Microsoft

# Be a Better DBA with Database Watcher: A Proactive Approach to Database Monitoring

Daniel Taylor
datay@microsoft.com
@dbabulldog
Daniel Taylor | LinkedIn
https://www.youtube.com/@Tales-from-the-Field

Matt Gordon
matt@sqlatspeed.com
@sqlatspeed

Microsoft

# Daniel Taylor, Microsoft

- Sr. Azure FastTrack CXP Engineer
- Started as Sybase ASE / IQ DBA
- SQL Server 2000 and beyond
- Proud Geek, Video Gamer, and lover of the mountains

@dbabulldog
datay@microsoft.com
https://github.com/dbabulldog-repo/SharedPresentations.git

Microsoft

# Matt Gordon, Centric Consulting

- Lead Microsoft Data Architect
- Working in data for (redacted) years
- SQL Server 2000 and beyond
- Data & AI consultant by trade, racing driver by hobby

@sqlatspeed
matt.gordon@centricconsulting.com
https://github.com/sqlatspeed/2024_presentations

CENTRIC

# Racing Family to #sqlfamily



https://www.youtube.com/@Tales-from-the-Field

# Agenda

aka.ms/dbwatcher

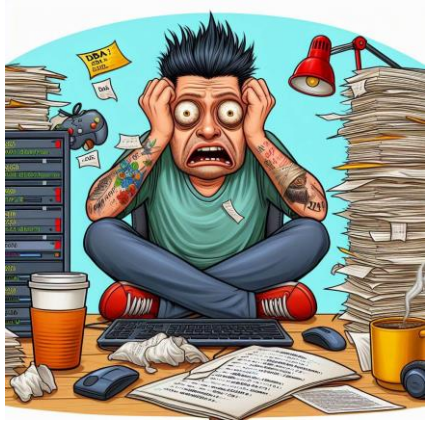
- DBAs are awesome
- Value prop of Database Watcher
- Deeper look into Database Watcher & Monitoring
- Price, roadmap, and timeline
- Final Q&A

# Types of DBAs

# DBAs can be reactive or proactive

## Reactive DBA

- focuses on fixing problems after they occur, rather than preventing them beforehand
- may spend most of their time troubleshooting errors, restoring data from backups, or resolving performance issues
- often work under pressure and have to deal with frequent interruptions and emergencies
- may not have enough time or resources to plan, test, and implement proactive measures, such as regular maintenance, optimization, or security audits
- can lead to higher costs, lower availability, and poor customer satisfaction.



# DBAs can be reactive or proactive

## Proactive DBA

- mainly focuses on preventing problems before they occur, rather than fixing them afterward
- may spend most of their time planning, testing, and implementing proactive measures, such as regular maintenance, optimization, or security audits
- often work with foresight and have a clear vision of the goals and requirements of their databases.
- may have more time and resources to optimize performance, ensure availability, and enhance security
- approach can lead to lower costs, higher customer satisfaction, and better business outcomes

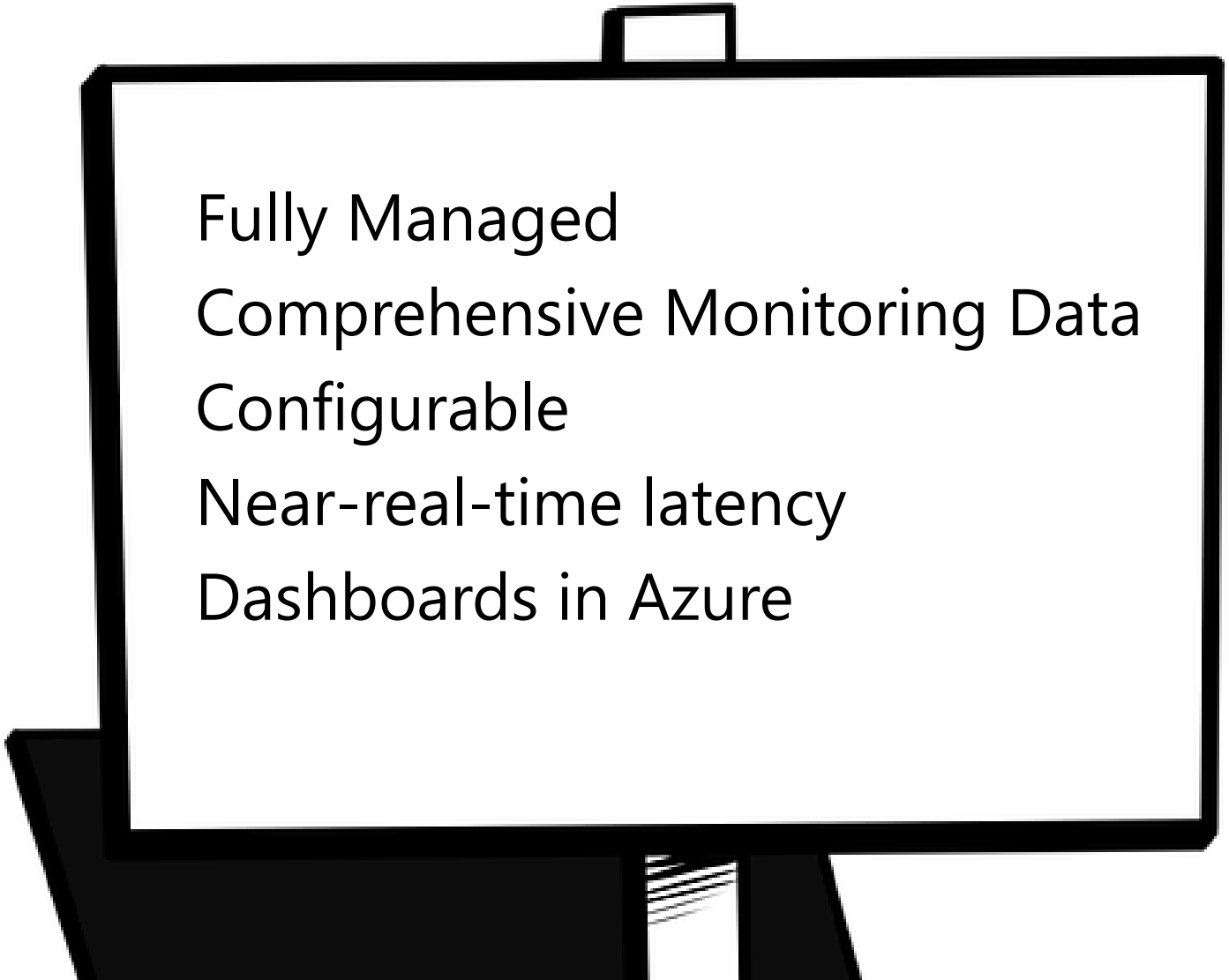
# What's one thing we can do to be Proactive?

# What choices do DBAs have for monitoring?

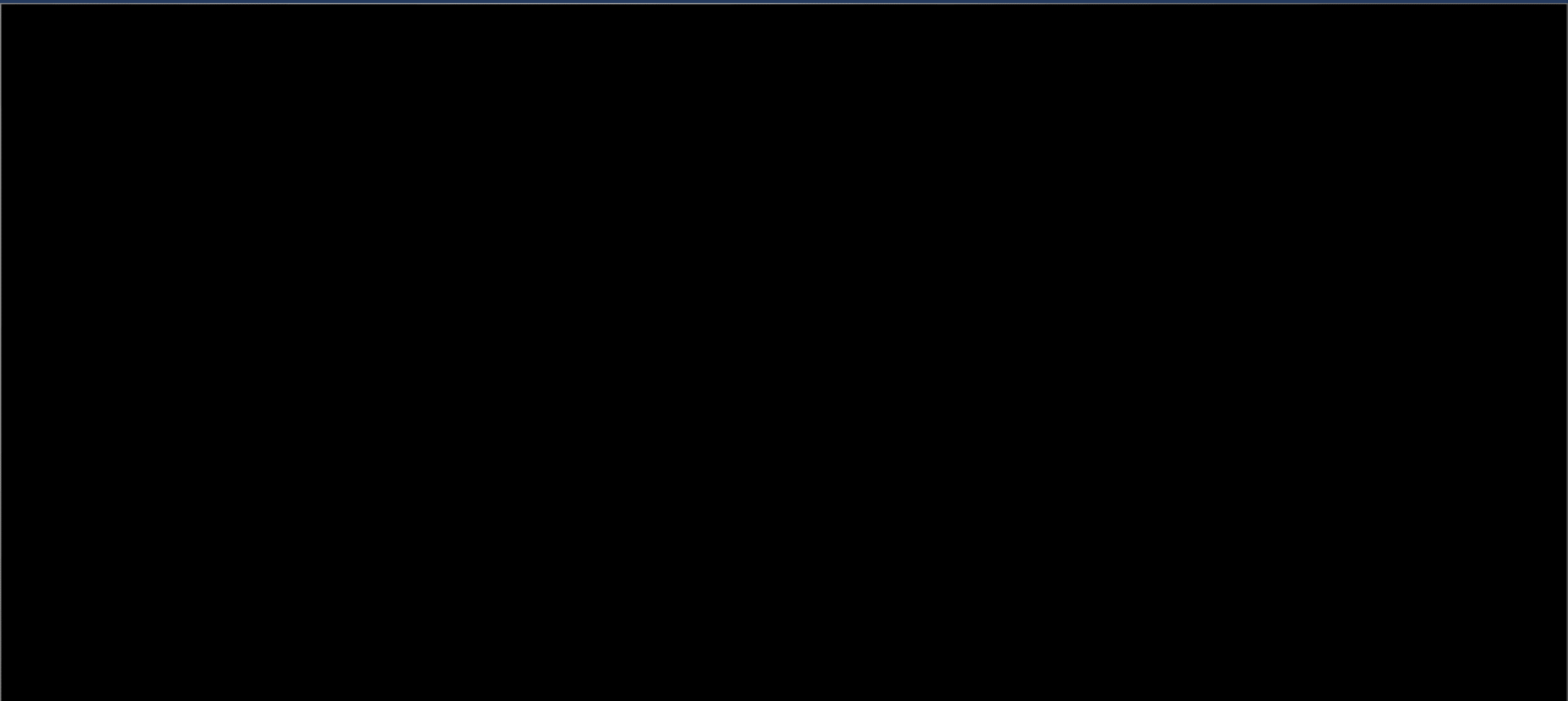
- Third party options
- Azure Options
  - Azure SQL Analytics
  - SQL Insights
- Homegrown written in-house

# DBAs are awesome!

# What do we want? When do we want it?



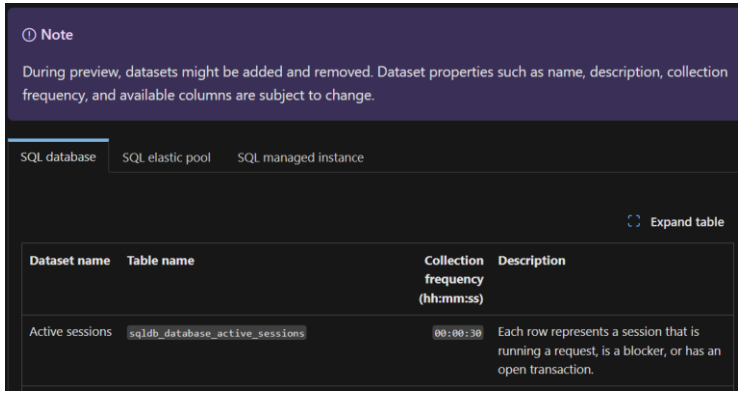
# A solution for you!



# Looking closer at Database Watcher

- Azure SQL Database and Azure SQL Managed Instance
  - SQL DB, Hyperscale, Elastic Pools, Replicas
- Microsoft managed data collection agent

- 10-20 datasets per target type, derived from 70+ system views

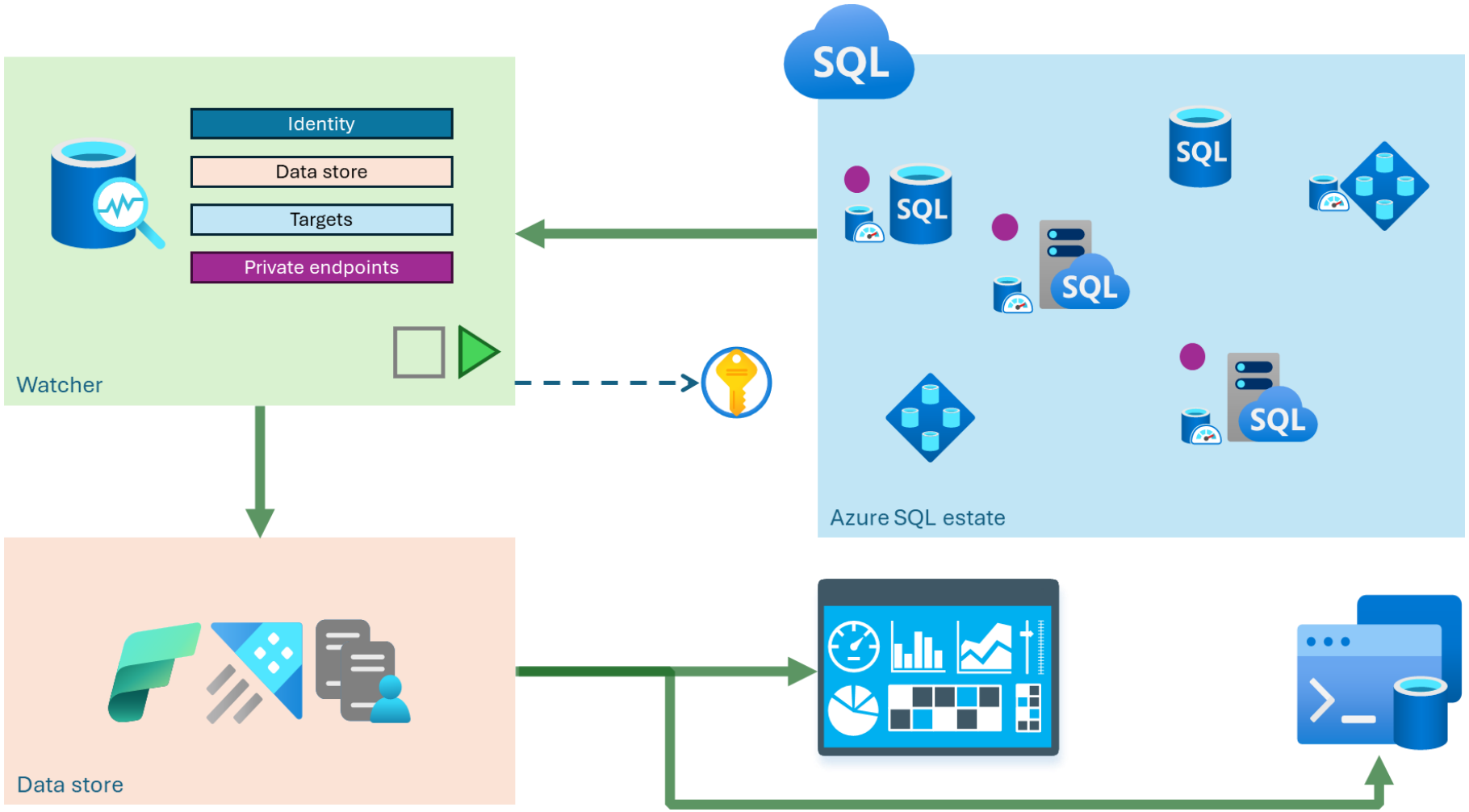


# Looking closer at Database Watcher


- Entra or SQL with key vault secrets
  - Entra is recommended
- Private Link or Firewall/NSG
- Single pane of glass with detailed visualizations
- Azure Data Explorer (including free ADX) or Fabric Real-Time ~~Analytics~~ Intelligence

# Introduction to Database Watcher

# Introducing Database Watcher



# Network and data access

## Principles

1. Customers fully control network connectivity and SQL permissions
2. Require minimal access to customer databases
3. Validate that only the necessary access is granted

## Private connectivity

- Watcher uses managed private endpoints
- Resource owner approves/rejects/deletes each private endpoint

## SQL permissions

| SQL Database | SQL Managed Instance |
|---|---|
| Membership in server roles:<br><br>##MS_DatabaseConnector##<br>##MS_ServerPerformanceStateReader##<br>##MS_DefinitionReader## | CONNECT SQL<br>CONNECT ANY DATABASE<br>VIEW ANY DATABASE<br>VIEW ANY DEFINITION<br>VIEW SERVER PERFORMANCE STATE<br>SELECT on specific tables in MSDB |

# Demo

Look at Database Watcher Deployment & Configurations

# Dataset categories

- **Active sessions**
  Connections, sessions, requests
- **Backup history**
- **CDC, change feed**
- **Geo-replication**
- **Indexes**
  Metadata, operational stats
- **Memory utilization**
  Memory clerks, out-of-memory events
- **Missing indexes**
- **Network connectivity**
- **Performance counters**
  200+ in total
- **Query Store**
  Query runtime statistics and wait categories
- **Replica health**
  Availability groups, secondary replicas
- **Resource utilization**
  CPU, Data IO, Log IO, workers, ...
- **Properties and configuration**
  Server and database level
- **Session statistics**
- **SQL Agent**
  Job state and history
- **Storage IO**
  IOPS, throughput, latency
- **Storage utilization**
  Volume and database file level
- **Tables**
  Metadata, schema, size
- **Wait statistics**
- **... and more to come**

# Demo

Look at Database Watcher Datasets & Brief Use Case

# Using Kusto for SQL monitoring data

## Highly scalable

- A single Kusto database can support thousands of SQL monitoring targets
- Vertical and horizontal scaling, manual scale or auto-scale, pause/resume
- Configurable data retention per database and per table
- Time-proven as a telemetry data store

## Optimized for time-series data and analytics

- Powerful KQL language
- Columnstore
- Streaming ingestion for low latency
- KQL explainer video here

## Short learning curve for SQL professionals

- Uses relational database concepts
- Has T-SQL support at protocol level
- Can offload data to a SQL database or Azure storage on a schedule

# Using Kusto for SQL monitoring data

SQL database | SQL elastic pool | SQL managed instance

Kusto

```
let duration = 1d;
sqlmi_instance_properties
| where sample_time_utc > ago(duration)
| summarize arg_max(sample_time_utc, *) by managed_instance_name, replica_type
| project-rename last_sample_time_utc = sample_time_utc
| sort by tolower(managed_instance_name),
         case(
             replica_type == "Primary", 0,
             replica_type == "Geo-replication forwarder", 1,
             replica_type == "HA secondary", 2,
             3) asc;
```

Use KQL
Or
Familiar T-SQL

SQL database | SQL elastic pool | SQL managed instance

SQL

```
DECLARE @DurationHours int = 24;

SELECT p.sample_time_utc,
       p.managed_instance_name,
       p.replica_type,
       p.service_tier,
       p.hardware_generation,
       p.product_update_level,
       p.logical_cpu_count,
       p.storage_space_used_mb,
       p.reserved_storage_mb,
       p.database_engine_memory_mb,
       p.database_engine_build_time,
       p.database_engine_start_time_utc
FROM sqlmi_instance_properties AS p
INNER JOIN (
           SELECT managed_instance_name,
                  replica_type,
                  MAX(sample_time_utc) AS last_sample_time_utc
           FROM sqlmi_instance_properties
           WHERE sample_time_utc > DATEADD(hour, -@DurationHours, SYSUTCDATETIME())
           GROUP BY managed_instance_name,
                    replica_type
           ) AS ls
ON p.managed_instance_name = ls.managed_instance_name
   AND
   p.replica_type = ls.replica_type
   AND
   p.sample_time_utc = ls.last_sample_time_utc
WHERE p.sample_time_utc > DATEADD(hour, -@DurationHours, SYSUTCDATETIME())
ORDER BY LOWER(managed_instance_name) ASC,
         CASE replica_type
              WHEN 'Primary' THEN 0
              WHEN 'Geo-replication forwarder' THEN 1
              WHEN 'HA secondary' THEN 3
         END ASC;
```

# Demo (Time Permitting)

Analyze monitoring data in
KQL and T-SQL

# Challenges I faced during deployment

- Azure Key Vault
  - Must use RBAC permission model
  - If not using private connectivity vault must have public access from all networks enabled
  - You will have 2 secrets (Login & Password)
- If adding a new SQL Target be sure to apply security to target
- If creating a Managed private endpoint be sure it is approved
- If adding a SQL Target stop and start the database watcher
- Elastic Pools?

# What's next for Database Watcher?

# What is the cost of Database Watcher?

## Free

- Watchers
- Dashboards
- **No per instance/database/elastic pool/user cost**

## Paid

- Azure Data Explorer compute and storage
  - Can use an existing cluster
  - Can use the free cluster for evaluations and POCs
- Or use Real-Time Analytics in Microsoft Fabric
- Key Vault, if used
- Azure network bandwidth for cross-region monitoring

# Feature roadmap

- Alerts
- Monitor SQL Server on Azure IaaS VM
- Increase number of SQL targets per watcher
- Improve manageability
  - PowerShell
  - Azure CLI
- Under consideration
  - Configurable data collection
  - Large estates: enable monitoring for logical server, resource group, subscription
  - Extended event collection
  - …

# Timeline


* GA timeline depends on public preview feedback

# Learn more and send feedback

## Documentation

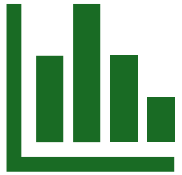
https://aka.ms/dbwatcher

- Overview
- Quickstart
- Create and configure
- Data collection and datasets
- Analyze monitoring data
- FAQ

## https://aka.ms/sqlfeedback

Post under the *Azure SQL* group

Include *dbwatcher* in the title



## Email

dbwatcherfeedback@microsoft.com

# Q & A


https://www.youtube.com/@Tales-from-the-Field