Microsoft

Azure SQL Database

The intelligent relational cloud database service

Patrick Alexander
Data Solution Architect - Microsoft

Patrick.Alexander@Microsoft.com
Twitter: @iLikeAzureSQL

Microsoft Azure

Microsoft

Microsoft Azure Account | Start Free with $200 Credit

https://azure.microsoft.com/en-us/free/

SQL Database – Cloud Database as a Service | Microsoft Azure

https://azure.microsoft.com/en-us/services/sql-database/

Azure Blog

https://azure.microsoft.com/en-us/blog/

Microsoft Azure SQL DB Certification

https://ready.azurewebsites.net/csslearning/2083

Free Online Courses

https://azure.microsoft.com/en-us/training/free-online-courses/

Free Hands-on Labs

https://www.microsoft.com/handsonlabs

Introduction to Azure Data Services

Microsoft Azure

Microsoft Azure Data Services

SQL Server in a VM	SQL Database	CosmosDB	Tables	Blobs
fully featured RDBMS				
transactional processing				
rich query				
	managed as a service	managed as a service	managed as a service	managed as a service
	elastic scale	elastic scale	elastic scale	elastic scale
		schema-free data model	schema-free data model	schema-free data model
		Internet accessible http/rest	Internet accessible http/rest	Internet accessible http/rest
			arbitrary data formats	arbitrary data formats

Azure SQL Database- The developer's intelligent cloud database service

Microsoft Azure

Azure SQL Database

The developer's intelligent cloud database service

- Built for application developers
- Gives you more time to innovate
- Accelerates your time to market
- Maximizes performance, reliability, and data protection
- Lets you use the languages and platforms you prefer

SQL Database Service

A relational **database as a service**, fully managed by Microsoft.

For cloud-designed apps when **near-zero administration** and **enterprise-grade** capabilities are key.

Perfect for cloud **architects and developers** looking for programmatic, DBA-like functionality.

Elastic scale and performance

- Predictable performance levels
- Programmatic scale-out
- Dashboard views of database metrics

Business continuity and data protection

- Self-service restore
- Disaster recovery
- Compliance-enabled

Familiar and self-managed

- Familiar and compatible
- Programmatic
- Self-managed

Azure SQL Database

The developer's intelligent cloud database service

Learn and adapt
with your app!

Scales on the fly
without app downtime!

Build multitenant apps
with customer isolation and efficiency!

Work within your environment
so you can focus on building great apps!

Protects and secures
your app data!

Azure SQL Database

delivering the latest

Security

- Advanced threat detection
- Transparent data encryption with AKV integration
- Encryption by default

Intelligence

- Advanced machine learning with R/Python support
- Graph support for relationship-focused apps
- Adaptive query processing
- Automatic performance tuning

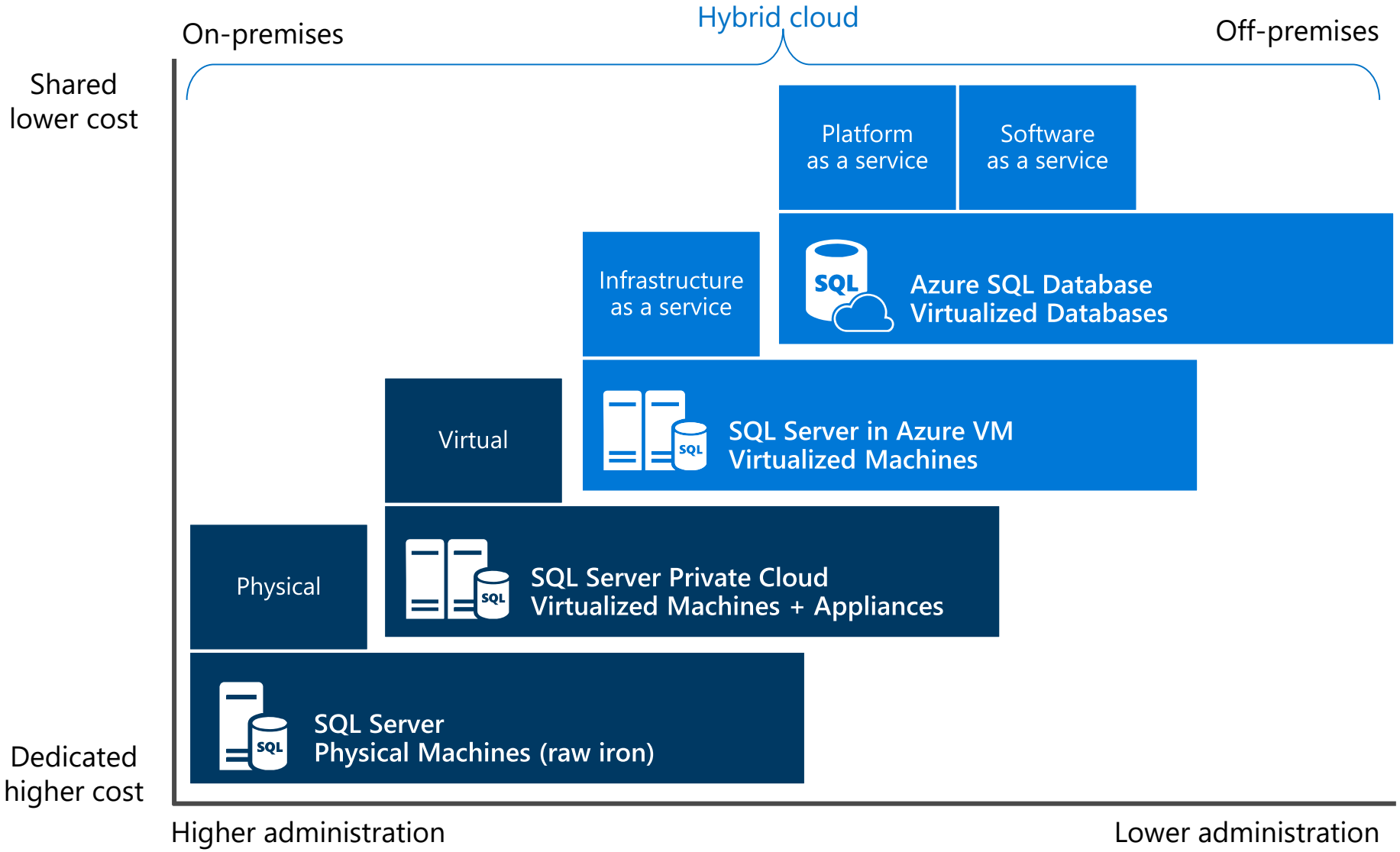
Continuity

- 4TB Premium DB/1TB Standard DB
- Automatic DR
- Next generation data sync
- Read scale
- Long term data retention

Azure SQL Database vs. VMs

	SQL Server in a VM	Azure SQL Database
Best for…	Existing applications that require full box product functionality	Applications that need elastic scale and/or reduced overhead
Resources	Customer has ecosystem of IT resources for support and maintenance	Customer does not want to add additional IT resources for support and maintenance
TCO benefits	Removing CAPEX	Avoiding CAPEX and OPEX
Scalability	Scale up to 343,000 TPS	Scale out to thousands of DBs, process TBs of OLTP data

Data platform continuum



Targeting applications with Azure SQL Database

Cloud-born SaaS applications

New SaaS providers

Goal: Compete and replace on-premises competitor applications

Needs:
- Hyper-scale, HADR, performance
- Write code once
- Predictable and flexible cost structure to build a business model

ISV → SaaS applications

Existing SQL ISV
Re-architecting for SaaS

Goal: Replace existing LOB packaged software with a SaaS solution

Needs:
- Hyper-scale, HADR, performance
- SQL compatibility
- Write code once
- Predictable and flexible cost structure to build a business model

Custom customer-facing applications

All entities

Goal: Scale as the customer base grows

Needs:
- Time to solution
- Hyper-scale, HADR, performance
- Predictable and flexible cost structure to build a business model

Employee-facing applications

Enterprises
Building custom applications

Goal: Create or re-architect apps to cloud or use SaaS solutions

Needs:
- Hyper-scale, HADR, performance
- SQL compatibility

Packaged LOB applications – Delivered by ISVs

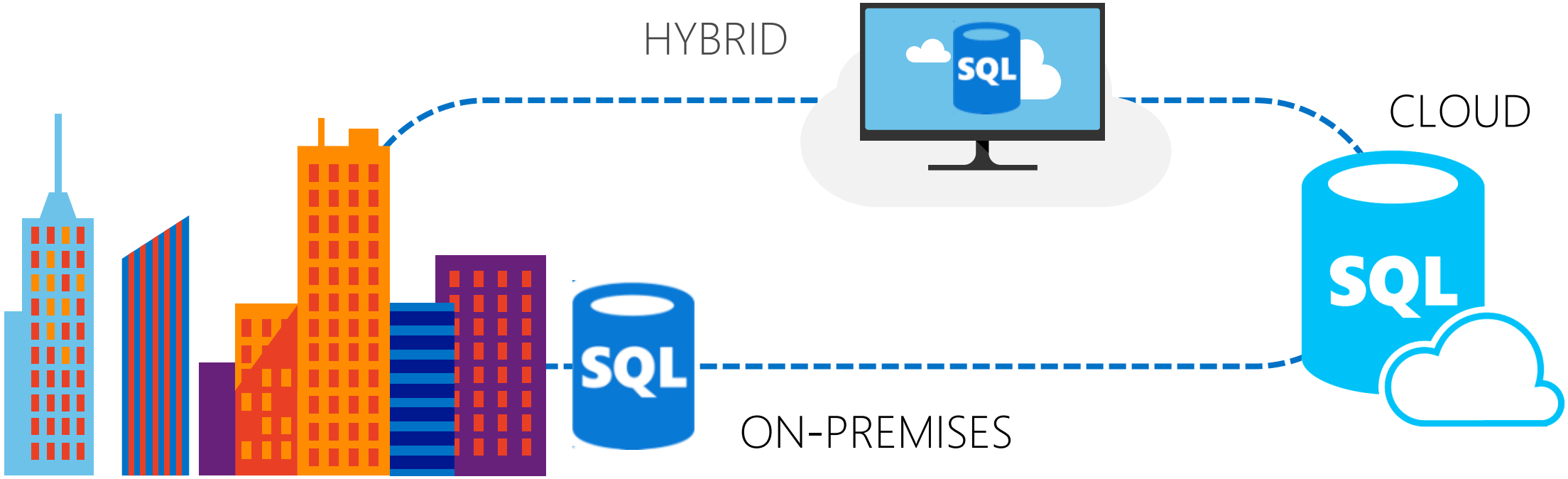
Elastic databases

Single databases

Evolution of SQL Database over the past year

SLA	Increased from 99.9% to **99.99%** uptime SLA
Performance	Service design enables scale up and out of resources, delivering **predictable throughput and performance** regardless of demand
Protection	Point-in-time restore, geo-restore, and active geo-replication **protect against human and environmental-initiated events**
Compliance	**Azure certifications**: ISO, HIPAA BAA, EU Model Clause **Auditing** on SQL Database
Flexibility	Hourly billing and broad set of price points

SQL Server: A consistent experience from on-premises to the cloud and in between



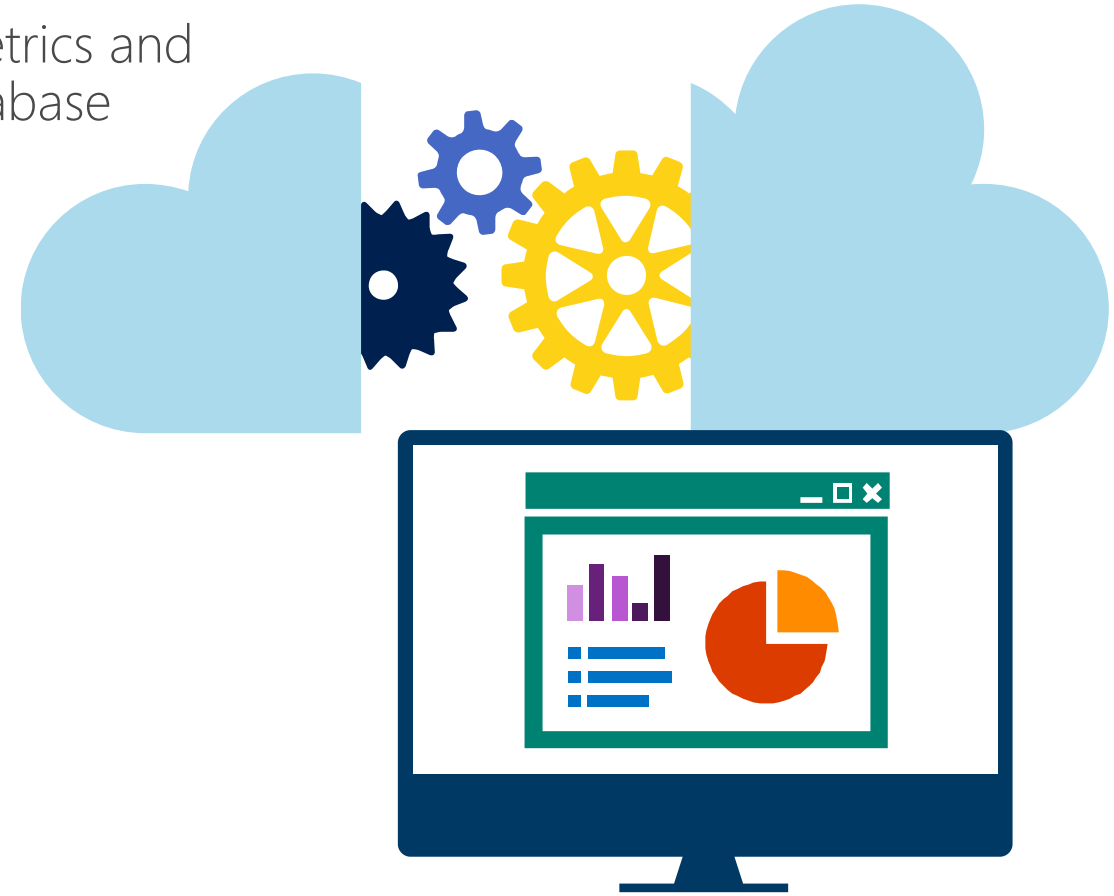
In-memory across all workloads

Learns and adapts with your applications

Microsoft Azure

Intelligent capabilities

- Built-in performance monitoring with intelligent advisors helps reduce troubleshooting time
- The Azure management portal exposes real-time metrics and dashboard, and you can set check and alerts on database performance
- Azure SQL Database includes intelligent advisors:
  - Index Advisor
  - Query Performance Insight
  - Threat Detection
  - Query Store
  - Elastic DB Advisor
  - Graph Support
  - Adaptive Query Processing
  - More…



Query Store

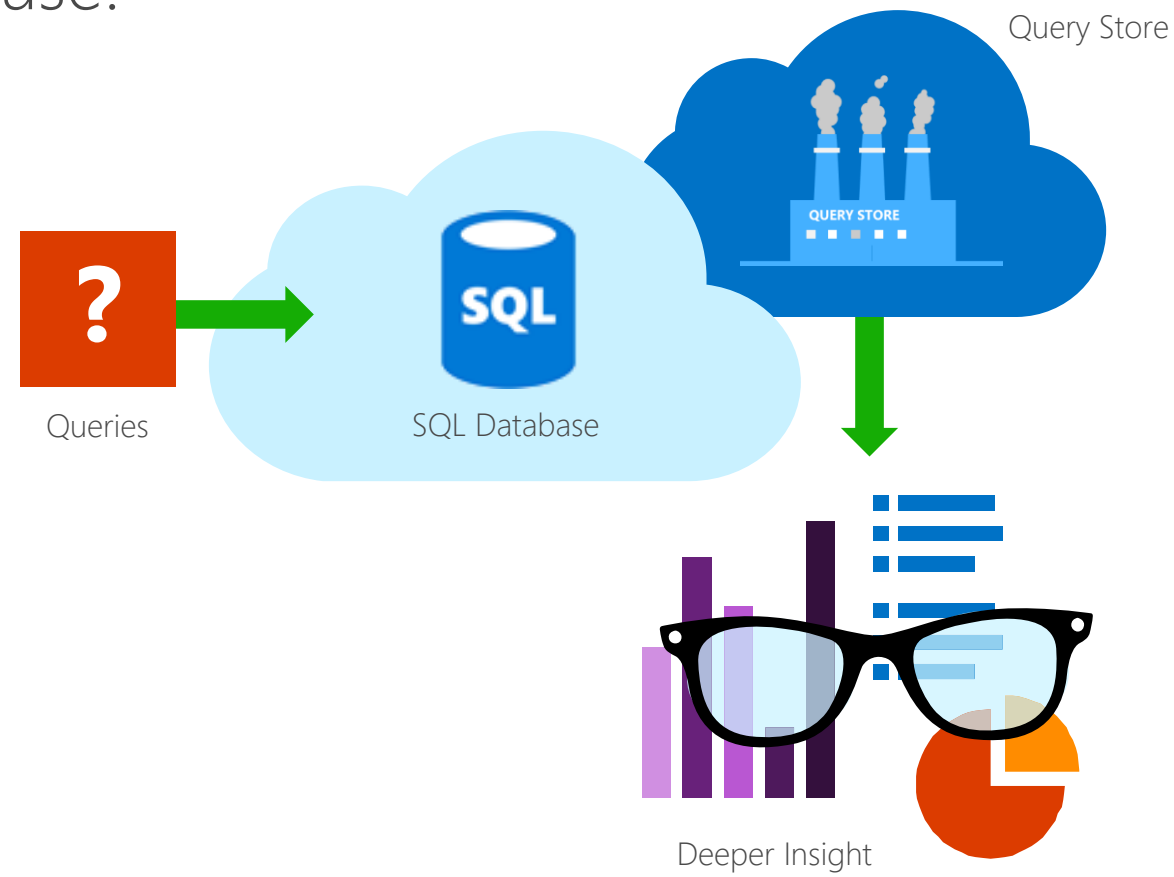
Comprehensive query-performance information when you need it most

Workload-data recorder for your database:

- Queries, plans, and compilation and runtime statistics available at your fingertips
- Get full history of query execution and quickly pinpoint the most expensive queries
- Allows you to easily identify and fix performance issues in minutes

Enables the following scenarios:

- Finding regressed queries
- Identifying top resource-consuming queries
- Optimizing ad hoc workload
- Streamlining application upgrades



Query Performance Insight

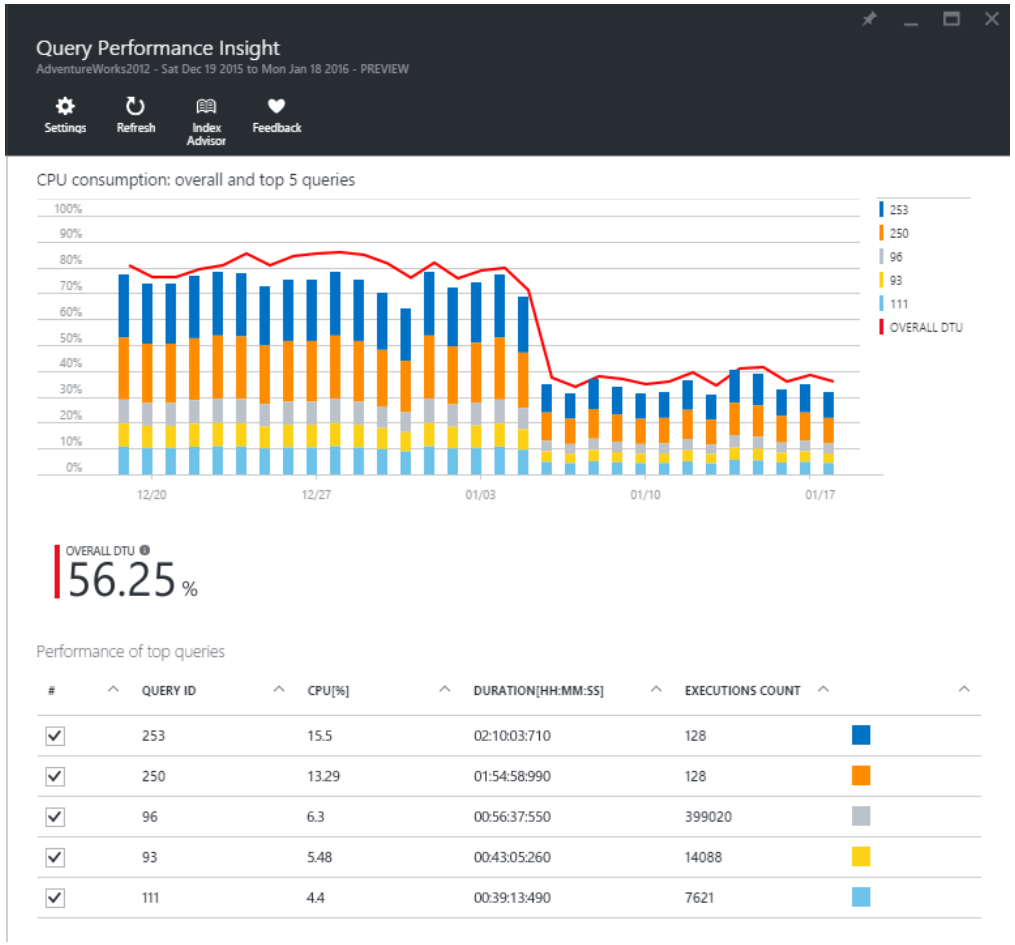
See how database resources are being consumed

See the most CPU-intensive queries:

- Customize your view by selecting observation interval, number of queries, and aggregation type
- View aggregated statistics about your workload: total duration and number of executions

Drill down to a specific query:

- Get granular view on query execution intervals
- View query text



#	QUERY ID	CPU[%]	DURATION[HH:MM:SS]	EXECUTIONS COUNT
☑	253	15.5	02:10:03:710	128
☑	250	13.29	01:54:58:990	128
☑	96	6.3	00:56:37:550	399020
☑	93	5.48	00:43:05:260	14088
☑	111	4.4	00:39:13:490	7621

SQL Database Advisor

Improve the database performance with index tuning recommendations

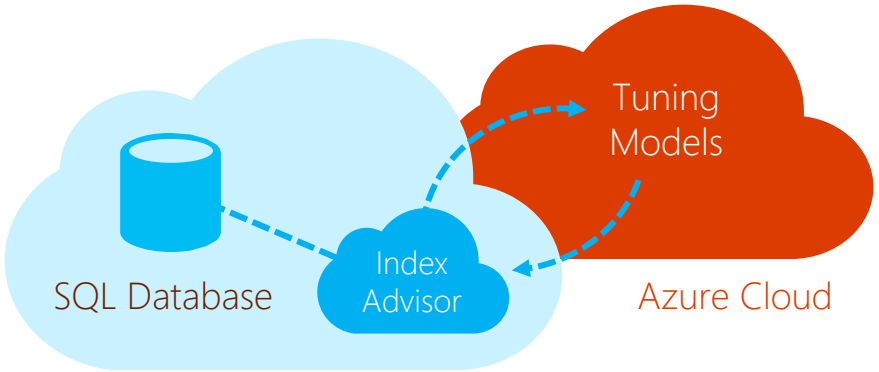
Index tuning recommendations tailored to each DB

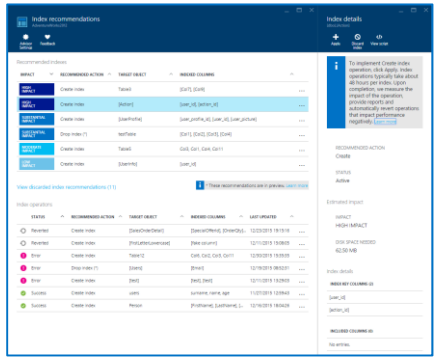
- Recommendations are based on the observed usage, and evolve as the DB workload changes
- Support for CREATE and DROP index, more types of recommendations underway

Intelligent service for implementing and validating index recommendations

- Full-auto mode: service takes full care of the indexes for your DB
- Manual “review and apply” mode for full control

Report + visualization of index impact

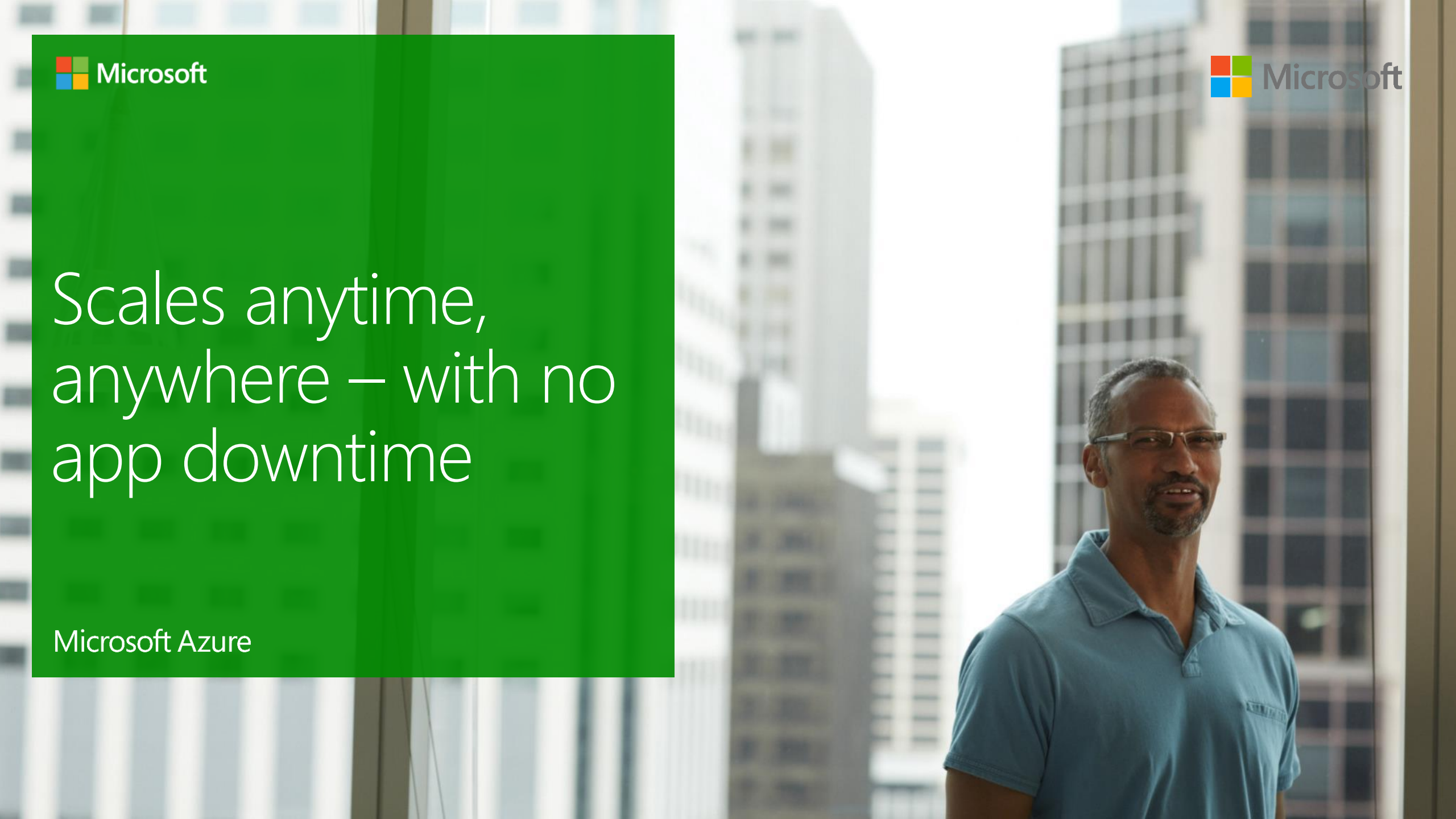




Microsoft

Scales anytime, anywhere – with no app downtime

Microsoft Azure



Learning objectives

Scalable performance when managing multiple and large numbers of databases

- Predictable performance
- Pay for what you need
- In-Memory OLTP
- Real-time analytics

Scalable performance, no downtime

Predictable performance

Pay for what you need, when you need it

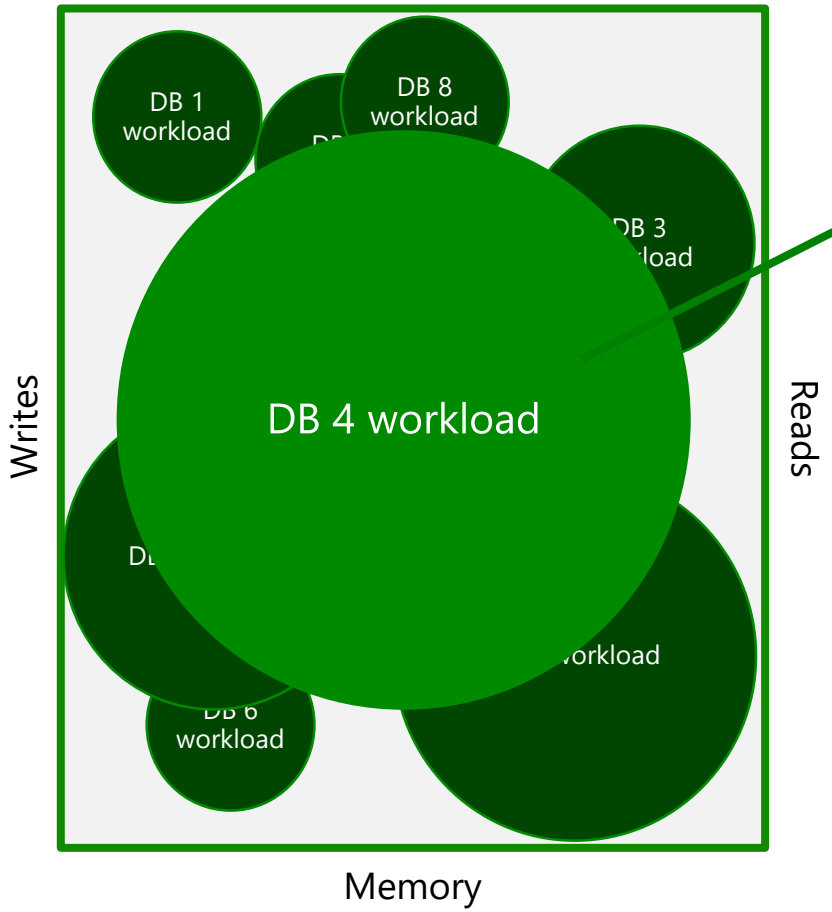
In-Memory OLTP

Real-time analytics

- With one mouse click or a single API call, scale up app resources to accommodate growth periods or peak workload demand
- Rescale resource needs of your application with no database downtime or application rewrites
- Pay only for the performance you need, when you need it
- Get guaranteed resources for your databases so that users get a predictable performance experience
- Get up to 30x performance gains with In-Memory OLTP
- Get real-time insights and 100x performance gains with Real-Time Operational Analytics

Performance predictability

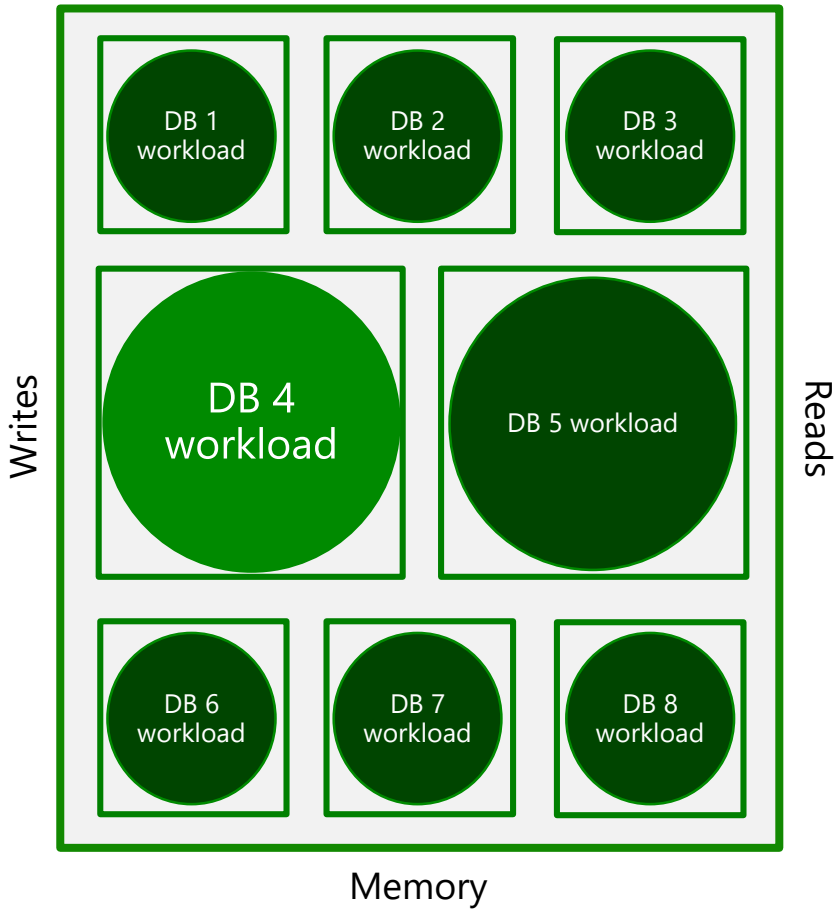
Web/business
machine
compute



Noisy neighbor

Introduction of
bounding boxes
to eliminate the
noisy neighbor problem

Basic/standard/premium
machine
compute



Pay only for what you need, when you need it

Service Tier Advisor recommendations combine **feature usage analysis** with **resource consumption analysis**

BASIC	STANDARD	PREMIUM	PREMIUM RS
Light transactional workloads	The go-to option Determine your service tier for modern business apps	Mission-critical databases	IO-intensive workloads

Elastic scale and performance: Six performance levels across four tiers let you scale up and down based on throughput needs, and offer better resource isolation and an improved billing experience

Business continuity: A spectrum of business-continuity features across tiers let you dial up control over data recovery and failover

Familiar and self-managed: Near-complete SQL Server compatibility and unprecedented efficiencies as your applications scale with a near-zero maintenance service and a variety of familiar management tools and programmatic APIs

For more information click **SINGLE DB** Model / **ELASTIC DB** Model

In-Memory OLTP

Customer benefits	High-performance data operations	Efficient business-logic processing	Frictionless scale-up	Hybrid engine and integrated experience
Architectural pillars	Main-memory optimized	Transact-SQL compiled to native machine code	High concurrency	SQL Server integration
	• Optimized for in-memory data • Indexes (hash and range) exist only in memory • No buffer pool or B-trees • Stream-based storage	• Transact-SQL compiled to machine code by using C code generator • Invoking a procedure is just a DLL entry-point • Aggressive compile-time optimizations	• Multi-version optimistic concurrency control with full ACID support • Core engine uses lock-free algorithms • No lock manager, latches, or spinlocks	• Same manageability, administration, and development experience • Integrated queries and transactions • Integrated high availability and backup/restore
Drivers	Hardware trends			Business
	Steadily declining memory price for non-volatile random-access memory (NVRAM)	Stalling CPU clock rate	Processors with many cores	Total cost of ownership (TCO)

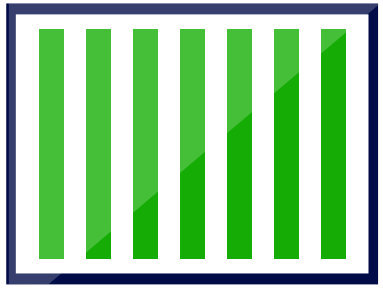
For more information click **Why In-Memory** / **Why In-Memory Columnstore**

In-Memory technology for real-time performance

Expedite query and transaction processing speed

AZURE SQL DATABASE

In-Memory Analytics



100x

performance gains

+

In-Memory OLTP

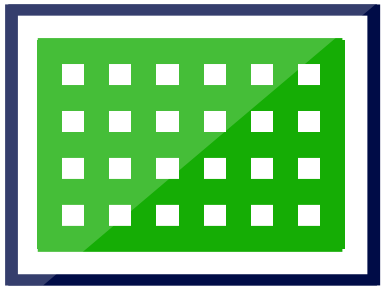
Up to

30x

faster transactions

=

Breakthrough performance
Real-time Operational Analytics



Real-time business insights based on operational data

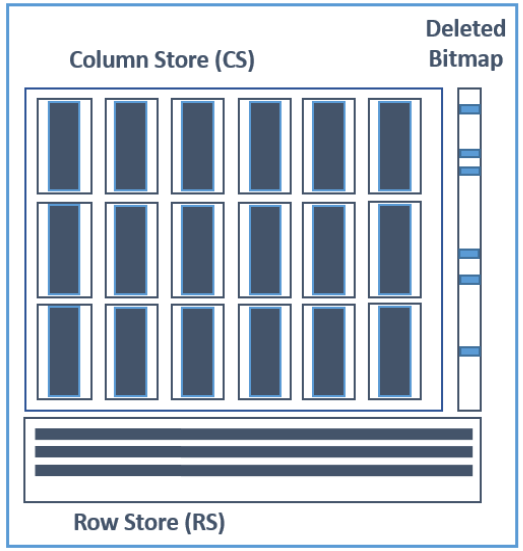
Why In-Memory OLTP

- Customers demand ever-higher throughput and performance for OLTP at lower costs
  - Hardware trends require architectural changes for relational database-management systems (RDBMS) to meet those demands
- In-Memory OLTP is:
  - High performance
  - Memory-optimized as an OLTP engine
  - Integrated into SQL Server
  - Architected for modern hardware

In-memory columnstore

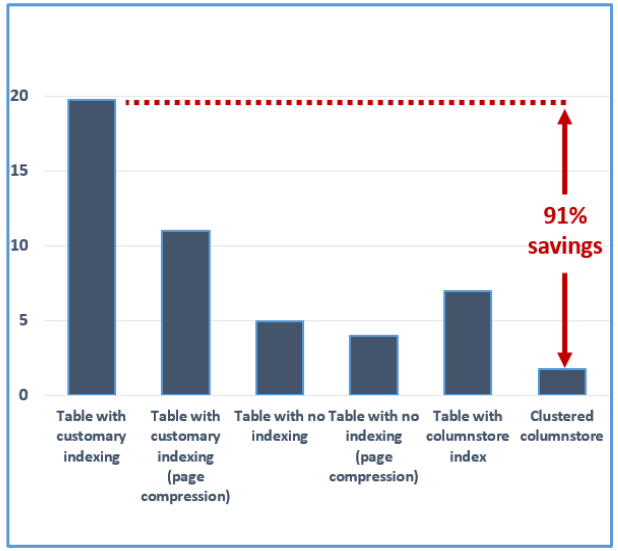
- Updatable non-clustered columnstore index
- In-Memory OLTP plus columnstore
- Faster batch-mode scans using CPU vector instructions
- Dynamic aggregate pushdown
- Primary key/foreign key (PK/FK) enforcement
- Offload reporting to AlwaysOn secondary replica

Clustered ColumnStore Index



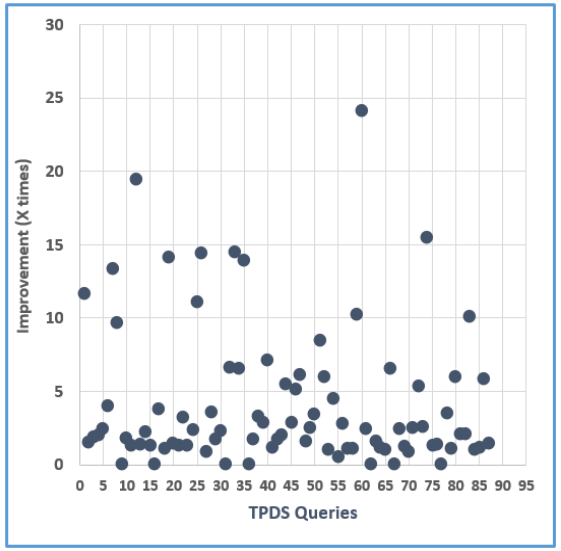
- Update-able columnstore
- Partition level statistics

Space Consumption



- 5-10X data compression
- Additional archival compression

Performance Characteristics



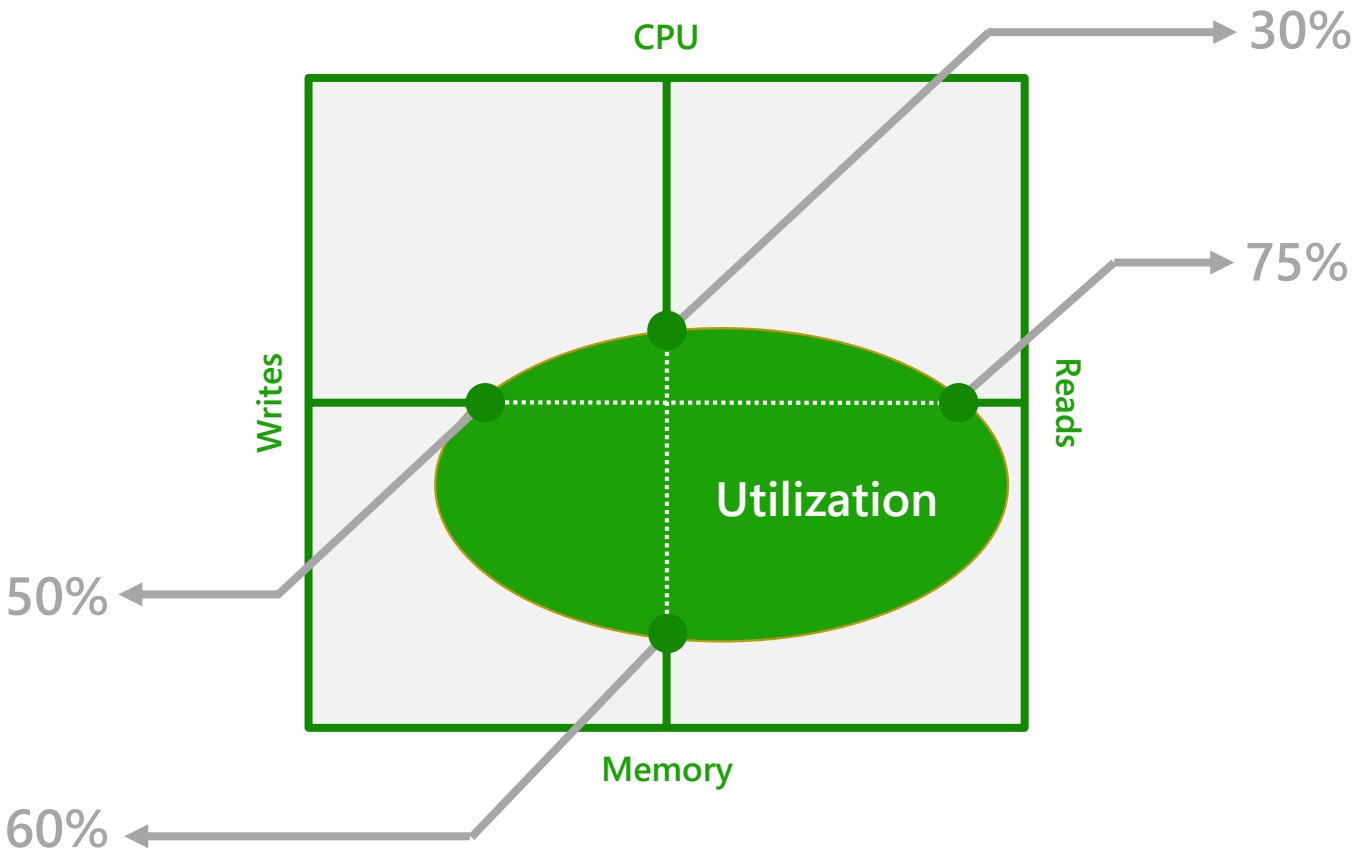
- Next generation query processing
- 5-10X improvement

Database Transaction Unit – DTU

Bounding box

Monitoring database workload utilization within bounding box

- Represents the relative power (resources) assigned to the database
- Blended measure of CPU, memory, and read-write rates
- Compare the power across performance levels
- Simplifies talking about performance, think IOPS vs. %



Designed for predictable performance

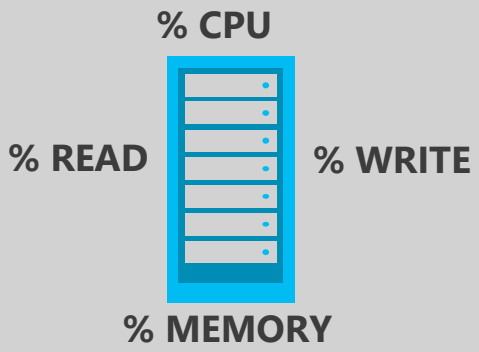
Redefined

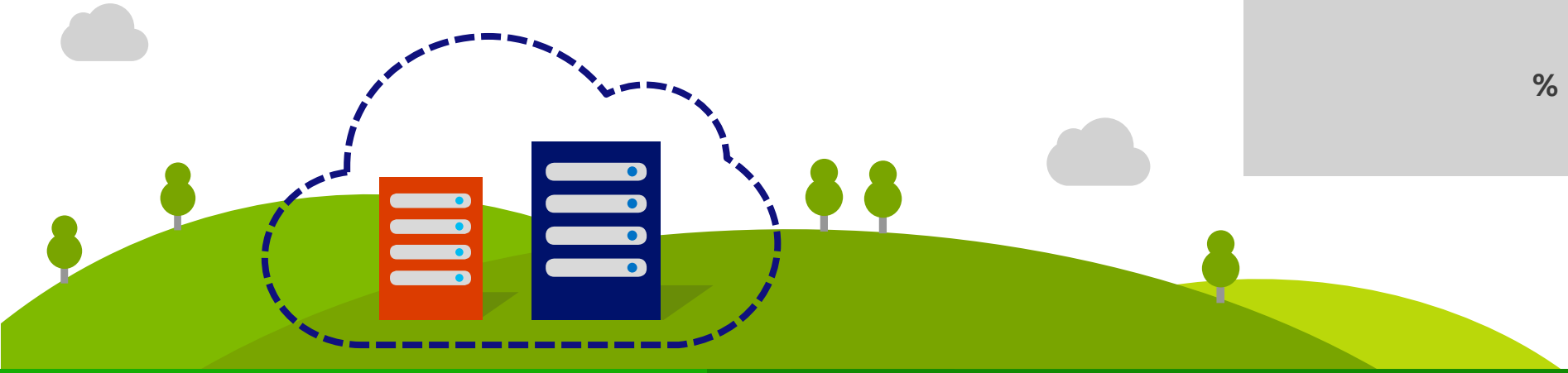
Across the service tiers, each performance level is assigned a defined level of throughput for a streamlined experience

Measure of power

Introducing the Database Transaction Unit (DTU), which represents the relative power of databases based on a real-world measure: **the database transaction**

DTU represents a set of operations that are typical for an online transaction processing (OLTP) request, and then measured by how many transactions could be completed per second under fully loaded conditions





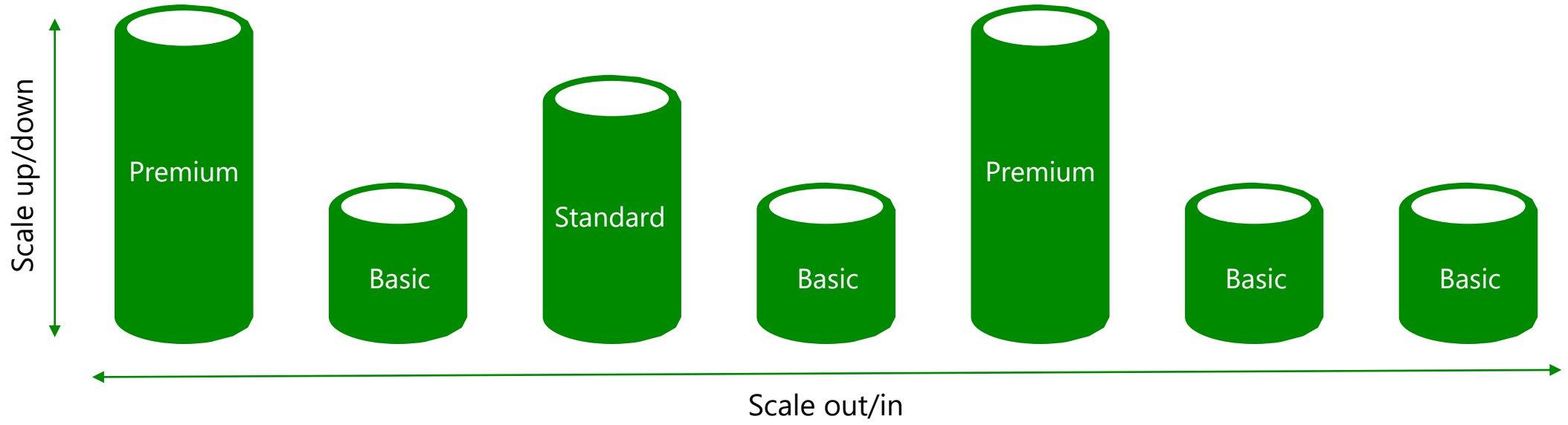
Scalability options in Azure SQL Database

Vertical: Scale up or scale down

Change service tiers for a given database as capacity needs fluctuate

Horizontal: Scale out or scale in

Add or remove databases (sharded and/or in a pool) as more or less capacity is needed



Service Tier Selection Overview

Choosing right service tier / performance level is dependent on:

- Feature usage and features available in different Service Tiers
- Resource usage and bounding boxes of different Performance Levels

Selection should take into account:

- Minimum database features required
- Resource consumption
- Database Size

SQL Database service tiers (single DB model)

	BASIC	STANDARD				PREMIUM						PREMIUM RS (Preview)			
		S0	S1	S2	S3	P1	P2	P4	P6	P11	P15	S0	S1	S2	S3
Built For	Light transactional workloads	Medium transactional workloads				Heavy Transactional Workloads						IO-intensive Workloads			
Available SLA	99.99%*											< 99.99%			
Database Max Size	2 GB	250 GB				500 GB				4 TB		500 GB			
Point-in-time Restore (“oops” Recovery)	Any point within 7 days	Any point within 35 days													
Business Continuity	Active geo-replication, up to four readable secondary backups														
Security	Auditing, row-level security, dynamic data masking														
Performance Objectives	Transactions per hour	Transactions per minute				Transactions per second									
Database Transaction Units (DTUs)	5	10	20	50	100	125	250	500	1,000	1,750	4,000	125	250	500	1000
Available Tiers ($/Month) and GA Price	$4.99	$15	$30	$75	$150	$465	$930	$1,860	$3,720	$7,001	$16,003	$116	$233	$465	$930

**The 99.99% availability SLA does not apply to the existing Web and Business editions, which will continue to be supported at 99.9% availability.*

Elastic databases at a glance

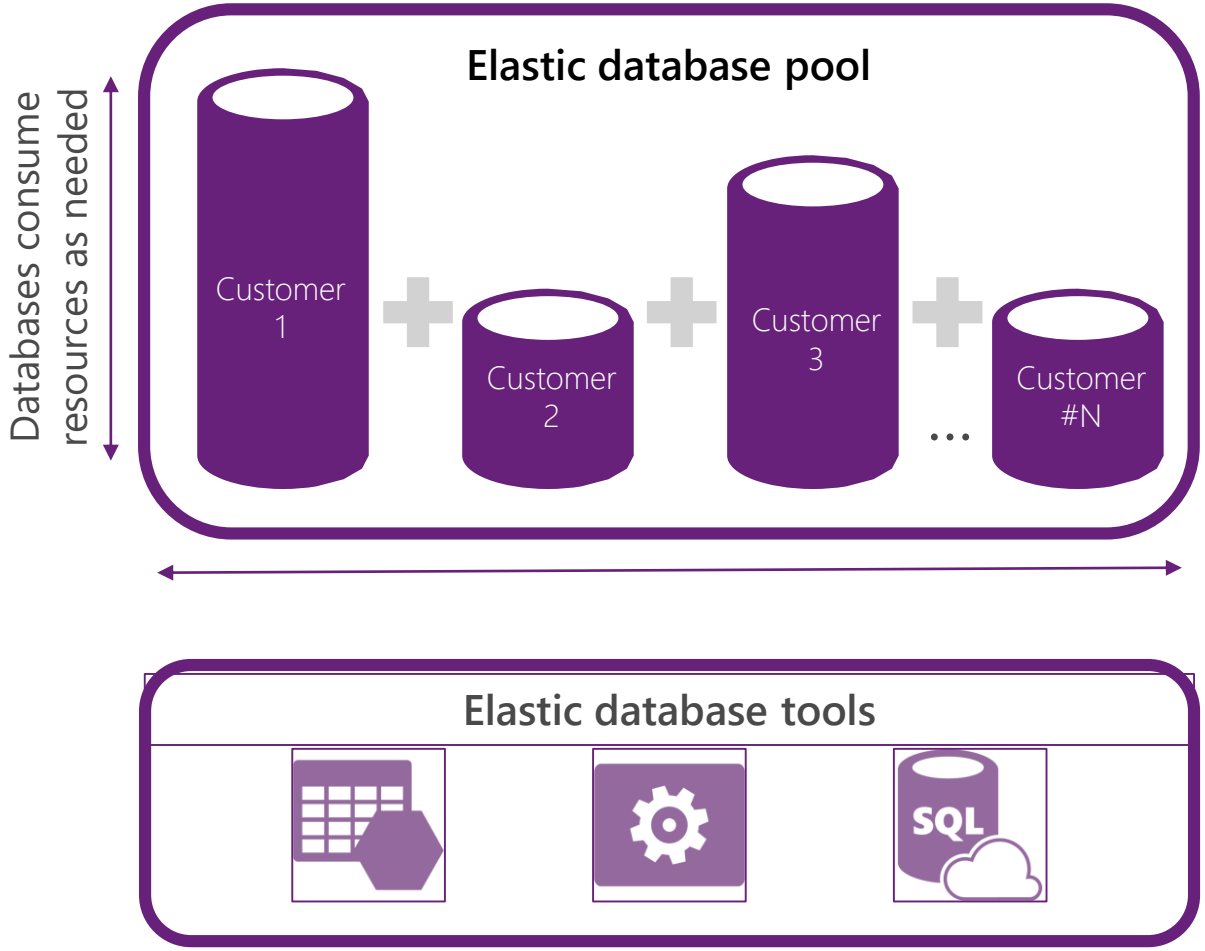
Elastic database pools and elastic database pricing model

Elastic database tools – client library and split-merge service (GA)

Elastic database job (preview)

Elastic database queries (preview) and transactions (GA)

Concepts



An elastic database pool is a collection of DTUs and storage that is used by multiple databases

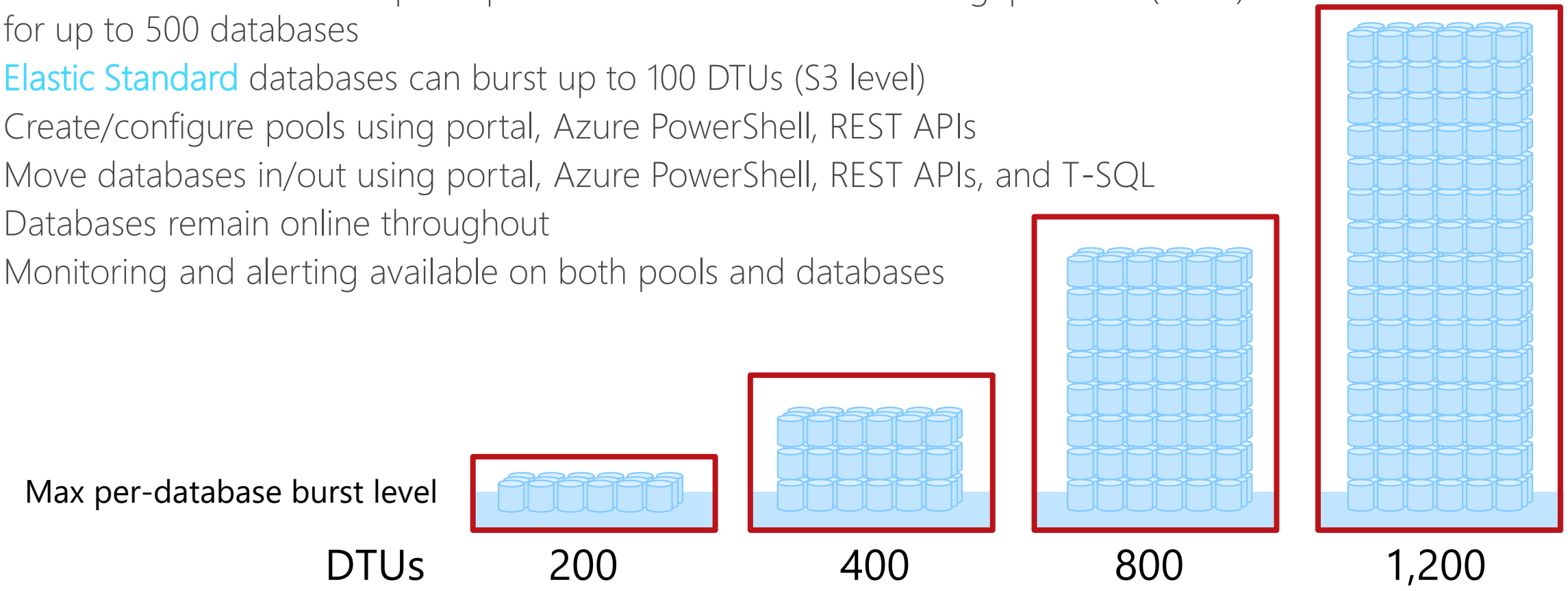
Elastic database jobs allow you to perform tasks across databases in the pool

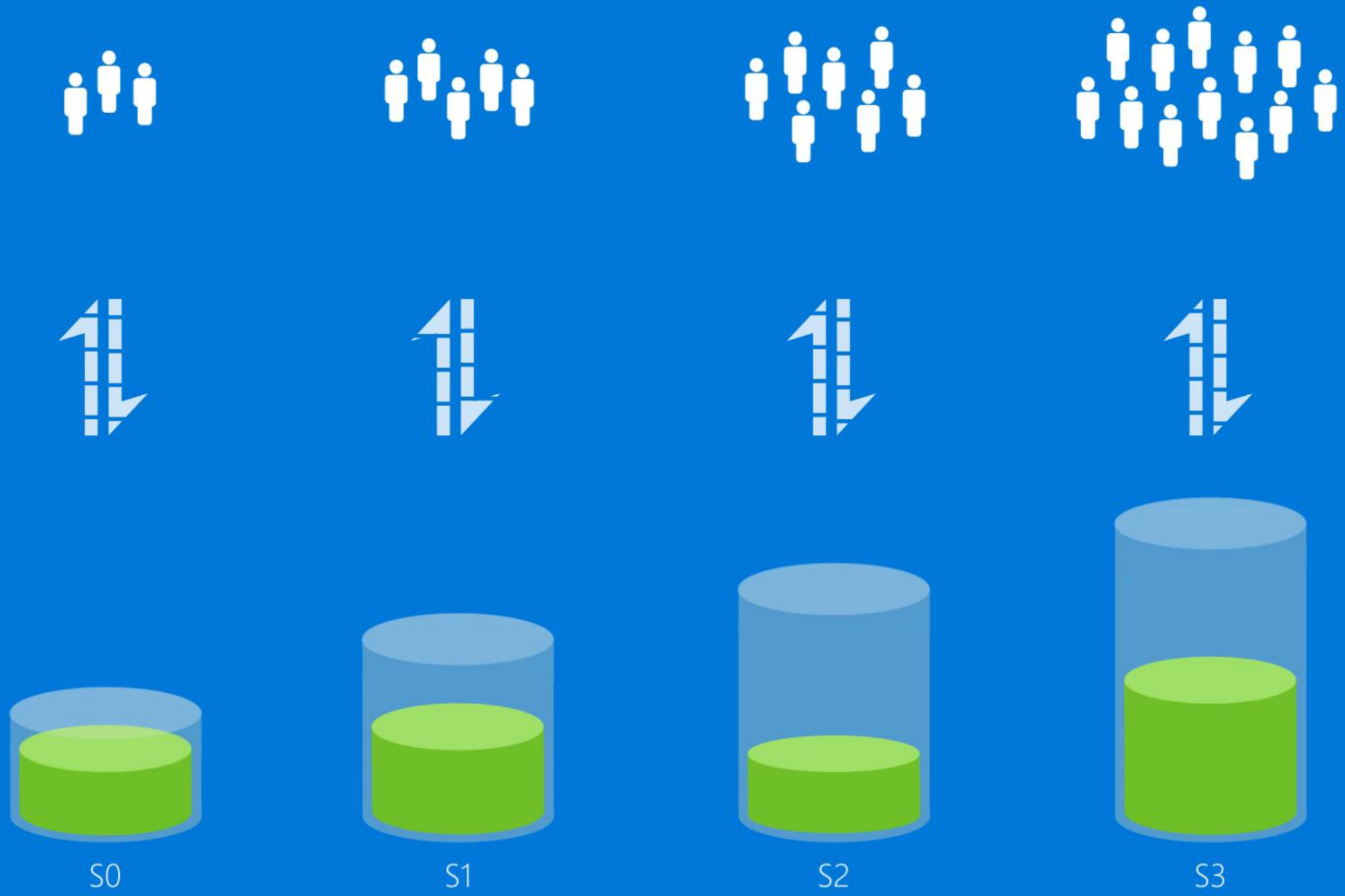
Scenarios include:

- Performing administrative tasks, such as deploying new schemas
- Updating reference data, such as making product information common across all databases
- Rebuilding indexes to improve query performance

Elastic database model

- Elastic databases in elastic database pools
- Pooled resources are used by many databases
- Standard elastic database pools provide 50-3000 database throughput units (DTUs) for up to 500 databases
- Elastic Standard databases can burst up to 100 DTUs (S3 level)
- Create/configure pools using portal, Azure PowerShell, REST APIs
- Move databases in/out using portal, Azure PowerShell, REST APIs, and T-SQL
- Databases remain online throughout
- Monitoring and alerting available on both pools and databases

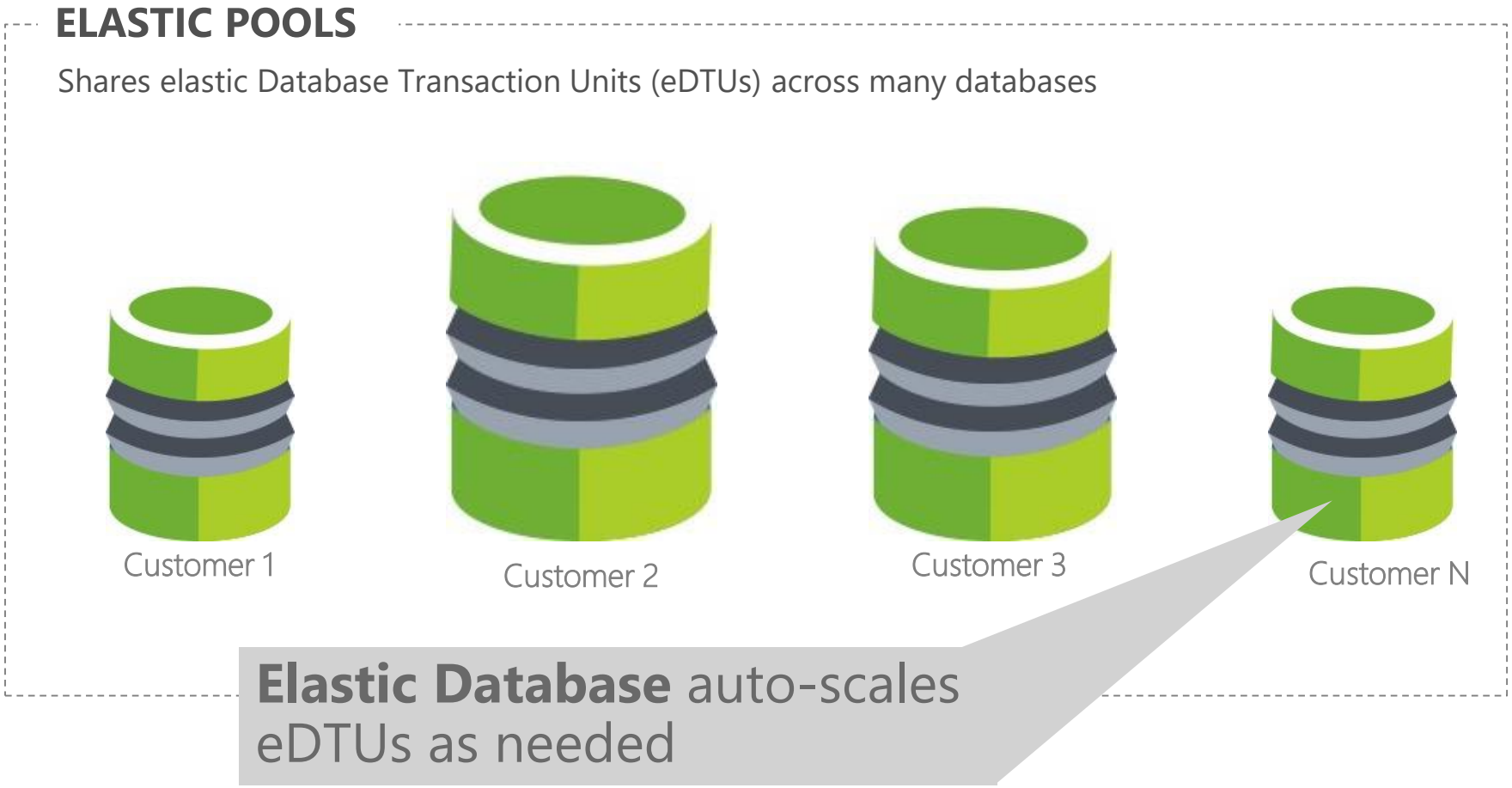



S0
S1
S2
S3

ELASTIC DATABASE POOL
MAX
100 DTUs
MIN
10 DTUs

Elastic Database

Auto-scaling you control.



- Pools automatically scale performance and storage capacity for elastic databases—anytime, anywhere
- You can control the performance assigned to a pool, add or remove elastic databases on demand, and define performance of elastic databases without affecting overall pool cost
- Don’t worry about managing usage needs of individual databases

Elastic database pools

Buy a fixed number of eDTUs, share compute across many databases

Basic	Standard	Premium
Auto-scale up to 5 eDTUs per database	Auto-scale up to 100 eDTUs per database	Auto-scale up to 1,000 eDTUs per database

Elastic tools

Manage operational activities across multiple databases

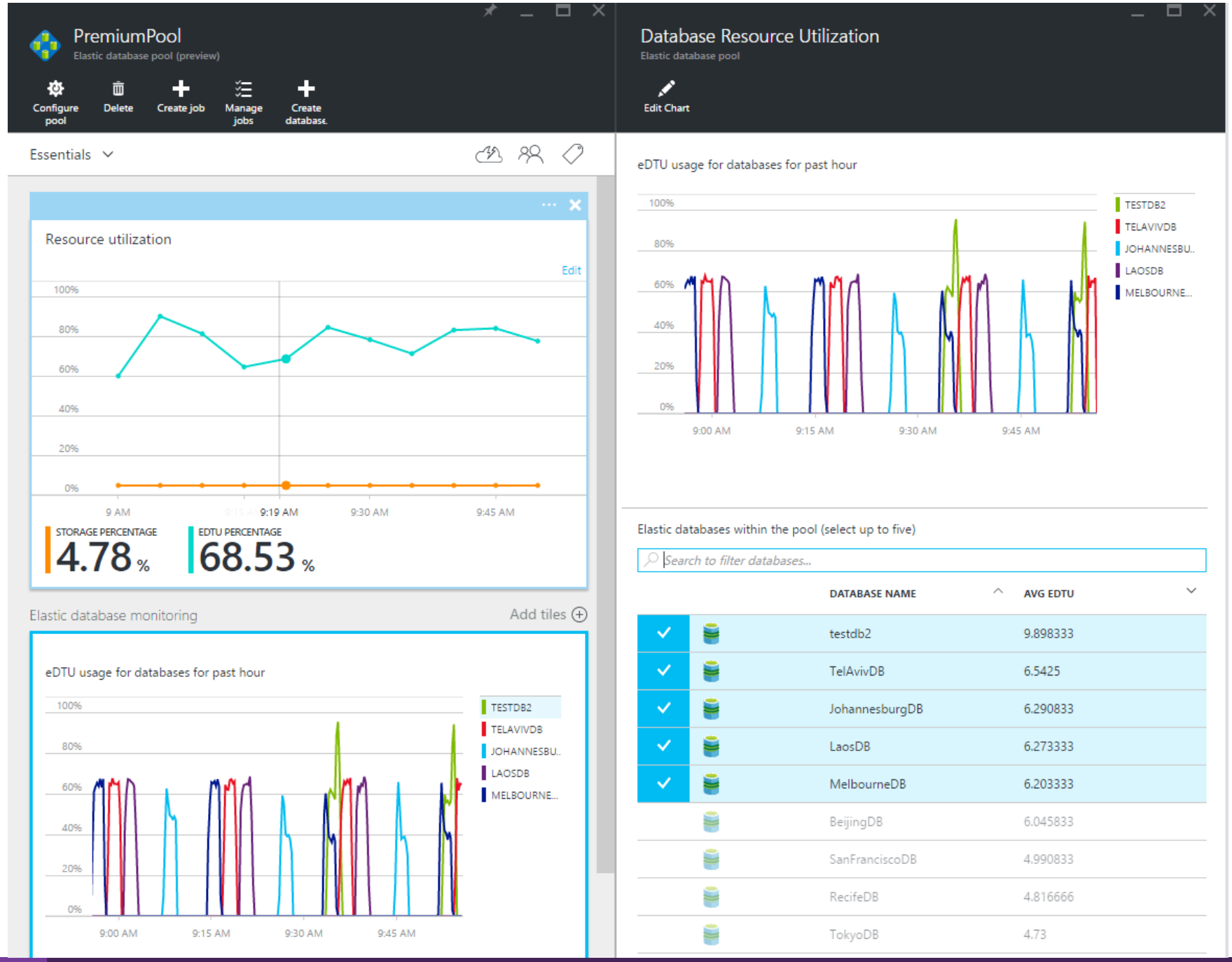
ELASTIC DATABASE POOLS

ELASTIC DATABASE TOOLS

- Elastic database jobs
- Elastic database queries
- Elastic database transactions

Support management and increased efficiency for multi-database environments

Monitor and Manage Elastic Pools



Elastic database jobs overview

Elastic database jobs enables you to run T-SQL scripts (jobs) against all databases in an elastic database pool

Define, maintain, and persist T-SQL scripts to be executed across an elastic database pool

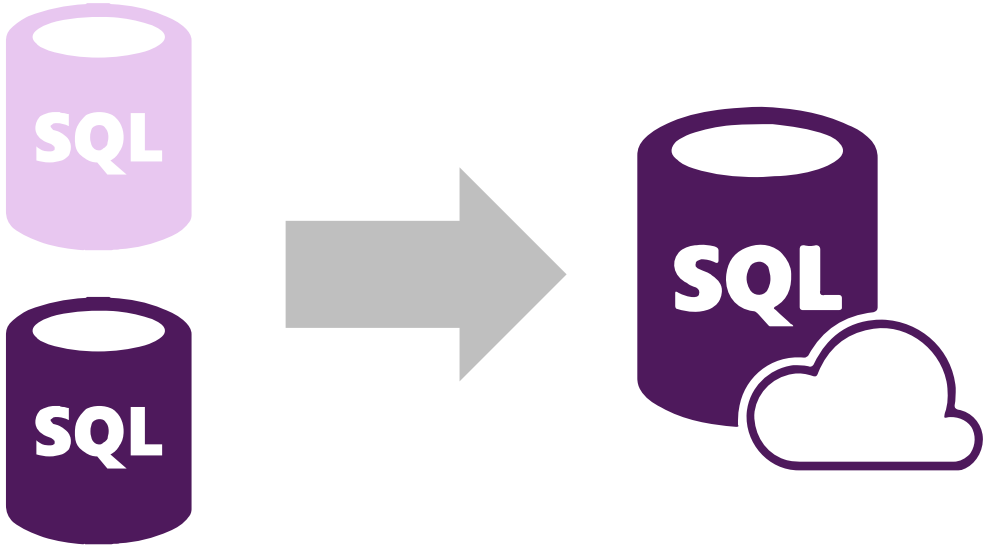
Execute T-SQL scripts reliably with automatic retry and at scale

Track job execution state

For more information click **Getting Started with Elastic Database Jobs**

Migrate to Azure SQL Database

Migrating to Azure SQL Database



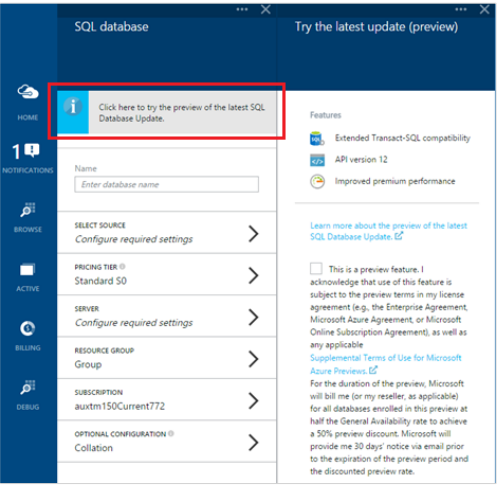
Migrate an on-premises SQL Server database to Azure SQL Database (v12)

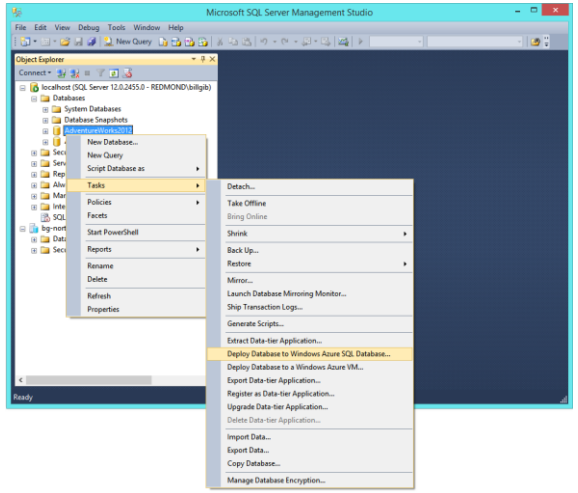
Get near-complete engine compatibility with SQL Server 2016*

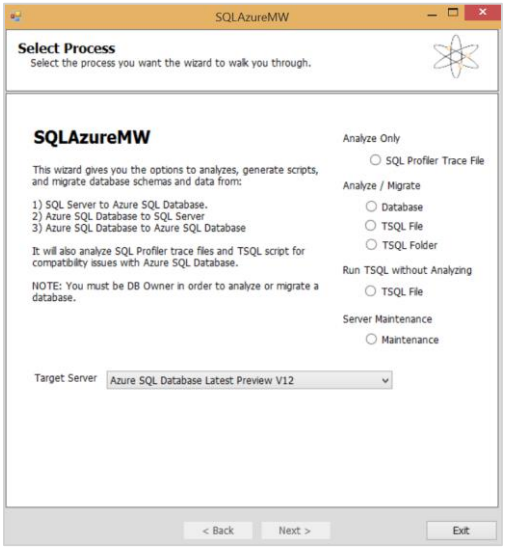
Simplify migration:

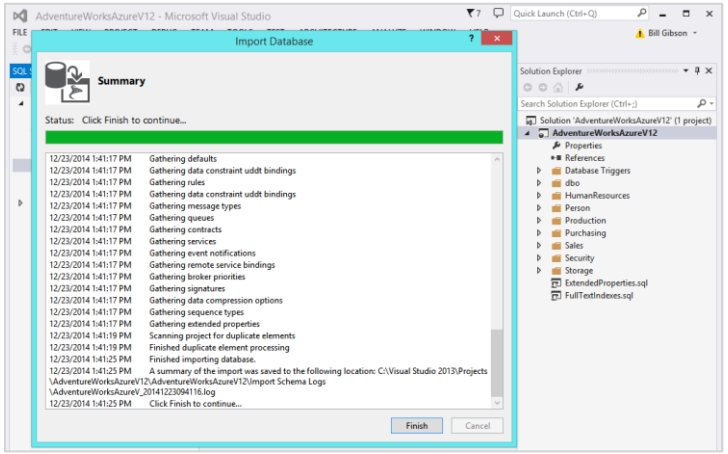
- Straightforward migration for the majority of databases
- Requires few or no changes to schema
- No re-engineering of applications**

Migration tools





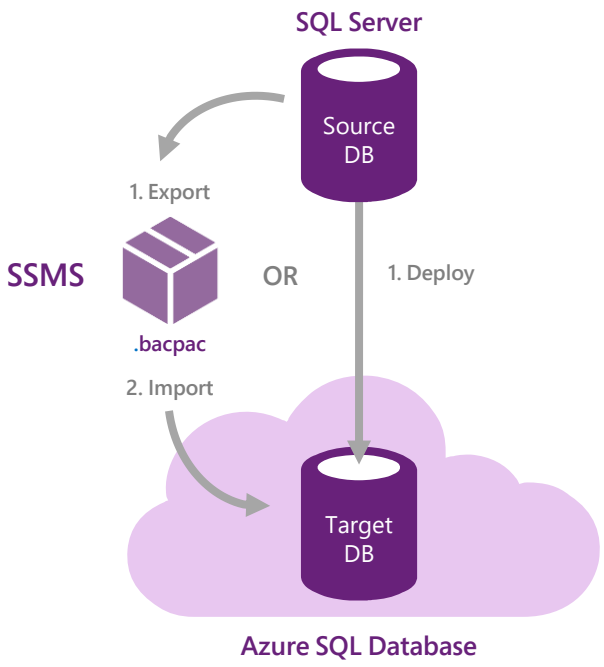




- Azure Management Portal
- SQL Server Management Studio (SSMS)
- SQL Azure Migration Wizard (SAMW)
- SQL Server Data Tools in Visual Studio
- Data Migration Assistant

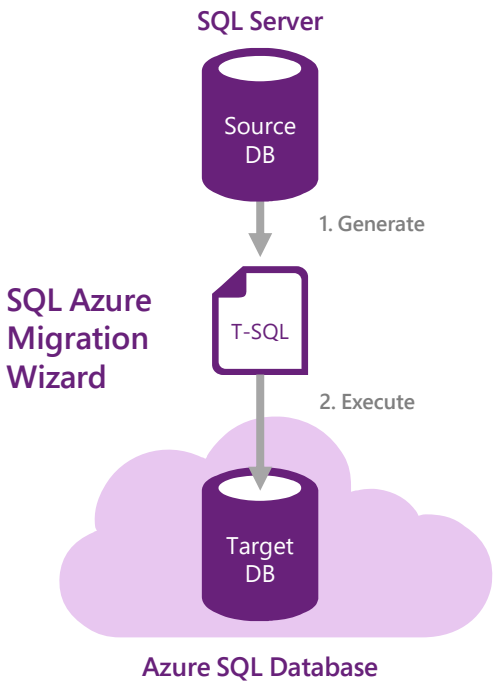
Migration methodologies

Method 1



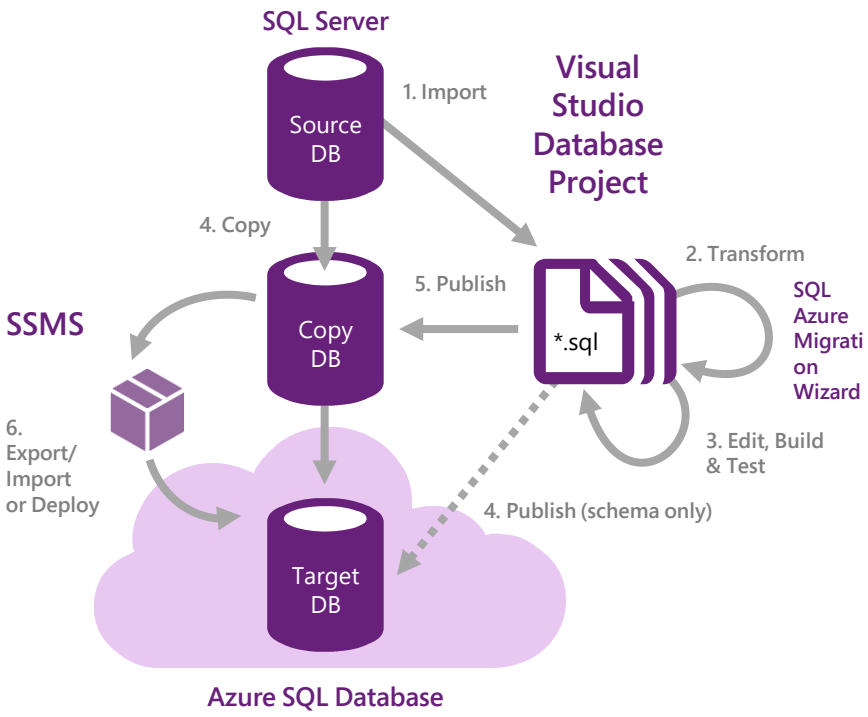
Migrate a compatible database using SSMS

Method 2



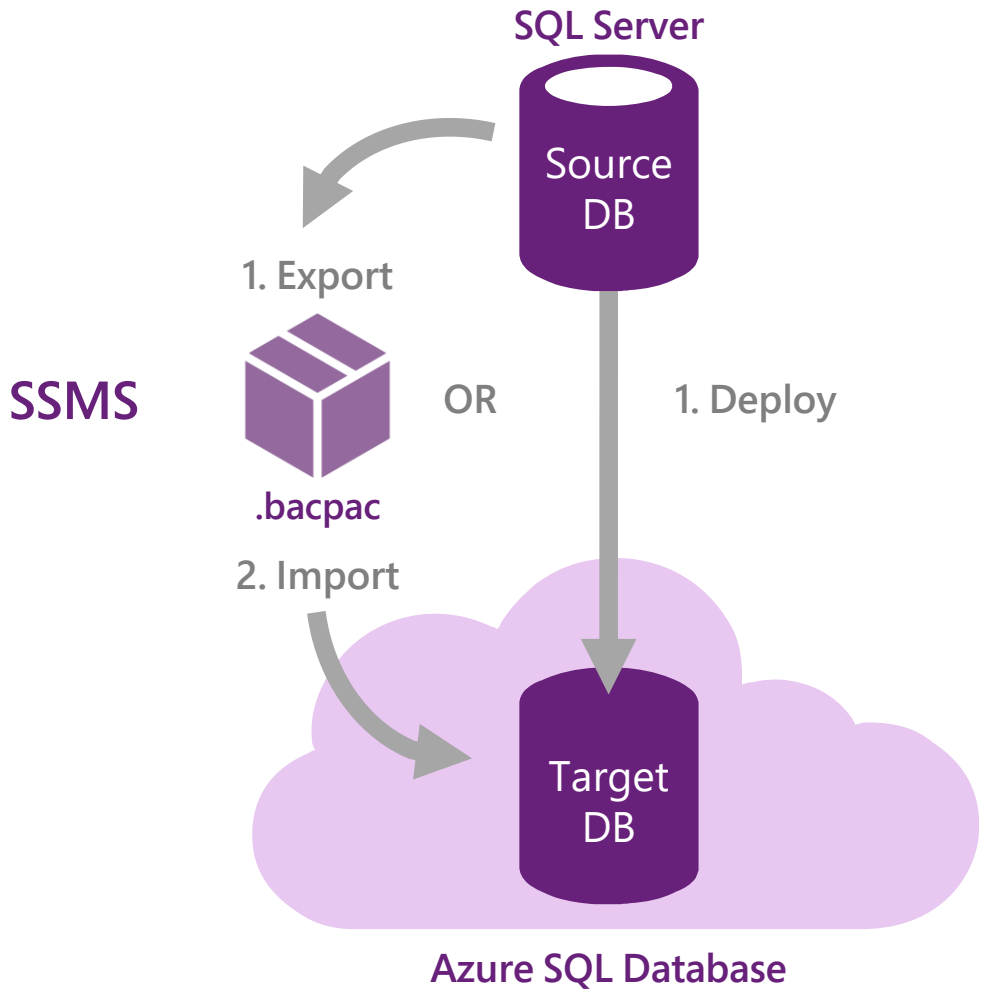
Migrate a near-compatible database using SAMW

Method 3



Update the database schema offline using Visual Studio and SAMW, and then deploy it with SSMS

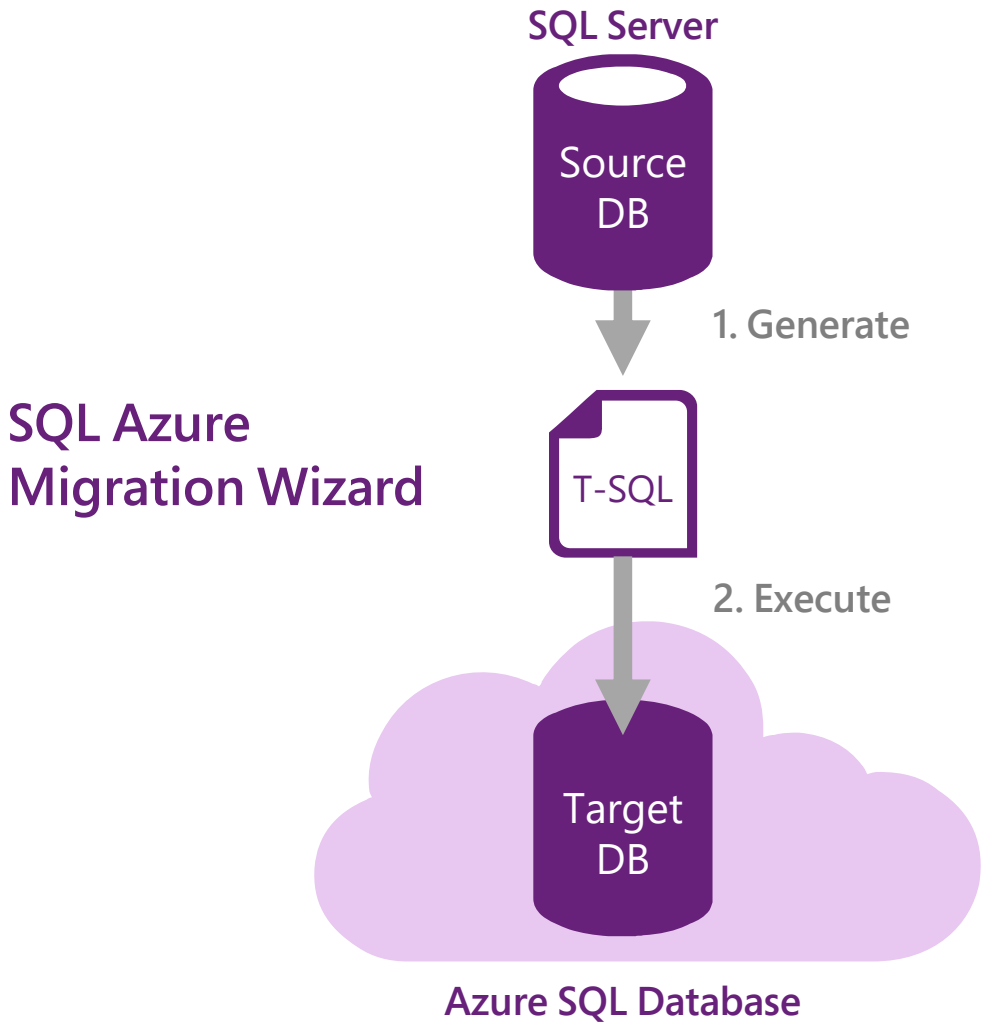
Method 1



Migrate a compatible database using SSMS

- Use SSMS to deploy to Azure SQL Database
- Use SSMS to export a BACPAC and then import it to Azure SQL Database

Method 2



Migrate a near-compatible database using SAMW

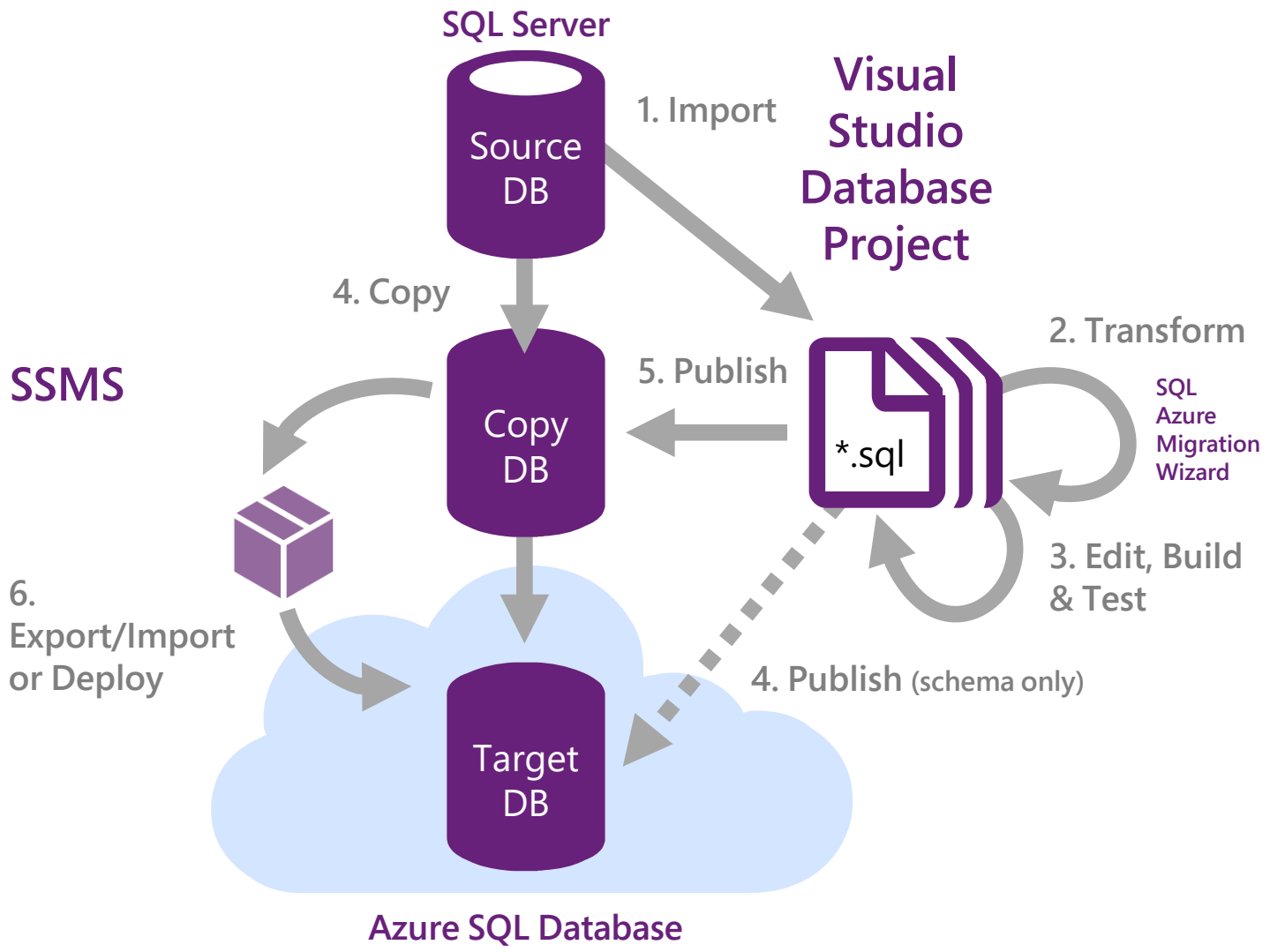
Provision a new database

Analyze and migrate a database with SAMW

Fix errors reported by SAMW

Deploy database to Azure SQL Database using SAMW

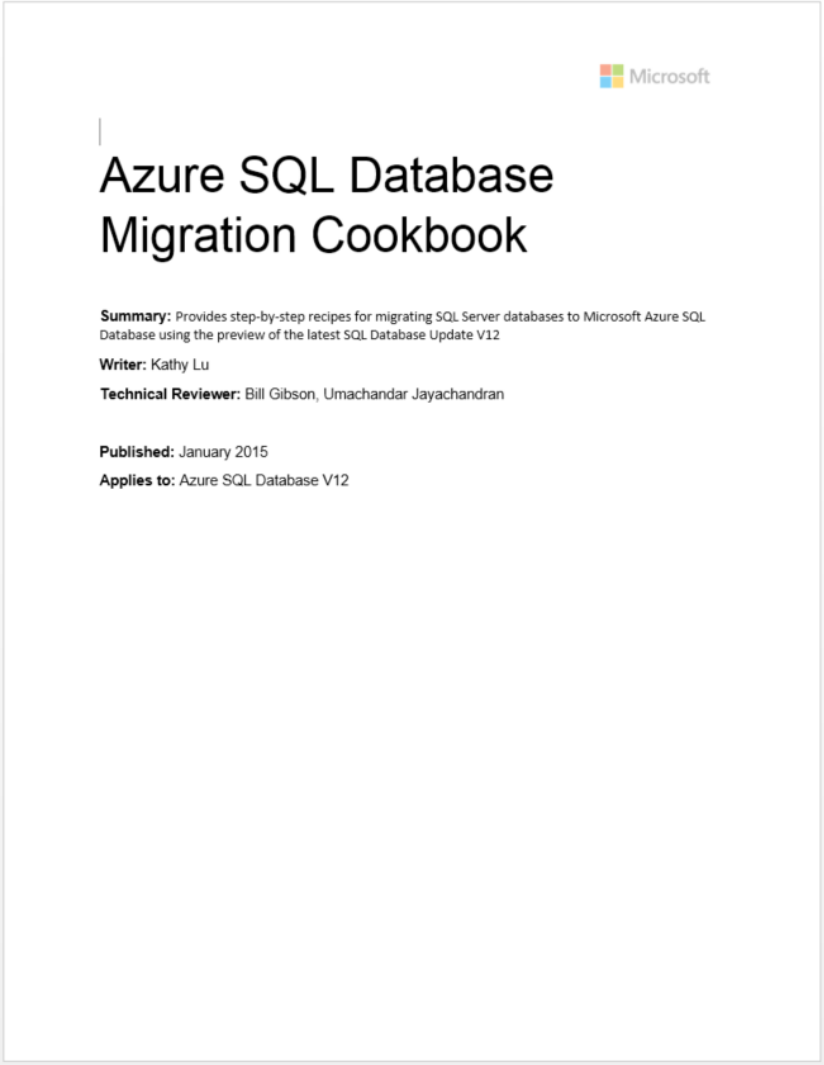
Method 3



Update the database schema offline using Visual Studio and SAMW, and then deploy it with SSMS

- Create a new database project using SSDT
- Import your database into the project
- Select the target database as Azure SQL Database v12 preview
- Generate the scripts
- Process the scripts using SAMW
- Correct any errors reported by SAMW and deploy the scripts to Azure SQL Database

Migration Cookbook



Microsoft

Azure SQL Database Migration Cookbook

Summary: Provides step-by-step recipes for migrating SQL Server databases to Microsoft Azure SQL Database using the preview of the latest SQL Database Update V12

Writer: Kathy Lu

Technical Reviewer: Bill Gibson, Umachandar Jayachandran

Published: January 2015

Applies to: Azure SQL Database V12

Migrate an on-premises SQL Server database to Azure SQL Database (v12).

The Migration Cookbook describes various approaches you can use to migrate an on-premises SQL Server database to the latest Azure SQL Database Update (v12).

Download: http://aka.ms/azuresqlmigration

Temporal Tables

Allows you to track and analyze the history of changes in data

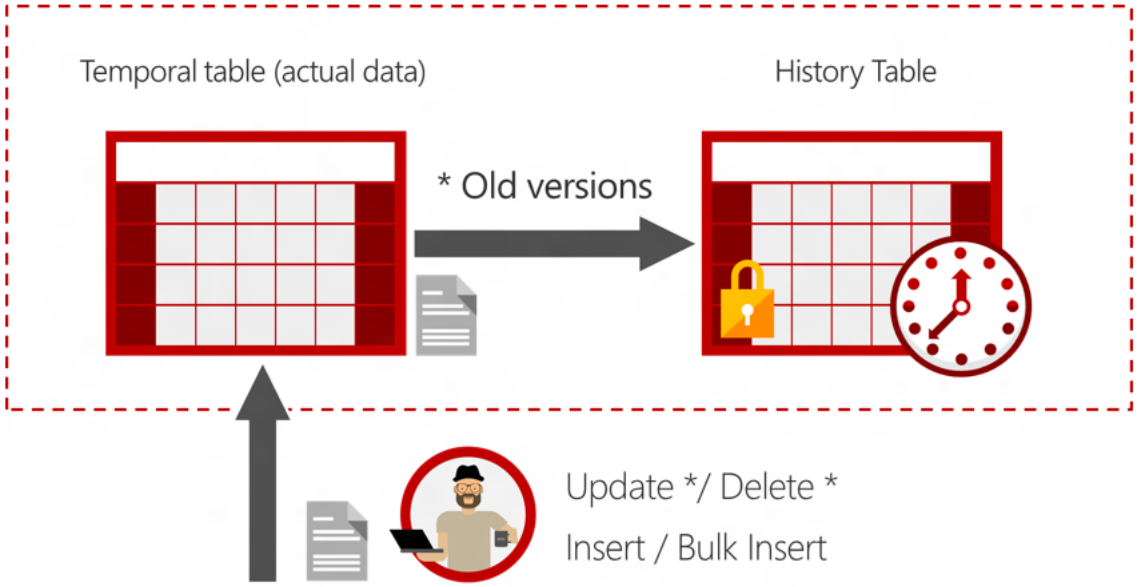
Useful for:

- Auditing data changes
- Reconstructing state of the data as of any time in the past
- Calculating trends over time
- Maintaining a slowly changing dimension for decision support applications

Requires no custom coding to support these scenarios

How does it work?

Implemented as a pair of tables, one current, one historical



The history table cannot have data inserted or deleted from it directly and its schema cannot be directly modified.

Creating a temporal table

Can be created via:

- Sql Server Management Studio
- Sql Server Data Tools
- T-SQL

```
CREATE TABLE WebsiteUserInfo
(
    [UserID] int NOT NULL PRIMARY KEY CLUSTERED
  , [UserName] nvarchar(100) NOT NULL
  , [PagesVisited] int NOT NULL
  , [ValidFrom] datetime2 (0) GENERATED ALWAYS AS ROW START
  , [ValidTo] datetime2 (0) GENERATED ALWAYS AS ROW END
  , PERIOD FOR SYSTEM_TIME (ValidFrom, ValidTo)
 )
 WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE =
dbo.WebsiteUserInfoHistory));
```

Reading from the history table

Data can be read from the history table in multiple ways…

To see the top 10 users ordered by the number of visited web pages as of an hour ago

- Use the AS OF clause

```
DECLARE @hourAgo datetime2 = DATEADD(HOUR, -1, SYSUTCDATETIME());
SELECT TOP 10 * FROM dbo.WebsiteUserInfo FOR SYSTEM_TIME AS OF @hourAgo
ORDER BY PagesVisited DESC
```

Reading from the history table

To perform basic statistical analysis for the previous day

Use the BETWEEN … AND … clause

```
DECLARE @twoDaysAgo datetime2 = DATEADD(DAY, -2, SYSUTCDATETIME());
DECLARE @aDayAgo datetime2 = DATEADD(DAY, -1, SYSUTCDATETIME());
SELECT UserID, SUM (PagesVisited) as TotalVisitedPages, AVG (PagesVisited) as
AverageVisitedPages,
MAX (PagesVisited) AS MaxVisitedPages, MIN (PagesVisited) AS MinVisitedPages,
STDEV (PagesVisited) as StDevVisitedPages
FROM dbo.WebsiteUserInfo
FOR SYSTEM_TIME BETWEEN @twoDaysAgo AND @aDayAgo
GROUP BY UserId
```

Retention History

Temporal Tables may increase database size more than regular tables

Particularly if

- You retain historical data for a long period of time
- You have an update or delete heavy data modification pattern

Developing a data retention policy for managing data in the history table is an important aspect of planning and managing the lifecycle of every temporal table

ALTER DATABASE <myDB> SET TEMPORAL_HISTORY_RETENTION ON

Setting Retention History

Ensure retention policy is enabled at the database level

```
ALTER DATABASE <myDB>
SET TEMPORAL_HISTORY_RETENTION ON
```

Retention policy can be set when the table is created or can be defined or altered after table creation

```
ALTER TABLE dbo.WebsiteUserInfo
SET (SYSTEM_VERSIONING = ON (HISTORY_RETENTION_PERIOD = 9 MONTHS));
```

JSON Support in SQL Database

Lets you work with JSON data easily and integrate your database with modern services

Built-in functions
ISJSON, JSON_VALUE
JSON_MODIFY, JSON_QUERY

```
[
  {
    "Number":"SO43659",
    "Date":"2011-05-31T00:00:00"
    "AccountNumber":"AW29825",
    "Price":59.99,
    "Quantity":1
  },
  {
    "Number":"SO43661",
    "Date":"2011-06-01T00:00:00“
    "AccountNumber":"AW73565“,
    "Price":24.99,
    "Quantity":3
  }
]
```

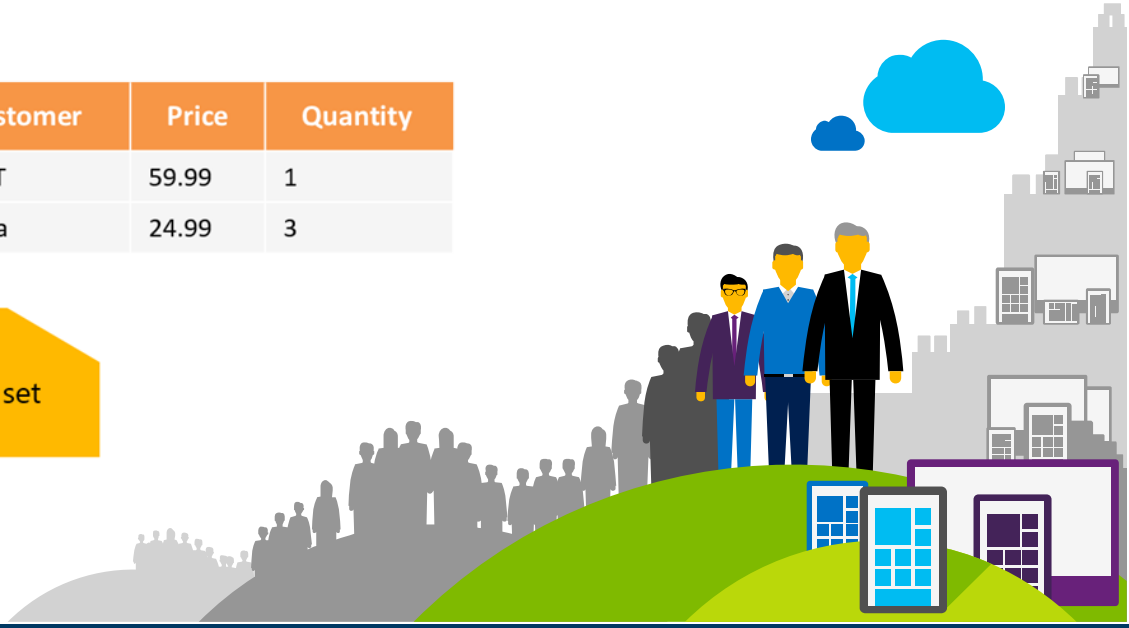
OPENJSON

Transforms JSON text to table

Number	Date	Customer	Price	Quantity
SO43659	2011-05-31T00:00:00	MSFT	59.99	1
SO43661	2011-06-01T00:00:00	Nokia	24.99	3

FOR JSON

Formats result set as JSON text



Formatting data as JSON

Format your database content as JSON directly in a SQL query

```
SELECT CustomerName, PhoneNumber, FaxNumber
FROM Sales.Customers
FOR JSON PATH
```


```
[
    {
        "CustomerName":"Eric Torres",
        "PhoneNumber":"(307) 555-0100",
        "FaxNumber":"(307) 555-0101"
    },
    {
        "CustomerName":"Cosmina Vlad",
        "PhoneNumber":"(505) 555-0100",
        "FaxNumber":"(505) 555-0101"
    },
    {
        "CustomerName":"Bala Dixit"
        ,"PhoneNumber":"(209) 555-0100",
        "FaxNumber":"(209) 555-0101"
    }
]
```

Querying JSON data

JSON functions let you use JSON data in any SQL query

Id	Data
1	{"Price":50, "Color":"White","tags":["toy","children","games"]}

```
SELECT Id, JSON_VALUE(Data, '$.Color'), JSON_QUERY(Data, '$.tags')
FROM Products
WHERE JSON_VALUE(Data, '$.Color') = 'White'
```

1	White	["toy","children","games"]

Modifying JSON data

JSON functions let you use JSON data in any SQL query

Id	Data
1	{"Price":50, "Color":"White","tags":["toy","children","games"]}

```
UPDATE Products
SET Data = JSON_MODIFY(Data, '$.Price', 60)
WHERE Id = 1
```

© 2016 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.

Layers of protection

Azure Active Directory	Encryption	Data protection
Row-Level Security	Auditing and threat detection	Stringent regulatory compliance

- Centrally manage and control identity and user access
- Encrypt a database, associated backups, and log files at rest—without changing your app
- Protect data at rest, in motion, or in use with encryption that uses the National Institute of Standards and Technology (NIST) Advanced Encryption Standard 256 cipher
- Control which users can access specific row-level data and how much specific data they can see
- Get notified of potential threats with auditing tools and anomalous activity alerting
- Leverage ISO/IEC 27001/27002, Fed RAMP/FISMA, SOC, HIPPA, and PCI DSS compliance

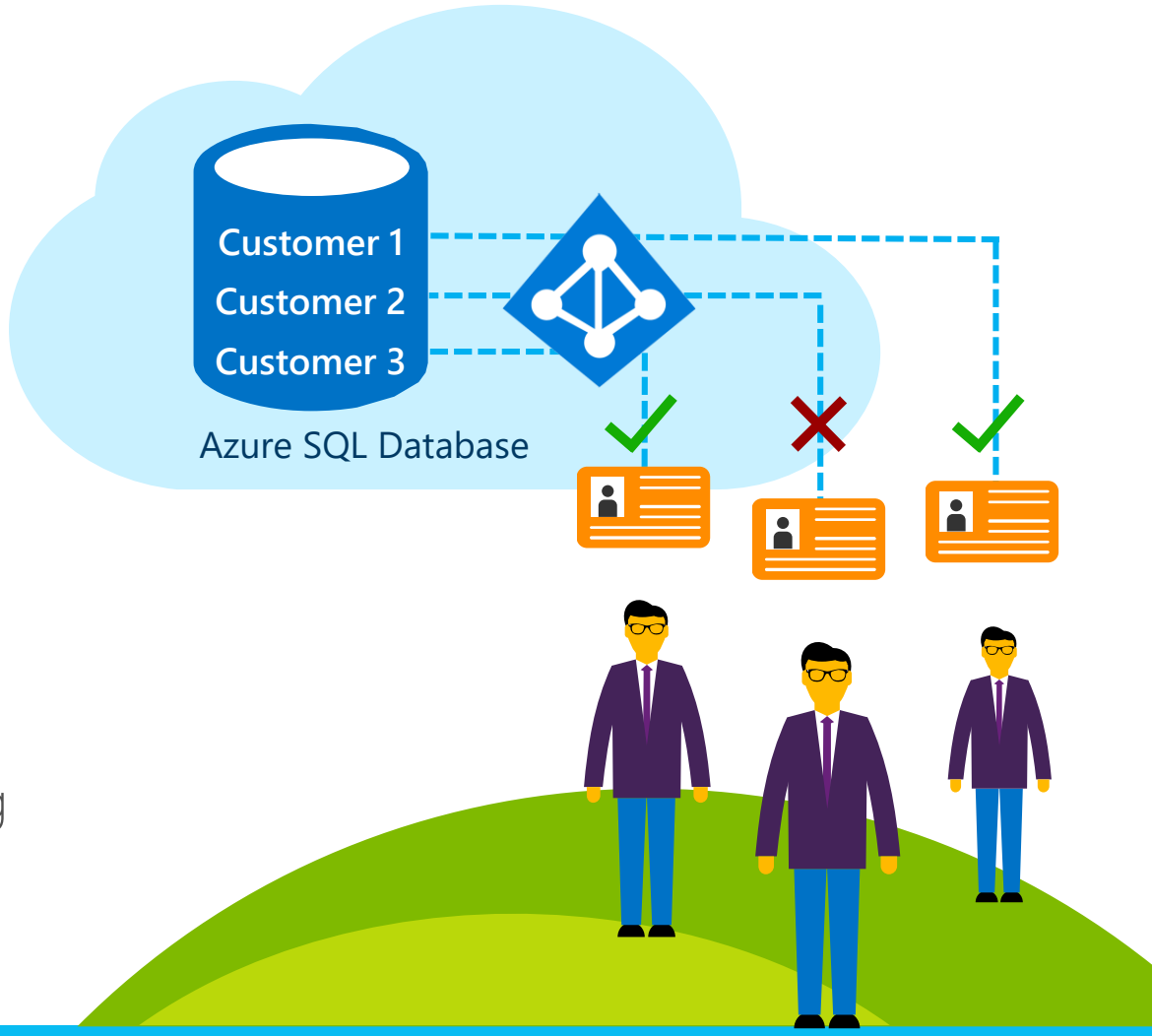
Azure Active Directory authentication

Manage user identities in one location.

Use Azure Active Directory user identities and groups to enable access to Azure SQL Database and other Microsoft services.

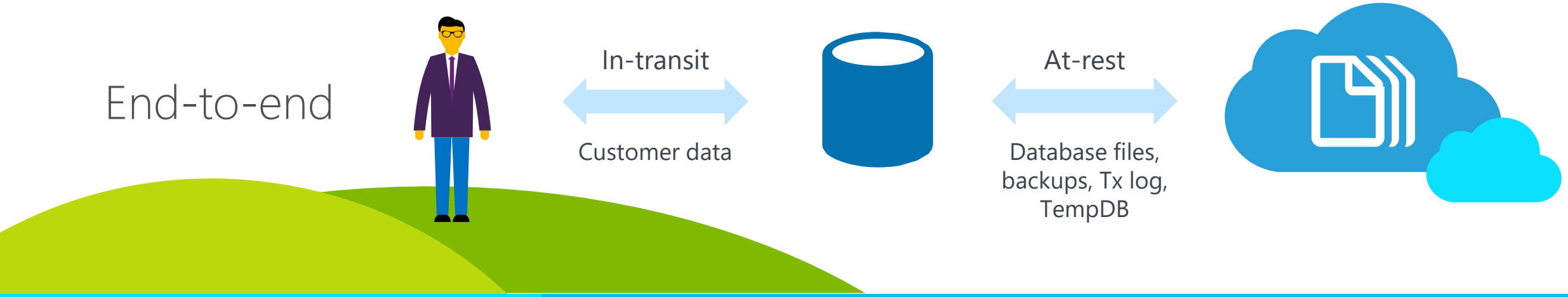
Benefits include:

- Alternative to SQL Server authentication
- Limits proliferation of user identities across databases
- Allows password rotation in a single place
- Enables management of database permissions by using external Azure Active Directory groups
- Elimination of the need to store passwords



Azure SQL Database encryption: overview

Encryption type	Type	Customer value
Encryption-in-transit	Transport Layer Security (TLS) from the client to the server	**Protects data between client and server against snooping and man-in-the-middle attacks.** Azure SQL Database is phasing out Secure Sockets Layer (SSL) 3.0 and TLS 1.0 in favor of TLS 1.2.
Encryption-at-rest	Transparent Data Encryption (TDE) for Azure SQL Database	Protects data on the disk. Key management is done by Azure, which **makes it easier to obtain compliance.**
Encryption-end-to-end	Always Encrypted for client-side column encryption	Data is protected end-to-end, but the application is aware of encrypted columns. **This is used in the absence of data masking and TDE for compliance-related scenarios.**



Transparent Data Encryption

All customer data encrypted at rest

Encrypted at rest, in flight, and while in use

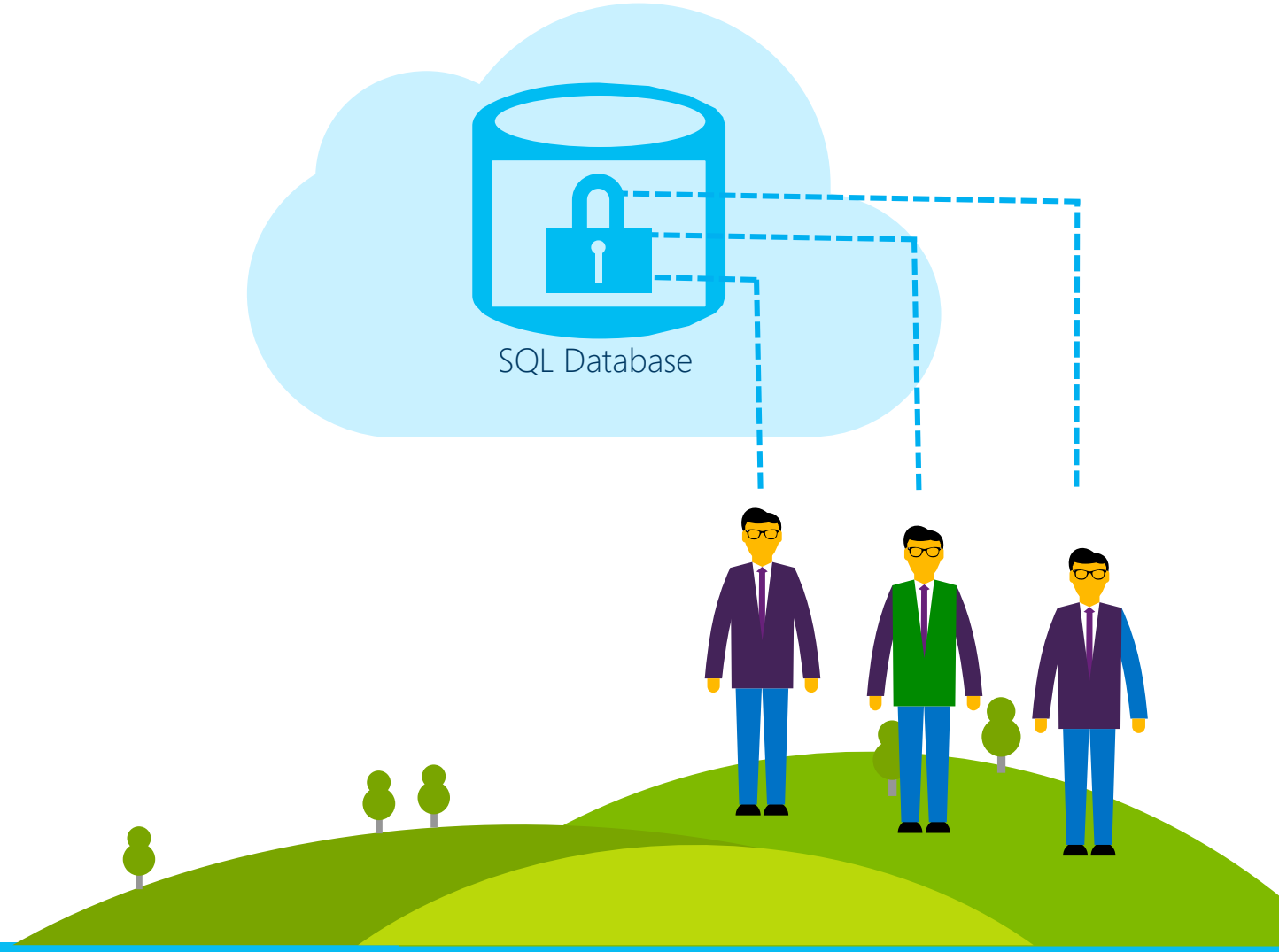
SQL Database service manages your keys

Keep application changes to a minimum

Encryption/decryption of data done transparently in TCE-enabled client driver

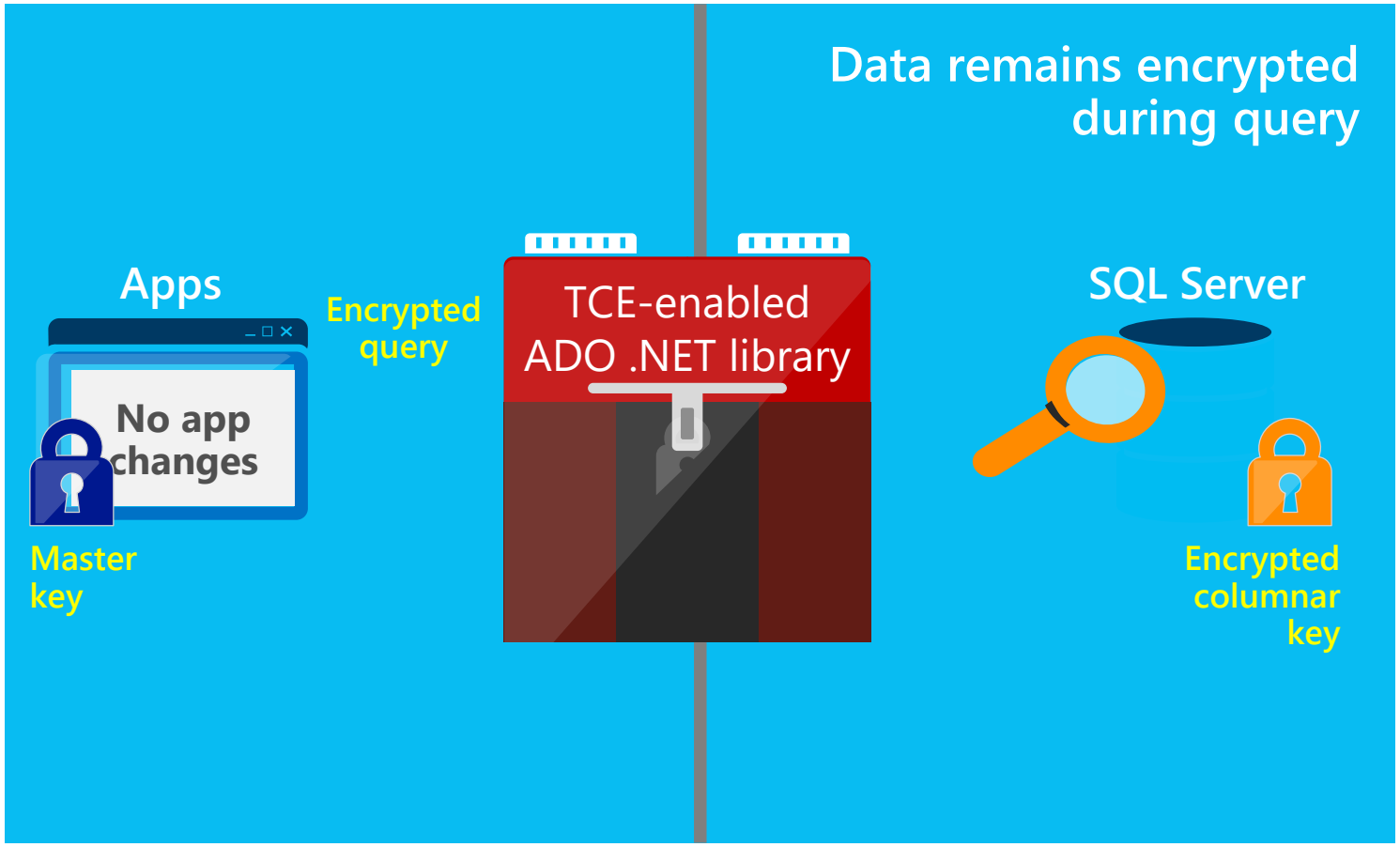
Support for equality operations (include joins) on encrypted data

Azure manages encryption keys



Always Encrypted

Protect data at rest and in motion, on-premises and in the cloud



Capability

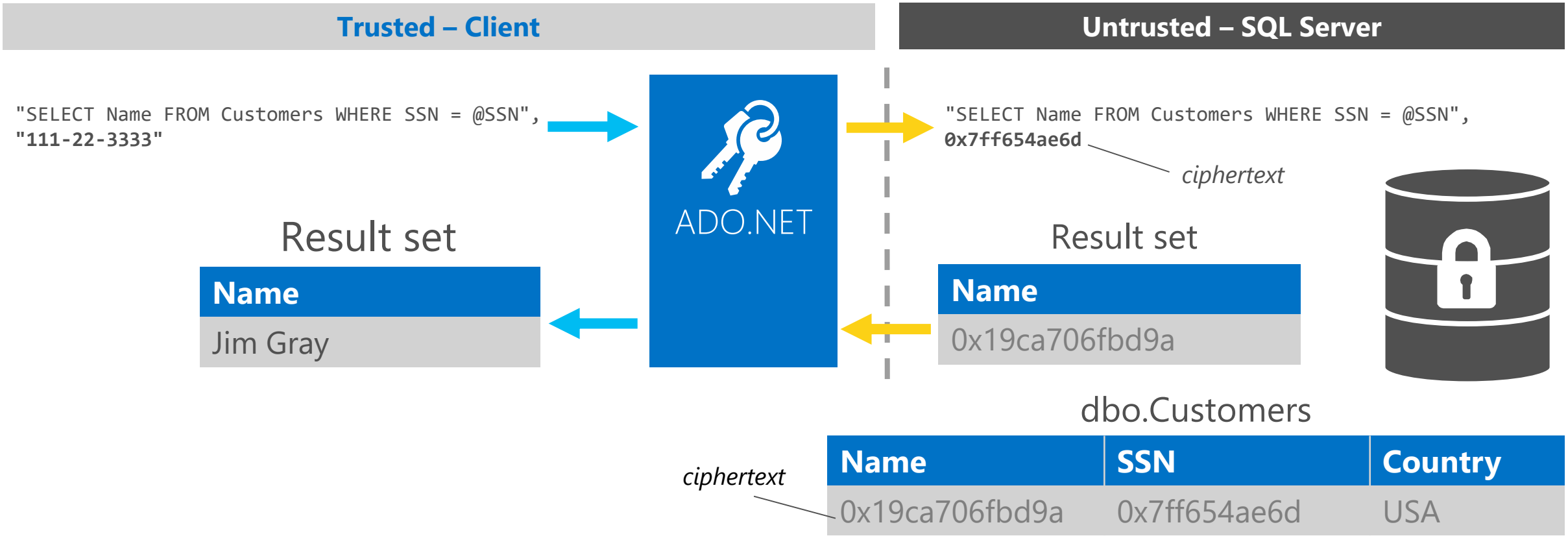
Transparent client-side encryption, while SQL Server executes T-SQL queries on encrypted data

Benefits

- Sensitive data remains encrypted and queryable at all times on-premises and in the cloud
- Unauthorized users never have access to data or keys
- No application changes

How it works

Encrypted sensitive data and its corresponding keys are never seen in plaintext in SQL Server



Always Encrypted

Protect sensitive data in use from high-privileged yet unauthorized SQL users both on-premises and in the cloud

Client-side encryption

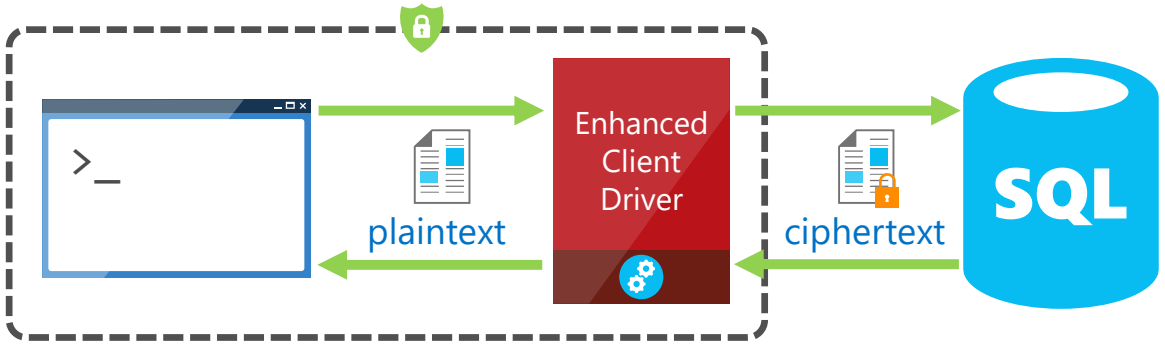
Sensitive data and related encryption keys are never revealed to the database engine

Encryption transparency

Client driver transparently encrypts query parameters and decrypts encrypted results

Queries on encrypted data

Support for equality comparison on columns encrypted using deterministic encryption



Limitations of Always Encrypted

Reduced functionality of queries on encrypted columns

Encrypted columns only allow equality comparisons

Data needs to be moved out of the database for initial encryption and key rotation

This process can be time-consuming and prone to network errors

Row-level security

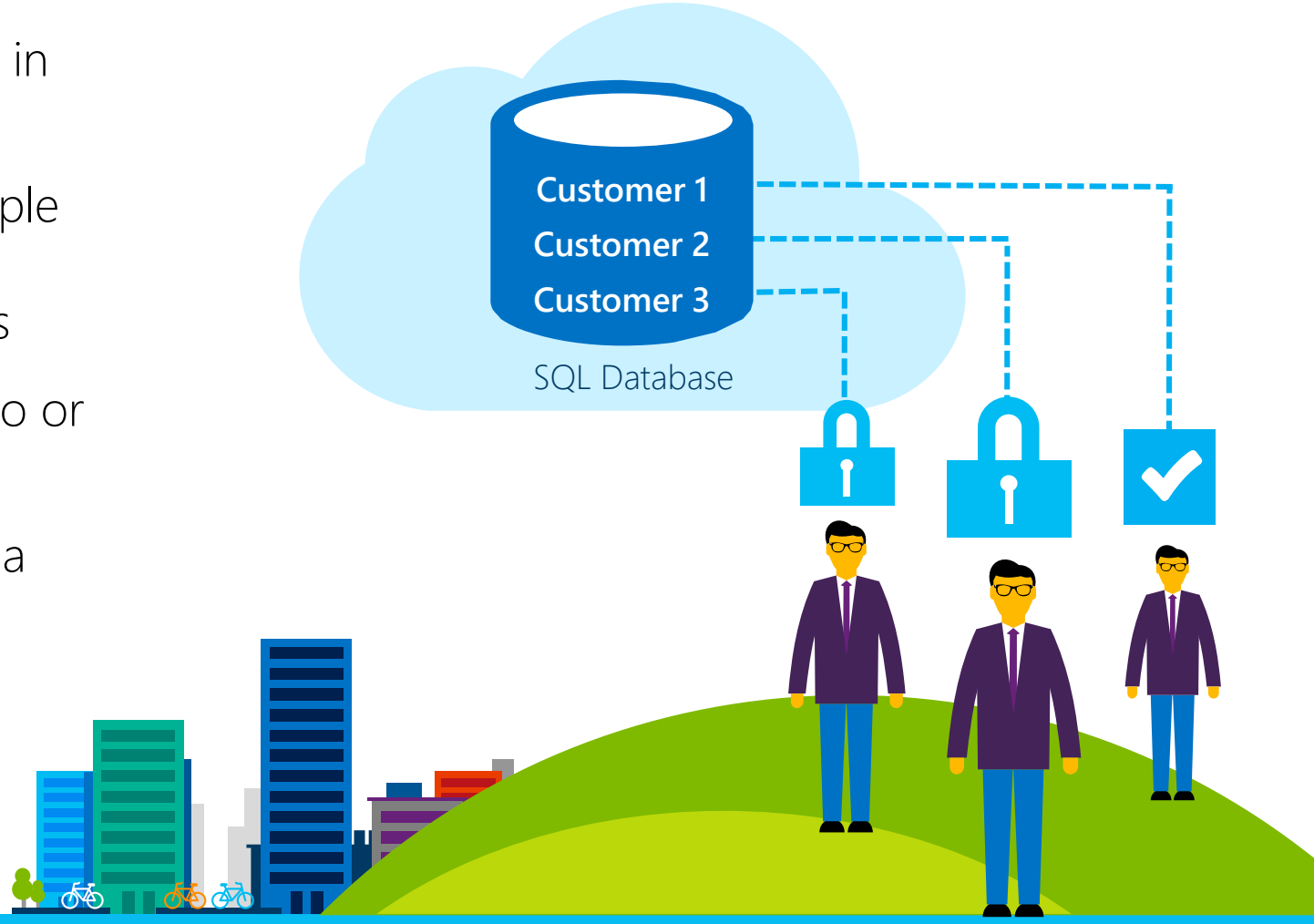
Protect data privacy by ensuring the right access across rows

Fine-grained access control over specific rows in database table

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

Administer via SQL Server Management Studio or SQL Server Data Tools

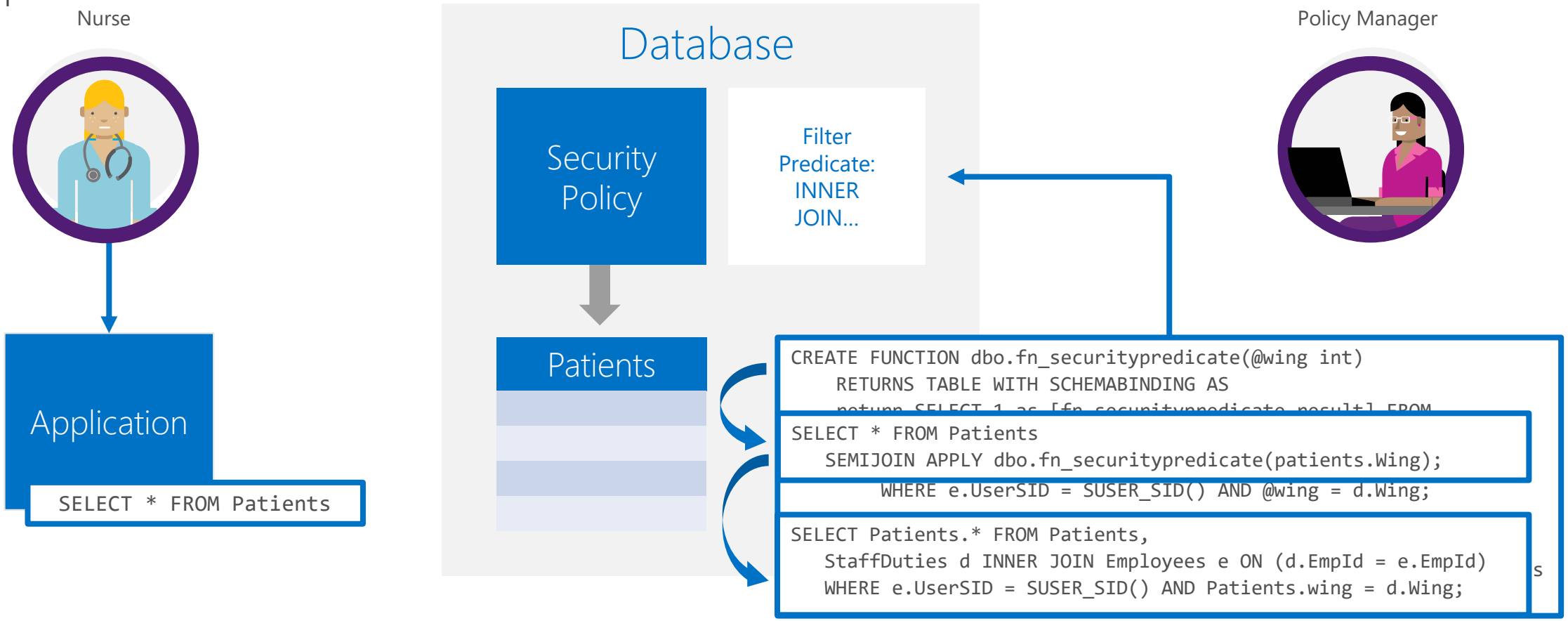
Enforcement logic inside database and schema bound to table



RLS in Three Steps

Three

1. Policy Manager defines security predicate and applies it to Patients table
2. Nurse (or any other) queries Patients table
3. Security Policy injects filter predicate into T-SQL, binding the predicate to the Patients table



Dynamic data masking

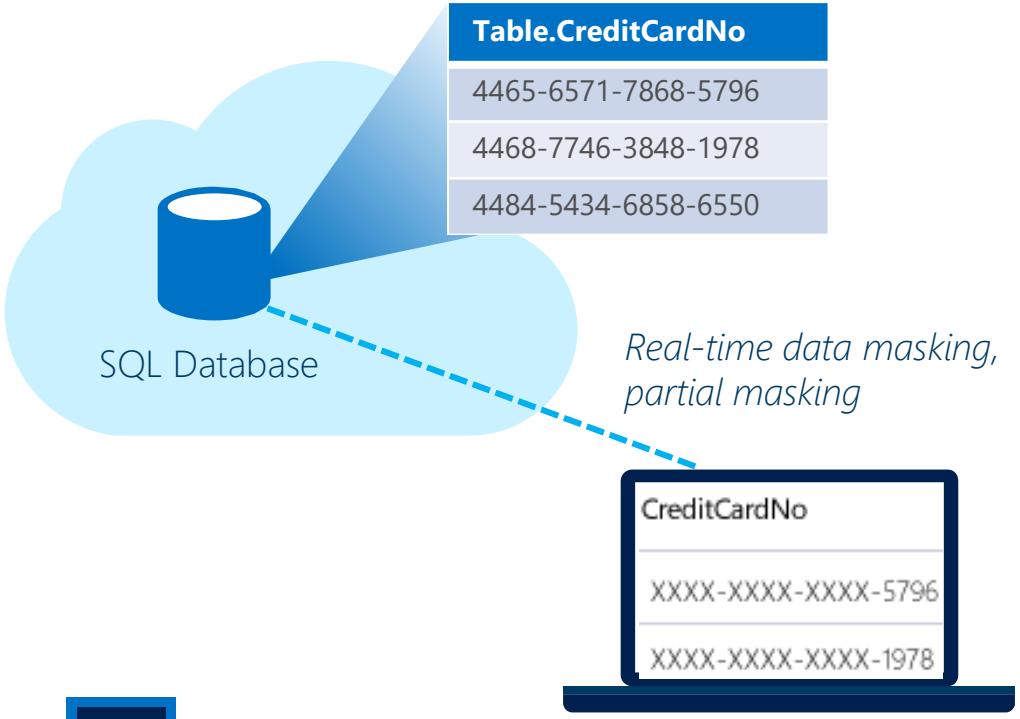
Prevent abuse of sensitive data by hiding it from users

Configuration made easy in new Azure Portal

Policy-driven at table and column level, for a defined set of users

Data masking applied in real-time to query results based on policy

Multiple masking functions available, such as full or partial, for various sensitive data categories (credit card numbers, SSN, etc.)



Status: Public preview

Dynamic data masking

1) The dba defines a masking policy on the sensitive data in the Employee table
2) The dba defines masking rules for the sensitive data in the Employee table
3) The dba authorizes the Security Officer to see sensitive data in results

```
ALTER TABLE [Employee] ALTER COLUMN [SocialSecurityNumber]
ADD MASKED WITH (FUNCTION = 'SSN()'

ALTER TABLE [Employee] ALTER COLUMN [Email]
ADD MASKED WITH (FUNCTION = 'EMAIL()')

ALTER TABLE [Employee] ALTER COLUMN [Salary]
ADD MASKED WITH (FUNCTION = 'RANDOM(1,20000)')

GRANT UNMASK to admin1
```

Security Officer

```
SELECT [Name],
       [SocialSecurityNumber],
       [Email],
       [Salary]
FROM [Employee]
```

other logon

	First Name	Social Security Number	Email	Salary
1	LILA	XXX-XX-XX37	lXX@XXXX.net	8940
2	JAMIE	XXX-XX-XX14	jXX@XXXX.com	19582
3	SHELLEY	XXX-XX-XX28	sXX@XXXX.net	3713
4	MARCELLA	XXX-XX-XX65	mXX@XXXX.net	11572
5	GILBERT	XXX-XX-XX87	gXX@XXXX.net	4487

admin1 logon

	First Name	Social Security Num...	Email	Salary
1	LILA	758-10-9637	lila.barnett@comcast.net	1012794
2	JAMIE	113-29-4314	jamie.brown@ntlworld.com	1025713
3	SHELLEY	550-72-2028	shelley.lynn@charter.net	1040131
4	MARCELLA	903-94-5665	marcella.estrada@comcast.net	1040753
5	GILBERT	376-79-4787	gilbert.juarez@verizon.net	1041308

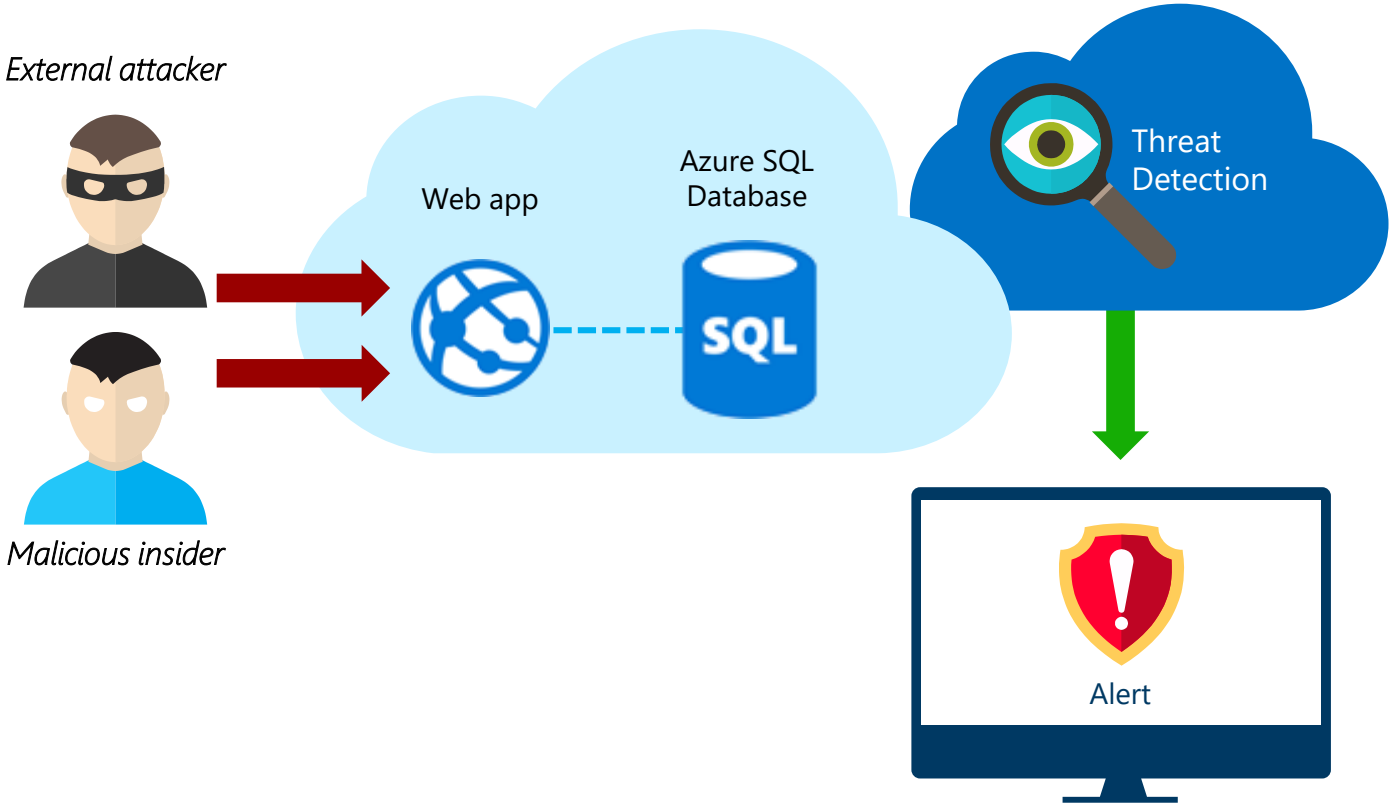
Threat detection

Detects anomalous database activities that could indicate potential threat

Configure threat detection policy in Azure Portal

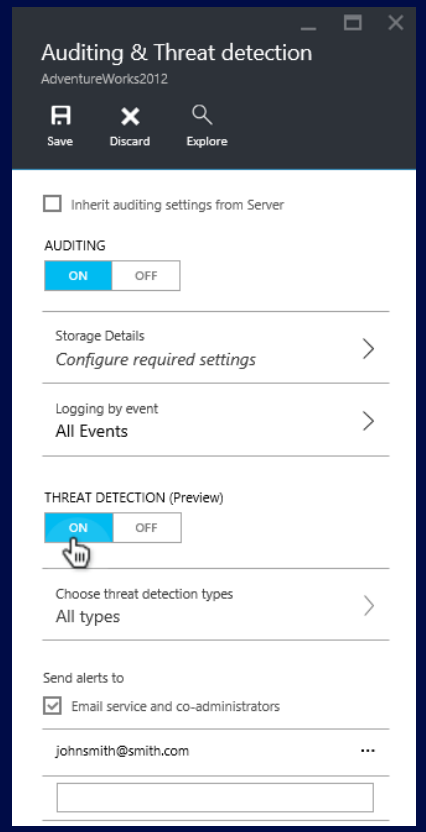
Receive alerts from multiple database-threat detectors that identify anomalous activities

Explore audit log around the time of an event

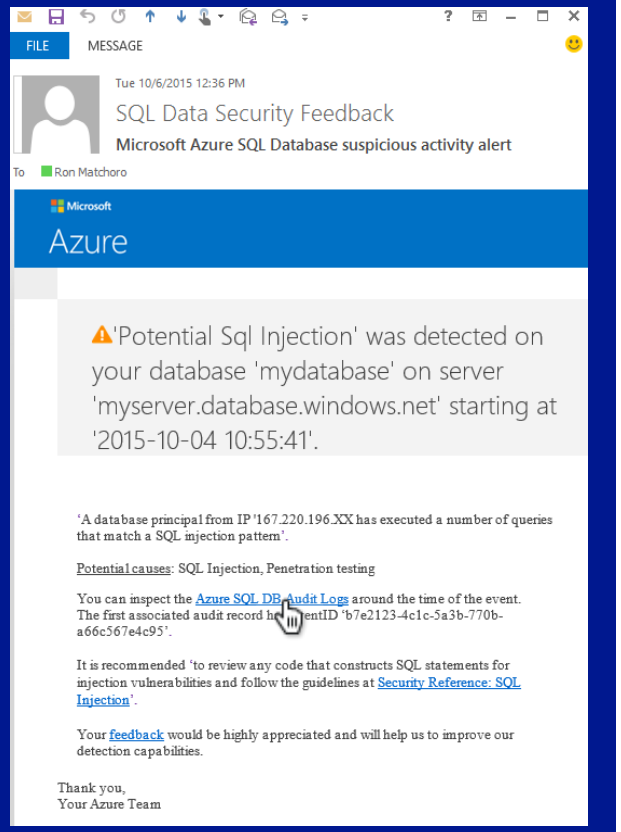


How it works

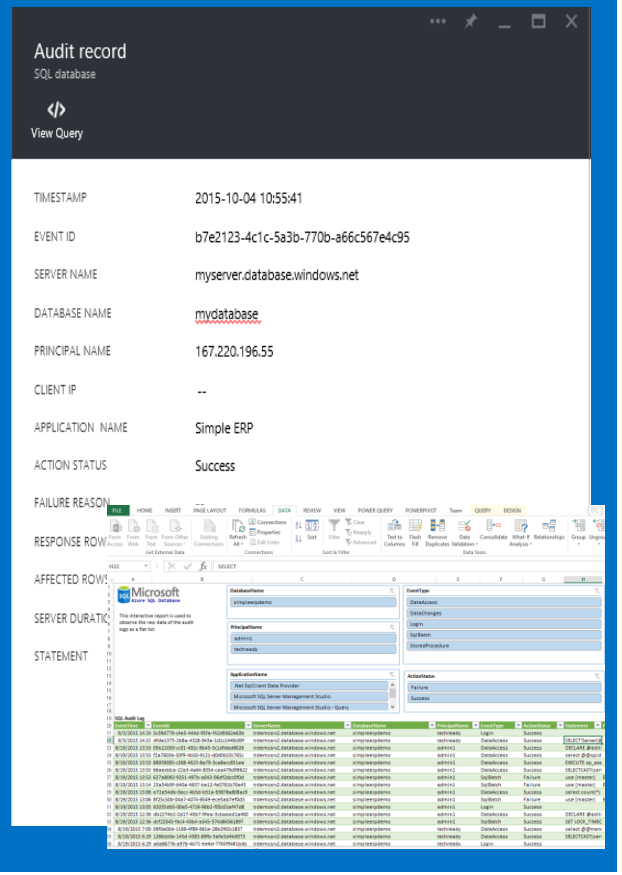
Set up



Alert



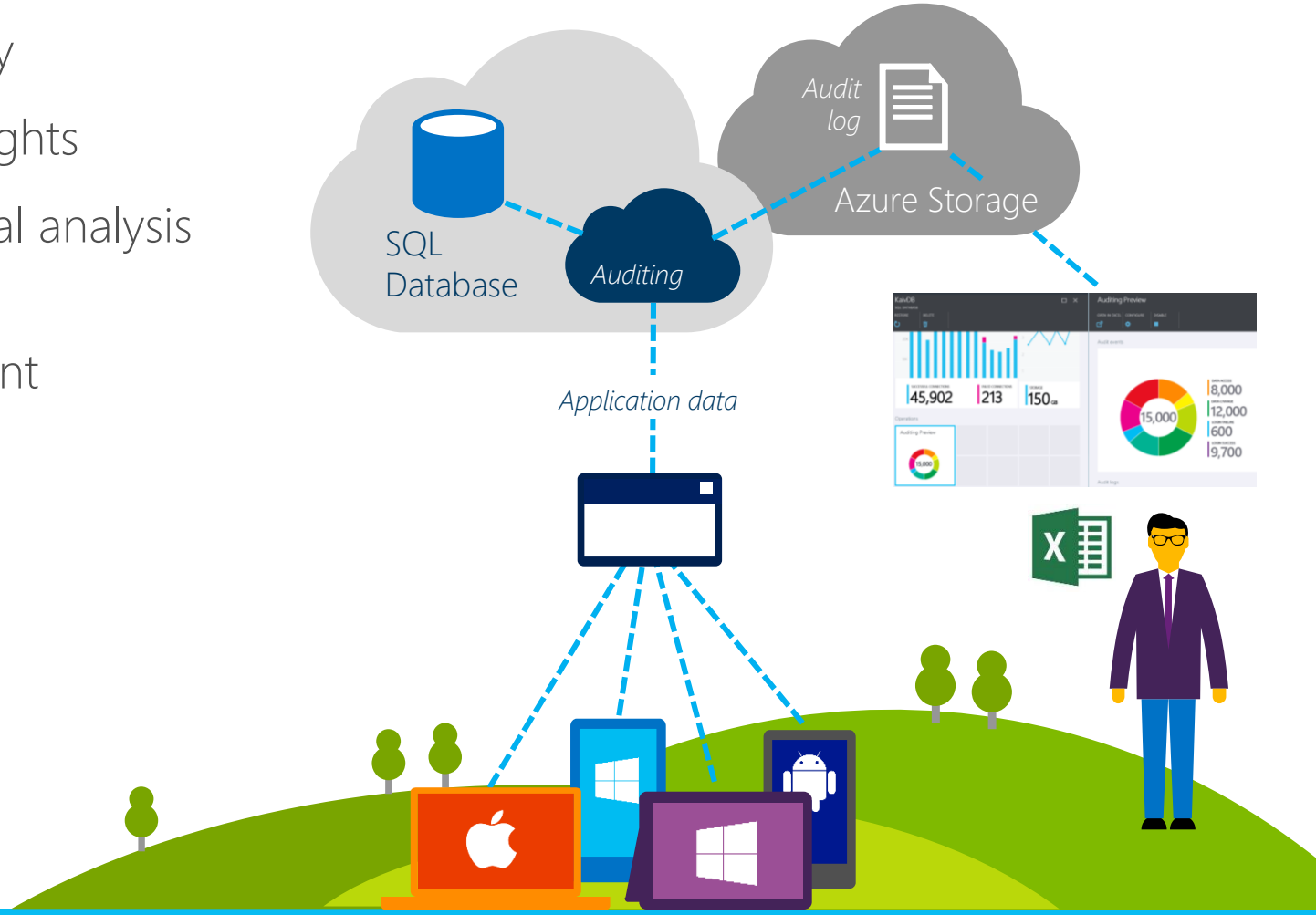
Explore



Auditing

Gain insight into database events and streamline compliance-related tasks

- Configurable to track and log database activity
- Dashboard views in portal for at-a-glance insights
- Pre-defined Power View reports for deep visual analysis on audit log data
- Audit logs reside in your Azure Storage account
- Available in Basic, Standard, and Premium
- Access via Azure portal



Compliance

ISO/IEC 27001

FedRAMP/FISMA

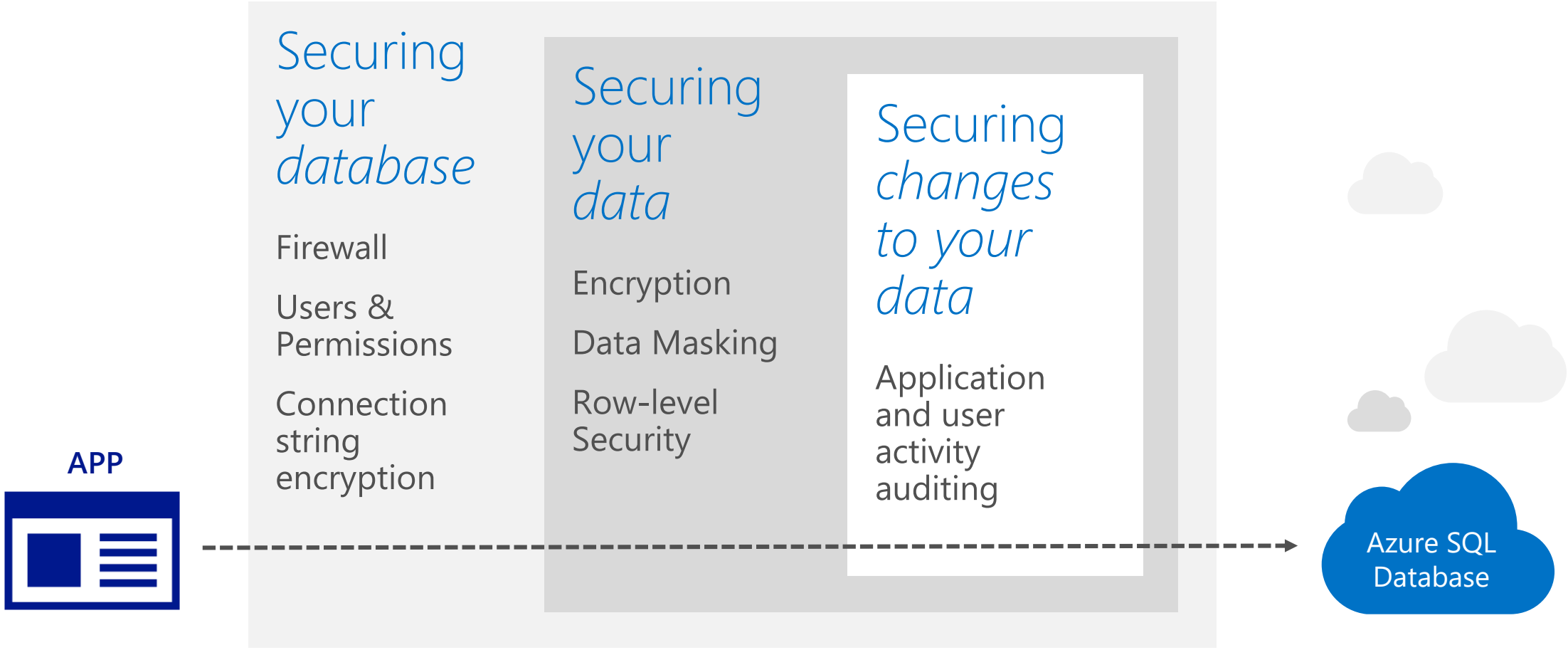
PCI DSS Level 1

SOC 1 Type 2 and
SOC 2 Type 2

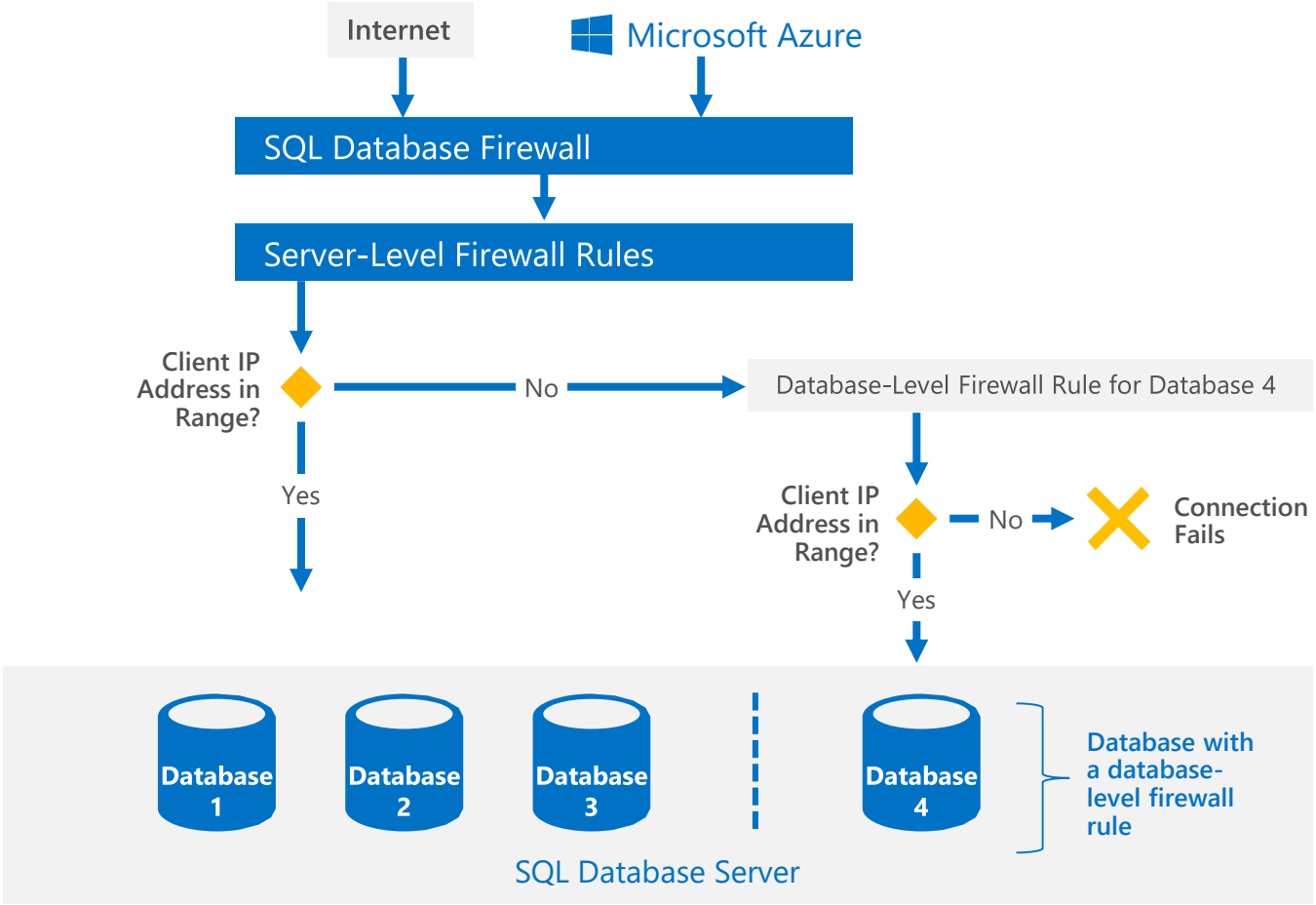
HIPAA
business associate
agreement (BAA)

EU Model Clauses

Layered Approach to Security



Securing your database with firewalls



Windows Azure Platform

Initially, all access to your Azure SQL Database server is blocked by the firewall.

In order to begin using your Azure SQL Database server, you must go to the Management Portal

Microsoft

Business Continuity and Disaster Recovery

Microsoft Azure

Learning objectives

Roles and responsibilities

Point-in-time restore

Geo-restore

Active geo-replication

Auto Failover Groups

Point-in-time restore

Automatic backup

Full backups weekly, different backup daily, log backups every 5 minutes

Daily and weekly backups automatically uploaded to geo-redundant Azure Storage

Self-service restore

Point-in-time up to a second granularity

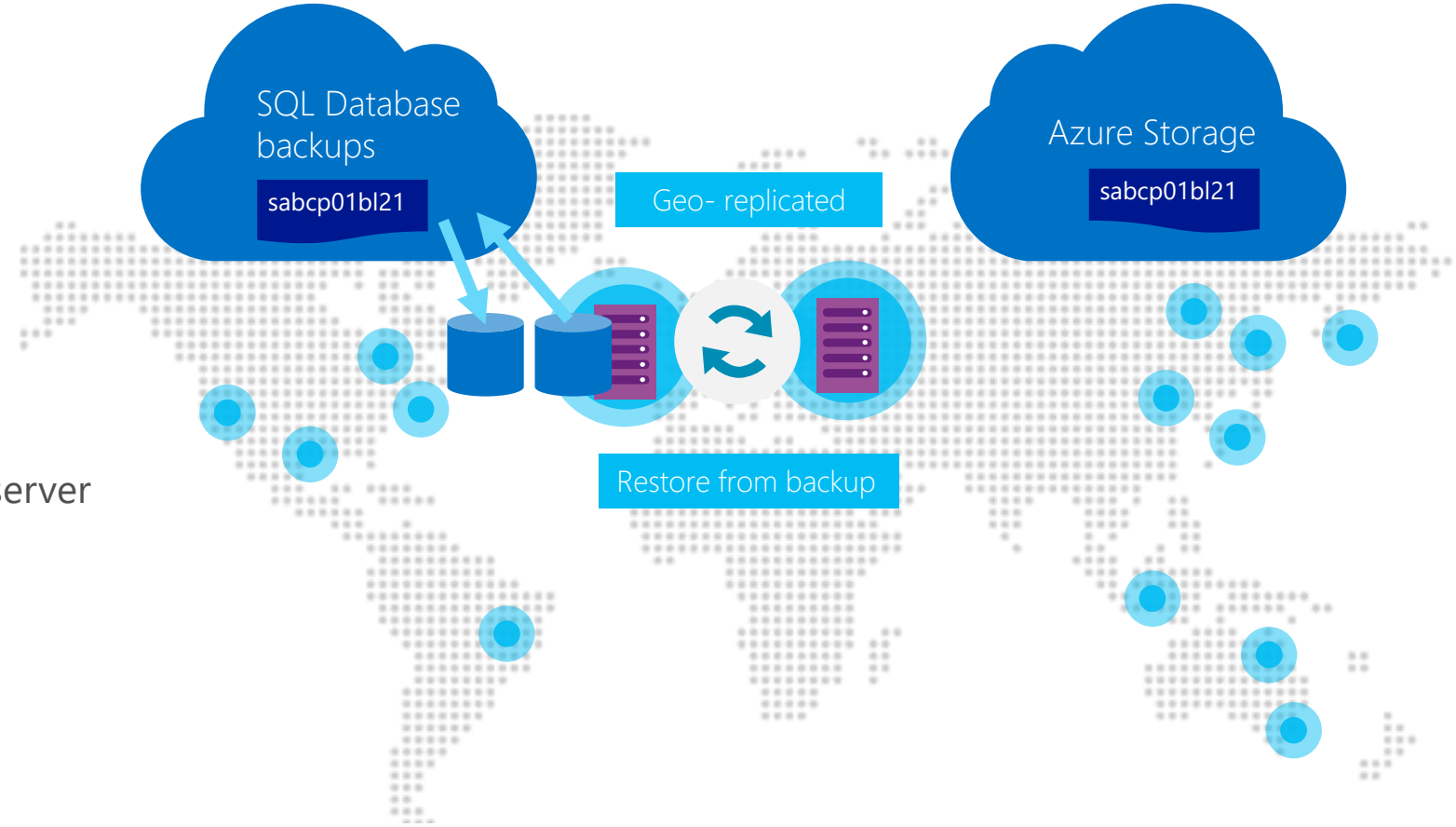
REST API, PowerShell, or Portal

Creates a new database in the same logical server

Tiered retention policy

Basic - 7 days
Standard - 35 days
Premium, Premium RS - 35 days

No additional cost to retain backups



Restore deleted database

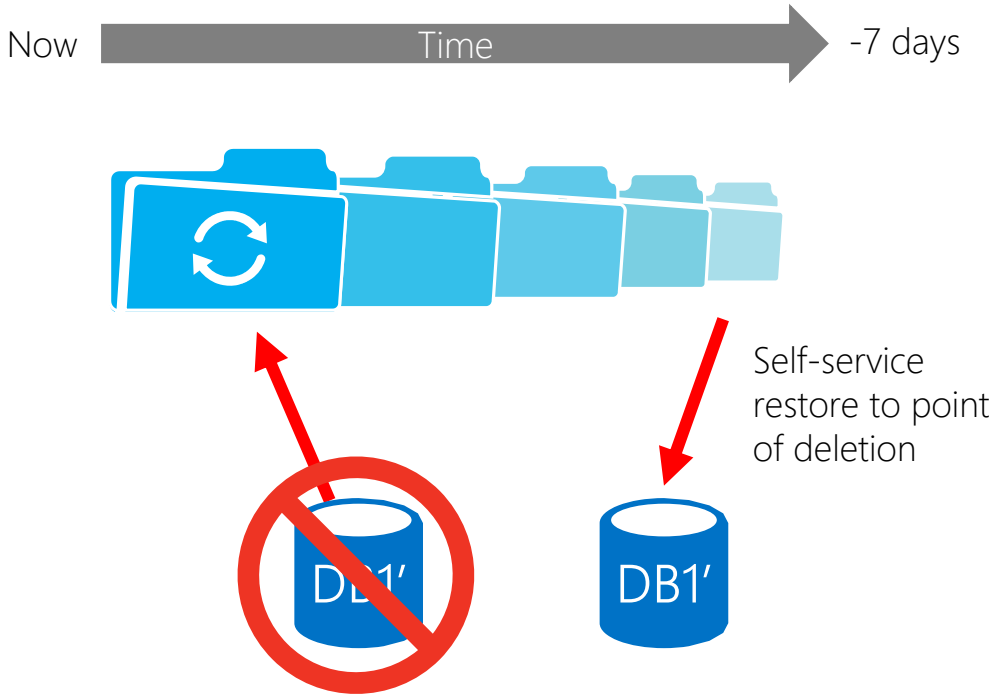
Recovery after accidental database deletion

Restores the database to the point of deletion (earlier backups are deleted)

Creates a new database on the server used by the original database

You can choose to failover to the restored database or use scripts to recover data

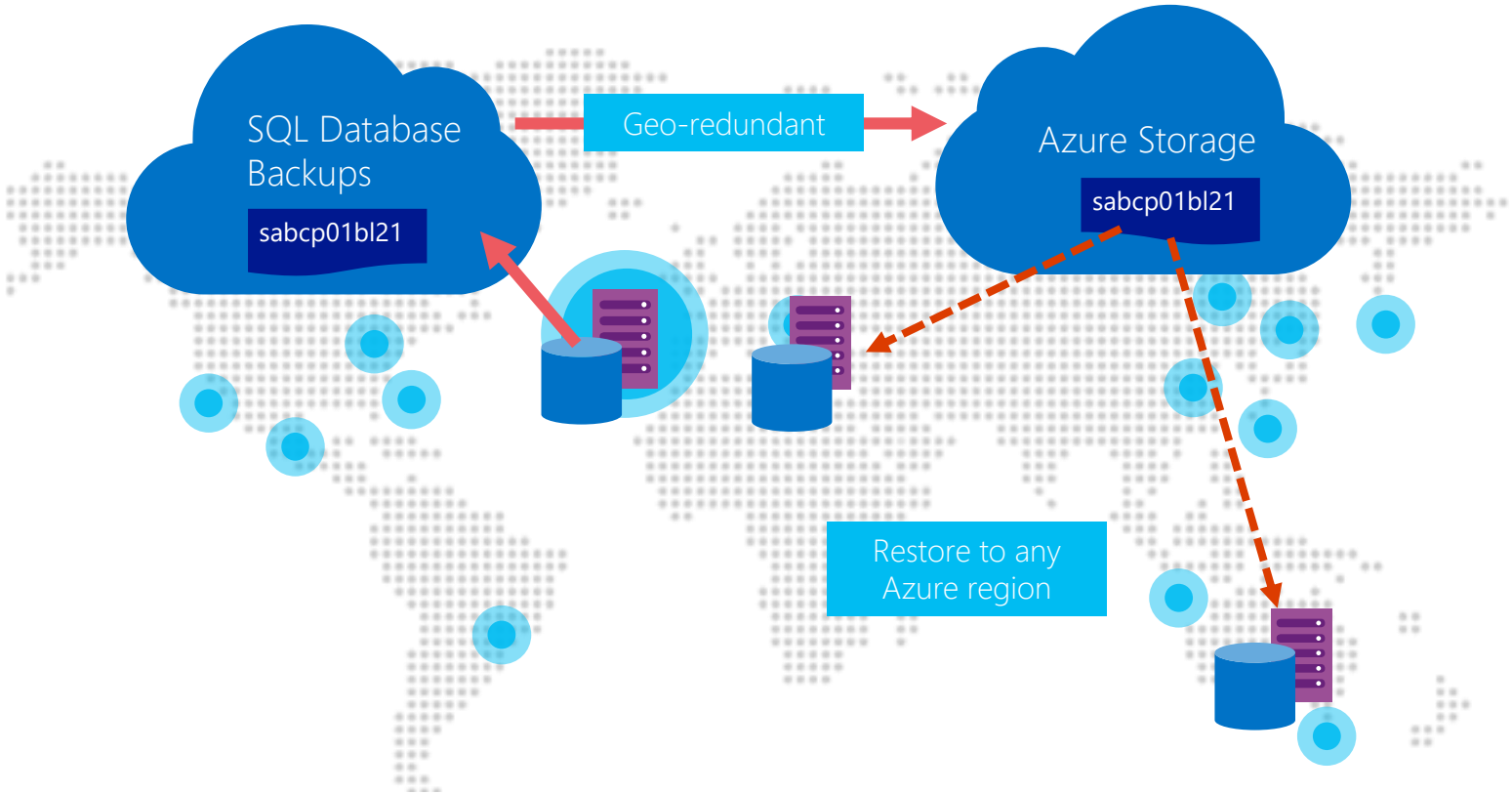
Backups retained for 7/35 days



Geo-restore protects from disaster

Built-in disaster recovery capability available for every database

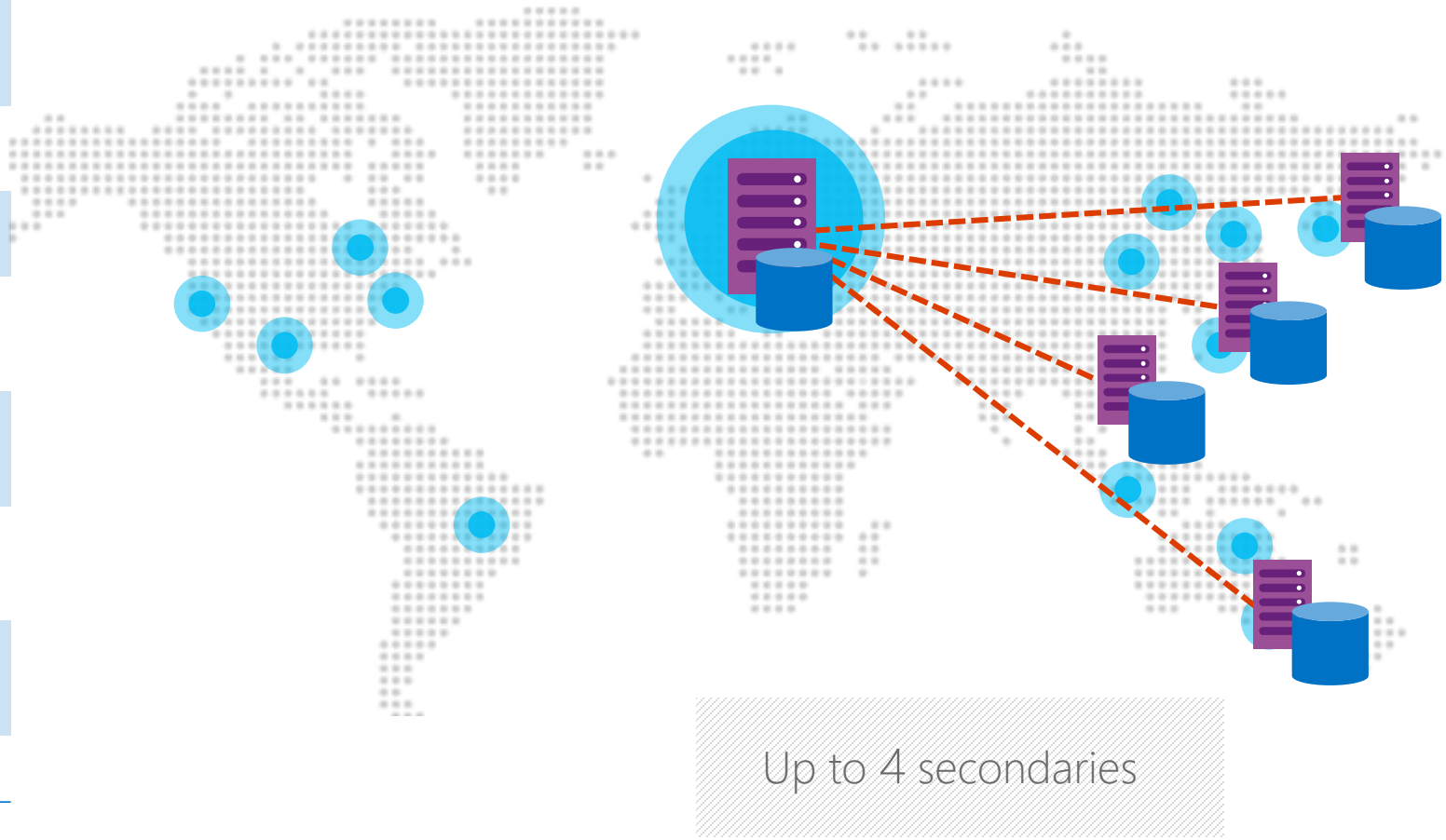
- Self-service restore API
- Restores last daily backup
- No extra cost, no capacity guarantee
- RTO>=24h, RPO=24h
- Database URL will change after restore



Active geo-replication

Mission critical business continuity

Service levels	Basic, Standard and Premium Self Service
Readable Secondaries	Up to 4
Regions available	Any Azure region
Replication	Automatic, Asynchronous
Manageability tools	REST API, PowerShell, T-SQL or Azure Portal
Recovery Time Objective (RTO)	<1 hour
Recovery Point Objective	<5 mins
Failover	On Demand



SQL Database Failover Groups

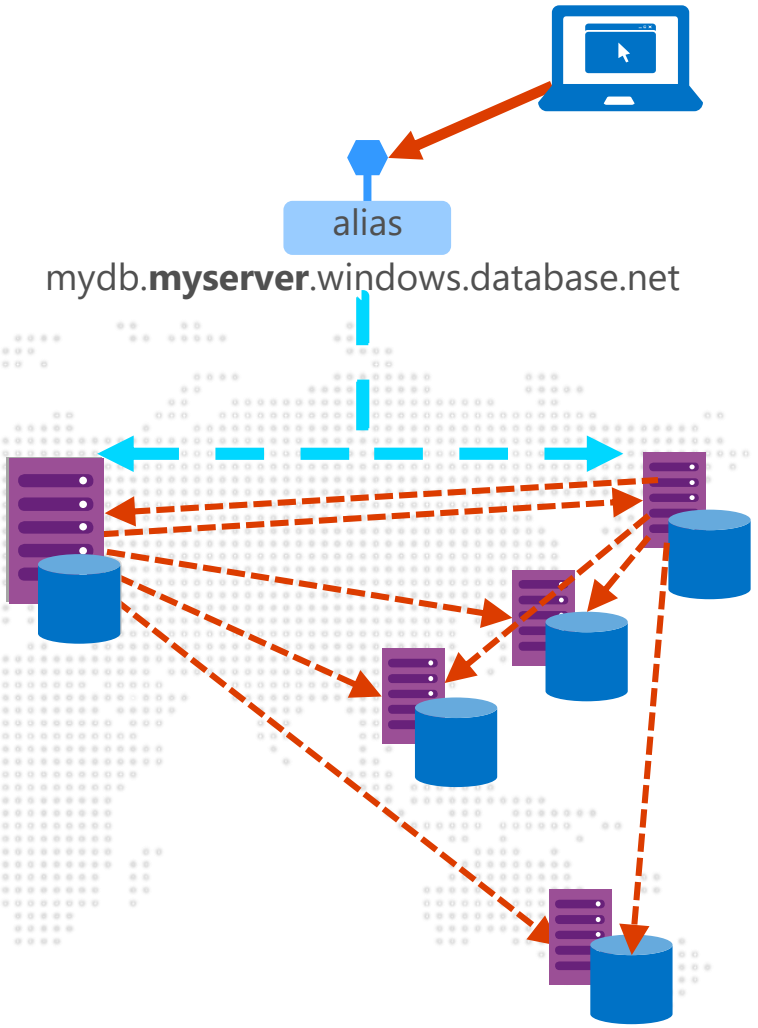
Extends geo-replication

Enable geo-replication for a group of databases within a server

Automatically or manually failover a group of databases

supports active-active setup for pools

All service tiers



**Currently in private preview - Microsoft Confidential – Shared Under NDA Only*

SQL Database Failover Groups

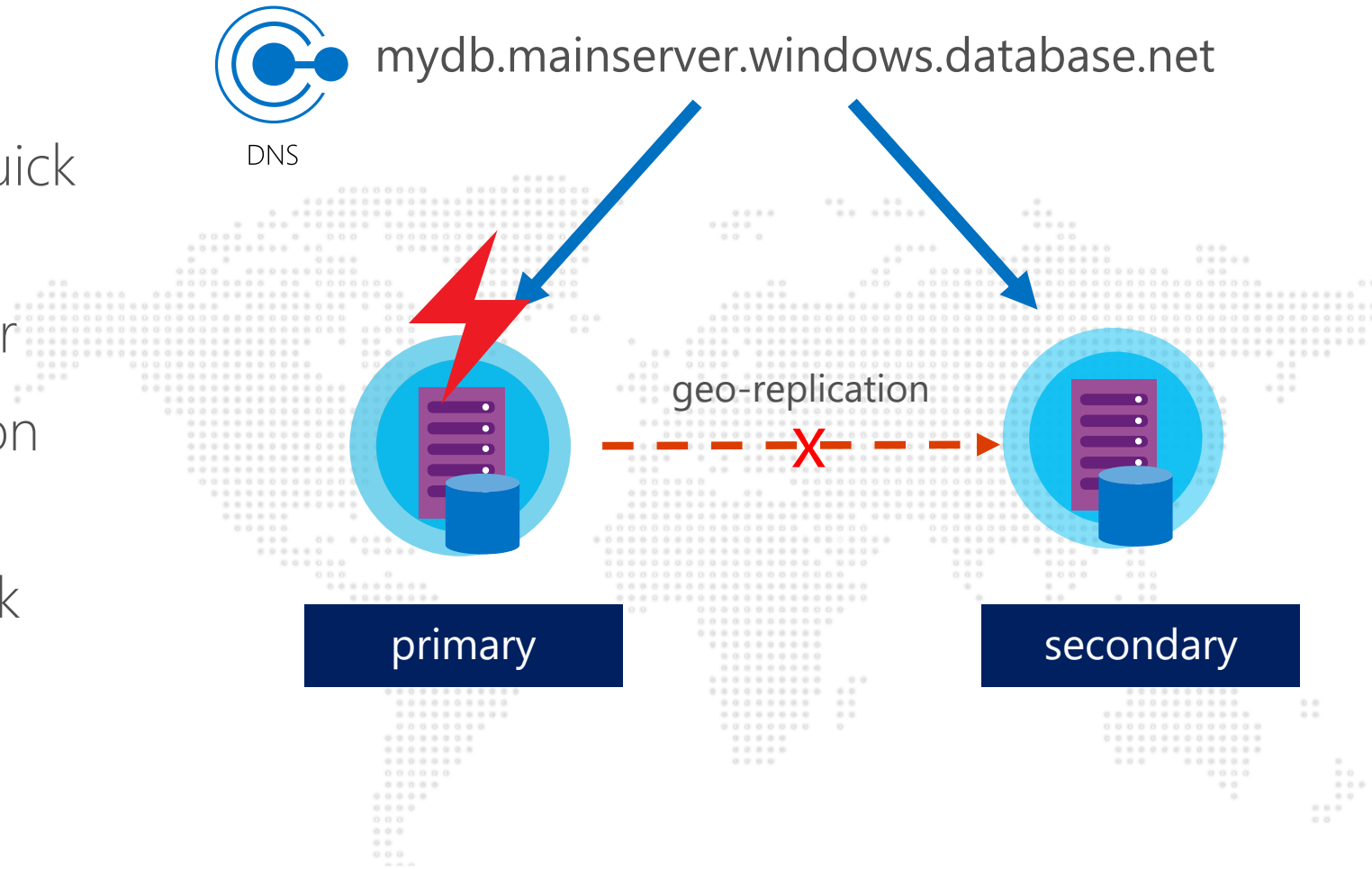
Outage in primary region

Allows cross region isolation with quick with minimal data loss

Automatic DNS swap during failover

Failover without changing application code or connection strings

Manually initiated failover or failback



**Currently in private preview - Microsoft Confidential – Shared Under NDA Only*

Point-in-time restore

Programmatic “oops recovery” of data deletion or alteration

Auto backups, transactional logs every 5 min

Backups in Azure Storage and geo-replicated

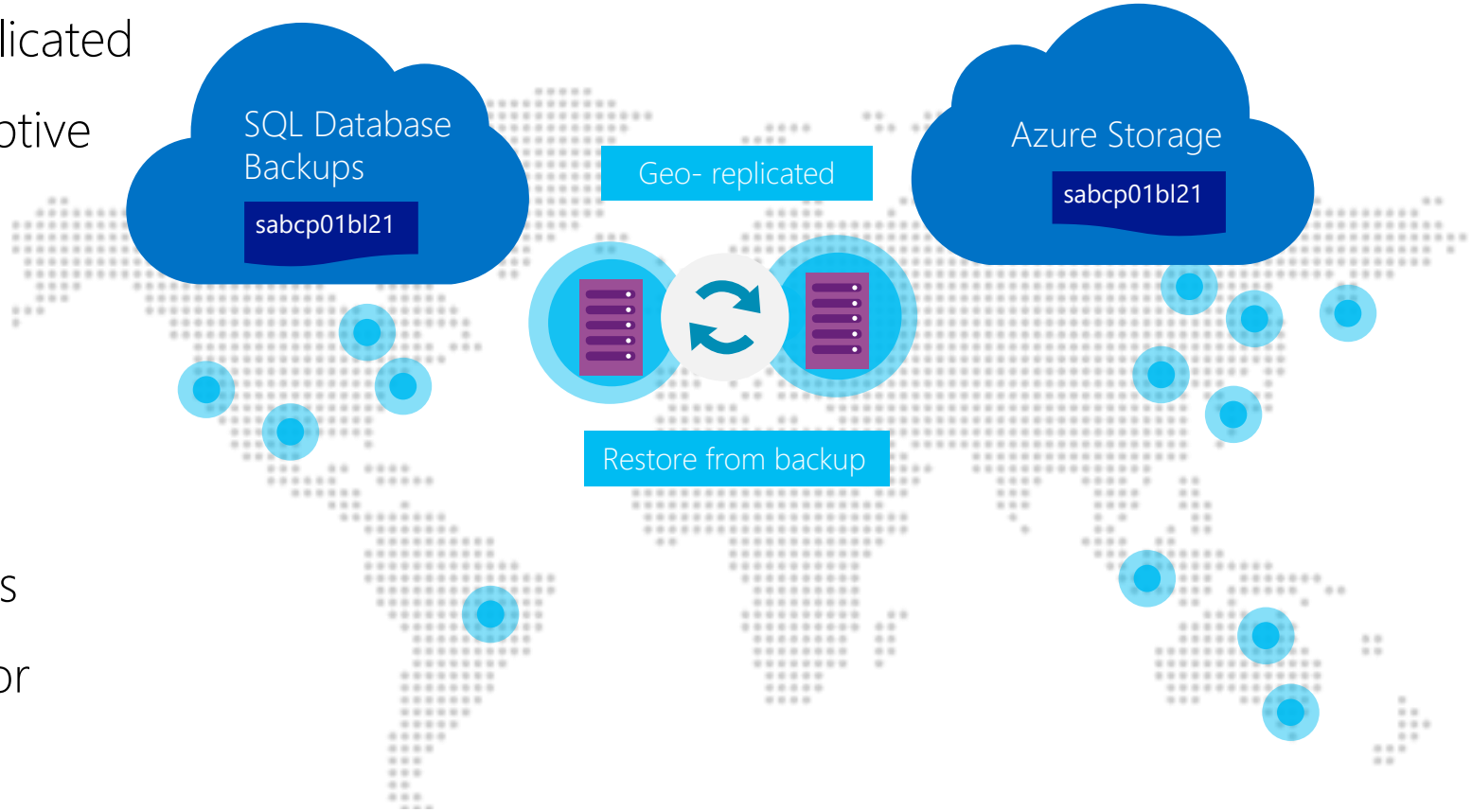
Creates a side-by-side copy, non-disruptive

REST API, PowerShell or Azure Portal

Backups retention policy:

- Basic, up to 7 days
- Standard, up to 35 days
- Premium, Premium RS up to 35 days

Automated export of logical backups for long-term backup protection

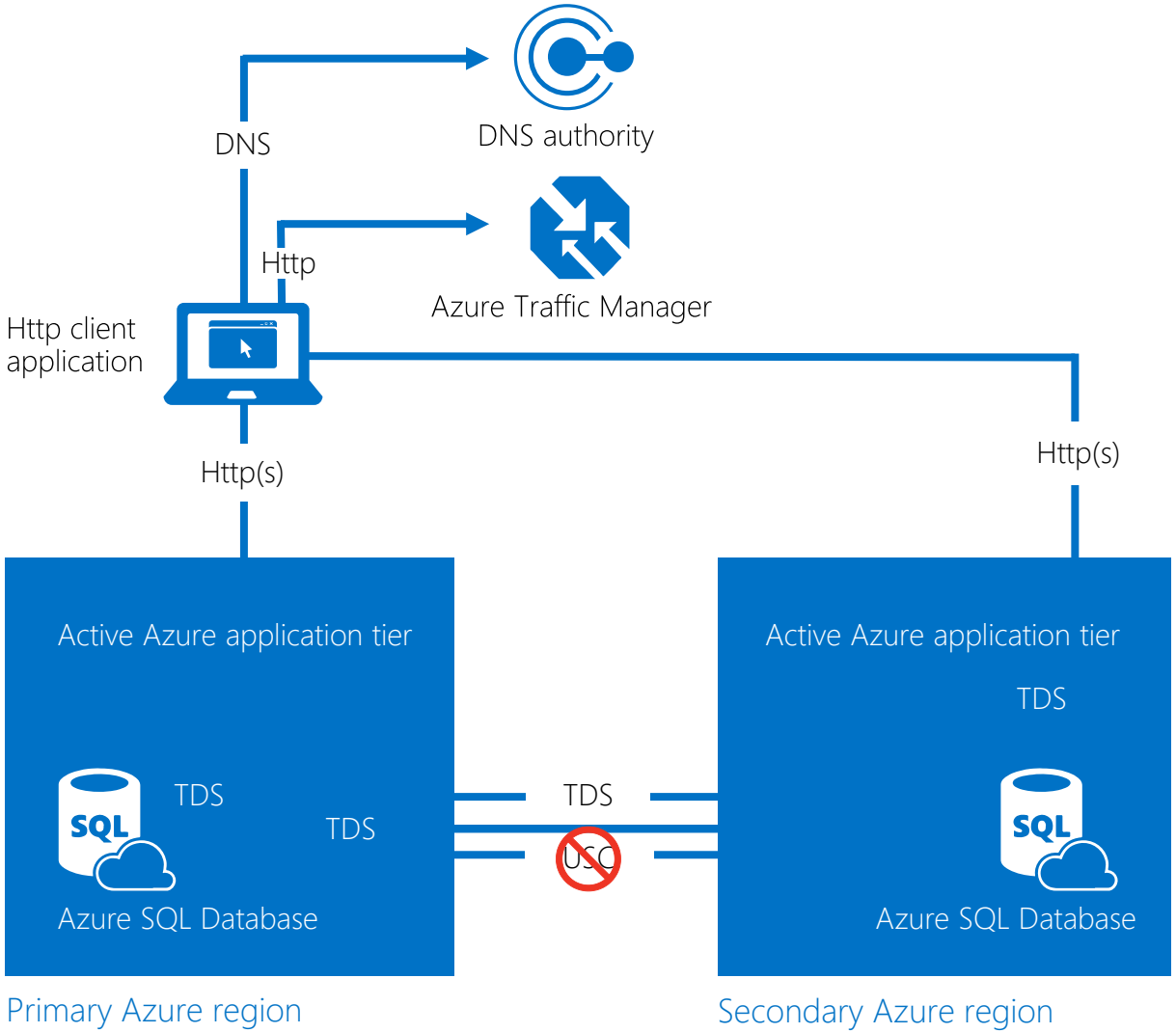


Active-active compute with decoupled failover

Optimized, isolated load balancing of end-user traffic

Secondary is used for read-only access

Failover does not involve DNS change

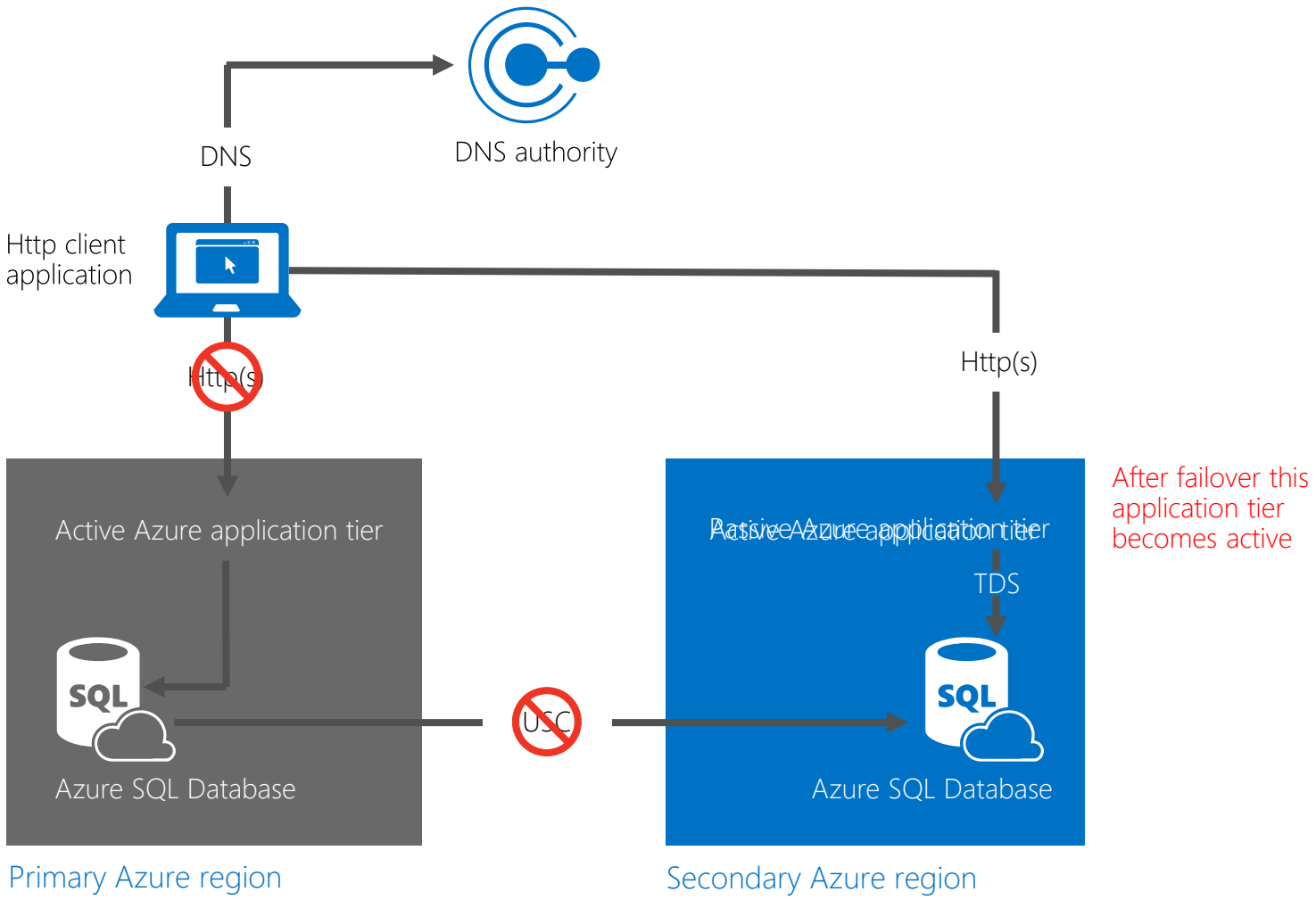


Active-passive compute with coupled failover

Optimized for co-located app and data tier

Secondary is not used for app access

Failover involves DNS change

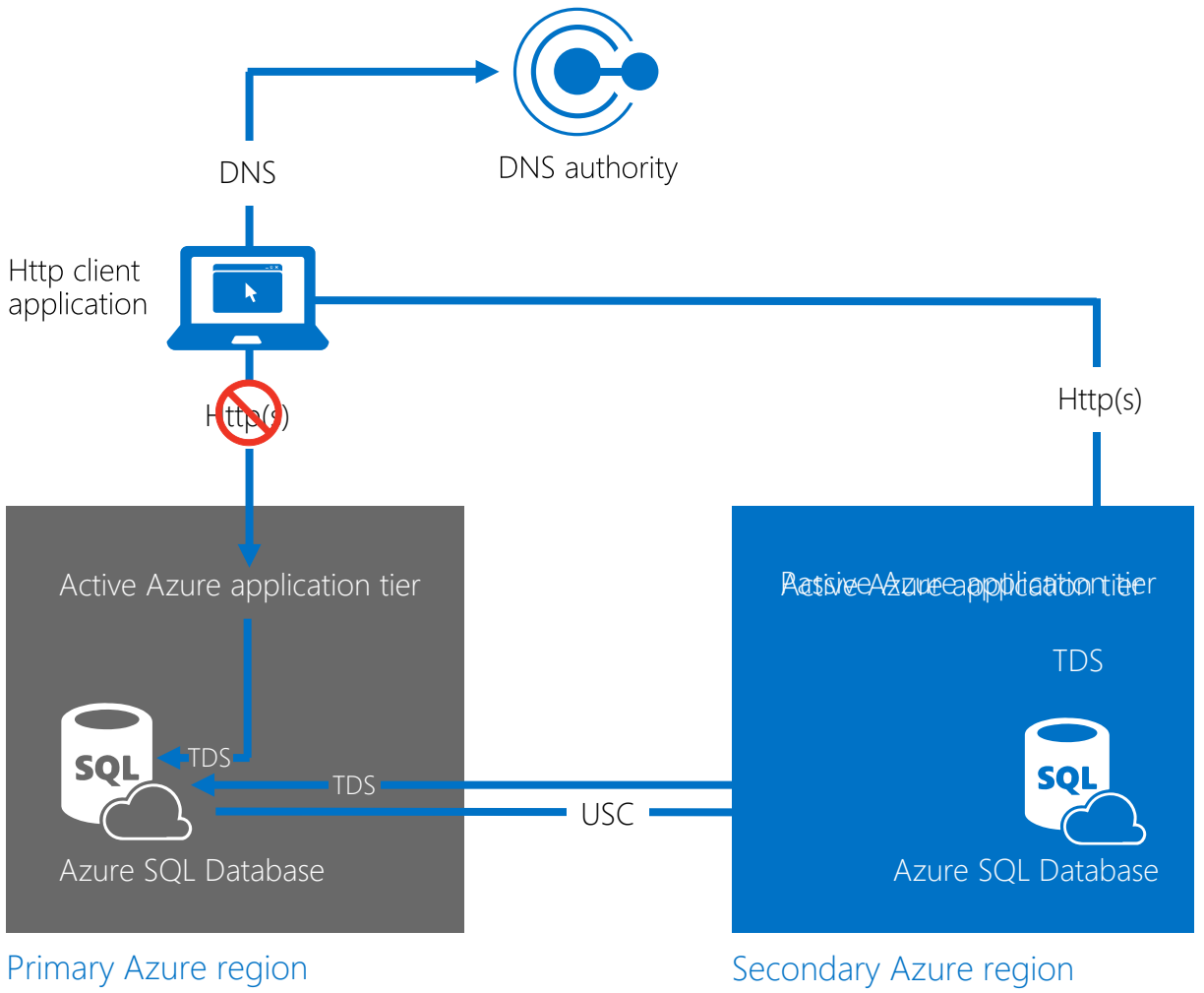


Active-passive compute with decoupled failover

Optimized for co-located app and data tier

Secondary is not used for app access

Failover involves DNS change



After failover this application tier becomes active

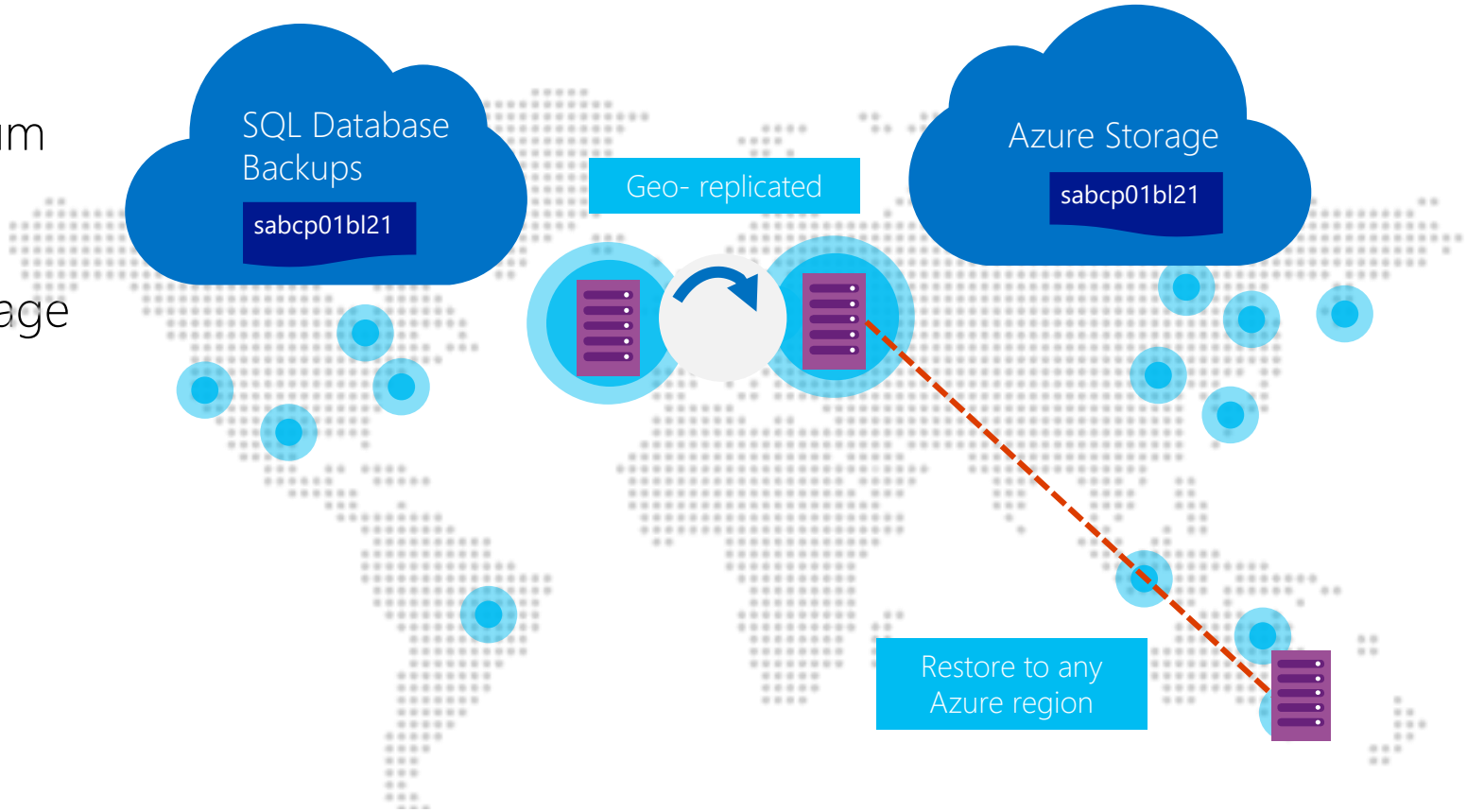
Geo-restore

Emergency data recovery when you need it most

Available in Basic, Standard, Premium and Premium RS

Built on geo-redundant Azure Storage

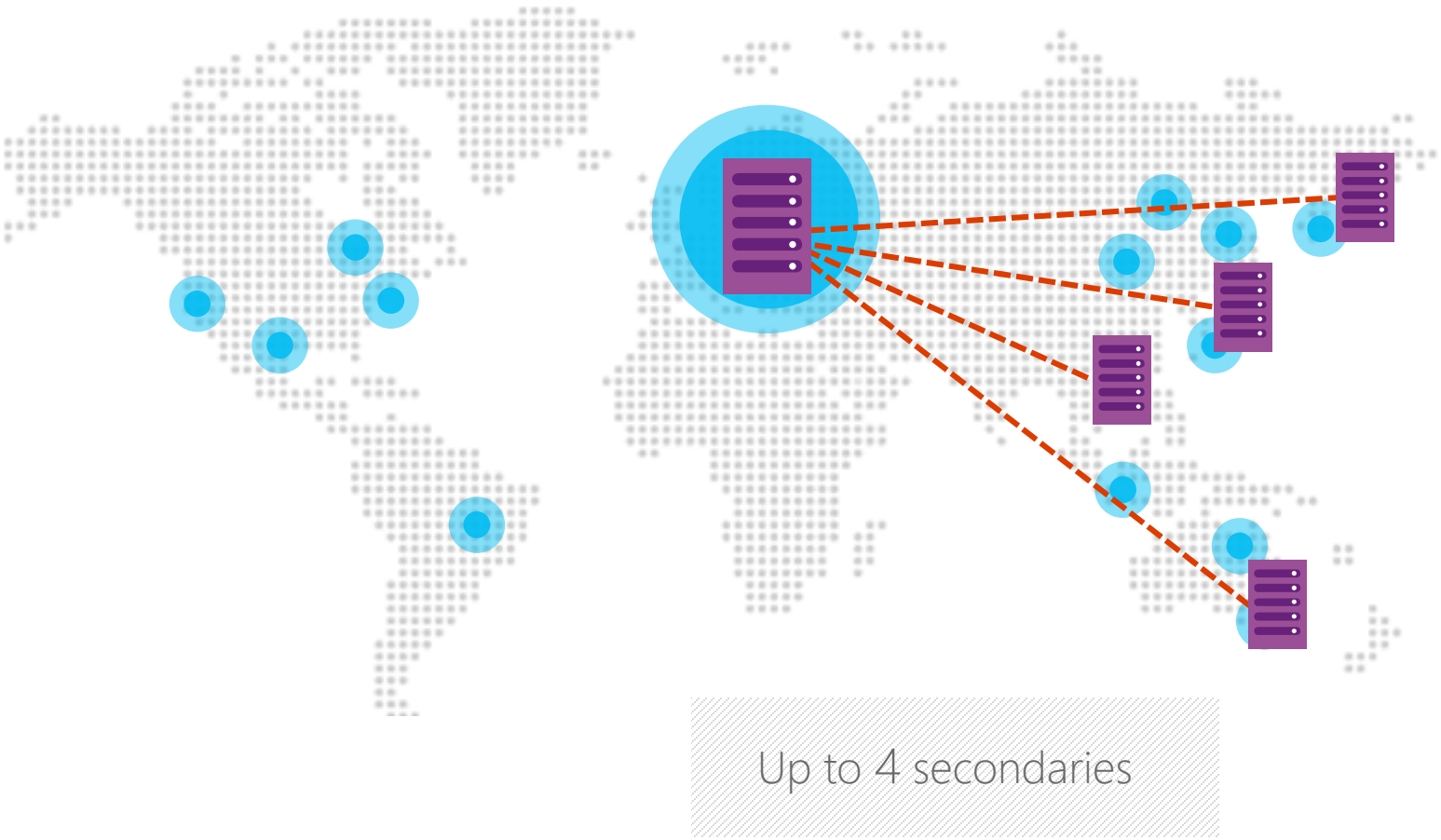
Recover to any Azure region



Active geo-replication

Mission-critical business continuity on your terms, via programmatic APIs

Service levels	Basic, Standard, Premium and Premium RS self service
Readable secondaries	Up to 4
Regions available	Any Azure region
Replication	Automatic, asynchronous
Manageability tools	REST API, PowerShell or Azure Portal
Recovery time objective (RTO)	<1 hour
Recovery point objective	<5 minutes
Failover	On demand



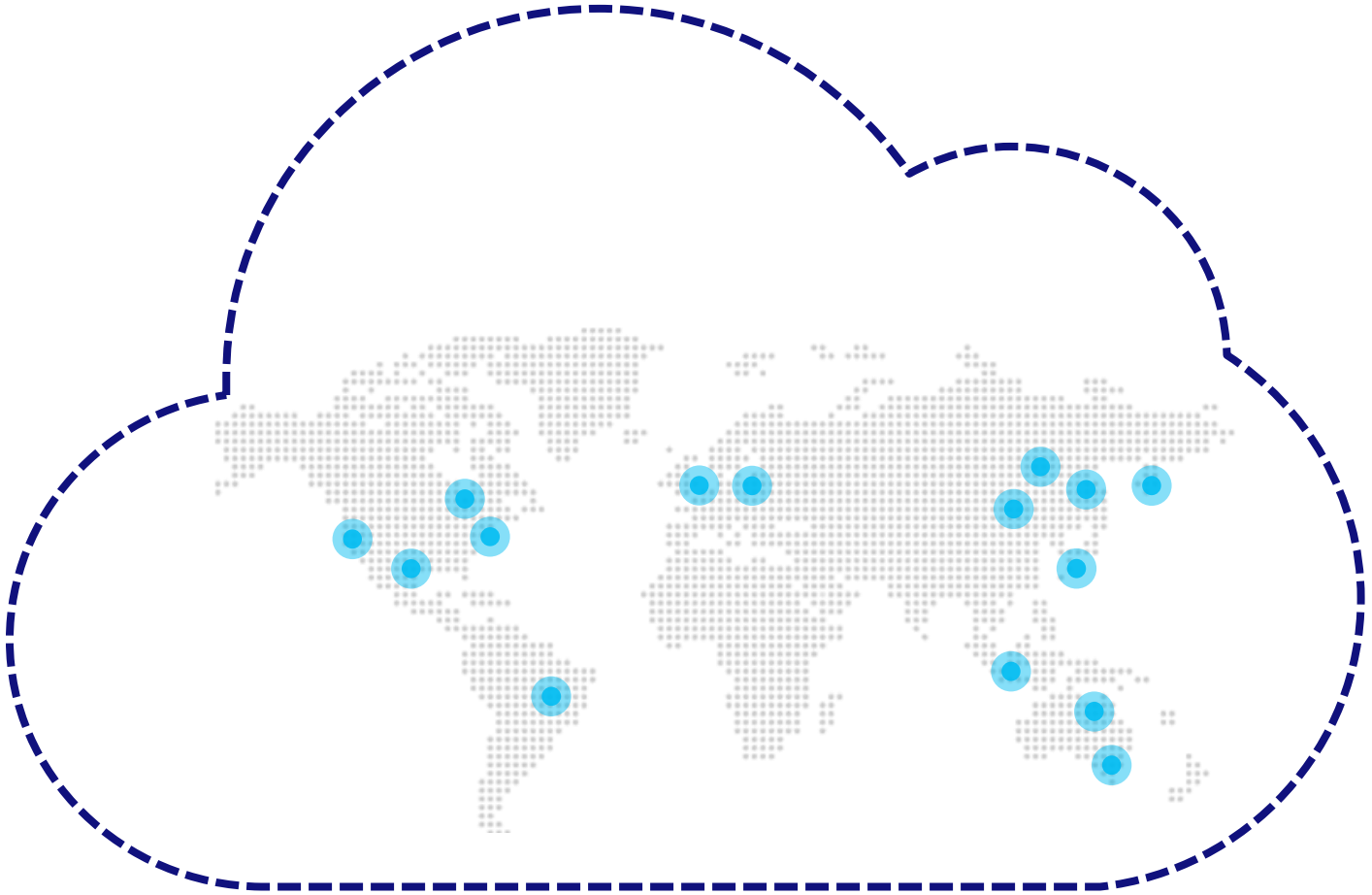
Microsoft-backed

Peace of mind over your cloud investments

Built-in regional database replicas for additional protection

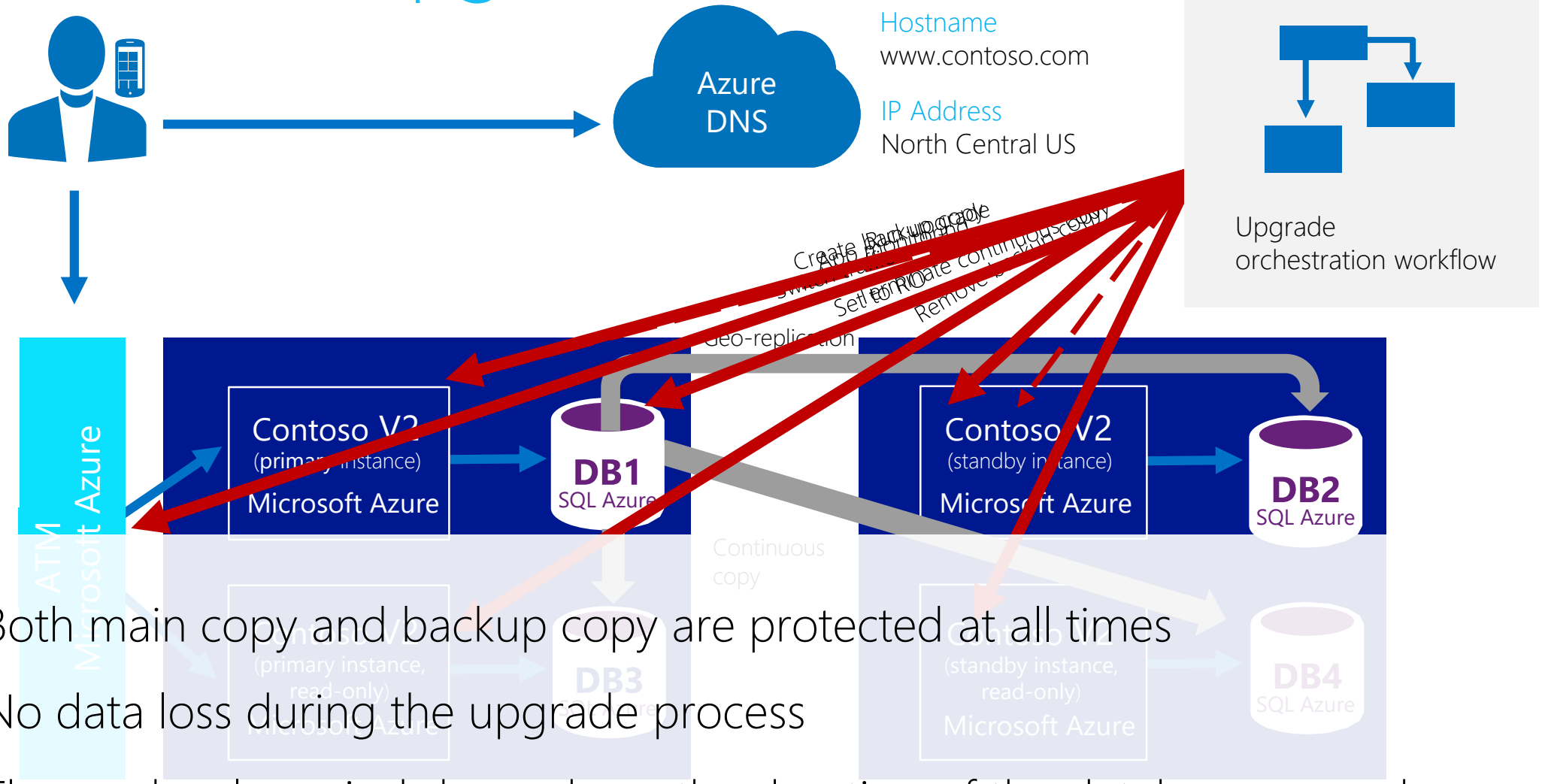
Uptime SLA of **99.99%***

Single support vendor across Azure cloud services



*Web & Business tiers remain backed by 99.9% uptime SLA.
Premium RS tier has a lower SLA.

Online service upgrade



Both main copy and backup copy are protected at all times

No data loss during the upgrade process

The read-only period depends on the duration of the database upgrade

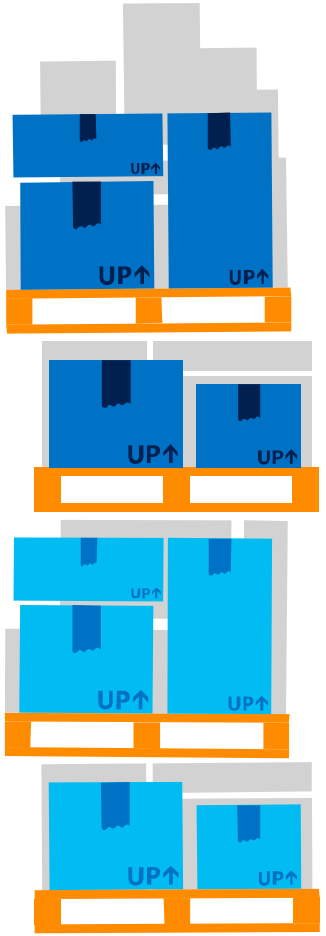
Long-term data retention

Export a database

Generate a BACPAC in external storage and hydrate as needed

Live database

Utilize lower performance tier SQL database to support quick data retrieval



Export a database

Flexible and portable option but incurs operational overhead

Pros	Cons
Portable data format – logical schema and data	Need workaround (DB-Copy) to ensure consistent database
Low cost	Slow to restore

Export a database

Ideal for customer needing portable dataset

Create a database copy

Ensure transactional consistent copy

Export backup to storage account

Export to customer storage account

Repeat as needed

Create additional archive copy as needed

Azure SQL Database wrap-up

Microsoft Azure

Learn more

Pricing: https://azure.microsoft.com/en-us/pricing/details/sql-database/?b=16.50

Services: https://azure.microsoft.com/en-us/services/sql-database/

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

Stack overflow: http://stackoverflow.com/questions/tagged/sql-azure

Microsoft