😊 Azure SQL – Transparent Data Encryption

What is Azure SQL Database?

Azure SQL Database is a **fully managed, Platform-as-a-Service (PaaS)** cloud-based relational database service provided by Microsoft Azure. It is built on **Microsoft SQL Server engine** and eliminates the need for managing infrastructure, updates, backups, and security patches.

Unlike **SQL Server on Azure Virtual Machines (VMs)**, which requires manual configuration and maintenance, Azure SQL Database is a **serverless, auto-scaling, and highly available** database service designed for modern applications.

It offers multiple deployment models:

- **Single Database** (isolated database, fully managed)
- **Elastic Pool** (multiple databases sharing resources)
- **Managed Instance** (compatible with on-prem SQL Server, designed for easy migration)

Key Benefits of Azure SQL Database

1. Fully Managed Service

- No need to handle software updates, backups, or patching.
- Automatic tuning and indexing improve performance without manual intervention.

2. High Availability & Reliability

- **99.99% SLA uptime** with built-in geo-redundancy and failover capabilities.
- Supports **Zone-redundant high availability** for business continuity.

3. Scalability on Demand

- **Auto-scale compute and storage** based on workload requirements.
- Serverless options available for cost-efficient usage.

4. Security & Compliance

- **Built-in Threat Detection, Advanced Data Security, and Transparent Data Encryption (TDE)**.
- Compliance with **ISO, GDPR, HIPAA, and SOC** standards.

5. Cost-Effective

- Pay only for resources used (**DTU-based and vCore-based pricing models**).
- Supports **Azure Hybrid Benefit** for additional savings.

6. Easy Integration with Azure Services

- Works seamlessly with **Power BI, Azure Synapse, Logic Apps, and Azure Functions**.
- Supports **AI-driven performance tuning**.

Use Cases of Azure SQL Database

1. Modern Web and Mobile Applications

- Used for SaaS applications requiring high availability and auto-scaling.

2. Multi-Tenant SaaS Applications

- Supports multiple customers by using **Elastic Pools** for cost-efficient database management.

3. Enterprise Data Warehousing

- Used for real-time analytics and business intelligence.

4. Disaster Recovery & Backup

- Ensures **automatic failover and geo-replication** for business continuity.

5. IoT and Real-Time Data Processing

- Stores **sensor data** and integrates with Azure services for analytics.

6. On-Premises to Cloud Migrations

- **Managed Instance** makes it easy to migrate on-prem SQL Server databases with minimal changes.

Conclusion

Azure SQL Database is an **intelligent, secure, and scalable** PaaS offering that **removes infrastructure management overhead** while ensuring **high availability, security, and cost efficiency**. It is ideal for businesses looking to **modernize applications, migrate on-prem SQL workloads, and build cloud-native solutions**.

What is Transparent Data Encryption (TDE)?

Transparent Data Encryption (TDE) is a security feature in **Azure SQL Database and SQL Server** that **encrypts data at rest** to protect it from unauthorized access. It ensures that stored data, including backups, is encrypted without requiring application-level changes.

How TDE Works?

- **Encryption occurs at the storage level** (data files and log files) using **AES (Advanced Encryption Standard) with a 256-bit key**.
- The encryption and decryption process is **transparent** to users and applications.
- The **Database Encryption Key (DEK)** is stored in the database but protected by **Azure Key Vault or a built-in server certificate**.
- When a user or application queries the database, data is automatically decrypted without manual intervention.

Key Features of TDE

1. **Automatic Encryption of Data at Rest**
    - Encrypts **data files, log files, and backups** to prevent unauthorized access.
2. **No Application Changes Required**
    - Works at the database engine level, so applications don't need modifications.
3. **Azure Key Vault Integration**
    - Can use **Azure Key Vault Managed HSM** for **Bring Your Own Key (BYOK)** security.
4. **Protection Against Physical Theft**
    - Prevents data theft even if someone gains access to the storage or backups.
5. **Supports Compliance and Regulations**
    - Helps meet **GDPR, HIPAA, ISO 27001, and PCI DSS** security standards.

Use Cases of TDE

1. **Protecting Sensitive Data**
    - Used in **financial, healthcare, and government** sectors to safeguard PII (Personally Identifiable Information).
2. **Securing Cloud Databases**
    - Ensures data security in **Azure SQL Database, Managed Instance, and SQL Server on VMs**.
3. **Preventing Data Breaches**
    - If storage media is stolen, **encrypted data remains unreadable** without the encryption key.
4. **Ensuring Regulatory Compliance**
    - Helps organizations meet compliance standards **without additional manual encryption efforts**.

Conclusion

Transparent Data Encryption (TDE) is a **built-in security feature** in SQL Server and Azure SQL Database that **encrypts data at rest**, ensuring compliance and protection against unauthorized access. It provides **seamless security** without requiring application changes, making it a **critical feature for enterprise-grade security**.

Process:

In this lab, we create an **Azure SQL Database**. First, navigate to **SQL databases** in the Azure Portal and click **Create**. Select your **subscription, resource group, and database name**. Create a new **SQL Server**, set **authentication credentials**, and configure **compute/storage**

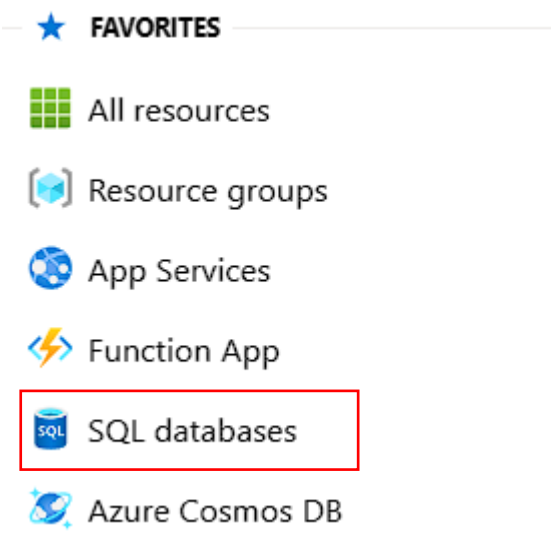
with **Basic DTU**. Enable **public endpoint networking** and firewall rules, then review and create the database. After deployment, connect via **Query Editor, Azure Data Studio, or SSMS**.

End Goal:

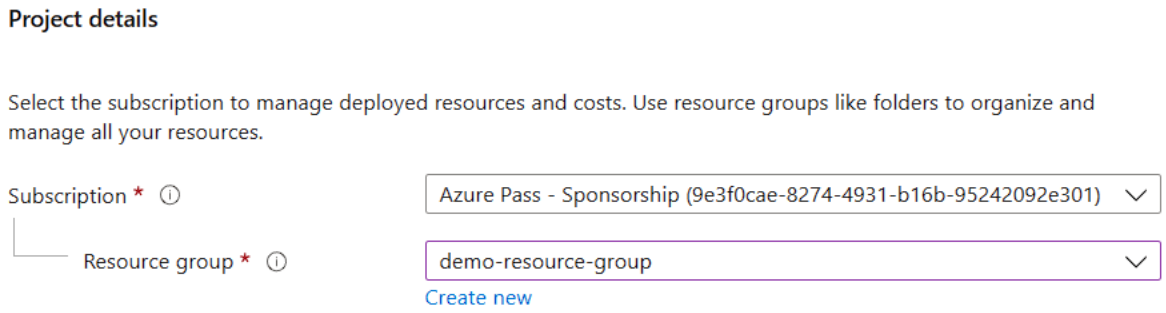
To set up an **Azure SQL Database with Transparent Data Encryption (TDE)** enabled by default, ensuring **secure, scalable, and managed cloud-based SQL storage** for applications.

😄 To begin with the Lab:

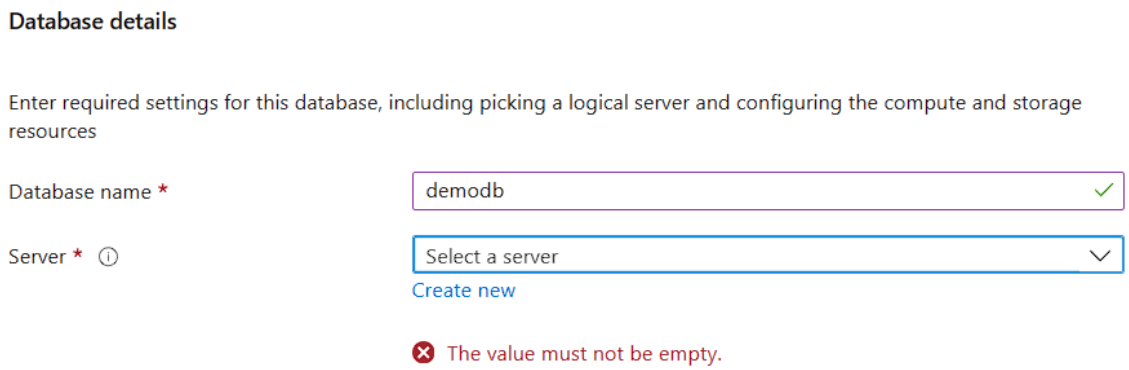
1. On your Azure Portal open All Resources. Then click on the hamburger icon.
2. There you will see the option for SQL databases. Click on it and from its dashboard click on create.



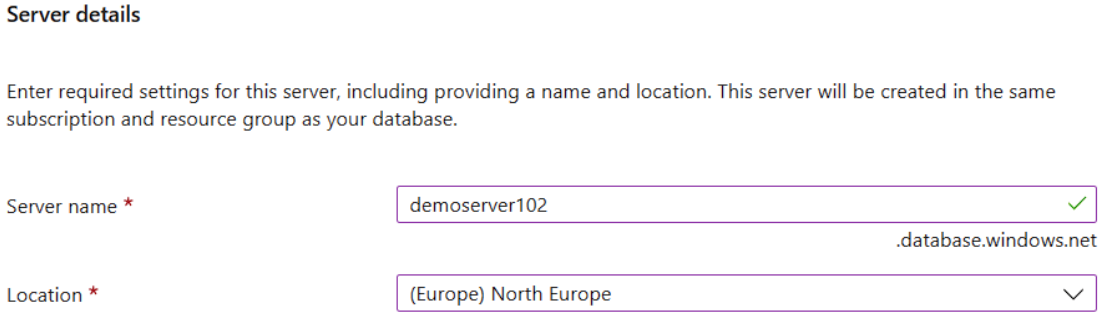
3. As always first select your subscription and resource group.



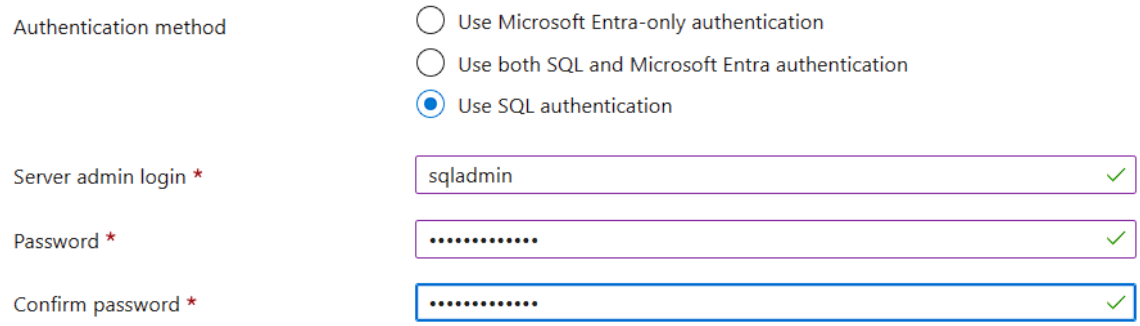
4. Now you have to give the database name then for the server click on create new and create a new server.



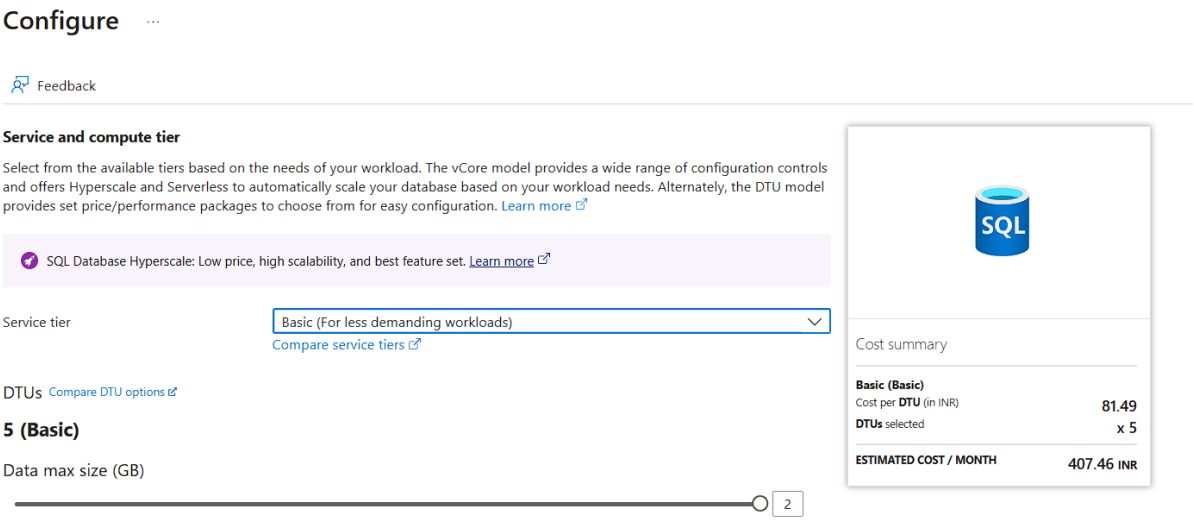
5. While creating your server first give it a unique name the select your location.



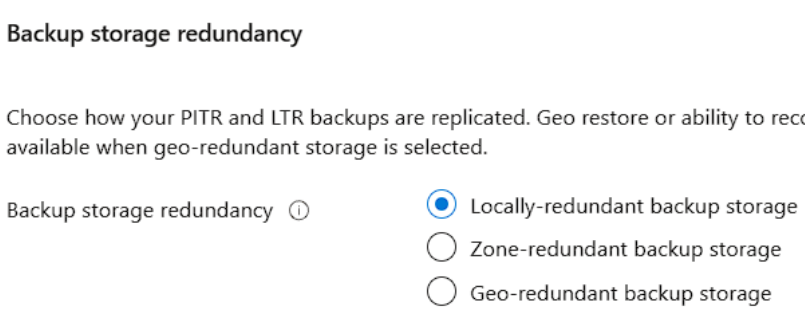
6. After that select the authentication method and create your user id and password.



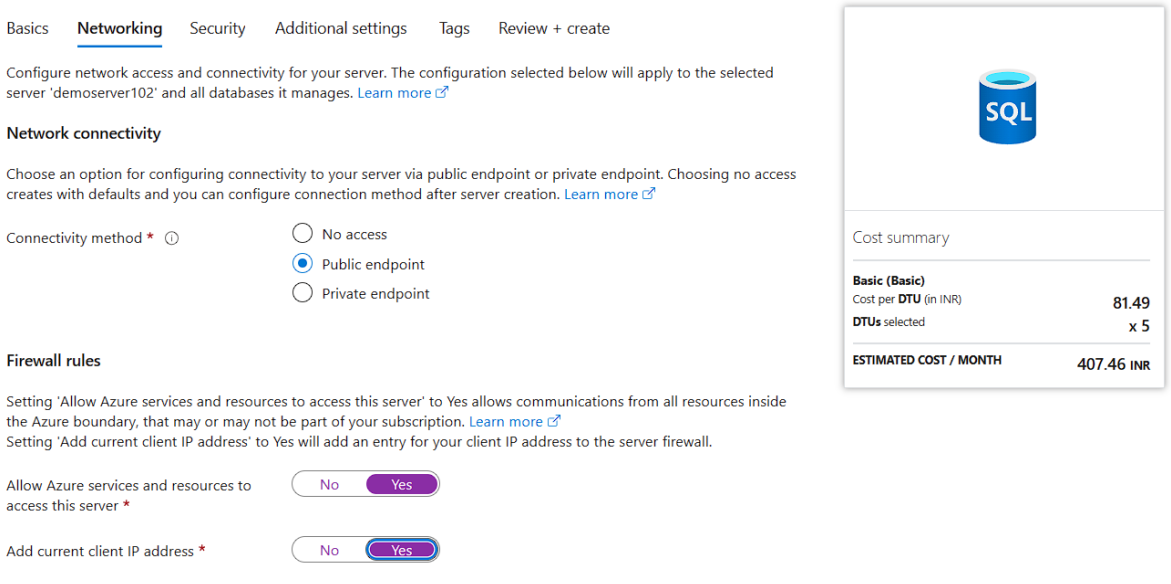
7. Now in the compute and storage option click on configure and select basic DTU for the workload.



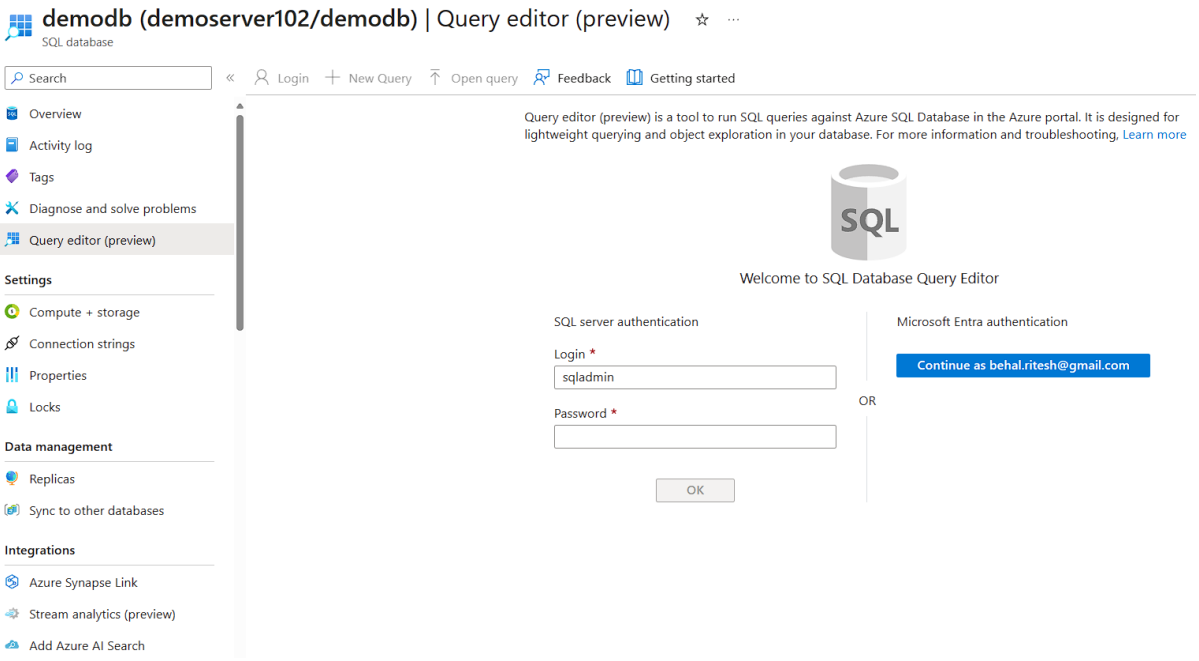
8. Choose LRS for your backup storage.



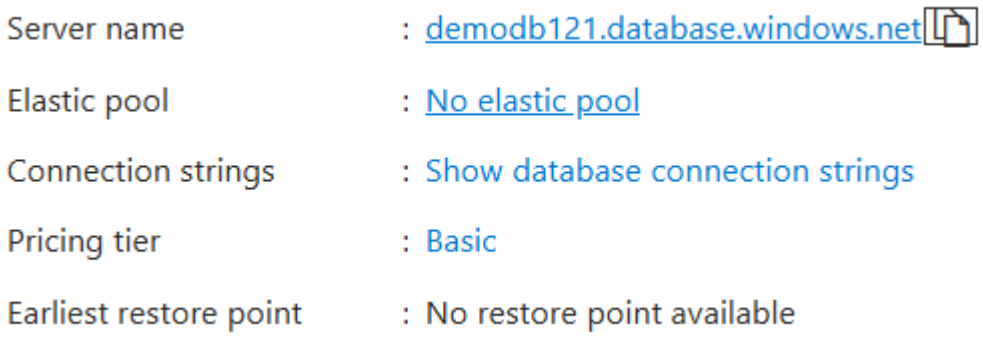
9. Now in the networking enable the public endpoint and say yes to both of the firewall rules.
10. After that directly jump to the review page and create your database.



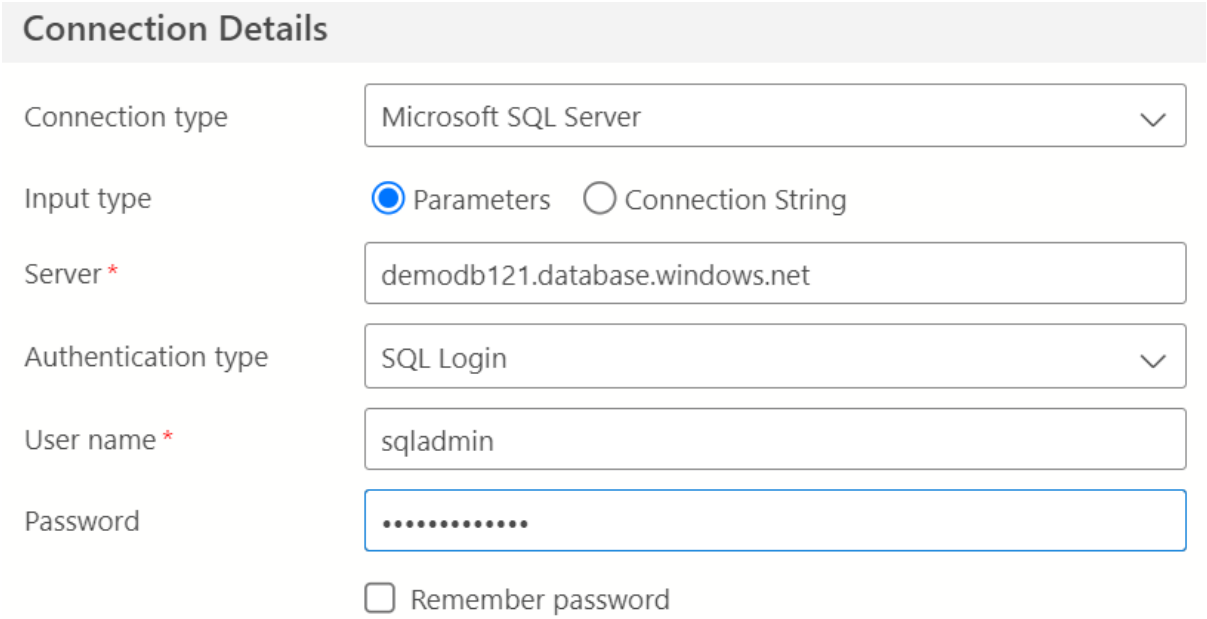
11. Now, this particular wizard is going to create two resources for us. One is a SQL database server and the other is a SQL database. This might take around three to five minutes.
12. Once the deployment is completed go to resources and open query editor.
13. Here you have to enter the password that you mentioned earlier while creating your database.



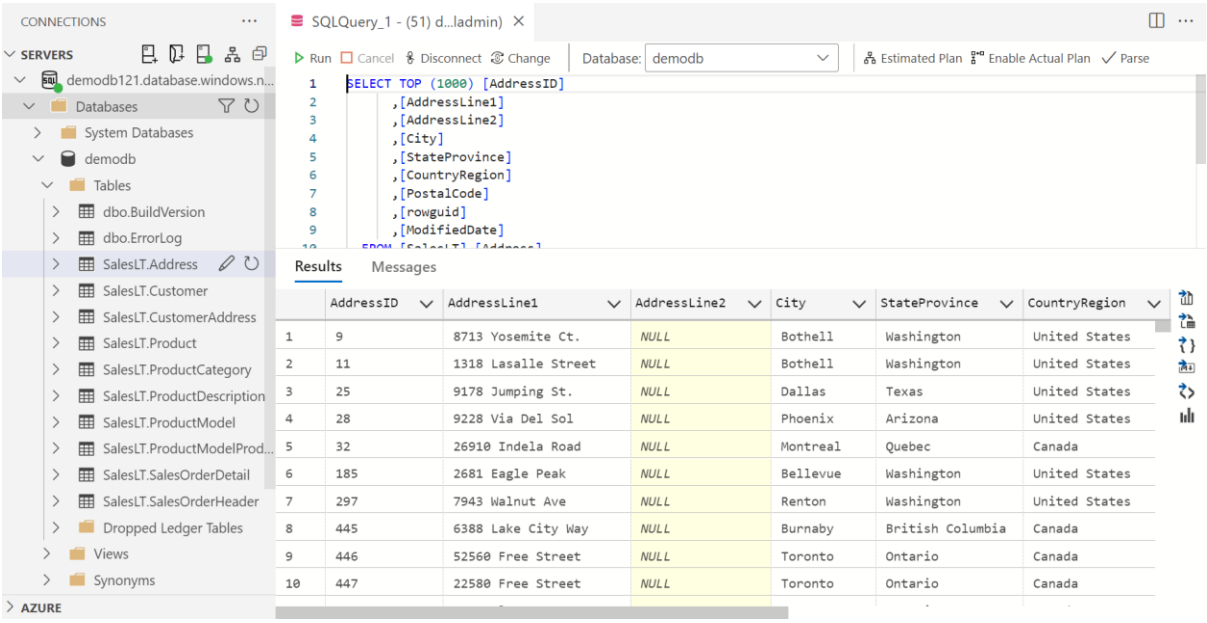
14. You can also connect your SQL Server with external applications like Azure Data Studio or SQL Server Management Studio.
15. Below I have used Azure Data Studio to connect my SQL Server. For that first you need to copy the server's name from the overview of your SQL Database and paste it in the Azure Data Studio.



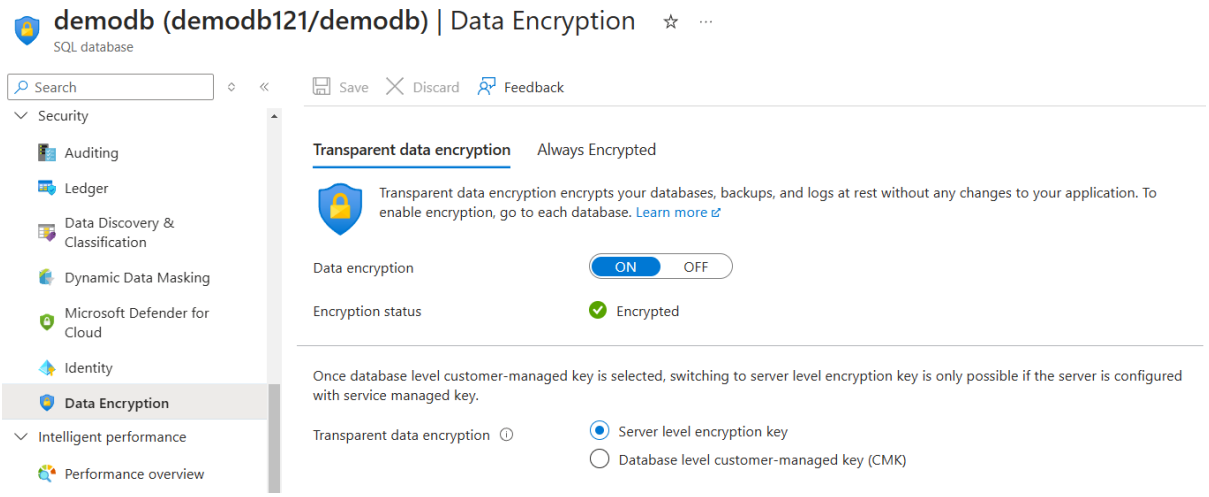
16. Also, add the username and password then click on connect.



17. You will be able to see the SQL Database. Also, if you expand the Demo DB then tables you will see the tables that come with the sample database which we enabled while creating the SQL Database.
18. You can right click on any table to select the top 1000 rows and columns of information.



19. Now on the Azure Portal for your Database if you go to security and open the Data encryption for your SQL Server then you will see that the Transparent data encryption option is enabled by default and the security is being managed by the SQL Database itself. This means that the key used here is called PMK or platform-managed keys.
20. You can also use the CMK or Customer managed keys by enabling the database level customer-managed-key option and choosing the keys from your Azure Key Vault.


demodb (demodb121/demodb) | Data Encryption
SQL database
Search
Security
Auditing
Ledger
Data Discovery & Classification
Dynamic Data Masking
Microsoft Defender for Cloud
Identity
Data Encryption
Intelligent performance
Performance overview
Save
Discard
Feedback
Transparent data encryption
Always Encrypted
Transparent data encryption encrypts your databases, backups, and logs at rest without any changes to your application. To enable encryption, go to each database. Learn more
Data encryption
ON
OFF
Encryption status
Encrypted
Once database level customer-managed key is selected, switching to server level encryption key is only possible if the server is configured with service managed key.
Transparent data encryption
Server level encryption key
Database level customer-managed key (CMK)