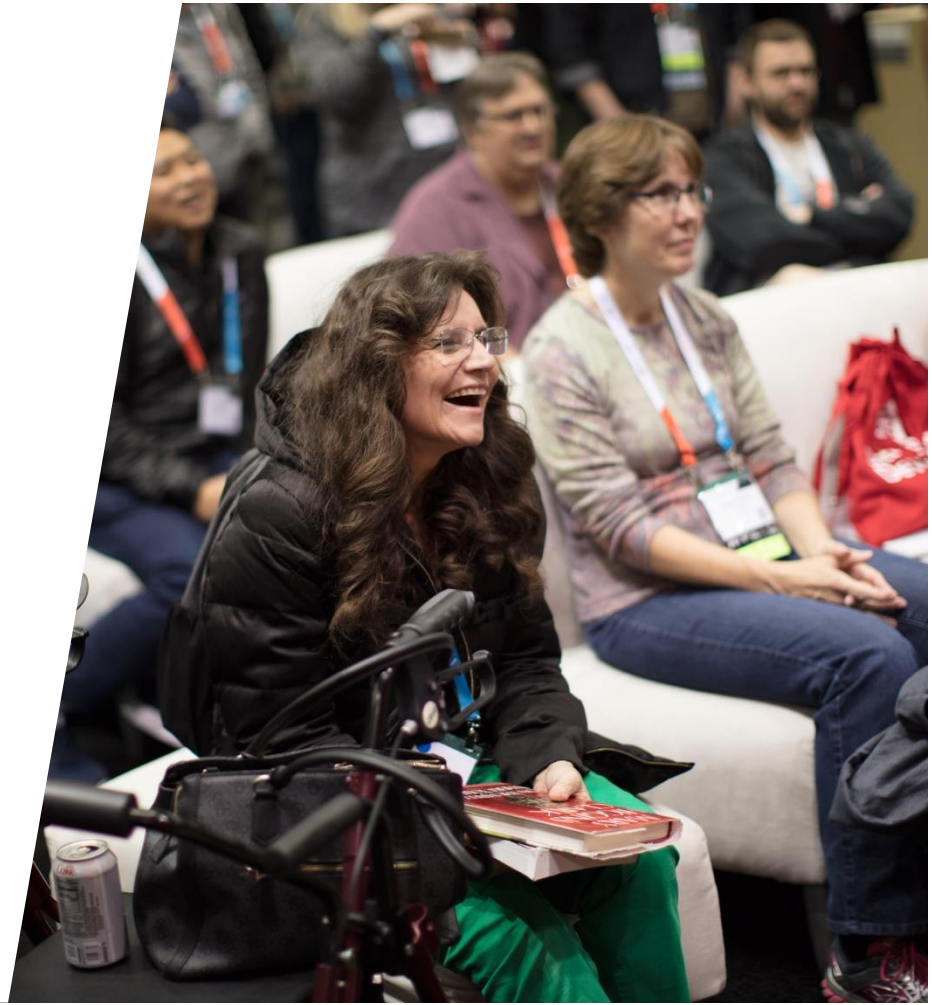
PASS SUMMIT2019

Modernizing your Database with Azure SQL

Azure SQL Team, Microsoft

Please silence cell phones



Explore
everything PASS has to offer

Free Online Resources

Newsletters

PASS.org

Free online webinar events

Local user groups around the world

Free 1-day local training events

Online special interest user groups

Business analytics training

Get involved

Azure SQL Team

Microsoft

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

@azuresql #AzureSQL

Sanjay Mishra
Denzil Ribeiro
Shreya Verma
Mladen Andzic
Andreas Wolter
Davide Mauri
Rohit Nayak
Dimitri Furman
Jovan Popovic
Joe Sack
Rie Irish

Getting Started

Agenda

Time	Topic	Speaker
08:40 - 08:50	Azure SQL: What, Why and How	Sanjay Mishra
08:50 - 09:30	Azure SQL VM: The Simplest way to run SQL in Azure	Shreya Verma
09:30 - 10:00	Azure SQL PaaS services: maximize your potential. What PaaS provides and how?	Sanjay Mishra
10:00 - 10:15	Refreshment Break	
10:15 - 10:45	Azure SQL MI: What and How? Migration to MI.	Mladen Andzic

Agenda

Time	Topic	Speaker
10:45 - 11:15	Networking in Azure SQL DB and MI	Rohit Nayak
11:15 - 12:00	Security in Azure SQL DB and MI	Andreas Wolter
12:00 - 1:00	Lunch Break	
01:00 - 01:30	Azure SQL DB: Best cloud database for developers	Davide Mauri
01:30 - 02:00	Azure SQL DB: Best cloud database for Enterprise applications	Dimitri Furman, Davide Mauri

Agenda

Time	Topic	Speaker
02:00 - 02:30	Azure SQL DB Hyperscale	Denzil Ribeiro
02:30 - 02:45	Refreshment Break	
02:45 - 03:25	Azure SQL DB: Best cloud database for DBAs	Dimitri Furman
03:25 - 03:55	Troubleshooting Common Performance and Scale problems	Denzil Ribeiro
03:55 - 04:25	Intelligent DB	Joe Sack

Session Evaluations

Submit by 5pm Friday, November 15th to win prizes.

3 WAYS TO ACCESS

Go to PASSsummit.com

Download the GuideBook App and search: PASS Summit 2019

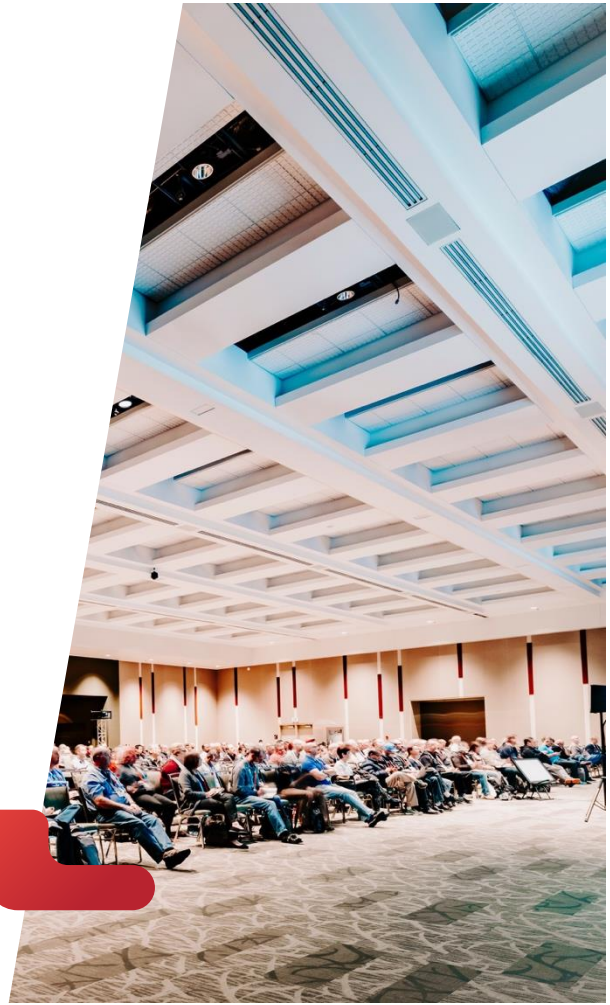
Follow the QR code link on session signage

PASS SUMMIT2019

Azure SQL

What, Why and How

Sanjay Mishra
Student of Life, for Life



Making the most of it

- Learning Continues …
- Sign up sheet
- Take a page out of your notebook

Azure SQL

SQL virtual machines

Best for migrations and applications requiring OS-level access

SQL virtual machine

- SQL Server and OS server access
- Expansive SQL And OS version support
- Automated manageability features for SQL Server

Managed instances

Best for most lift-and-shift migrations to the cloud

Single instance

- SQL Server surface area (vast majority)
- Native virtual network support
- Fully managed service

Instance pool

- Resource sharing between multiple instances to price optimize
- Enables migration of many small instances at scale
- Fully managed service

Databases

Best for modern cloud applications. Hyperscale and serverless options are available

Single database

- Hyperscale storage (up to 100TB)
- Serverless compute
- Fully managed service

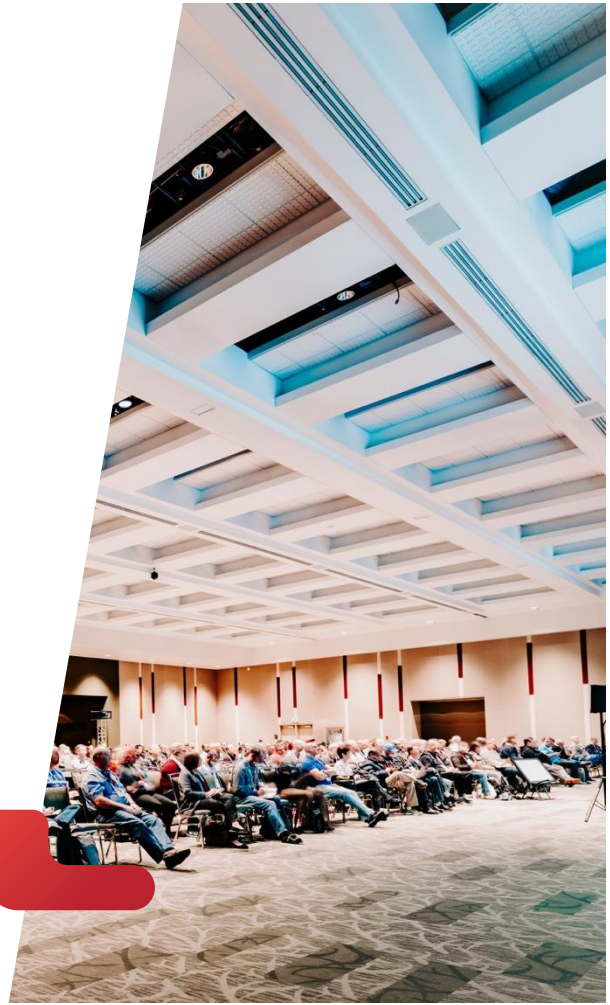
Elastic pool

- Resource sharing between multiple databases to price optimize
- Simplified performance management for multiple databases
- Fully managed service

PASS SUMMIT2019

SQL Server in Azure Virtual Machine

Shreya Verma
Program Manager, Microsoft



SQL Server on Azure Virtual Machines

Secure and compliant

Most compliant cloud

ISO CSA cloud security alliance HIPAA

Seamless cloud migration

100% SQL Server compatibility and only cloud with pre-configured Developer edition

Leading TCO

Save up to 43% vs. AWS EC2 with Azure Hybrid Benefit

Flexibility and control with automation

Easier to maintain than EC2 with automated security patches and automated backup

Note: Savings vs. AWS based on 8-Core D13 v2 Azure VM in East US 2 region at CentOS/Ubuntu Linux rate (combining Azure Hybrid Benefit for SQL Server with Azure Hybrid Benefit for Windows Server); comparable AWS VM is r3.2xlarge in US East (N. Virginia) at Windows rate (with SQL Server License Mobility). Savings analysis excludes Software Assurance costs for SQL Server and Windows Server, which may vary based on EA or any applicable volume licensing agreement. Actual savings may vary based on region, instance size, and compute type. Publicly-available prices as of June 5, 2018, subject to change.

PASS

SQL VM Resource Provider

SQL Server on Azure VM

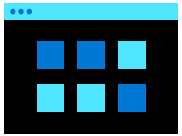
What is Resource Provider?

Resource Provider brings the functionality of Azure Marketplace images to SQL Server instances self-installed on Azure VMs.

Comprehensive feature set

Self-installed VMs registered with RP now can access automation features in Azure Marketplace images

Leverage auto-backup, auto-patching, and automate Always On Availability Groups to avoid time-consuming admin and VM customization



Dashboard view for VM awareness

Azure VMs are now discoverable on the new Azure SQL blade in Azure Marketplace

Easily manage your SQL VM and SQL PaaS deployments from one central location

Simple license conversions

Self-installed VMs with RP can be easily converted to PAYG images

Save money by converting variable workloads with Software Assurance to PAYG images

Learn more aka.ms/sqlvm_rp and aka.ms/sqlvm_rp_documentation

How to register with SQL VM RP?

- Register the resource provider to your subscription (one-time only) via the Azure Portal or Azure CLI

```
# Register the new SQL resource provider to your subscription
az provider register --namespace Microsoft.SqlVirtualMachine
```

- Register single VM with SQL VM RP

```
# Register your existing SQL Server VM with the new resource provider
az sql vm create -n <VMName> -g <ResourceGroupName> -l <VMLocation> --license-type <PAYG/AHUB>
```

- NEW Bulk register with SQL VM RP

```
# Register all existing SQL Server VM with the new resource provider
Import-Module .\RegisterSqlVMs.psm1 Register-SqlVMs -SubscriptionList SubscriptionId1, SubscriptionId2
```

*If you're unable to select Full or Lightweight RP, choose an Agentless approach. Note this option does not enable verification of Resource Provider.

PASS

DEMO

Azure SQL VM – Resource Provider

SQL Server Performance on Azure VM

SQL VM Performance with TPC-E

For read heavy workloads, Azure Blob Cache is the key for the best price/perf

SQLVM on Azure: E64(Cores=64, RAM =432GB) - Workload TPCE (800K-Full scale-800GB)																	
No Of users	TPS	Client %CPU	CPU% Total	CPU% Priv	Batch Req	Log Flushes Sec	Log MB Flushed/sec	IOPS/sec	IO MB\Sec	READ IOPS	RIO MB\Sec	WRITE Total IOPS	WIOPS MB\Sec	WRITE/Data IOPS	WRITE/Log IOPS	RQ-Depth	WQ-Depth
256	490	23	19	3	9,156	394	5	18,403	164	8,735	72	9,668	92	9,285	381	247	1,703

Improved application throughput by adding more storage capacity

SQLVM on Azure: E64(Cores=64, RAM =432GB) - Workload TPCE (80K-Full scale-700GB)																	
No Of users	TPS	CPU% Total	CPU% Priv	Batch Req	Log Flushes Sec	Log MB Flushed/sec	IOPS/sec	IO MB\Sec	READ IOPS	RIO MB\Sec	WRITE T	WIOPS MB\Sec	WRITE/Data IOPS	WRITE/Log IOPS	RQ-Depth	WQ-Depth	
500	823	45	13	14,672	190	8	36,787	323	17,809	146	18,978	177	18,805	196	347	1640	
500	801	47	14	14,525	166	7	37,227	324	17,598	144	19,630	179	19,482	145	363	1473	

Maximize throughput with lower storage latencies

SQLVM on Azure: E64(Cores=64, RAM =432GB) - Workload TPCE (80K-Full scale-700GB) Ultra Disk																	
No Of users	TPS	CPU% Total	CPU% Priv	Batch Req	Log Flushes Sec	Log MB Flushed/sec	IOPS/sec	IO MB\Sec	READ IOPS	RIO MB\Sec	WRITE T	WIOPS MB\Sec	WRITE/Data IOPS	WRITE/Log IOPS	RQ-Depth	WQ-Depth	
500	1,489	92	20	26,614	417	14	60,682	547	31,431	258	29,250	288	28,730	377	35	106	
500	1,185	94	20	26,216	381	14	59,498	534	30,879	253	28,619	281	28,130	343	26	46	

Full study https://techcommunity.microsoft.com/t5/SQL-Server/Optimize-OLTP-Performance-with-SQL-Server-on-Azure-VM/ba-p/916794

Performance Guidance for SQL Server on Azure VM

It is still just SQL Server….

Locked Pages in Memory (Windows)	Instant File Initialization (Windows)	Standard SQL Query Performance Best Practices and Tuning	SQL Linux Performance Best Practices	RHEL8 has XFS optimizations	Don't forget about File-Snapshot Backups in Azure

IOPS Testing

1- Start with collecting peak IOPS requirement for Data, Log and Temp DB.
2- Choose the VM size that can scale to the total IOPS requirement
3- If workload is Temp DB heavy, locate Temp DB on Local SSD
4- Use RO cache enabled Premium SSD disks <4TB for Data Files
5- Use disk striping to achieve Data IOPS requirement
6- Provision Ultra Disk for Log file
7- Monitor SQL DB performance, write queues on Data Files becoming the bottleneck for throughput then move Data files to Ultra SSD.
8- https://techcommunity.microsoft.com/t5/SQL-Server/Optimize-OLTP-Performance-with-SQL-Server-on-Azure-VM/ba-p/916794

SQL Server IaaS on Windows

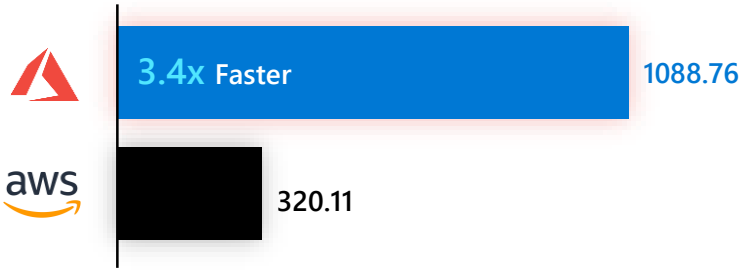
SQL Server 2017 Enterprise (Windows) on Azure and AWS

GIGAOM

- SQL Server on Azure includes Azure Hybrid Benefit
- SQL Server on AWS includes license mobility

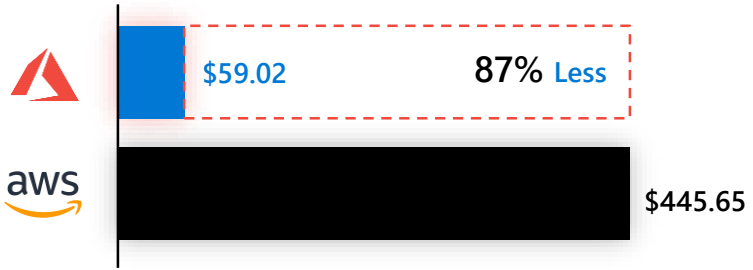
Performance

Transactions per second (TPS)



Price/performance*

3-year pricing / TPS



*Includes Azure Hybrid Benefit and Reserved Instances with three-year commitment.

**Price-performance claims based on data from a study commissioned by Microsoft and conducted by GigaOm in October 2019. The study compared price performance between SQL Server 2017 Enterprise edition on Windows Server 2016 Datacenter edition in Azure E64s_v3 instance type with 4x P30 1TB Storage Pool data (Read Only Cache) + 1x P20 0.5TB log (No Cache) and the SQL Server 2017 Enterprise edition on Windows Server 2016 Datacenter edition in AWS EC2 r4.16xlarge instance type with 1x 4TB gp2 data + 1x 1TB gp2 log. Benchmark data is taken from a GigaOm Analytic Field Test derived from a recognized industry standard, TPC Benchmark™ E (TPC-E). The Field Test does not implement the full TPC-E benchmark and as such is not comparable to any published TPC-E benchmarks. The Field Test is based on a mixture of read-only and update intensive transactions that simulate activities found in complex OLTP application environments. Price-performance is calculated by GigaOm as the cost of running the cloud platform continuously for three years divided by transactions per second throughput. Prices are based on publicly available US pricing in West US for SQL Server on Azure Virtual Machines and Northern California for AWS EC2 as of October 2019. Pricing incorporates three-year reservations for Azure and AWS compute pricing, and Azure Hybrid Benefit for SQL Server and Azure Hybrid Benefit for Windows Server and License Mobility for SQL Server in AWS, excluding Software Assurance costs. Price-performance results are based upon the configurations detailed in the GigaOm Analytic Field Test. Actual results and prices may vary based on configuration and region.

SQL Server
High Availability on Azure VM

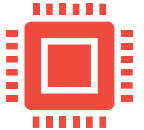
SQL Server HADR technologies for Azure IaaS

Always On Availability Groups

Failover Cluster Instances

Log Shipping

SQL Server Backup and Restore with Azure Blob Storage Service

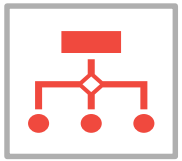


Database Mirroring - Deprecated in SQL Server 2016

NEW SQL Server FCI with Azure Premium File Share

Ease of management

File shares are fully managed by Azure.



Lower the work on your virtual machines

Input or output is offloaded to your managed file share

Burstable input or output capacity

Zonal Redundancy

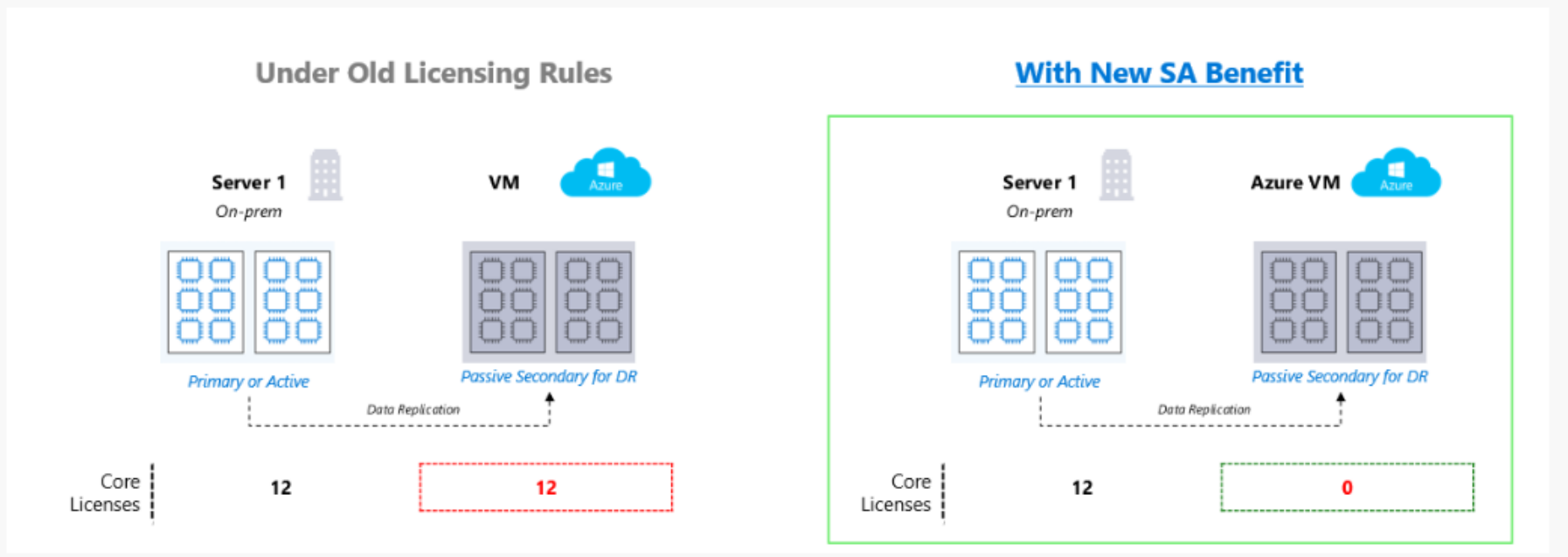
	Premium scale factor
IOPS	1 * provisioned GiB. (max 100,000 IOPS).
Burst IOPS	3 * Baseline IOPS. (max 100,000 IOPS)
Egress	60 MiB/s + 0.06 * provisioned GiB
Ingress	40 MiB/s + 0.04 * provisioned GiB

	Example 10TB Share
IOPS	10,000
Burst IOPS	30,000
Egress	660 MiB/s
Ingress	440 MiB/s

Step-by-step tutorial to configure SQL Server FCI with Premium File Shares

PASS

NEW Free Failover servers for disaster recovery in Azure



Thank You

Shreya Verma

@shreyavermakale

PASS

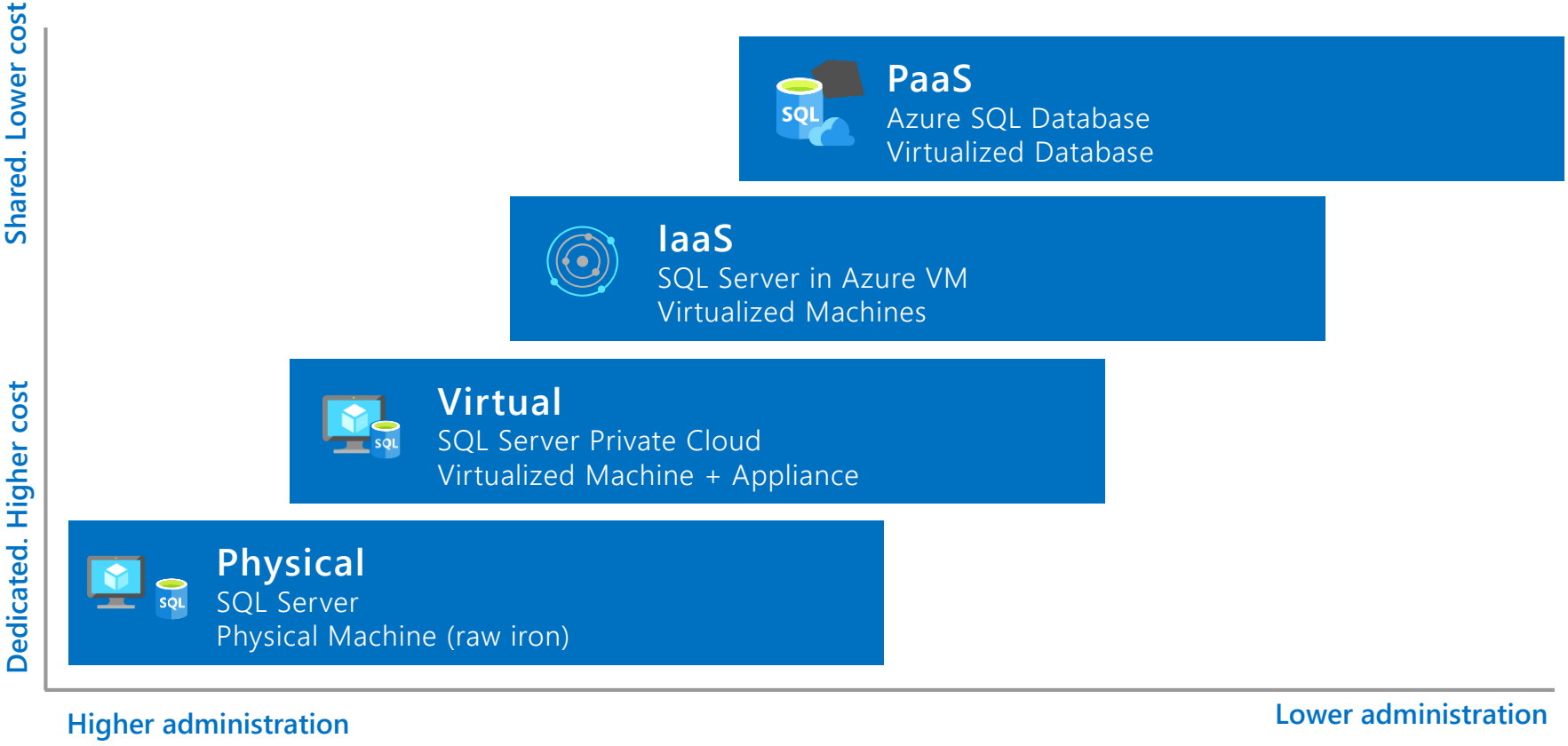
PASS SUMMIT2019

Can you please PASS the PaaS !

Azure SQL PaaS Services

Sanjay Mishra (sanjaymi@microsoft.com)
Student of Life, for Life

Data platform continuum



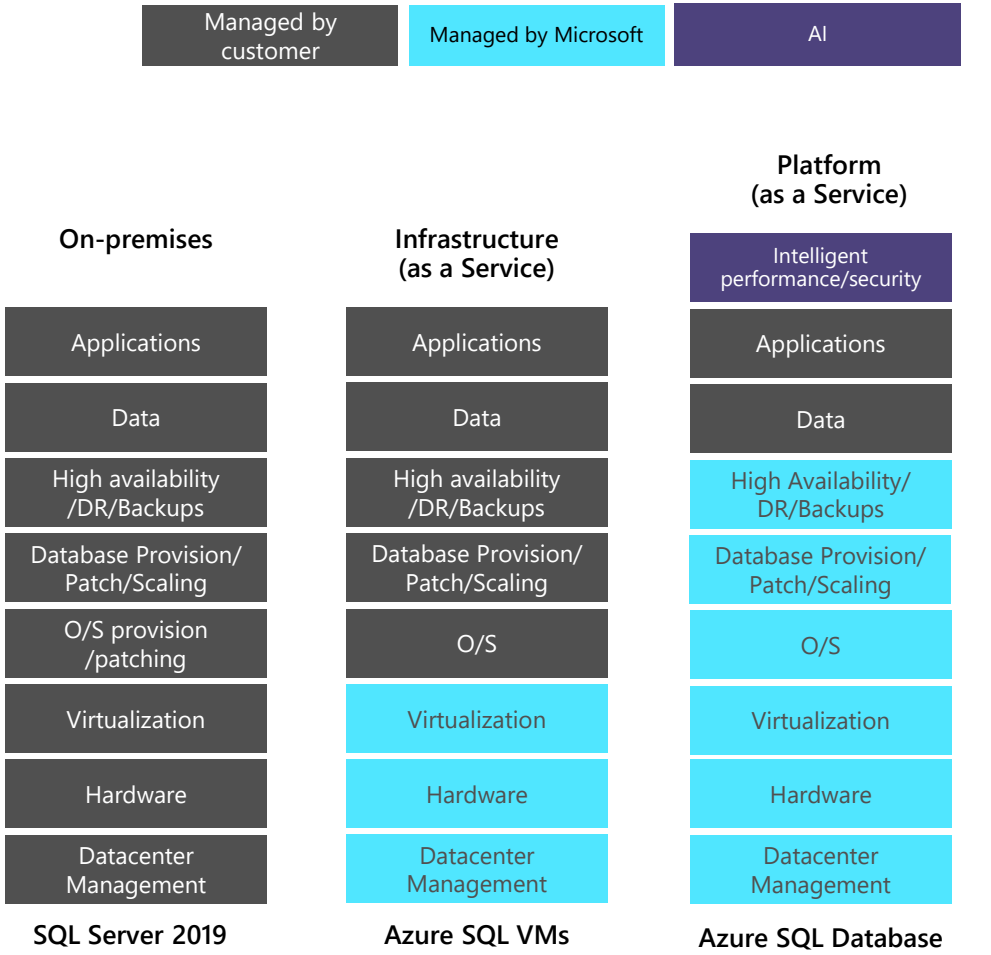
Opportunity for modernization

On-premises costs tend to be driven by hardware and data center management costs

Infrastructure-as-a-Service reduces cost categories related to data center and compute

Platform-as-a-Service off-loads customers' most administrative tasks to Azure, further improving efficiency with machine-learning capabilities for performance and security

- **Managed Instance**: instance-level deployment for lift-shift existing apps to Azure, fully backward compatible
- **Single database**: database-level deployment for new apps



Azure SQL

SQL virtual machines

Best for migrations and applications requiring OS-level access

SQL virtual machine

- SQL Server and OS server access
- Expansive SQL and OS version support
- Automated manageability features for SQL Server

Managed instances

Best for most lift-and-shift migrations to the cloud

Single instance

- SQL Server surface area (vast majority)
- Native virtual network support
- Fully managed service

Instance pool

- Resource sharing between multiple instances to price optimize
- Enables migration of many small instances at scale
- Fully managed service

Databases

Best for modern cloud applications. Hyperscale and serverless options are available

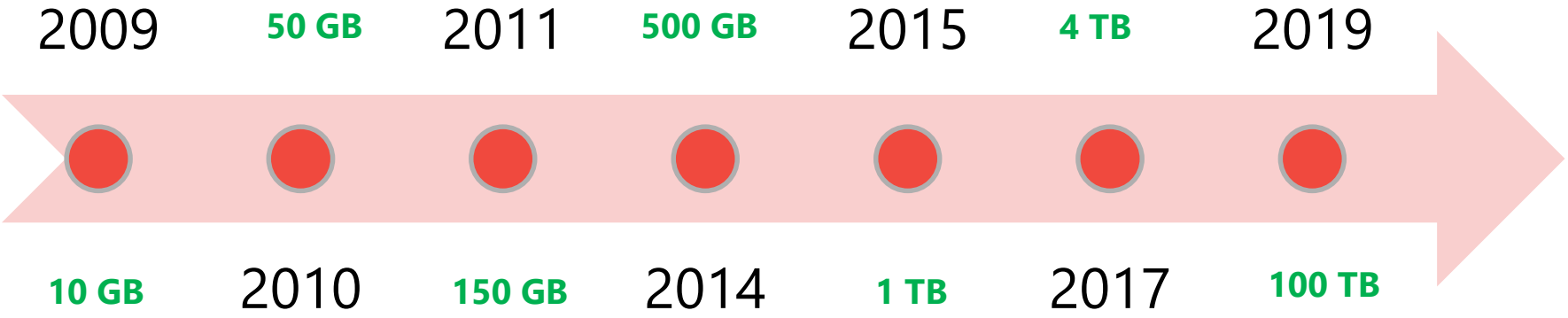
Single database

- Hyperscale storage (up to 100TB)
- Serverless compute
- Fully managed service

Elastic pool

- Resource sharing between multiple databases to price optimize
- Simplified performance management for multiple databases
- Fully managed service

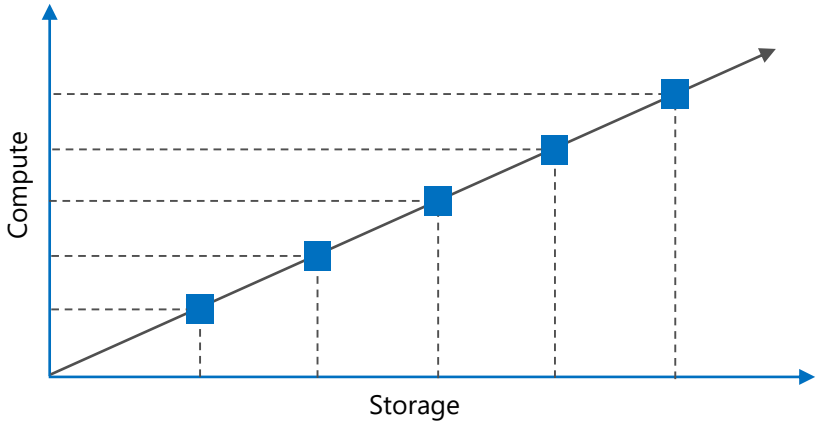
Azure SQL DB Journey



Flexible compute & storage options

DTU model

Simple, preconfigured



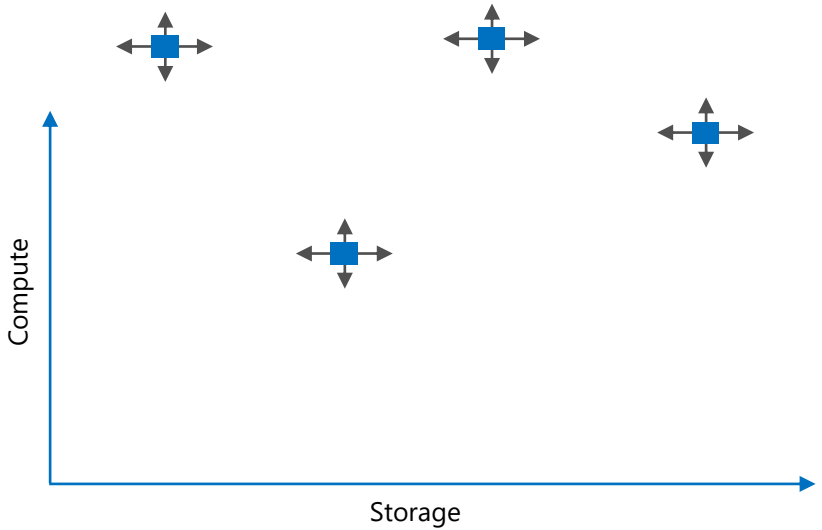
Pre-packaged, bundled unit that represents the database power

Designed for predictable performance, but somewhat inflexible and limited in options

DTU sizing offers simplicity of choice

vCore model

Independent scalability



This model allows you to independently choose compute and storage resources. It also allows you to use Azure Hybrid Benefit for SQL Server to gain cost savings.

Best for customers who value flexibility, control and transparency

Flexible compute, storage & performance options

Service tier	General purpose		Business critical		Hyperscale
Best for	Offers budget oriented balanced compute and storage options		OLTP applications with high transaction rate and low IO latency.		Most business workloads. Auto-scaling storage size up to 100 TB, fluid vertical and horizontal compute scaling, fast database restore
Deployment option	Single / Elastic Pools	Managed Instance	Single / Elastic Pools	Managed Instance	Single
Compute tiers	Gen4: 1 to 24 vCore Gen5: 2 to 80 vCore	Gen4: 4 to 24 vCore Gen5: 2 to 80 vCore	Gen4: 1 to 24 vCore Gen5: 2 to 80 vCore	Gen4: 4 to 24 vCore Gen5: 4 to 80 vCore	Gen4: 1 to 24 vCore Gen5: 2 to 80 vCore
Storage	**Premium remote**		**Local SSD**		**Local SSD Cache**
	32GB – 4TB per instance	32GB – 8TB per instance	32GB – 4TB per instance	32GB – 4TB per instance	Scale from 5GB to 100TB of storage in 10GB increments

For latest information reference: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers-general-purpose-business-critical

Azure SQL Database — Everything built-in

Intelligent performance

Realize automatic performance improvements from continuous assessment and innovation

Scales on the fly

Change service tiers, performance levels, and storage dynamically without downtime

Business continuity

Easily manage and monitor business critical functions for reliable operations

Works in your environment

Develop your app and connect to SQL Database with the tools and platforms you prefer

Advanced threat pprotection

Build security-enhanced apps with built-in protection and industry-leading compliance

The Total Economic Impact of Azure SQL Database Managed Instance

Microsoft commissioned Forrester Consulting to conduct a Total Economic Impact™ study to examine potential cost savings and business benefits enterprises would achieve from migrating on-premises workloads to Azure SQL Database Managed Instance.

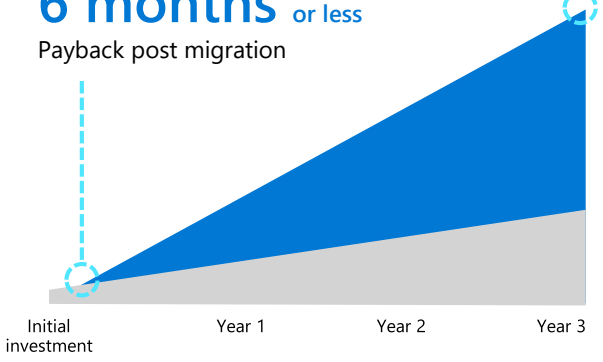
Key report benefits and findings

212% Total ROI

3 years after initial investment & migration to Azure SQL Database Managed Instance

6 months or less

Payback post migration



$1.8M

Savings from avoided hardware, network, storage and ongoing maintenance costs

40%

Improvement in productivity for in-house DBAs

20%

Increase in productivity for IT teams

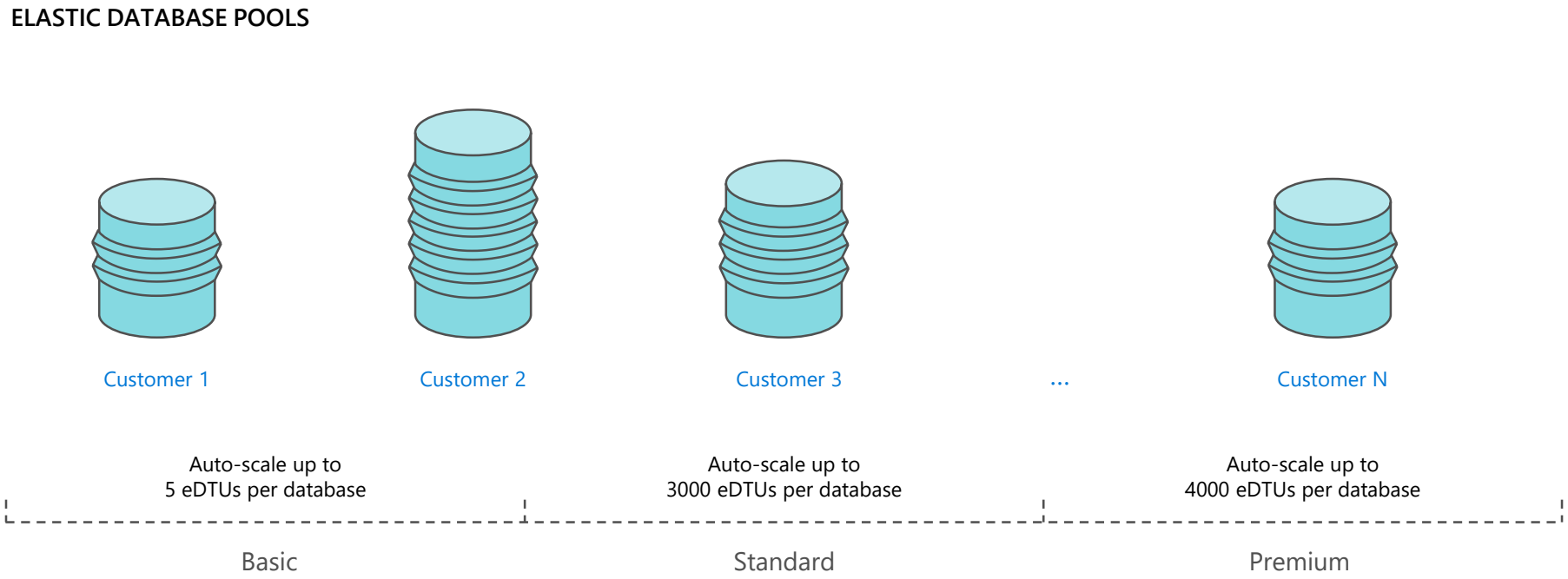
> **Azure SQL Database Managed Instance** is part of our strategic mandate to move all of our application, database, and services footprint to the Azure cloud. **We can quickly integrate and are more nimble and more efficient as a result.**
>
> *Head of development, technology company*

Download the full Total Economic Impact™ of Azure SQL Database Managed Instance report

PASS

Elastic database pool service tiers

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

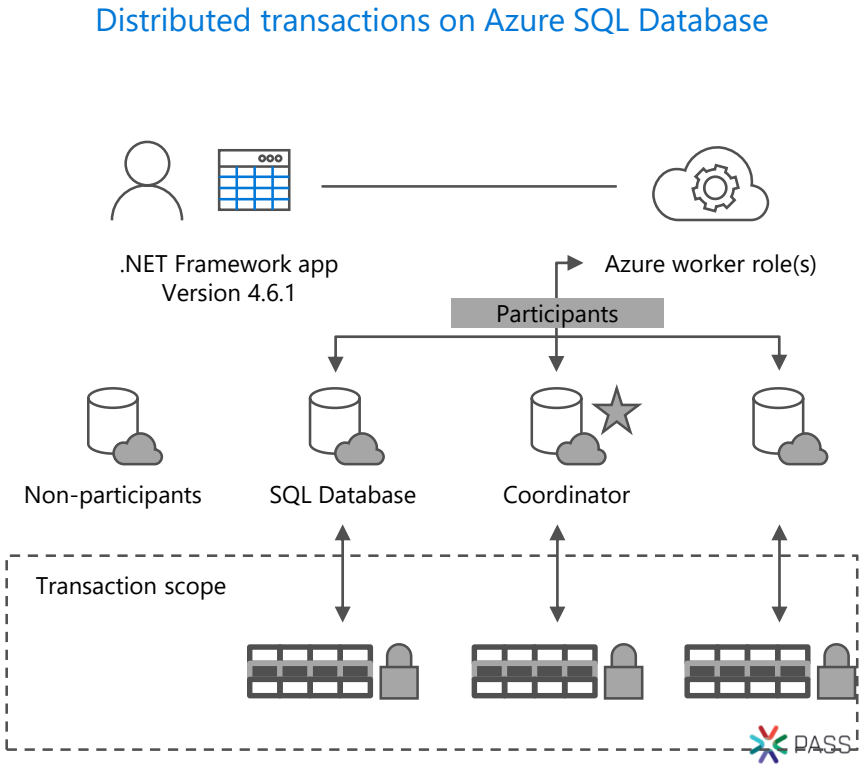


Elastic database jobs overview

- Elastic database jobs enables you to run T-SQL scripts (jobs) against a set of databases
- Execute T-SQL scripts reliably with automatic retry and at scale
- Track job execution state

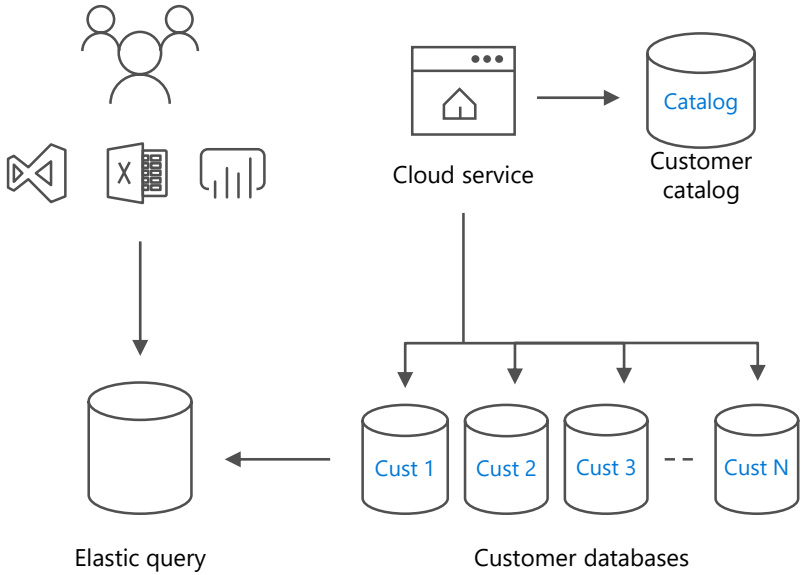
Elastic database transactions

- Distributed transactions: perform operations across several databases with transactional properties in Azure SQL Database
- Ensure that changes are made to all databases or not at all
- Use the same .NET APIs that are used on-premises today



Querying across many databases

- Connect to a single Azure SQL Database instance using familiar Azure SQL Database connection strings
- Simplify setup with Transact-SQL Data Definition Language (DDL)
- Transparently query many databases from a single database
- Familiar programming experience with Transact-SQL, ADO .NET LINQ, Entity Framework (EF), and others
- Use familiar development and business-intelligence (BI) tools to query across many databases
- Designed for interactive querying



Serverless supports common industry scenarios of sporadic or unpredictable usage

Line of business apps

Expense reporting and employee tracking apps

Procurement systems

E-commerce

Opening new marketplaces, marketing campaigns, sales promotions

Content management systems

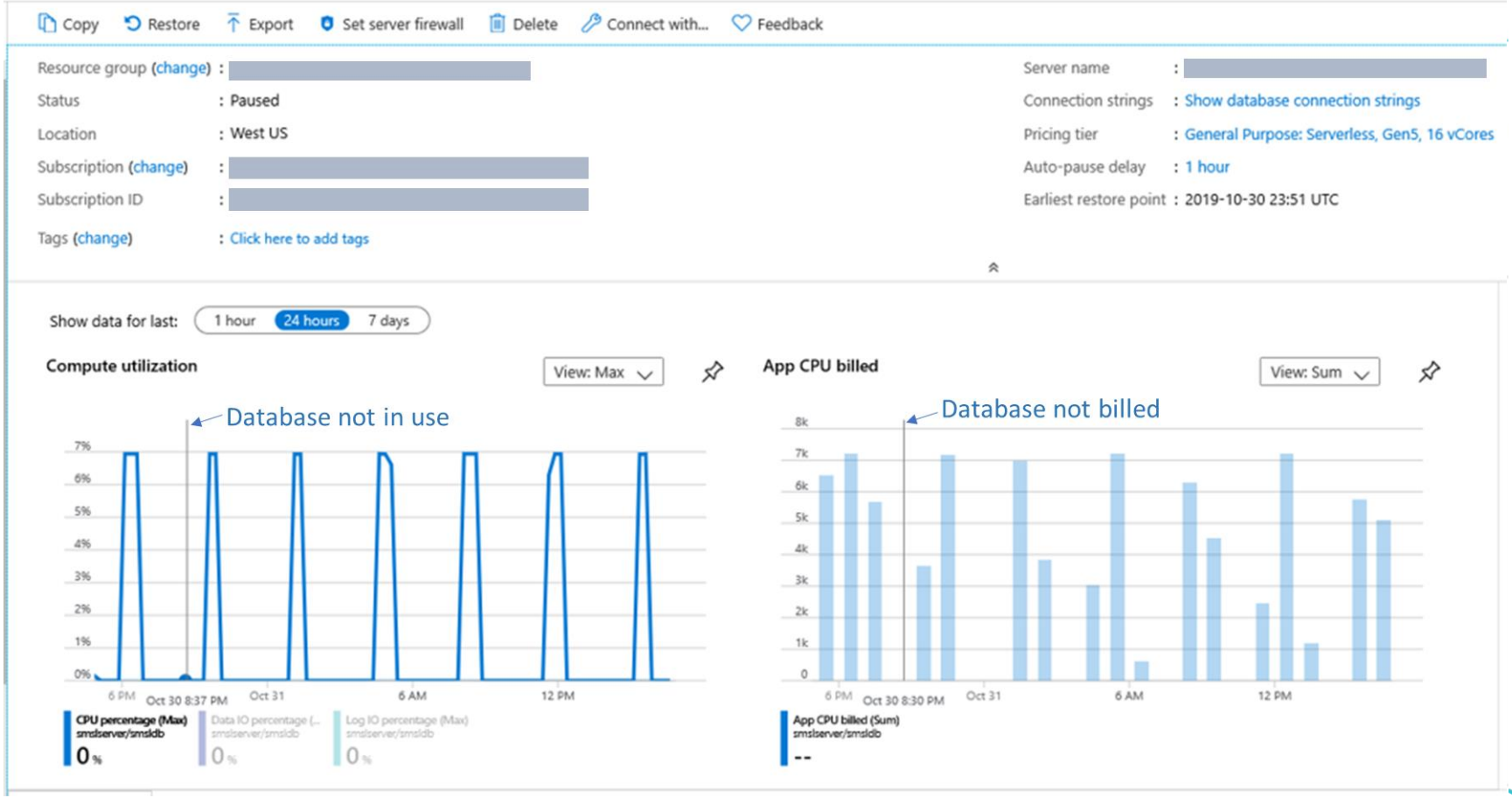
Updating and publishing web content

Content clearinghouses that pull select content by third parties

Dev/test workloads

Handling unpredictable workload needs

Azure SQL Serverless



PASS

Provisioned compute and serverless meet different needs

Optimize compute provisioning and billing for your workload

Serverless databases...

Scale up or down to meet workload requirements, instead of pre-provisioning

Bill on a per-second basis

Common scenarios

Workloads with unpredictable and intermittent usage patterns or performance requirements

Workloads where the requirements are unknown and you can delegate compute sizing to the service

Databases with provisioned compute...

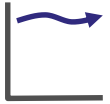
Provision compute resources upfront

Bill on an hourly basis

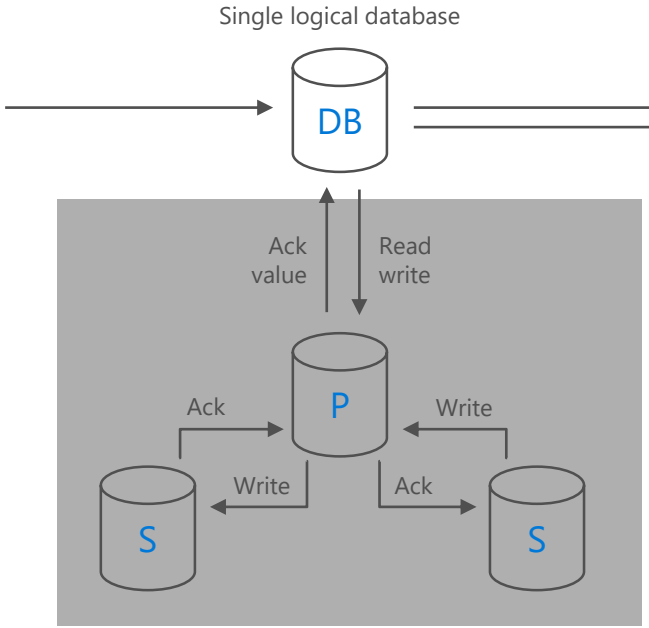
Common scenarios

Workloads with regular and substantial compute utilization

Multiple databases with bursty usage patterns that can be consolidated into a single server and use *elastic pools* for better price optimization

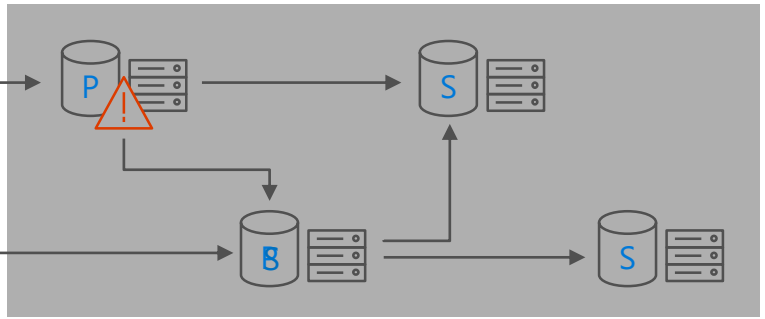


High-availability platform



Reads are completed at the primary

Writes are replicated to secondaries



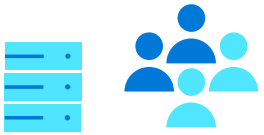
Recovery from machine failure

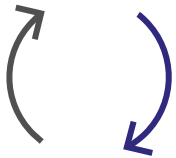
Critical capabilities:

- ✓ Create new replica
- ✓ Synchronize data
- ✓ Stay consistent
- ✓ Detect failures
- ✓ Failover
- ✓ 99.995% SLA

BCDR: Roles and responsibilities

- Azure SQL Database
  - Geo-distributed service
  - Customer metadata protection and recovery
  - Transparent high availability and data protection from local platform failures
  - Automatic geo-distributed backups
  - Automatic data synchronization of geo-replicated databases
  - Platform compliance testing and certification
  - Alert to impacted customers during regional failures
- Customer
  - Implementing retry logic in your application
  - Detecting user errors and initiating point-in-time restore
  - Planning, database prioritization, and region selection for disaster recovery
  - Initiating geo-restore to selected region
  - Application disaster recovery drills

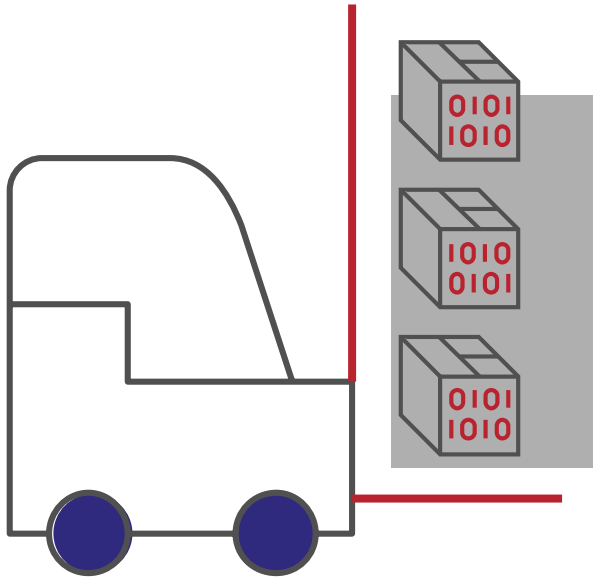




Long-term data retention

Automatically created with LTR capability in Azure SQL Database

- Full database backups
- Store backups for up to for up to 10 Years
- Read-access geo-redundant storage (RA-GRS)



Export a database

- Generate a BACPAC in external storage and hydrate as needed

Point-in-time restore

Automatic backups

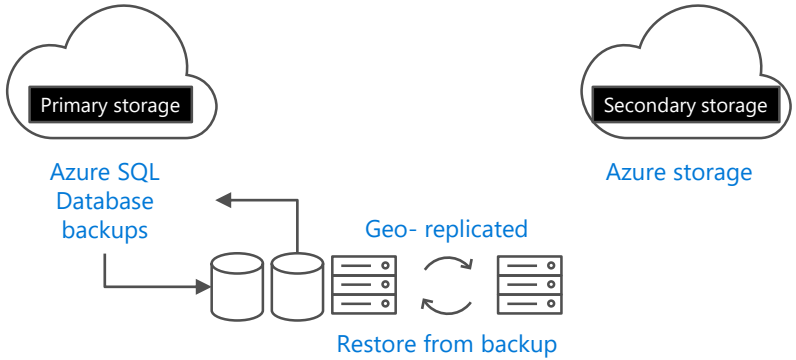
- Full backups weekly, differential backup daily, log backups every 5 minutes
- Daily and weekly backups automatically uploaded to geo-redundant Azure Storage

Self-service restore

- Point-in-time up to a second granularity
- REST API, PowerShell, Azure CLI, Azure portal
- Creates a new database in the same logical server

User-controlled retention policy

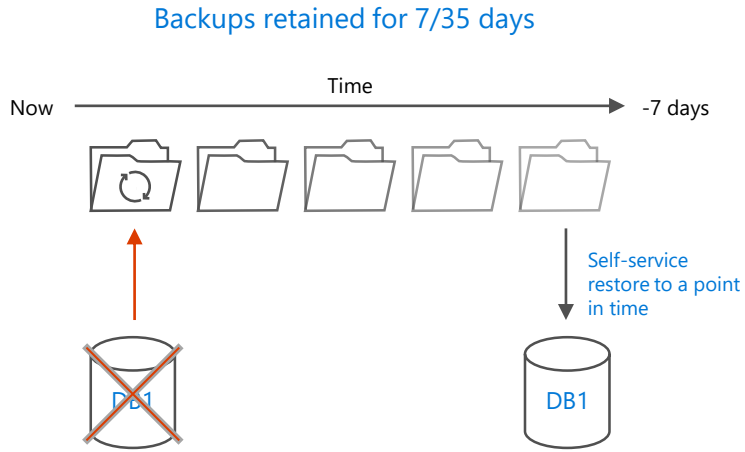
- 7 days default retention in all service tiers
- Up to 35 days of additional retention if required at additional cost
- Choice of storage tier for data sovereignty



Database recovery

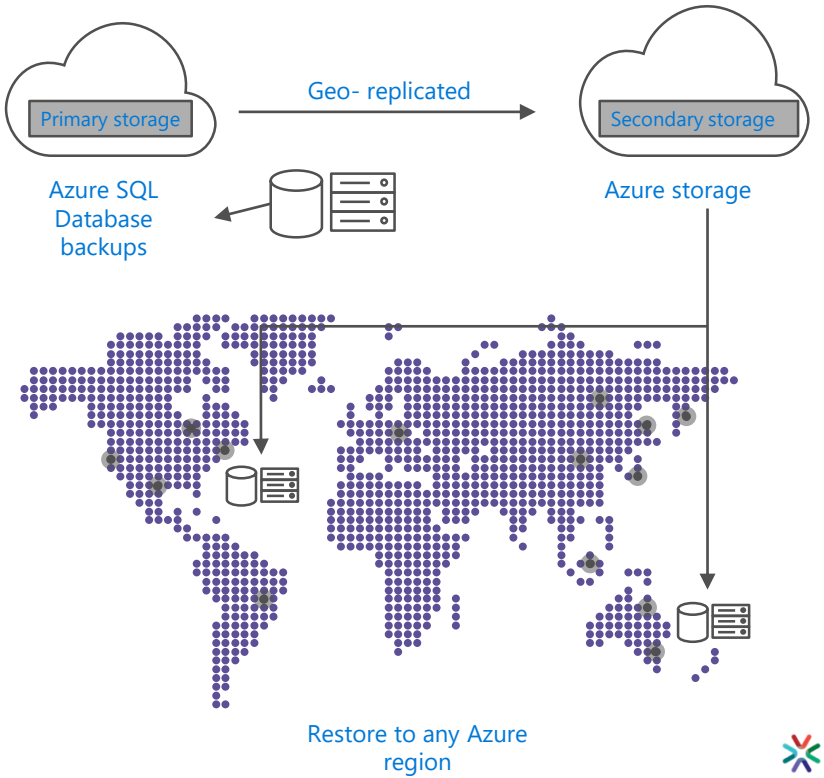
Restoring a deleted database

- Restores the database to any point in time within the retention period
- Creates a new database on the server used by the original database
- You can choose to switch over to the restored database or use queries to recover data



Geo-restore protects from disaster

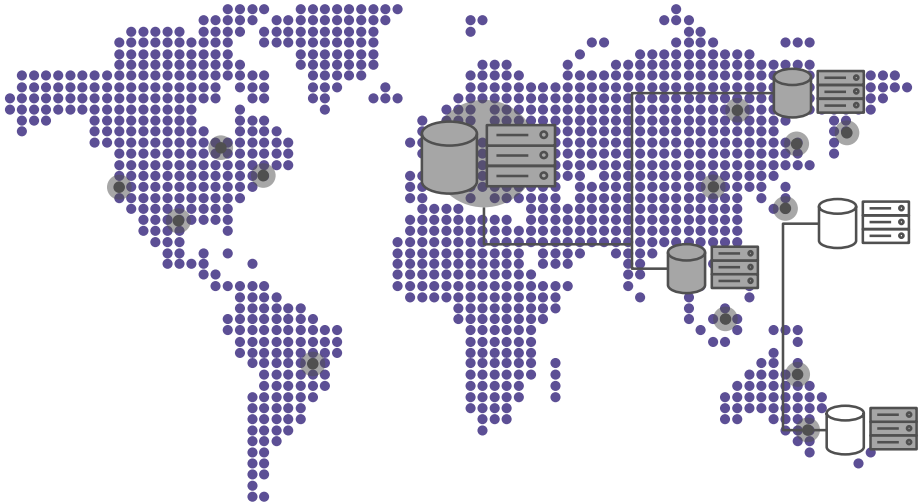
- Self-service restore API
- Built on geo-redundant Azure Storage
- Restores last replicated backup to any Azure region as a new database
- No extra cost, no capacity guarantee
- Estimated recovery time <12h, RPO=1h



Active geo-replication

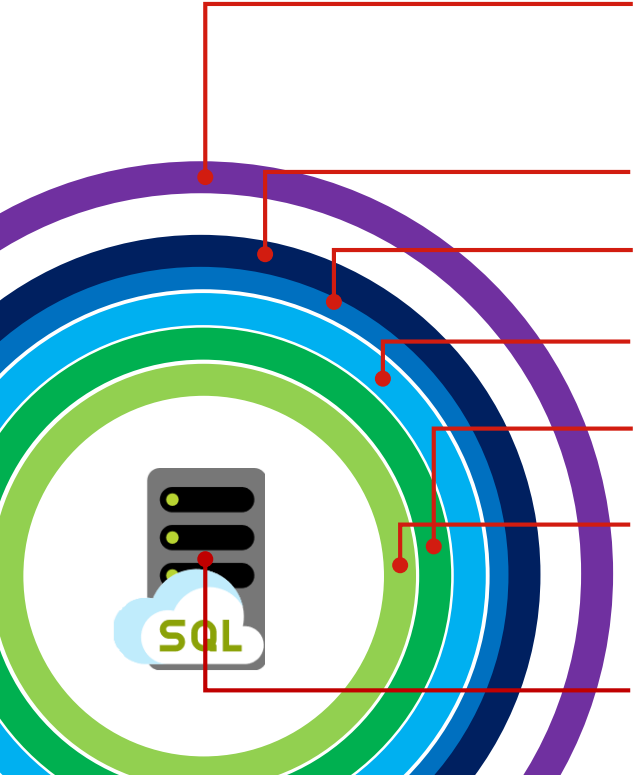
Mission-critical business continuity on your terms, via programmatic APIs

Service levels	All
Readable secondaries	Up to 4 (16 possible)
Regions available	Any Azure region
Replication	Automatic, asynchronous
Manageability tools	REST API, PowerShell, Azure CLI, or Azure Portal
Recovery time objective (RTO)	<30 sec
Recovery point objective	<5sec
Failover	On demand



Up to 4 secondaries

Enterprise-grade security that is easy to use



PHYSICAL SECURITY

NETWORK SECURITY

CLUSTER SECURITY

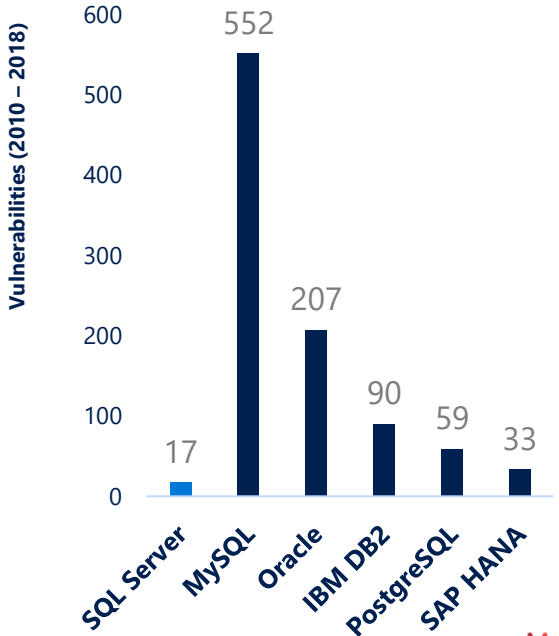
ACCESS MANAGEMENT

THREAT PROTECTION

INFORMATION PROTECTION

CUSTOMER DATA

Trusted: most secure over last 7 years



PASS

Azure SQL: The database for all seasons

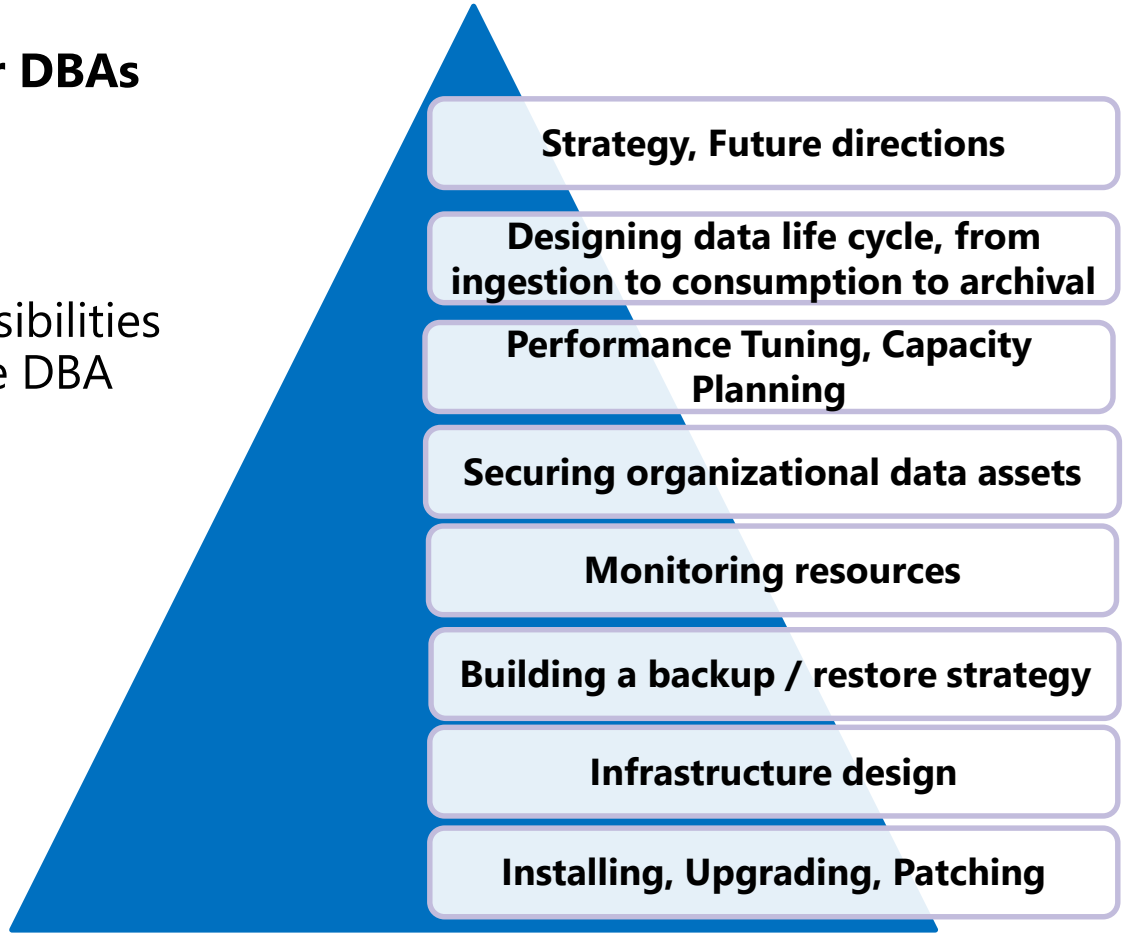
Best for all Developers

- "Batteries Included"
- Awesome relational database with 25+ years for product development and 40+ years of research
- Wide range of development languages and tools
- Write code once, deploy anywhere (in any form factor)
- Control and abstraction on storage format (row store and column store)
- Features for modern apps (JSON, Graph, Spatial, memory-optimized tables, temporal tables, and so on)
- Encryption, Masking and Row-Level Security
- Blocking and Non-Blocking Transactions
- Wide range of prices and elasticity

Azure SQL: The database for all seasons

Best for DBAs

Responsibilities of the DBA



Value to the organization

Azure SQL: The database for all seasons

Best for SaaS Applications

- SQL Server has grown in association with a variety of ISV applications
- Wide range of prices and elasticity:
- Flexible design and deployment choices for multi-tenant applications
- Enterprise-class performance, scale, reliability, security
- Intelligence Query Processing, and Intelligent insights

Azure SQL: The database for all seasons

Best for Enterprise Applications

- Enterprise-class built-in Business Continuity (high availability, disaster recovery) backed by financial SLA
- Security "Always a step ahead"
- Meets varieties of government and industry compliance requirements
- World's best price / performance
- Well-integrated with overall data ecosystem

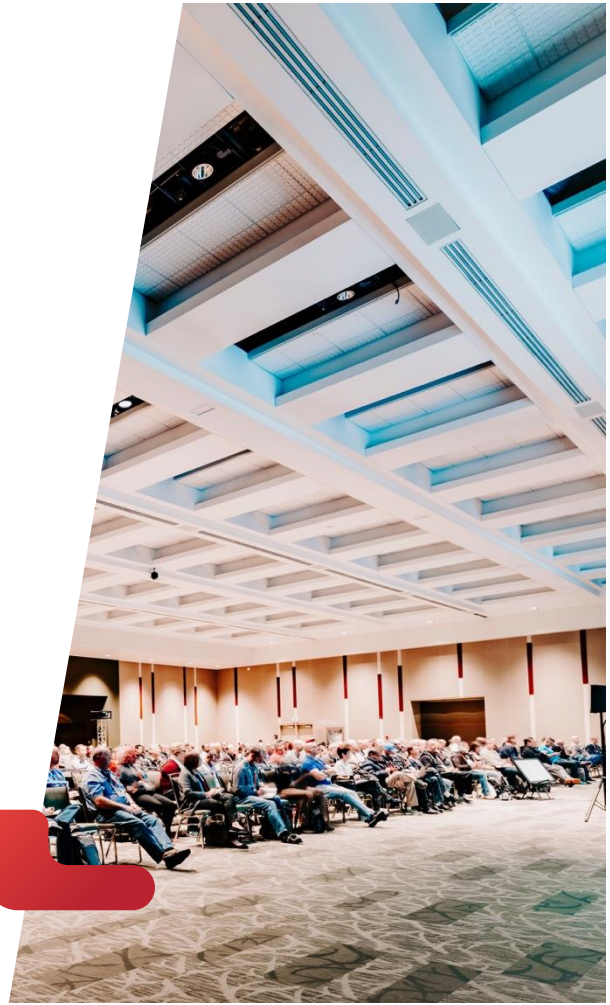
PASS

PASS SUMMIT2019

Azure SQL Database

Managed Instance

Jovan Popovic, Mladen Andzic
Program Managers, Microsoft



Azure SQL

SQL virtual machines

Best for migrations and applications requiring OS-level access

SQL virtual machine

- SQL Server and OS server access
- Expansive SQL and OS version support
- Automated manageability features for SQL Server

Managed instances

Best for most lift-and-shift migrations to the cloud

Single instance

- SQL Server surface area (vast majority)
- Native virtual network support
- Fully managed service

Instance pool

- Resource sharing between multiple instances to price optimize
- Enables migration of many small instances at scale
- Fully managed service

Databases

Best for modern cloud applications. Hyperscale and serverless options are available

Single database

- Hyperscale storage (up to 100TB)
- Serverless compute
- Fully managed service

Elastic pool

- Resource sharing between multiple databases to price optimize
- Simplified performance management for multiple databases
- Fully managed service

PASS

Why Azure SQL Managed Instance?

Instance-level features

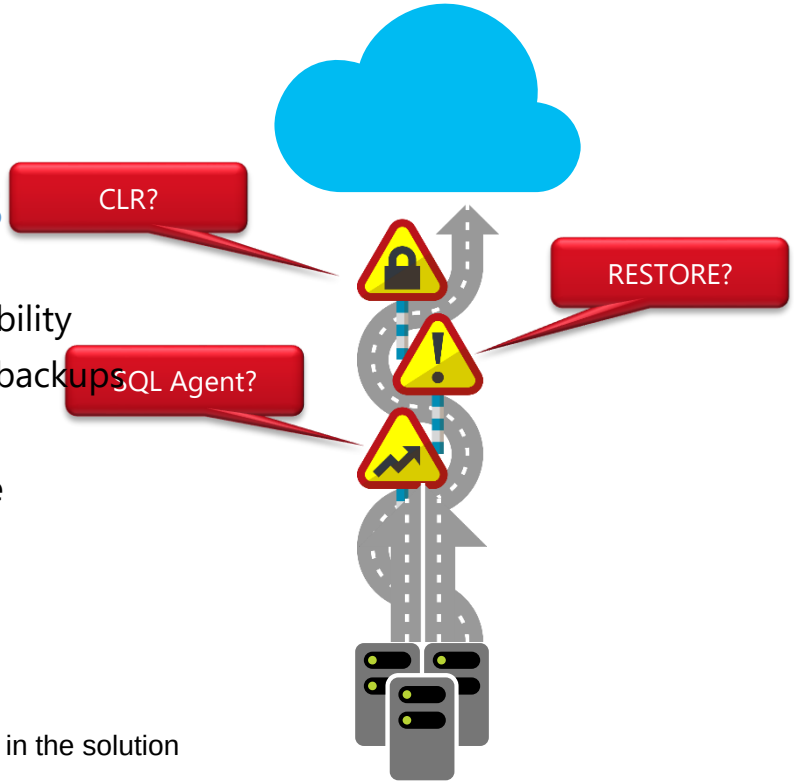
- Cross-DB queries
- Linked servers
- CLR
- SQL Agent
- Restore
- Service Broker
- Server-level objects
- Database mail

Network isolation

Deployed in VNet
Private IP address

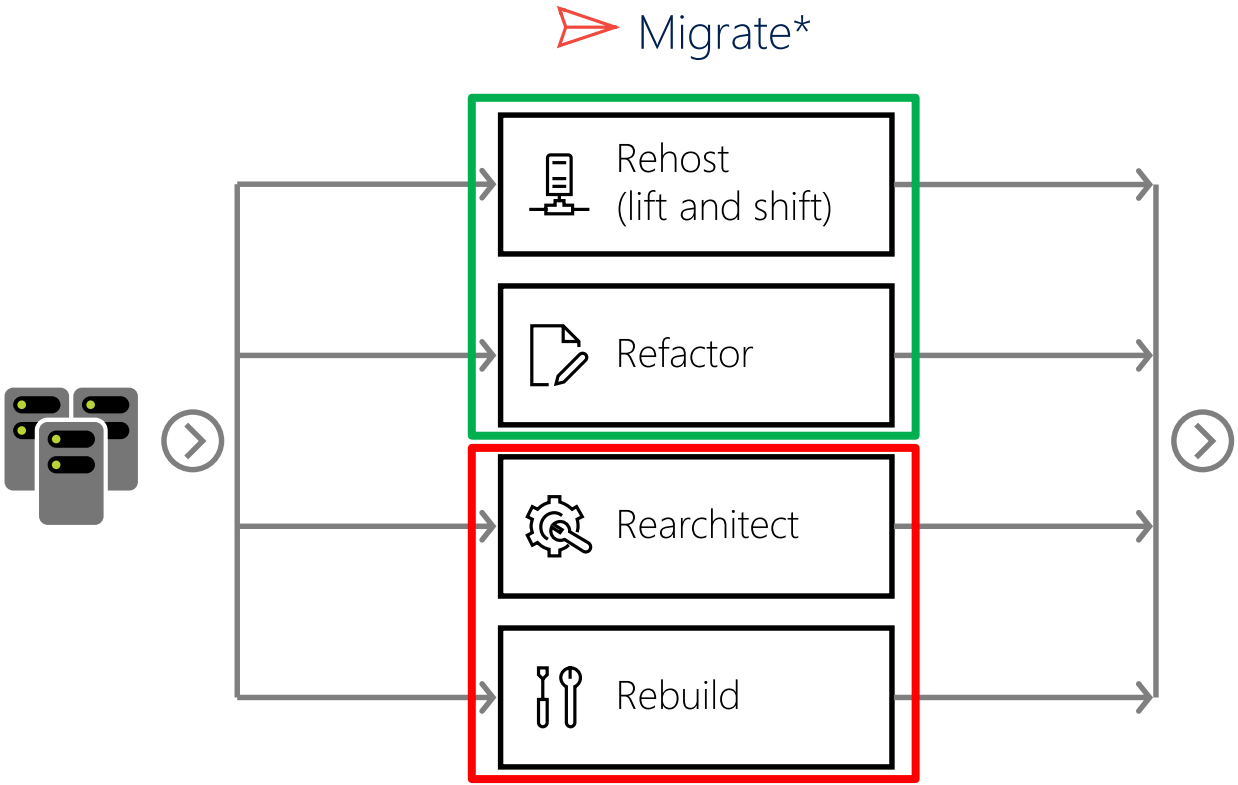
Still full PaaS

High availability
Automatic backups
Patching
Intelligence



Migration to cloud may require some changes in the solution

Migration to Cloud



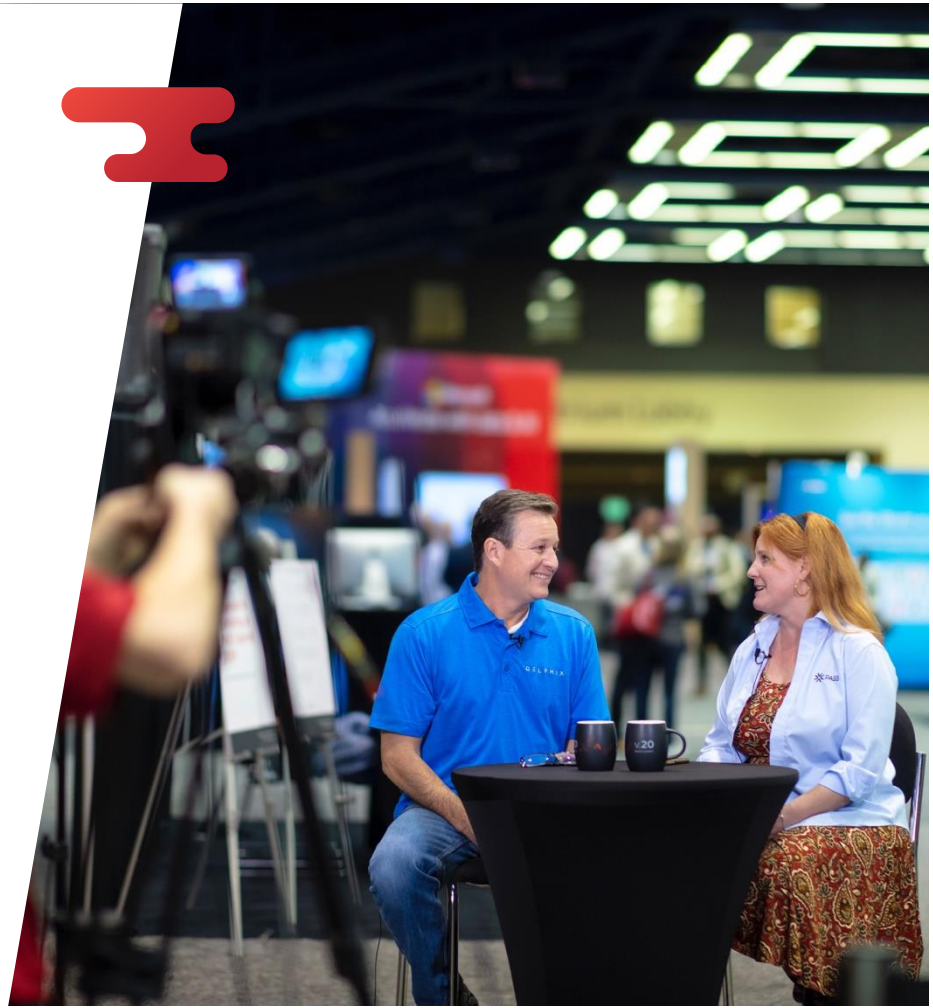
Enables rehosting or light refactoring for most SQL apps

Reduces the need to rearchitect or rebuild your apps

DEMO

Migration Assessment

Data Migration Assistant (DMA)



Proof of Concept – Best Practice

Start simple

Default settings

Public endpoint

Native backup / restore

General Purpose service tier

Evolve and optimize

Customized settings

Customized security

Online migration

Consider Business Critical

Auto-failover groups

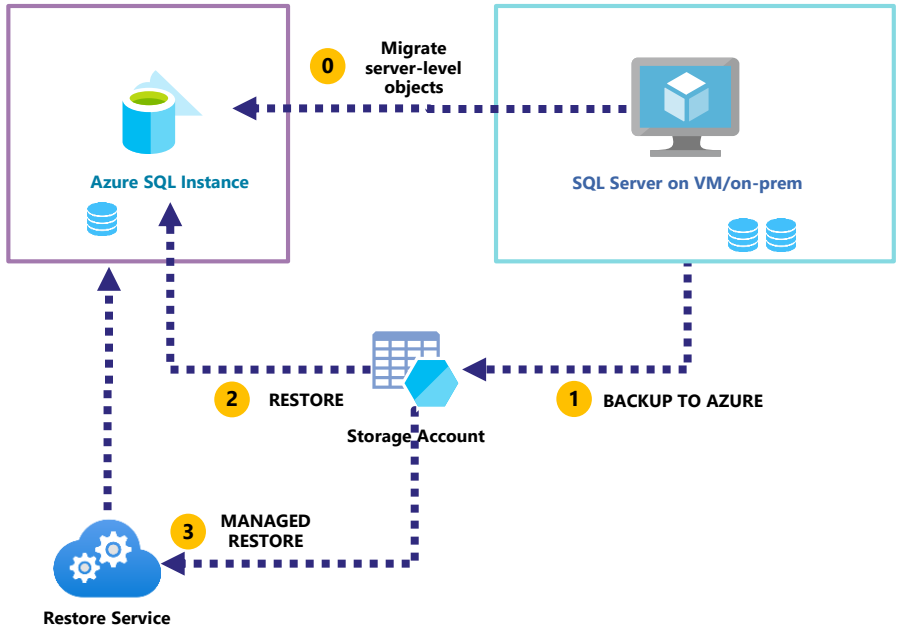
Easy Database Migration

Offline

- Native backup/restore

Online

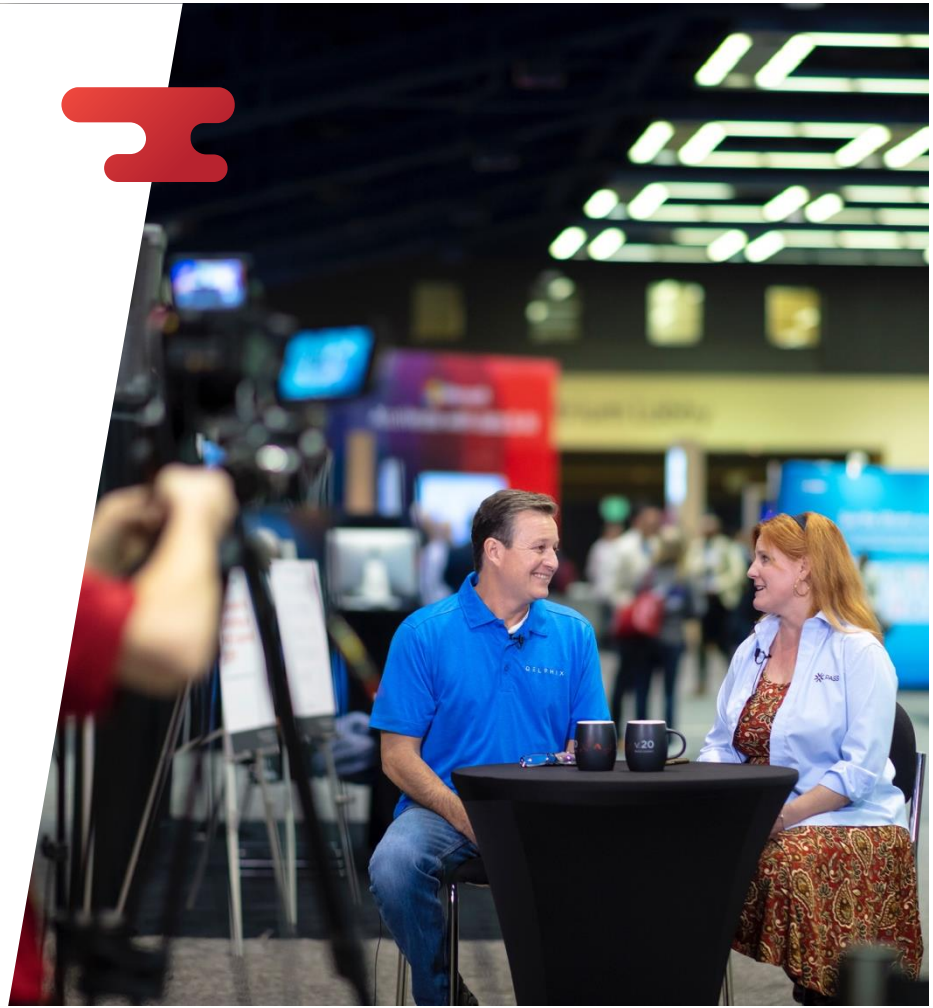
- Data Migration Service
- Log shipping



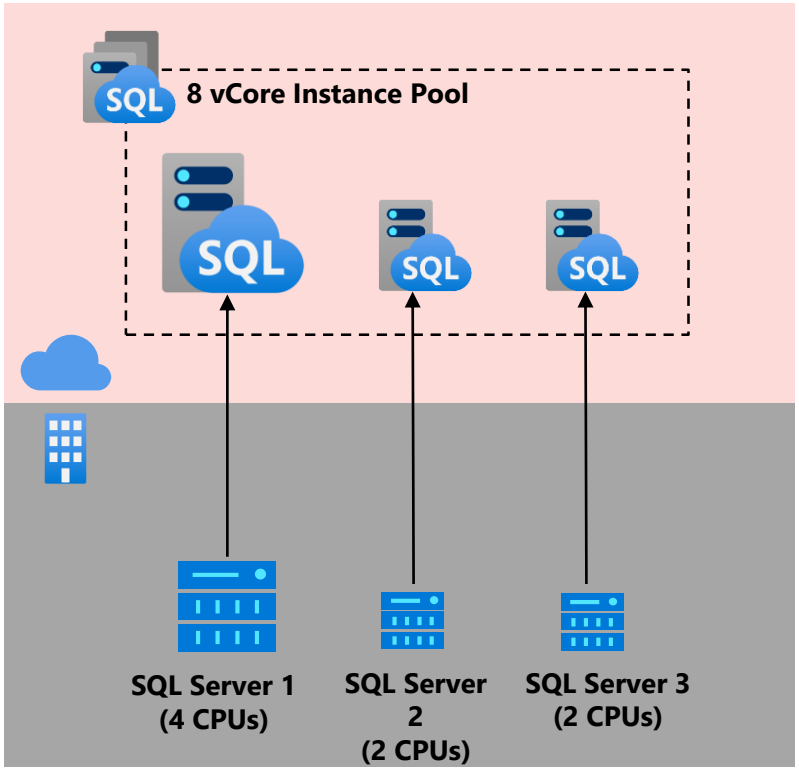
DEMO

Migration

Database restore



Instance Pools (preview)



Migrate multiple **small SQL instances** together to a fully managed instance pool

Add instances starting from 2 vCores up to your pool's limit.

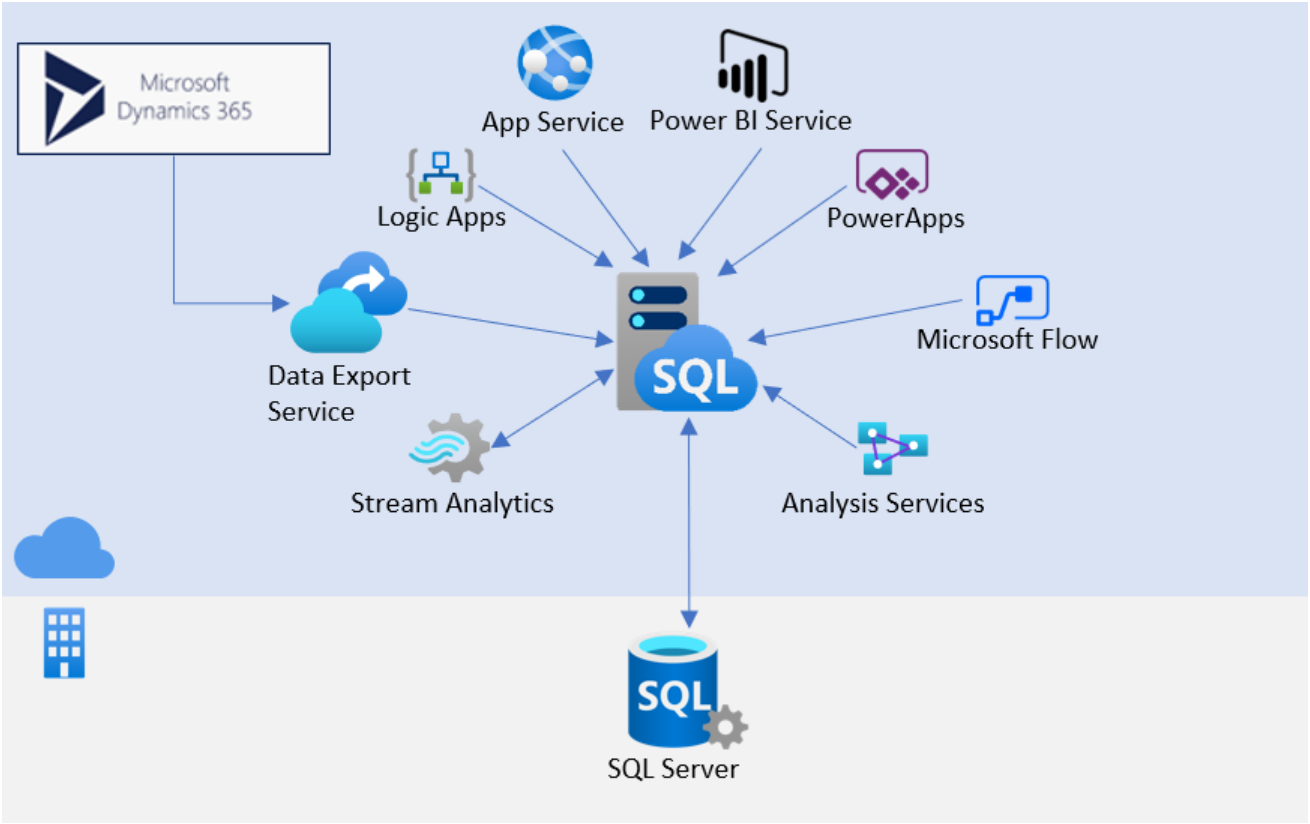
Higher "DBs per vCore" density

Benefit from fast provisioning and scaling operations at the instance level

Modernization

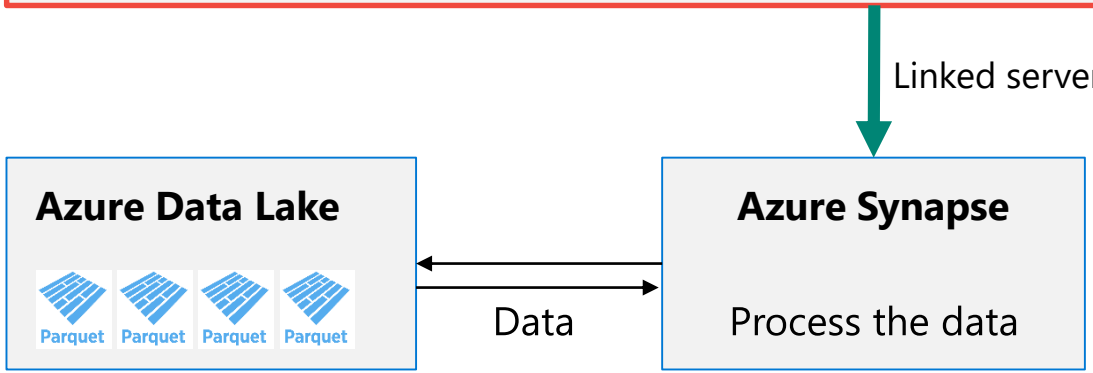
…because migration is just an appetizer

Integration with Cloud Services

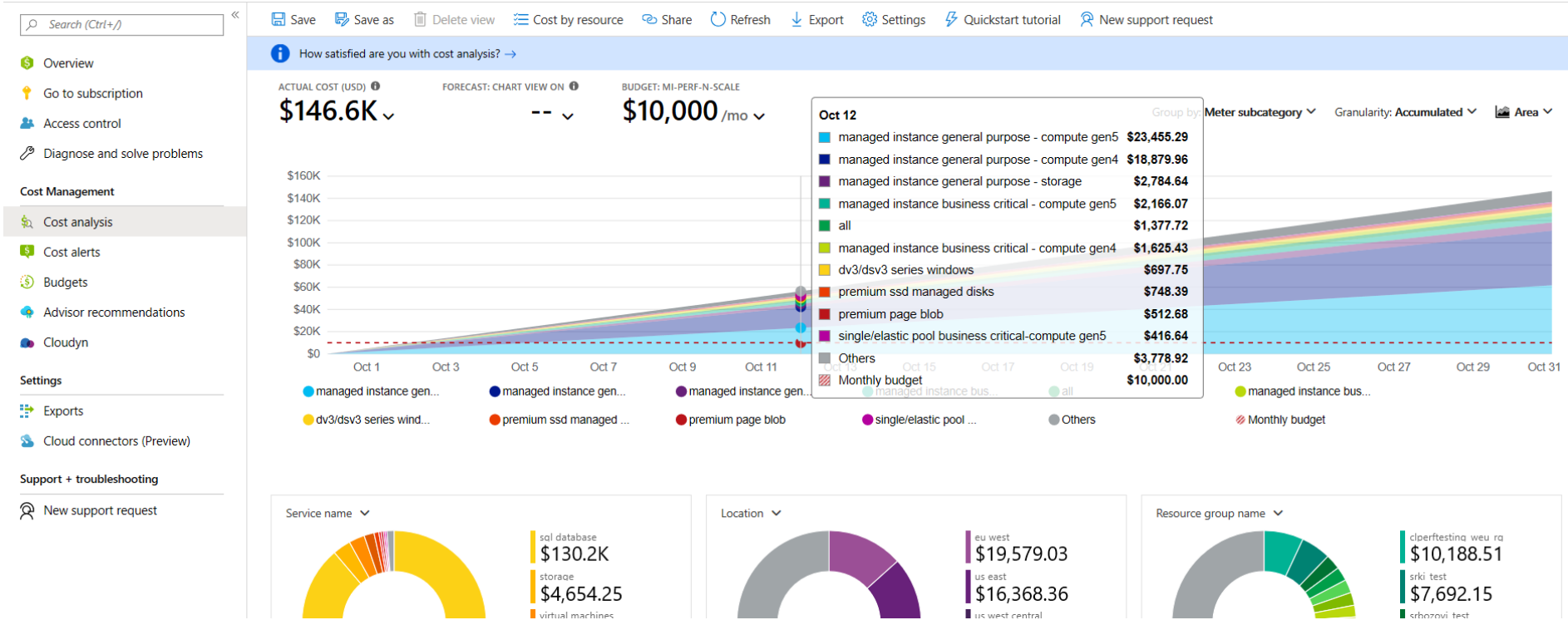


Query big data via Azure Synapse[NEW]

```
SELECT TOP 100 *
FROM        OPENROWSET(
                    BULK 'https://<storage>/path/to/files/*.parquet',
                    FORMAT = 'Parquet'
            ) AS [r]
```



Monitor and Manage Cost



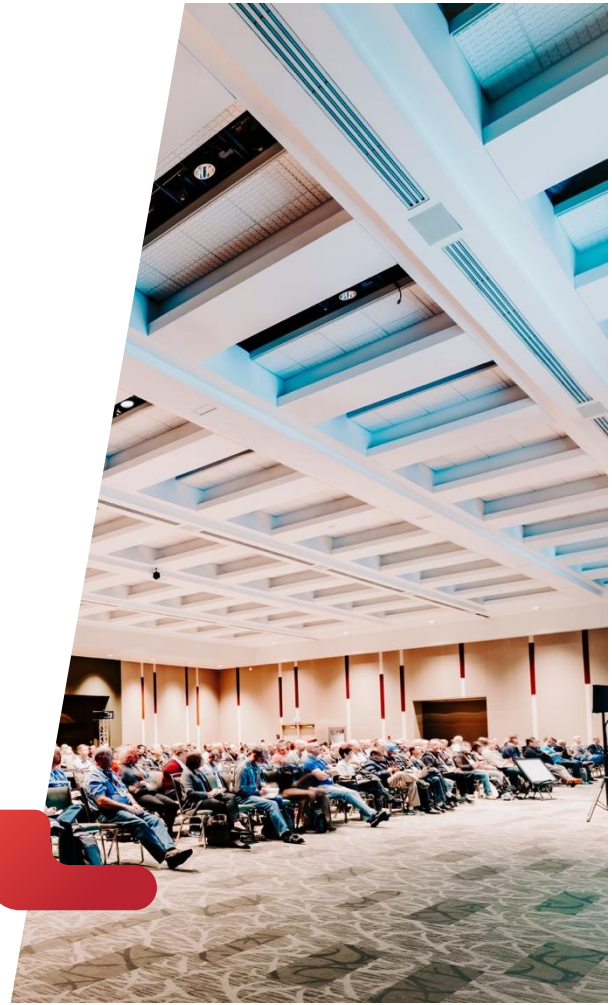
Learn more

Friday, Nov 8, 9:30 AM
SQL Managed Instance: a Fully Managed SQL Server in the Cloud

PASS SUMMIT2019

Networking for Azure SQL DB and Managed Instance (MI)

Rohit Nayak, Sr. Program Manager, Microsoft

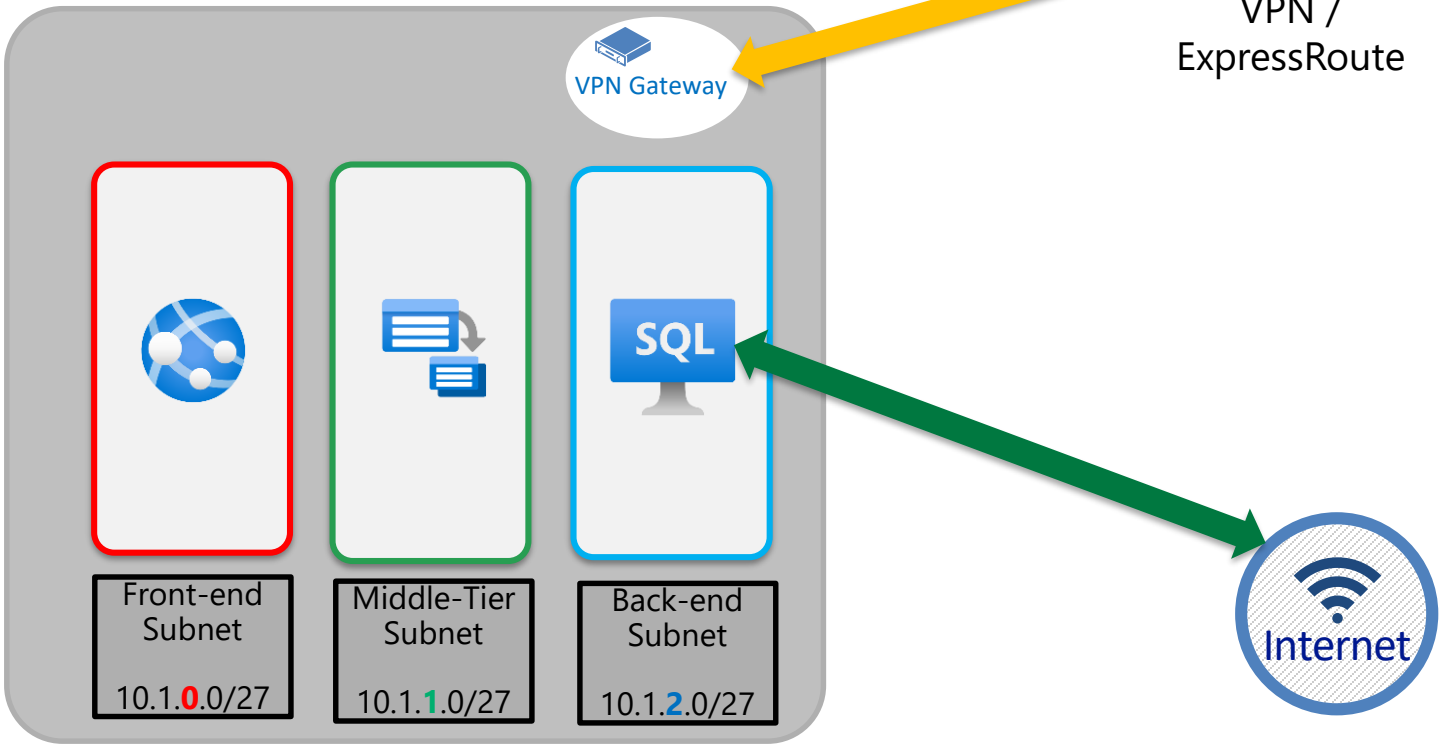


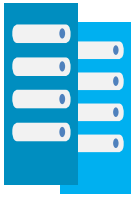
Agenda

1. Azure Networking 101
2. Connectivity Architecture – SQL DB & Managed Instance (MI)
3. Network Access Controls – SQL DB

Azure Networking 101

Virtual **Net**work
10.1.0.0/16 -> [10.1.0.0 : 10.1.255.255]

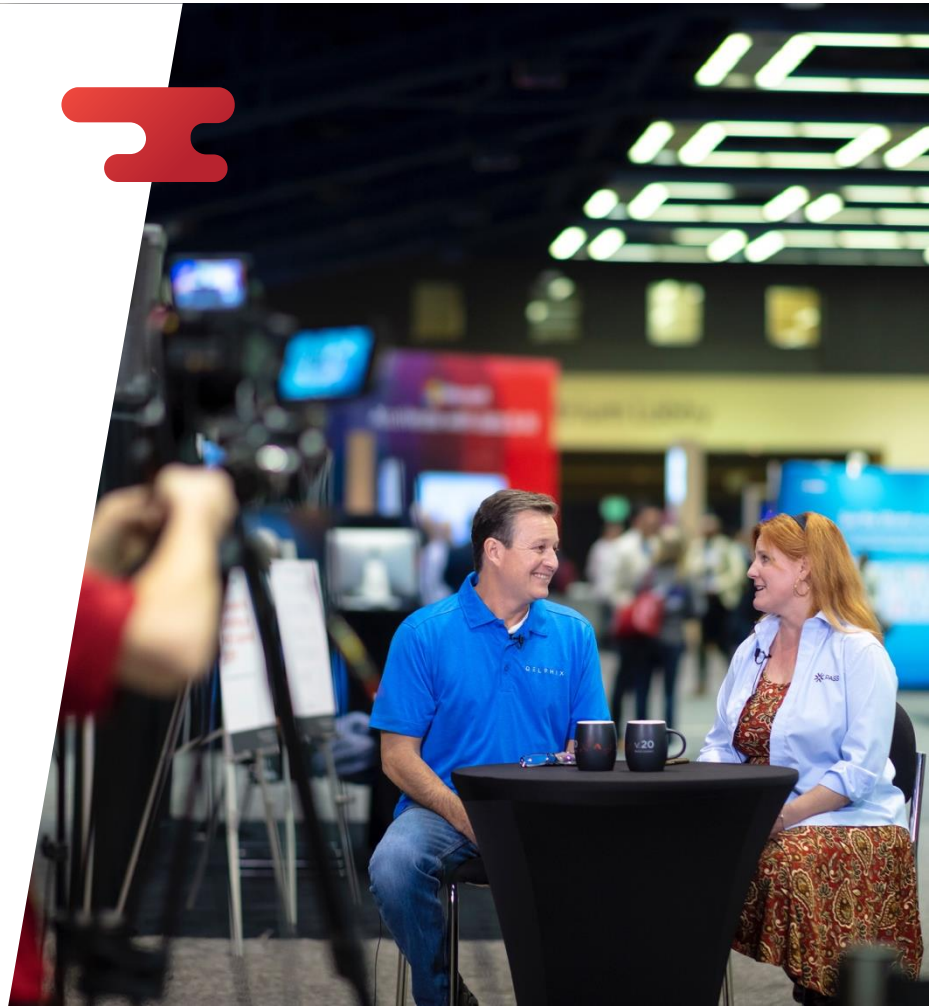




VPN /
ExpressRoute

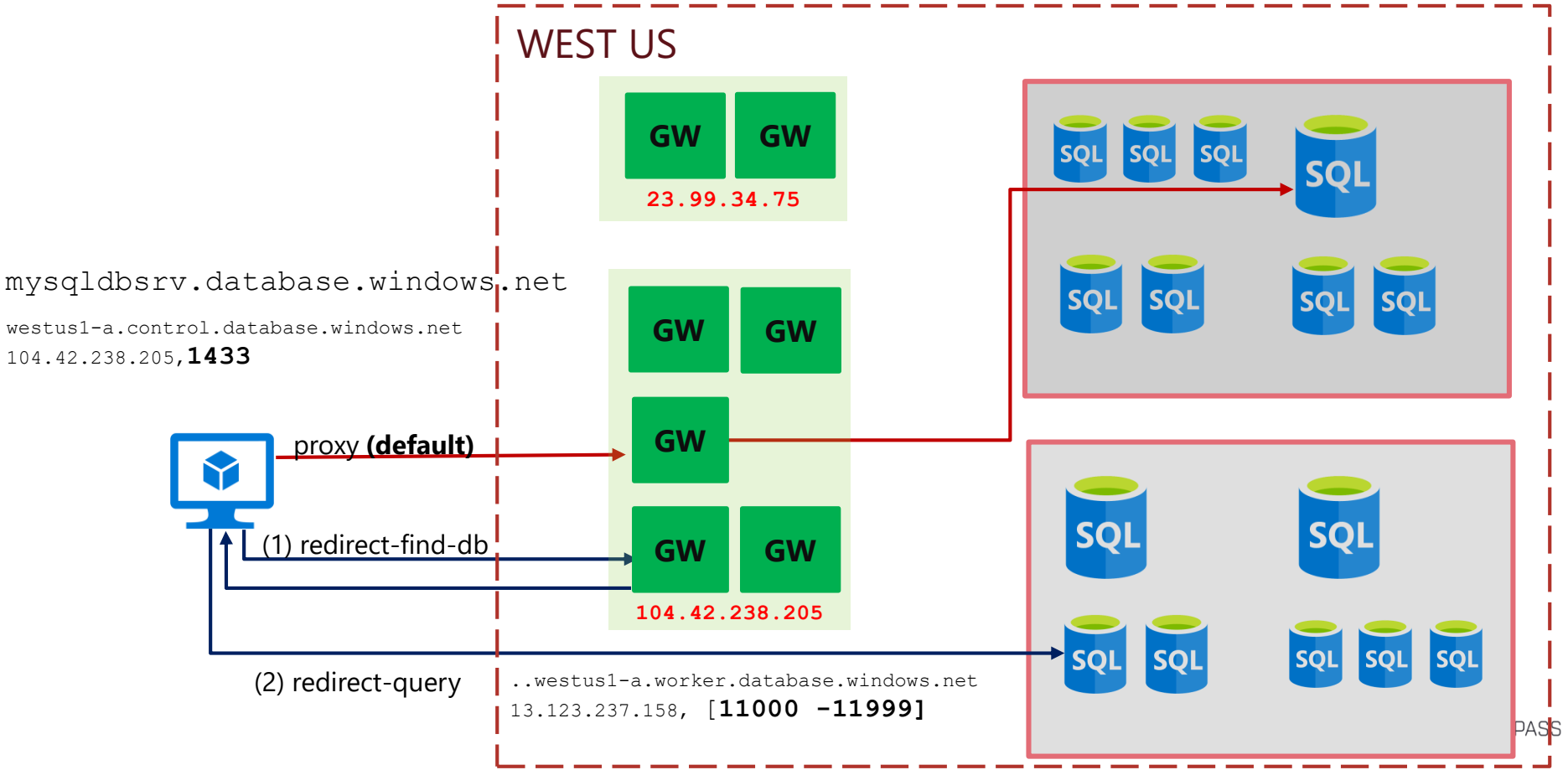
DEMO

Quick tour of Azure Networking

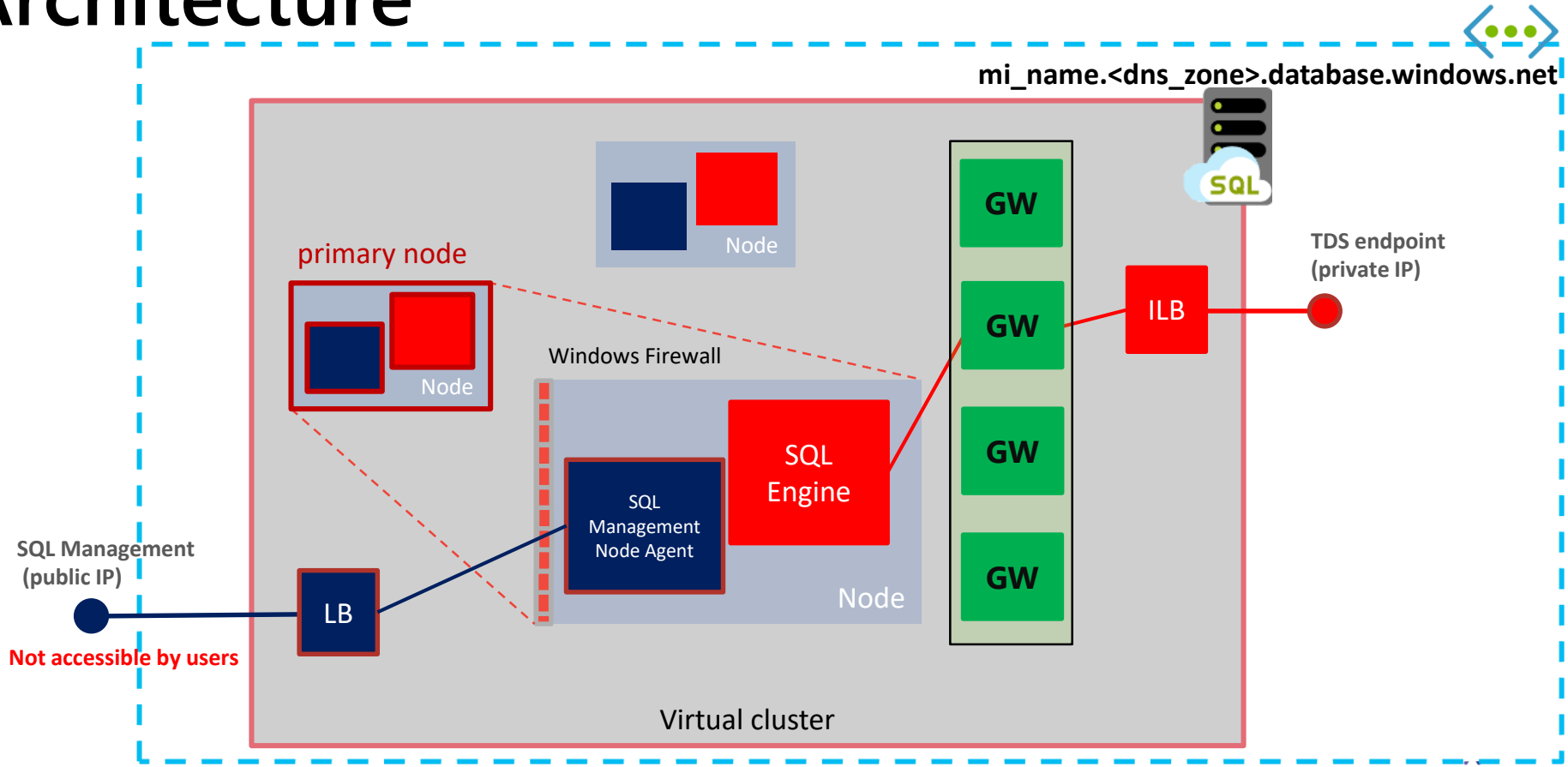


Connectivity Architecture – SQL DB vs MI

Sql Database Connectivity Architecture

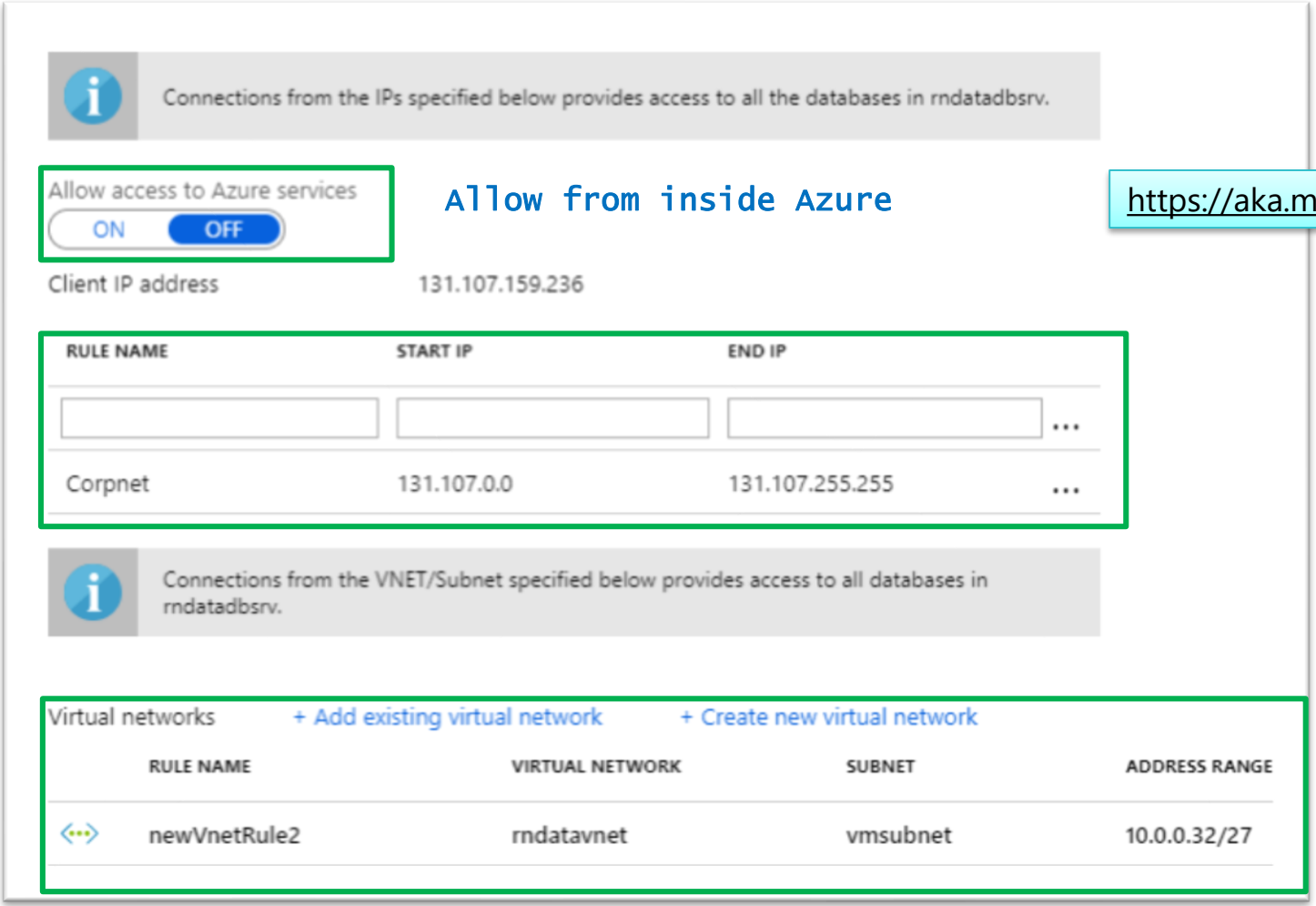


Sql Managed Instance Connectivity Architecture



Network Access Controls – SQL DB

SQL Database network access controls



Allow from inside Azure

https://aka.ms/sqlnetworkaccesscontrols

IP based Firewall rules

Allow from your laptop or specific Azure VM

VNET Firewall rules

Allow from specific subnet over private IP

PASS

Private Link

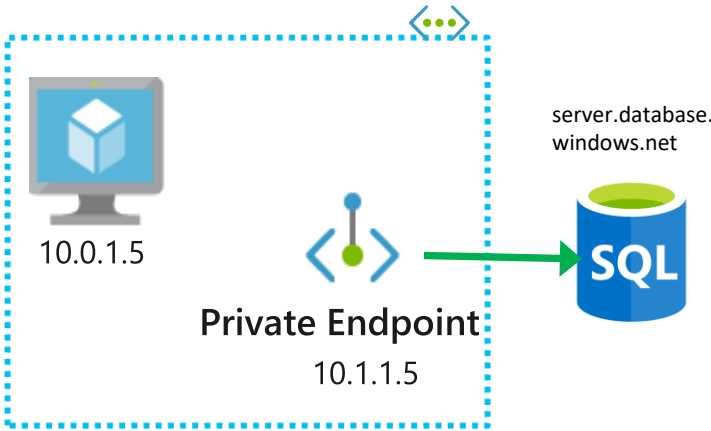
Private connectivity to PaaS Service from within customers VNET

Benefits:-

- Private endpoint for Azure SQL (server level)
- On-premises connectivity via Express Route/VPN
- Data exfiltration protection
- Can be part of network monitoring strategy using a Network Virtual Appliance (NVA)

Currently in public preview for:

- Azure SQL Database (singleton databases)
- Azure SQL Data Warehouse
- Azure Storage



VIDEO

Network Access Controls Demo

Resources

- SQL Database Connectivity Architecture [https://docs.microsoft.com/en-us/azure/sql-database/sql-database-connectivity-architecture]
- SQL Managed Instance Connectivity Architecture[https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-connectivity-architecture]
- Connect to Azure SQL Database V12 via Redirection [https://techcommunity.microsoft.com/t5/DataCAT/Connect-to-Azure-SQL-Database-V12-via-Redirection/ba-p/305362]
- Overview of Network Access Controls[http://aka.ms/sqlnetworkaccesscontrols]
- Impact of removing Allow Access to Azure Services [https://docs.microsoft.com/en-us/azure/sql-database/sql-database-networkaccess-overview#allow-azure-services]
- IP Based Firewall [https://docs.microsoft.com/en-us/azure/sql-database/sql-database-firewall-configure]
- Vnet Firewall Rules [https://docs.microsoft.com/en-us/azure/sql-database/sql-database-vnet-service-endpoint-rule-overview]

Thank You

Rohit Nayak

@sqlrohit

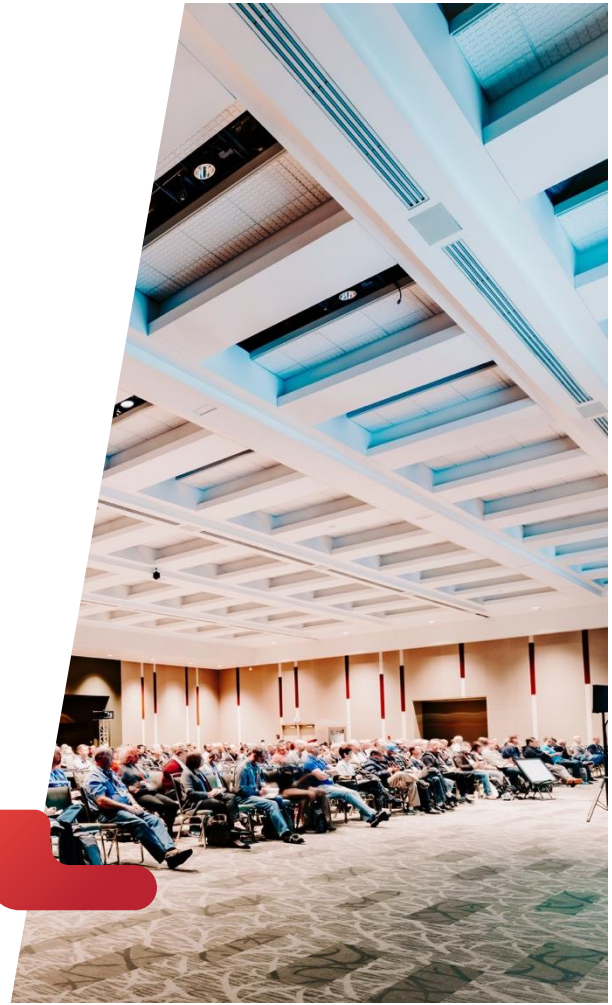
rohit.nayak@microsoft.com

PASS

PASS SUMMIT2019

Security for Azure SQL

Andreas Wolter
Program Manager SQL Security



Enterprise-class Security Controls





Identity & access management

- Authentication options: Azure AD, SQL Auth, Windows Auth
- Azure RBAC
- Row-level security
- Roles & Permissions

Data protection

- Encryption-in-use (Always Encrypted)
- Encryption-at-rest (TDE)
- Encryption-in-flight (TLS)
- User-managed keys
- Dynamic Data Masking

Network security

- VNET
- Firewall Rules, NSG
- Private Link

Monitoring, logging & auditing

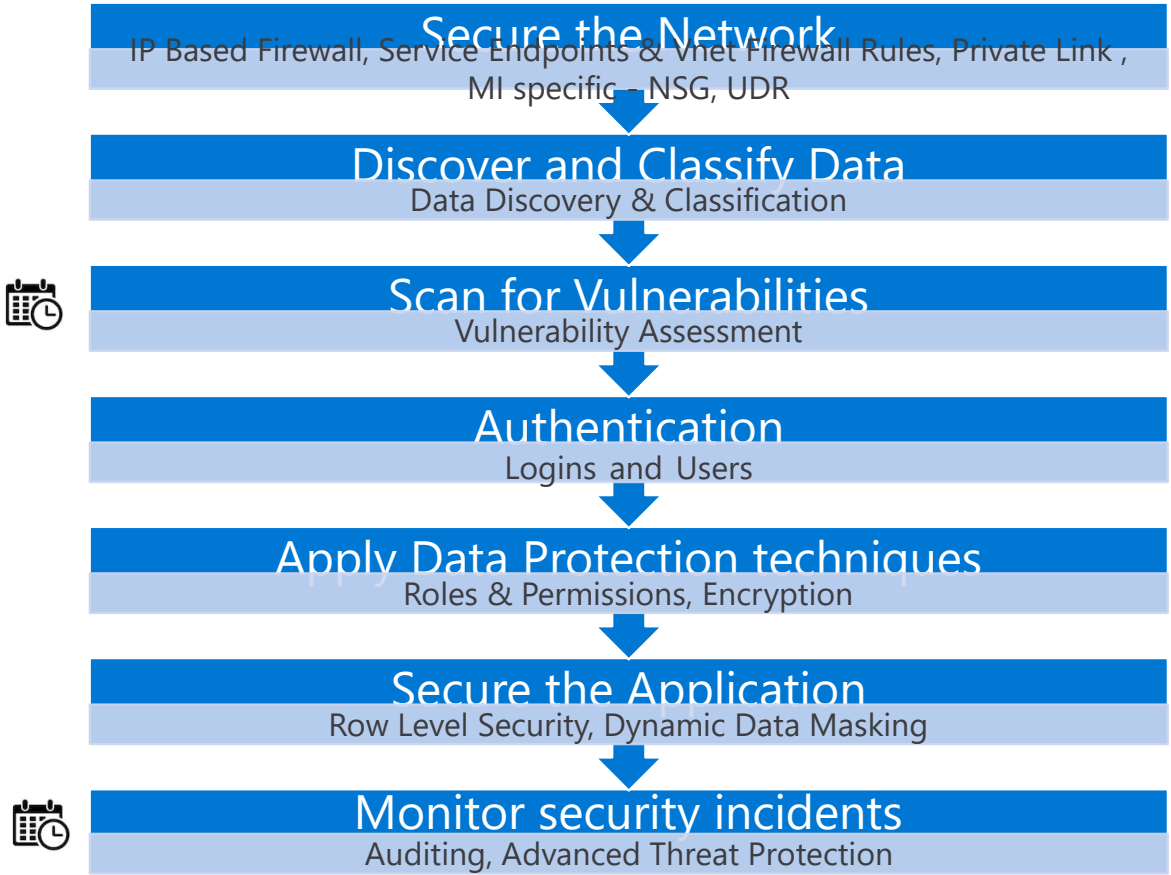
- Advanced Threat Detection
- SQL Audit
- Audit Integration with Log Analytics and Event Hubs

Security management

- Vulnerability Assessment
- Data Discovery & Classification
- Azure Security Center

+ Partner Solutions: Imperva SecureSphere, IBM Guardium

Securing your databases in Azure SQL



Every step may need to be repeated in case of schema changes, new users or other needs!

= Run continuously /on schedule

PASS

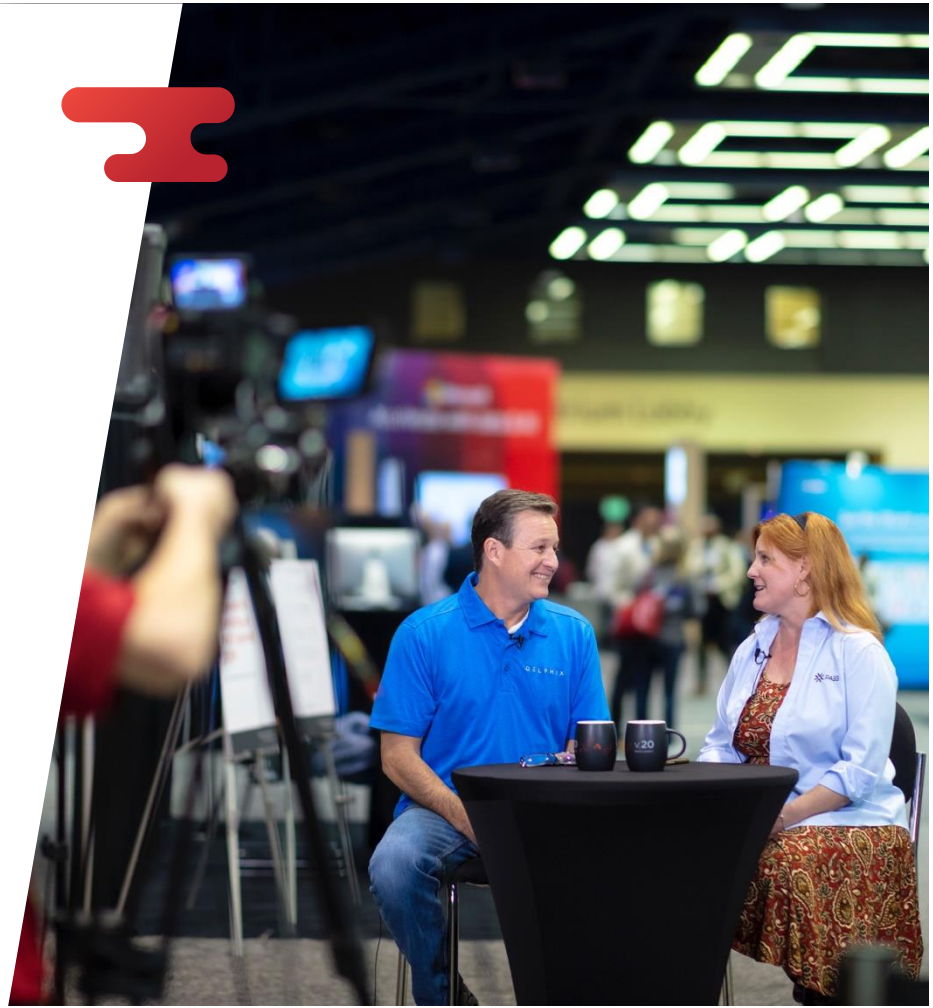
Check your assets

Where to start

- You need to know your assets before you know what to protect and how
- Azure SQL supports you in this process:
  - Use Data Discovery to learn about your data
  - Use Vulnerability Assessment (VA) to check your current configuration and set baselines
    - this should be run on a scheduled basis

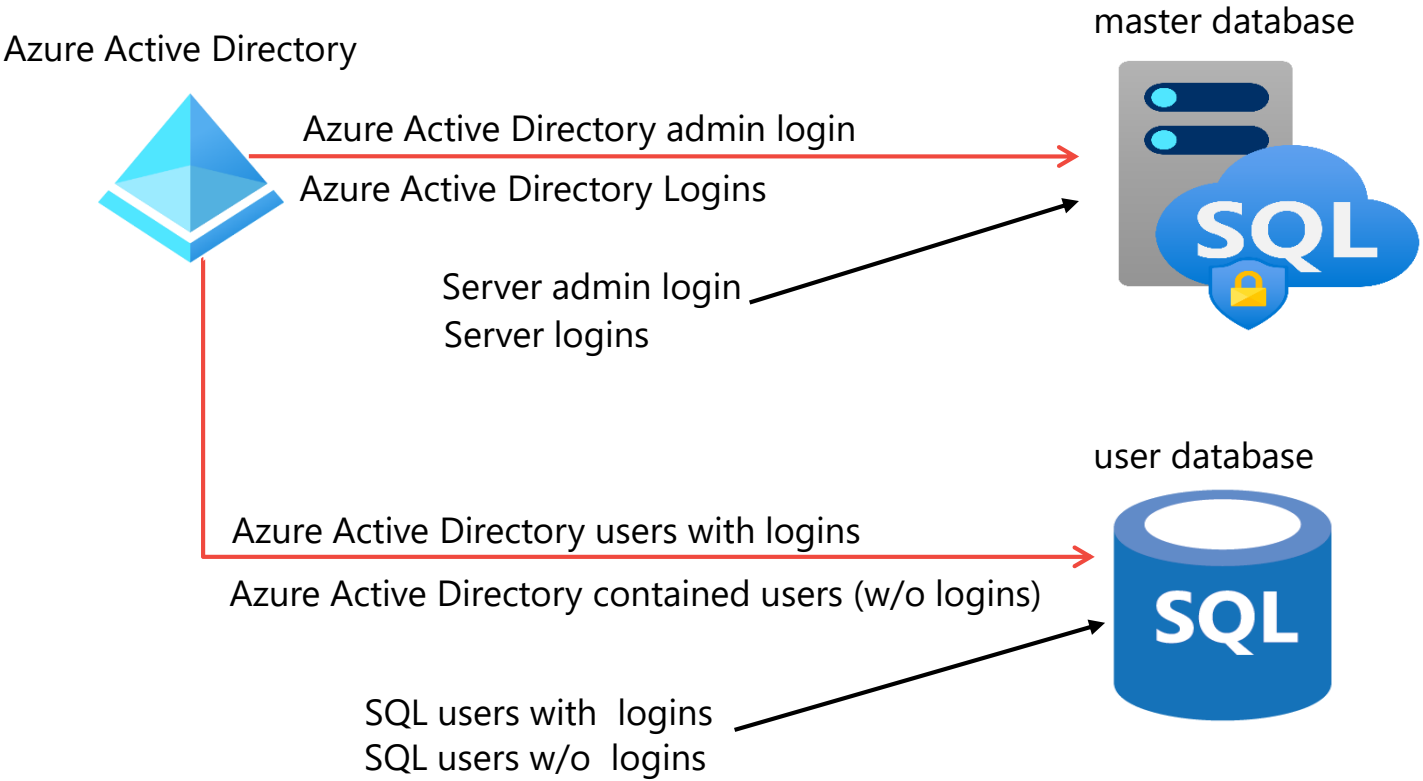
DEMO

Data Discovery & Classification, Vulnerability Assessment

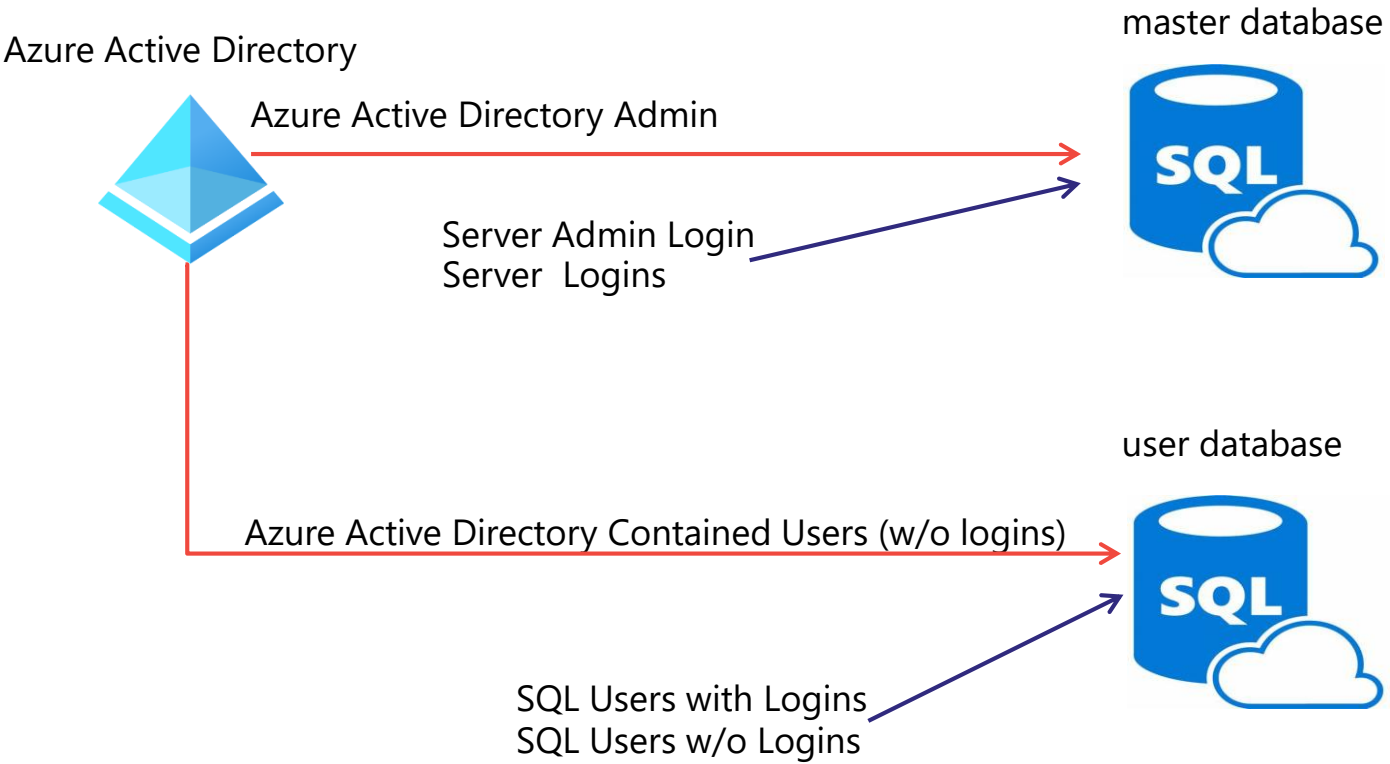


Authentication

GA of Azure AD server principals (Azure AD logins) for Managed Instance



Azure AD for SQL DB



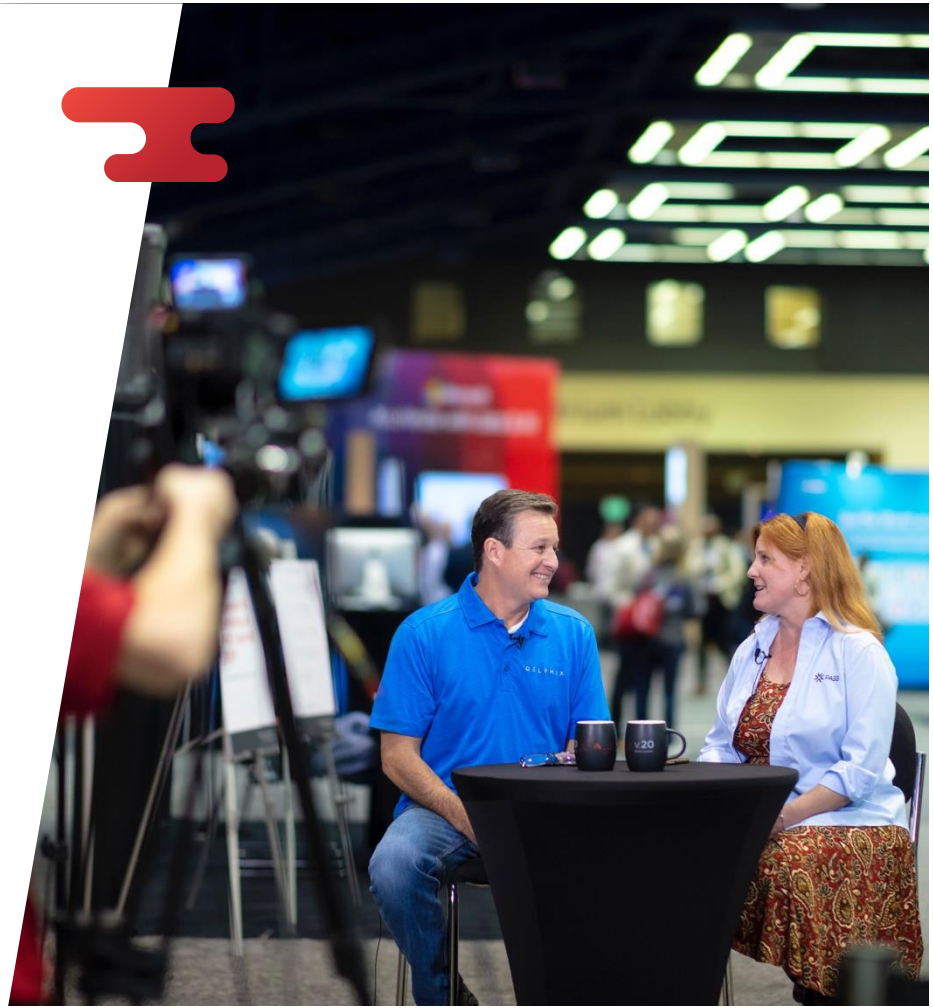
Data Protection techniques

Data Protection techniques

- Permissions and Schemas
  - Use the Principle of Least Privilege when granting permissions
    - Azure SQL comes with up to 240 permissions depending on the SKU
  - Inside the databases, schemas can be used to group objects with similar security requirements
- Roles
  - Roles group permissions together and make security manageable
  - There are built-in roles and you can create custom roles that have permissions that suit your needs
- Add Users to Roles, Grant Permissions to Roles

DEMO

Permissions and Roles



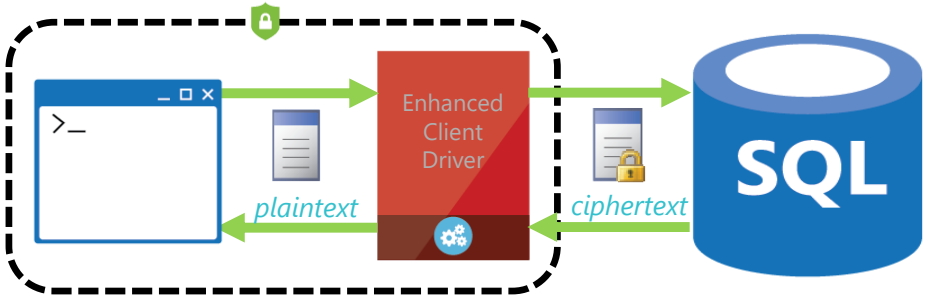
Encryption

- TDE – protects at rest (by default in Azure since 2017)
- Encryption further strengthens security and can prevent data exfiltration
- Use Always Encrypted to protect data in use from high privileged users by applying Separation of Duties between data owner and data manager (like a DBA)

In SQL Server 2016/17 and Azure SQL DB

Always Encrypted

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



Client side Encryption

Client-side encryption of sensitive data using keys that are ***never*** given to the database system

Encryption Transparency

Client driver transparently encrypts query parameters and decrypts encrypted results

Queries on Encrypted Data

Support for equality comparison, including join, group by and distinct operators via deterministic encryption

How Always Encrypted works

Client

```
using (SqlCommand cmd = new SqlCommand(
"SELECT Name FROM Patients WHERE SSN =
@SSN"
 , conn))
{
 cmd.Parameters.Add(new SqlParameter(
  "@SSN", SqlDbType.VarChar, 11).Value =
   "111-22-3333");
 SqlDataReader reader =
  cmd.ExecuteReader();
}
```

SQL Server or Azure SQL Database

```
exec sp_describe_parameter_encryption
@params = N'@SSN VARCHAR(11)'
, @tsql = N'SELECT Name FROM Patients WHERE SSN =
@SSN'
```

Param	Encrypted CEK Value	CMK Store Provider Name	CMK Path
@SSN		AZURE_KEY_VAULT	https://my.vault.azure.net:443/keys/CMK1/b94d985

```
EXEC sp_execute_sql
N'SELECT Name FROM Patients WHERE SSN = @SSN'
, @params = N'@SSN VARCHAR(11)', @SSN=0x7ff6a54ae6d
```

Param	Encrypted CEK Value	CMK Store Provider Name	CMK Path
@Name		AZURE_KEY_VAULT	https://my.vault.azure.net:443/keys/CMK1/b94d985

Name
0x19cae706fbd9

Result set (plaintext)

Name
John Smith

Patients table

Name	SSN
0x19cae7...	0x7ff6a54...
0xfbd9ae...	0x654ae6...

PASS

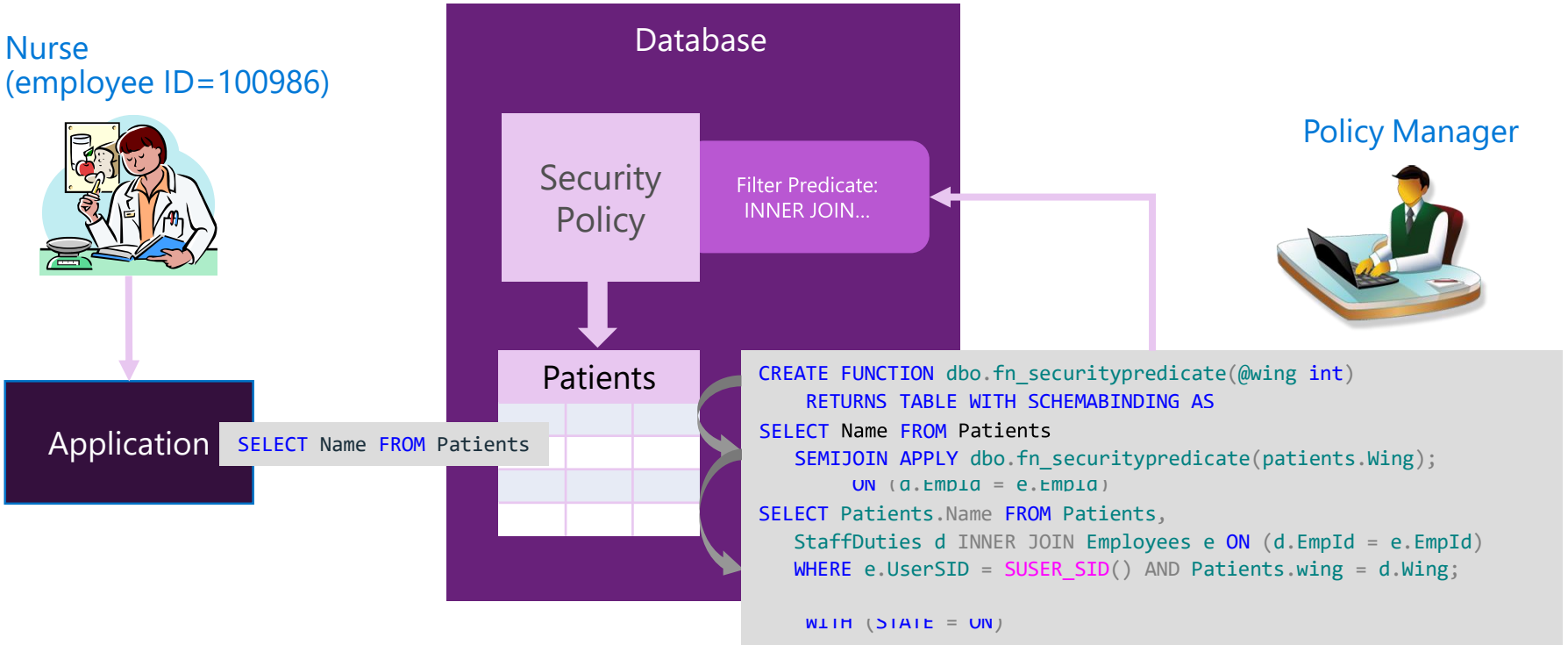
Secure the Application

Application security features

- intended for scenarios where the queries that a user can execute are controlled by a middle-tier application.
- users with ad-hoc query access to the database may be able to infer the existence of filtered data, using side channels.
- Dynamic Data Masking
  - Obfuscates sensitive data in the application, using built-in or custom masking rules
- Row Level Security use case:
  - limit users' access to only certain rows of data in a database

How Row Level Security works

2) Security Policy and predicate are applied to the Patients table



How Row Level Security works

4) Filtered rows are returned and nurse sees only the patients on her wing

Nurse
(employee Id=100986)

Application

Database

Security Policy

Filter Predicate:
INNER JOIN…

Patients

Policy Manager

Patients

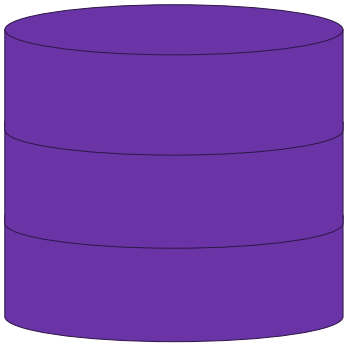
Wing	PatientId	Name
a1	100-207	Jones, Walker
b1	201-677	Smith, Weston
b2	898-045	Jarvis, Anne
a1	009-451	Rickmann, Earnst
b2	922-769	Gray, Jim
a1	801-345	Polka, Dorothy

StaffDuties

EmployeeId	Wing
100986	a1
206985	a2
345906	a3
331225	b1

How Dynamic Data Masking works

1) Security officer defines dynamic data masking policy in T-SQL over sensitive data in Employee table
2) App user selects from Employee table
3) Dynamic data masking policy obfuscates the sensitive data in the query results



```
ALTER TABLE [Employee] ALTER COLUMN [SocialSecurityNumber]
ADD MASKED WITH (FUNCTION = 'Partial(0,"XXX-XX-",2)')

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 DataOfficers
```

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

Normal Endusers

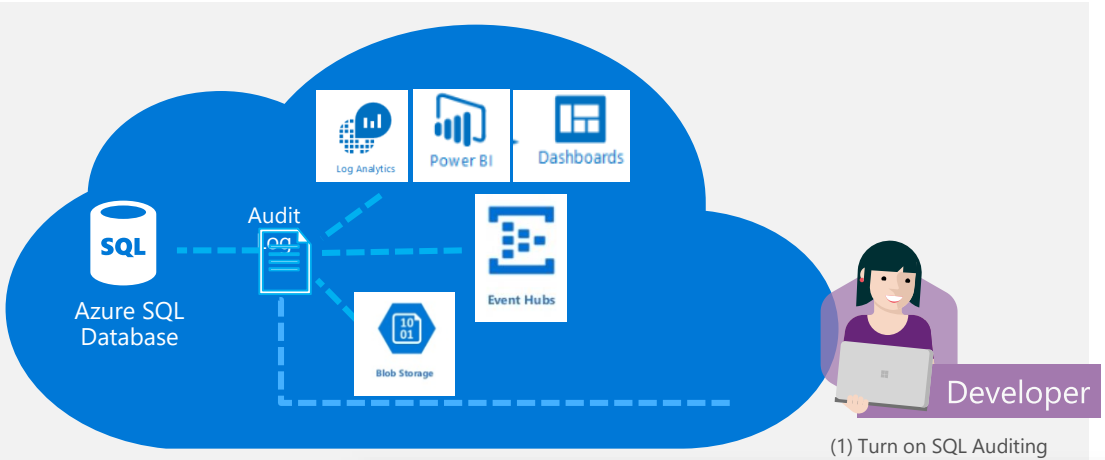
	First Name	Social Security Number	Email	Salary
1	LILA	XXX-XX-XX37	lXX@XXXX.net	8940
2	JAMIE	XXX-XX-XX14	jXX@XXXX.com	19582
3	SHELLEY	XXX-XX-XX28	sXX@XXXX.net	3713
4	MARCELLA	XXX-XX-XX65	mXX@XXXX.net	11572
5	GILBERT	XXX-XX-XX87	gXX@XXXX.net	4487

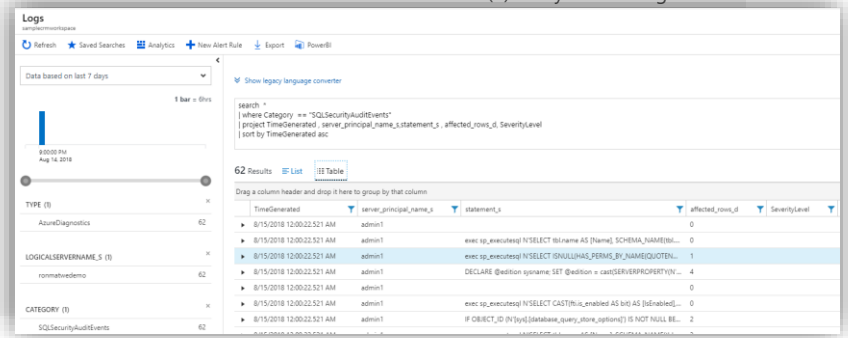
Role DataOfficers

	First Name	Social Security Num...	Email	Salary
1	LILA	758-10-9637	lila.bamett@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

Monitor security incidents

SQL Auditing with Log Analytics and Event Hubs



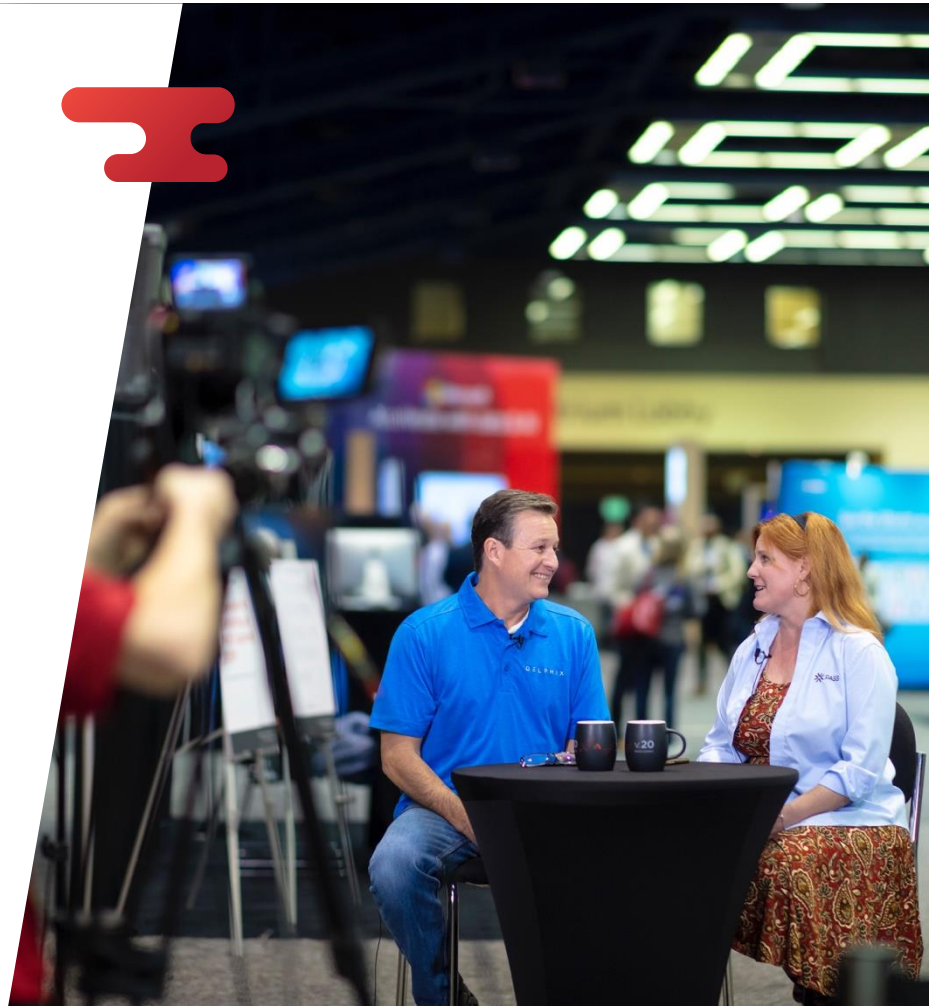


- ✓ Configurable via audit policy
- ✓ SQL audit logs can reside in
  - Azure Storage account
  - Azure Log Analytics
  - Azure Event Hubs
- ✓ Rich set of tools for
  - Investigating security alerts
  - Tracking access to sensitive data

PASS

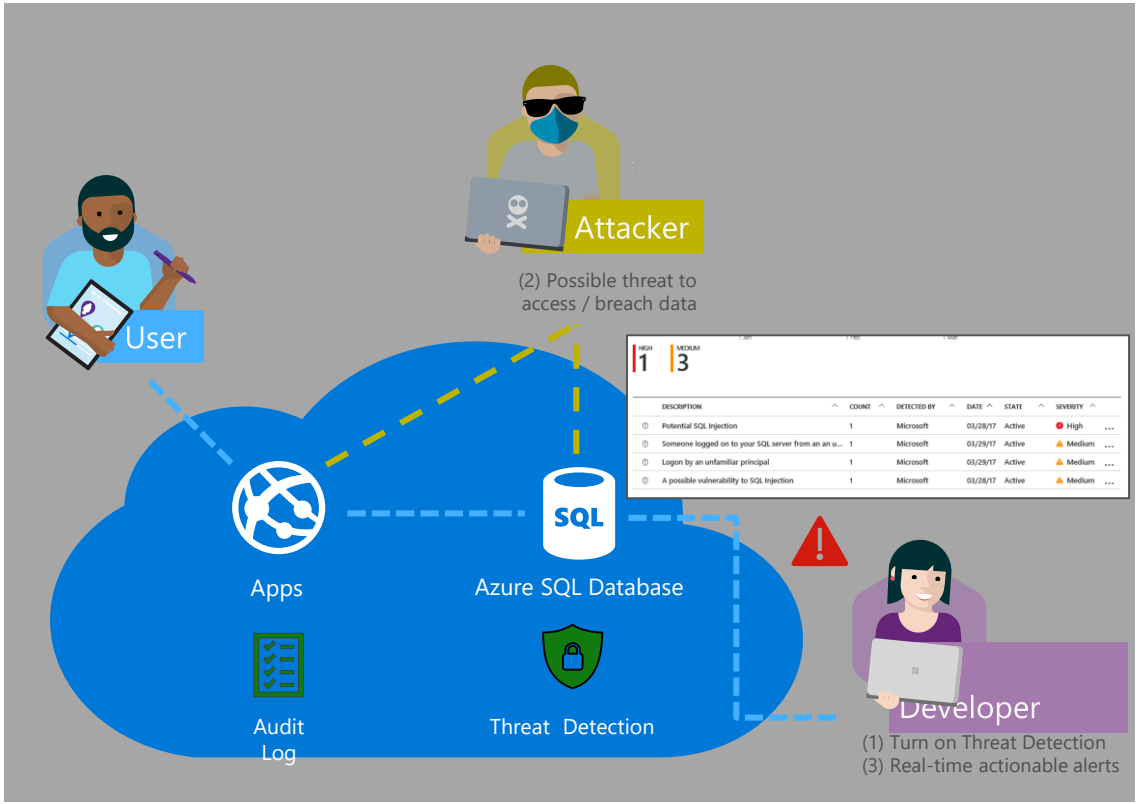
DEMO

Auditing Sensitive Data Access



Advanced Threat Protection

Detect unusual and harmful attempts to breach your database.



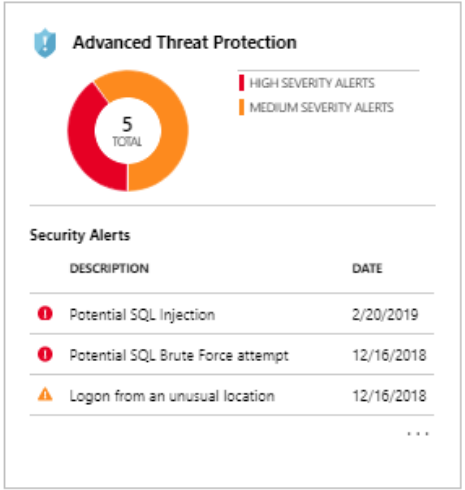
- ✓ Detects potential SQL injection attacks
- ✓ Detects unusual access & data exfiltration activities
- ✓ Actionable alerts to investigate & remediate
- ✓ View alerts for your entire Azure tenant using Azure Security Center

PASS

SQL Threat Protection Suite

Anomalous access patterns

1. Someone has logged from an unusual location - change in the access pattern from an unusual geographical location
2. An unfamiliar principal successfully logged- change in the access pattern using an unusual SQL user.
3. Someone is attempting to brute force SQL credentials abnormally high number of failed logins with different credentials.
4. Someone has logged from a potentially harmful application



Potential SQL injection attacks

5. SQLi attempt (SQL Server 2019) - An application generated a faulty SQL statement, which may indicate a potential vulnerability of the application to SQL injection.
6. SQLi attack (SQL Server 2019) - Potential exploitation of application code vulnerability to SQL Injection, which may indicate a SQL Injection attack.

Anomalous queries patterns

8. Unsecure commands (SQL Server 2019) - Someone has executed unsecure commands (e.g. xp_cmdshell...)
9. Data exfiltration by volume (coming at GA) - someone has extracted anomalous amounts of data in an hour or using a single query

PASS

Advanced Threat Protection

Thank You

Andreas Wolter

@AndreasWolter

andreas.wolter@microsoft.com

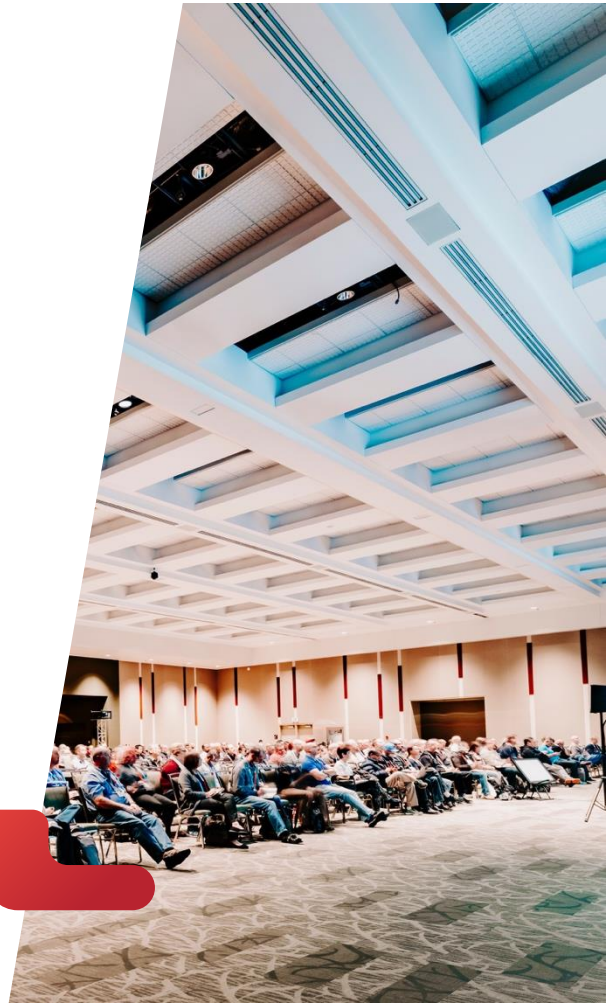
PASS

PASS SUMMIT2019

Azure SQL Database

Best cloud database for DEVs

Davide Mauri
Senior Program Manager

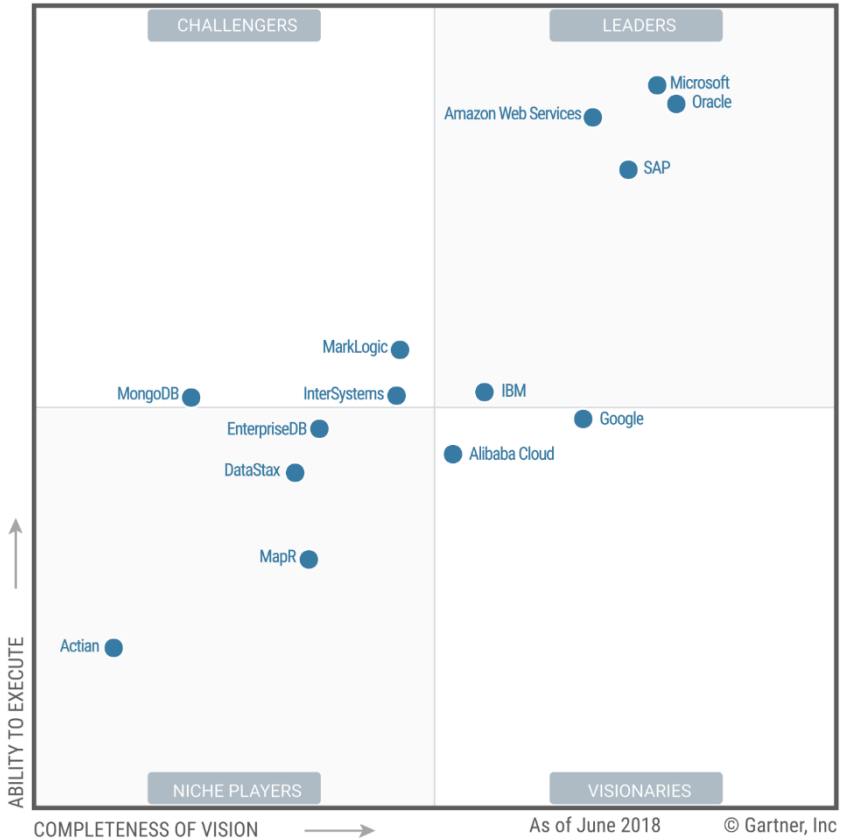


Azure SQL is the best cloud db for devs

Really! Why?

- "Batteries included" philosophy.
  - Yep, that's a cit. from Python ☺
  - Means that you have access to *a lot* of cool features, all in one place
    - Columnstore/Rowstore, Graph, JSON, XML(!), Spatial, Encryption, Analytics and, of course, Relational (with its declarative power)
- Based on the fined tuned SQL Server engine…just cloud-scale
  - Up to 100TB. Several tiers (General Purpose, Business Critical, Hyperscale, Serverless)

Azure SQL is the best cloud db for devs



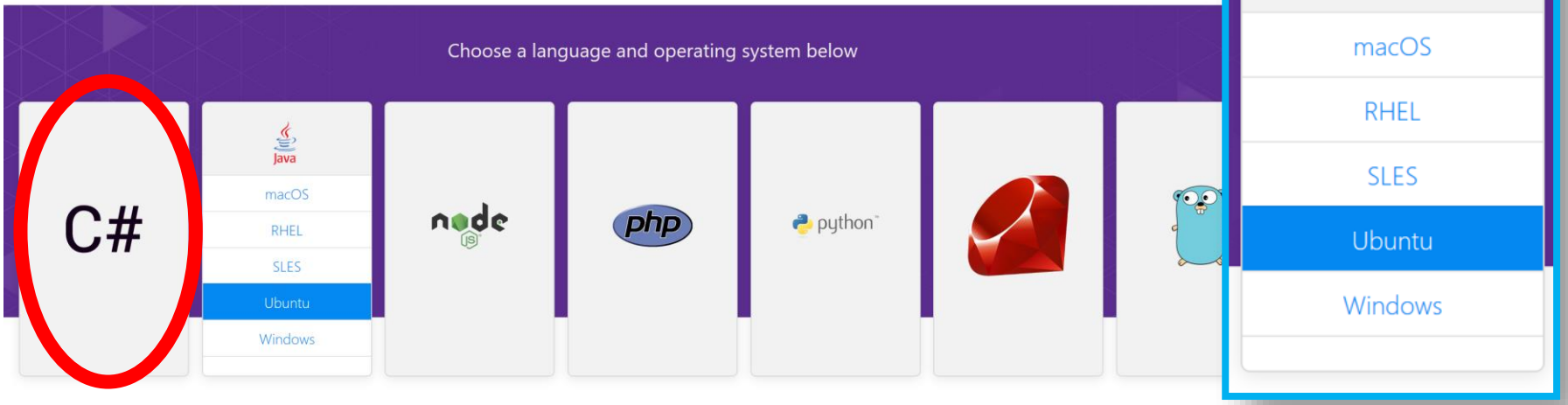
PASS

Choose your own language

Azure SQL is the SQL Server in the cloud

Build an app using SQL Server

Get started with SQL Server on macOS, Linux, and Windows.

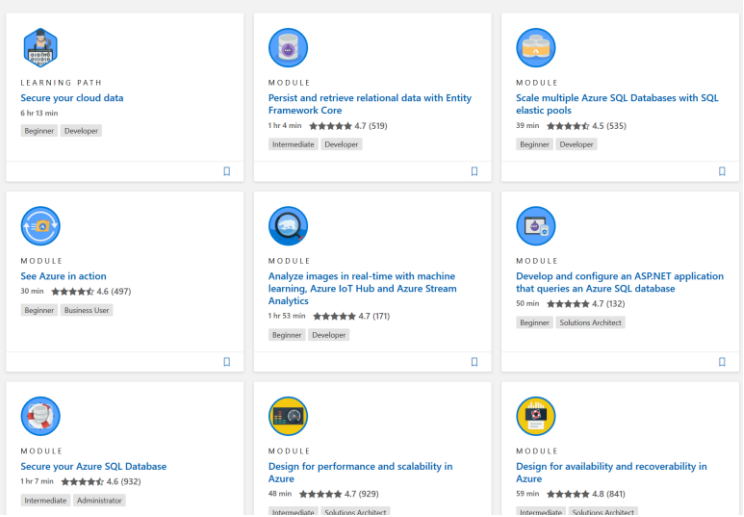


https://www.microsoft.com/en-us/sql-server/developer-get-started

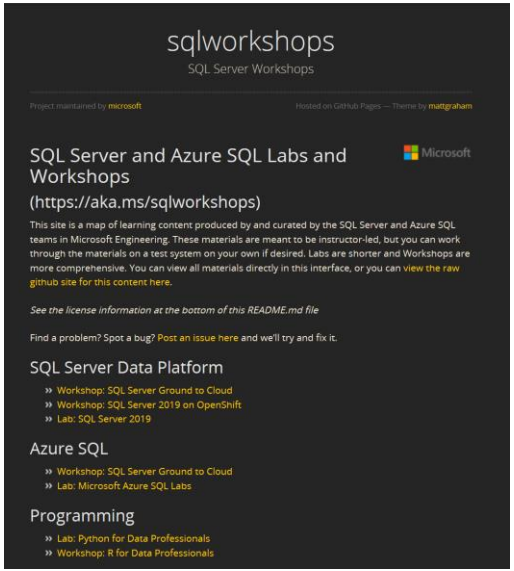
PASS

Learning Paths

Learning for everyone



https://aka.ms/learn-azure-sql

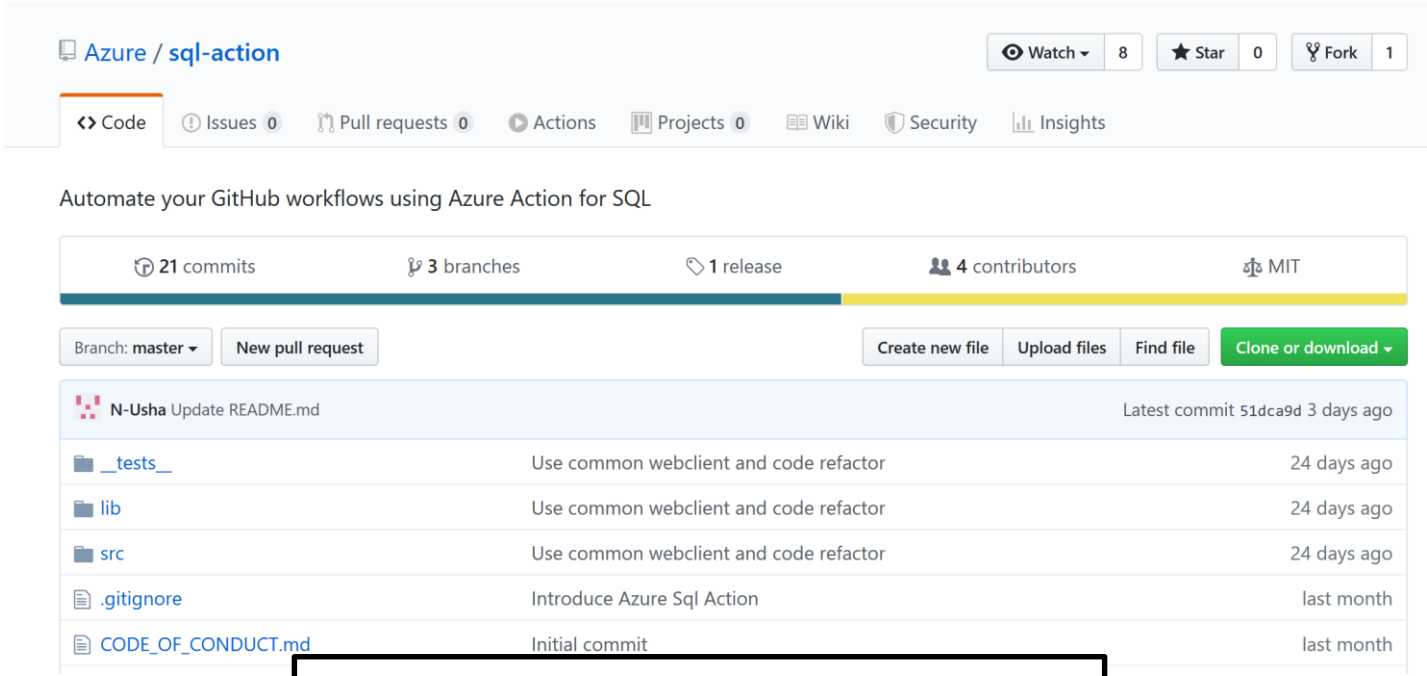


https://aka.ms/sqlworkshops

PASS

CI/CD support in GitHub

GitHub Actions



https://github.com/Azure/sql-action

PASS

Getting Started

Many sample databases ready to be used

- WideWorldImporters
  - .bak => Azure SQL MI
  - .bacpac => Azure SQL Database

- SqlPackage.exe to import .bacpac or Azure Portal/PowerShell/AZ CLI/REST API
  - Azure SQL Database Import

```powershell
$importRequest = New-AzSqlDatabaseImport `
    -ResourceGroupName "dm-pass-2019" `
    -ServerName "dmsqlsrv-pass2019" `
    -DatabaseName "WideWorldImportersFull2" `
    -DatabaseMaxSizeBytes "$(100 * 1024 * 1024 * 1024)" `
    -StorageKeyType "StorageAccessKey" `
    -StorageKey $(Get-AzStorageAccountKey -ResourceGroupName "dm-pass-2019" -StorageAccountName "dmpass2019").Value[0] `
    -StorageUri "https://dmpass2019.blob.core.windows.net/bak/WideWorldImporters-Full.bacpac" `
    -Edition "BusinessCritical" `
    -ServiceObjectiveName "BC_Gen5_2" `
    -AdministratorLogin "<admin-login>" `
    -AdministratorLoginPassword $(ConvertTo-SecureString -String "<admin-password>" -AsPlainText -Force)

Get-AzSqlDatabaseImportExportStatus -OperationStatusLink $importRequest.OperationStatusLink
```

```bash
storageKey=$(az storage account keys list -g dm-pass-2019 -n dmpass2019 --query "[0].value" -o

az sql db import\
    -g dm-pass-2019 \
    -s dmsqlsrv-pass2019 \
    -n $sqlDatabase \
    -u $sqlLogin \
    -p $sqlPassword \
    --storage-key-type StorageAccessKey \
    --storage-uri https://dmpass2019.blob.core.windows.net/bak/WideWorldImporters-Full.bacpac \
    --storage-key $storageKey
```

Connecting to Azure SQL

Microsoft.Data.SqlClient

- The new future-proof library to connect and use Azure SQL
  - https://devblogs.microsoft.com/dotnet/introducing-the-new-microsoftdatasqlclient/
  - Supports both .NET and .NET Core
  - Open Source
    - https://github.com/dotnet/SqlClient
  - Install via NuGet
  - Drop-In replacements for System.Data.SqlClient

create-new-app.bat

```
mkdir myapp

dotnet create console

dotnet add package Microsoft.Data.SqlClient

[ ... ]

dotnet run
```

Program.cs > ...

```
using Microsoft.Data.SqlClient;

namespace Azure.SQLDB.Samples.Connection
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            // Connection string can be found in the portal
            using(var conn = new SqlConnection("<connection string>"))
            {
                conn.Open();
                var cmd = new SqlCommand("SELECT Column1 FROM dbo.Table1", conn);
                var result = cmd.ExecuteScalar();
                conn.Close();
            }
        }
    }
}
```

Using Azure SQL from .NET

Three main options

- SqlClient classes
  - You're in control of everything
- Full-featured ORM like Entity Framework
  - You want the tool to take care of all database interaction for you
- Lightweight ORM like MicroORM
  - You just want to execute queries as fast a possible and map it to a class
    - Dapper, Massive, PetaPoco, Simple.Data, OrmLite

Using Azure SQL from .NET

Dapper Sample

```
public class Dog
{
    public int? Age { get; set; }
    public Guid Id { get; set; }
    public string Name { get; set; }
    public float? Weight { get; set; }

    public int IgnoredProperty { get { return 1; } }
}

var guid = Guid.NewGuid();
var dog = connection.Query<Dog>("select Age = @Age, Id = @Id", new { Age = (int?)null, Id = guid });
```

https://medium.com/dapper-net

Pushing compute to Azure SQL

Use everything you're paying for

- Azure SQL offer a great amount of features to do complex data manipulation
  - And uses a lot of advanced optimization techniques
    - Batch (or vector) execution, using CPU SIMD instructions
    - Intelligent Query Execution (adaptive joins, memory grant feedbacks, etc..)
    - Data and Execution Plan (aka DAG) caching
    - Data pre-fetching
    - Data Distribution Statistics awareness
  - Keep your application as simple as possible and focused on business logic processing (eg: stored procedure)

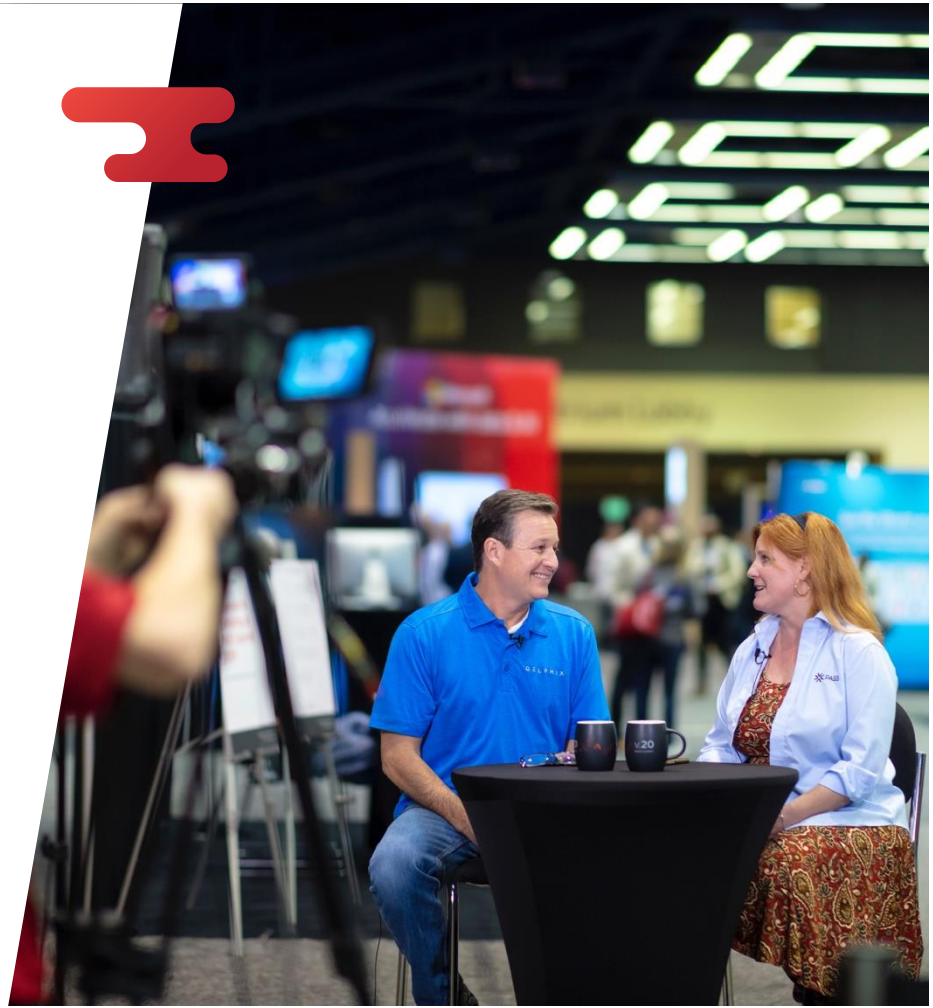
Pushing compute to Azure SQL

Windowing Functions

- Allows complex data manipulation over subset of data (**w**indows)
  - Ranking, Running Totals, Moving Averages, One-vs-Total Ratios,
  - Allows access to previous and next rows in the same (ordered) window
  - Learning how to master them can save *hours or days* of development
    - Simpler applications
    - Less maintenance, better performance scale

DEMO

Windowing Functions



Pushing compute to Azure SQL

String Manipulation

- Historically a weak point of SQL Server / Azure SQL
  - Required data to be sent to the app and back: No more!
    - Less chatty app = less roundtrips = better performances + save money
  - STRING_SPLIT, STRING_AGG, CONCAT, CONCAT_WS, TRANSLATE

Keeping Data Consistent

Transaction

- Azure SQL uses a row-version approach to keep consistency
  - READ COMMITTED SNAPSHOT ISOLATION LEVEL enabled by default
  - Reads will not be blocked by writes!
  - Reads will not block writes!
  - What if you *need* locking instead?
    - Usual Isolation Levels are always available
      - READCOMMITTEDLOCK hint

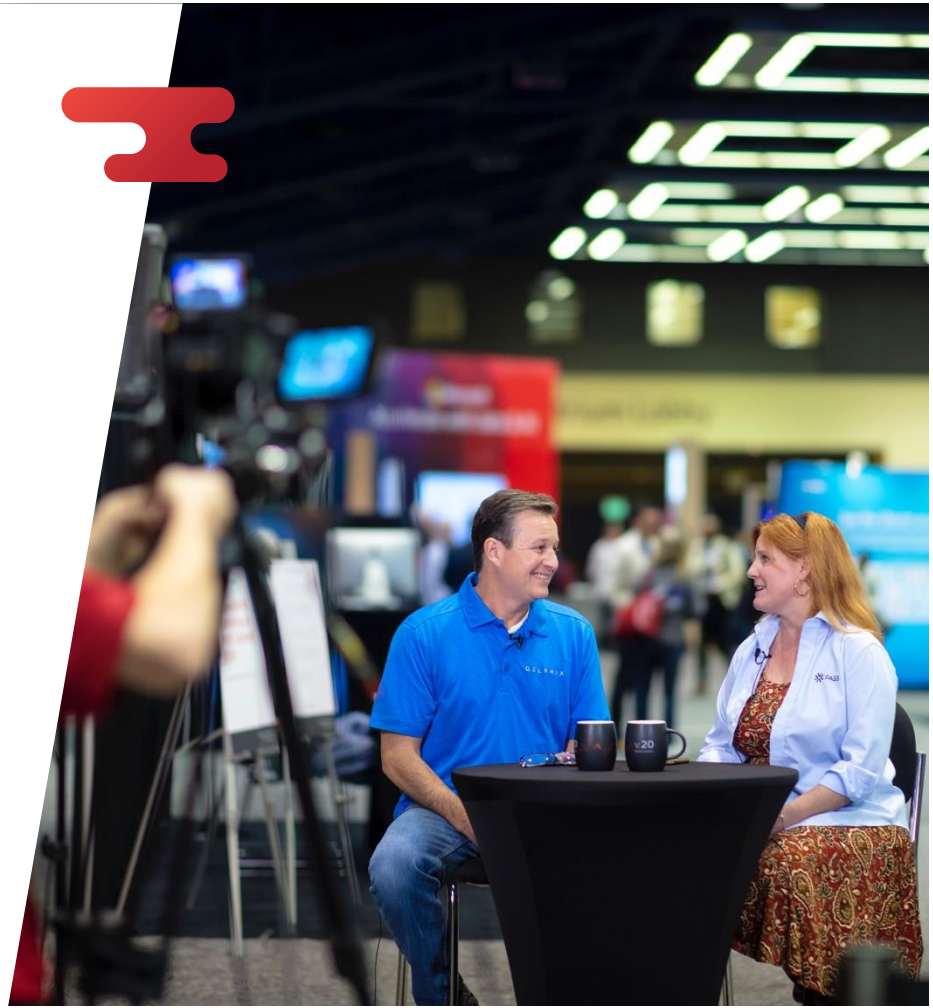
Going Fast

Storage Engine Options

- Azure SQL / SQL Server support both Column-Store and Row-Store
  - Column-Store: ideal for read-intensive app that scans a lot of data
  - Row-Store: ideal for highly transactional app with a lot of small reads and writes
    - Or sequential (range-scan) reads
  - OLTP and HTAP scenarios 100% supported
    - Technically also DWH scenarios supported but Azure DW is a better choice

DEMO

Columnstore



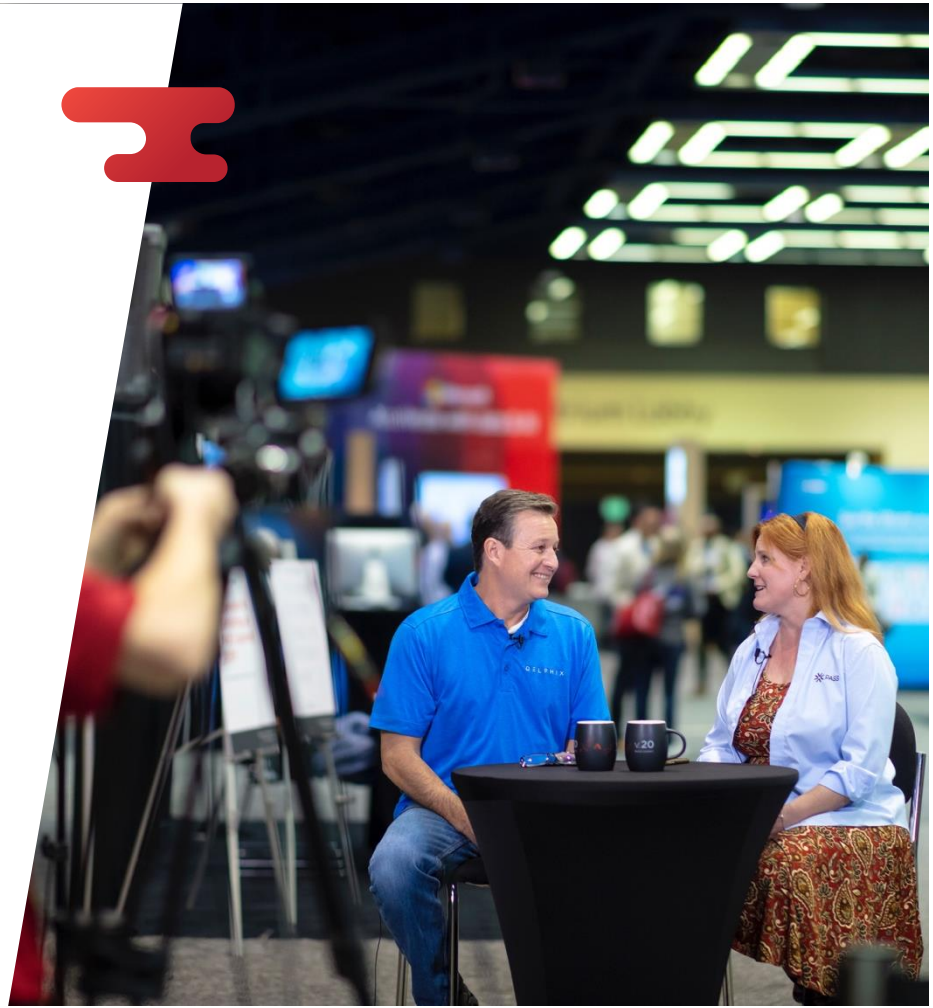
Azure SQL is really NewSQL

Going Beyond Relational

- Full support to a lot of non-strictly relational features
  - Graph database support
    - EDGE, NODE tables, dedicated constructs (MATCH) and functions (SHORTEST_PATH => Transitive Closure)
  - JSON Support
    - FOR JSON, OPENJSON, JSON_VALUE, JSON_MODIFY
  - Spatial Support
    - Geography and Geometry data types (Open Geospatial Consortium)
    - Full Globe, Polygons, Spatial Indexes, WKT/WKB support, Functions

DEMO

Graph Support



A Serverless World

Azure Integration

- Easy to use Azure SQL from other Azure services
  - With Azure Functions it is easy to create complete serverless solutions

- Azure SQL Serverless SKU
  - Automatically scales based on workload demands
  - Automatic Pause-Resume
    - Resume latency: ~1 minute

Can Azure SQL DB Do <>?

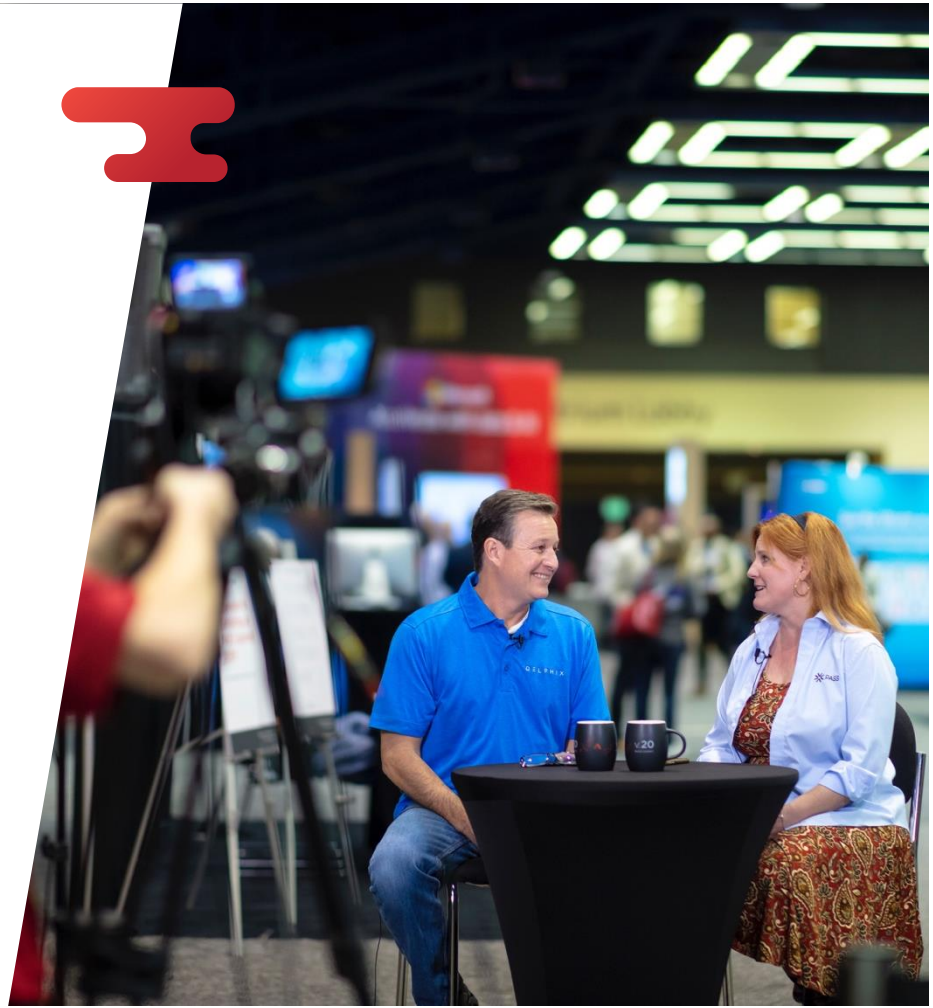
Some myths are still around…

- Can Azure SQL be used to create really, really, scalable solution?
  - Like loading 1 Billion of rows per day
  - No Bulk-load
    - With bulk load you could do even more
  - 10K inserts per seconds?
  - Keep a scoreboard updated in real-time?

https://github.com/Azure-Samples/streaming-at-scale/

DEMO

Can SQL do...?



Can Azure SQL DB Do <>?

Some myths are still around…

- Can Azure SQL be used to create r… solution?
  - Like loading 1 Billion of rows per da…
  - No Bulk-load
    - With bulk load you could do even r…
  - 10K inserts per seconds?
  - Keep a scoreboard updated in real-t…

Azure SQL is the best cloud db for devs

Right?

Thank You

Davide Mauri

@mauridb

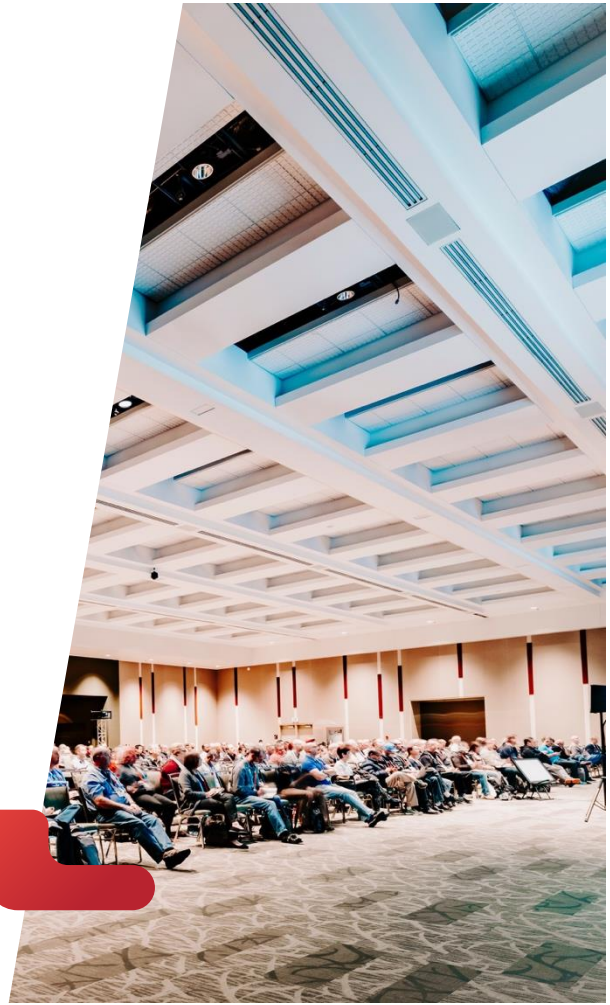
davide.mauri@microsoft.com

PASS SUMMIT2019

Azure SQL Database

Best cloud database for enterprise apps

Davide Mauri, Dimitri Furman
Senior Program Managers
Microsoft



Agenda

1. Robustness and resiliency
2. Scalability and performance
3. Workload monitoring
4. Database design

Best cloud db for enterprise apps

Azure SQL Database is built on SQL Server

SQL Server:

- 40+ years of R&D
- 25+ years of real-world usage and optimizations
  - Fine-tuned to run critical enterprise applications

Azure SQL: leverages all the enterprise experience of the last 25 years

- It is the only cloud database doing that

Enterprise Apps

What is an Enterprise app?

- Enterprise applications are about the display, manipulation, and storage of large amounts of often complex data and the support or automation of business processes with that data
  - Martin Fowler
- Although there is no single, widely accepted list of enterprise software characteristics, they generally include performance, scalability, and robustness.

https://en.wikipedia.org/wiki/Enterprise_software

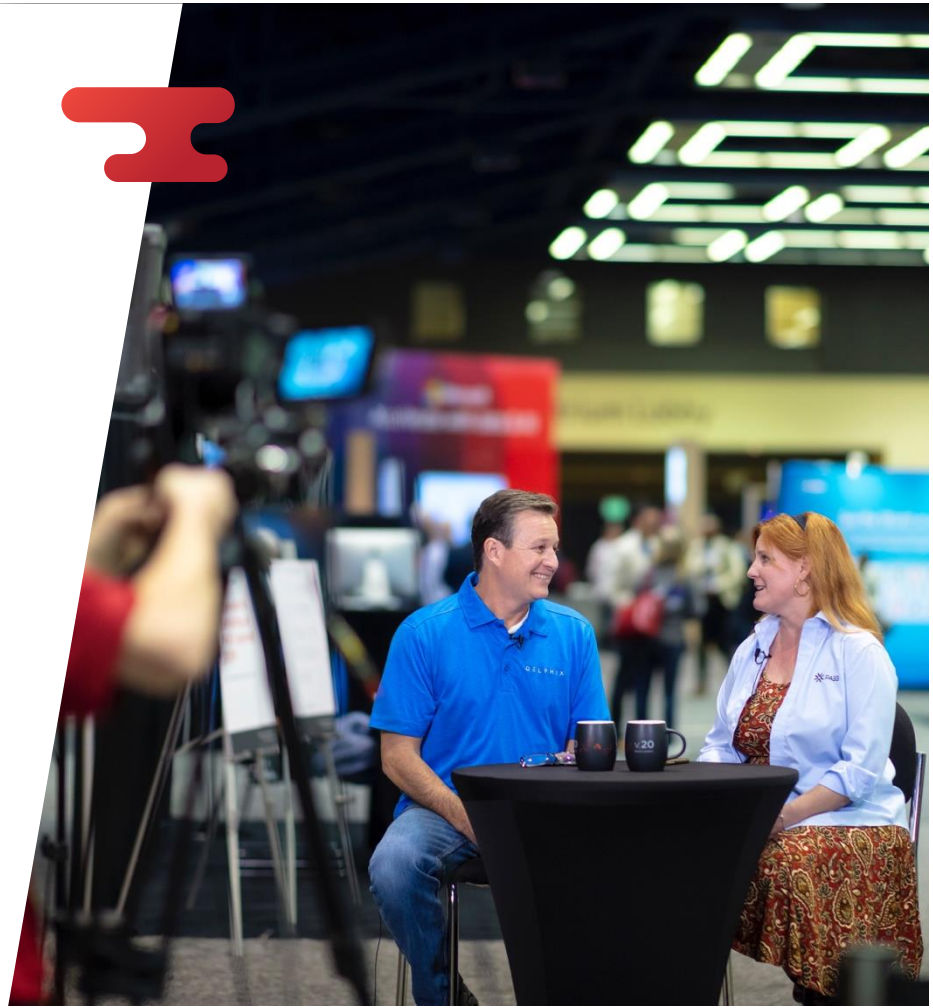
Robustness and Resiliency

In the cloud, be prepared to handle disconnections

- Connection Resiliency
  - Best practice: every mission critical app, right from the start
  - DIY
    - For .NET: Use System.Data.SqlClient >= 4.6.1 or Microsoft.Data.SqlClient
    - Handle Azure SQL DB Connectivity Issues
    - Resiliency in Java Apps
  - Or using existing retry framework
    - Polly

DEMO

Connection Resiliency



Scalability and Performance

Network latency

- Every time your app talks to the database, it is affected by network latency. For chatty apps:
  - Scalability will be harder and more expensive
  - Performance won't be as good as it could be
  - Same best practices can be applied to on-prem
    - (write once deploy everywhere)
- So, Avoid Chatty Apps
  - Push data processing logic to database
  - Avoid sending unnecessary data on the wire (do not use SELECT *)
  - Send sets of data (arrays, lists) and process them as a whole
    - Table Valued Parameters
    - JSON

Scalability and Performance

Storage latency

- Every transaction commit takes 1-10 milliseconds
  - Avoid unnecessarily frequent commits
- Latency depends on storage type

Basic, Standard, General Purpose	remote storage	5-10 milliseconds
Premium, Business Critical	local storage	1-2 milliseconds
Hyperscale	local/remote storage	1-10 milliseconds

- Scaling database up or down
  - Changes IOPS/throughput
  - Does not change latency

Query Monitoring

Know what's happening or what happened

- What's going on *right now*
  - https://github.com/amachanic/sp_whoisactive
- DMV
  - Dynamic Management Views
- Query Store
  - Query performance flight recorder
- Extended Events
  - X-Rays into Azure SQL

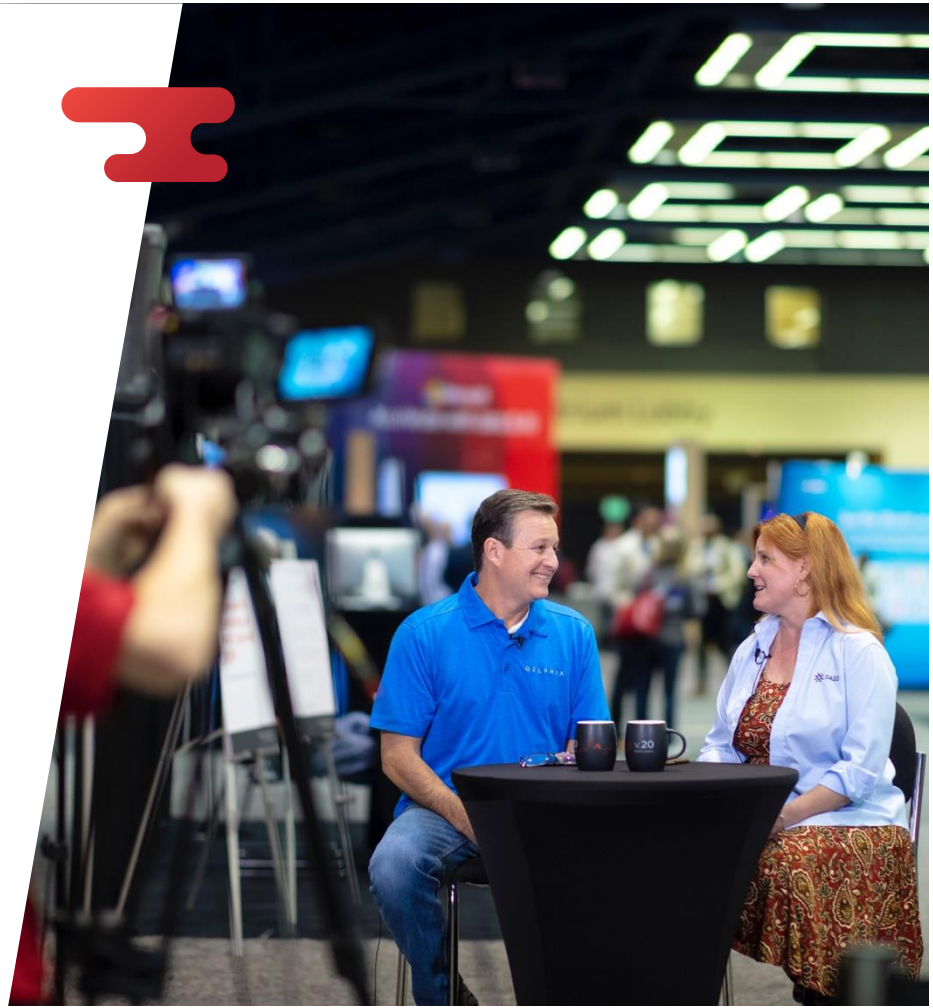
Import external data

What if you need to import CSV or JSON data?

- Azure SQL can access Azure blob storage
  - But *not* files on user local disk or on the network
  - OPENROWSET, BULK IMPORT (including BCP native/character formats)
  - An easy way is to integrate Azure SQL with Azure Functions
    - https://medium.com/@mauridb/automatic-import-of-csv-data-using-azure-functions-and-azure-sql-63e1070963cf

DEMO

Importing JSON/CSV



Database design for enterprise apps

Azure SQL Database vs. SQL Server: What is different

Instance-level features

- Cross-database joins
- SQL Agent
- Linked servers
- Resource Governor
- Database Mail
- …

Authentication

Database size limit

Database options and features

- Recovery model
- Compatibility level
- Files/filegroups
- RCSI
- Auto-tuning
- MAXDOP
- Latest SQL Server features

Database design for enterprise apps

Azure SQL Database vs. SQL Server: What does not change

Database design best practices

SQL Server database engine

Performance tuning methods

- Indexing
- Query rewrite
- Statistics and query plans
- Locking, blocking, deadlocks
- Hints

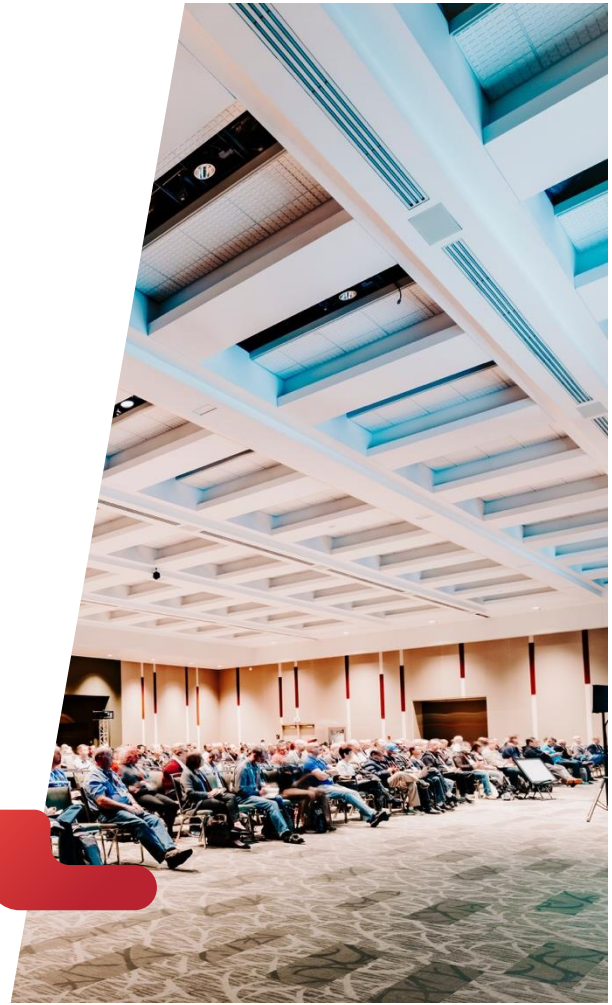
Thank You

Dimitri Furman,
Davide Mauri

PASS SUMMIT2019

Azure SQL Database Hyperscale

Denzil Ribeiro
Principal Program Manager, Microsoft

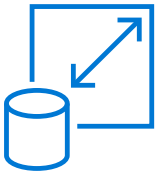


Challenges with Large Databases

Size of data

- Provisioning more storage to expand the database can be painful
- Operations take a LONG time (days in some cases)
- Ongoing operations degrade database performance
- Even in cloud environments, need to shard to fit in a tier

Scaling Compute



- Logistics of moving to larger box
- Economics of sizing for max peaks

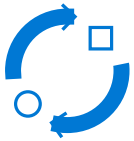
Azure SQL Database Hyperscale?

Cloud native

Architected for cloud

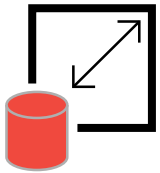
Storage

Scalable new tiered storage architecture



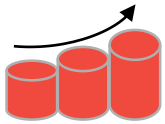
Seamless compatibility

Fully compatible with Azure SQL Database



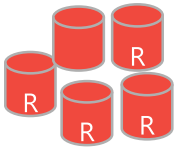
No limits

Scale compute and storage separately
Scale up in constant time.



Large database

Support for 100TB+
Infinite Log



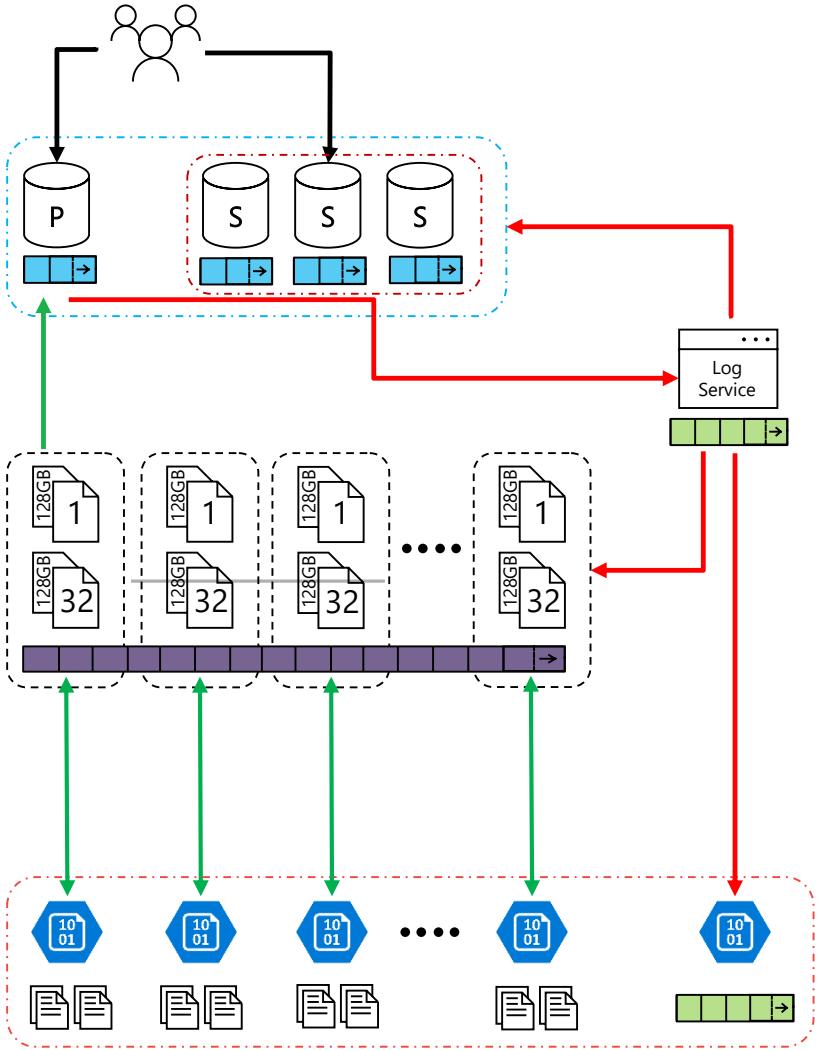
Read scale with replicas

Supports 4 Read replicas

Performance

VLDB operations without headaches

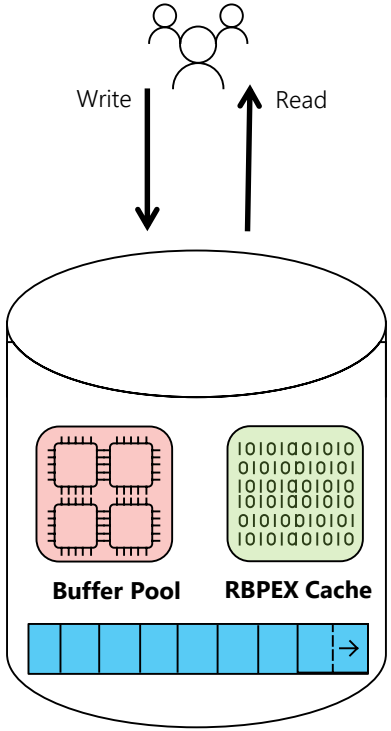
- Restore in constant time
- No impact of backups on compute
- No impact of checkpoints on compute
- Accelerated database recovery



- Features that make this possible:
  - Resilient Buffer Pool Extension (RBPEx)
  - Snapshot-based backups
  - Accelerated Database Recovery (ADR)
- External log service
- Tiered storage
- Completely separated compute and storage
- No single point of failure
- Storage encrypted at rest

PASS

Primary



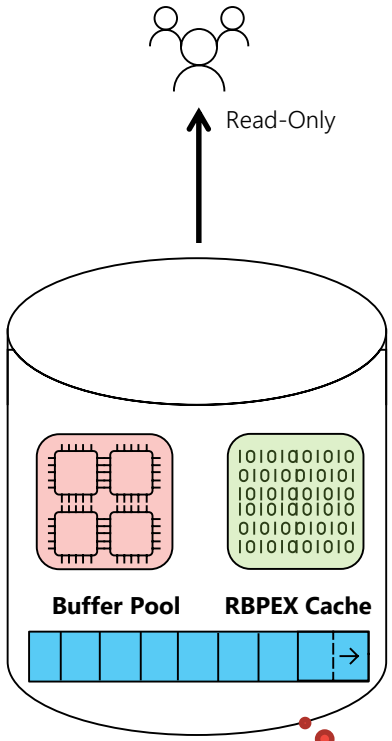
- Configurable vCores 2 – 80
- Memory and directly attached SSD proportional to number of cores
  - Memory dependent on hardware generation
  - **Non-covering** RBPEX
  - 1-2 ms data access latency
- Read-Write Workloads
- Seamless end user interactions

Compute

P

PASS

Readable Secondary

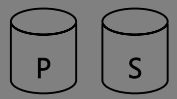


- Currently same size as the Primary
- Up to 4 readable secondary replicas
- Read-only workloads
  - ApplicationIntent = READONLY
  - Round robin distribution
- Hot failover target

Replica priced at AHUB pricing

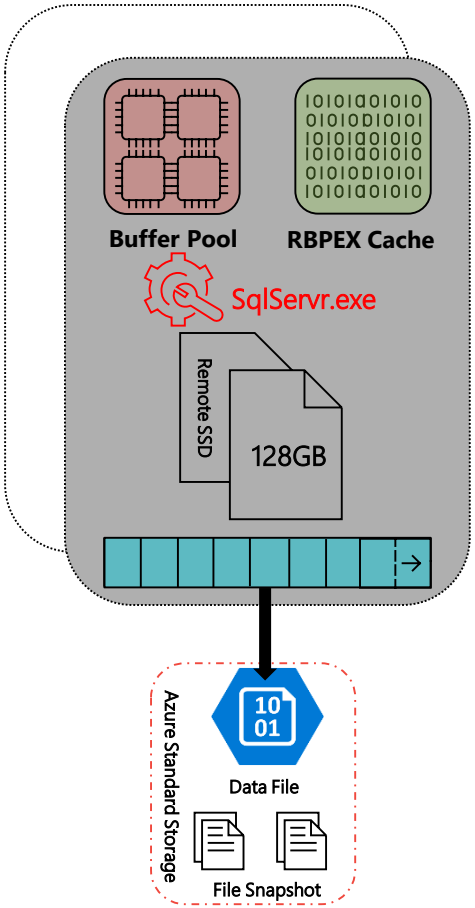
Higher SLA with > 0 replicas

Compute



PASS

Page Server



- "Remote" SSD - 128GB per Page Server
- **Covering** RBPEx
  - >=2ms data access latency
- Each page server has its data file on Azure Standard storage
- Offload operations from compute
  - Checkpoints executed on Page Servers
  - File snapshots for backup operations are isolated from compute
- Built-in high availability for page servers and storage
- Allocations are 10 GB increments

Compute

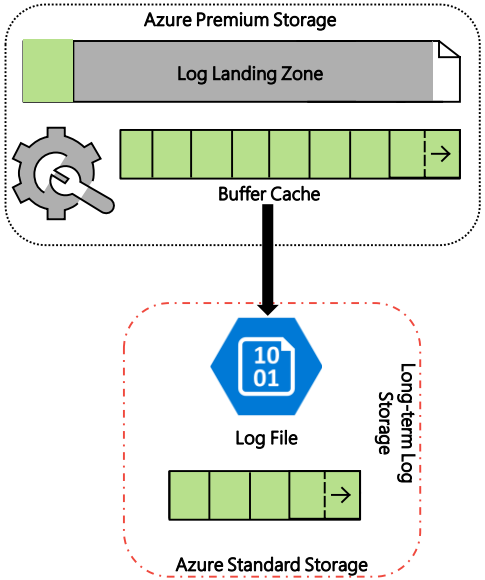
P S

Storage

128GB

PASS

Log Service



- Log Service consists of 3 components
  - Landing Zone
  - The actual Log Service
  - Long-Term Log Storage
- Availability and Resiliency built-in for Storage
- New Transactions are hardened to the Log Landing Zone (Tx Commit)
- Log broker/ Cache will stage Log blocks
  - Apply Log blocks to secondaries / Page servers.
- Data is offloaded to Long-Term log storage

Compute

P S

Storage

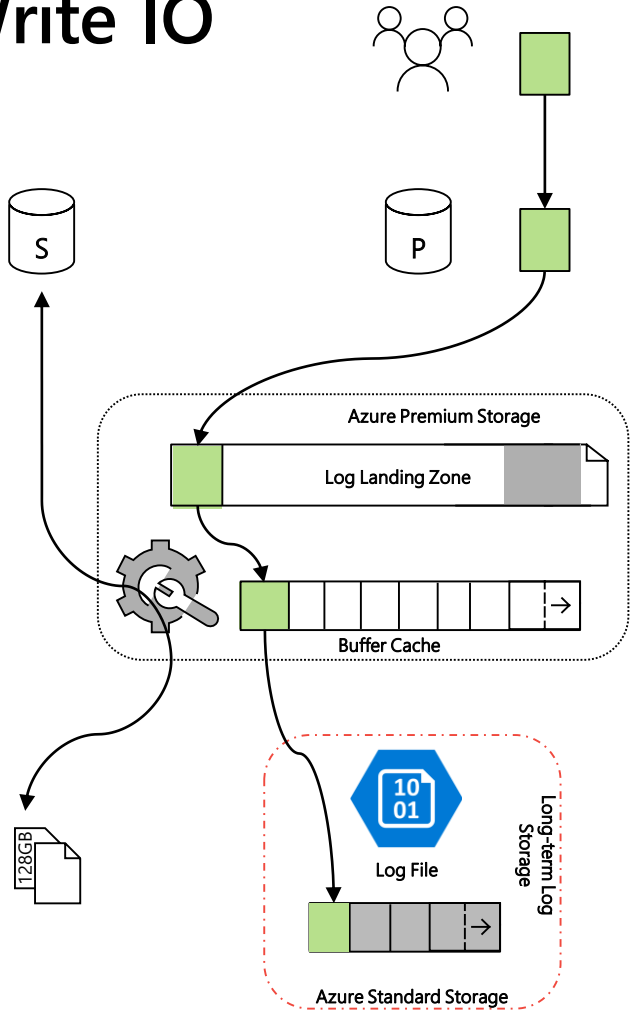
128GB 1

128GB 32

Log Service

PASS

Write IO



- A write Tx is generated by a user or application to the primary
- Transaction is written to the log landing zone
  - **Tx Commit** on hardening of Tx to Log
- Log service stages the new Tx in its cache
- Log service will asynchronously and simultaneously apply the Tx to the secondaries and the page servers
- Tx Log is destaged from the buffer cache and shipped to the long-term log storage

Compute

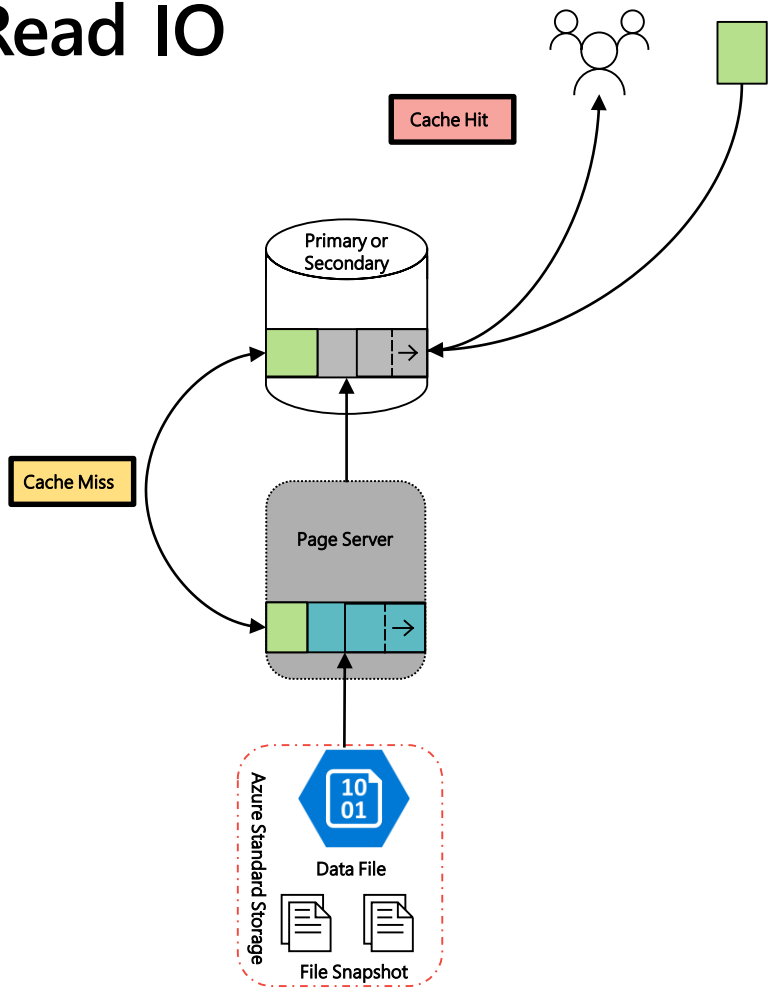
P S

Storage

128GB 1
128GB 32
Log Service

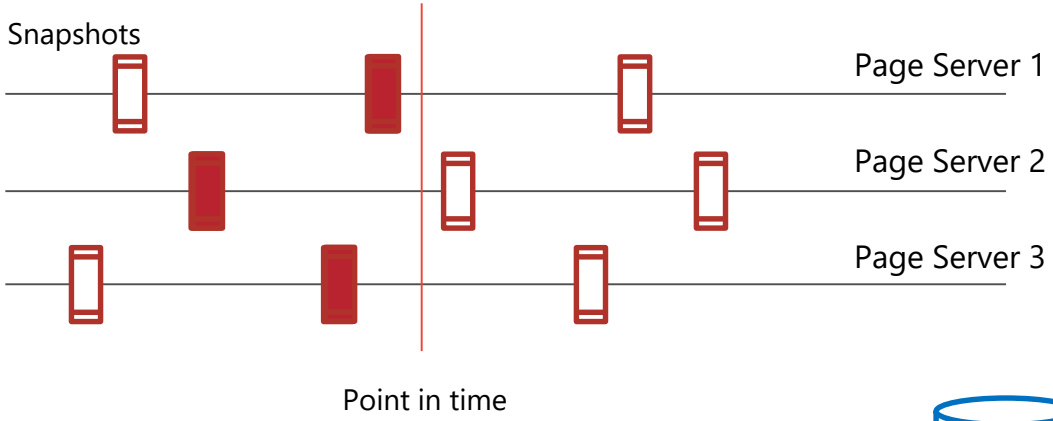
PASS

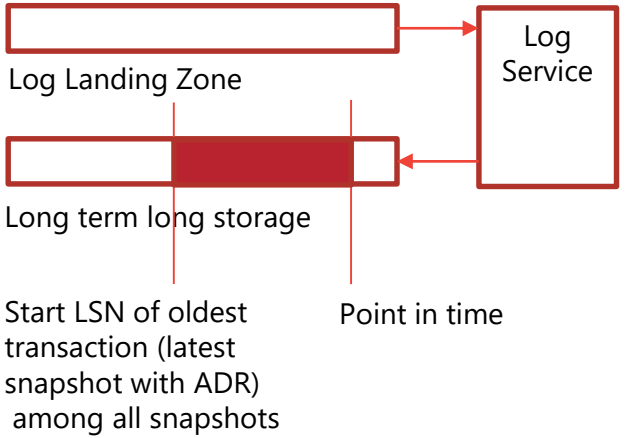
Read IO



- From a Cold Start – Data Files stored on Azure Standard Storage
- Data files will populate the covering RBPEx on the Page Servers
- Page servers will populate the non-covering RBPEx on the Primary and Secondary compute
- Query to return data pages is issued by a user or an application
- Cache Hit – Data is in the local RBPEx 1-2ms latency
- Cache Miss – Data has not on local RBPEX, to be retrieved from Page Server > 2ms data latency

Backup & Restore



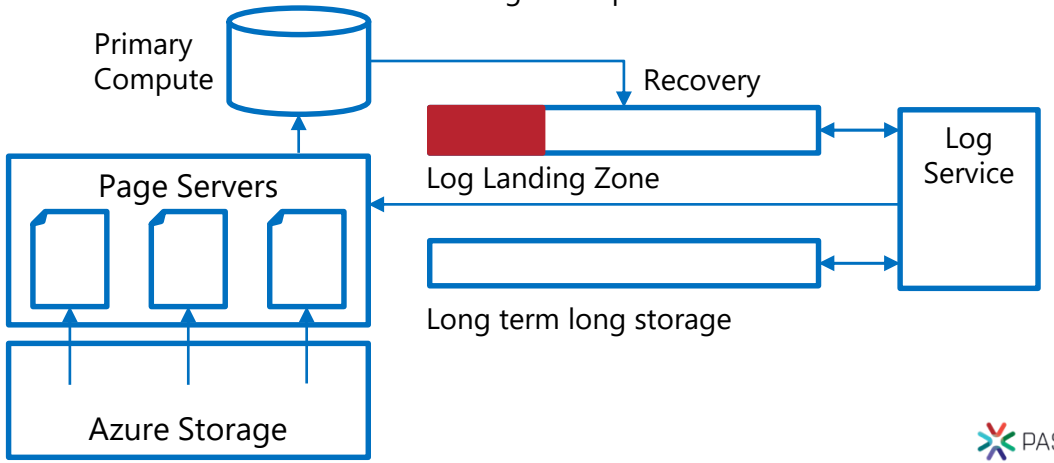


Backup has no impact to apps

Parallelized copy (metadata)

Constant time PITR

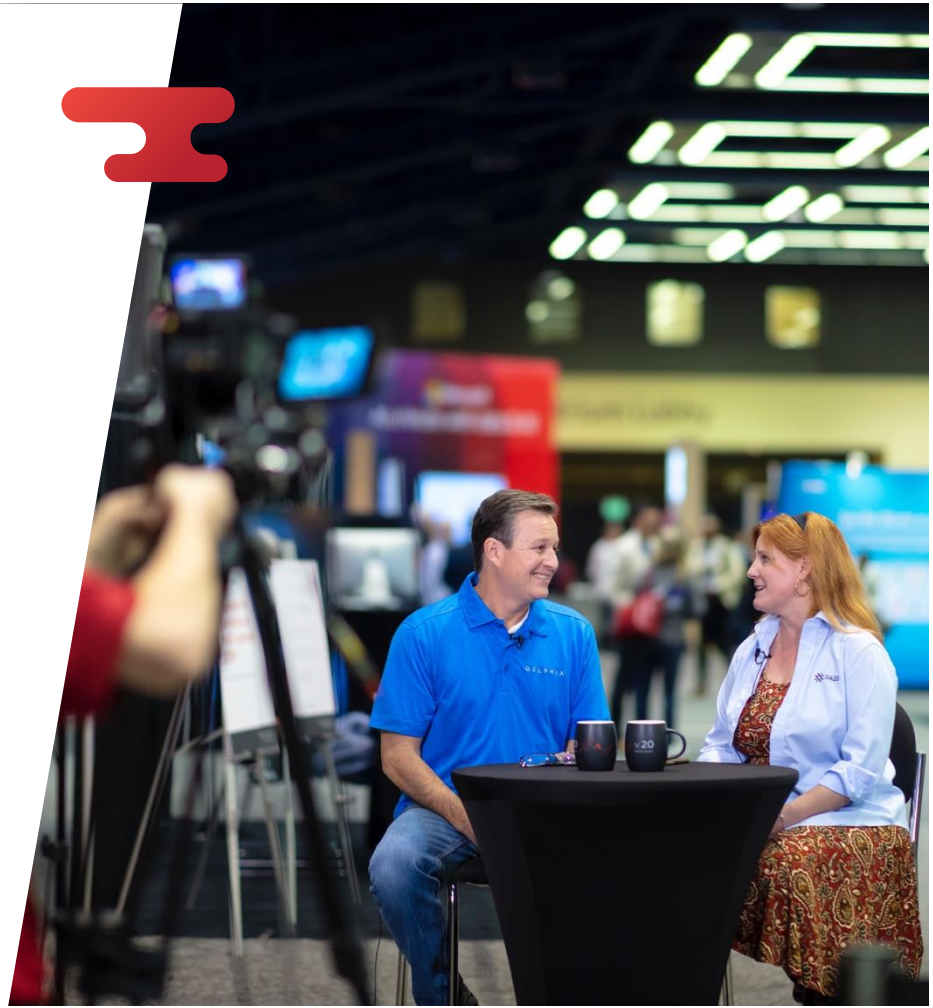
Accelerated database recovery



PASS

DEMO

Read Scale



Hyperscale benefits

Performance

- Better performance than Standard/General Purpose overall, for **all** databases
- Offloading backups/checkpoints to page servers
- Rapid scale up/down
- Read scale out with Primary + 4 replicas
- Snapshot based restores (fast for any database size)
- Workloads requiring very low latency commit may work better on Premium/Business Critical today

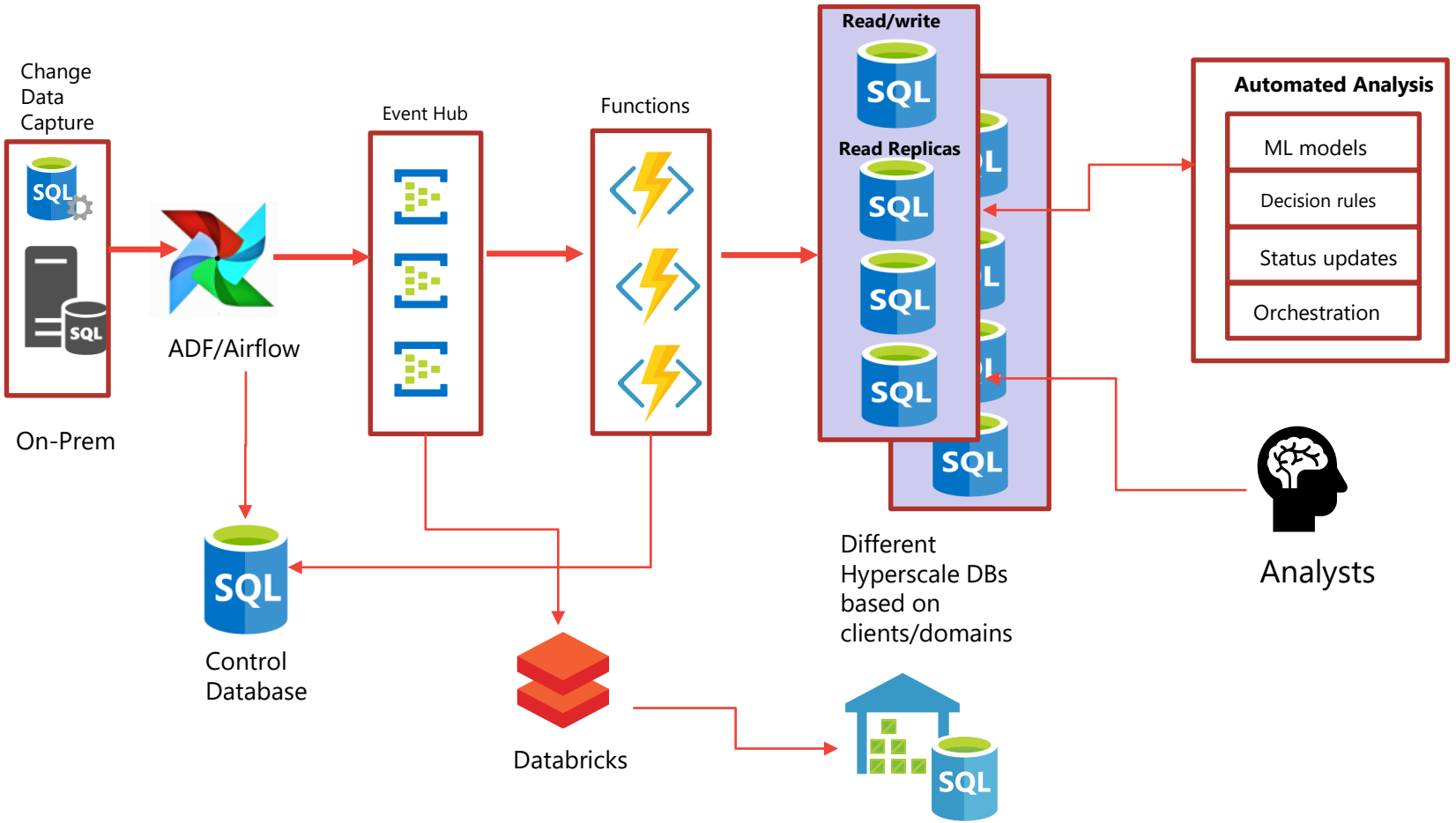
Size

- Large databases up to 100 TB
- Auto-scaling size
- Storage decoupled from compute

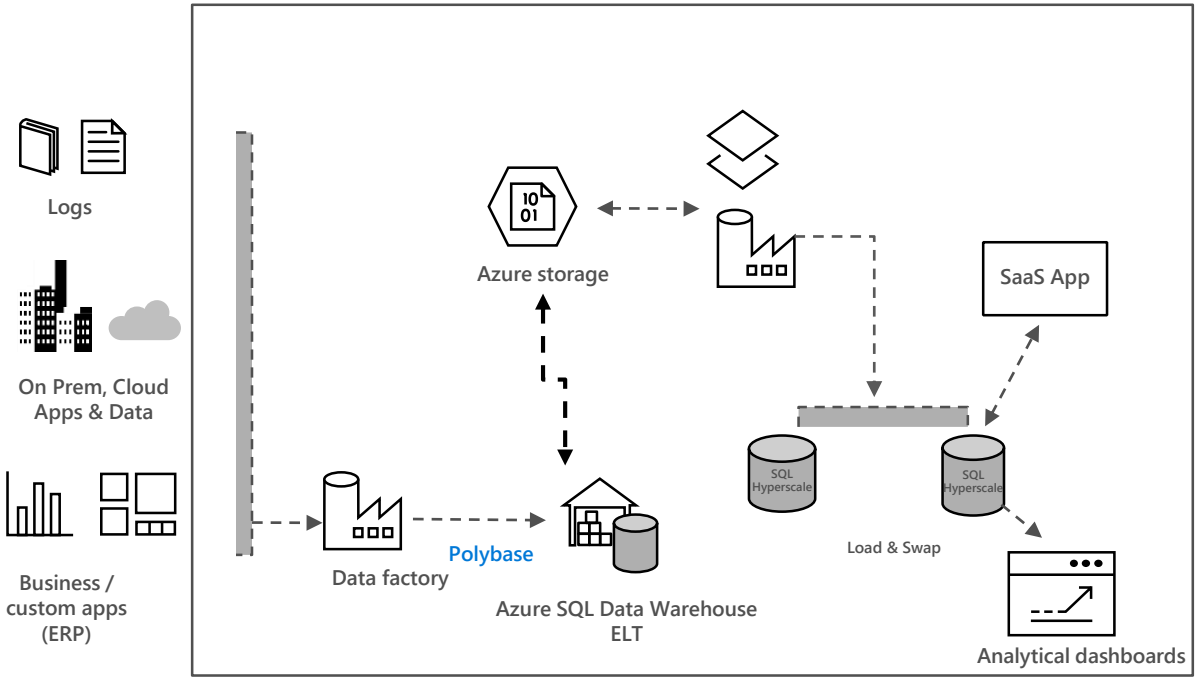
Workload Patterns

- OLTP
- HTAP
- Operational data stores/marts
- IOT
- PRS migrations

Scenario: Near-Realtime Fraud Detection



Scenario: Hyperscale for Data Marts



Several customers use Hyperscale as ODS/data mart for reporting.

Patterns:

- Load and swap (2 separate HS database), allows different tiers on each.
- Batch/ETL loads with ADF or Spark.
- Database sizes in 5-20 TB range currently

PASS

Migration – Moving Data to Hyperscale

Azure Data Migration Service

Bulk load/Bulk copy

- Using Azure Data Factory
- Using Spark connector for SQL
- Using Sqlbulkcopy Sample: smart bulkcopy
- Bulk load into heaps for larger tables, create indexes later
- If using clustered Columnstore, bulk load directly into the table with CCI

Transactional replication

- Use concurrent snapshots
- Requires primary keys

Migration from existing Azure SQL DB

Migration from another service tier is just a service objective change
Currently cannot go back to prior tier

Existing DBs moved to Hyperscale
- Existing files > 128 GB, land on 1 TB page servers
- Existing files <=128 GB land on 128 GB page servers (more efficient)

If performance is an issue in testing
Option 1:
Create new Hyperscale DB, Load data.

Option 2:
- Add files at the source, Rebuild indexes to redistribute data, shrink
- Once files are < 128GB each then SLO update

Hyperscale Perf diagnostics

Log throttling has new set of waits (RBIO_RG*)
Page Server reads (Remote reads) reported in all the DMVs below

- sys.dm_exec_requests
- sys.dm_exec_query_stats
- sys.dm_exec_procedure_stats

Page Server reads added to Actual Execution Plan to distinguish remote reads.

RPBEX IO exposed in DMVS

- FileID 0 in sys.dm_io_virtual_file_stats() accounts for RBPEX IO.
- All RBPEX IO is 8KB

Additional details: Hyperscale performance diagnostics

Index Creation/Rebuild

- Tempdb is still a finite resource, size depends on compute size

- Consider resumable indexes for index creation/rebuild on very large tables
  - Persists index build state which is replicated, can be resumed.
  - CREATE INDEX is also resumable
  - New database scoped options:

    ELEVATE_RESUMABLE= { OFF | WHEN_SUPPORTED | FAIL_UNSUPPORTED }
    ELEVATE_ONLINE = { OFF | WHEN_SUPPORTED | FAIL_UNSUPPORTED }

- Consider MAXDOP hint for index create/rebuild
  - Particularly if doing concurrent index create/rebuild
- Consider partition level index rebuilds

*Offline indexing is faster than online/resumable

Thank You

Denzil Ribeiro

@denzilribeiro

denzilr@microsoft.com

PASS

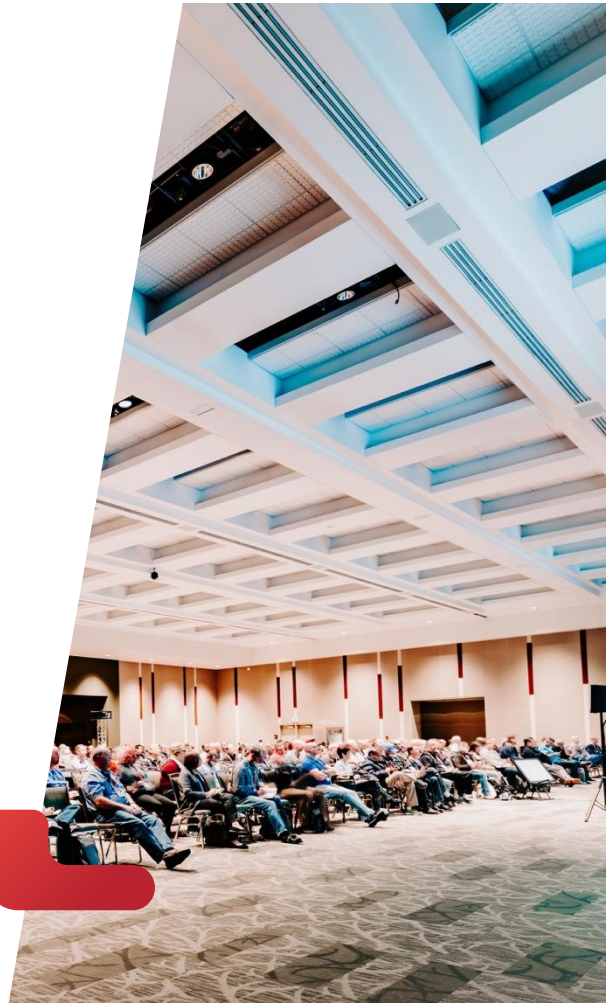
PASS SUMMIT2019

Azure SQL for DBAs

How it Works, and How to Solve Common Problems

Dimitri Furman, Denzil Ribeiro

Microsoft



Agenda

1. DBA tasks in the Azure SQL world
2. Solving common performance and scale problems

Traditional DBA Tasks

Platform DBA

- High availability
- Perf tuning (resources)
- Backup/restore
- Business continuity/DR
- Monitoring
- Infrastructure design
- Installation, upgrades, patching

Application DBA

- Database design and development
- Code deployment
- Perf tuning (query)
- Data lifecycle management
- Data security

How does this change in Azure SQL Database?

High Availability

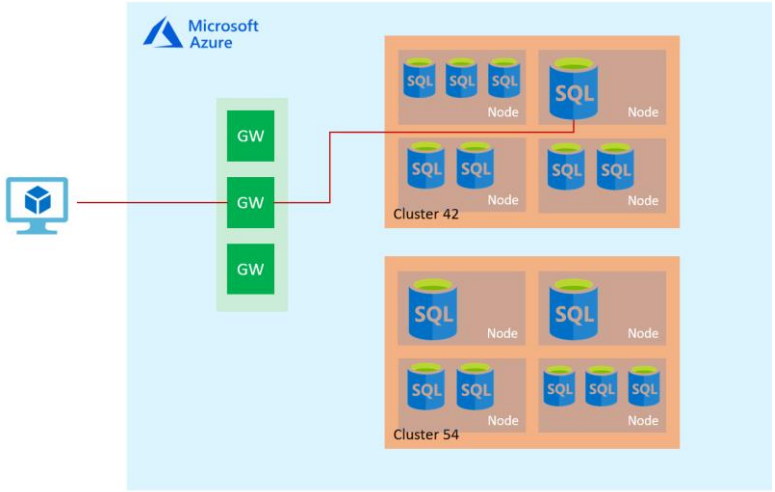
Built-in, for any Azure SQL database

- Understand HA capabilities
- Which vary by service tier
  - Basic, Standard, General Purpose
  - Premium, Business Critical
  - Hyperscale

High Availability

How it works: all service tiers

- Highly redundant connectivity layer, multiple gateways
- Large pools of compute capacity (tenant rings), hundreds of nodes
- Managed by Service Fabric



High Availability

How it works: Basic, Standard, General Purpose

- Single compute replica (VM) running SQL Server
- Data is remote in Azure Storage
  - 99.999999999% durability guarantee over a year
- Standby VMs replace failed replicas
- Fast compute scaling in minutes, regardless of database size
  - Storage is re-attached to new replica
  - Failover takes up to a minute

High Availability

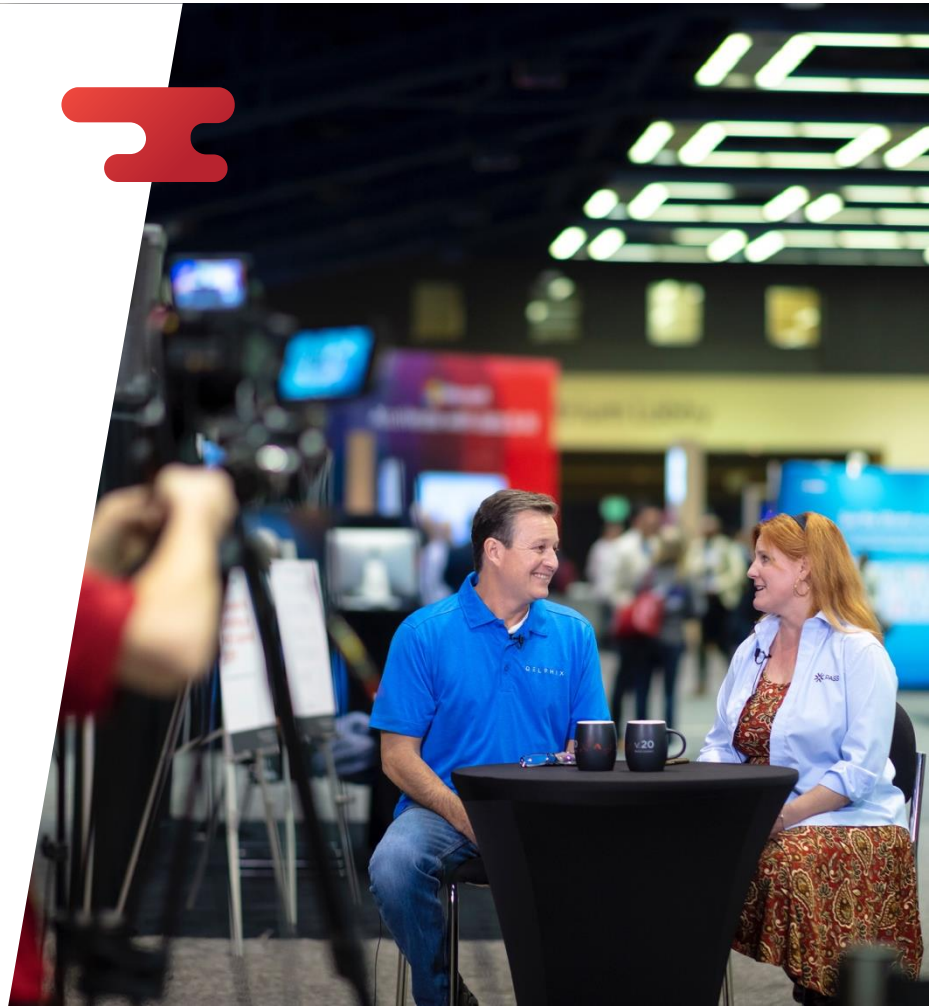
How it works: Premium, Business Critical

- Multiple compute replicas, typically 4
- Availability Groups with quorum commit
- Data is local on SSD storage
- Failed replicas are replaced from compute pool and reseeded
- Time to scale compute is proportional to database size
  - New replicas are built, data is copied at 1 GB/minute
  - Failover to a sync commit secondary takes <8 seconds on average
- Zone-redundant databases (optional)

DEMO

Manual failovers in Azure SQL Database

Invoke-AzSqlDatabaseFailover
Invoke-AzSqlElasticPoolFailover



Resource Management

Current hardware generations in Azure SQL

Gen5

- Intel Broadwell CPU (E5-2673 v4)
- 5.1 GB/vCore memory ratio
- Up to 80 vCores (hyperthreaded)
- NVMe SSD
- Up to 4 TB of local storage
- Accelerated Networking is guaranteed

Gen4

- Intel Haswell CPU (E5-2673 v3)
- 7 GB/vCore memory ratio
- Up to 24 vCores (physical)
- SAS SSD
- Up to 1 TB of local storage
- Accelerated Networking is not guaranteed

New hardware generations in preview

M128m

- 128 vCores (hyperthreaded)
- 3.8 TB memory
- Business Critical

Fsv2

- 72 vCores (hyperthreaded)
- 3.4 GHz sustained turbo clock speed
- General Purpose

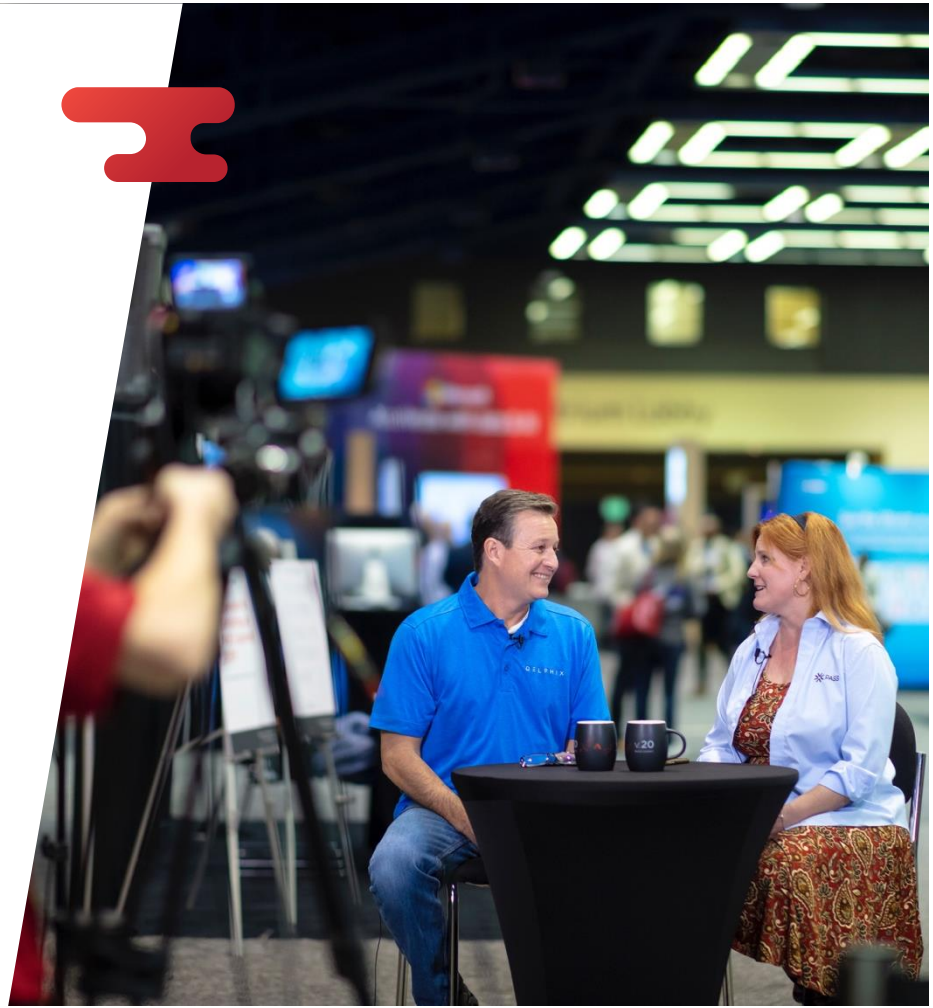
Resource Management

System resource governance in Azure SQL

- Database service objective: a set of resource limits
  - CPU, memory, storage size, IOPS, throughput, sessions, worker threads, etc.
  - Single database limits and Elastic pool limits
- Provides balanced Database-as-a-Service
  - Backup, HA, DR, predictable performance, multi-tenant fairness, etc.
- Investigating performance? Check resource usage first

DEMO

Troubleshooting “Request limit reached”



Backup and Restore

Built-in, for any Azure SQL database

- Automatic full, differential, and log backups
- Configurable backup retention period, up to 35 days*
- Optional long-term backup retention (LTR), up to 10 years*
- Automated point-in-time restore
  - Restore time depends on service objective
- Fast backup and restore for VLDBs in Hyperscale

* Temporary limitations for Hyperscale
- Fixed 7-day retention only
- LTR is not yet supported

Business Continuity/DR

Built-in, for any Azure SQL database

- Geo-Restore
  - Backups copied to paired Azure region
  - Can be restored in any region

Opt-in, for any Azure SQL database

- Geo-Replication
  - Database replicas in any Azure region
- Failover Groups
  - Geo-replication for a group of databases
  - Connection strings not affected by failovers

Monitoring and alerting

Built-in, for any Azure SQL database

- Resource metrics in Azure Monitor
- Query Store
  - Automatic tuning
  - Query performance insights
- Resource health
- DMVs

sys.dm_db_resource_stats
sys.resource_stats (in master)
sys.dm_db_wait_stats
sys.dm_os_wait_stats
sys.dm_exec_session_wait_stats
sys.dm_os_performance_counters
sys.dm_os_job_object
sys.dm_user_db_resource_governance
sys.dm_resource_governor_resource_pools
sys.dm_resource_governor_resource_pools_history_ex
sys.dm_resource_governor_workload_groups
sys.dm_resource_governor_workload_groups_history_ex

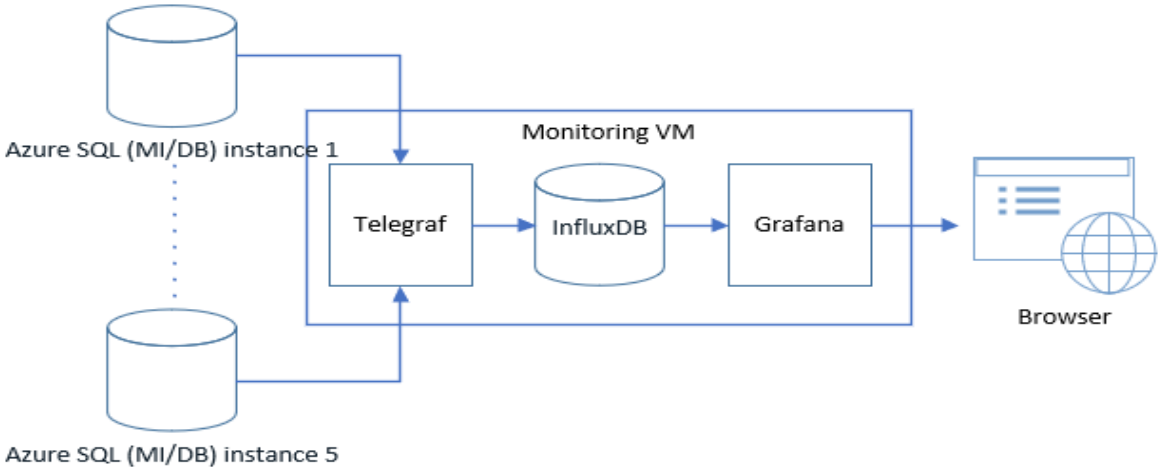
Monitoring and alerting

Opt-in, for any Azure SQL database

- Azure SQL Analytics, Intelligent Insights
  - Early detection, root cause analysis
  - Monitoring at scale
- Extended Events for in-depth troubleshooting
- Alert rules in Azure Monitor
  - Define an alert for multiple databases/managed instances
  - Alert on resource metrics exceeding thresholds
  - Alert on management activity events

Additional monitoring options

- Third party tools
- Query performance insight library
- Near-realtime monitoring for Azure SQL with telegraf/InfluxDB/Grafana
  - Ability to monitor Managed Instance, Azure SQL DB, SQL Server



Solving performance and scale problems

Data loading in Azure SQL Database

All service tiers have necessary resource limits

Cost/performance and workload specific decisions

Hyperscale: same log generation rate in any service objective

Full recovery model!

- On Heaps still use TABLOCK (#locks acquired)
- On Columnstore tables use batch size > 102,400

Partitioning very large tables is important

- Provides flexibility on index rebuilds, incremental stats

Business Critical has HADR replicas

Log rate governance

Why throttle log generation?

- Required for replicas to be up to date
- Allows frequent log backups to sustain recoverability SLAs

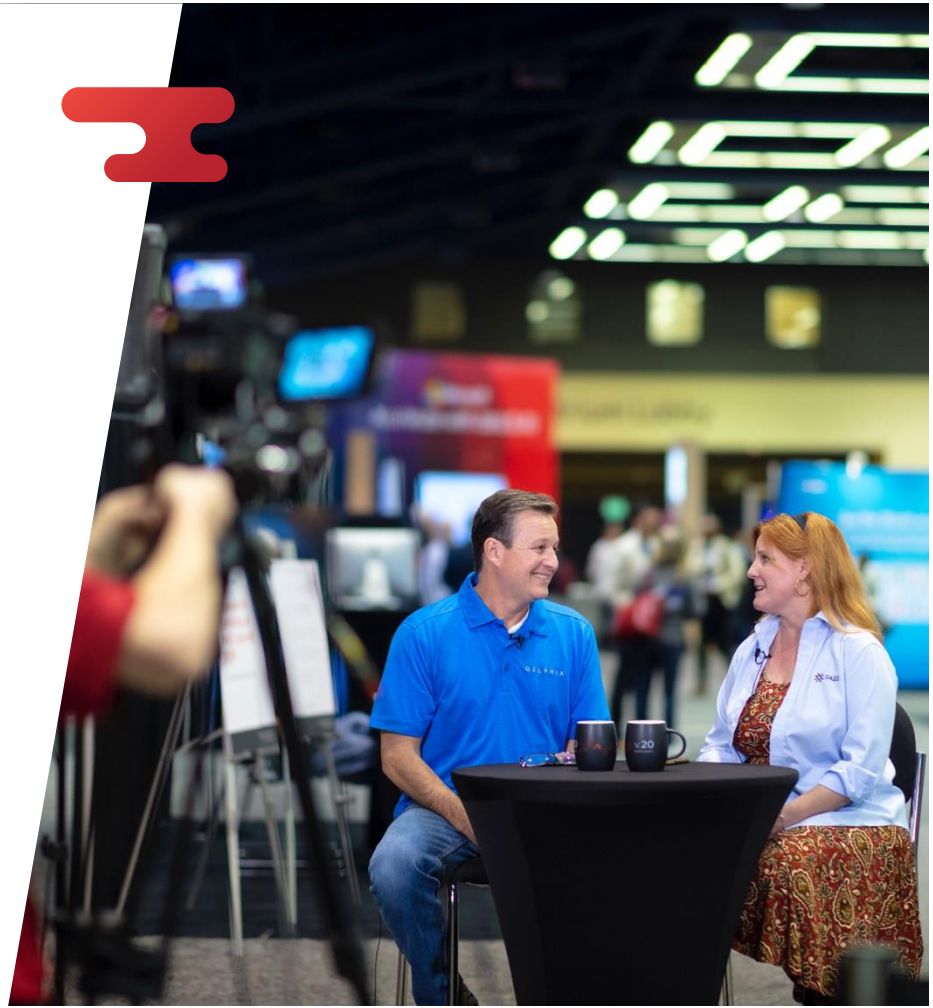
Applied to log generation itself to avoid storage throttling

Surfaced as waits

Wait Type	Notes
LOG_RATE_GOVERNOR	Database limiting
POOL_LOG_RATE_GOVERNOR	Pool limiting
INSTANCE_LOG_RATE_GOVERNOR	Instance level limiting
HADR_THROTTLE_LOG_RATE_SEND_RECV_QUEUE_SIZE	Feedback control, availability group physical replication in Premium/Business Critical not keeping up
HADR_THROTTLE_LOG_RATE_LOG_SIZE	Feedback control, limiting rates to avoid an out of log space condition

DEMO

Data loading in Azure SQL DB



Other data loading tips

Load data in parallel improves throughput

- Reduce concurrent parallel loads on columnstore due to memory grants

Loading from a compute (ADF/Spark) offloads the “read” of the files to a compute engine rather than SQL DB

If loading by partition

- Load into staging table
- Build aligned indexes
- Switch into partitioned table

When comparing to a baseline, compare with Full recovery model

Load into Columnstore when feasible (> 102400 batch size)

For very large data warehouse loads (>150 MB/sec), Azure SQL DW would be a better choice.

Storage

Local storage

General Purpose: used for tempdb only

Business Critical: used for *all* database files

Remote storage

Database files are page blobs in Azure Premium Storage

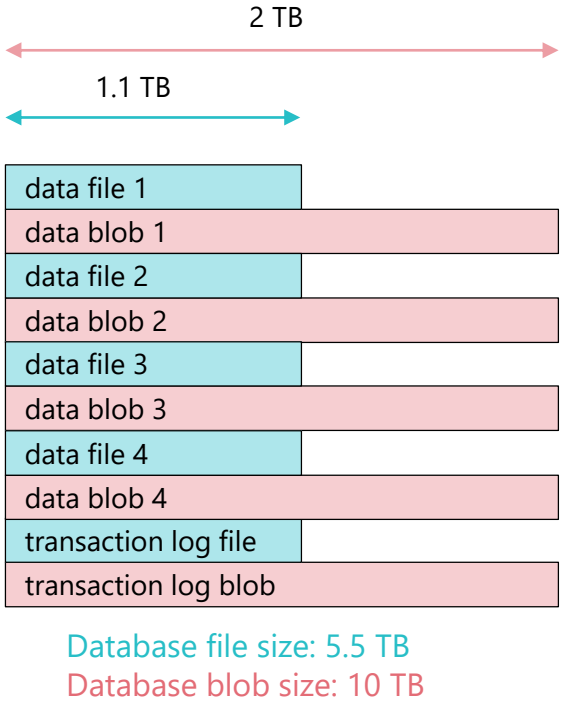
IOPS/throughput is tied to blob size

5-10 ms latency, unless throttled

Blob size >= database file size

Managed Instance storage limit is based on file size, not blob size

Premium storage limits



DEMO

Maximizing storage perf

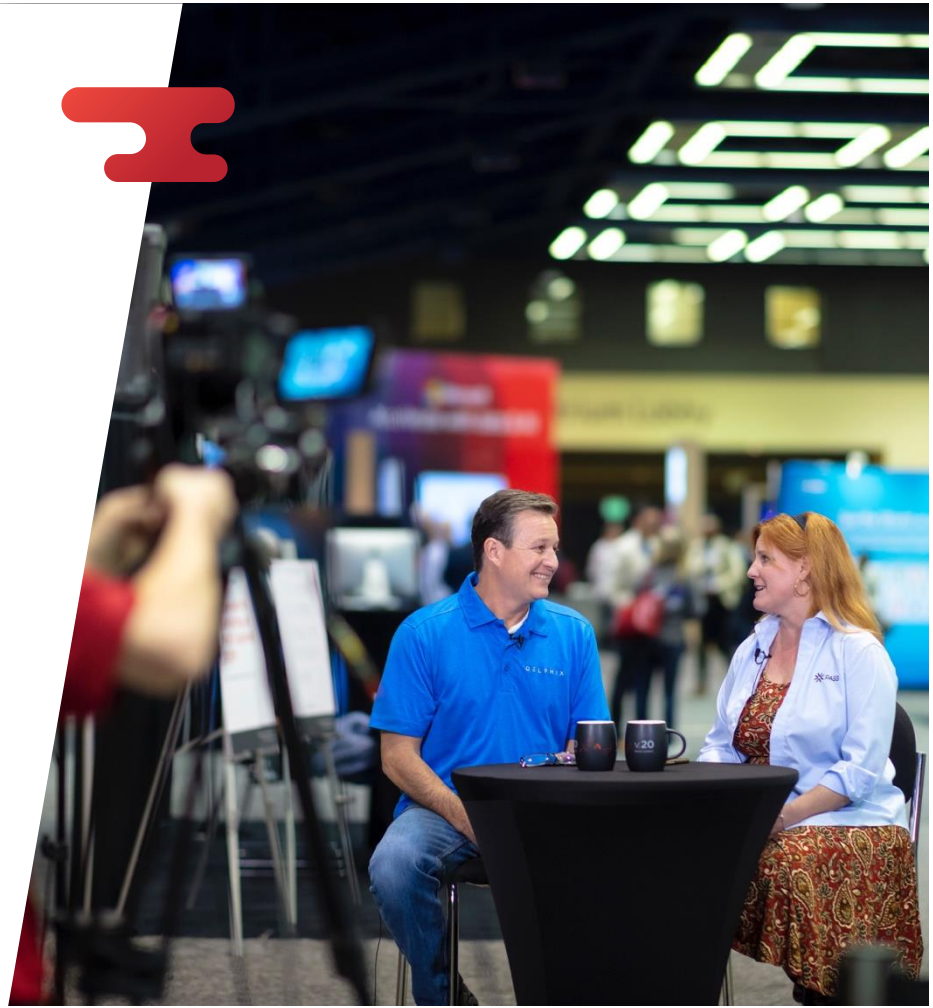


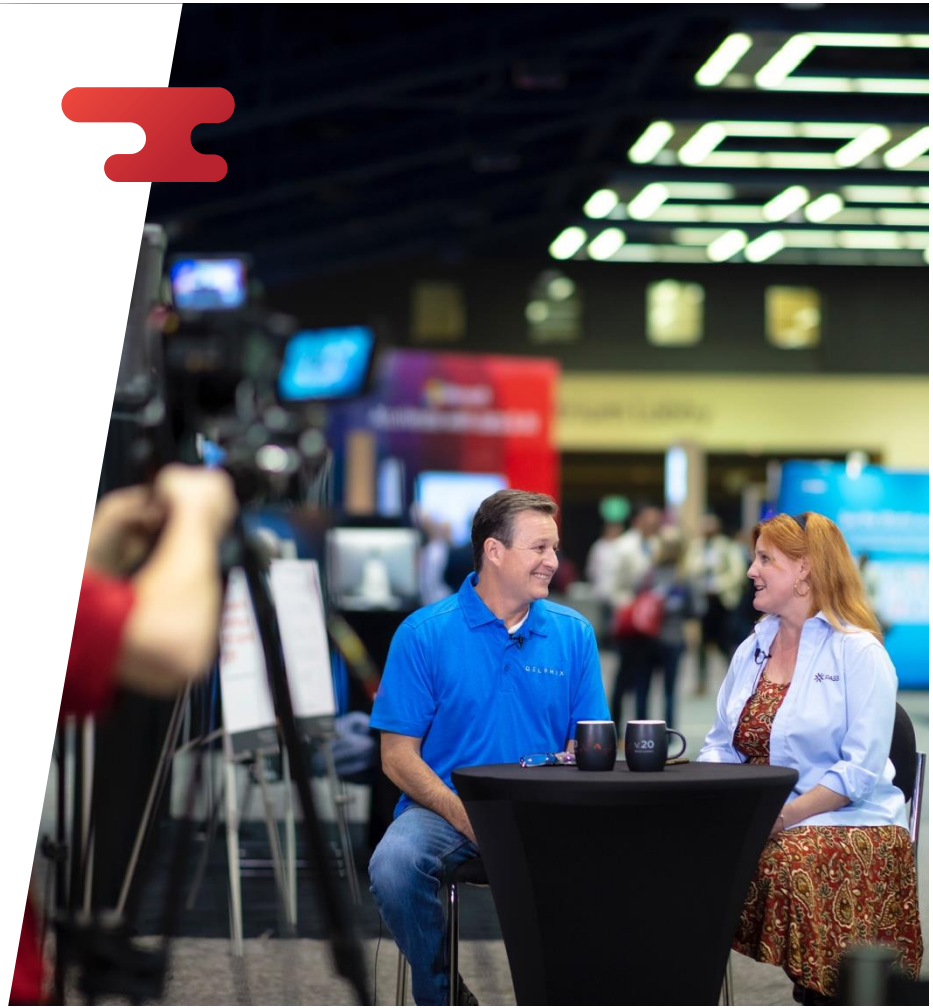
Table variables vs. Temp Tables

Functionality	Table variables	Temp Tables
Transactional behavior	Not transactional	Transactional
Indexes	Only in-line definition	Yes
Constraints	Only PK, Unique, Check	Yes
Auto Stats	No	Yes
Manual Stats	No	Yes
Compilation if used in same batch	TV compiled with the batch, estimate always 1	Deferred until first execution of statement using temp table

Compile deferred till first statement
Compatibility LEVEL = 150

DEMO

Troubleshooting high CPU due to bad estimates



Networking perf recommendations

Use redirect mode

Collocate app & database in same region

Use Accelerated Networking on App VM when possible

Networking tools: psping, Latte for Windows, Sockperf for Linux

If possible, batch data together

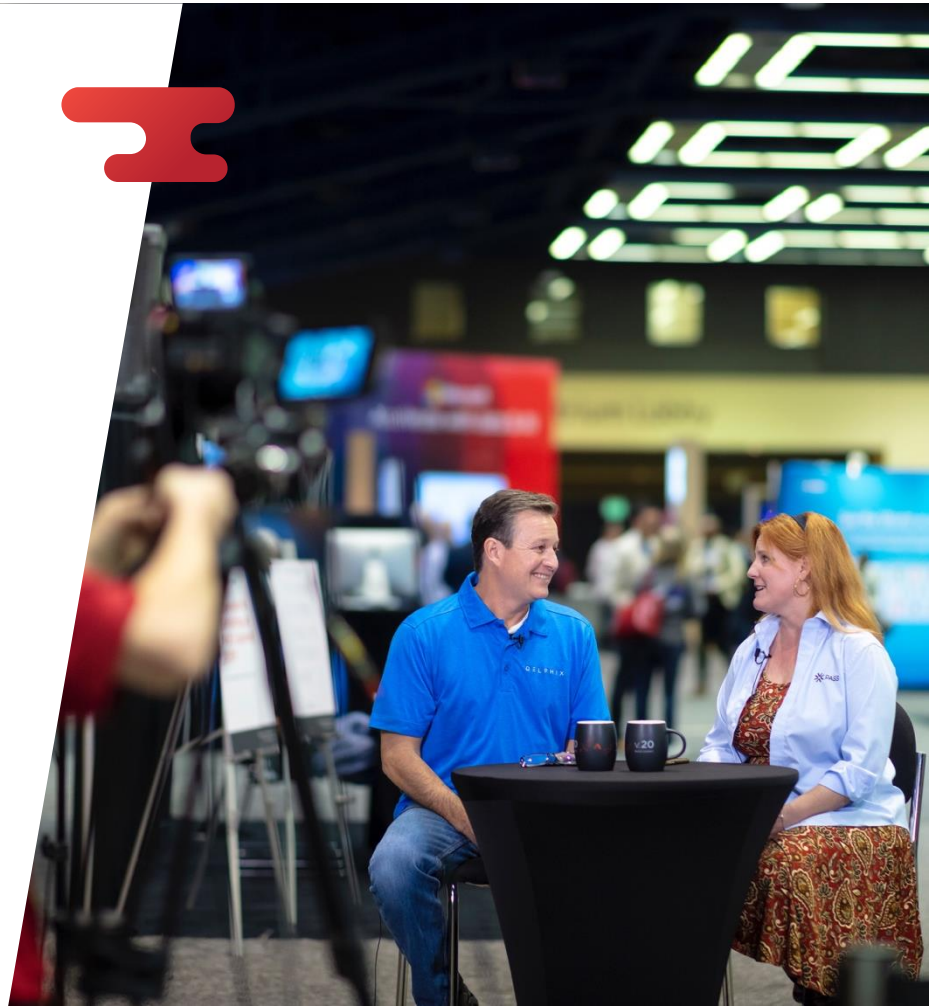
- Using TVPs
- Using multi-row parameterized statements
- Using SQLBulkCopy

Where applicable use SET NOCOUNT ON

Recent versions of drivers and tools are recommended

DEMO

Chatty applications



Thank You

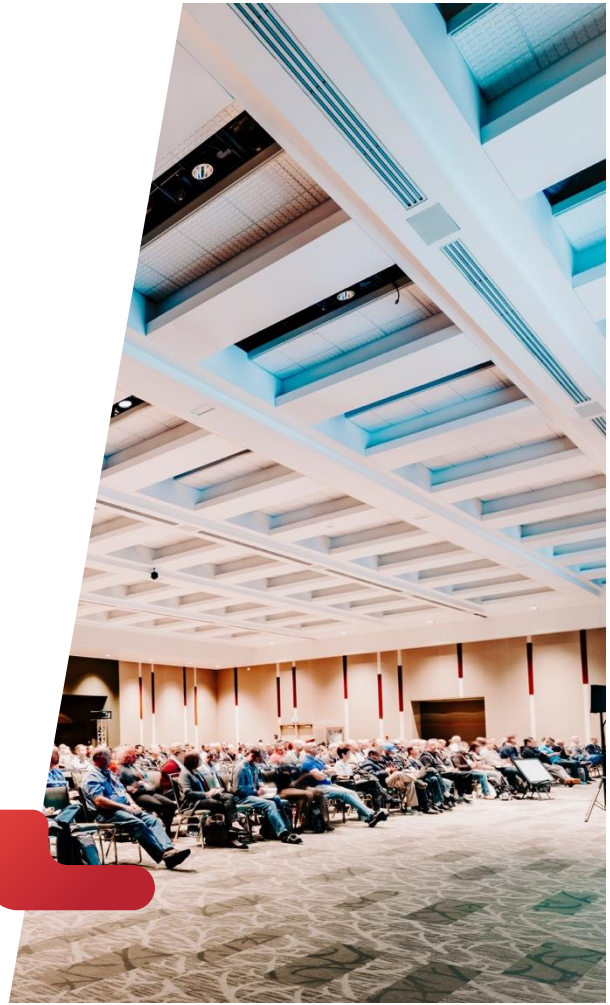
Dimitri Furman, Denzil Ribeiro
Microsoft

PASS SUMMIT2019

Azure SQL Database

Intelligent Database

Joe Sack
Program Manager



Intelligent Database landscape

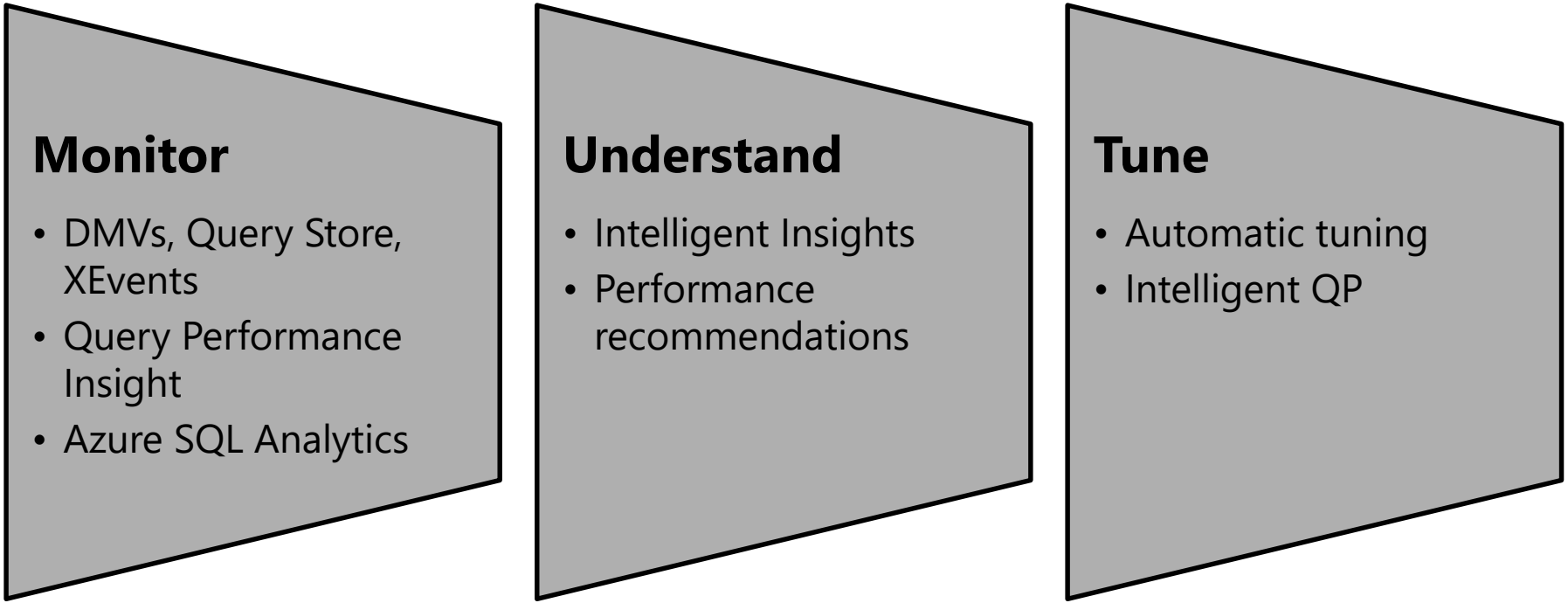
Azure SQL Database full intelligence surfaced across:

SQL Advanced Threat Protection

- Data Classification
- Vulnerability Assessment
- Threat Detection

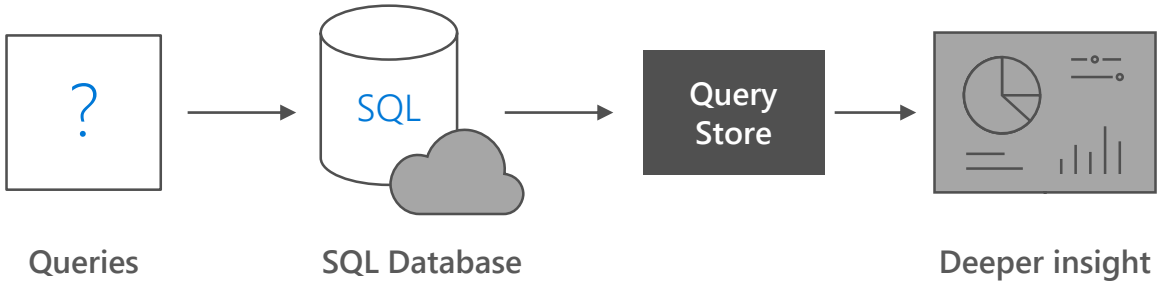
Maximizing app performance…

Maximizing app performance

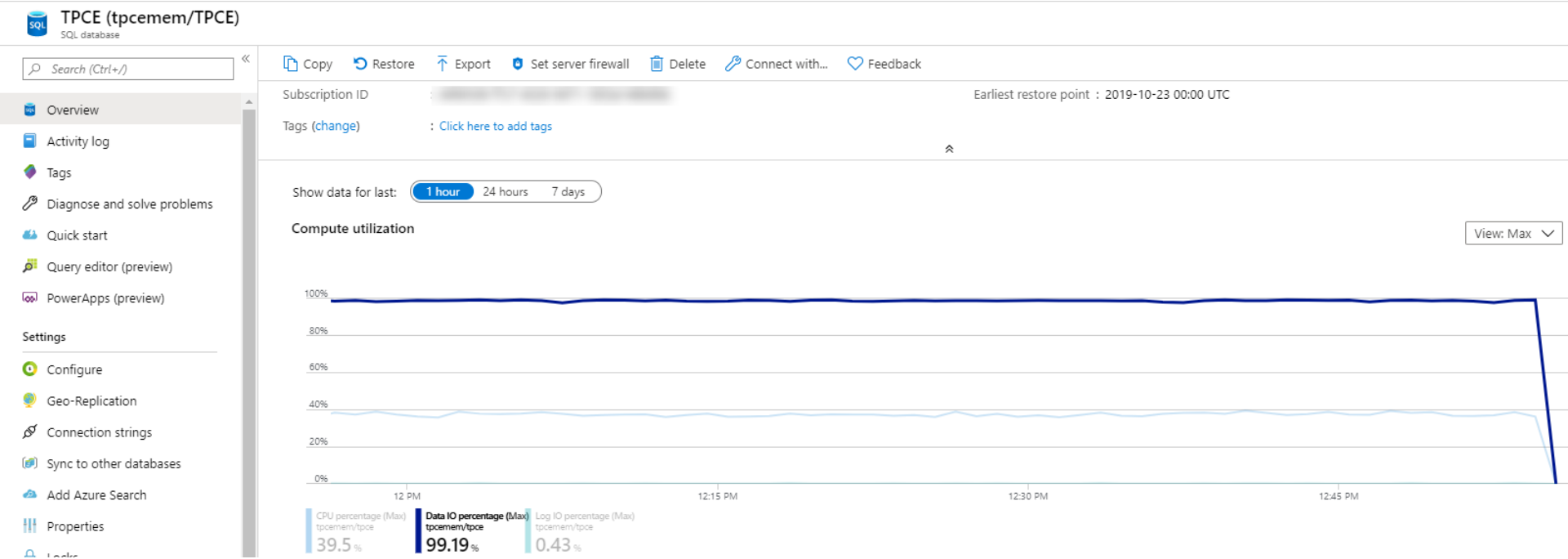


Crucial default: Query Store

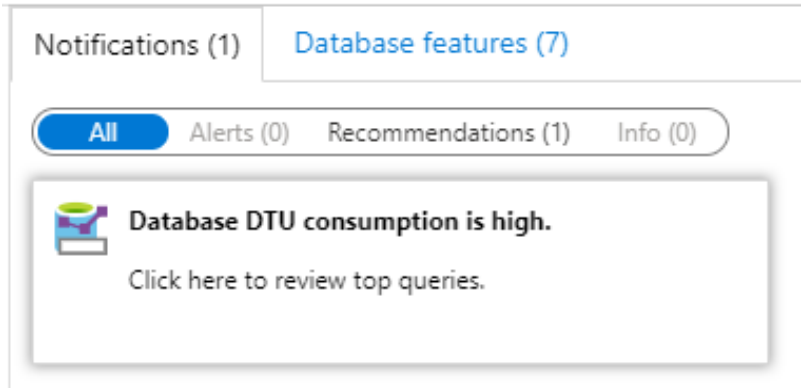
- "Flight data recorder" that simplifies query performance troubleshooting
- On *by default* in Azure SQL Database
- Foundation for performance analysis and tuning features



Portal: Overview, Compute utilization

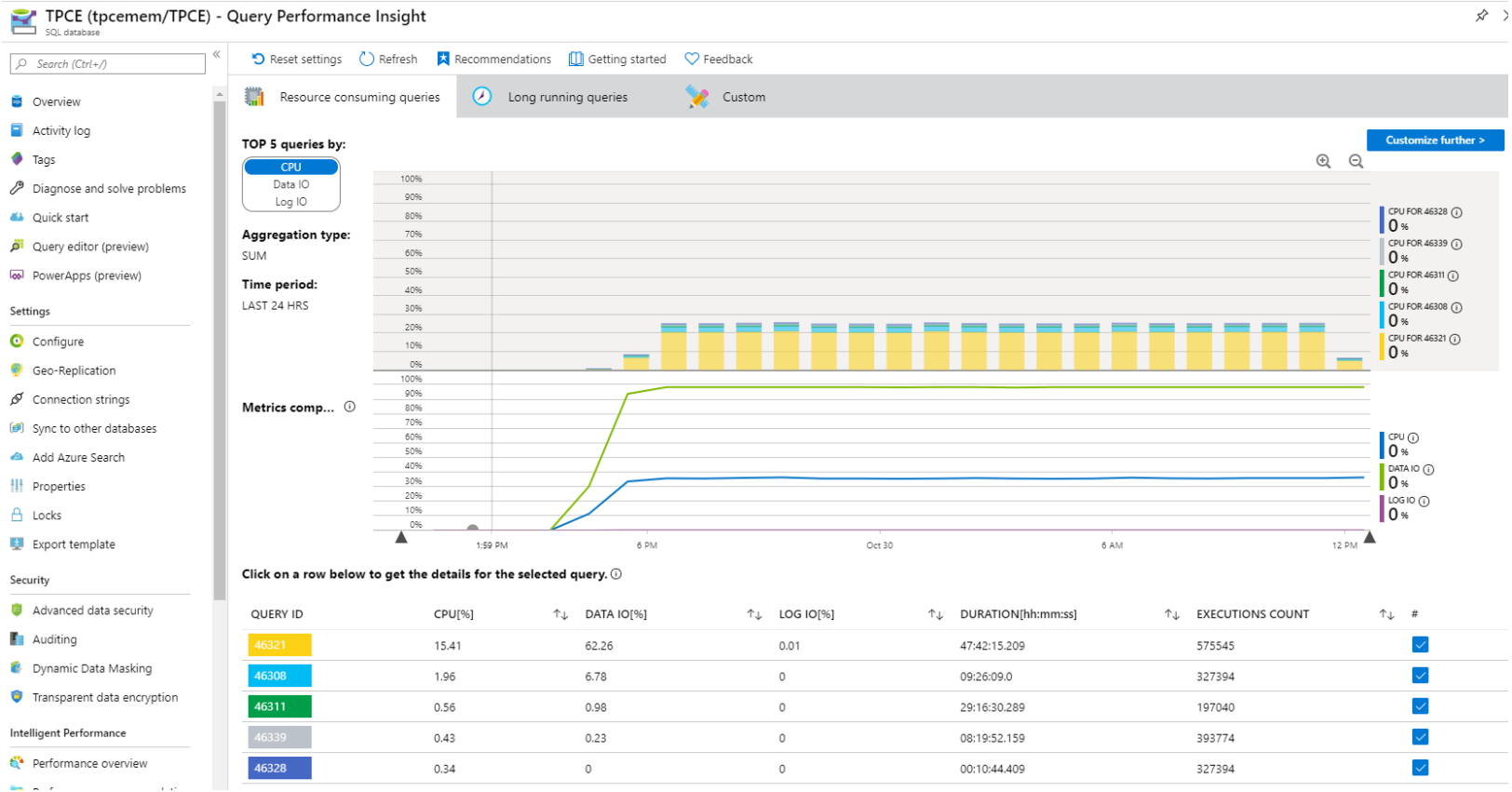


Portal: Overview Notifications

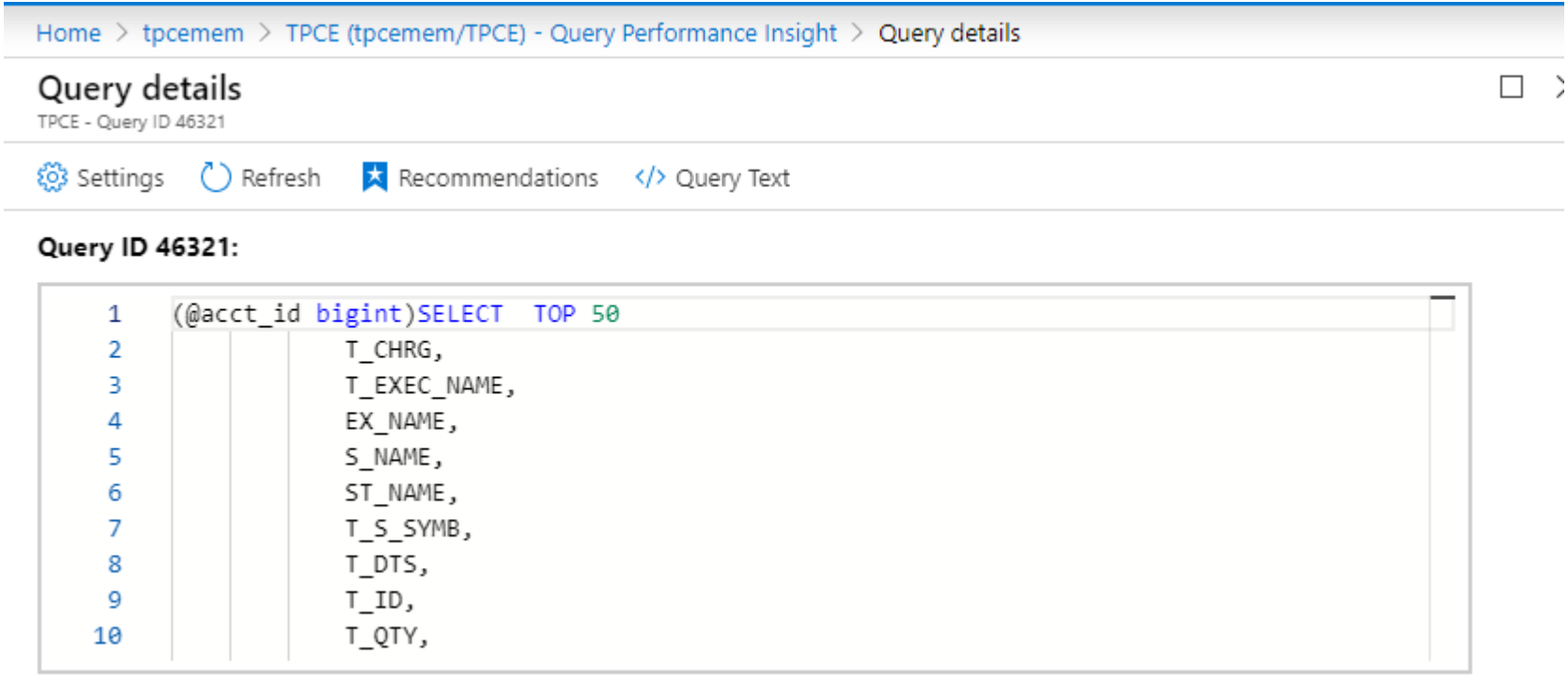


Takes you to Query Performance Insight

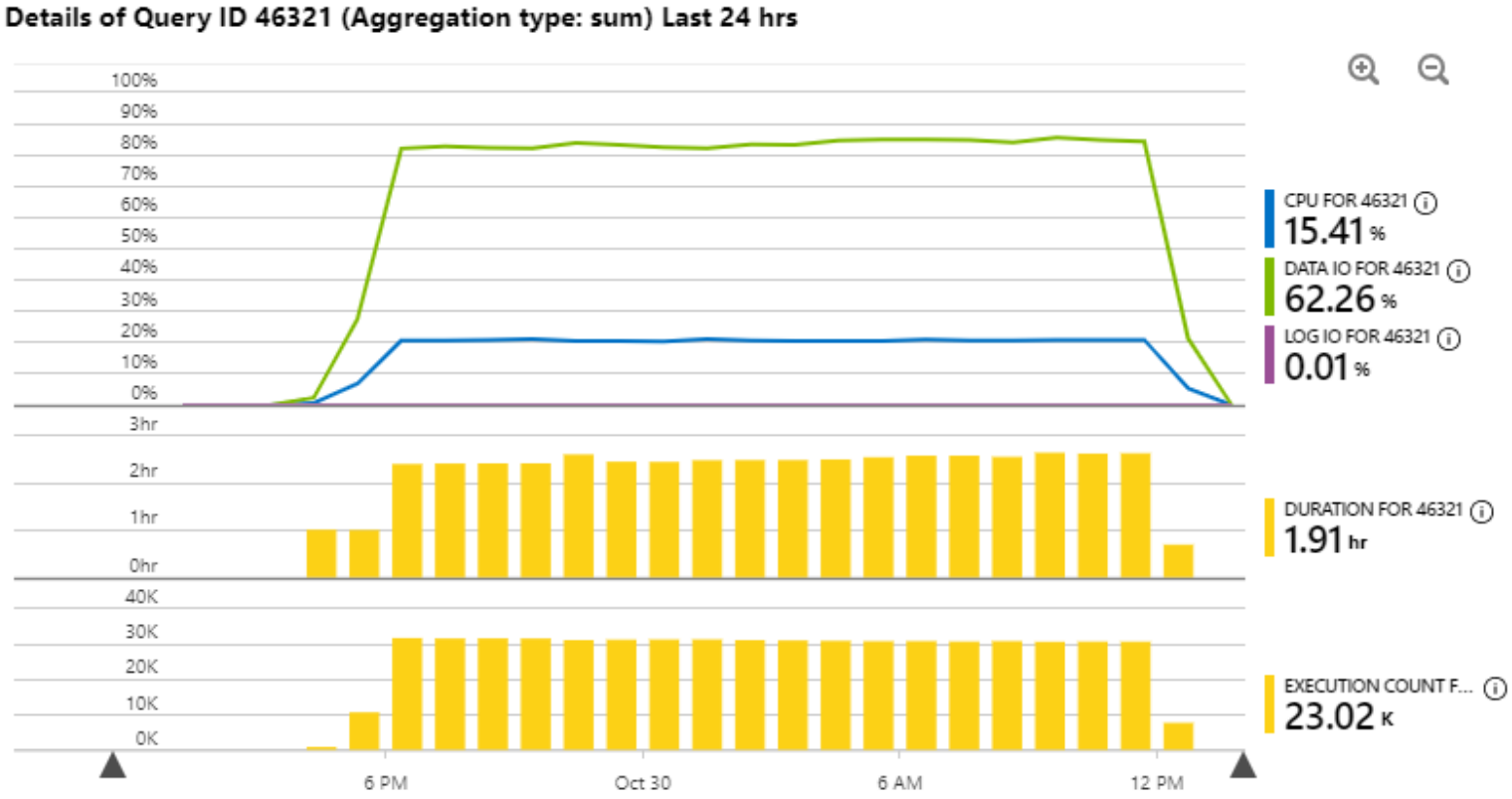
Query Performance Insight



Query Performance Insight, Query Details

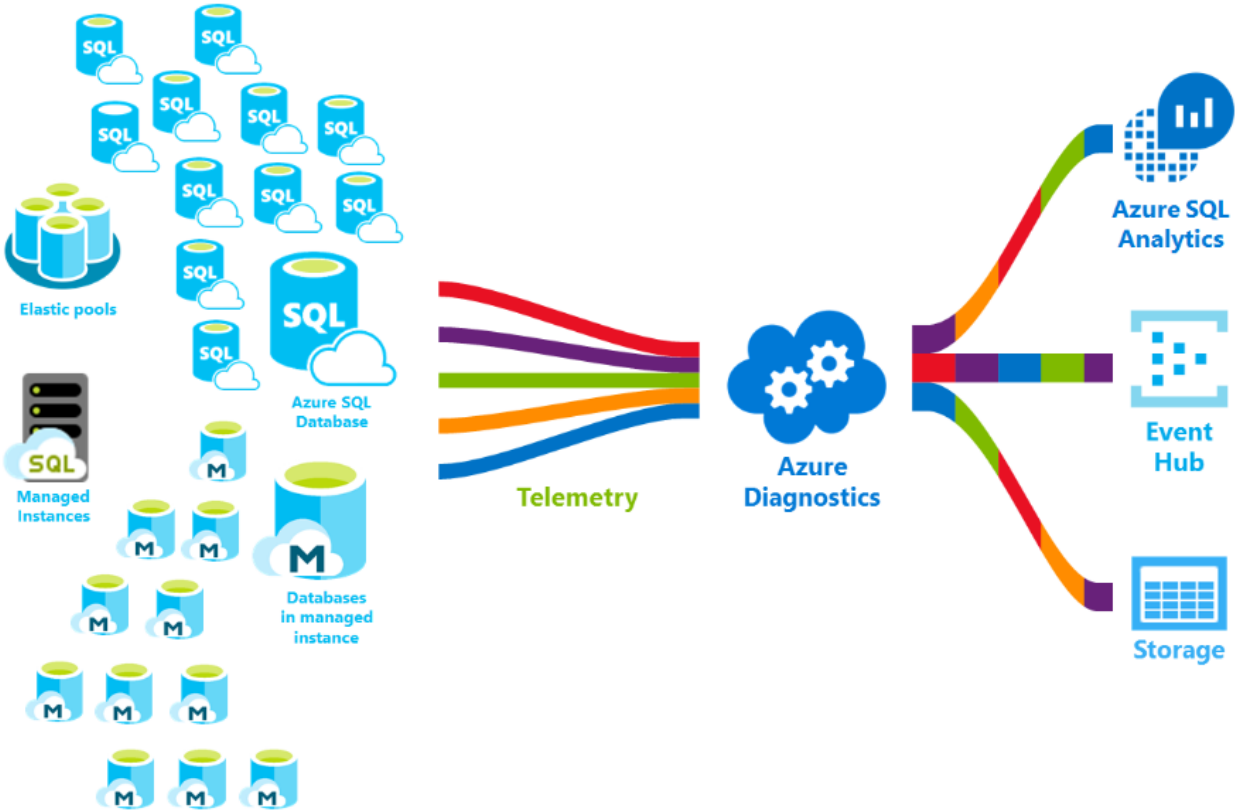


Query Performance Insight, Query Details



PASS

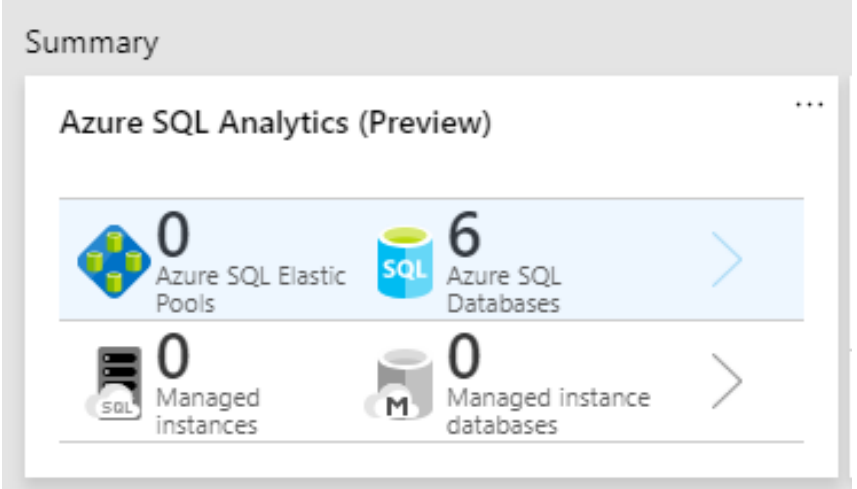
Monitoring and Analysis at Scale



Azure SQL Analytics

Monitor performance of Azure SQL databases, elastic pools, and Managed Instances at scale and across multiple subscriptions through a single pane of glass

- Resource usage
- Query performance
- SQL errors
- Timeouts
- Blocking
- Intelligent insights



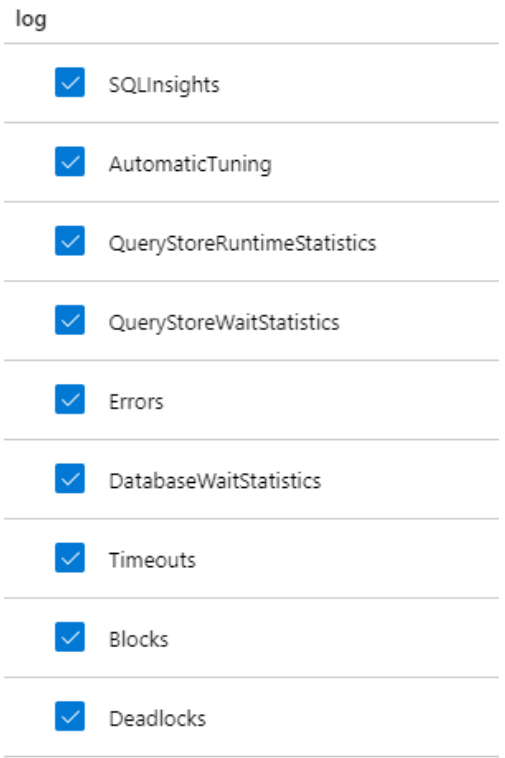
Enabling Azure SQL Analytics

Select and create **Azure SQL Analytics** from marketplace

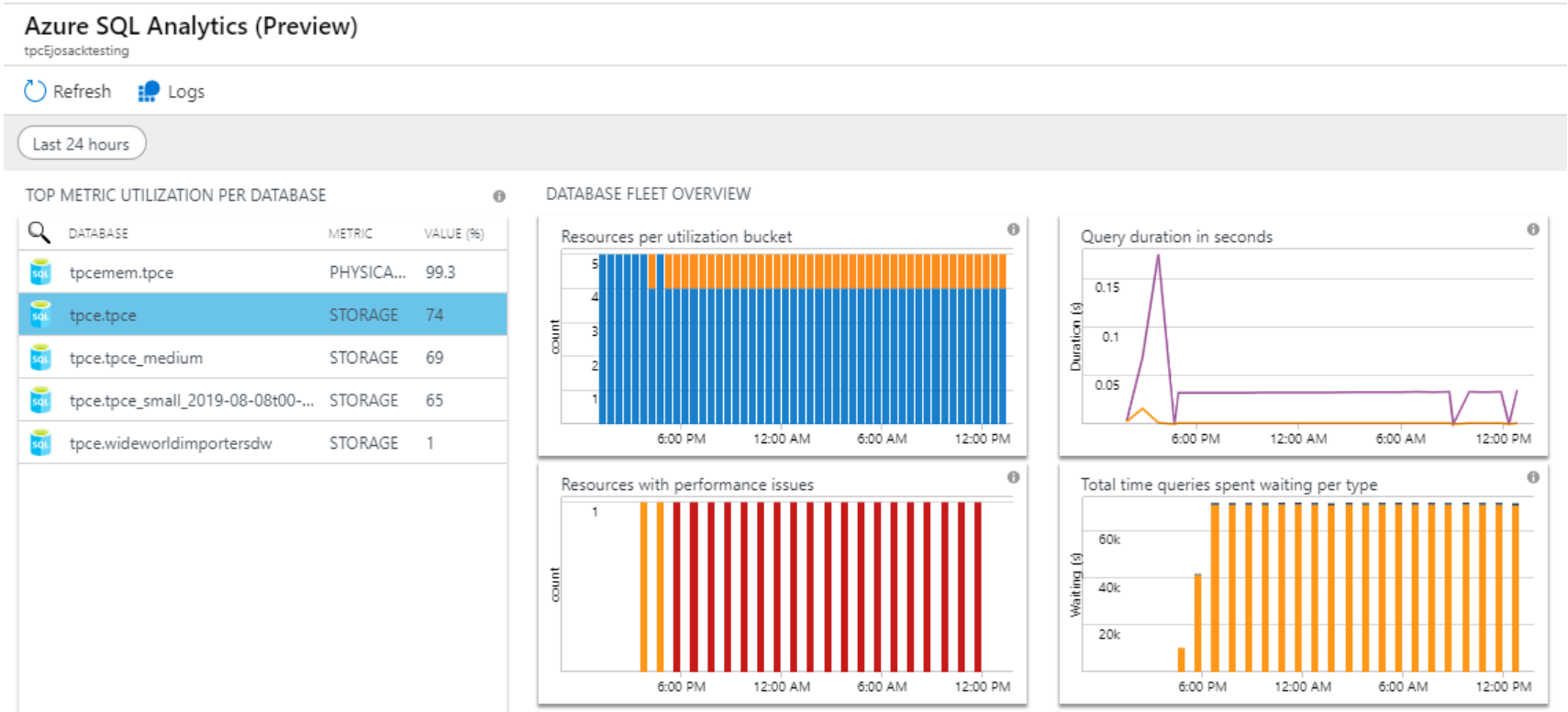
Create or re-use a **Log Analytics workspace**

For each DB resource, enable under **Diagnostic settings** blade, sending to Log Analytics

Select log types for capture



Azure SQL Analytics

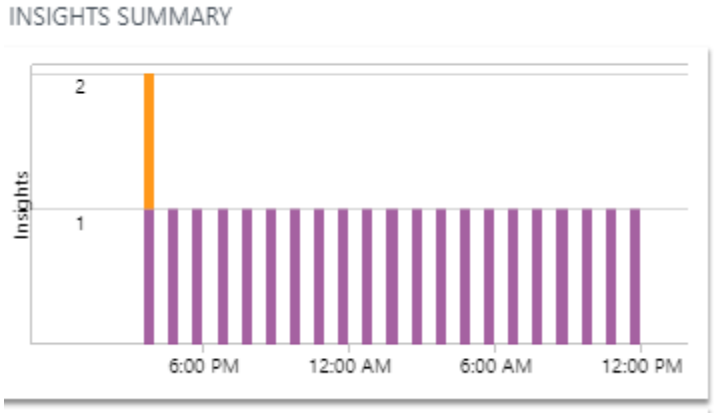


PASS

Intelligent Insights

- Built-in intelligence that continuously monitors database usage
- Uses Artificial Intelligence to detect disruptive events that cause poor performance
  - Current hour vs. last 7 days
- Generates diagnostic log with intelligent assessment for disruptive events
- Performs root cause analysis and provides remediation where possible for performance improvements

Intelligent Insights



View full insights report

Summary

Start: 2019-10-29T21:00:00Z

Worst: 2019-10-30T17:00:00Z

The database is hitting its IO limits. Average consumption has reached 98.7%.

Last Update: 2019-10-30T17:00:00Z

The database is hitting its IO limits. Average consumption has reached 98.7%.

Detected patterns:

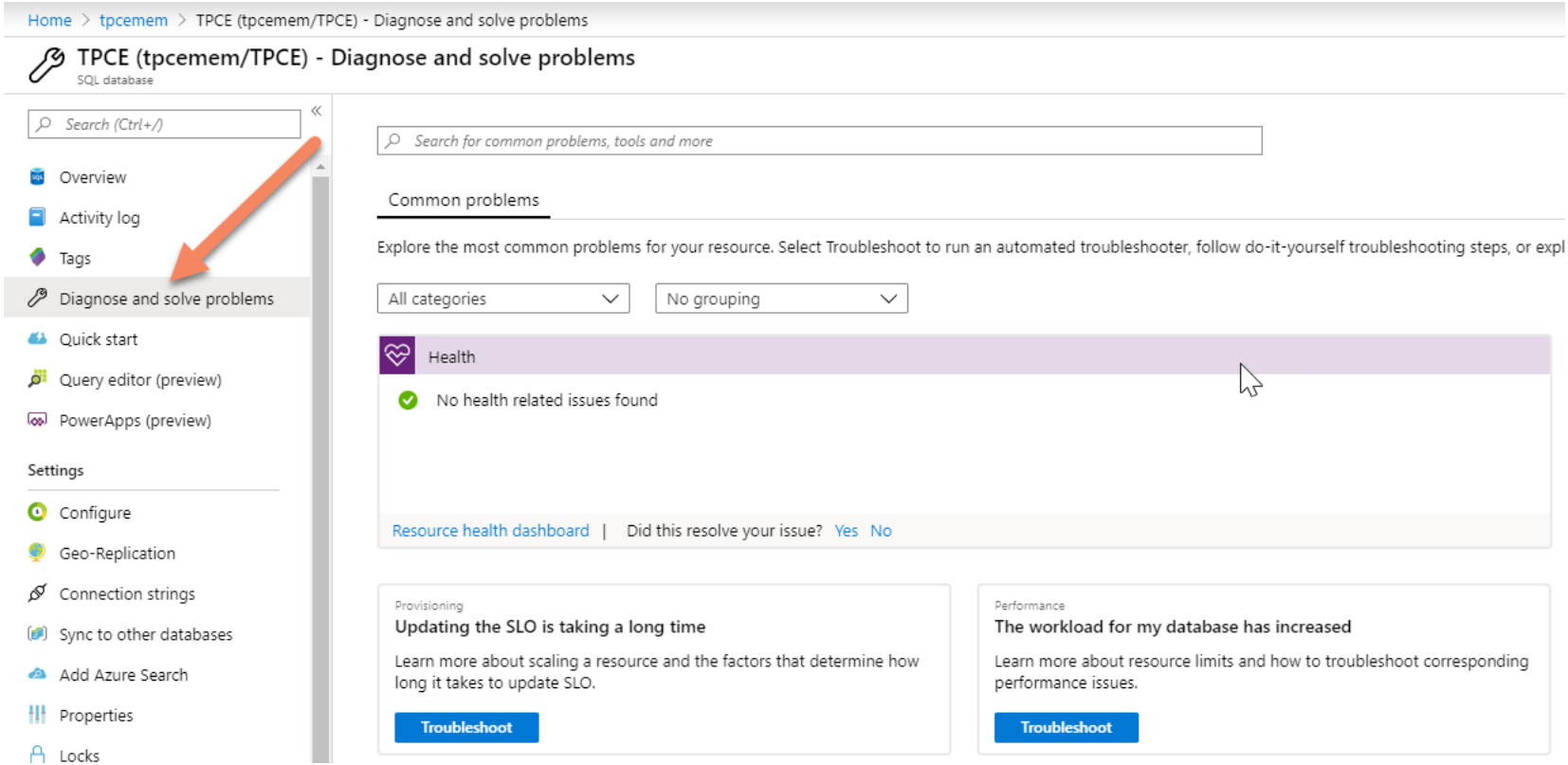
1. Critical SQL Errors

Intelligent Insights patterns

Reaching resource limits	Workload increase	Memory pressure	Locking	Increased MAXDOP
Pagelatch contention	Missing Index	New Query	Increased wait stat	Tempdb contention
DTU shortage	Plan regression	Database scoped configuration change	Slow client	Pricing tier downgrade

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-intelligent-insights-troubleshoot-performance#critical-sql-errors

Diagnose and solve problems



Automated Troubleshooter

Our automated troubleshooter will diagnose Performance issues for TPCE. You can also choose to manually troubleshoot your issue using our recommended steps.

Automated troubleshooter

Troubleshooting "The workload for my database has increased" for TPCE

Our internal service telemetry detected high IO usage greater than 90% for more than 5 consecutive minutes on your Azure SQL DB database **TPCE** in server **tpcemem**.

Recommended Steps

This indicates possible application slowness, timeouts due to lack of IO resources to execute the requested workload during that specific period. Please see the recommended articles for more details.

Recommended Documents

- Consider batching to reduce IO consumption when doing large operations
- Increase Azure Database tier to get more DTUs and hence accommodate for more IO in your environment

Rerun troubleshooter | Download report | Did this resolve your issue? Yes No

Automatic Tuning

One-click to enable | Prevent and mitigate issues | No app changes needed

Tuning actions

- Create missing indexes
- Drop unused/duplicate indexes
- Force last good plan

Configure the automatic tuning options ⓘ

Option	Desired state	Current state
FORCE PLAN	ON OFF INHERIT	**ON** Forced by user
CREATE INDEX	ON OFF INHERIT	**ON** Forced by user
DROP INDEX	ON OFF INHERIT	**ON** Forced by user

PASS

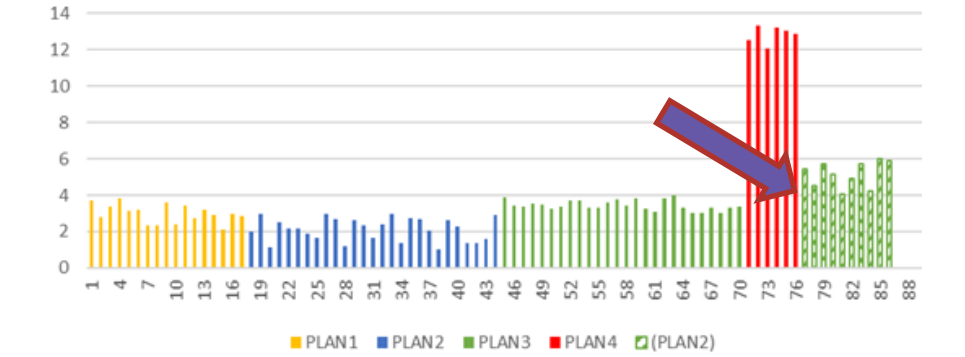
Plan Regression Correction

We can now detect and correct these scenarios without manual intervention

Recommended actions surfaced via sys.dm_db_tuning_recommendations

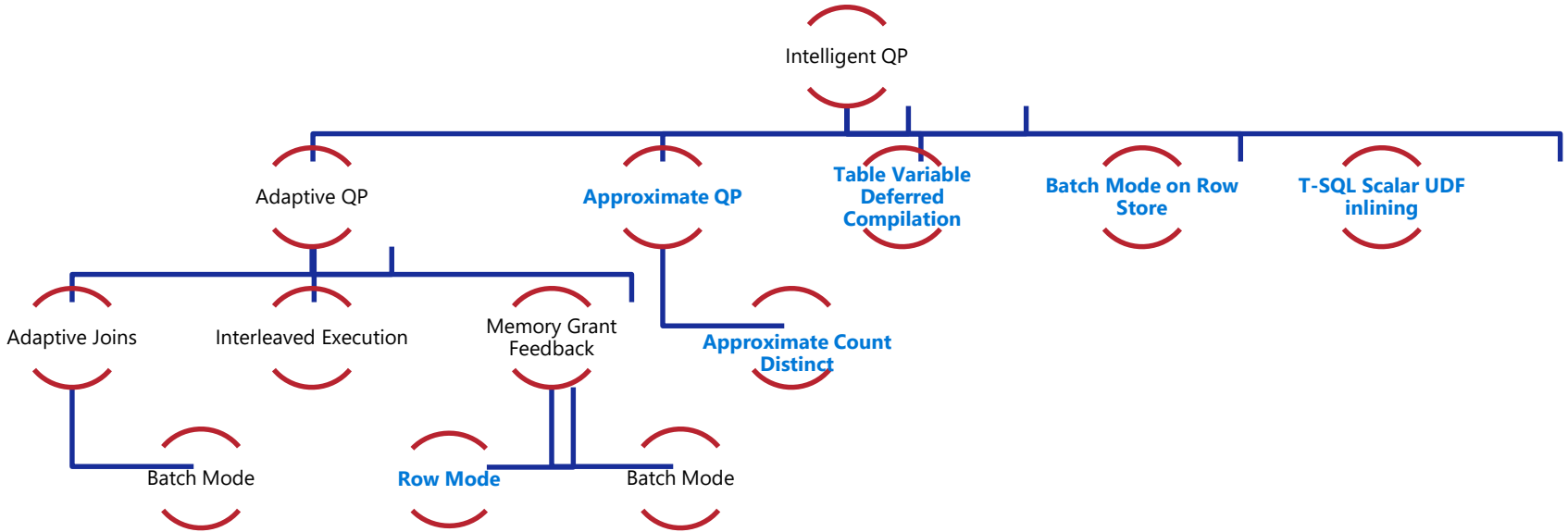
We can now automatically switch to the last known good plan whenever the regression is detected

```
ALTER DATABASE CURRENT
SET AUTOMATIC_TUNING (FORCE_LAST_GOOD_PLAN = ON);
```



Intelligent Query Processing

Query processing features that automatically improve workload performance



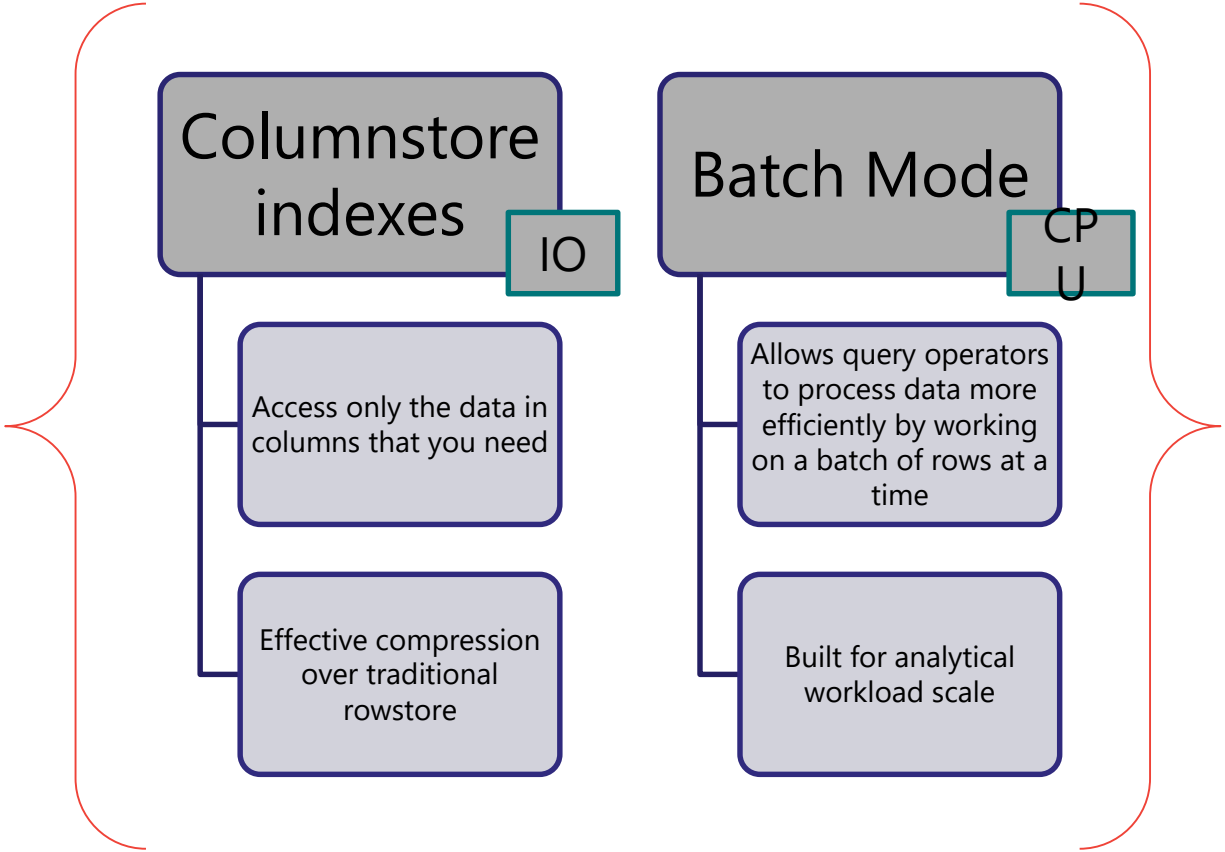
Azure SQL Database

SQL Server 2017 **SQL Server 2019**

PASS

CLICK TO EDIT MASTER TITLE STYLE

Since SQL Server 2012 – we've bound these two features together



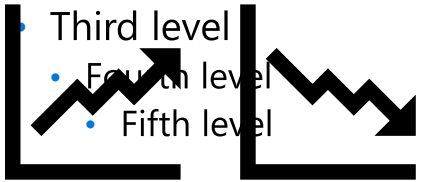
CLICK TO EDIT MASTER TITLE STYLE

- Click to edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - Fifth level

When approximate is good enough…

Approximate Query Processing

- Click to edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - Fifth level



Dashboard scenarios against big data sets with many distinct values (for example, distinct orders counts over a time period) – and many concurrent users

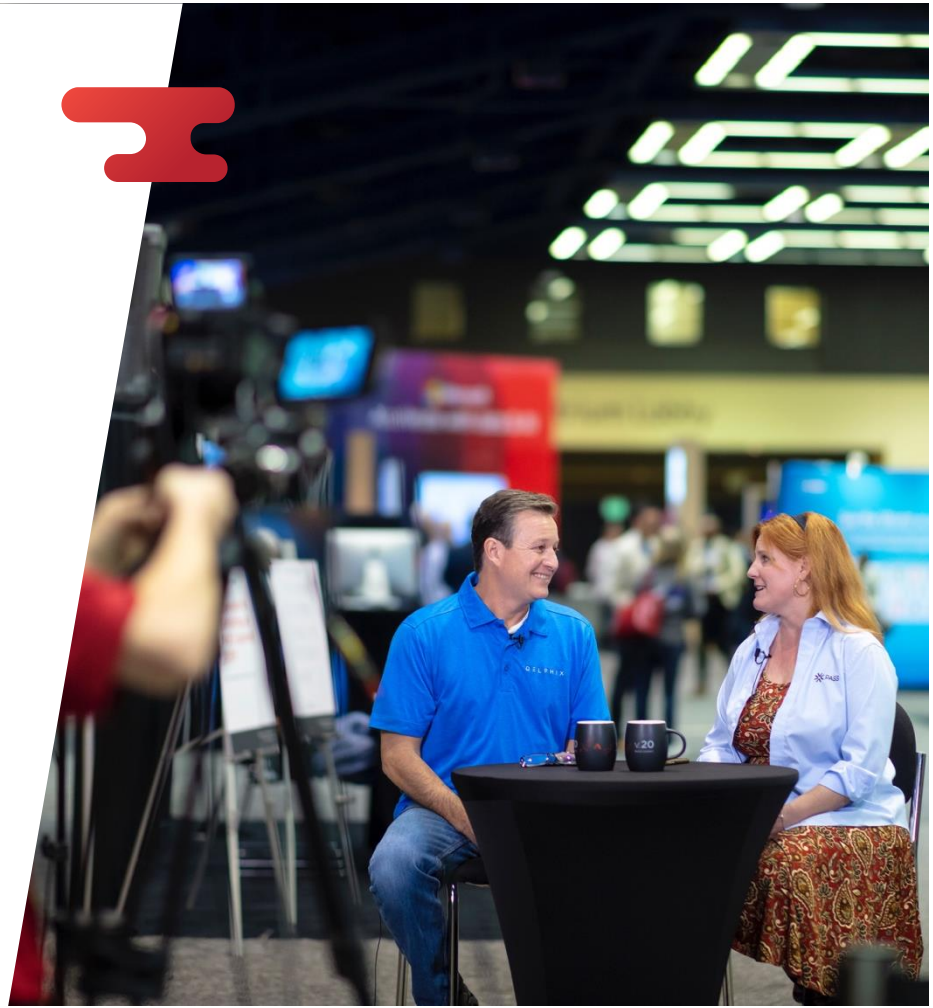
Data science big data set exploration. Need to understand data distributions quickly and exact values are not paramount

CLICK TO EDIT MASTER TITLE STYLE

- Click to edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - Fifth level

DEMO

Intelligent QP



Q&A