

# Monitor and optimize operational resources in Azure SQL

Introduction to performance monitoring, understand performance problems, and proper configuration options

# Objective

---



Monitor activity and compare to a baseline

---



Explore maintenance tasks related to performance

---



Identify major causes of performance problems

---



Configure databases and resources for optimal performance

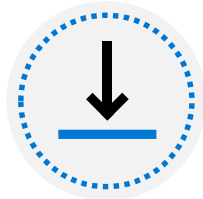
Describe performance monitoring



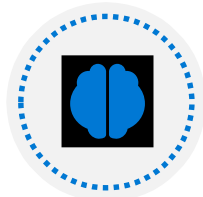
# Objectives



Understand methods to review potential performance issues



Learn how to collect metrics for an established baseline



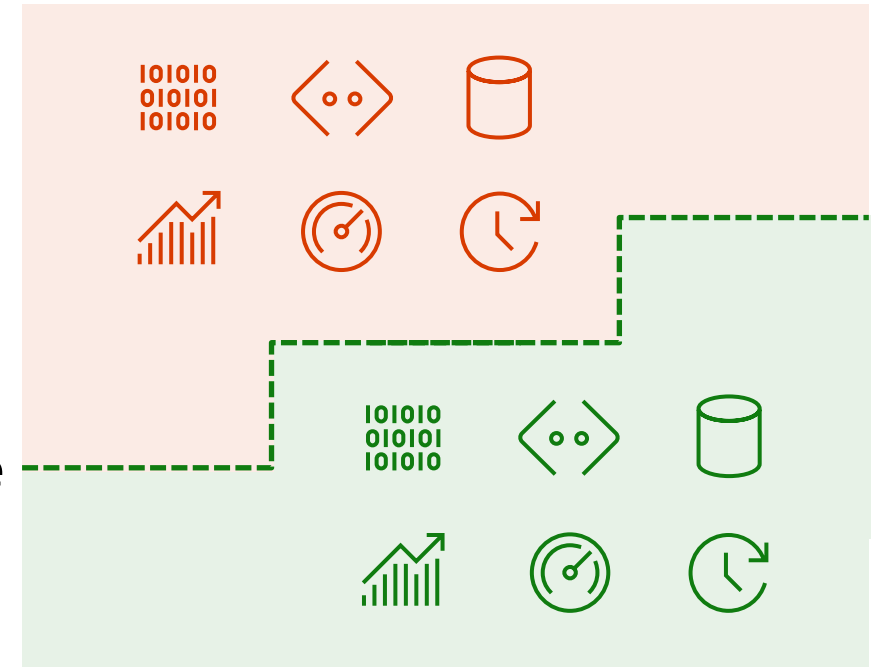
Understand Extended Events and Azure Intelligent Insights

# What is a baseline?

## A baseline

A general set of metrics around performance and utilization that allow you to easily separate performance anomalies from normal usage patterns

Baseline



# Establishing a baseline

---

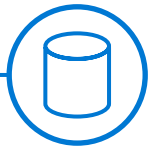
Having a baseline of system performance data allows you to make better decisions for performance tuning

You can separate code problems from simple increased utilization

You may also proactively allocate more compute resources as your workload increases over time

# Performance on an Azure Virtual Machine

Built-in tool in the Windows operating system



Collect detailed, granular performance data about SQL Server and Windows operations



Group Performance Monitor counters together to easily gather and analyze performance data



Data can be easily exported to archive

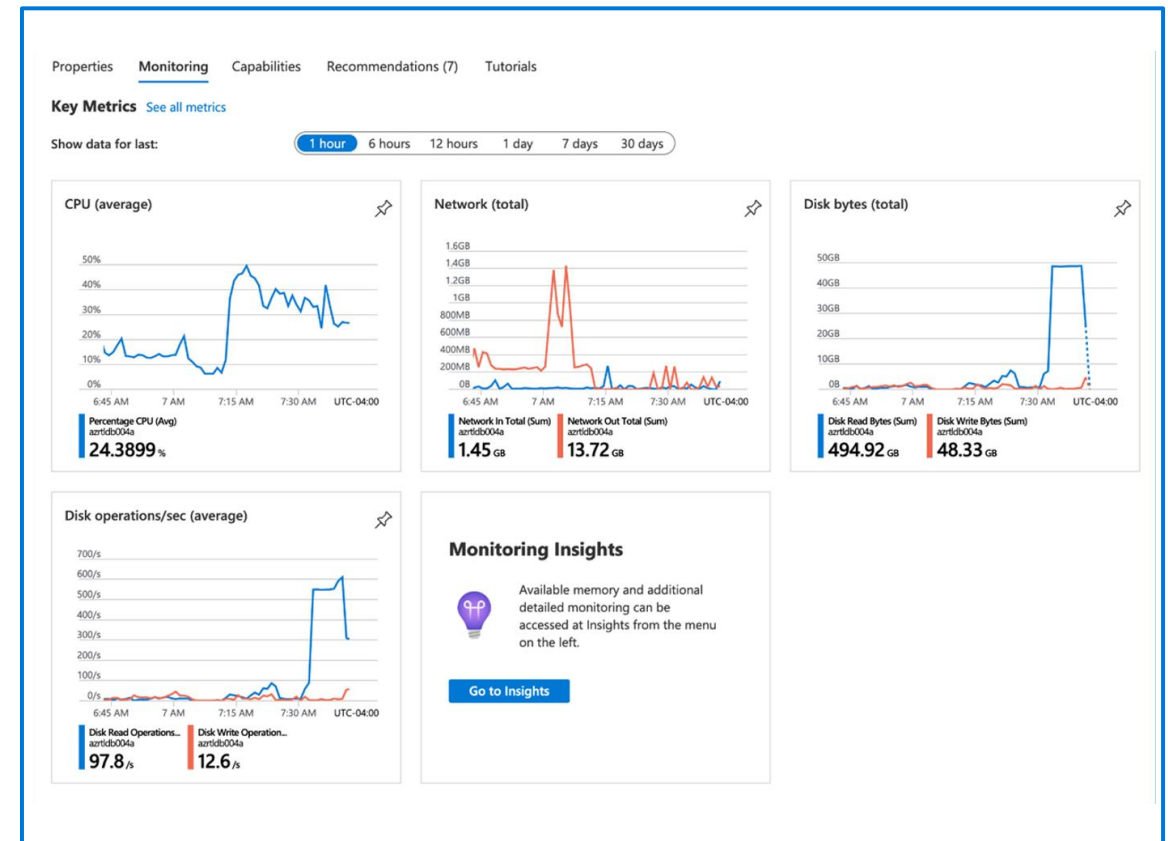
# Performance monitoring tools – Azure Monitor

All Azure resources collect a set of metrics through the Azure Monitor service

Enhanced data may be collected through the Azure Monitoring Insights for virtual machine resources

Data is stored in Azure Log Analytics

Metrics available to monitor will vary depending on the type of resource





# Review of Azure metric alerts

---

## Overview

Azure has a built-in alerting system that allows you to create alerts

---

The alerts can be configured:

- In a static manner
  - In a dynamic fashion using Dynamic Thresholds
- 

## Creating metric alerts

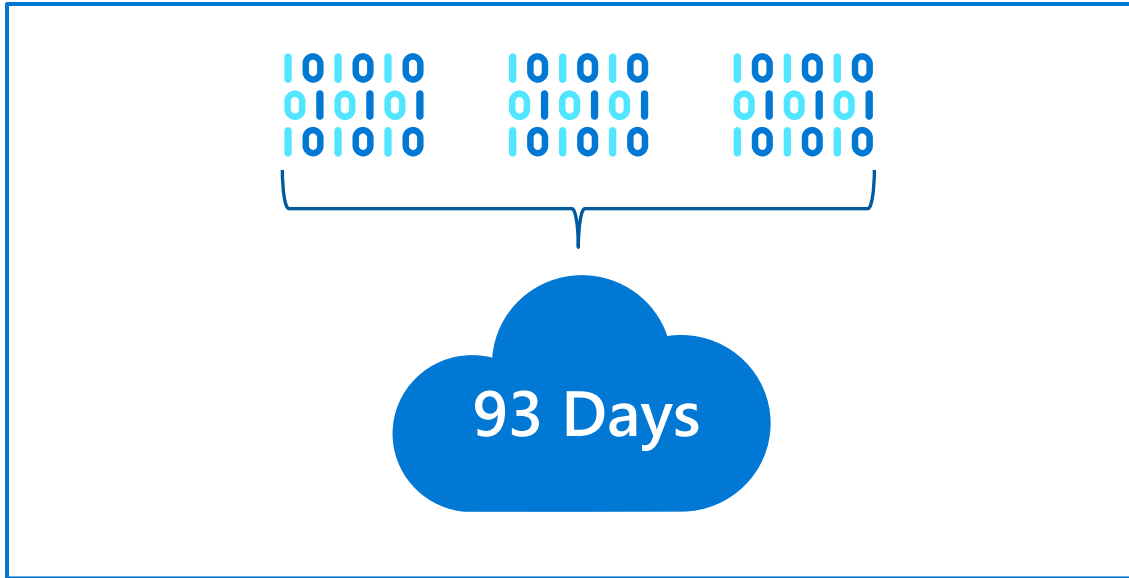
Metric alerts can be defined on IaaS or PaaS resources

---

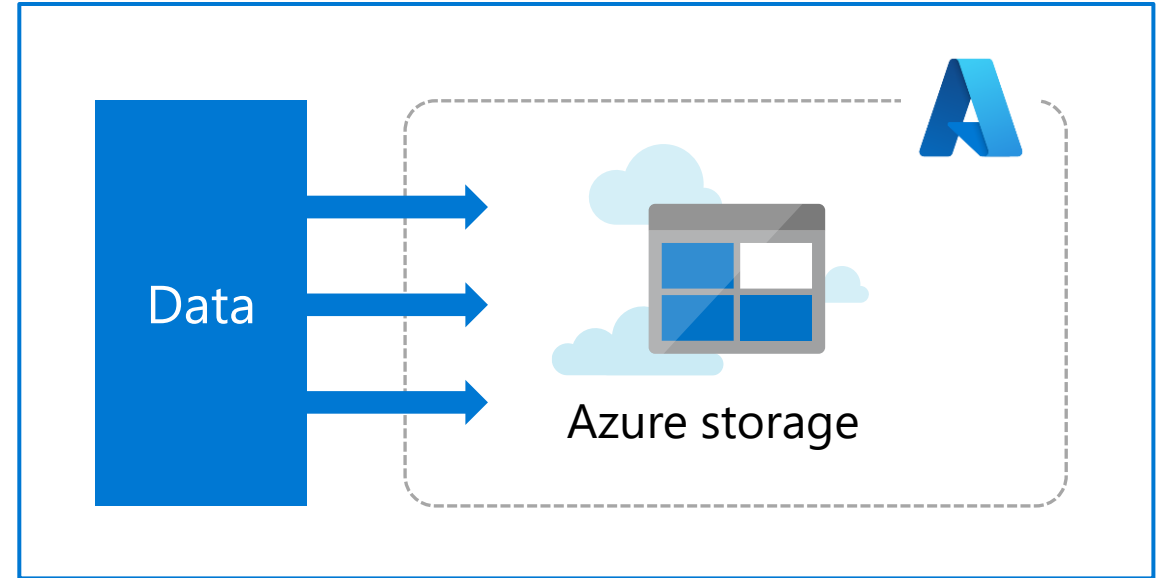
Alerts can be scoped by:

- VM
  - Subscription
  - Resource Group
-

# Reviewing performance data history



Azure Monitor stores data from **93 days**, which covers three months of data



You may archive data to Azure storage for historical analysis

# SQL Server metrics that matter

---

Processor(\_Total)% Processor Time

Paging File(\_Total)% Usage

PhysicalDisk(\_Total)\Avg. Disk sec/Read and  
Avg. Disk sec/Write

System\Processor Queue Length

SQLServer:Buffer Manager\Page life expectancy

SQLServer:SQL Statistics\Batch Requests/sec

SQLServer:SQL Statistics\SQL Compilations/sec and SQL Re-Compilations/sec

# What are Extended Events?

A lightweight and extensive diagnostic system that is built into the SQL Server engine

- ▶ Extended Events are supported on all the Azure SQL platforms
- ▶ Event sessions can be used to trace activity within the database which can help you troubleshoot issues like:
  - Blocking and deadlocking performance issues
  - Identify long-running queries
  - Observing Memory Pressure in your database
- ▶ The Extended Event framework allows you to filter your data collection

# What can I monitor with Extended Events?

---

- Extended events cover all aspects of the database engine. Specifically they are divided into four channels:

## **Admin**

Events targeted for end users and administrators, like a deadlock report

## **Operational**

Used for analysis and diagnosis of common problems, like an Availability Group failover

## **Analytical**

Commonly used for tracing query execution

## **Debug**

Used in conjunction with Microsoft support. May not be fully documented

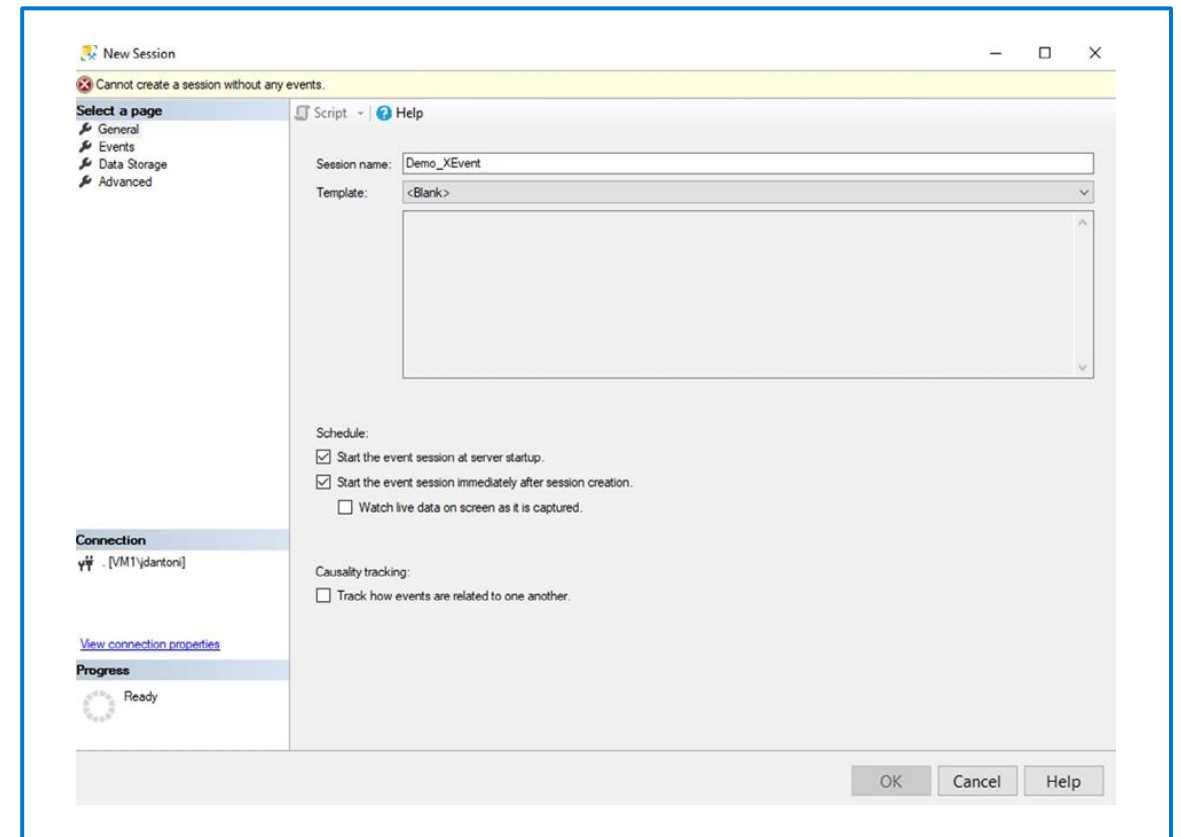
- Events can be groups together into sessions for easier monitoring

# How to create an Extended Events session

SSMS has a wizard to create an Extended Events session

SSMS also includes predefined templates for monitoring specific resources

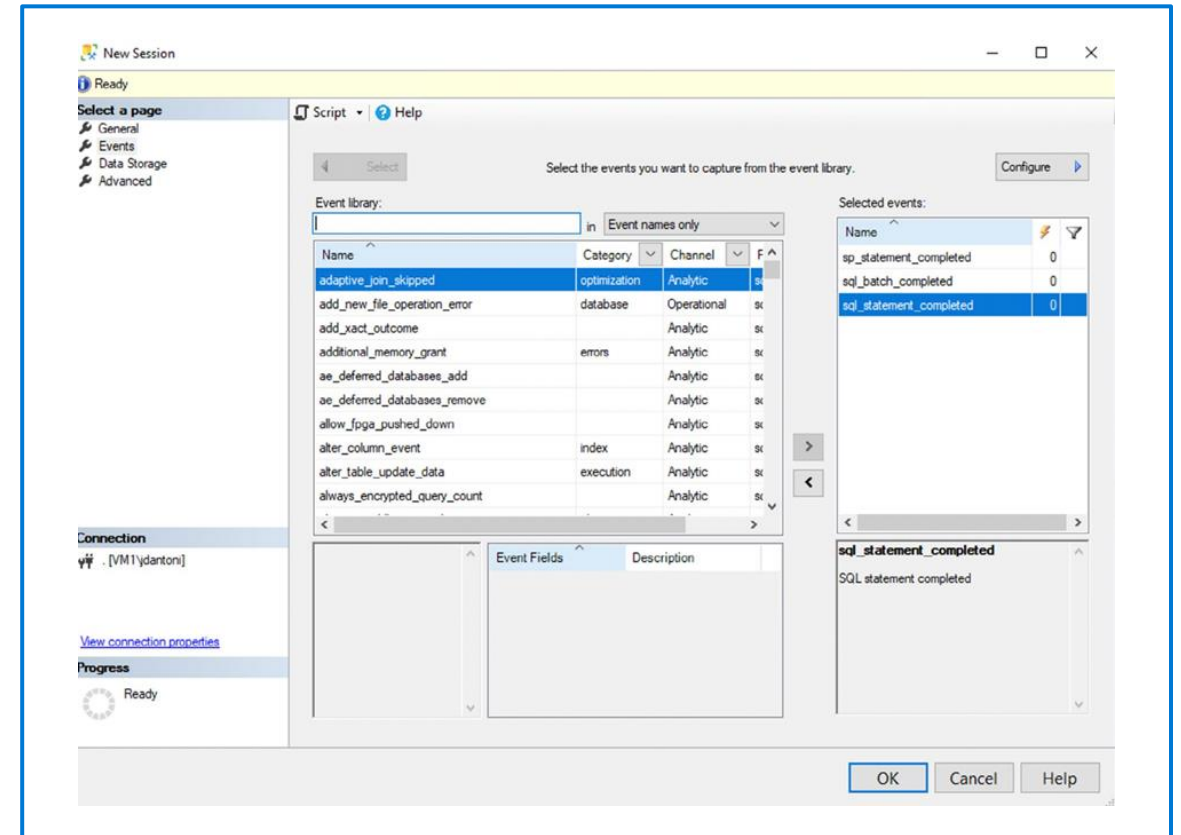
You can also use T-SQL which is more common for advanced administrators



# Extended Events components

Event Sessions consists of events and targets

Events represent the engine action you are capturing and include fields that can both be filtered and select individually



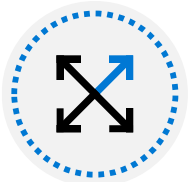
# Extended Events storage

---



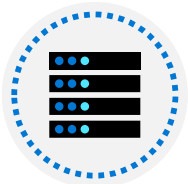
Events are persisted to disk for further analysis

---



There are multiple targets depending on the nature of your data collection

---



The most commonly used storage target is the event file, which writes the file to a local disk

---



You can also use the ring buffer, a non-persistent area of memory to perform real-time analysis of an event session



# Extended Events with T-SQL

```
CREATE EVENT SESSION test_session
ON SERVER
    ADD EVENT sqllos.async_io_requested,
    ADD EVENT sqlserver.lock_acquired
    ADD TARGET package0.etw_classic_sync_target
    (SET default_etw_session_logfile_path =
    N'C:\demo\traces\sqletw.etl' )
    WITH (MAX_MEMORY=4MB, MAX_EVENT_SIZE=4MB);
GO
```

# Azure SQL Insights

1

Monitoring profile can include SQL Database, SQL Managed Instance, and SQL Server on Azure VMs

2

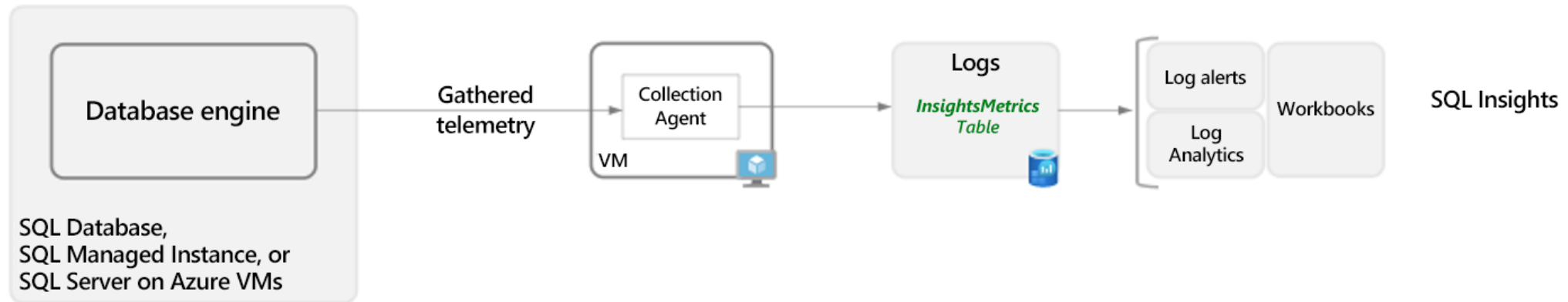
Collection agent is installed on a dedicated virtual machine that will monitor and remotely collect data from your SQL servers

3

Insight metrics are stored in Azure Log Analytics

4

Access performance data from the SQL Insights workbook template, or directly from the monitoring logs



# Azure SQL Insights – Creating a new profile

The screenshot displays the Azure Monitor SQL Insights (Preview) interface. The left-hand navigation pane includes a search bar and a list of categories: Overview, Activity log, Alerts, Metrics, Logs, Service Health, Workbooks, Insights, and a list of resource types. The 'SQL (preview)' option under the Insights category is highlighted with a red box. The main content area is titled 'Monitor | SQL (preview)' and features a breadcrumb 'Home > Monitor'. Below the title bar, there are filters for 'Subscription' (set to 'Contoso Subscription'), 'Monitoring profile' (with a red warning icon), and 'Time range' (set to 'Last 4 hours'). The main heading is 'Onboarding to Azure Monitor SQL Insights (Preview)'. The text below explains that SQL insights offers a flexible canvas for telemetry collection, analysis, and rich custom visualization, and is available in preview for SQL Database, SQL Managed Instance, and SQL Server on Azure Virtual Machines. It also states that billing is based on data ingested into Log Analytics and data retention settings. Two links are provided: 'Learn more about SQL performance monitoring' and 'Learn more about pricing'. A blue button labeled 'Create new profile' is highlighted with a red box. At the bottom, a preview of the 'Azure SQL Database' monitoring view is shown, including a 'Time range' selector and a 'Replica' dropdown.

Home > Monitor

Monitor | SQL (preview)

Search (Ctrl+ /)

Overview  
Activity log  
Alerts  
Metrics  
Logs  
Service Health  
Workbooks

Insights

Applications  
Virtual Machines  
Storage accounts  
Containers  
Networks  
**SQL (preview)**  
Azure Cosmos DB

Subscription: Contoso Subscription | Monitoring profile: [Warning] | Time range: Last 4 hours

### Onboarding to Azure Monitor SQL Insights (Preview)

SQL insights offers you a flexible canvas for telemetry collection, analysis, and rich custom visualization. You can customize telemetry collection and frequency, and also combine data from multiple sources into a single monitoring experience for your entire SQL estate. SQL insights is enterprise-ready and easily scales to meet your resource requirements. This feature is available in preview for SQL Database, SQL Managed Instance and SQL Server on Azure Virtual Machines.

You will be billed based on the amount of data ingested into Log Analytics and your data retention settings.

[Learn more about SQL performance monitoring](#)

[Learn more about pricing](#)

**Create new profile**

Home > Monitor > Azure SQL Databases > Azure SQL Database

Time range: [Selector] | Replica: [Dropdown]

# Azure SQL Insights – Creating a new profile cont'd

## Create new profile

Azure Monitor

Auto refresh: Off

### Enable SQL monitoring

#### 1. Create monitoring profile

A profile allows you to group servers, instances or databases to monitor and analyze as a combined set. It allows you to set the scope of monitoring - whether it is collection environments (development or production), application (billing or customer), the collection settings (e.g. high fidelity data collection vs. low), etc.

**Subscription**

Contoso Subscription

**Resource group**

myresourcegroup

**Profile name**

SQLservers-profile

Profile will be created in the same location where the Log Analytics workspace is.

#### 2. Set destination

Specify the Log Analytics workspace to send the SQL monitoring data to.

**Workspace subscription**

Contoso Subscription

**Log Analytics workspace**

DefaultWorkspace-xxxxxx-xxxx-xxxx-xxxxcd01efa-EUS

#### 3. Collection settings

Configure SQL monitoring data collection for your profile.

**Common settings** Customized collection (advanced)

Use this section to configure common collection settings. The default settings cover the majority of monitoring scenarios and usually does not need to be changed.

**Collection interval**

60 seconds

**Azure SQL Database settings**

7 selected

**Azure SQL Managed Instance settings**

7 selected

**SQL Server settings**

7 selected

#### 4. Create monitoring profile

Use the button below to create your monitoring profile

Create monitoring profile

# Azure SQL Insights limitations

---

**Limited support** or  
**No support** for the  
following components:

- Non-Azure instances
- Azure SQL Database elastic pools
- Azure SQL Database running on Basic, S0, S1, and S2 service tiers
- Azure SQL Database serverless tier
- Multiple secondary replicas
- Authentication with Azure Active Directory. Only SQL authentication is supported

# Query Performance Insights

---

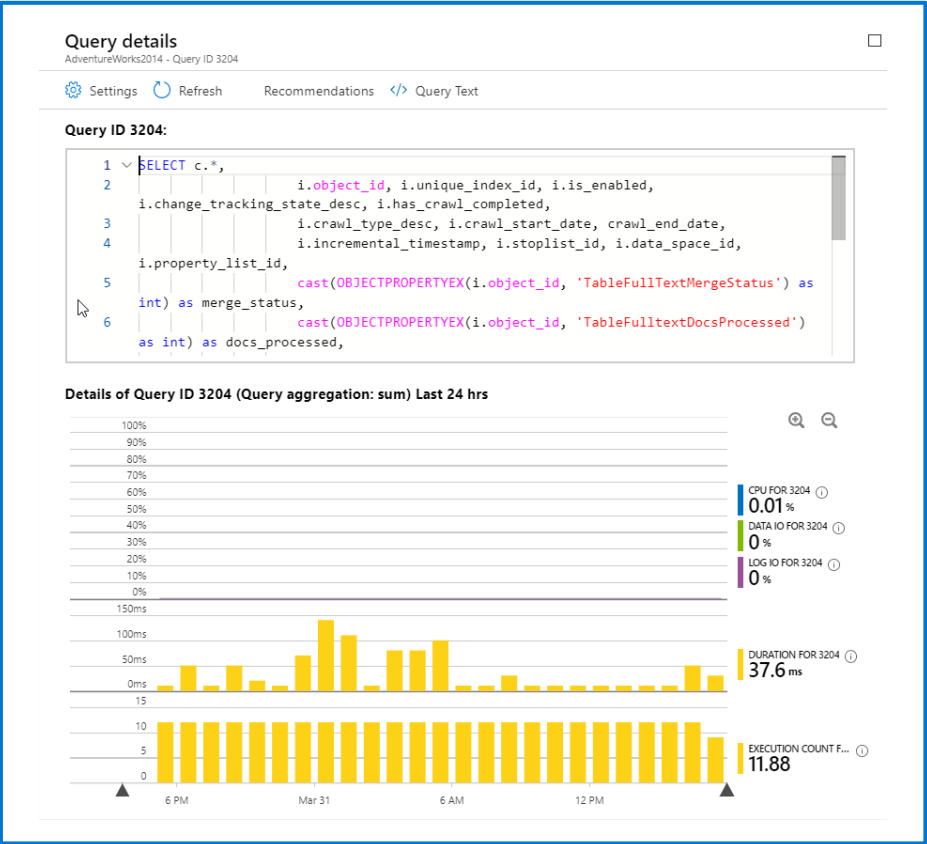
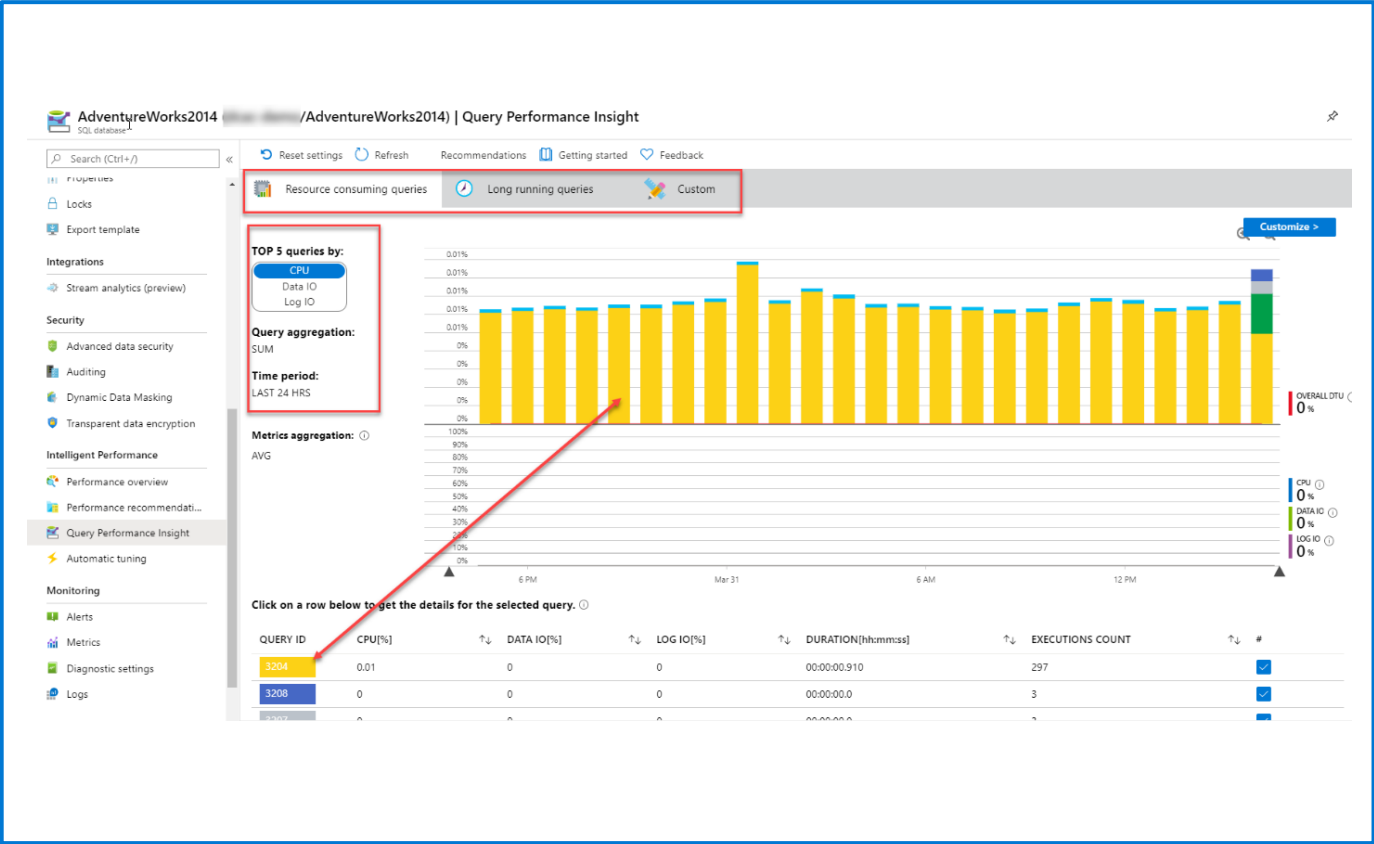
Query Performance  
Insight  
Allows you...

Allows DBAs to quickly identify expensive queries

You can drill into individual queries

You can combine the Query Performance Insight dashboard with Query Store data to quickly identify the most expensive execution plan

# Query Performance Insights



# Instructor led labs: Isolate performance problems through monitoring

---

Review CPU utilization in Azure portal  
Identify high CPU queries



The background is a dark navy blue. It features a series of faint, embossed rectangular shapes of varying sizes, some of which are slightly offset from each other, creating a sense of depth. A prominent glowing line starts as a small white circle with a soft halo in the upper right quadrant. This line extends horizontally to the left, then turns 90 degrees downward, and finally turns 90 degrees to the left again, ending in a vibrant purple color at the bottom right. The line has a subtle gradient and a soft glow around it.

Configure SQL Server resources for  
optimal performance

# Objectives



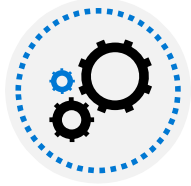
Understand your options for configuration of Azure Storage



Learn how to configure TempDB data files in SQL Server



Learn how to choose the right type of VM for SQL Server workloads



Understand the use cases and configuration of Resource Governor in SQL Server

# Azure Storage

---



Azure Storage is a scalable, secure storage platform that meets a variety of application data storage needs beyond just VM storage.

SQL Server can use three types of Azure Storage:

## **Blob Storage**

Can be used for database backups

## **File Storage**

File shares that can be used for Failover Cluster Instances

## **Disk Storage**

Managed block storage that is used to provide storage to VMs

# Azure Managed Disks

---

- Azure VMs use Managed Disks for their storage
- SQL Server VMs should use either Premium SSD or Ultra Disk for optimal performance

Azure Managed Disks are offered in four types:

## Ultra Disk

Support high-IO workloads for mission critical databases with extremely low latency

## Premium SSD

High-throughput and low latency and can meet the needs of most database workloads running in the cloud

## Standard SSD

Designed for lightly used dev/test workloads or web servers that do a small amount of IO and require predictable latency

## Standard HDD

Suitable for backups and file storage that is infrequently accessed

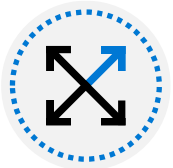
# Striping disks for maximum throughput

---



Disks are striped with no redundancy at the operating system level

---



This allows for you volumes to have the sum of IOPS and data size

---



For **Premium SSDs** it can be beneficial to scale both IOPs and storage volume

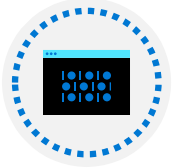
---



For **Ultra Disk**, you can scale IOPs, throughput, and maximum size independently on a single disk

# SQL Server storage best practices

---



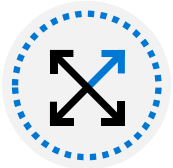
Create a separate volume for data and transaction log files

---



Enable read caching on the data file volume

---



Do not enable any caching on the log file volume

---



Use the D: drive (the locally attached SSD) for TempDB files

---



Enable instant file initialization to reduce the impact of file-growth activities

# Azure SQL VM resource provider

The SQL VM resource provider can configure storage based on your workload

You specify the number of IOPs and data volume your database requires and the provider will allocate the disks and configure them in the O/S

You can also configure TempDB on the D: drive

### Configure storage

Storage optimization ⓘ

GeneralTransactional processingData warehousing

**Data storage**

These disks will be attached to your virtual machine as data disks and will be stored in storage as page blobs.

Data drive location \* ⓘ

F:\data ✓

Disk type \* ⓘ

Premium SSD ▾

Disk type	Size (GiB)	Max IOPS	Max throughput	Number of disks
1024 GiB, Premium SSD (... ▾)	1024	5000	200	1

ⓘ 1024 GiB, 5000 IOPS, 200 MB/s

**Log storage**

Transaction logs are a critical component of the database as they record all transactions and database modifications made by each transaction.

Shared drive space \* ⓘ

Use a separate drive for lo... ▾

Log drive location \* ⓘ

G:\log ✓

Disk type \* ⓘ

Premium SSD ▾

Disk type	Size (GiB)	Max IOPS	Max throughput	Number of disks
1024 GiB, Premium SSD (... ▾)	1024	5000	200	1

ⓘ 1024 GiB, 5000 IOPS, 200 MB/s

**TempDb storage**

The tempDb system database is a global resource that is available to all users connected to the instance of SQL Server. It is used to store temporary user objects and internal objects created by the database engine.

Shared drive space \* ⓘ

Use local SSD drive ▾

TempDb drive location \* ⓘ

D:\tempDb

# TempDB configuration

SQL Server uses TempDB for many internal operations



The I/O performance of TempDB can be critical to the performance of the SQL Server



TempDB should have the same number of data files as the server has CPU cores, up to 8



The data files should all be the same size and have the same autogrowth settings



# Resource Governor

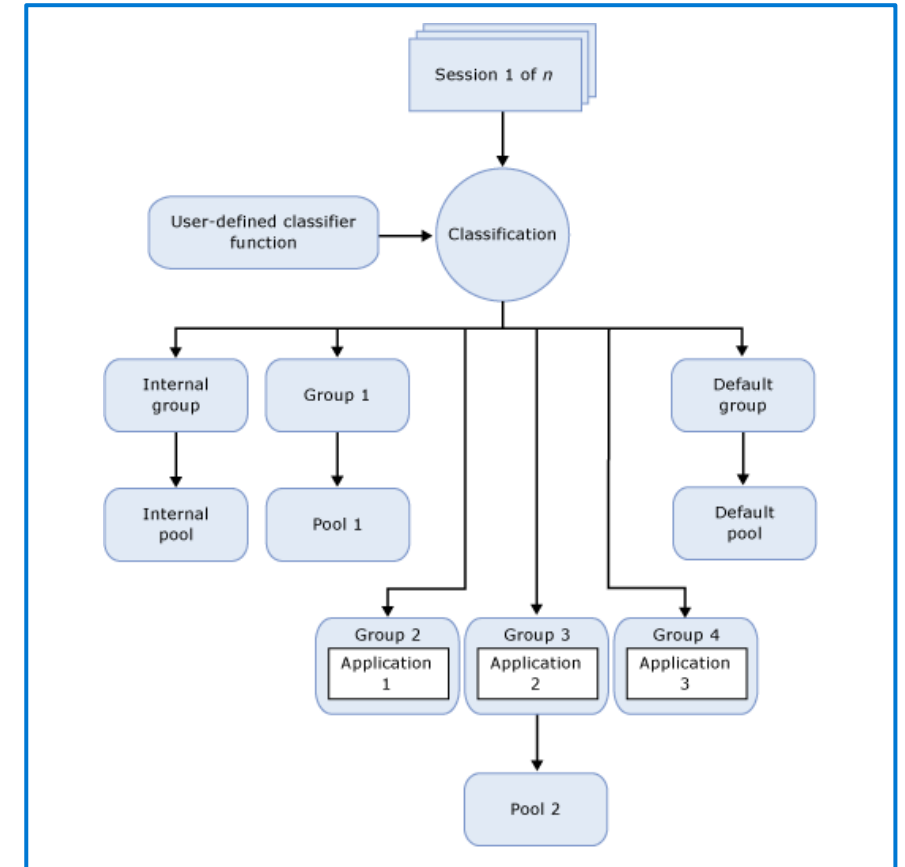
- Allows to balance conflicting workloads that have different resource demands at different times
- Workloads are divided into resource pools which allow you to limit CPU, memory, and IOPs for user sessions
- Workloads are classified using a classifier function

## Resource Governor components:

Resource pool

Workload group

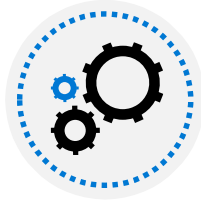
Classification



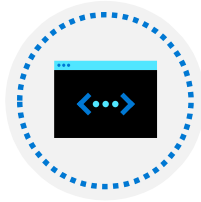
Configure databases for optimal  
performance



# Objectives



Understand database scoped configuration options



Understand the features of Intelligent Query Processing (IQP)



Explore the automatic tuning feature in Azure



Understand related maintenance tasks related to indexing and statistics

# SQL Server maintenance activities

---

Databases need regular maintenance activities. Some common examples of these activities include:



**Database  
Backups**

**Database  
Consistency  
Checks**

**Index  
Maintenance**

**Statistics  
Updates**

# Fragmentation in data files

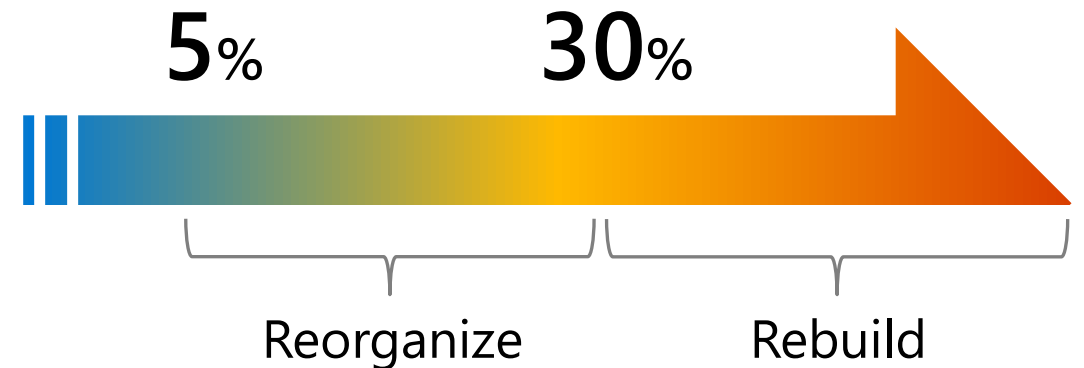
---

- ▶ Fragmentation in database files can occur within indexes or tables
- ▶ As data is inserted and updated, records can move into pages which no longer match the physical ordering of the data pages on the disk
- ▶ Fragmentation can degrade the performance of queries
- ▶ Fragmentation can be reduced by **reorganizing** or **rebuilding indexes**

# Index maintenance tasks

**Indexes should be regularly rebuilt and reorganized as they reach defined levels of fragmentation**

SQL Server and Azure SQL support resumable index maintenance as well as online index rebuilds to allow your workload to continue as maintenance operations occur



# Index maintenance tasks – REORGANIZE vs. REBUILD

## Reorganize

- Physically reorder the leaf-level index pages to match the logical sorted order of the leaf nodes
- Compacts the index pages based on the index's fill factor setting
- It is an **online** activity

```
ALTER INDEX ALL ON Production.Product  
REORGANIZE
```

## Rebuild

- Drops and recreates the pages of the index
- Causes the statistics to be updated
- Can be either online or offline
- When fragmentation is greater than **30%**

```
ALTER INDEX ALL ON Production.Product  
REBUILD
```

# Index maintenance tasks on SQL Server

---



Since you have the full surface area of Windows and SQL Server (including the SQL Server Agent) you can use the Agent to perform maintenance activities



If you are migrating from on-premises your maintenance activities should be the same as your on-premises workloads



You can use **SQL Agent** or **Task Scheduler** to create custom maintenance jobs



# Index maintenance tasks on Azure SQL Database

---



While the Azure platform performs activities like consistency checks and backups, index and statistics maintenance are still the responsibility of the DBA



Azure SQL Database does not have the SQL Server Agent, but there are other options for scheduling tasks:

- Azure Automation Runbooks
- Azure SQL Elastic Jobs
- SQL Agent Job from SQL Server in an Azure VM

# Index maintenance tasks on Azure SQL Managed Instance

---



Azure SQL Managed Instance includes the **SQL Agent** and the MSDB database



This means you can use scheduled tasks like on-premises servers or Azure VMs



Any tasks that need to access the file system will need to be modified as that is not supported

# Statistics maintenance tasks

---



The database engine uses statistics on columns and indexes to build execution plans

---



It is important that statistics be kept up to date for all objects

---



SQL Server defaults to having auto-update statistics on

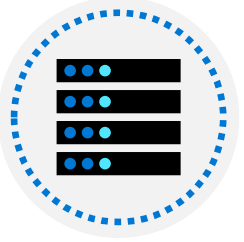
---



The database engine uses a sliding scale to determine the number of modifications required to kick off a statistics update

# Database scoped configuration options

---



Databases have two types of configuration options, which are defined based on how they are set:

- Database Scoped Configuration options are configured using ALTER DATABASE SCOPED CONFIGURATION syntax
  - Database options are configured using ALTER DATABASE SET syntax
- 



Options include recovery model, isolation level, automatic tuning, and the Intelligent Query Processing features

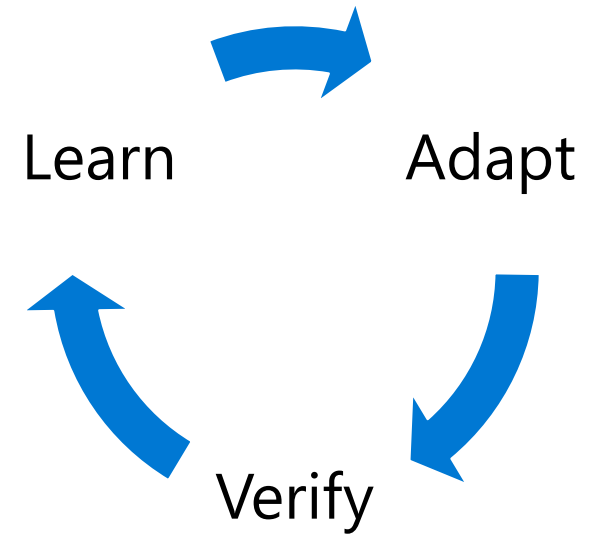
# Automatic tuning

---

SQL Server and Azure SQL Database have an automatic tuning features that uses the query store for a data source

This can be helpful for sudden unexpected query execution plan changes that cause regressions in performance


Automatic tuning will revert to the last known good execution plan after fifteen executions of a significantly regressed query




# Automatic index management

Azure SQL Database supports index analysis through automatic index management




This feature can add and remove indexes, in addition to forcing a better execution plan in the event of a regression

 Azure SQL Database built-in intelligence automatically tunes your databases to optimize performance. Click here to learn more about automatic tuning.

Inherit from: ⓘ  
Server Azure defaults Don't inherit

 The database is inheriting automatic tuning configuration from the server. You can set the configuration to be inherited by going to: [Server tuning settings](#)

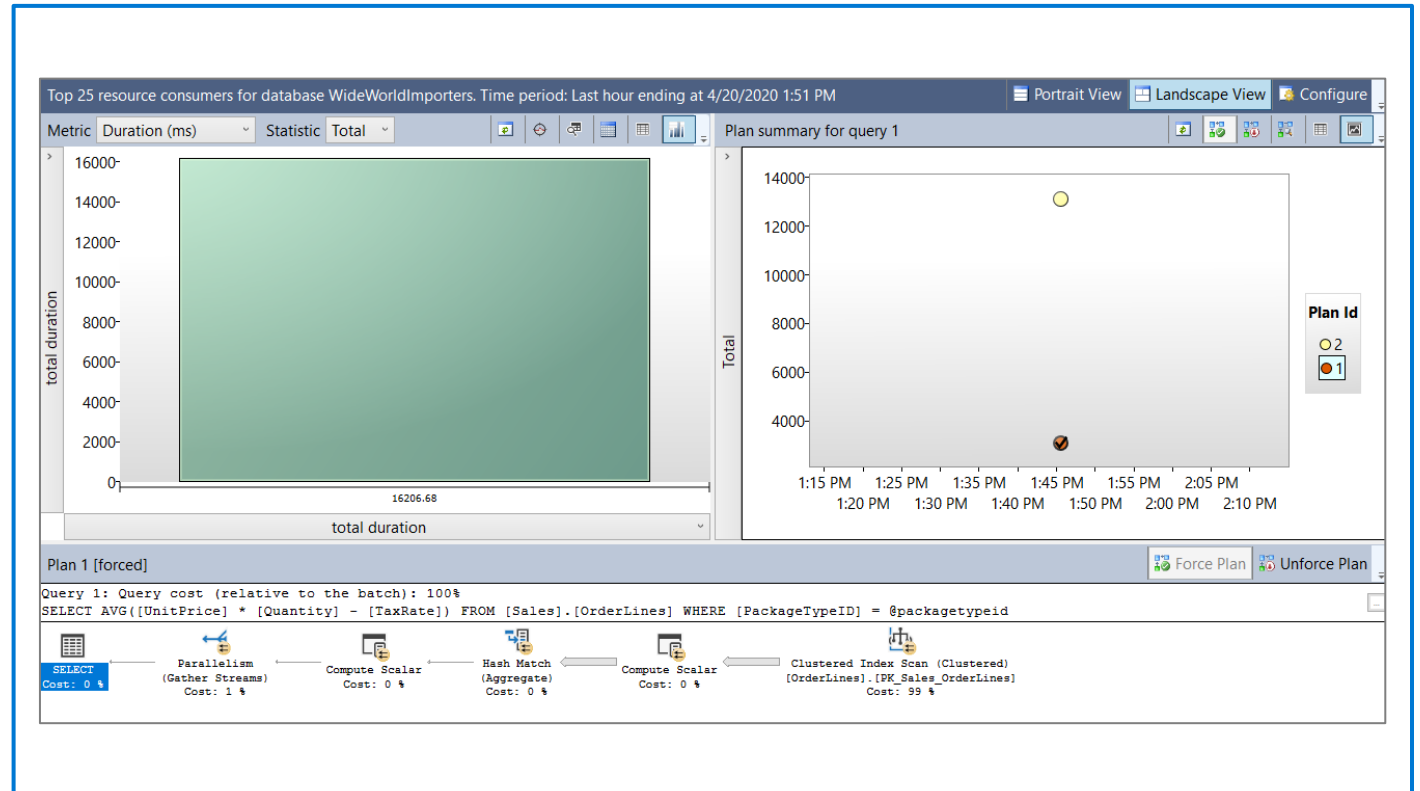
Configure the automatic tuning options ⓘ

Option	Desired state	Current state
 FORCE PLAN	<span>ON</span> <span>OFF</span> <span>INHERIT</span>	<b>ON</b> Inherited from server
 CREATE INDEX	<span>ON</span> <span>OFF</span> <span>INHERIT</span>	<b>OFF</b> Inherited from server
 DROP INDEX	<span>ON</span> <span>OFF</span> <span>INHERIT</span>	<b>OFF</b> Inherited from server

# Automatic plan correction

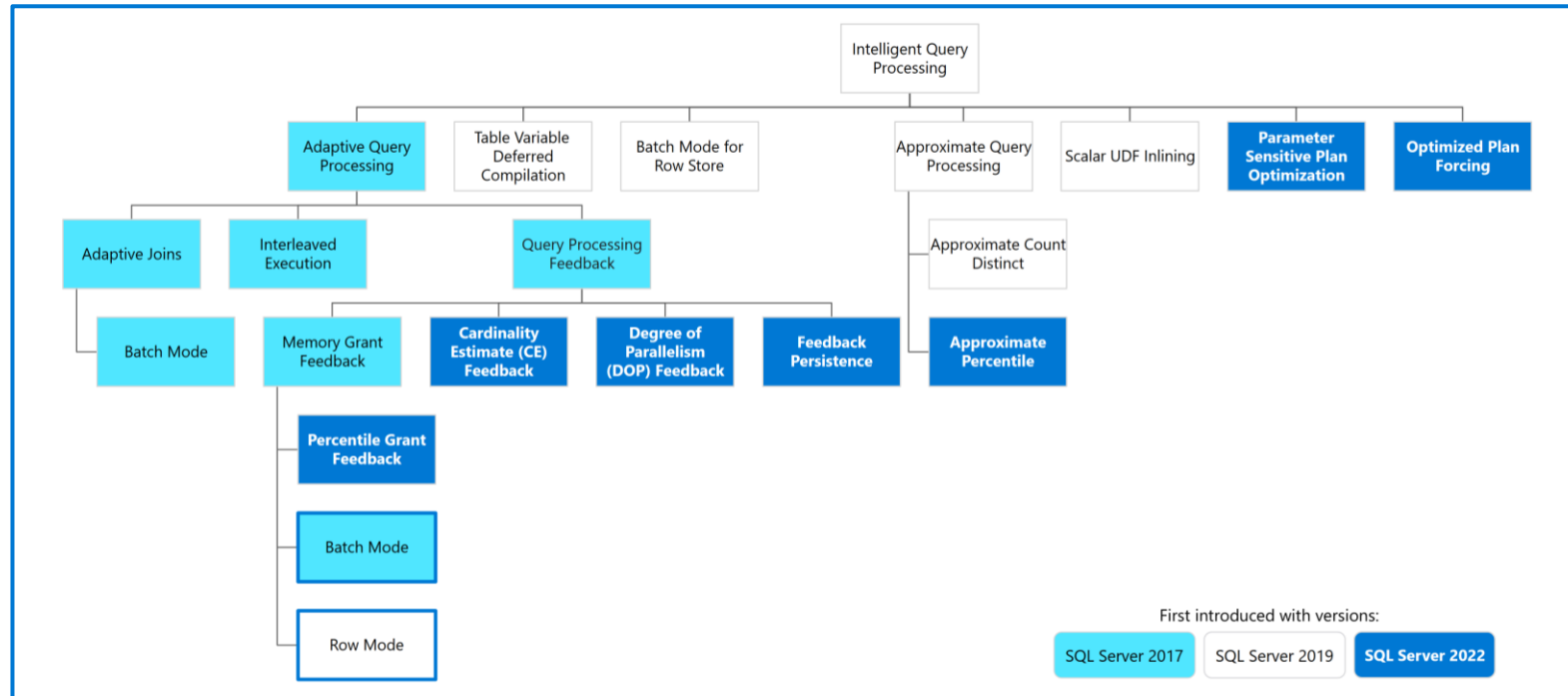
Identifies when query execution plans have regressed in their performance

```
ALTER DATABASE [<Database Name>] SET  
AUTOMATIC_TUNING (FORCE_LAST_GOOD_PLAN = ON);
```



# Intelligent Query Processing (IQP)

Intelligent query processing is a family of features that allows the database engine to make better choices to provide better overall performance.





# Instructor led labs: Detect and correct fragmentation issues

---

Investigate index fragmentation  
Rebuild fragmented indexes  
Validate performance improvements

# Summary

## **Describe performance monitoring:**

- Understanding methods to review potential performance issues
- Identify critical Azure metrics
- Explore Azure SQL Insights

## **Configure SQL Server resources for optimal performance:**

- Understand your options and configuration for Azure Storage
- Learn how to configure TempDB data files in SQL Server
- Understand Resource Governor in SQL Server

## **Configure databases for optimal performance:**

- Understand Database Scoped Configuration Options
- Understand related maintenance tasks related to indexing and statistics
- Obtain knowledge on the auto-tuning feature in Azure