

# Azure Database SQL Database Setup and Cost Breakdown - As of 8th February

To set up an Azure Databricks workspace and create a SQL database using a SQL warehouse, follow these steps:

## 1. Create an Azure Databricks Workspace:

- **Sign in to the Azure Portal:** Navigate to <https://portal.azure.com> and log in with your Azure credentials.
- **Create a Resource:** In the Azure portal, select **Create a resource > Analytics > Azure Databricks**.
- **Configure Workspace Settings:** Provide the following details:
  - **Workspace Name:** Choose a unique name for your workspace.
  - **Subscription:** Select your Azure subscription.
  - **Resource Group:** Create a new resource group or select an existing one.
  - **Location:** Choose the region closest to your users.
  - **Pricing Tier:** Select between Standard, Premium, or Trial.
- **Review and Create:** After configuring, click **Review + create**, then **Create**. Deployment may take a few minutes.

## 2. Launch the Azure Databricks Workspace:

- Once the deployment is complete, navigate to your Azure Databricks resource and select **Launch Workspace**.

## 3. Create a SQL Warehouse:

- **Access SQL Warehouses:** In the Azure Databricks workspace, click on the **SQL** persona (icon) on the sidebar.
- **Create a New Warehouse:** Navigate to the **SQL Warehouses** tab and click on **Create Warehouse**.
- **Configure Warehouse Settings:** Provide the following details:
  - **Name:** Enter a name for your SQL warehouse.
  - **Cluster Size:** Choose an appropriate size based on your workload. (Make sure you choose a smaller cluster size as per your dataset and experimentation needs)
  - **Auto Stop:** Set the auto-stop interval to manage costs. (Keep auto stop to a minimum to avoid unnecessary accumulated cost).
- **Create:** After configuring, click **Create**.

## 4. Create a SQL Database:

- **Access SQL Editor:** In the Azure Databricks workspace, click on the **SQL** persona (icon) on the sidebar.
- **Create a New Query:** Navigate to the **Queries** tab and click on **New Query**.
- **Write SQL Commands:** Use the SQL editor to write commands to create and manage your database. For example:  

```
CREATE DATABASE my_database;
USE my_database;
CREATE TABLE my_table (id INT, name STRING);
```

### **Cost Considerations:**

Azure Databricks pricing is based on Databricks Units (DBUs), which are a unit of processing capability per hour. The cost depends on the workload type and the chosen pricing tier.

- **SQL Compute:** \$0.22 per DBU-hour
- **SQL Pro Compute:** \$0.44 per DBU-hour
- **Serverless SQL:** \$0.70 per DBU-hour

### **Example :**

When configuring a Serverless SQL Warehouse in Azure Databricks, selecting a larger instance size, such as **X-Large**, will impact your costs due to increased resource consumption.

Selecting an appropriate instance is of utmost importance otherwise the costs might escalate quickly, here's a cost breakdown for extra large instance with Serverless SQL to guage a maximum range.

### **Cost Breakdown for the Upper Limit:**

- **Databricks Unit (DBU) Consumption:** Each instance size corresponds to a specific number of DBUs. For example, an **X-Large** instance consumes 80 DBUs per hour.
- **DBU Pricing:** For Serverless SQL, the cost is \$0.70 per DBU-hour.

### **Estimate for an X-Large Instance and :**

- **DBU Cost:**  $80 \text{ DBUs} \times \$0.7 \text{ per DBU-hour} = \$56 \text{ per hour.}$

Therefore, operating an **X-Large** Serverless SQL Warehouse will cost approximately \$56 per hour.

It's important to note that Serverless SQL pricing includes both the DBU and underlying cloud instance costs.

These prices are estimates and can vary based on region and specific configurations. For the most accurate and up-to-date pricing, use the [Azure Pricing Calculator](#) and refer to the [Azure Databricks Pricing page](#).

To connect to a Databricks SQL Warehouse, you need the following access credentials:

1. **Databricks Server Hostname**
2. **HTTP Path**
3. **Personal Access Token**

## How to Find Access Credentials in Azure Databricks

### 1. Get the Server Hostname and HTTP Path

Follow these steps to retrieve the connection details:

1. **Open Azure Databricks Workspace**
  - Go to [Azure Portal](#).
  - Navigate to **Databricks** and launch your workspace.
2. **Go to SQL Warehouses**
  - In the Databricks UI, click the **SQL persona** (top-left corner).
  - Click **SQL Warehouses** from the left-hand menu.
3. **Select Your SQL Warehouse**
  - Click on the **SQL warehouse** you want to connect to.
  - Under the **Connection Details** section, you will find:
    - **Server Hostname** (e.g.,  
`adb-1234567890123456.15.azuredatabricks.net`)
    - **HTTP Path** (e.g., `/sql/1.0/warehouses/abcdef123456`)

### 2. Generate a Personal Access Token

A personal access token (PAT) is required for authentication when connecting through APIs or Python.

1. **Go to User Settings**
  - In the Databricks UI, click on your profile picture (top-right corner).
  - Select **User Settings**.
2. **Create a New Token**
  - Under the **Access Tokens** tab, click **Generate New Token**.
  - Provide an optional description and expiration date.
  - Click **Generate**.
  - **Copy the token immediately**—you won't be able to see it again.

## Using These Credentials to Connect

### 1. Connect Using Python

Install the Databricks SQL Connector:

```
pip install databricks-sql-connector
```

Use the credentials to connect:

```
from databricks import sql

connection = sql.connect(
    server_hostname="adb-1234567890123456.15.azuredatabricks.net",
    http_path="/sql/1.0/warehouses/abcdef123456",
    access_token="your-generated-token"
)

cursor = connection.cursor()
cursor.execute("SELECT * FROM my_database.my_table")
result = cursor.fetchall()

for row in result:
    print(row)
```

### **Other Components:**

- Open AI API Call Charges for Text-to-SQL conversion
- Ec2 deployment on t3 micro

### **Cost Breakdown**

Component	Description	Usage (Experimentation Phase)	Estimated Cost (GPT-4o)	Estimated Cost (GPT-4o Mini)
<b>API Calls</b>	OpenAI API for text-to-SQL query generation	100 queries (100,000 input + 50,000 output tokens)	\$0.75	\$0.045
<b>Compute Resources</b>	AWS EC2 t3.micro instance	10 hours @ \$0.0104/hour	\$0.104	\$0.104

GPT-4o - 128k context length

Price

Input:

\$2.50 / 1M tokens

Cached input:

\$1.25 / 1M tokens

Output:

\$10.00 / 1M tokens

GPT-4o mini - 128k context length

Price

Input:

\$0.150 / 1M tokens

Cached input:

\$0.075 / 1M tokens

Output:

\$0.600 / 1M tokens

**Total Estimated Cost for Experimentation Phase:**

**USD 5 should be enough to experiment and run multiple queries on the API.**

**Recommendation:**

- Opt for GPT-4o Mini for cost savings unless higher accuracy is required.
- Monitor API usage to optimize costs and minimize unnecessary token consumption.
- Utilize AWS and Databricks cost optimization tools to manage expenses effectively.