

# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS



Authored by:  
**Doddipalli Vamsy Reddy,**  
Senior Database Administrator

**Kaviarasan Selvaraj,**  
Marketing Manager

# C O N T E N T S

	Page No.
→ Introduction	3
→ What is Azure SQL Server Database?	3
→ What are Elastic Pools?	3
→ Is this really a cost-effective solution?	4
→ When to consider the Elastic Pools?	4
→ How to determine the databases good for Elastic Pool	5
→ Creating and Managing the Elastic pool	6
→ Tools to administer the Elastic pools	8
→ Monitoring the Elastic Pools	8
→ Conclusion	9
→ INDEX	9
→ DTU	9
→ eDTU	9
→ References	9

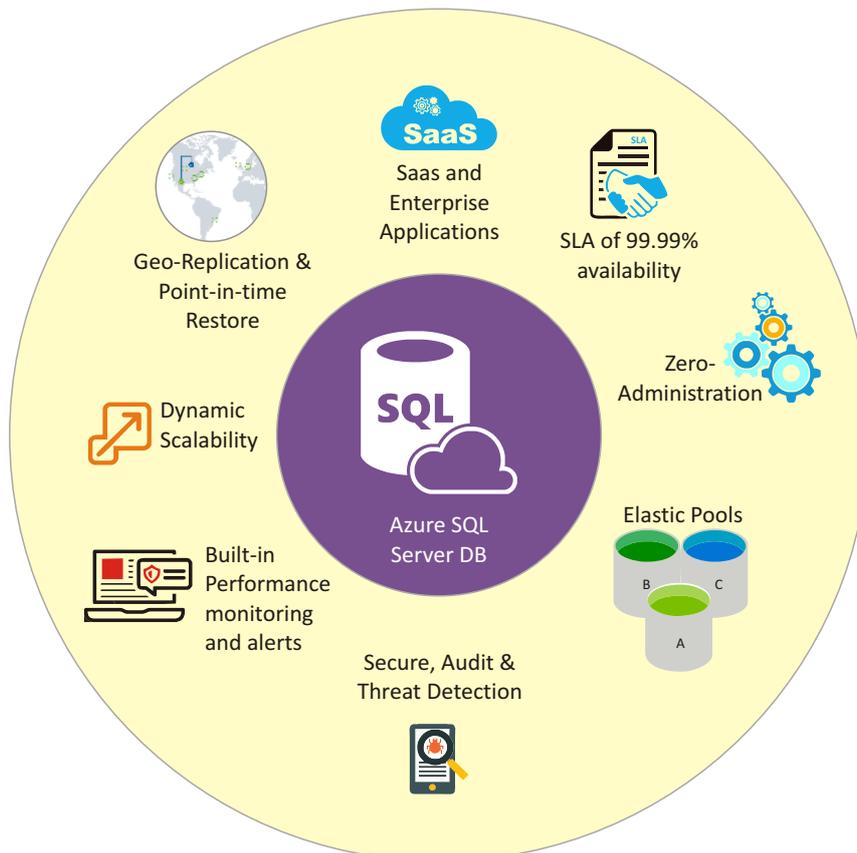
# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS

## Introduction

This whitepaper covers a feature called Elastic pools available in Azure SQL Server Database and the way it's implemented for optimal utilization to make for cost-effective solutions. You will know how to determine the right methodologies for implementation to boost ROI.

## What is Azure SQL Server database?

It's a relational database service that is capable of handling mission-critical workloads, multiple databases with unpredictable varying workloads and provisioning the utmost snug for those organizations focused in rapid app development to deploy into the production environment. Azure SQL Server database assimilates your unique app patterns, tunes the performance automatically and improves the reliability and data protection. It efficiently scales the performance with no-downtime and allows you to build multi-tenant applications with isolation benefits of one-customer-per-database.



## What are Elastic Pools?

Many organizations struggle with unpredictable workloads upon their multiple databases holding various applications. Propensity of leaking money can be in the below two ways:

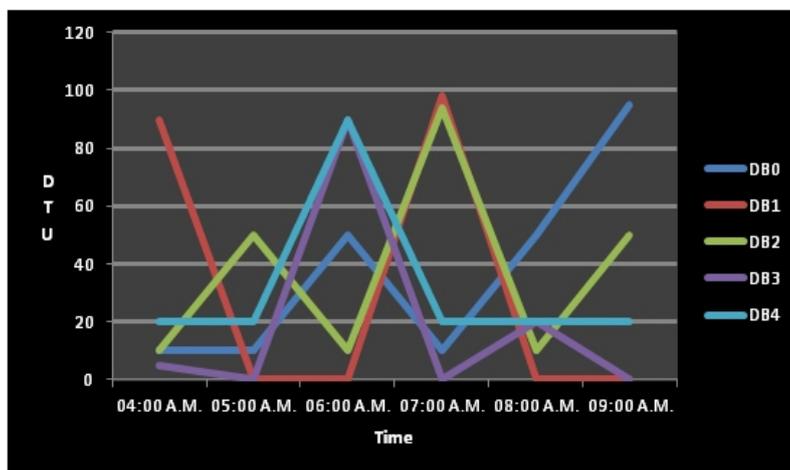
- ➡ Over-pay to the high resources all the time which brought based on peak usage calculations.
- ➡ Compromise on performance by provisioning the lower resources and experience the ineffective solutions.

# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS

A singleton database having dedicated resources always makes sense for a particular application having a detailed workload pattern. Besides, for multiple databases with varying and unpredictable usage patterns, an Elastic pool helps you provision the simple, cost-effective solution for managing and administering for eccentric workloads. The databases in the Elastic Pools leverage their performance by utilizing a bunch of shared resources (eDTUs\*\*) allocated as per the prescribed budget.

## Are Elastic Pools really a cost-effective solution?

Yes, at certain scenarios where the customer has multiple databases having highly varying workloads and it needs an abundant performance at infrequent times. Take an example of 5 databases, DB1 to DB5 which has unpredictable usage patterns and unknown peak and idle hour occurrences.



All the five databases consume the resources heterogeneously as per their disparate usage loads. So, based on the analysis chart, the aggregate DTU\* utilization of all databases never exceeds 100 and the high workload duration exists for a minimal stint of time. Elastic Pool allows sharing eDTUs across multiple databases within the pool. Apparently, this reduces cost up to 80% comparing with a single database expense. The estimated calculation is as follows.

### Single Database Price Estimation at Central US:

➔ S3 (100 DTU's) of 5 DB's \* \$149.99/Month = \$749.95/Month

### Elastic Pool Price Estimation at Central US:

- ➔ Basic (100 eDTUs with 200 DBs per pool) = \$148.80/Month (Up to 80% reduction of cost)
- ➔ Standard (100 eDTUs with 200 DBs per pool) = \$223.20/Month (Up to 70% reduction of cost)
- ➔ Premium (125 eDTUs with 50 DBs per pool) = \$174.38/Month (Up to 76% reduction of cost)

Additionally, you can include the new databases having similar usage pattern to the current Elastic Pool

## How to determine the databases good for Elastic Pool?

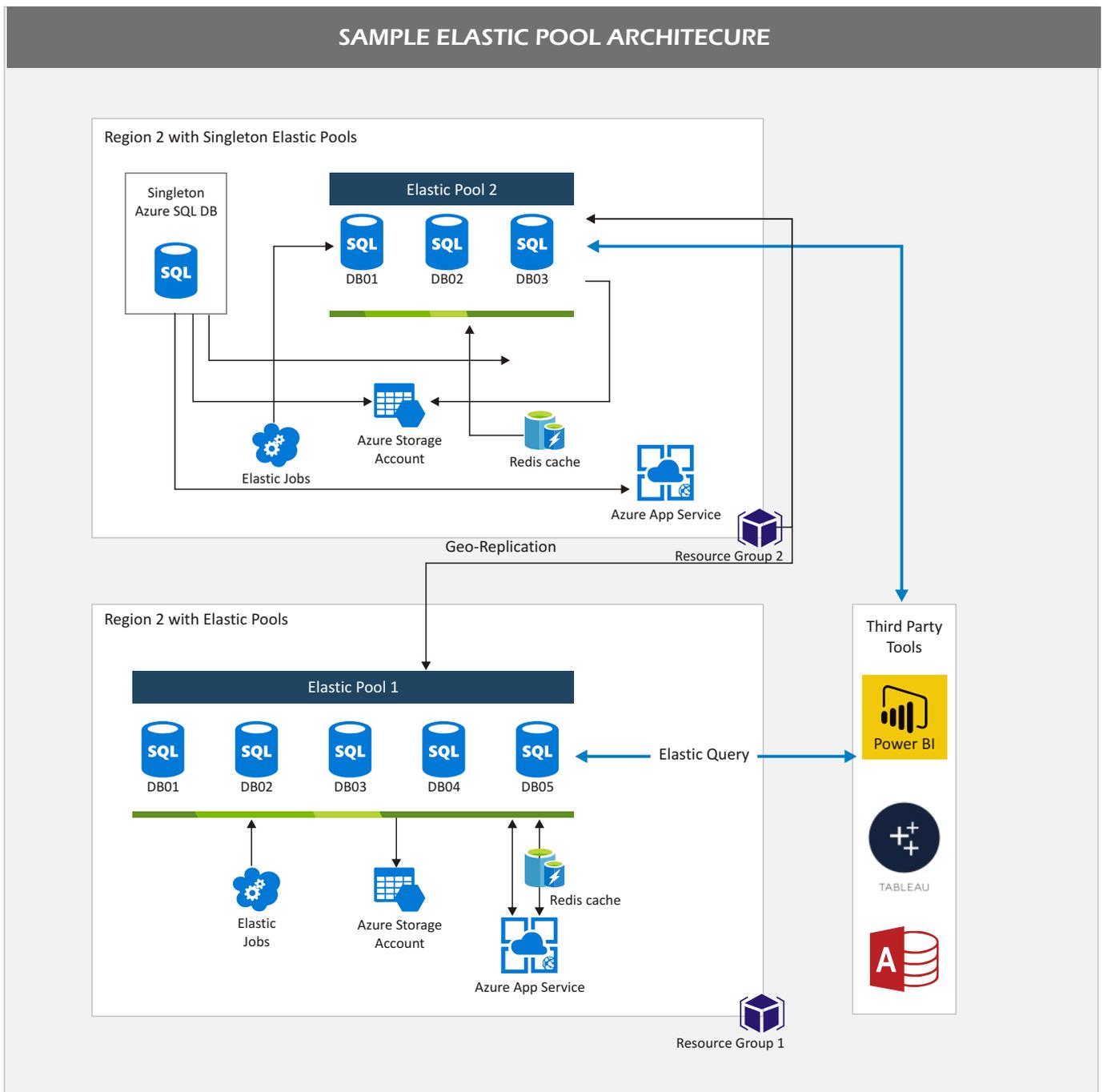
Taking advantage of the above example, on average DB4 uses 20 DTUs and rarely climbs to use 90 DTUs in the given period. This can be considered a good candidate for sharing eDTUs in Elastic Pools. On the contrary, if you have a database 'SDB1' that consumes the 90 DTUs for a long time and easily predicted heavy workloads frequently is bad for Elastic Pools.

### Essential considerations for adapting Elastic Pools:

- ➔ A pool is an ideal solution for databases that experience peak utilizations at infrequent times.
- ➔ The capacity of holding the databases varies by pricing model.
- ➔ Consider other vital limitations like Max concurrent users, Max concurrent logins, Max concurrent sessions, Max eDTUs per database, Max storage per database.

# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS

- ➔ If your business model changes to demand the dedicated resources to the database. You can remove the database from pool additionally Microsoft allows you to pick the right pricing tier before its release.
- ➔ Microsoft allows you to restrict minimum and maximum levels of utilization of eDTUs of each database individually.
- ➔ You still can configure high-Availability and disaster recovery solution 'Geo-replication' from one database in an Elastic pool to another datacenter either as a singleton database or a different Elastic pool with considering some limitations.



# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS

## Sizing Elastic Pools:

The efficient Elastic Pool can be determined by the Maximum DTUs utilized by all the databases in the pool and the total storage bytes by all databases in the pool. Microsoft at the backend always analyzes the consumption of resources on your databases and recommends the best elastic pool to optimize the performance.

**Estimate whether a pool is more cost-effective than single databases by the below calculations:**

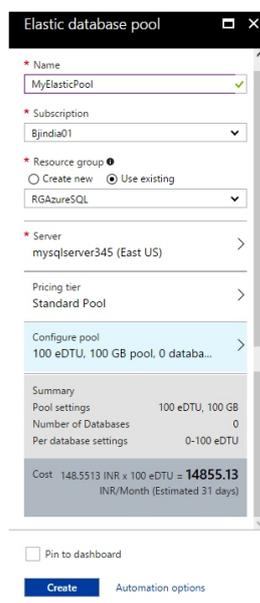
1. Estimate the eDTUs needed for the pool as follows:  
 $\text{MAX}\{<\text{Total number of DBs} \times \text{average DTU utilization per DB}>, <\text{Number of concurrently peaking DBs} \times \text{Peak DTU utilization per DB}>\}$
2. For pool storage limits based on eDTU pool size, see [eDTU and storage limits for elastic pools and elastic databases](#).
3. Take the larger of the eDTU estimates from Step 1 and Step 2.
4. See the [SQL Database pricing page](#) and find the smallest eDTU pool size that is greater than the estimate from Step 3.
5. Compare the pool price from Step 5 to the price of using the appropriate performance levels for single databases.

## Creating and Managing the Pool

You can create and manage the pool through Azure portal, PowerShell, the REST API & C#.

**Here are the some of the screenshots to illustrate how to create and manage the pools.**

1. Creating an Elastic Pool
  - a. **Login to Azure Portal -> More Services -> SQL Elastic Pools -> New Elastic Pool**  
(You've the option to add the databases either now or later after creating the pool)

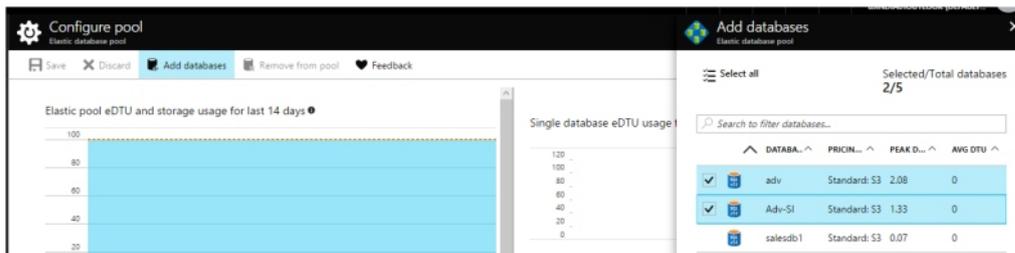


# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS

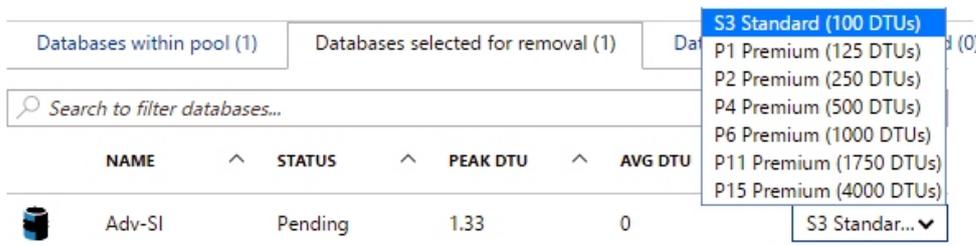
2. Choosing the Pricing Tier (Basic, Standard, Premium RS, Premium)
3. Configuring the pool
  - a. **SQL Elastic Pools -> MyElasticPool (Elastic Pool Name) -> Overview (or Settings) -> Configure Pool**



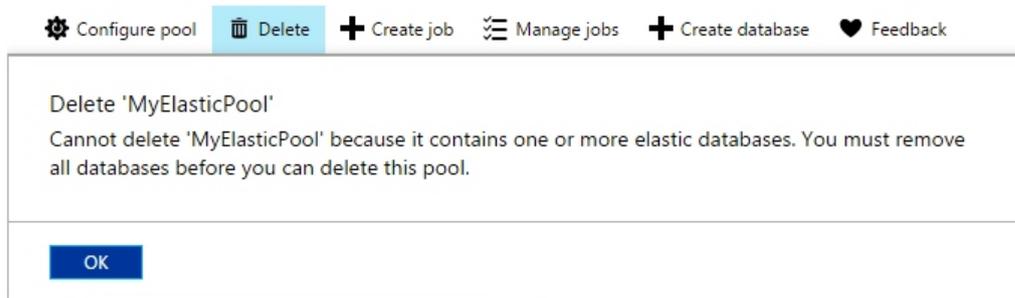
- b. **Configure pool -> Add databases -> Select databases on right corner blade -> Save**



4. Adding the existing database
  - a. **SQL Elastic pool -> Configure pool -> Add databases -> Select databases -> Save**
5. Removing the databases
  - a. **SQL Elastic pool -> Configure pool -> Databases within the pool -> Select the databases -> Remove from pool -> Save**
  - b. **NOTE: You have an access to choose the Single database pricing tier before releasing the pool.**



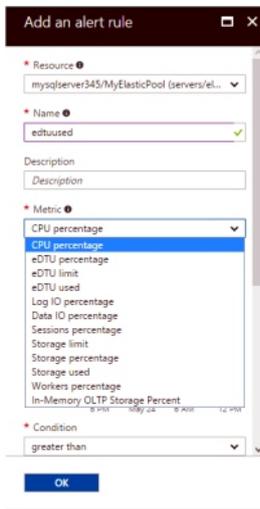
6. Deleting the Elastic Pool
  - a. **NOTE: The Elastic pool can't be deleted while the database is utilizing it.**



- b. Follow the 5.a, 5.b to remove all databases in the Elastic pool.
    - c. MyElasticpool (Elastic pool Name) -> delete

# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS

7. Configuring the alerts
  - a. MyElasticpool -> monitoring -> Alerts -> (provide name, metric, condition, threshold, Period, operators) -> Ok



## Tools to administer the Elastic pools:

**Elastic Database Split-Merge tool:** Used for data migration from Single-Tenant database to Multi-Tenant database and vice versa.

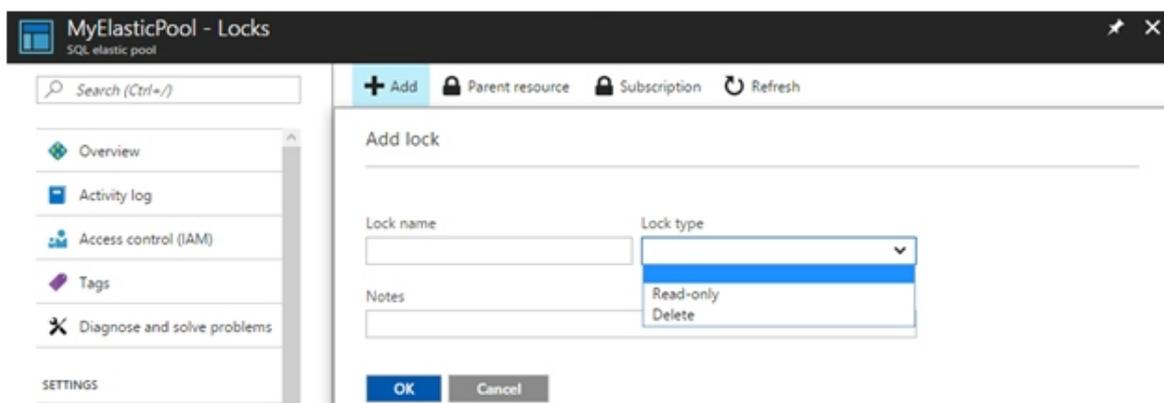
**Elastic Database Jobs (Preview state):** Used to administer the as schema changes, credentials management, reference data updates, performance data collection or tenant (customer) telemetry collection using jobs.

**Elastic Query (Preview state):** Allows you to trigger cross database queries, to access remote tables and to connect Microsoft Third party tools (Excel, PowerBI, Tableau, etc)

**Elastic Transactions:** Capable of running transactions that span multiple SQL DBs.

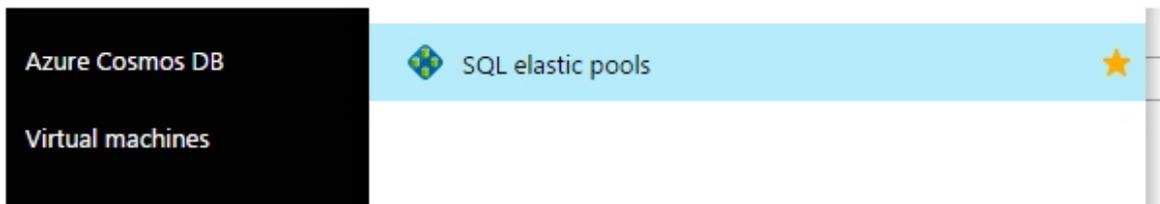
## Monitoring the Elastic Pools:

1. You can add and remove the database from Elastic Pool at any time.
2. Secure the accidental deletion of Elastic pool by locking the parent resource or subscription lock through settings blade.



# Reduce Azure SQL Server cost up to 70% through ELASTIC POOLS

3. To toggle the SQL Elastic pool to the favorites click 'Shift + Space' or highlight the start icon



## Conclusion

Given the unpredictable workloads that elastic pools need to contend with, flexibility of database utilization is optimized to make a superior cost-effective solution. ISVs and small to medium enterprises would serve to benefit with the right consultant to choose the right databases within the elastic pool for a self-fulfilling business ecosystem.

## Index

**\*Database Transaction Units “DTU”:** Isolated Azure SQL Server database relatively utilizes an amount of resources consumed to provide the predictable performance that is calculated as Database Transaction Units (DTUs). This is an equitable blended measure of CPU, Memory, I/O that provides the comparative performance levels at each price tier. The ratio amongst these resources was originally determined by an OLTP benchmark workload designed to be typical of real-world OLTP workloads. Increasing the DTUs upon a database apparently boosts the performance capabilities of the database.

**\*\* Elastic Database Transaction Units “eDTU” :** Regardless of dedicated single Azure SQL Server utilizing the dedicated set of resources (DTUs) whether needed or not, an elastic pool that shares a pool of resources to the SQL server databases that benefits the customer to manage the performance of multiple databases that have widely varying and unpredictable usage patterns. These shared resources in an elastic pool are measured by elastic Database Transaction Units or eDTUs. In an elastic pool, you can guarantee that no one database uses all of the resources in the pool and also that a minimum amount of resources is always available to a database in an elastic pool.

## References

<https://docs.microsoft.com/en-us/azure/sql-database/>



Aspire Systems is a global technology service firm serving as a trusted technology partner for its customers. The company works with some of the world's most innovative enterprises and independent software vendors, helping them leverage technology and outsourcing in Aspire's specific areas of expertise. Aspire System's services include Product Engineering, Enterprise Solutions, Independent Testing Services, Oracle Application Services and IT infrastructure & Application Support Services. The company currently has over 1,600 employees and over 100 customers globally. The company has a growing presence in the US, UK, Middle East, Europe and Singapore. For the seventh time in a row, Aspire has been selected as one of India's "Best Companies to Work For" by the the Great Place to Work® Institute, in partnership with The Economic Times.

SINGAPORE  
+65 3163 3050

NORTH AMERICA  
+1 630 368 0970

EUROPE  
+44 203 170 6115

INDIA  
+91 44 6740 4000

MIDDLE EAST  
+971 50 658 8831

For more info contact  
info@aspire.com or visit [www.aspiresys.com](http://www.aspiresys.com)

