ASSIGNMENT-7

NAME-KONAKALLA AKHILA ROLL NUMBER-DXC262AB1225

BATCH-DXC-262-ANALYTICS-B12-AZURE

COMPANY-DXC TECHNOLOGY

EMPLOYEE DOMAIN-AZURE ANALYTICS

TRAINING UNDER-MANIPAL PRO LEARN

TRAINER NAME-MR.AJAY KUMAR

DATE OF SUBMISSION -7th JUNE 2022

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

**Components of Spark**

1. Shark (SQL)
2. Spark Streaming (Streaming)
3. ML Lib (Machine Learning)
4. Graph X (Graph Computation)
5. Spark R (R on Spark)

**Functionality of components:**

Shark (SQL):

Shark is one of the Spark Ecosystem components. It is used to perform structured data analysis, especially if the data is too voluminous. Shark also allows running unmodified Hive queries on existing Hadoop deployment.

Spark Streaming (Streaming):

Spark Streaming is one of those unique features, which have empowered Spark to potentially take the role of Apache Storm. Spark Streaming mainly enables you to create analytical and interactive applications for live streaming data. You can do the streaming of the data and then, Spark can run its operations from the streamed data itself.

ML Lib (Machine Learning):

ML Lib is a machine learning library like Mahout. It is built on top of Spark, and has the provision to support many machine learning algorithms. But the point difference with Mahout is that it runs almost 100 times faster than Map-Reduce.

Graph X (Graph Computation):

The graphs and graphical computations, Spark has its own Graph Computation Engine, called Graph X. It is similar to other widely used graph processing tools or databases.

Spark R (R on Spark):

There are many people from data science track, who must be aware that for statistical analysis, R is among the best. There is already an integration of R with Hadoop. Now, Spark R is a package for R language to enable R users to leverage the power of Spark from R shell.

|  |
| --- |
| Shark(SQL) |
| Spark streaming(Streaming) |
| ML Lib(Machine learning) |
| Graph X (Graph computation) |
| Spark R(R on spark) |
| Components of spark | |

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

Spark Core is the underlying general execution engine for the Spark platform that all other functionality is built on top of. It provides in-memory computing capabilities to deliver speed, a generalized execution model to support a wide variety of applications, and Java, Scala, and Python APIs for ease of development. Spark Core is also home to the API that defines resilient distributed datasets (RDDs), which are Spark's main programming abstraction.

RDD was the primary user-facing API in Spark since its inception. At the core, an RDD is an immutable distributed collection of elements of your data, partitioned across nodes in your cluster that can be operated in parallel with a low-level API that offers transformations and actions.

RDD is related to spark by following reasons:

1. You want low-level transformation and actions and control on your dataset;
2. Your data is unstructured, such as media streams or streams of text;
3. You want to manipulate your data with functional programming constructs than domain specific expressions;
4. You don’t care about imposing a schema, such as columnar format while processing or accessing data attributes by name or column; and
5. You can forgo some optimization and performance benefits available with Data Frames and Datasets for structured and semi-structured data.

Program for pyspark data frame from RDD:

rdd = spark.sparkContext.parallelize([

(1,2.,’string1’,date(2022,6,7),datetime(2022,6,7,4,20)),

(4,5.,’string1’,date(2022,6,8),datetime(2022,6,8,4,22)),

(6,7.,’string1’,date(2022,6,9),datetime(2022,6,9,4,23)),

])

df =spark.createDataFrame(rdd,schema[‘a’,’b’,’c’,’d’,’e’])

df

**3.Explain various Mlib algorithms spark is supporting ?**

Spark MLlib is used to perform machine learning in Apache Spark. MLlib consists popular algorithms and utilities

Spark MLlib provides the following tools:

* ML Algorithms: ML Algorithms form the core of MLlib. These include common learning algorithms such as classification, regression, clustering and collaborative filtering.
* Featurization: Featurization includes feature extraction, transformation, dimensionality reduction and selection.
* Pipelines: Pipelines provide tools for constructing, evaluating and tuning ML Pipelines.
* Persistence: Persistence helps in saving and loading algorithms, models and Pipelines.
* Utilities: Utilities for linear algebra, statistics and data handling.

**ML lib Algorithms**

The popular algorithms and utilities in Spark MLlib are:

1. Basic Statistics
2. Regression
3. Classification
4. Recommendation System
5. Clustering
6. Dimensionality Reduction
7. Feature Extraction
8. Optimization

**4.Explain the benfits spark SQL & how relational data will be inserted into spark ?**

**1.Integrated**  
Apache Spark SQL mixes SQL queries with Spark programs. With the help of Spark SQL, we can query structured data as a distributed dataset (RDD). We can run SQL queries alongside complex analytic algorithms using tight integration property of Spark SQL.  
**2. Unified Data Access**  
Using Spark SQL, we can load and query data from different sources. The Schema-RDDs lets single interface to productively work structured data. For example, Apache Hive tables, parquet files, and JSON files.  
**3. High compatibility**  
In Apache Spark SQL, we can run unmodified Hive queries on existing warehouses. It allows full compatibility with existing Hive data, queries and UDFs, by using the Hive fronted and MetaStore.  
**4. Standard Connectivity**  
It can connect through JDBC or ODBC. It includes server mode with industry standard JDBC and ODBC connectivity.  
**5. Scalability**  
To support mid-query fault tolerance and large jobs, it takes advantage of RDD model. It uses the same engine for interactive and long queries.  
**6. Performance Optimization**  
The query optimization engine in Spark SQL converts each SQL query to a logical plan. Further, it converts to many physical execution plans. Among the entire plan, it selects the most optimal physical plan for execution. Read more about[Apache Spark performance tuning techniques](https://data-flair.training/blogs/apache-spark-performance-tuning/) in detail.  
**7. For batch processing of Hive tables**  
We can make use of Spark SQL for fast batch processing of Hive tables.

**5.Explain spark streaming in detail ?**

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**

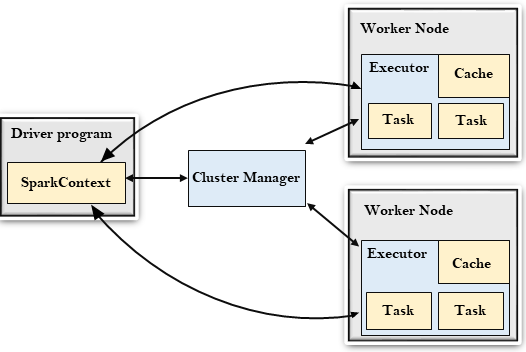
* Fast recovery from failures and stragglers
* Better load balancing and resource usage
* Combining of streaming data with static datasets and interactive queries
* Native integration with advanced processing libraries (SQL, machine learning, graph processing)

**6.Explain spark architecture ?What is master-slave architecture ?**

**spark architecture:**

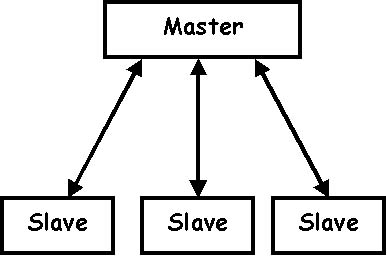
Apache Spark has a well-defined layered architecture where all the spark components and layers are loosely coupled. This architecture is further integrated with various extensions and libraries. Apache Spark Architecture is based on two main abstractions:

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



**Master-Slave architecture :**

Master-slave architectures are used to help stabilize a system. Master is the true data keeper while a slave is a replication of master. Cache/caching is an option but using it as complementary to the master-slave system would be better. Replication is the process of synchronizing data from the master to slave.



**7.Explain various cluster managers in spark ?**

Cluster manager is a platform 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.

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.

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

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

#### **Standalone Cluster Manager**

It is a part of spark distribution and available as a simple cluster manager to us. Standalone cluster manager is resilient in nature, it can handle work failures. It has capabilities to manage resources according to the requirement of applications.

#### **Hadoop Yarn**

This cluster manager works as a distributed computing framework. It also maintains job scheduling as well as resource management. In this cluster, masters and slaves are highly available for us. We are also available with executors and pluggable scheduler.

#### **Apache Mesos**

It is a distributed cluster manager. As like yarn, it is also highly available for master and slaves. It can also manage resource per application. We can run spark jobs, Hadoop MapReduce or any other service applications easily.

**8.Explain with screenshots & steps how to create cosmos DB ?**

**A**zure Cosmos DB is Microsoft's globally distributed multi-model database service.

An Azure subscription or free Azure Cosmos DB trial account

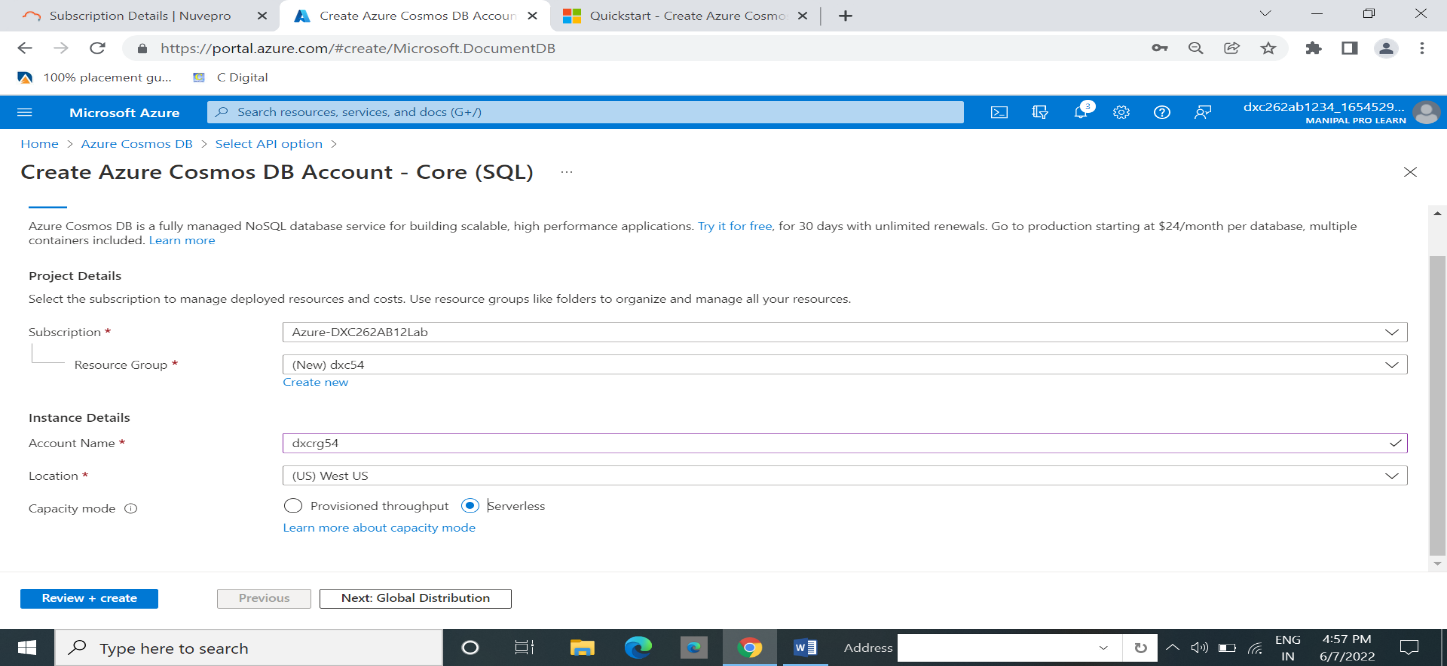
* If you don't have an [Azure subscription](https://docs.microsoft.com/en-us/azure/guides/developer/azure-developer-guide#understanding-accounts-subscriptions-and-billing), create an [Azure free account](https://azure.microsoft.com/free/?ref=microsoft.com&utm_source=microsoft.com&utm_medium=docs&utm_campaign=visualstudio) before you begin.
* You can [try Azure Cosmos DB for free](https://azure.microsoft.com/try/cosmosdb/), without an Azure subscription, and with no commitment required. Alternatively, you can create an [Azure Cosmos DB free tier account](https://docs.microsoft.com/en-us/azure/cosmos-db/optimize-dev-test#azure-cosmos-db-free-tier), with the first 400 RU/s and 5 GB of storage for free.

## Create an Azure Cosmos DB account

1.From the Azure portal menu or the Home page, select Create a resource.

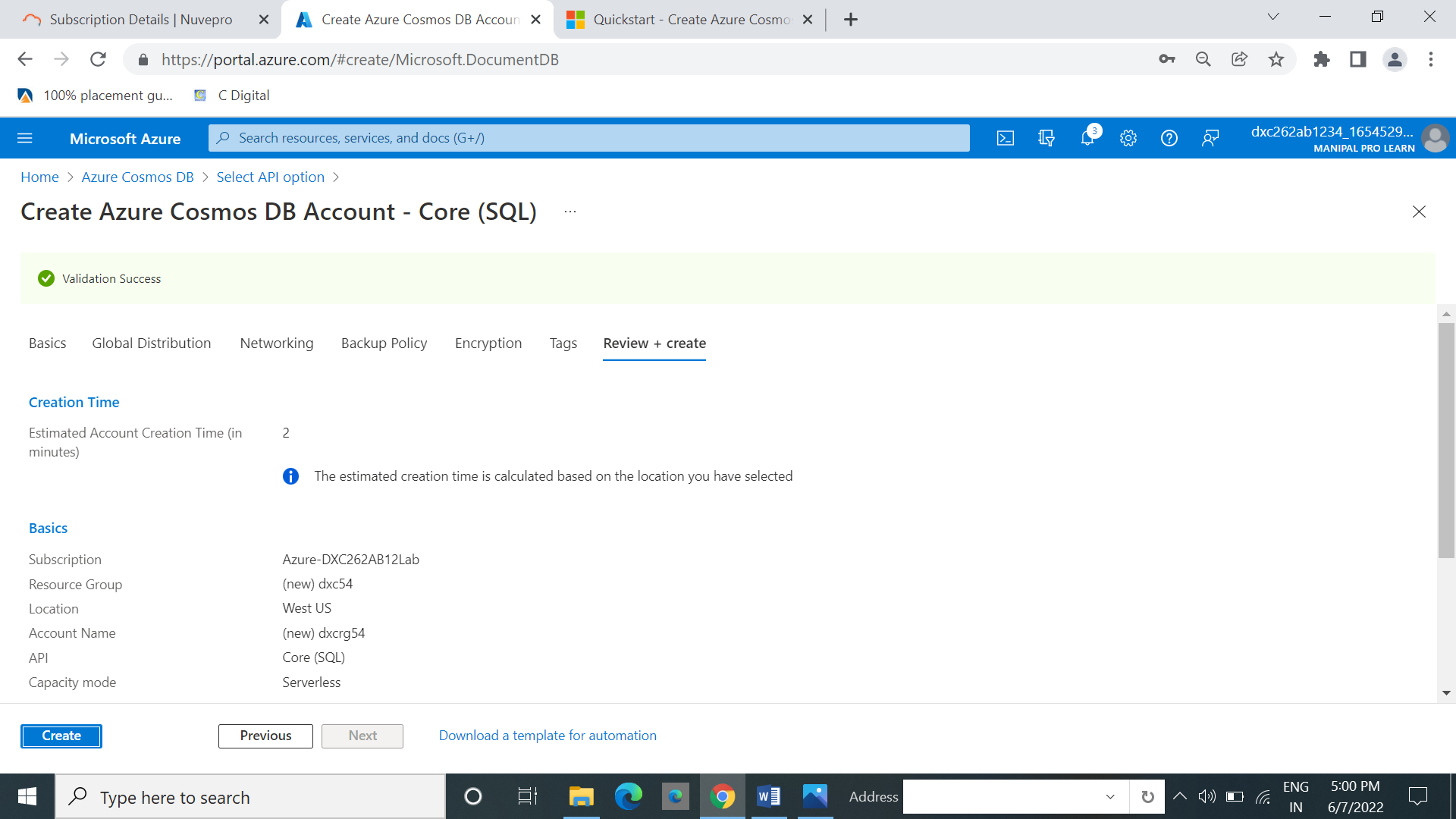
2.On the New page, search for and select Azure Cosmos DB.

3.On the Select API option page, select the Create option within the Core (SQL) - Recommend section. Azure Cosmos DB provides five APIs: Core (SQL)

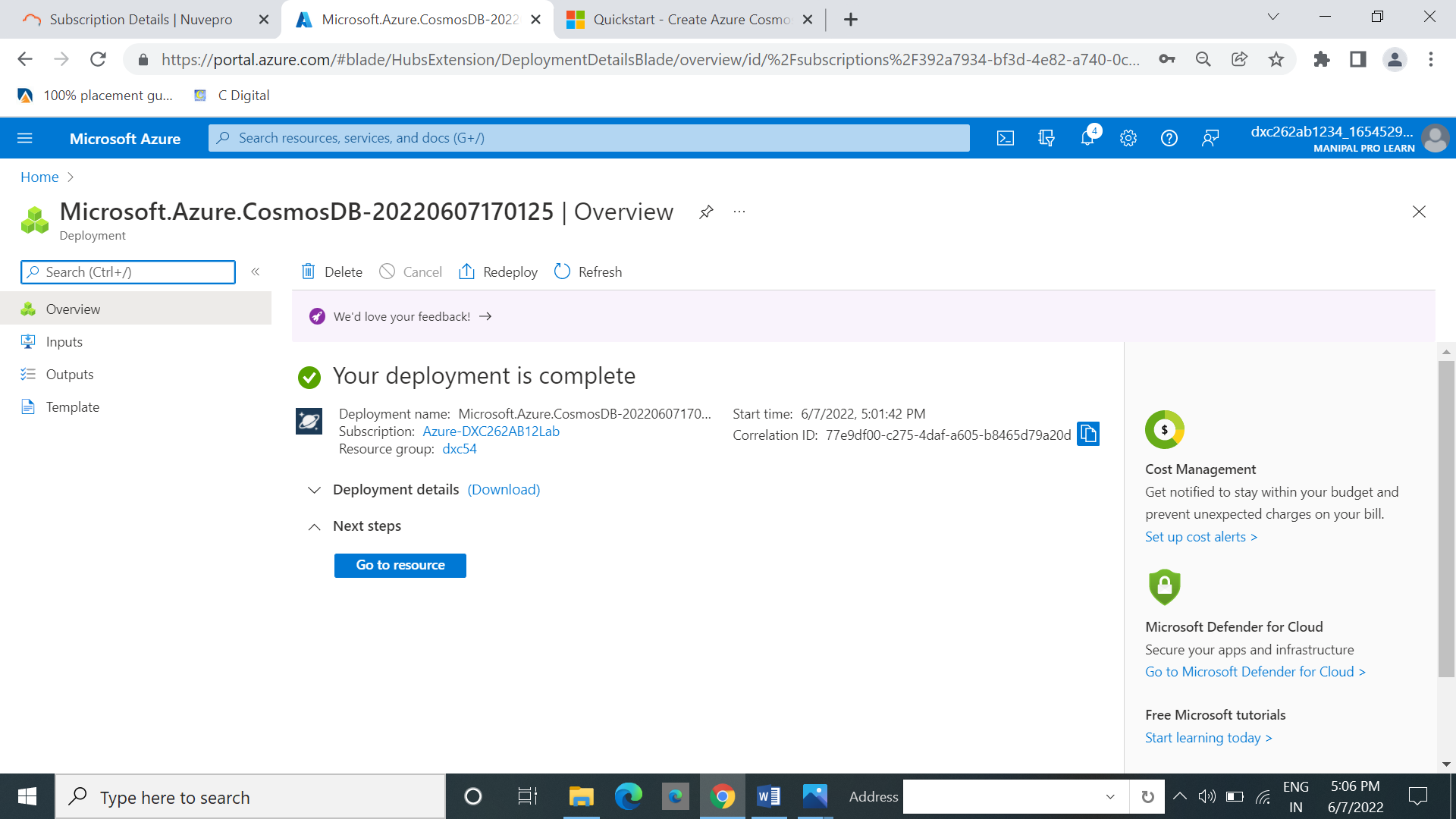
4.In the **Create Azure Cosmos DB Account** page, enter the basic settings for the new Azure Cosmos account. 

5. In the **Global Distribution** tab, configure the following details. You can leave the default values for this quickstart

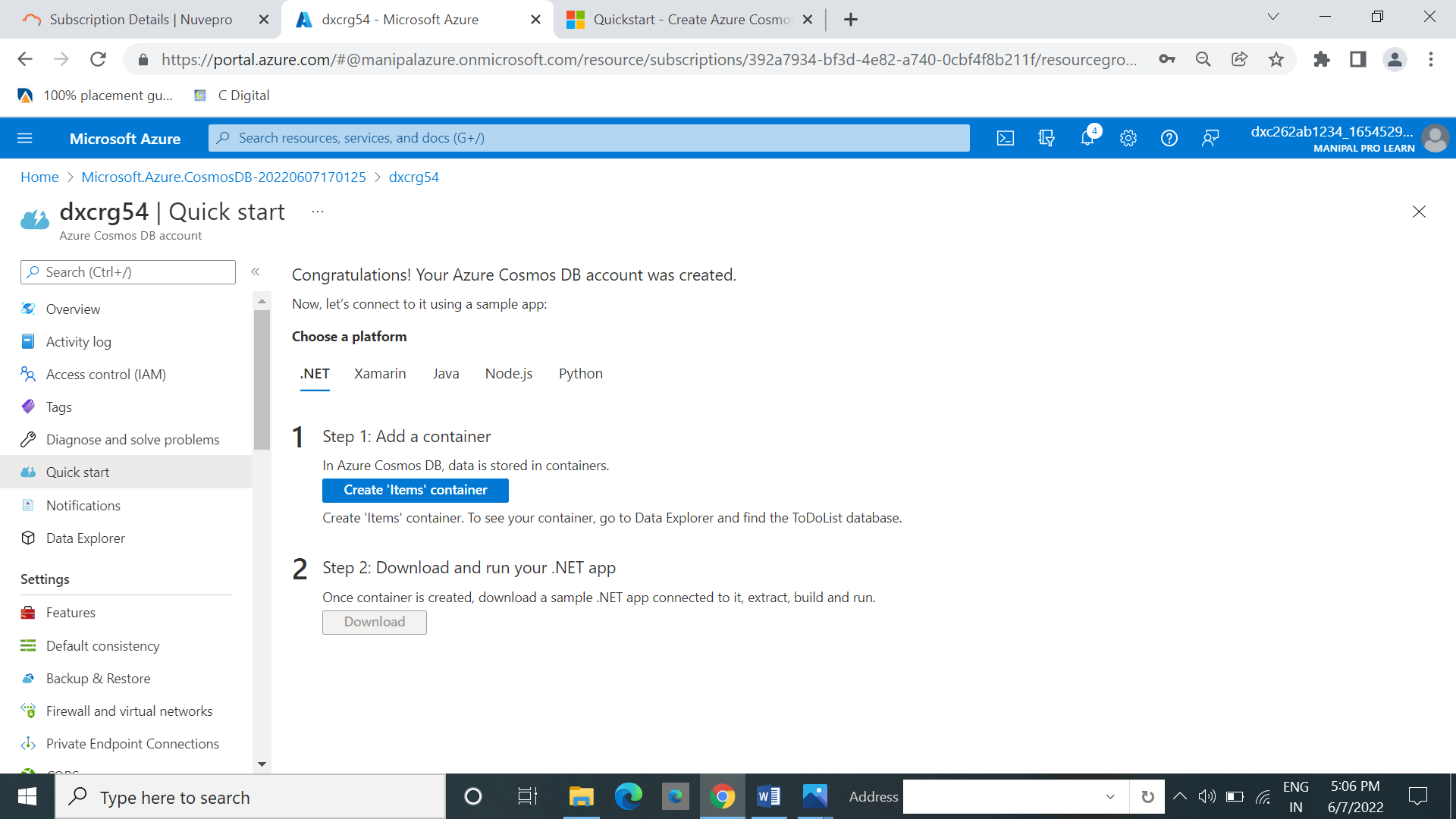
6. Select Review + create.



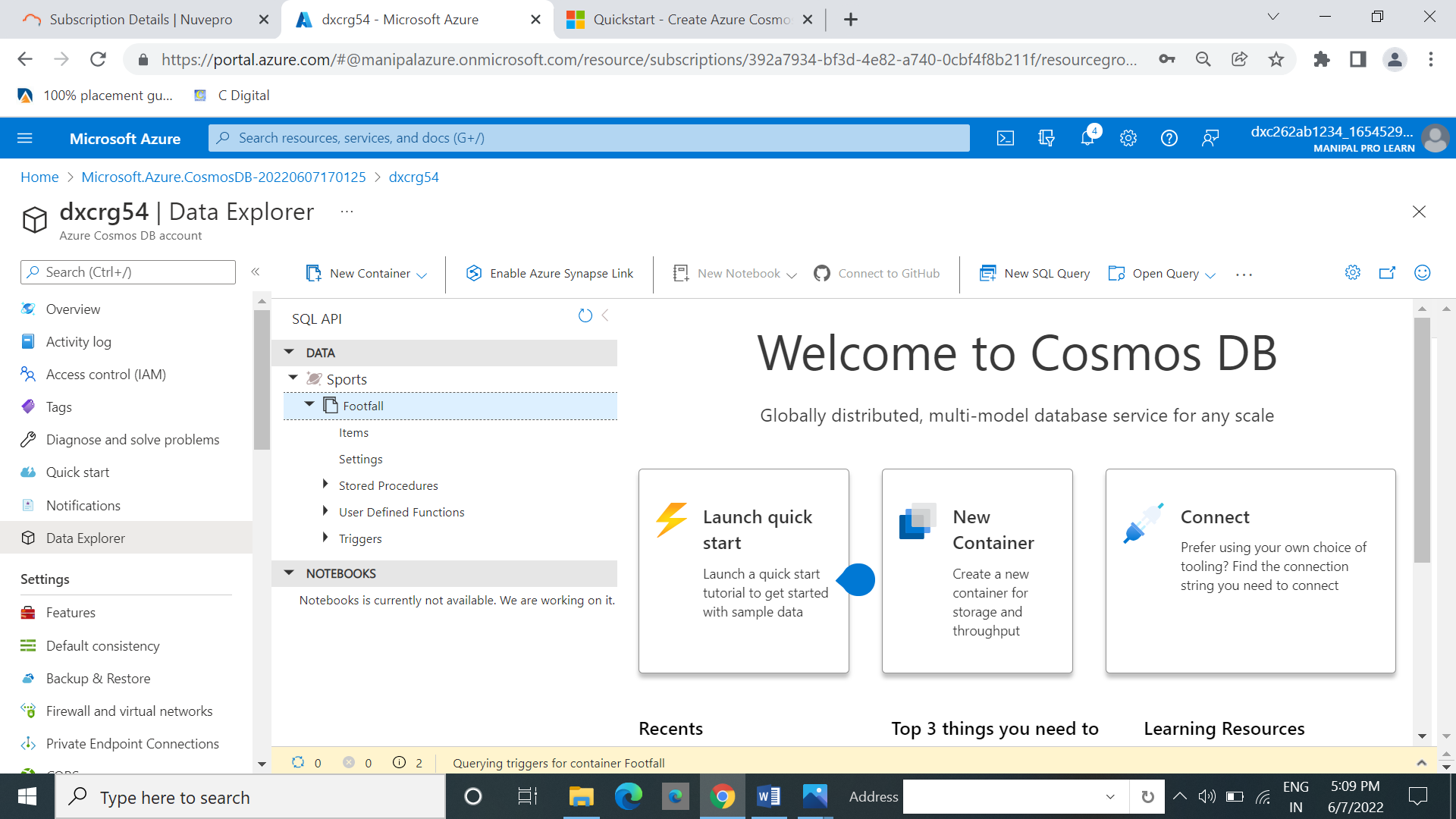
7.Select **Create.** It takes a few minutes to create the account. Wait for the portal page to display **Your deployment is complete.**



8.Select **Go to resource** to go to the Azure Cosmos DB account page.



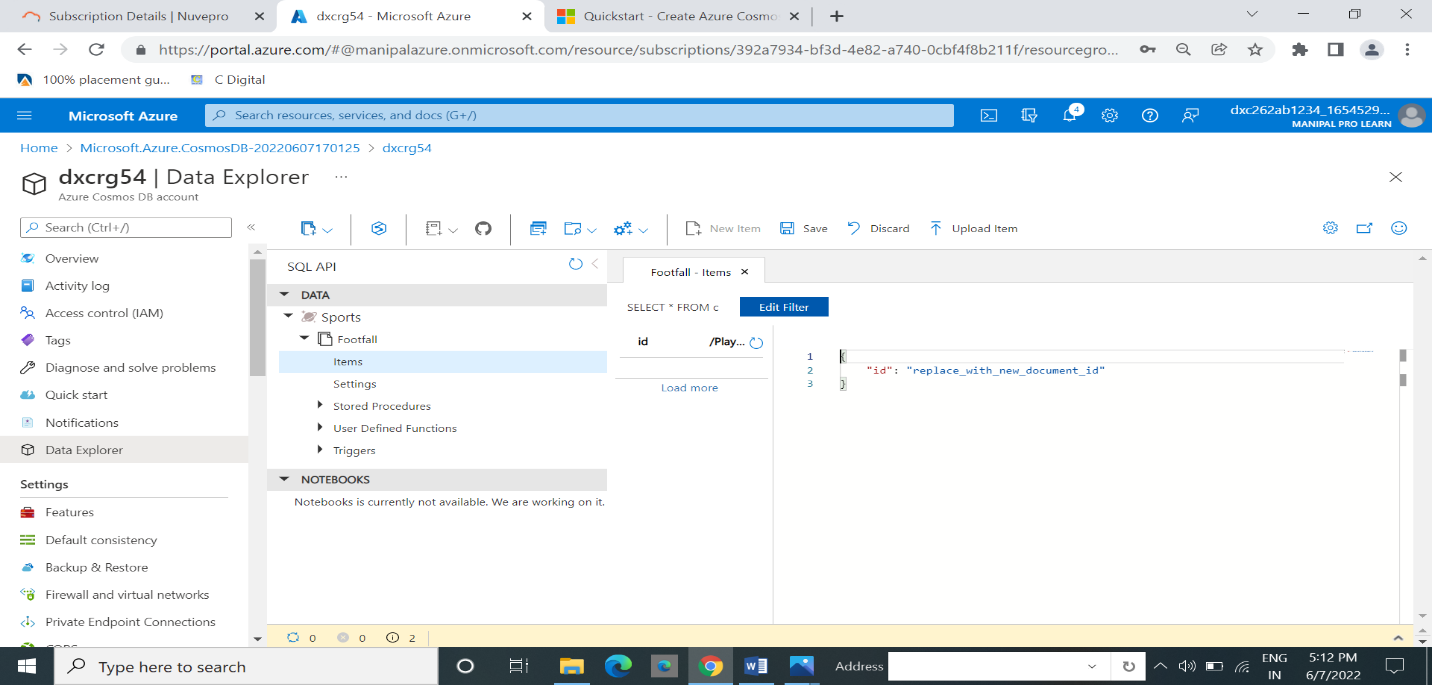
1. Select **Data Explorer** from the left navigation on your Azure Cosmos DB account page, and then select **New Container.**



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

1.In data explore .The the data is present .we insert some data into it.

2.From that data select items. And then select new items.

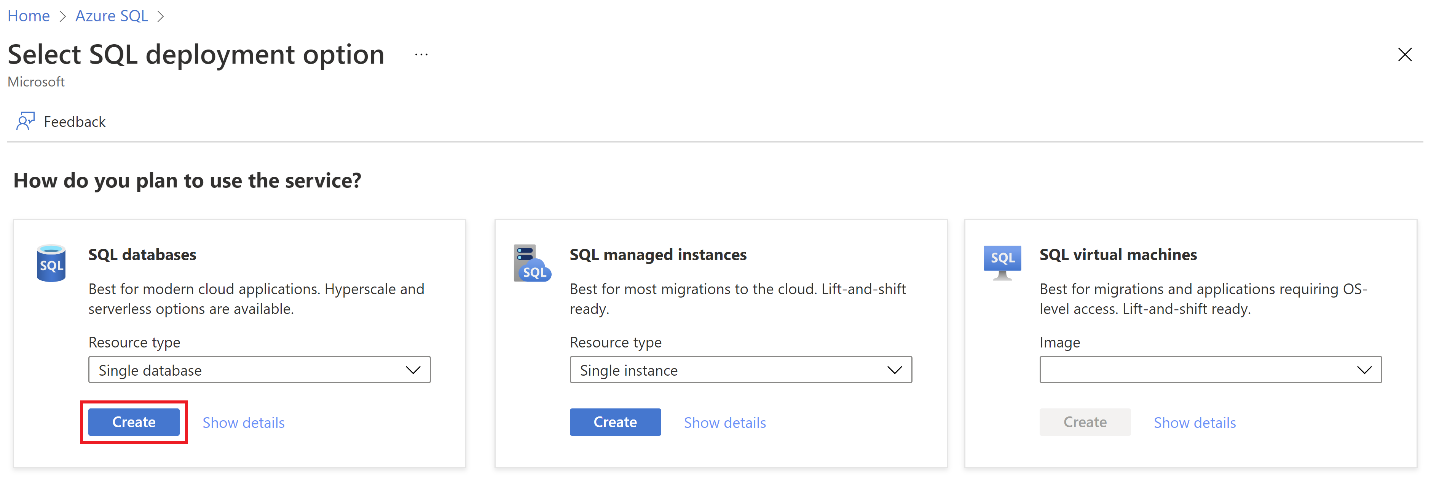


3.In the left side there is editer in that we can create and click on save.

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

Browse to the [Select SQL Deployment option](https://portal.azure.com/#create/Microsoft.AzureSQL) page.

Under SQL databases, leave Resource type set to Single database, and select Create.



On the Basics tab of the Create SQL Database form, under Project details, select the desired Azure Subscription.

For Resource group, select Create new, enter myResourceGroup, and select OK.

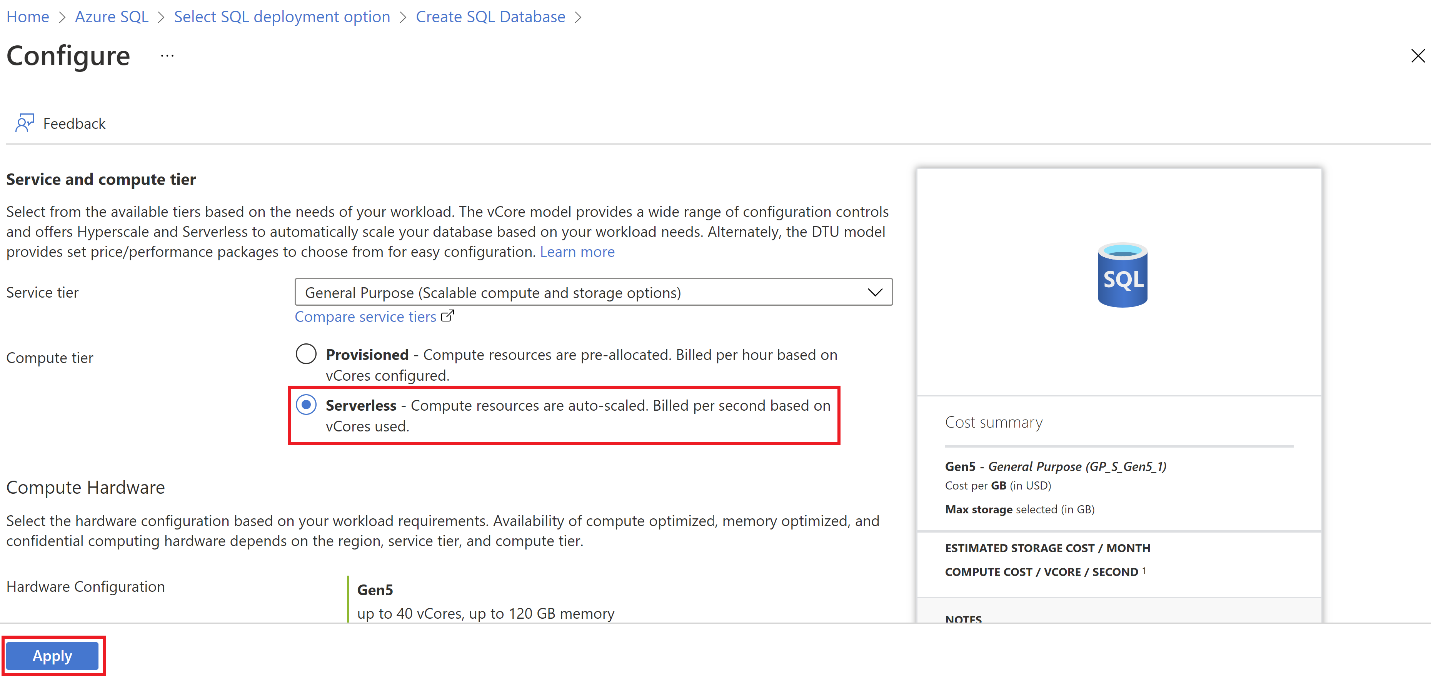
For Database name, enter mySampleDatabase.

For Server, select Create new, and fill out the New server form with the following value

Leave Want to use SQL elastic pool set to No.

Under Compute + storage, select Configure database.

This quickstart uses a serverless database, so leave Service tier set to General Purpose and set Compute tier to Serverless. Select Apply.



select **Next: Networking** at the bottom of the page.



On the Networking tab, for Connectivity method, select Public endpoint.

For Firewall rules, set Add current client IP address to Yes. Leave Allow Azure services and resources to access this server set to No.

Select Next: Security at the bottom of the page.

1. In the portal, search for and select SQL databases, and then select your database from the list.
2. On the page for your database, select Query editor in the left menu.
3. Enter your server admin login information, and select OK.

