SQL Azure Database V12 - Lessons learned from the trenches

José Manuel Jurado Diaz

Azure SQL Database Support Engineer / Subject Master Expert

NORDIC SQL Nexus

it-Craft

kapacity Business Analytics

Cognizant

Microsoft

About me..

Jose Manuel Jurado Diaz

Azure SQL DB Support Engineer

Subject Master Expert

Speaker SQL Saturday, channel9 and TechReady

Email: jmjurado@microsoft.com

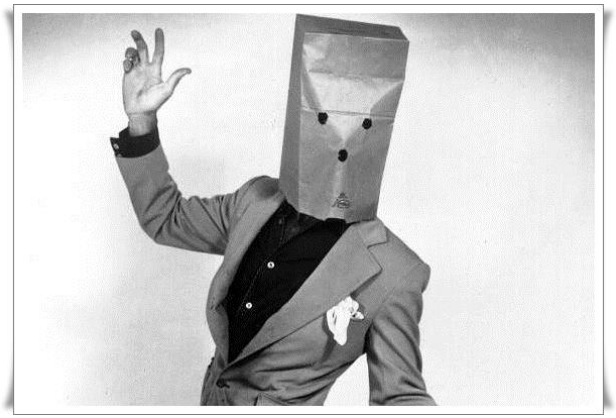
Session Objectives And Takeaways

- Session Objectives:
  - Raise awareness of Azure SQL DB common issues, help to improve customer's business continuity and prevent Azure SQL DB performance issues.
- Key Takeaway 1.
  - New Options in Azure SQL DB v12
- Key Takeaway 2.
  - New diagnostic tools for performance.
- Key Takeaway 3.
  - New database models and business continuity.

Microsoft

Before Getting Started - Questions

Did you work with SQL Azure previously? V11 or V12?



If not, do you know what is it?

If you know, did you have any trouble or issue working with SQL Azure previously?

SQL Azure Database V12

Introduction

To Azure SQL DB

Microsoft

Azure SQL Database – What is it?

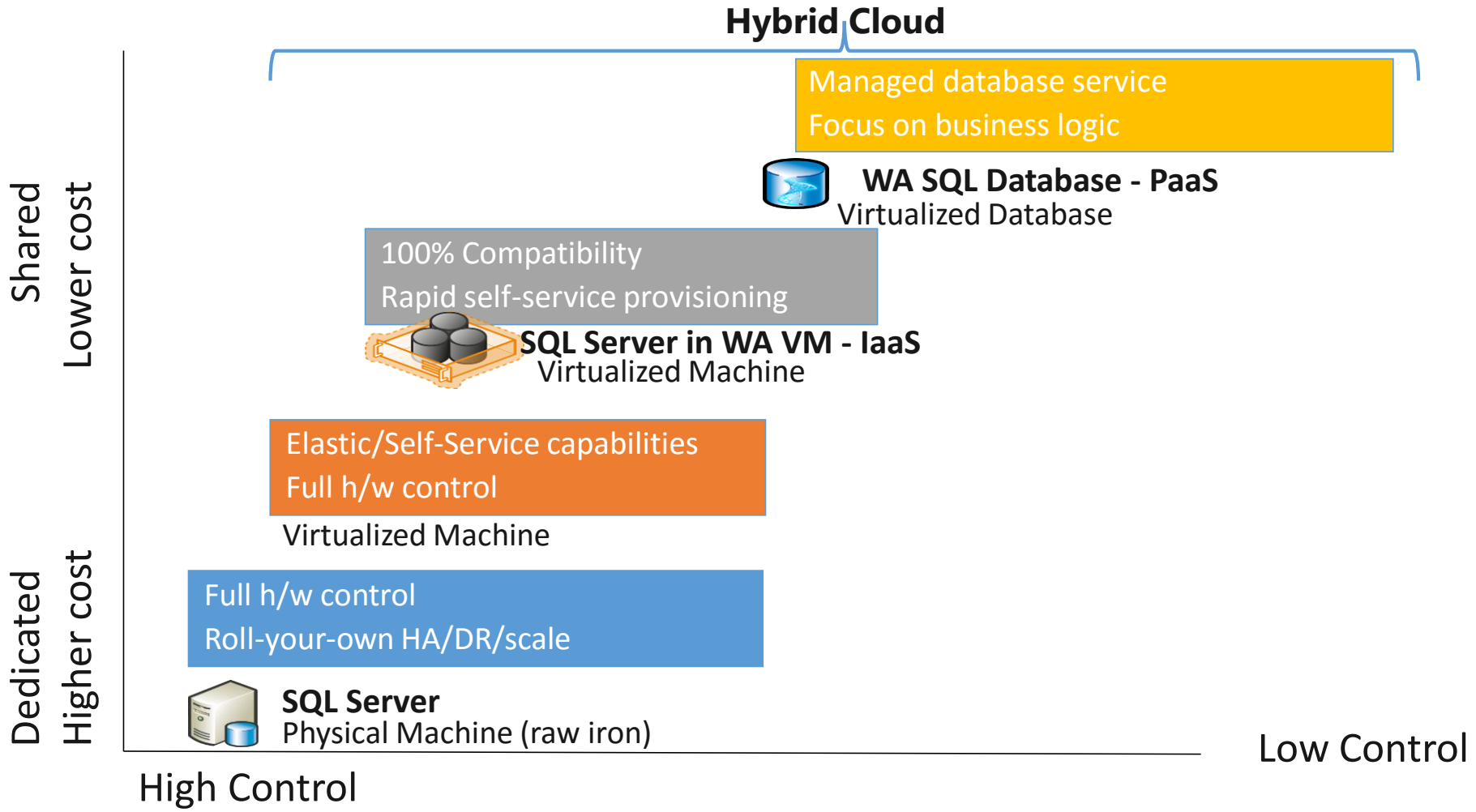
- → SQL Server as Service
- → Service managed by Microsoft
- → Enterprise Edition options, limited only by hardware characteristics depending on the model.
- → Almost 0 maintenance
- → Resources oriented to use your database

Microsoft

Azure SQL Database – Cloud Models

crosoft

Azure SQL Database – Database Platform



Azure SQL Database – Five Goals

Simple

Easy to create.

Dashboard metrics DB.

No need to change the application.

Multiple migration tools.

Familiarization.

Scalability

Predictive performance.

Multiple database model depending on use.

Parallel Data-Warehouse

Trusted

Azure AAD or SQL Login

Multiple security mechanisms.

Compliance

Auditing

Business Continuity

Disaster Recovery

Geo-Replication

Geo-Restore

Restore Point-in-Time.

Database Replicas.

Backups

Always Updated

Latest options available at company level.

Multiple monitoring tools.

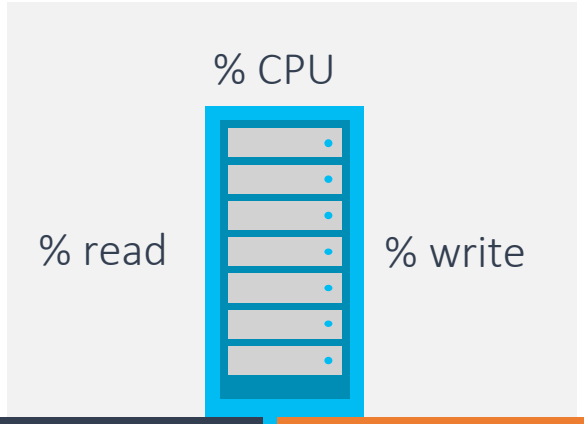
Microsoft

Azure SQL Database – Database Level/DTUs

Basic | Standard | Premium | Elastic DB Pools

Redefined

Across Basic, Standard, and Premium, each performance level is assigned a defined level of throughput



DTU is defined by the bounding box for the resources required by a database workload and measures power across the six performance levels.

Basic — 5 DTU
S0 — 10 DTU
S1 — 20 DTU
S2 — 50 DTU
P1 — 125 DTU
P2 — 250 DTU
P4 — 500 DTU

Measure of power

Introducing the Database Throughput Unit (DTU) which represents database power and replaces hardware specs

Azure SQL Database – Create First DB

Demo: Creating my first database

Microsoft

Agenda

- Differences with V11.
  - The good and the bad
- Changes and new improvements in Azure SQL Database V12
  - Engine
  - Security
- High Availability and Disaster Recovery
  - Geo-Replication and Geo-Restore
- The new database model and Elastic Tools
  - Elastic Database Pools
  - Elastic Query, Jobs, Scale and Transaction
- New Tools and connectivity improvements
  - Connectivity
  - Query Data Store, Query Performance Insights, Index Advisor, Extended Events, Alerts, DMVs.
- Q&A

Microsoft

SQL Azure Database V12

Differences with Azure SQL DB v11

Microsoft

Differences with V11 – The Good

- ✓No noisy neighbors.
- ✓Instance per database non-shared.
- ✓20%-30% improvement.
- ✓Direct connection and differences depending on data provider.
- ✓Predictive performance and DTUs
- ✓Our customer is able to restore backups.
- ✓Dedicated Administrative Connection.
- ✓A lot of new DMVs.
- ✓Deleted server.
  - ✓**Tip:**If you drop a server you wouldn't be able to reuse the name after
  - ✓5 days.

Microsoft

Differences with V11 – The Bad

- ✓ Compare Basic/Standard vs Web/Business.
  - ✓ Tip: Update stats, rebuild index, missing index. http://blogs.msdn.com/b/sqlblog/archive/2013/11/01/do-i-need-to-upgrade-my-dba-skills-for-the-cloud.aspx
  - ✓ Postpone the migration from V11 to V12: http://blogs.msdn.com/b/azuresqldbsupport/archive/2015/06/05/stopping-or-postponing-an-upgrade-to-sql-database-v12.aspx
  - ✓ Business and Web will be migrated in automatic way started at: 12th September - https://azure.microsoft.com/en-us/blog/azure-sql-database-web-and-business-edition-retirement-september-12th-2015/
- ✓ Retry-Logic is more needed than before.
- ✓ Once moved V12 not possible to go back.
- ✓ BulkCopy works but not loading entire files you could use bcp instead of.
- ✓ No more federations, now, Sharding framework. https://azure.microsoft.com/en-us/documentation/articles/sql-database-elastic-scale-use-entity-framework-applications-visual-studio/
- ✓ Lost HTML to manage the database.

SQL Azure Database V12

Changes and Improvements of Azure SQL DB

Microsoft

What is new using V12 Azure SQL Engine

Engine

- Partitioning
- Partial Full-Text Search
- Parallel Execution for >P2
- Contained database
- Tables In-Memory
- Temporal Tables (Versioning)
- DBCC commands
- CLR Integration – Stopped
- Table as Heap
- Data Compression
- Enhanced Session Context
- Setting changes Database
- Change tracking
- New functions and TSQL
- Strech Database

Compatibility Level to 130

- SQL Server 2016 (Preview)
- Azure SQL DB V12

Alter Database SCOPE Configuration

Available for:

- SQL Server 2016 (Preview)
- Azure SQL DB V12

Options:

- Procedure cache cleaning
- Maxdop
- Legacy Cardinality Estimation
- Parameter sniffing

Indexes

- Online Index.
- Column Store Index.
- XML Index.

Replication

- Transactional
- SQL Data-Sync

New Product

- Azure SQL DataWarehouse – Parallel Data Warehouse

High Availability

- Geo-Replication Offline/Online

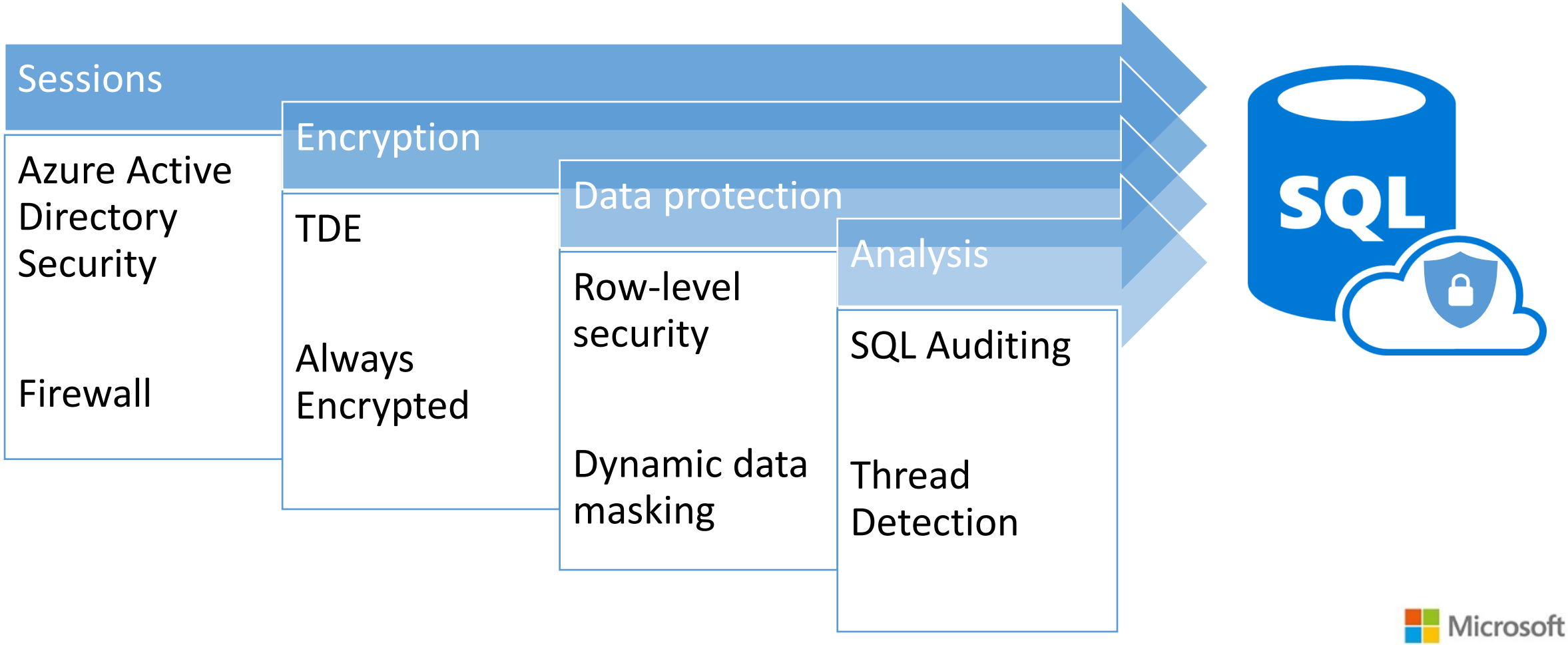
Microsoft

What is new using V12 Azure SQL Engine

Demo: Alter Database Scope Configuration

Microsoft

What is new using V12 Azure SQL Security



Transparent Data Encryption

Data is encrypted at rest, in flight, and while in use

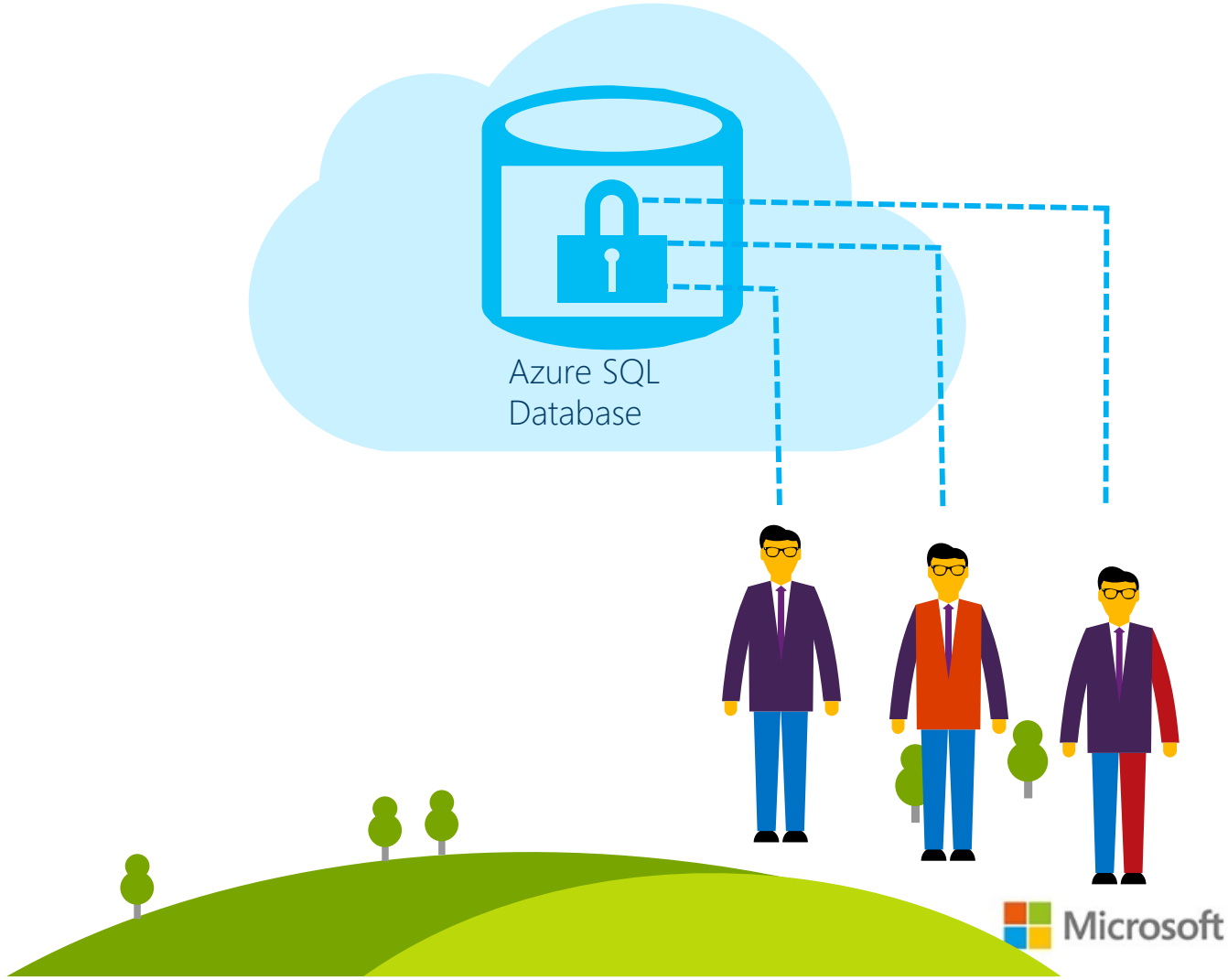
Azure SQL Database service manages your keys

You can keep application changes to a minimum

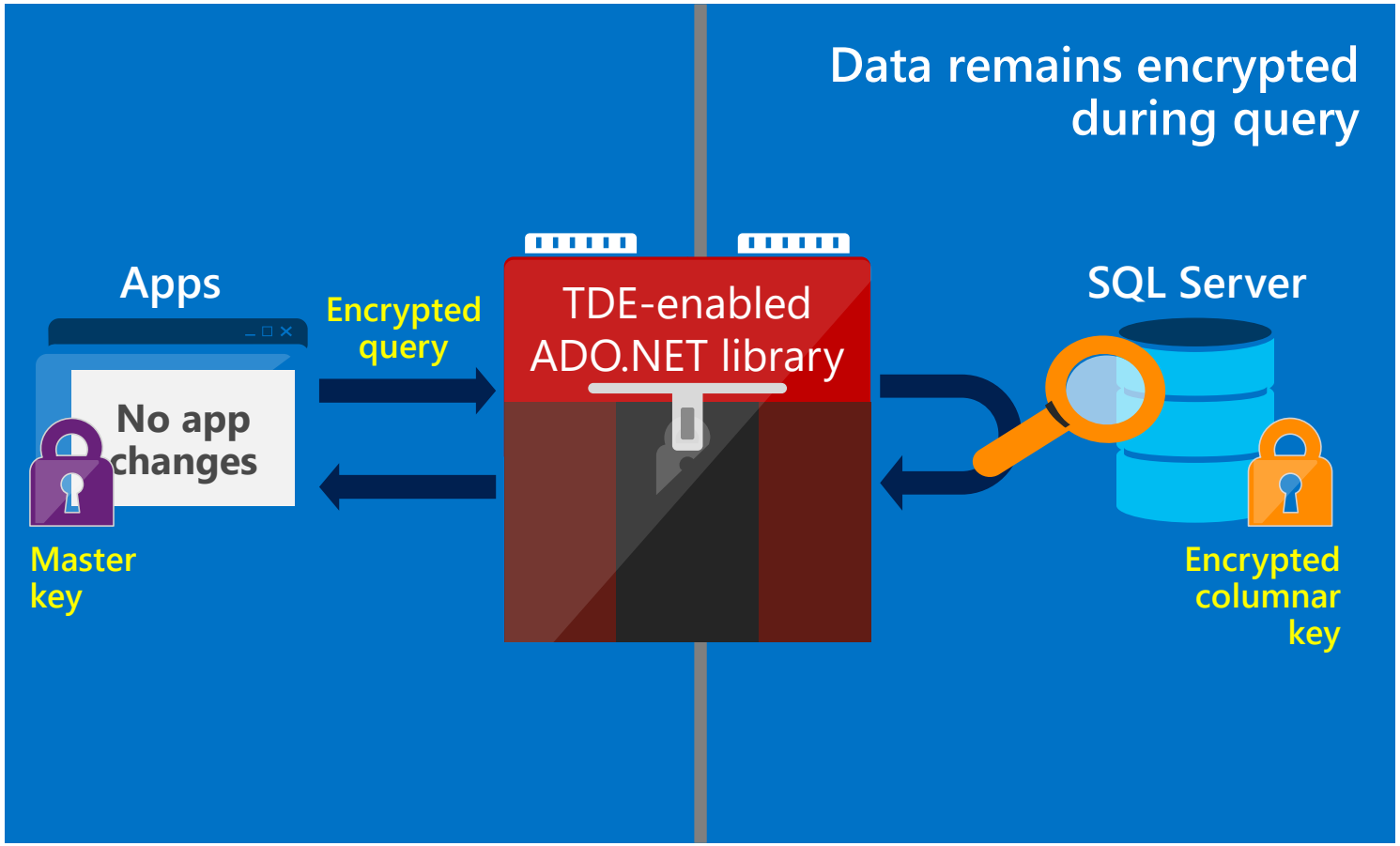
Encryption and decryption of data is done transparently in a transparent data encryption (TDE)–enabled client driver

Support for equality operations (including joins) on encrypted data

Azure manages encryption keys



Always Encrypted



Capability

Transparent client-side encryption, while SQL Server executes T-SQL queries on encrypted data

Benefits

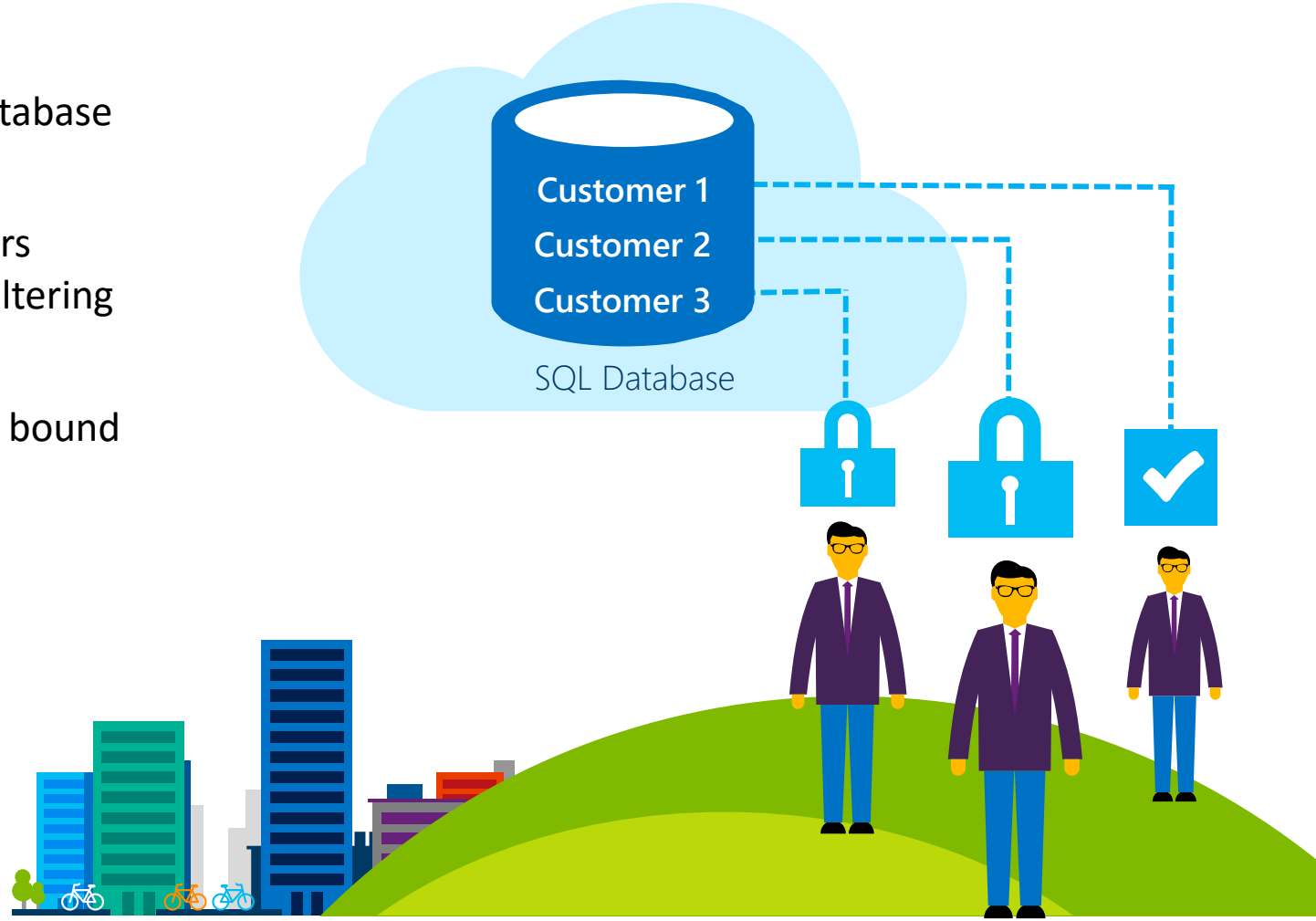
- Sensitive data remains encrypted and queryable at all times
- Unauthorized users never have access to data or keys
- No changes to applications are necessary

Row Level Security

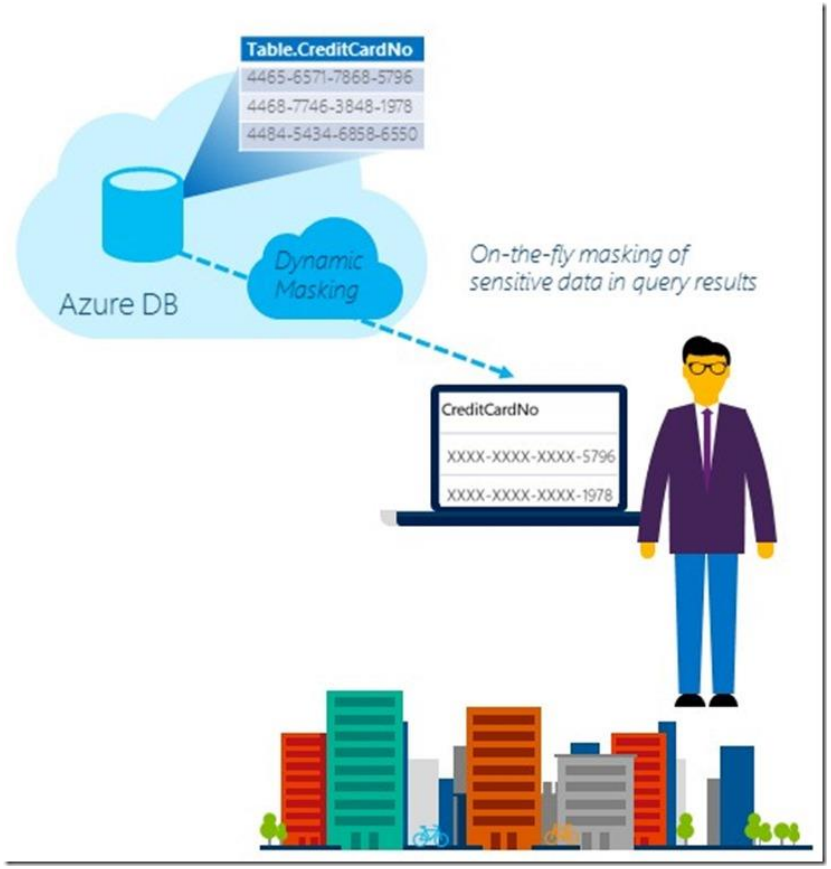
Fine-grained access control over specific rows in a database table

Help prevent unauthorized access when multiple users share the same tables, or to implement connection filtering in multitenant applications

Enforcement logic inside the database and schema is bound to the table



Dynamic Data Masking



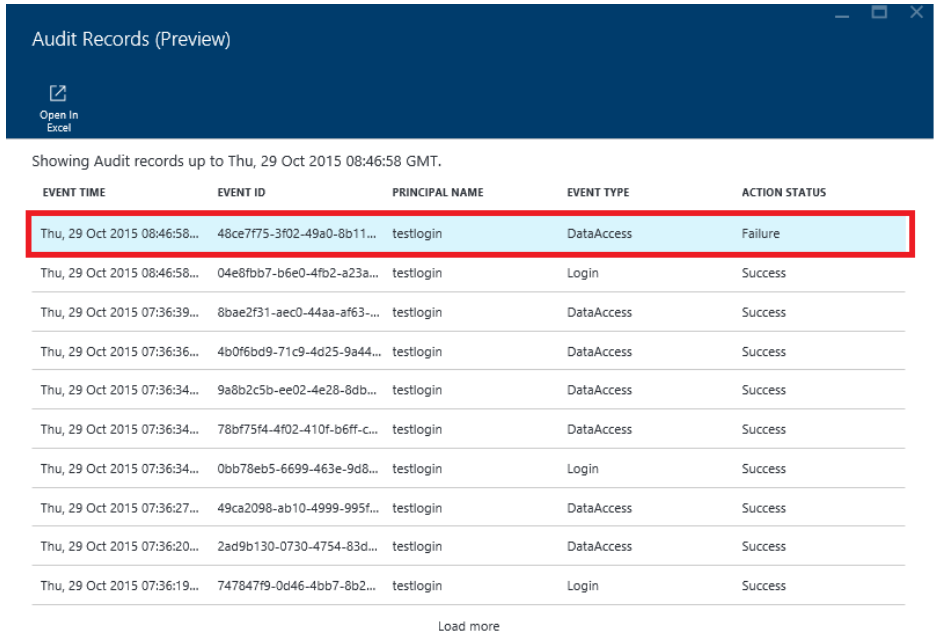
Dynamic Data Masking - Example

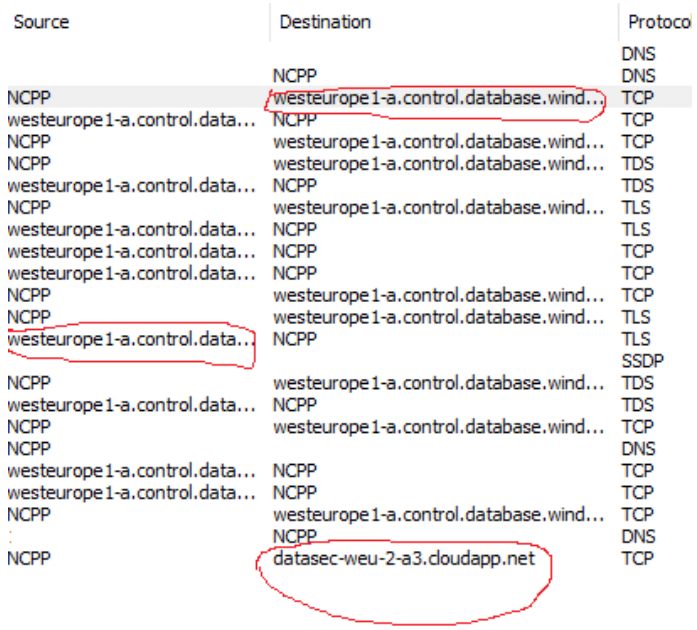
```
// Data Masking
CREATE TABLE Contacto
 (ID int IDENTITY PRIMARY KEY,
  Nombre varchar(100) MASKED WITH (FUNCTION = 'partial(1,"XXXXXXX",0)') NULL,
  Apellido varchar(100) NOT NULL,
  NrTlf varchar(12) MASKED WITH (FUNCTION = 'default()') NULL,
  Email varchar(100) MASKED WITH (FUNCTION = 'email()') NULL);
// Row Level Security
CREATE FUNCTION SecPred(@userId int)
RETURNS TABLE
WITH SCHEMABINDING
AS
RETURN SELECT 1 as valor WHERE @userId = user_id()
CREATE SECURITY POLICY [secpol] ADD FILTER PREDICATE
[dbo].[SecPred]([UserId]) on [dbo].[Protegido]
```

Microsoft

SQL Auditing

- Azure SQL Database Auditing tracks database events and writes audited events to an audit log in your Azure Storage account



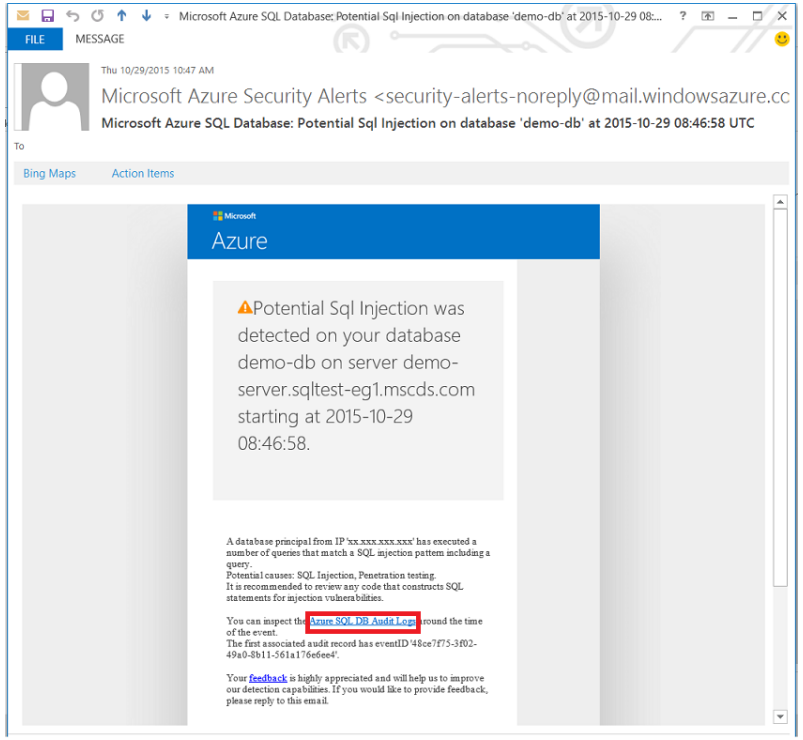


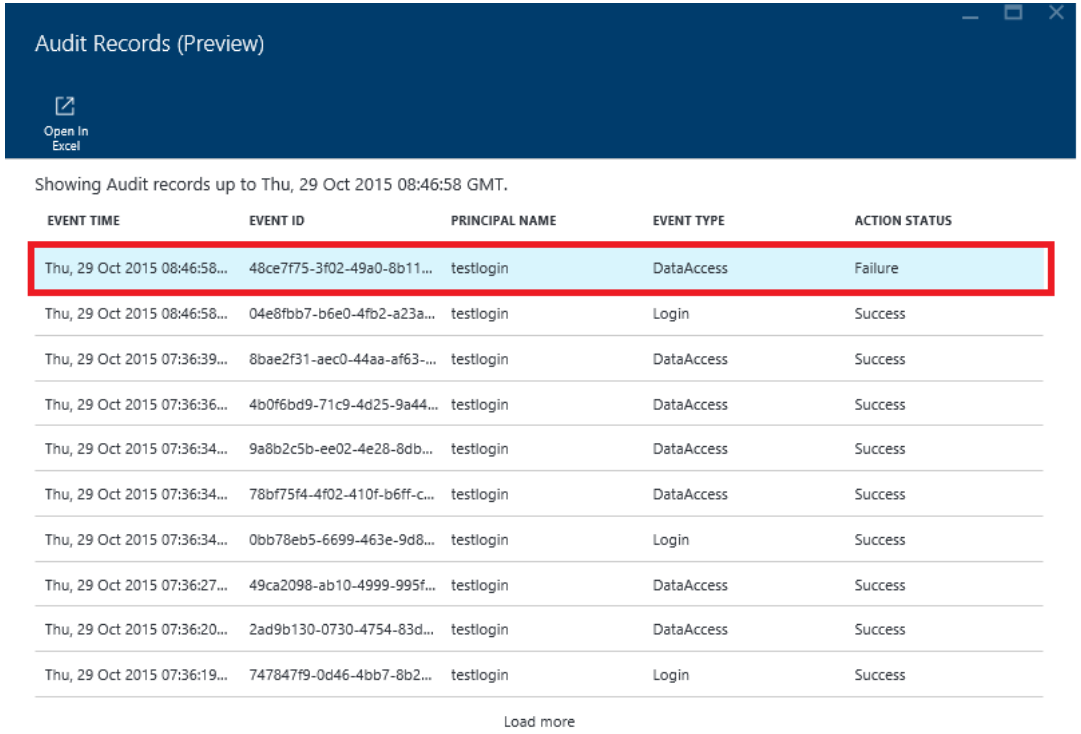
Events: Plain SQL,Parameterized SQL, Stored procedure, Logins,Transaction management

Microsoft

Thread Detection

- Threat Detection detects anomalous database activities indicating potential security threats to the database.





What is new using V12 Azure SQL Security

Demo: Row Level Security

Microsoft

SQL Azure Database V12

High Availability and Disaster Recovery

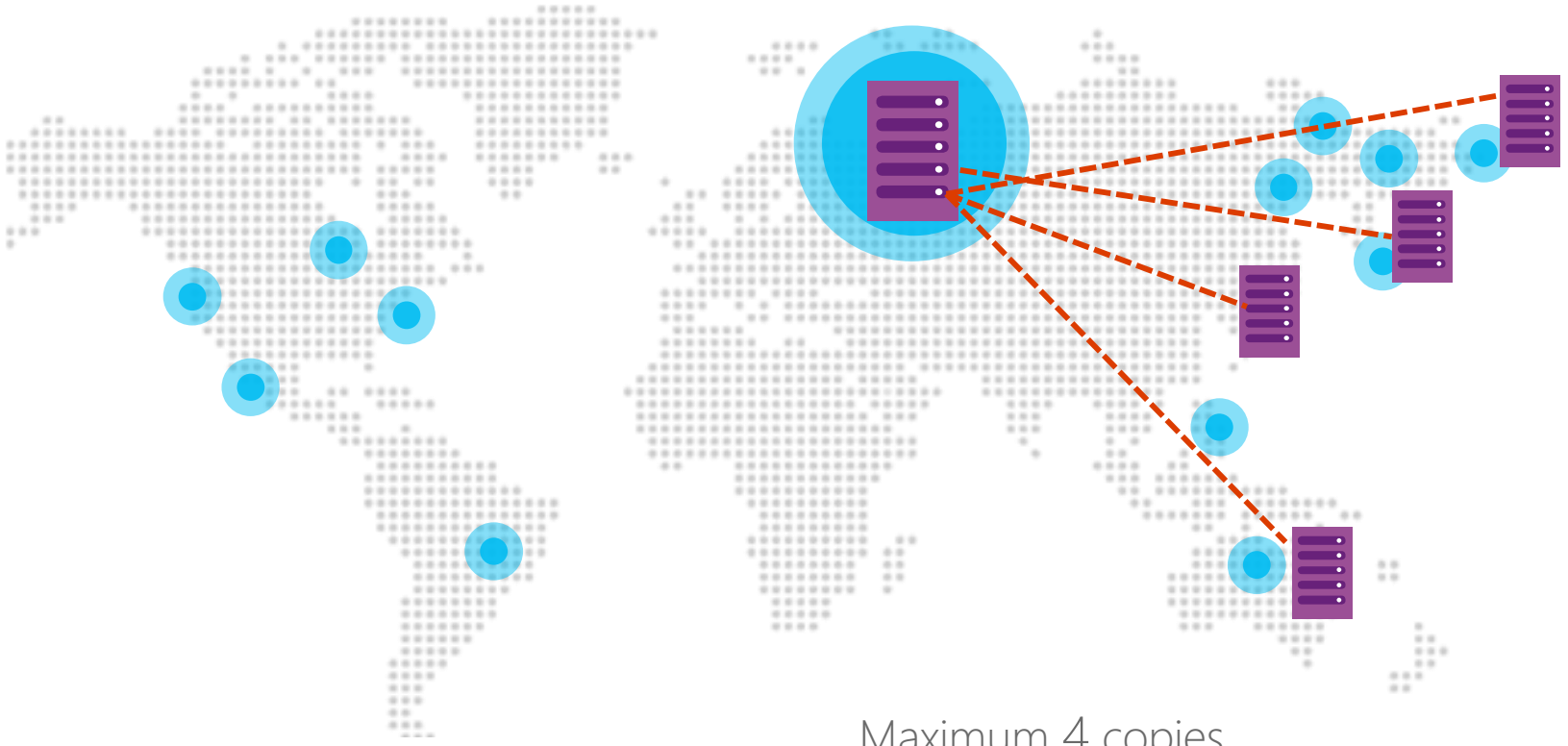
Microsoft

SQL Azure Database V12 – High Availability and Disaster Recovery

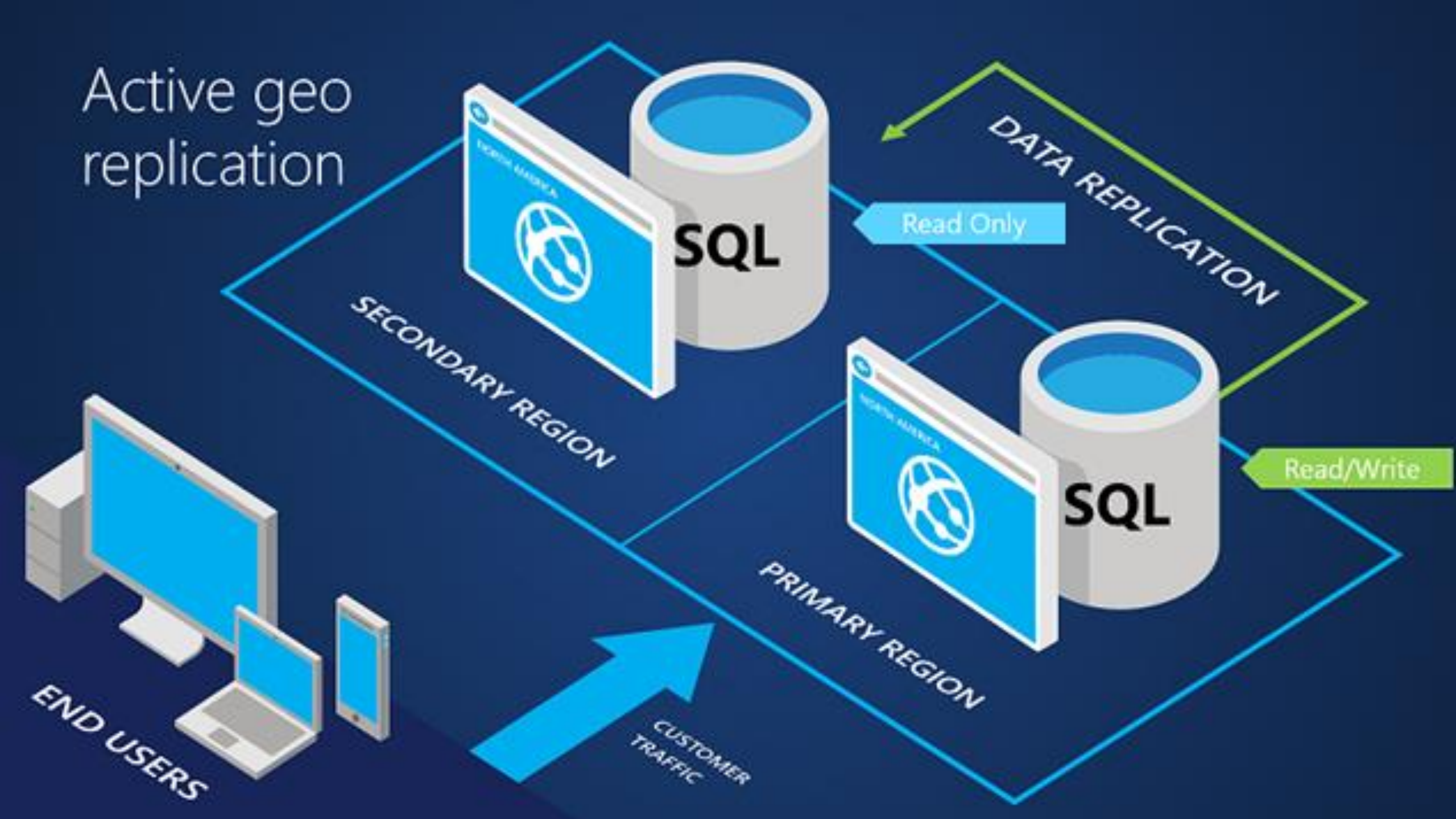
- ✓SQL Azure HA and DR is in our DNA.
- ✓Geo-Replication – more than 1 replica (Online). http://blogs.msdn.com/b/timomta/archive/2015/03/26/script-to-perform-azure-sql-premium-failover.aspx
- ✓Geo-Restore.
  - ✓All database backups are geo-replicated. SQL DB supports geo-restore – eg: any database can be restored anywhere in the world. https://azure.microsoft.com/en-us/documentation/articles/sql-database-business-continuity/
- ✓SQL DataSync is possible to use but not supported.
- ✓Replication between databases. Others: DBMoto

Microsoft

SQL Azure Database V12 – High Availability and Disaster Recovery



Maximum 4 copies


Active geo replication
DATA REPLICATION
Read Only
SQL
SECONDARY REGION
Read/Write
SQL
PRIMARY REGION
END USERS
CUSTOMER TRAFFIC

SQL Azure Database V12 – High Availability and Disaster Recovery

Demo: Implement Geo-Replication

Microsoft

SQL Azure Database V12

Elastic Database Pool

Microsoft

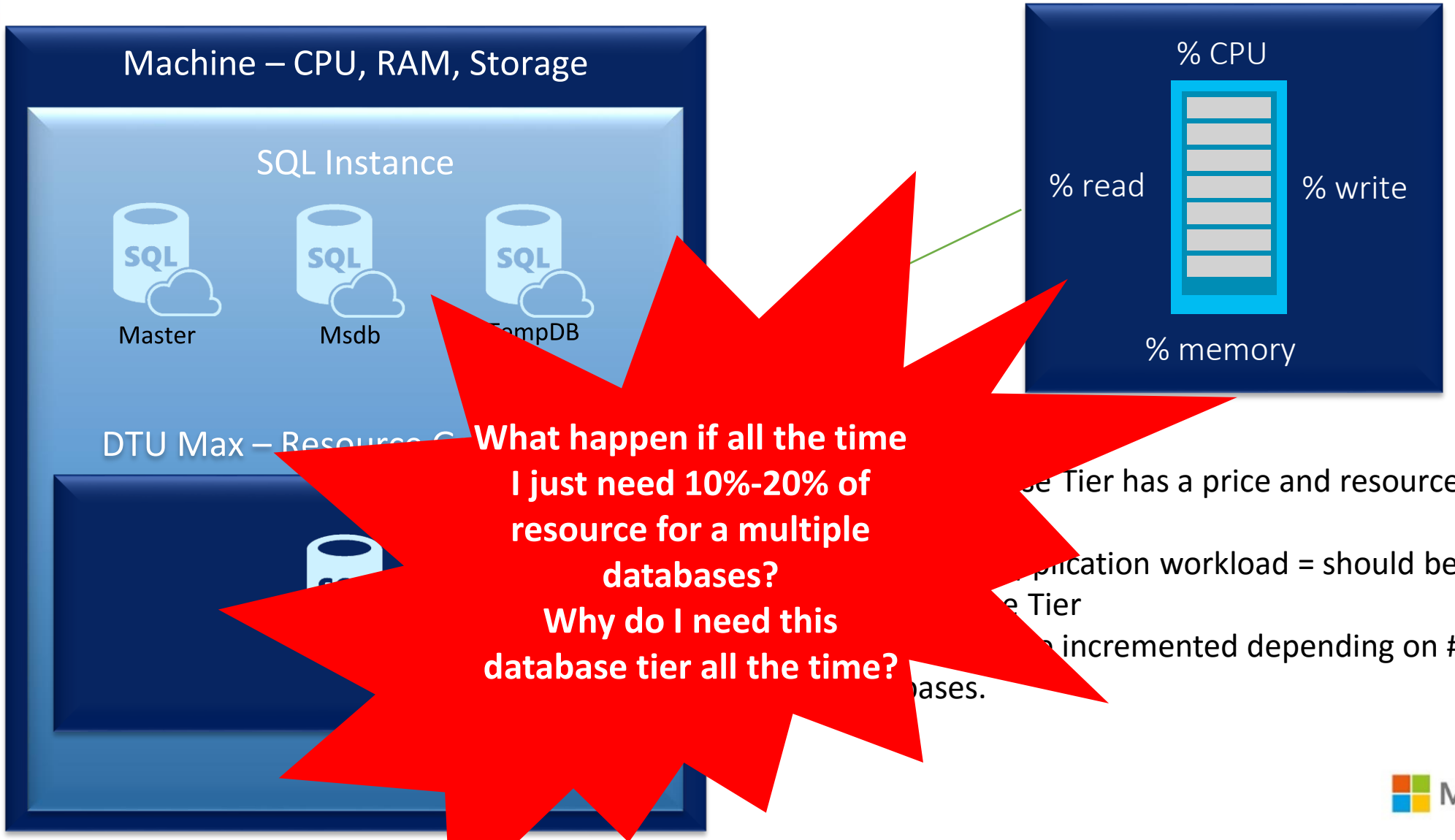
SQL Azure Database V12 - New Database Tiers

- ✓Basic/Standard vs Web/Business
- ✓Premium P6 y P11
- ✓Elastic Database Pool

	Basic	Standard				Premium				
		S0	S1	S2	S3	P1	P2	P4	P6/P3	P11
Maximum database size	2 GB	250 GB				500 GB				1 TB
DTUs	5	10	20	50	100	125	250	500	1,000	1,750
Point-in-time restore	Any point last 7 days	Any point last 14 days				Any point last 35 days				
Disaster recovery	Geo-Restore, restore to any Azure region	Standard Geo-Replication, offline secondary				Active Geo-Replication, up to 4 online (readable) secondary backups				
Max In-Memory OLTP storage	NA	NA	NA	NA	NA	1 GB	2 GB	3 GB*	8 GB	10 GB*
Max concurrent requests	30	60	90	120	200	200	400	800	1,600	2,400
Max concurrent logins	30	60	90	120	200	200	400	800	1,600	2,400
Max sessions	300	600	900	1,200	2,400	2,400	4,800	9,600	19,200	32,000

** In-Memory OLTP storage limits will soon adjust to 4 for P4 and 14 for P11.*

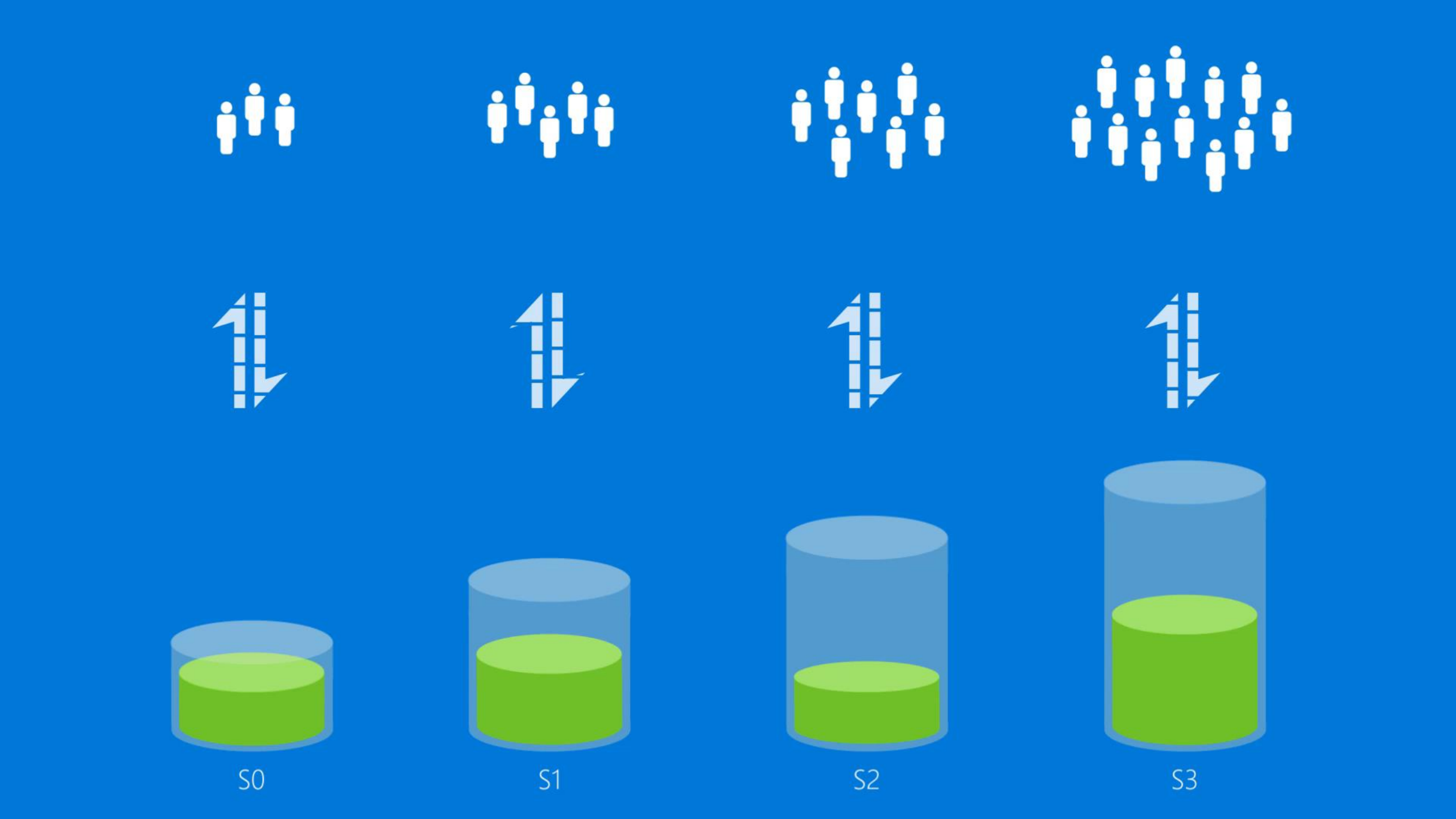
SQL Azure Database V12 - Single Database



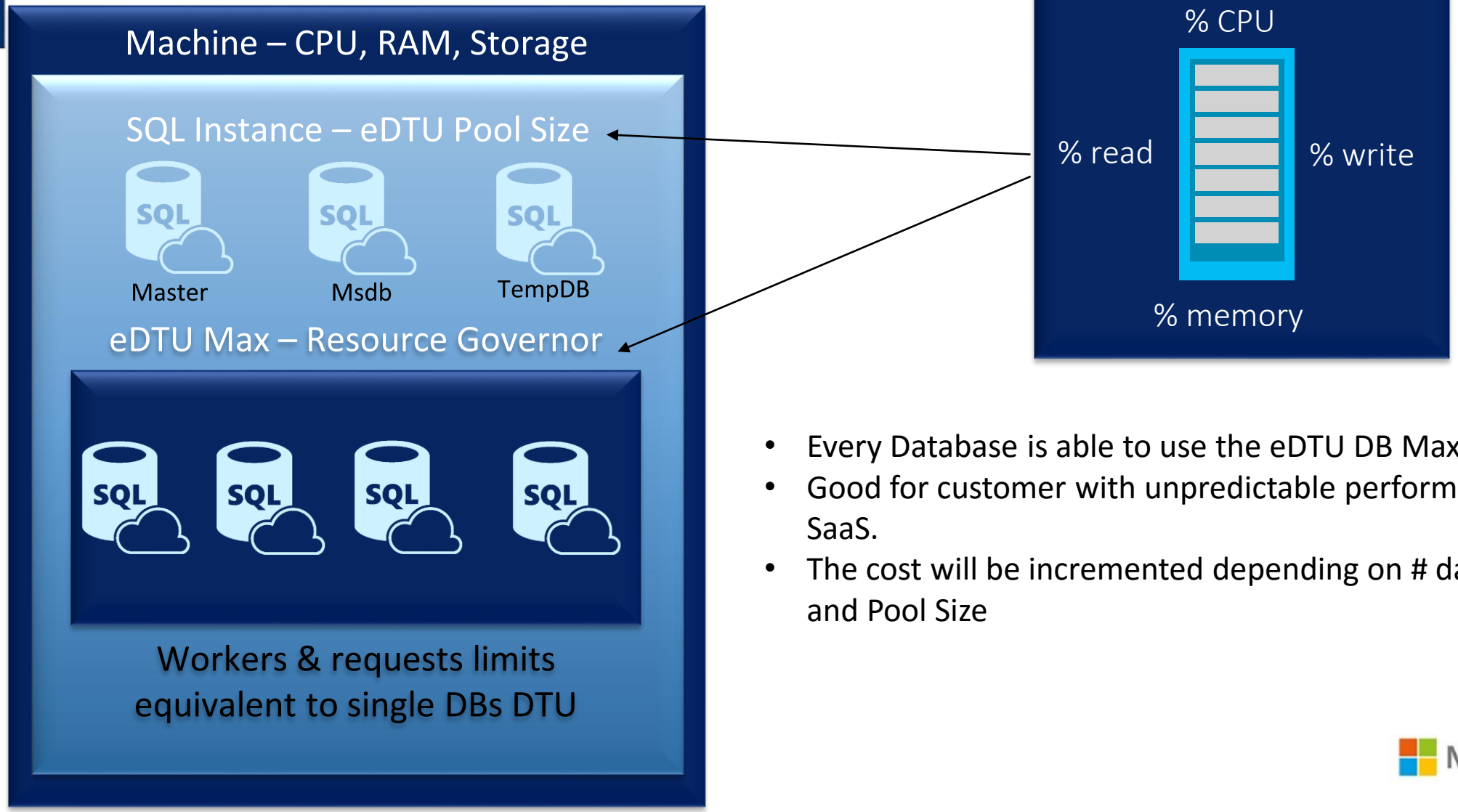
Tier has a price and resources

plication workload = should be
Tier

incremented depending on #
ases.


S0
S1
S2
S3

SQL Azure Database V12 – Elastic DB



- Every Database is able to use the eDTU DB Max.
- Good for customer with unpredictable performance or SaaS.
- The cost will be incremented depending on # databases and Pool Size


ELASTIC DATABASE POOL
MAX
100 DTUs
MIN
10 DTUs

SQL Azure Database V12 – Elastic Database

- A pool is given a set number of eDTUs, for a set price.

eDTUS consumption		
Individual databases are given the flexibility to auto-scale within set parameters.	Under heavy load a database can consume more eDTUs to meet demand.	Databases under light loads consume less, and databases under no load don't consume any eDTUs.

Provisioning resources for the entire pool rather than for single databases simplifies your management tasks. Plus you have a predictable budget for the pool.

Microsoft

SQL Azure Database V12 – Elastic Database - Think that…

Reconfiguration/failover impact

- Slow memory cached Warm-Up for 63 DBs running 200 eDTU/100 eDTU max

Microsoft

SQL Azure Database V12 – Elastic Database – Choosing Db Tier

Analysis From VM or On-premise

PSSDIAG for SQL Server
SQL Nexus.
Performance counters.
Reviewed instance configuration.
Azure SQL database calculator tool.
http://dtucalculator.azurewebsites.net/

Analysis from Single SQL DBs

Elastic pool creation experience.

Microsoft

SQL Azure Database V12 – Elastic Database

Demo: Using Dtu-Calculator

SQL Azure Database V12 – Elastic Database

Elastic Tools

Microsoft

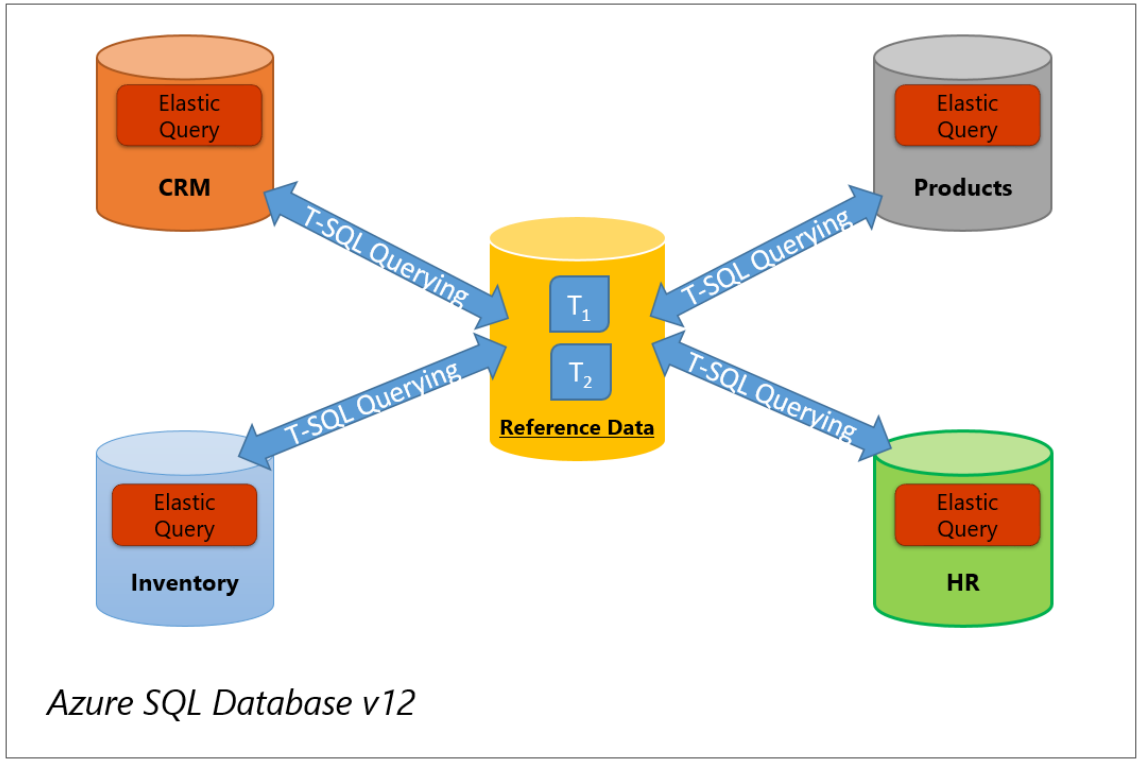
SQL Azure Database V12 – Elastic database query

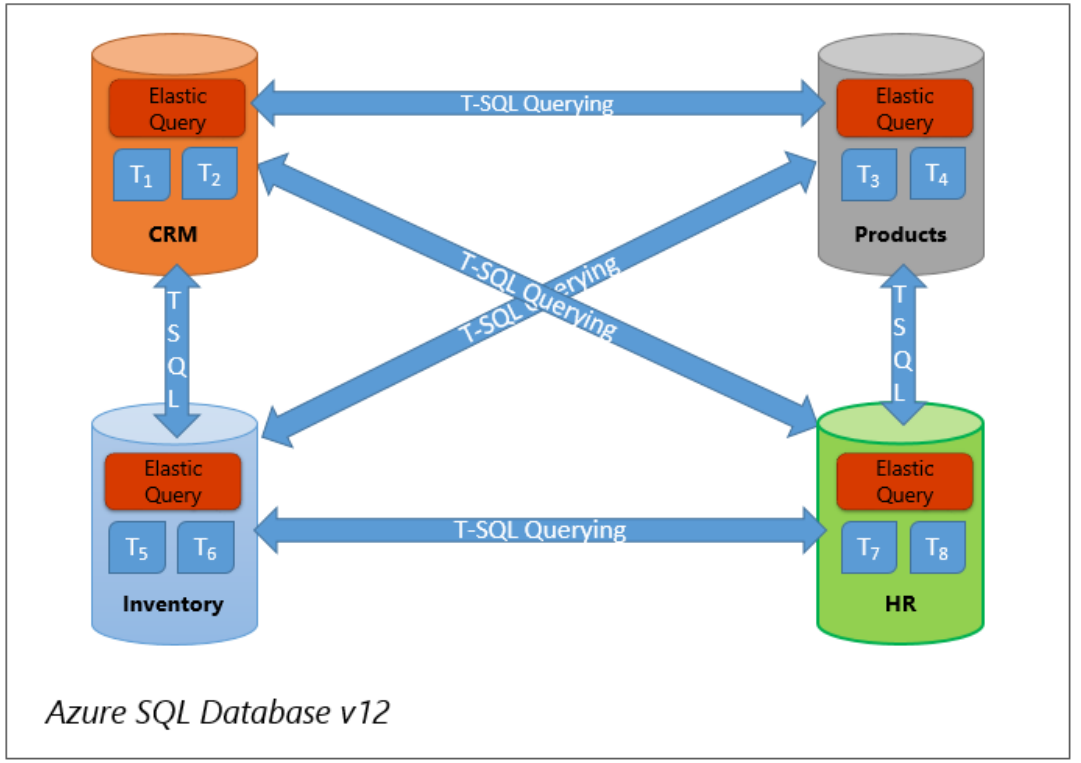
- ✓**Cross-database** query: read-only querying of remote databases.
- ✓**Migrate** applications: using three- and four-part names or linked server to SQL DB.
- ✓**Push SQL parameters:** to remote databases
- ✓**Sp_execute_fanout:** parameters similar to sp_executesql
  - Announcement blog post:
  - Step-by-step tutorial
  - Overview documentation

Microsoft

SQL Azure Database V12 – Elastic database query

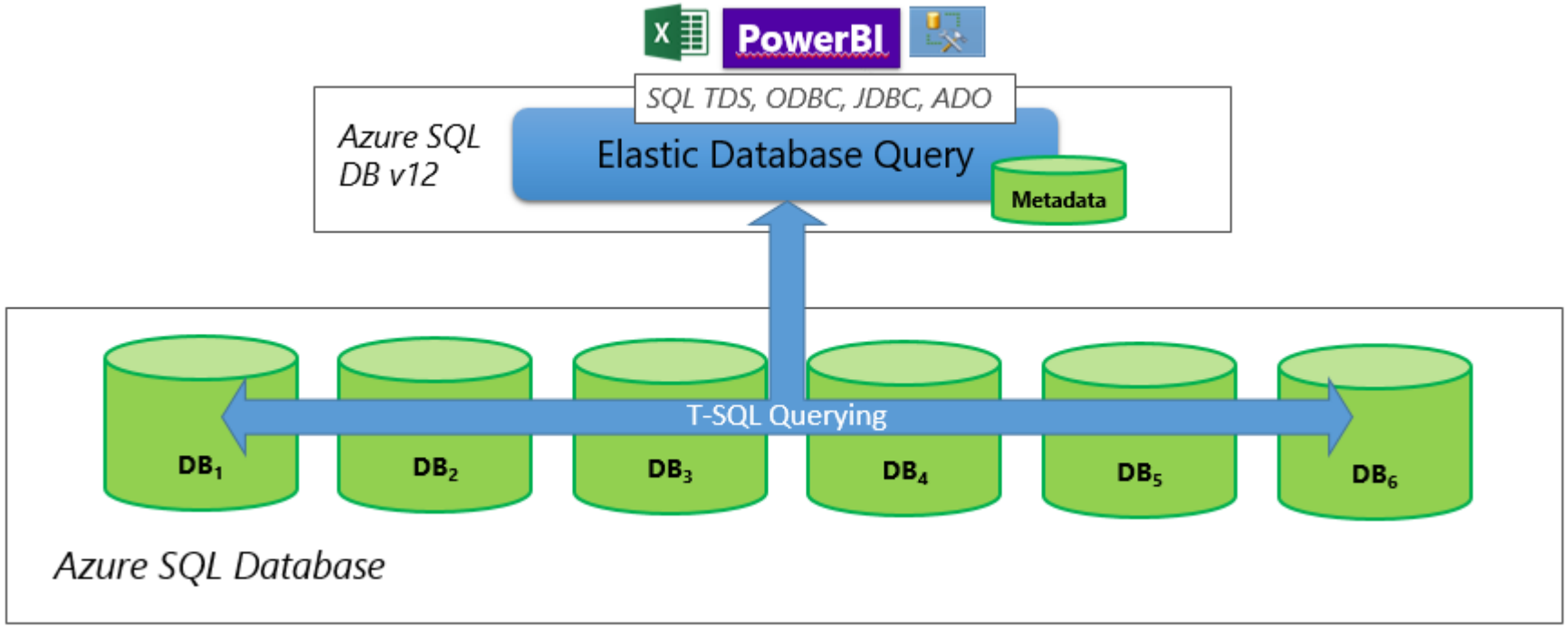
✓Vertical partitioning





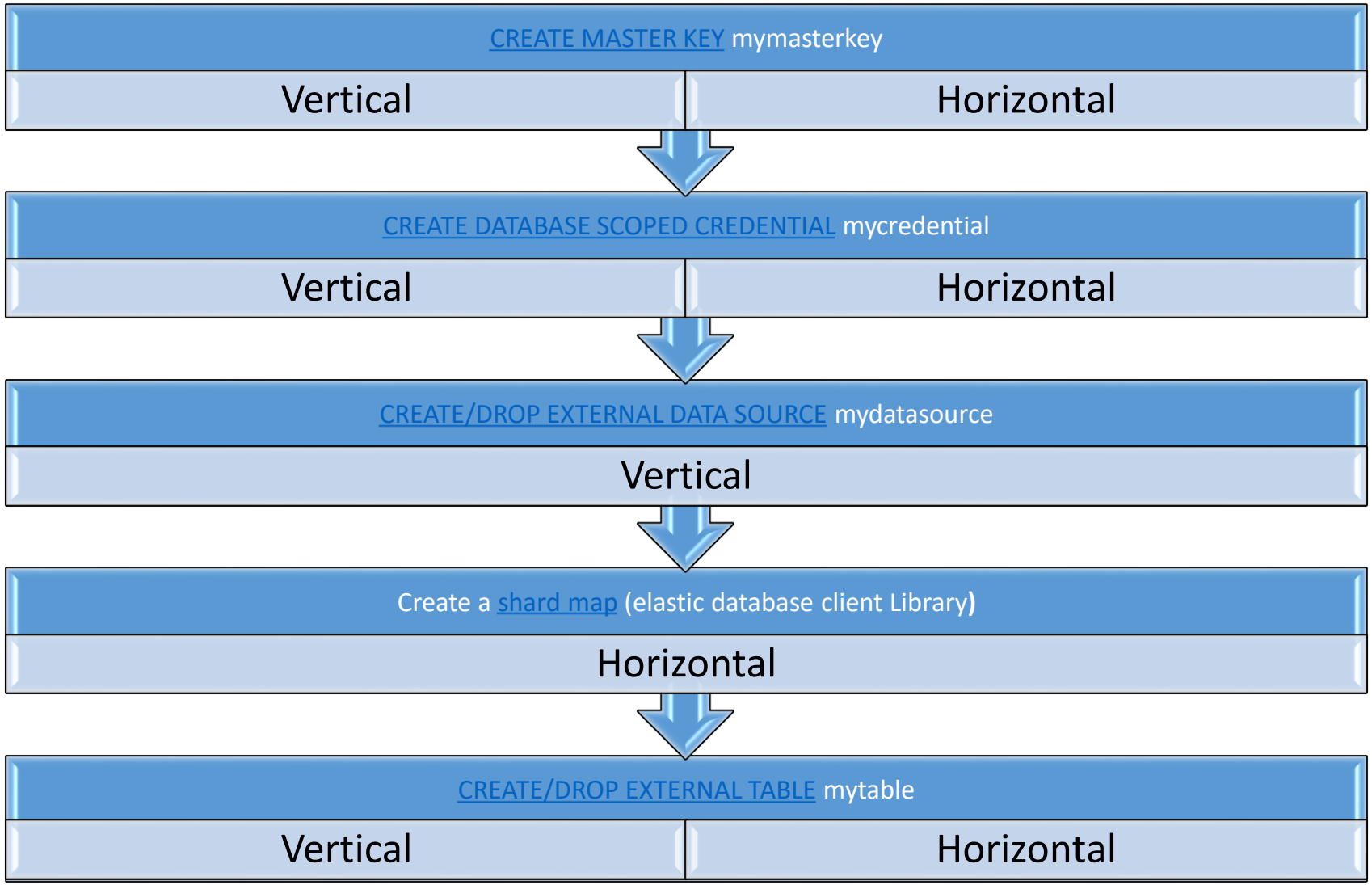
SQL Azure Database V12 – Elastic database query

- ✓Horizontal partitioning



Microsoft

SQL Azure Database V12 – Elastic database query



SQL Azure Database V12 – Elastic database query

Demo: Elastic Query

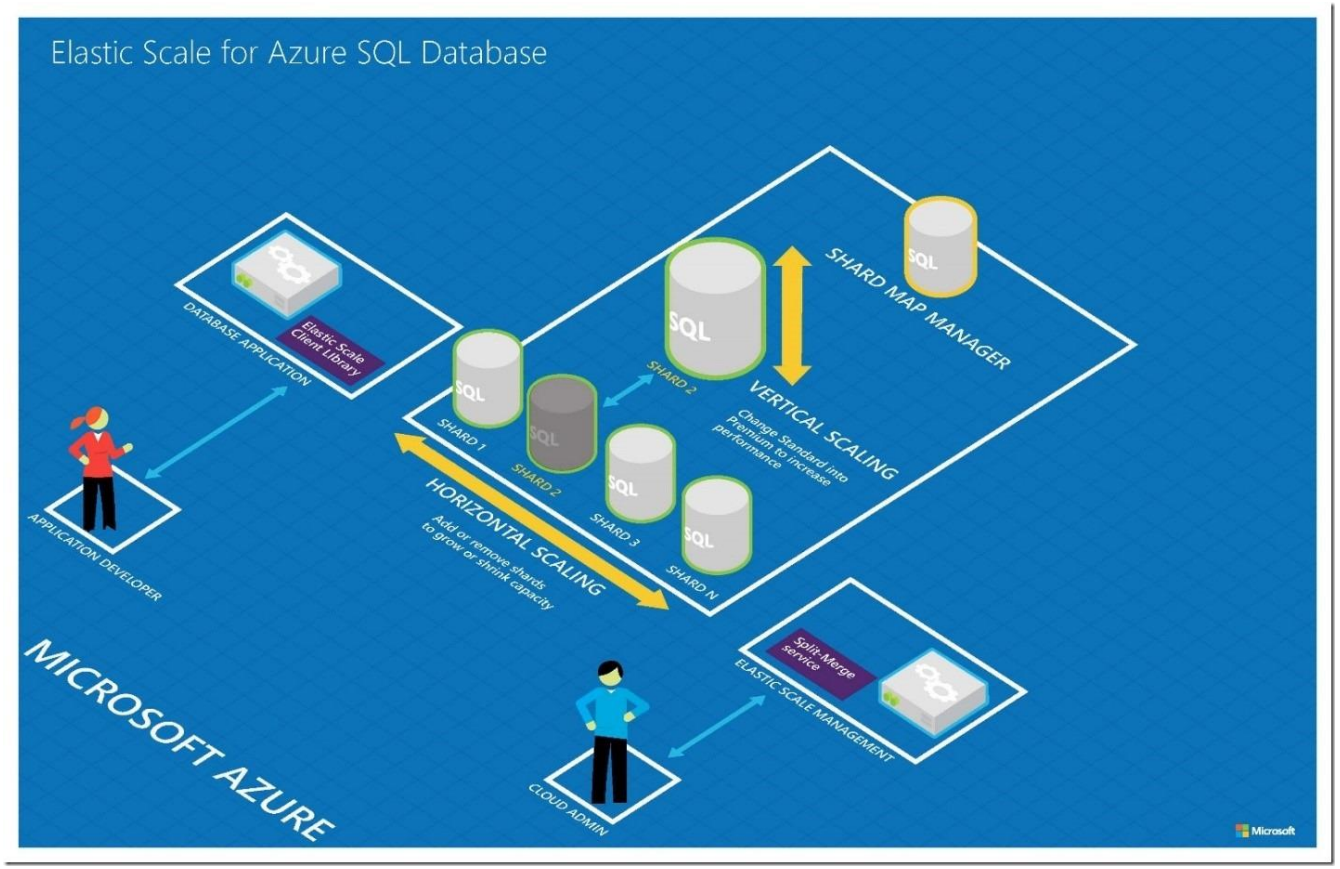
Microsoft

SQL Azure Database V12 - Elastic Scale

- ✓The new Elastic Scale capabilities simplify the process of scaling out (and in) a cloud application's data tier.
- ✓Elastic Scale offers functionality to help you build and maintain applications that can span a few databases or thousands.
- ✓**High-volume OLTP**: Single application process massive data volumes by employing #number of database shards.
- ✓**Multi-tenant SaaS**: Each customer is assigned their own database for higher isolation
- ✓**Continuous data collection**: New shards for new date ranges. Newer shards can use higher service tiers, and scale down over time as usage is reduced

Microsoft

SQL Azure Database V12 - Elastic Scale



SQL Azure Database V12 - Elastic jobs

- ✓Elastic Database jobs is a customer-hosted Azure Cloud Service that enables the execution of ad-hoc and scheduled administrative tasks, which are called jobs.
- ✓**Administrative**: Deploy new schema
- ✓**Update** data: product information common across all databases
- ✓**Schedule:** automate updates after hours
- ✓**Rebuild** indexes: regularly across a collection of databases
- ✓**Centralize** query results: from many databases into a central table
- ✓**Reference:** https://azure.microsoft.com/en-us/documentation/articles/sql-database-elastic-jobs-powershell/
- ✓**Service:** https://azure.microsoft.com/en-us/documentation/articles/sql-database-elastic-jobs-service-installation/
- ✓**Nuget:** https://www.nuget.org/packages/Microsoft.Azure.SqlDatabase.Jobs/

Microsoft

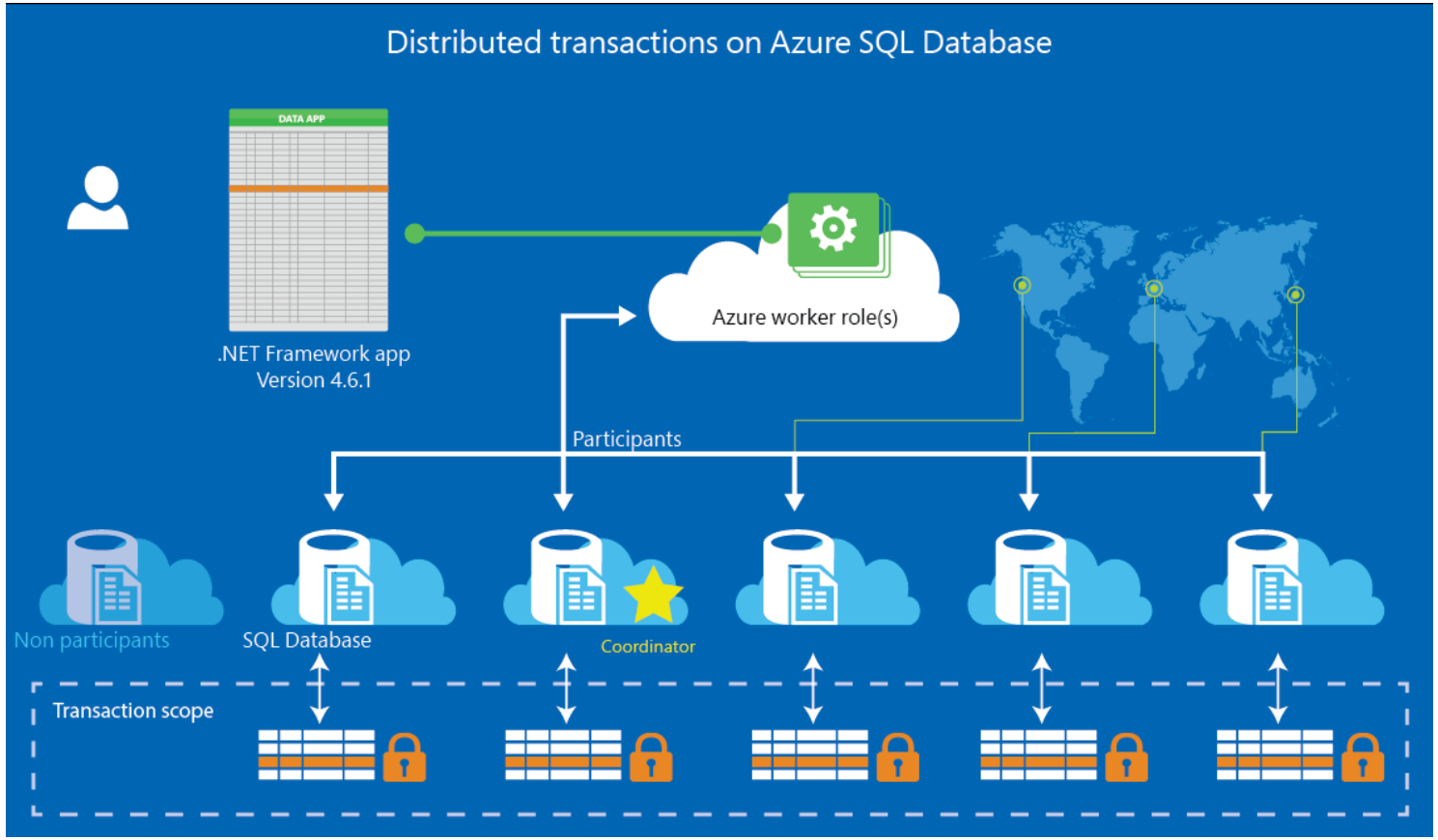
Elastic jobs

SQL Azure Database V12 - Elastic Database Transactions

- ✓ Elastic database transactions for Azure SQL Database (SQL DB) allow you to run transactions that span several databases in SQL DB
- ✓ Available for **.NET 4.6.1** applications: ADO .NET ->System.Transaction classes
- ✓ Coordinate distributed transactions is **directly integrated** into SQL DB As MSDTC is not an Azure PaaS.
- ✓ Applications: Connect to any SQL Database-> Launch distributed transactions-> one of the databases will coordinate **transparently**
- ✓ **Multi-database** applications: Vertically partition
- ✓ **Sharded database** applications: Horizontal partitioning
- ✓ **Only SQL DB supported**
- ✓ Not Other X/Open XA resource providers
- ✓ Not SQL Server
- ✓ Only **client-coordinated** transactions
- ✓ Only databases on **Azure SQL DB V12** are supported.

Microsoft

SQL Azure Database V12 - Elastic Database Transactions



SQL Azure Database V12 - Elastic Database Transactions

```
using (var scope = new TransactionScope())            {
            using (var conn1 = new SqlConnection(connStrDb1))
            {
               conn1.Open();
               SqlCommand cmd1 = conn1.CreateCommand();
               cmd1.CommandText = string.Format("insert into T1 values(1)");
               cmd1.ExecuteNonQuery();
            }
            using (var conn2 = new SqlConnection(connStrDb2))              {
               conn2.Open();
               var cmd2 = conn2.CreateCommand();
               cmd2.CommandText = string.Format("insert into T2 values(2)");
               cmd2.ExecuteNonQuery();
            }
            scope.Complete();
```

Microsoft

SQL Azure Database V12 - Elastic Database Transactions

Demo: Elastic Transaction

Microsoft

SQL Azure Database V12

Connectivity

Microsoft

What is new using V12 Azure SQL Connectivity

.Net Framework

[.NET 4.5.1 or above: ConnectRetryCount and ConnectRetryInterval]

[.NET 4.6.1, retry logic implemented at sqlConnection.Open]

[Entity Framework 6.0: SqlAzureExecutionStrategy]

To measure

[SQLConnection object properties]

[Stopwatch]

[EF: Optimizar query performance and modelo]

Tips & Tricks

Documentation on handling connections to Azure SQL Db: https://azure.microsoft.com/en-us/documentation/articles/sql-database-connect-central-recommendations

Retry logic for Azure SQL Db: http://social.technet.microsoft.com/wiki/contents/articles/4235.retry-logic-for-transient-failures-in-windows-azure-sql-database.aspx

Error Code List for Retry Logic in Azure SQL Db: https://azure.microsoft.com/en-us/documentation/articles/sql-database-develop-error-messages/

SqlConnection.ClearPool(cnn) - To clean up the connection from the connection pool using SqlConnection.ClearPool(cnn) - http://blogs.msdn.com/b/bartr/archive/2010/06/18/sql-azure-connection-retry.aspx

New connectivity method

Direct connection, not proxy

Differences depending on data provider.

Microsoft

SQL Azure Database V12 - StopWatch and SQLConnection

```
//Call StopWatch function
Stopwatch stopWatch = new Stopwatch();
            stopWatch.Start();
            stopWatch.Stop();
 //Elapsed time TimeSpan.
            TimeSpan ts = stopWatch.Elapsed;
//Obtained the time spent and data from SQLConnection property.
IDictionary currentStatistics = awConnection.RetrieveStatistics();
 long bytesReceived = (long)currentStatistics["BytesReceived"];
            long bytesSent = (long)currentStatistics["BytesSent"];
            long selectCount = (long)currentStatistics["SelectCount"];
            long selectRows = (long)currentStatistics["SelectRows"];
```

Microsoft

SQL Azure Database V12 – Connectivity

Demo: SQL Connection Properties

Microsoft

SQL Azure Database V12 – Retry Policy

- ✓ Caches physical connections with the same conn string in the process
- ✓ Reduces the cost of opening/closing operations
- ✓ If a connection fails to connect, it will be dropped from the pool
- ✓ What if a connection succeed to connect, but fails to execute commands?
  - ✓ It may remain in the pool for a long time and create additional issues
- ✓ Workaround: after multiple failed retries (e.g. 3) on the same command execution, clean up that connection from the pool
  - ✓ SqlConnection.ClearPool(cnn) - **To clean up the connection from the connection pool using SqlConnection.ClearPool(cnn)** - http://blogs.msdn.com/b/bartr/archive/2010/06/18/sql-azure-connection-retry.aspx

Microsoft

SQL Azure Database V12 – Retry Policy

- ✓Open connection late and close asap (especially disposable resources)
- ✓Leverage connection pooling
- ✓Retrieve only the data you need
- ✓Cache metadata and data where possible, avoid metadata retrieval at runtime
- ✓If possible, batch SQL statements together
- ✓Use Stored Procedures especially to encapsulate complex data access logic and transactional behaviors, this will naturally reduce round trips and resource locking on server side
- ✓Leverage Table Value Parameters
- ✓Avoid application based transaction management
- ✓Keep transactions as short as possible, avoid to involve unnecessary resources, use appropriate isolation level

Microsoft

SQL Azure Database V12 – Retry Policy

```
int retryIntervalSeconds = 10;
bool returnBool = false;
  for (int tries = 1; tries <= 5; tries++)
  {
  try
     {
       if (tries > 1)
                {
                        H.Thread.Sleep(1000 * retryIntervalSeconds);
                  retryIntervalSeconds = Convert.ToInt32(retryIntervalSeconds * 1.5);
                  C.SqlConnection.ClearAllPools();
                }
                sqlConnection.ConnectionString = clsConexion;
                sqlConnection.Open();….
              catch (C.SqlException sqlExc)
              {
                if (sqlExc.Number == 4060 || sqlExc.Number == 40197….
```

Microsoft

SQL Azure Database V12

New Tools

Microsoft

SQL Azure Database V12 - Common Performance Issues

- **Performance issues**
  - ✓Query tuning, investigating execution plans.
  - ✓Query Timeouts.
  - ✓Blocking.
  - ✓Deadlocks?.
  - ✓Database Tier in Azure SQL Database

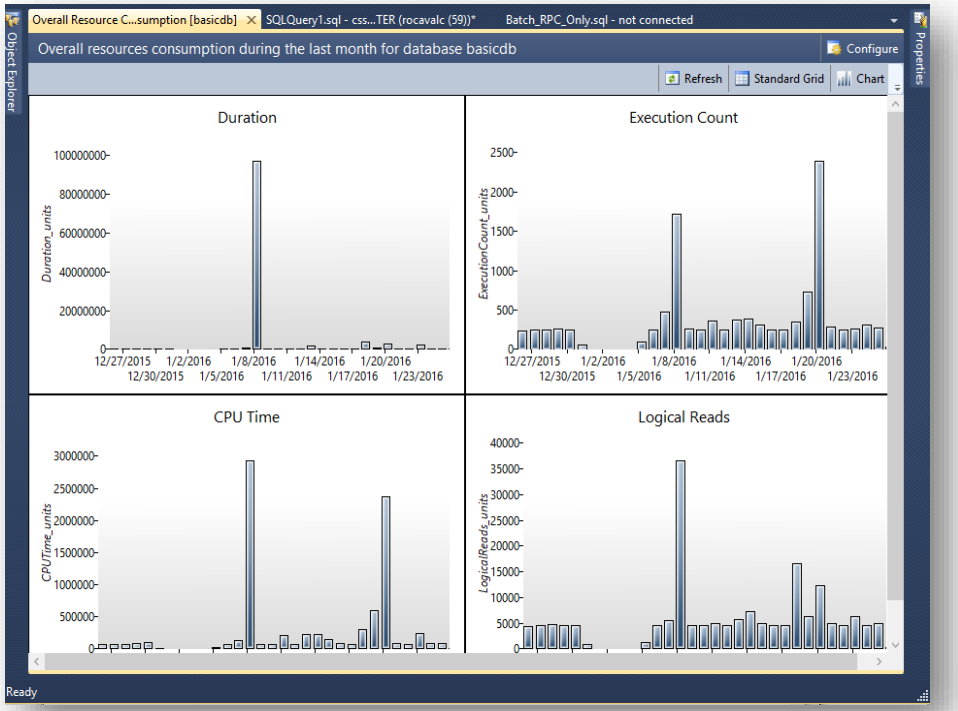
- **Frequently Simple**
  - ✓Outdated statistics.
  - ✓Missing Indexes.
  - ✓Throttling for Azure SQL Database.

Way to resolve

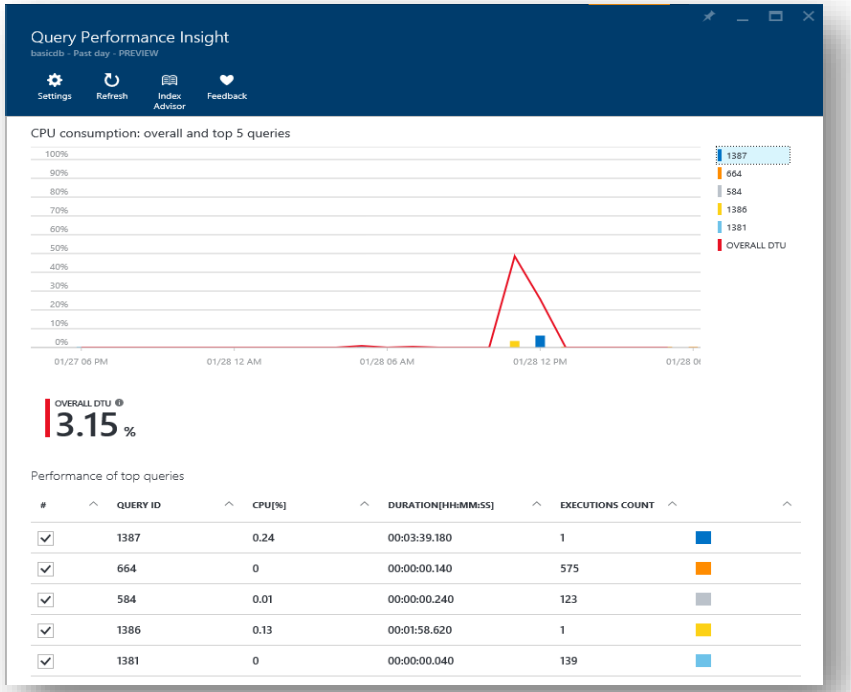
- ✓ Blocking - SDP, SQL Nexus
- ✓ Attentions, Errors, etc. – Xevents.
- ✓ Query behavior – QDS, DMVs

Microsoft

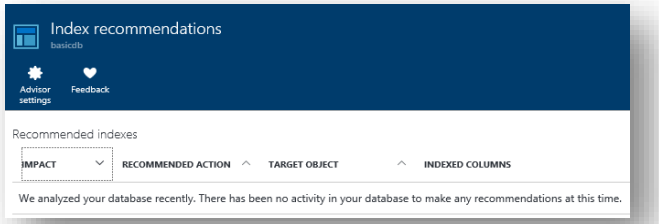
What is new using V12 Azure SQL Tools



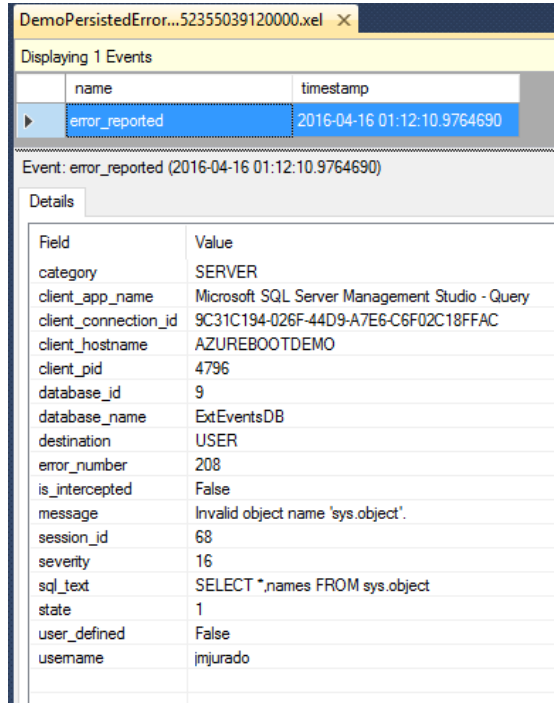
Query Data Store



Query Performance Insights



Index Advisor

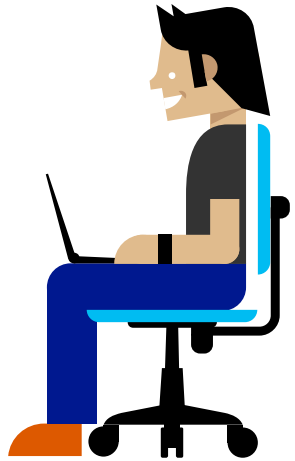


Extended Events

SQL Azure Database V12 - Query Data Store

- ✓ **It is a *persisted* database with *query execution information for***
  - ✓ SQL Server 2016 (Preview)
  - ✓ Azure SQL DB V12

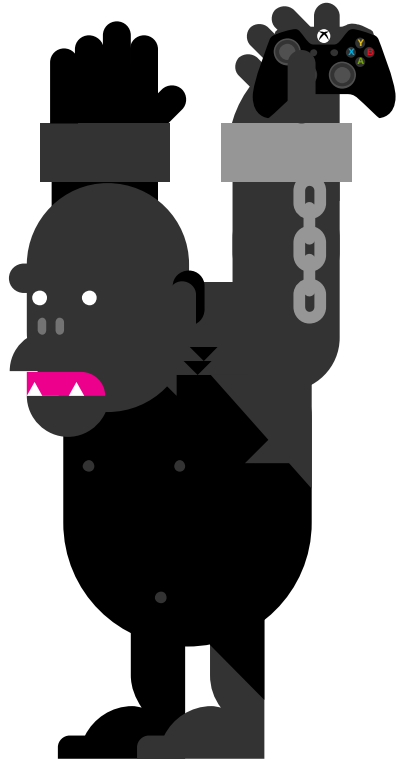
- ✓ **Query Performance Tuning and Troubleshooting**
  - ✓ SQL Profiler replacement in some parts.
  - ✓ Minimum impact for SQL Engine in SQL Server or Azure SQL DB.
  - ✓ Other tools like Query Performance Insight, Index Advisor are using QDS Information.
  - ✓ SSMS Reports Supported for SQL Server and Azure SQL Database.
  - ✓ Catalog Views all available.
  - ✓ Active or disable this feature demand.
  - ✓ Use Extended Events to capture the information.



Microsoft

SQL Azure Database V12 - Information collected

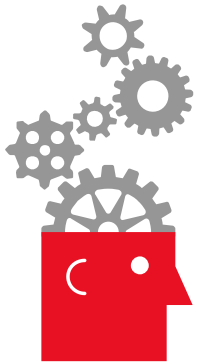
- ✓ ***A Query Flight Recorder to store:***
  - ✓ Queries/statements, Query Plans, Runtime statistics with connection properties.
  - ✓ Maintains the information even an instance restart.
  - ✓ Just save DML: SELECT, DELETE, INSERTE, UPDATE.
  - ✓ Doesn't record details about: DDL: BULK INSERT or DBCC commands.
  - ✓ The information will be recorded with a limited time.
  - ✓ Details of queries whether they are cached or not.
  - ✓ Analyze the history
  - ✓ Look at properties and statistics for queries not available in DMVs
  - ✓ Allows a simpler and more robust method to force plans vs plan guides
  - ✓ Independent of what is cached
  - ✓ Statement text appears in parameterized form
  - ✓ Including cursors and SET assignment
  - ✓ Persistence is async
  - ✓ Works with In-Memory tables in Azure SQL DB and SQL SERVER 2016.
- ✓ ***Other Information***
  - ✓ Use the scenarios page
  - ✓ The docs. Start here
  - ✓ Query Store Best Practices



Microsoft

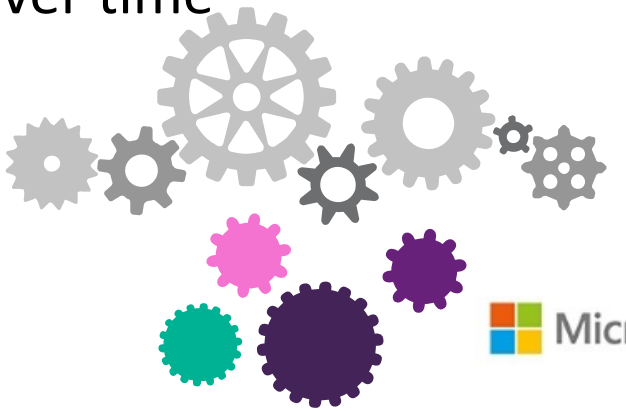
SQL Azure Database V12 - CAPTURE MODE

- **Query Capture Mode**
  - ✓ Query Store supports adjustable query capture.
    - ✓ The goal of capture option is to provide users with additional control over ingress of new queries with main value prop to enable collection of data that is highly relevant.
  - ✓ This is the list of supported values:
    - ✓ ALL - all queries are captured. Infrequent queries and queries with insignificant compile and execution duration are ignored. Thresholds for execution count, compile and runtime duration are internally determined.
    - ✓ AUTO - capture relevant queries based on execution count and resource consumption. This is the default configuration value
    - ✓ NONE: stop capturing new queries. Query Store will continue to collect compile and runtime statistics for queries that were captured already.
  - ✓ To change capture mode use one of the following statements:
    - ✓ ALTER DATABASE <database_name> SET QUERY_STORE (QUERY_CAPTURE_MODE = AUTO);
      - ✓ **Query capture mode** of AUTO (Default is ALL) = capture relevant queries based on execution count and resource consumption. NONE = Stop capture new queries but track and persist stats for existing.
    - ✓ ALTER DATABASE <database_name> SET QUERY_STORE (QUERY_CAPTURE_MODE = NONE);
      - ✓ **Sized based cleanup** of AUTO (Default is OFF) = Remove oldest queries and least expensive when 90% of max (until 80% reached). Overrides time retention (days kept in store)
  - ✓ Query capture = AUTO; Query retention = 30 days; Sized based cleanup = ON
  - ✓ We do limit memory for Query Store and may revert to read-only mode memory overcommitted or when the disk is full.



SQL Azure Database V12 - QDS vs DMVs

- ✓ **Using sys.query_context_settings**
  - ✓ set_options
  - ✓ cursor_options - Go here to interpret bitmask
  - ✓ schema_id
- ✓ **Comparing DMVs vs Query Store**
  - ✓ Compilation statistics
  - ✓ Plan properties including engine version and database compatibility level
  - ✓ Details and stats for plan variations used for a query over time
  - ✓ Recorded as queries are compiled
  - ✓ Record query execution stats ***even if aborted***



Dynamic Management Views

Plan store

Runtime stats

sys.database_query_store_options	• Exposes various Query Store configuration options for user database.
sys.query_store_query_text	• Exposes information about captured query text for a query
sys.query_context_settings	• Exposes information about content settings
sys.query_store_query	• Exposes information about queries in every execution
sys.query_store_plan	• Exposes information about different plans SQL Server used to execute queries in the system.
sys.query_store_runtime_stats_interval	• Returns all intervals created in Query Store
sys.query_store_runtime_stats	• Returns runtime statistics for executed query plans, aggregated on per-interval basis.

Microsoft

Dynamic Management Views

Force plan	• **sp_query_store_force_plan @query_id, @plan_id**
Stop plan forcing	• **sp_query_store_unforce_plan @query_id, @plan_id**
Clear all data for a single query	• **sp_query_store_remove_query @query_id**
Remove a single plan	• **sp_query_store_remove_plan @plan_id**
Clear all stats	• **sp_query_store_reset_exec_stats @plan_id**
Flush data	• **sp_query_store_flush_db**

Microsoft

SQL Azure Database V12 - Maintenance

- Enabling, Clearing, and State
  - ✓ **ON** = Enable; **OFF** = Disabled (existing state and data kept); **CLEAR** = TRUNCATE tables
  - ✓ **READ_WRITE** = Default when ON; **READ_ONLY** = intentional or *problem* (desired != actual)
- Size, limits, and retention
  - ✓ Default **max size** = 100Mb (limited by overall database size). If you hit max state = READ_ONLY
  - ✓ Default **max plans per query** = 200
  - ✓ Default **days queries kept in store** = 367 days

Azure SQL Database – Create First DB

Demo: Elastic Query

Microsoft

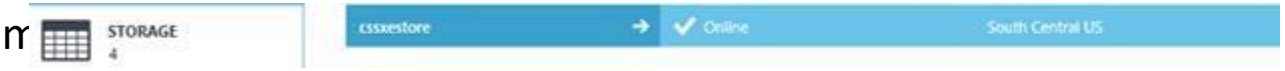
SQL Azure Database V12 - Extended Events

- You have the documentation in this URL
- In order to use Extended Events with your SQL Database server there are three basic steps:

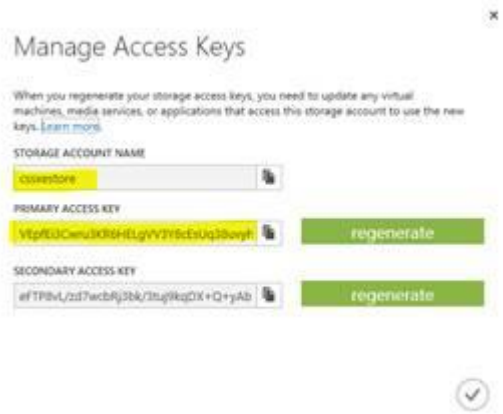
    - Setup a storage account and container
    - Enable access to storage from SQL Database
    - Collect Data using Extended Events

Setup storage account and container

XEvents use blob storage as their target. You shall need access to a storage account that will be the target for writing out the XEL files. Typically, this shows up under the Storage node in Management Portal. For exam



After storage is created, note the "Storage Account Name" and the "Primary Access Key". Both will be needed to access the .xel output files within the storage.



SQL Azure Database V12 - Extended Events

- Enable access to storage from SQL Database.

- You shall use a tool called Azure Storage Explorer to download the XEL files from the container to your local machine. Download and install Azure Storage Explorer (http://azurestorageexplorer.codeplex.com/ on your client machine.
- We used the current version (Version 6) as of this writing; feel free to use the latest version if another one is available.

    - Here are the steps to do for the same:

        - Configure Azure Storage Explorer to point to your storage account using the "Storage Account Name" and "Storage Account Key" from previous section titled Setup storage account and container. Check the Use HTTPS" checkbox and then click on Test Access

SQL Azure Database V12 - Extended Events

- The “Test Access” button click should return a success message. Click Save to save the configuration

You should see a similar screen to this once you are done

The next step is to add a container within the storage account. Click Blob Containers and add a new container

In my case I created a container named xe_container (with access level=off)

SQL Azure Database V12 - Extended Events

In my case I created a container named xe_container (with access level=off)

that show up like this

Next click on the specific container and then click on Security

Go to Shared Access Signatures tab and choose Read and Write Permission and ensure that End Date is well in future so the access does not expire before data collection is done.

SQL Azure Database V12 - Extended Events

Go to Shared Access Signatures tab and choose Read and Write Permission and ensure that End Date is well in future so the access does not expire before data collection is done.

Select the "Generate Signature" button, then copy the output to notepad

The output of this step should be a SAS key for the container that looks like this:

http://your_storage_account.blob.core.windows.net/your_container_name?sr=c&sv=start_date&sr=c&sig=%signature

SQL Azure Database V12 - Extended Events

- Collecting Data using Extended Events

Open the SQL SERVER Management Studio and follow up the following instructions:

- Create a master key specifying a strong password

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'xxxxxx!';
```

- Define the blob storage where the XEL will be save. Use the SAS key that was provided by the definition of the blob storage.

```
CREATE DATABASE SCOPED CREDENTIAL [https://xxxx.blob.core.windows.net/xe-container]
WITH IDENTITY='SHARED ACCESS SIGNATURE',
SECRET = 'sv=2014-02-14&sr=c&sig=Hz2n9vs%3D&st=2016-01-25T23%3A00%3A00Z&se=2016-02-02T23%3A00%3A00Z&sp=rw'
```

- Define the Extended Event. The filter result <> 0 will capture when the error gives an error.

```
CREATE EVENT SESSION
ON DATABASE
ADD EVENT
    ACTION
pid
WHERE
  ADD TARGET
     SET filename='https://xxx.blob.core.windows.net/xe-container/DemoPersistedEvents.xel')
```

- Start the event and wait to reproduce the issue.

```
ALTER EVENT SESSION                ON DATABASE STATE
```

- Once the issue has been reproduced, stop the event.

```
ALTER EVENT SESSION [              ON DATABASE STATE = STop;
```

- You could delete it when you don't need it anymore.

```
DROP EVENT SESSION                 ON DATABASE
```

- You should see XEL files in the storage container in Azure Storage Explorer
- You can then download to your laptop/local machine
- (Note that the files are in the format server_name_database_name_session_name.– so you can tell which server & database they were collected from)

Microsoft

Azure SQL Database – Create First DB

Demo: Elastic Query

Microsoft

SQL Azure Database V12 - DMVs

- SELECT end_time , (SELECT Max(v) FROM (VALUES (avg_cpu_percent) , (avg_data_io_percent) , (avg_log_write_percent)) AS value(v)) AS [avg_DTU_percent] FROM sys.dm_db_resource_stats ORDER BY end_time DESC;

- Tip:To find a resource usage:
  - sys.dm_db_resource_stats. Has 15 second granularity.
  - sys.resource_stats has 5 minute granularity.
  - sys.elastic_pool_resource_stats for elastic pool databases.
  - sys.dm_os_performance_counters - http://blogs.msdn.com/b/dfurman/archive/2015/04/02/collecting-performance-counter-values-in-sql-azure.aspx

Microsoft

SQL Azure Database V12 - Engine - Tips &Tricks

- Deadlock:

```
SELECT *,   CAST(event_data as XML).value('(/event/@timestamp)[1]',
'datetime2') AS timestamp ,CAST(event_data as
XML).value('(/event/data[@name="error"]/value)[1]', 'INT') AS error
,CAST(event_data as XML).value('(/event/data[@name="state"]/value)[1]', 'INT')
AS state   ,CAST(event_data as
XML).value('(/event/data[@name="is_success"]/value)[1]', 'bit') AS is_success
,CAST(event_data as
XML).value('(/event/data[@name="database_name"]/value)[1]', 'sysname') AS
database_name
FROM sys.fn_xe_telemetry_blob_target_read_file('el', null, null, null)
   where object_name = 'database_xml_deadlock_report'
```

Microsoft

SQL Azure Database V12 - Code Best Practices – Tips & Tricks

- Batching considerations: https://azure.microsoft.com/en-us/documentation/articles/sql-database-use-batching-to-improve-performance/
- Performance considerations: https://azure.microsoft.com/en-us/documentation/articles/sql-database-performance-guidance/
- To avoid any parameter sniffing issue. http://blogs.technet.com/b/mdegre/archive/2012/03/19/what-is-parameter-sniffing.aspx

Microsoft

SQL Azure Database V12 - Entity Framework - Tips & Tricks

- **Link describing Performance Considerations with Entity Framework – with regards to optimize query performance and model.** http://msdn.microsoft.com/en-us/library/cc853327.aspx

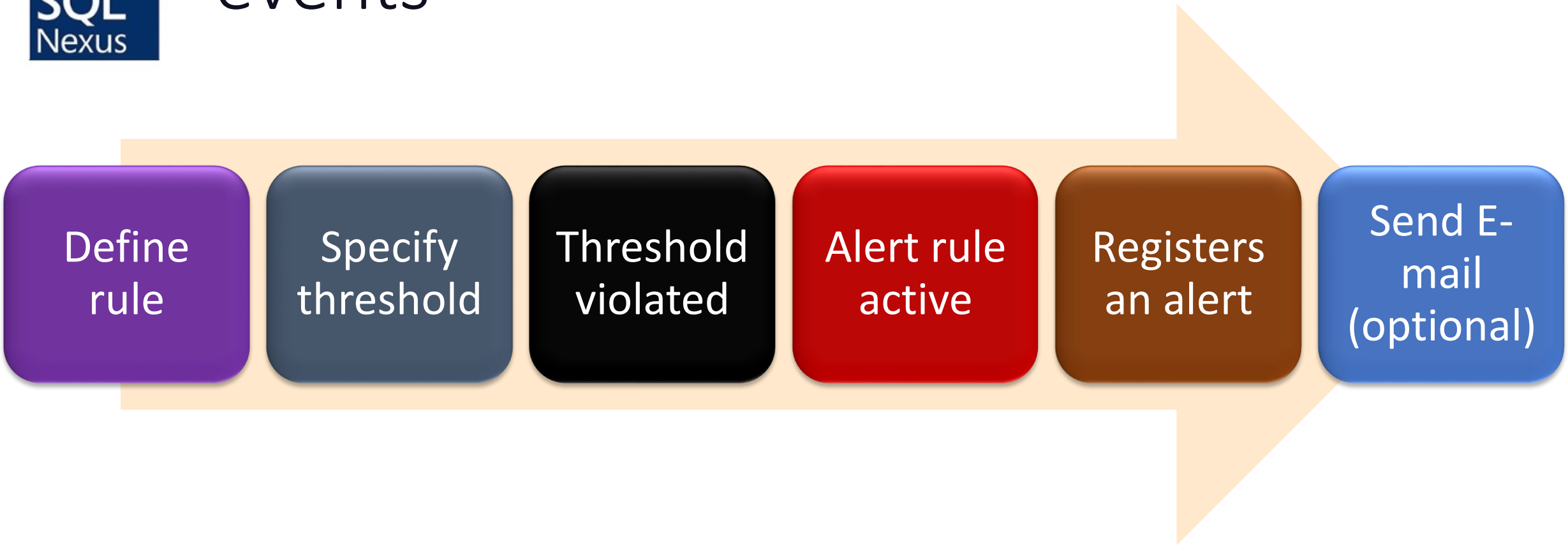
Microsoft

SQL Azure Database V12

Alerts

Microsoft

SQL Azure Database V12 - Alerts and events



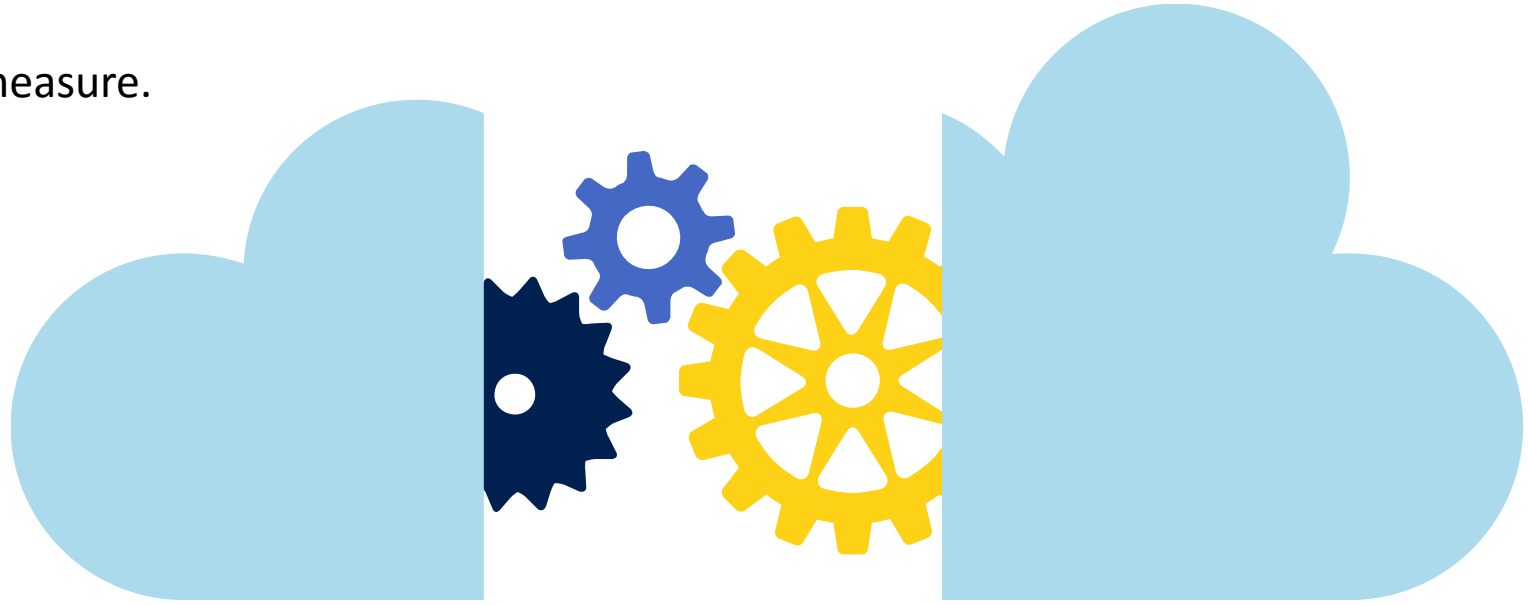
SQL Azure Database V12 - Alert Rules

- Blocked by Firewall
- Failed Connections
- Successful Connections
- CPU Percentage
- Deadlocks
- DTU Percentage
- Log IO Percentage
- Data IO Percentage
- Total Database Size

Microsoft

SQL Azure Database V12 - Alerts – 3rd Party tool

- New Relic issue causing a high memory usage, must be to review the actual tool that customer has.
- Review with Nagios some timeouts and our customer needs to implement other alternatives.
- Performance issue with you have a lot of row in the extended events.
- Review the unit of measure.



Microsoft

SQL Azure Database V12

Q&A

If you have any question, please send an email

jmjurado@microsoft.com

Microsoft

SQL Azure Database V12

Thanks!!

Microsoft