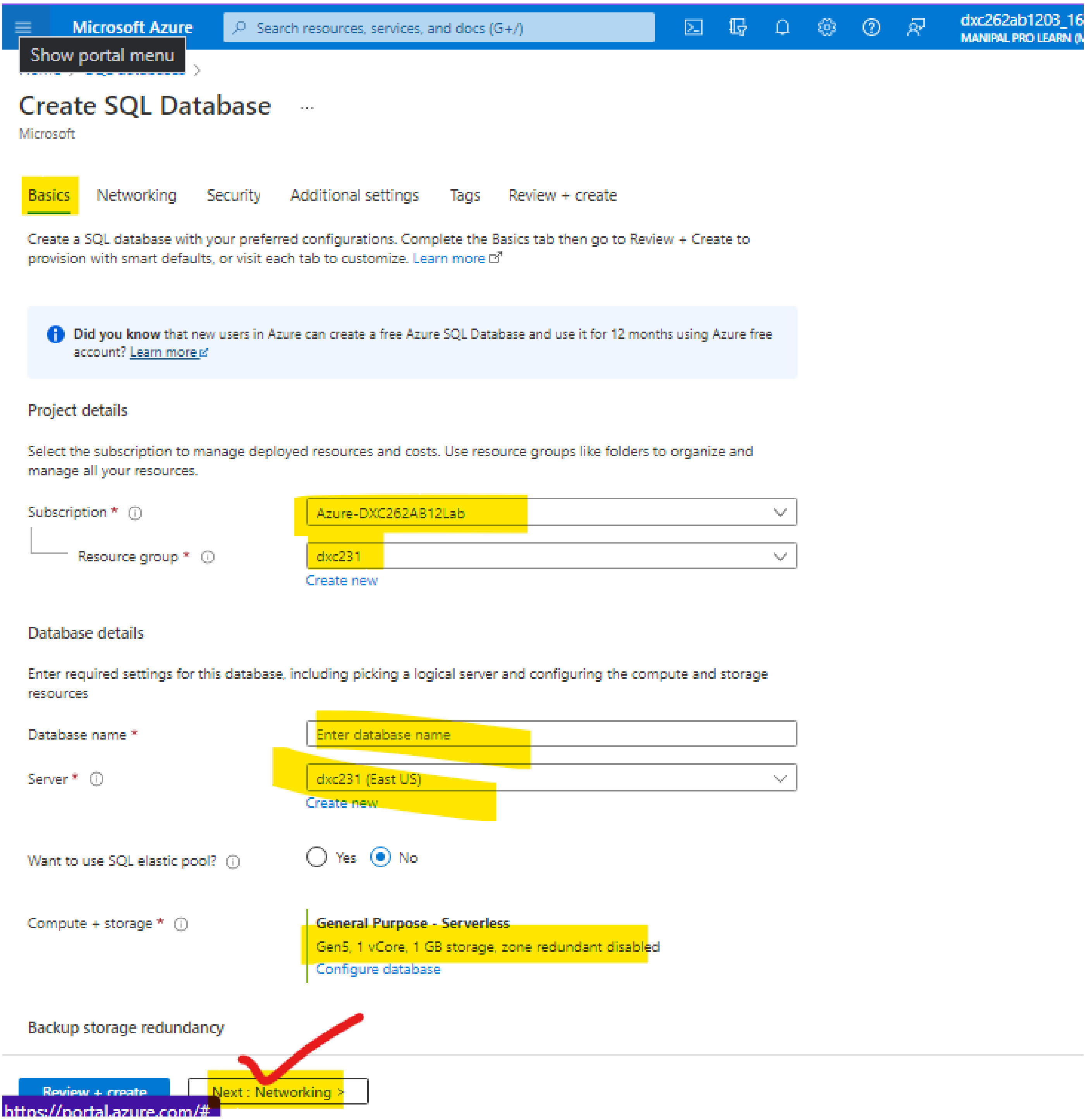
Step-1: search for Azure sql databases and select the AZURE SQl databases from the search results


Microsoft Azure
sql databaes
All
Services (22)
Marketplace (7)
Documentation (27)
Resources (0)
Resource Groups (0)
Azure Active Directory (0)
Azure se
Create
resour
Services
See all
SQL databases
Azure Database for MySQL flexible servers
Azure Database for MySQL servers
Azure SQL
Resourc
Reservations
Azure Database for PostgreSQL flexible servers
SQL Server stretch databases
Azure Database for PostgreSQL Hyperscale (Citus)
Recent
Marketplace

Step-2 : after selecting the Sql databases. Click on create button as shown below



Step-3: select all options as mentioned below



Create SQL Database ...

Microsoft

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Did you know that new users in Azure can create a free Azure SQL Database and use it for 12 months using Azure free account? Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure-DXC262AB12Lab

Resource group * dxc231

Create new

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * Enter database name

Server * dxc231 (East US)

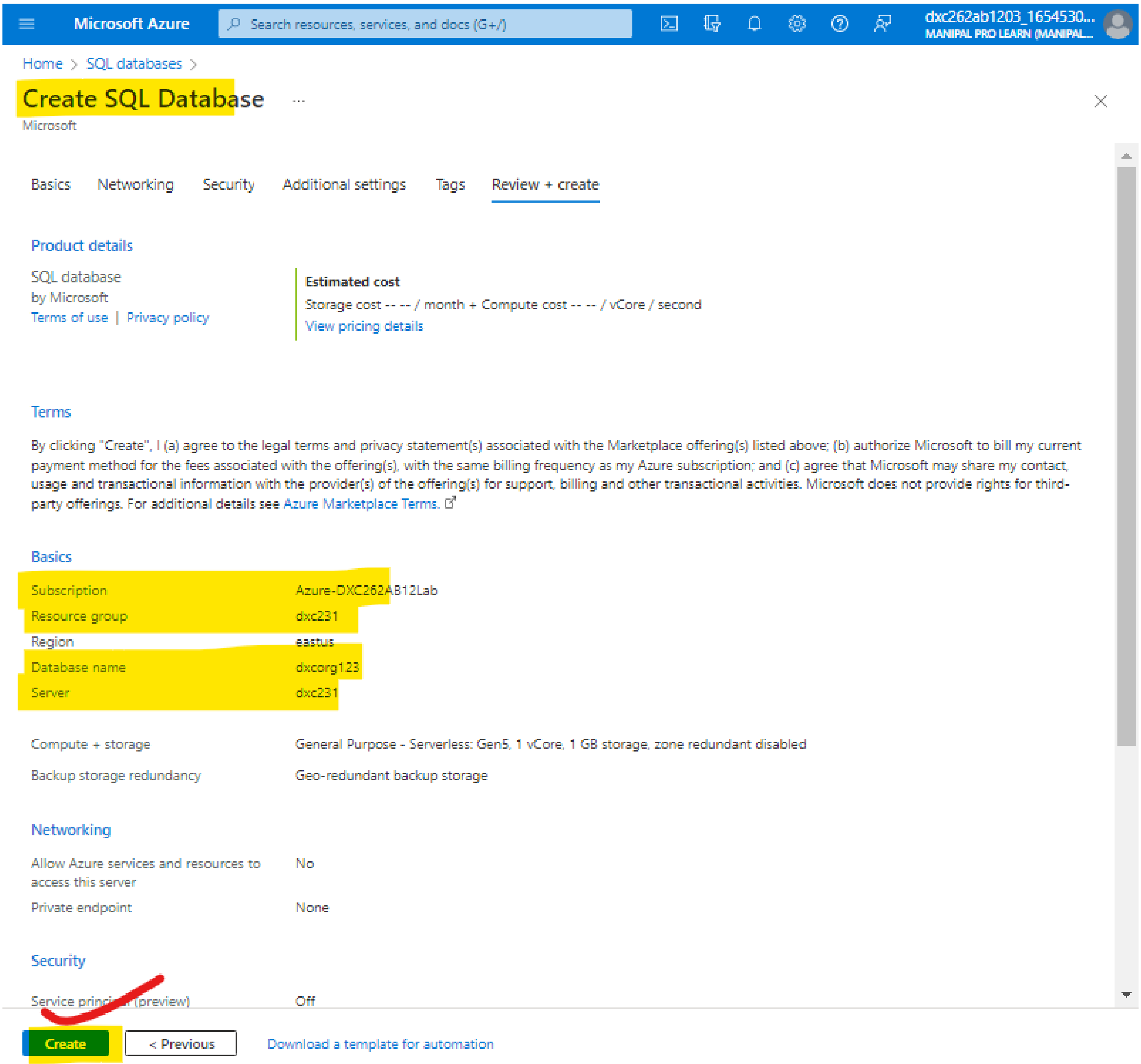
Create new

Want to use SQL elastic pool? ○ Yes ● No

Compute + storage * **General Purpose - Serverless**
Gen5, 1 vCore, 1 GB storage, zone redundant disabled
Configure database

Backup storage redundancy

Step-4: after completing this step navigate to the next menus with out changing any settings and click on create



Create SQL Database ...

Microsoft

Product details

SQL database
by Microsoft
Terms of use | Privacy policy

Estimated cost
Storage cost -- -- / month + Compute cost -- -- / vCore / second
View pricing details

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. For additional details see Azure Marketplace Terms.

Basics

Subscription	Azure-DXC262AB12Lab
Resource group	dxc231
Region	eastus
Database name	dxcorg123
Server	dxc231
Compute + storage	General Purpose - Serverless: Gen5, 1 vCore, 1 GB storage, zone redundant disabled
Backup storage redundancy	Geo-redundant backup storage

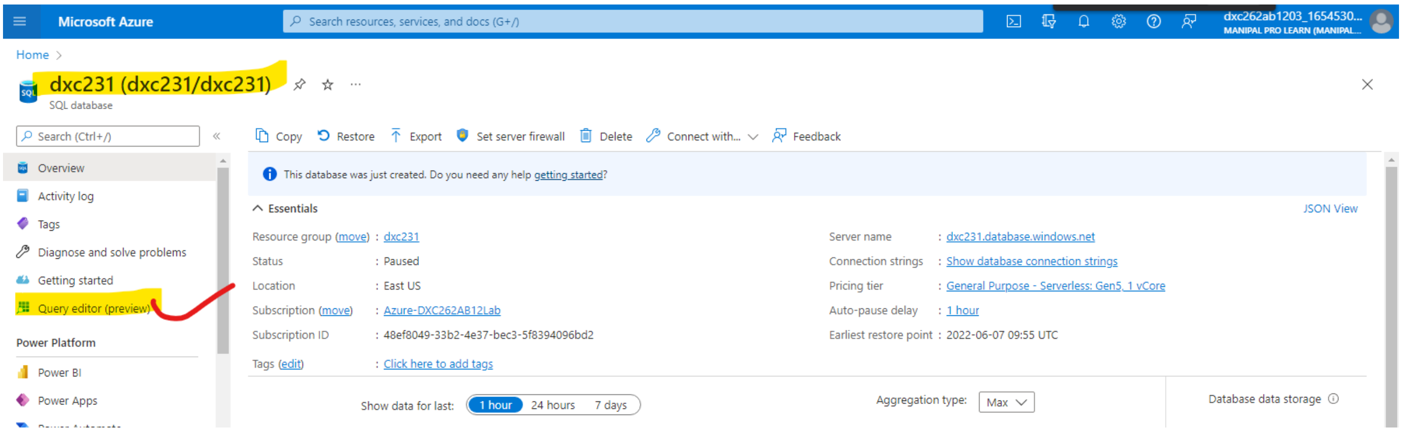
Networking

Allow Azure services and resources to access this server	No
Private endpoint	None

Security

Service principal (preview)	Off

Step-5: after clicking on create it takes some time to deploy after that we can access the Database and click on query editor



dxc231 (dxc231/dxc231)

SQL database

This database was just created. Do you need any help getting started?

Essentials

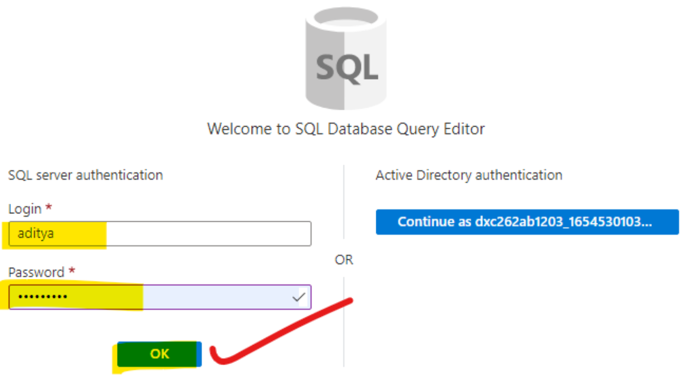
Resource group (move)	: dxc231
Status	: Paused
Location	: East US
Subscription (move)	: Azure-DXC262AB12Lab
Subscription ID	: 48ef8049-33b2-4e37-bec3-5f8394096bd2
Tags (edit)	: Click here to add tags
Server name	: dxc231.database.windows.net
Connection strings	: Show database connection strings
Pricing tier	: General Purpose - Serverless: Gen5, 1 vCore
Auto-pause delay	: 1 hour
Earliest restore point	: 2022-06-07 09:55 UTC

Show data for last: 1 hour 24 hours 7 days

Aggregation type: Max

Database data storage

Step-6: login with your login credentials



Welcome to SQL Database Query Editor

SQL server authentication

Login *

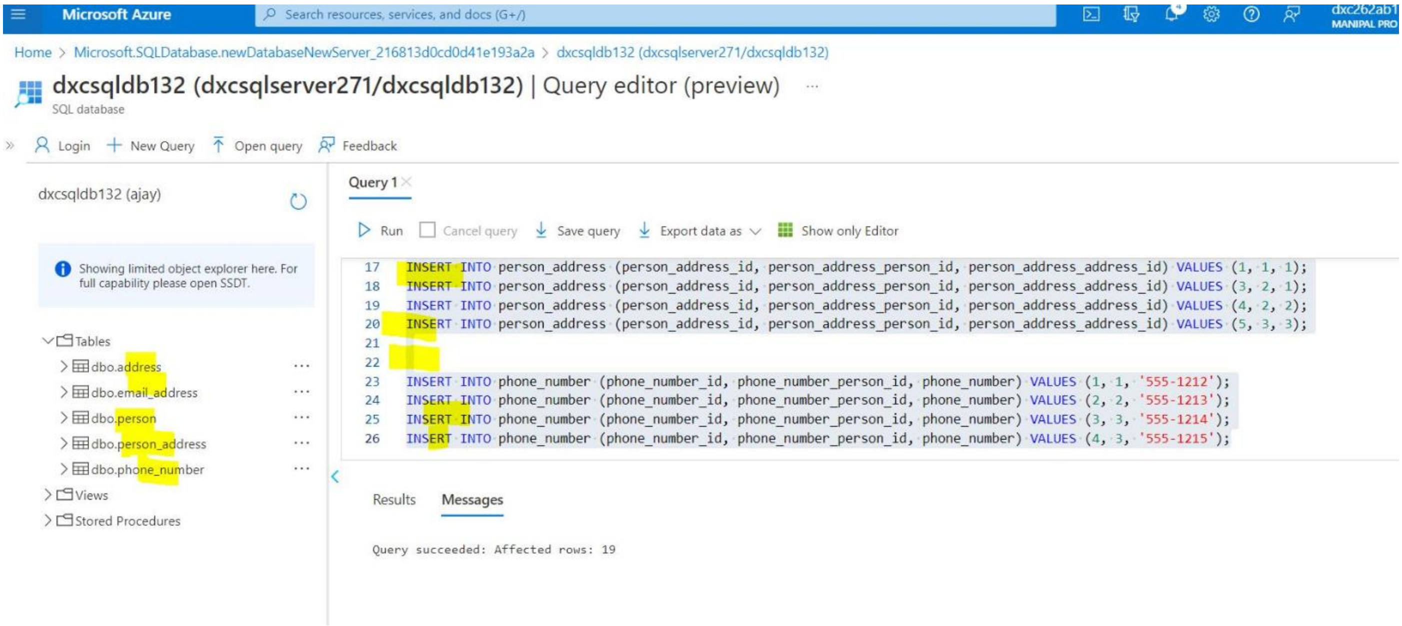
aditya

Password *

•••••••••

Active Directory authentication

Step-7: insert the data by using the traditional SQL methods.



Microsoft Azure
Search resources, services, and docs (G+/)
dxc262ab1
MANIPAL PRO
Home > Microsoft.SQLDatabase.newDatabaseNewServer_216813d0cd0d41e193a2a > dxcsqldb132 (dxcsqlserver271/dxcsqldb132)

dxcsqldb132 (dxcsqlserver271/dxcsqldb132) | Query editor (preview)

SQL database

Showing limited object explorer here. For full capability please open SSDT.

```sql
INSERT INTO person_address (person_address_id, person_address_person_id, person_address_address_id) VALUES (1, 1, 1);
INSERT INTO person_address (person_address_id, person_address_person_id, person_address_address_id) VALUES (3, 2, 1);
INSERT INTO person_address (person_address_id, person_address_person_id, person_address_address_id) VALUES (4, 2, 2);
INSERT INTO person_address (person_address_id, person_address_person_id, person_address_address_id) VALUES (5, 3, 3);


INSERT INTO phone_number (phone_number_id, phone_number_person_id, phone_number) VALUES (1, 1, '555-1212');
INSERT INTO phone_number (phone_number_id, phone_number_person_id, phone_number) VALUES (2, 2, '555-1213');
INSERT INTO phone_number (phone_number_id, phone_number_person_id, phone_number) VALUES (3, 3, '555-1214');
INSERT INTO phone_number (phone_number_id, phone_number_person_id, phone_number) VALUES (4, 3, '555-1215');
```

```
Query succeeded: Affected rows: 19
```