Microsoft

Monitoring and Tuning Azure SQL Database

Module 5

Learning Units covered in this Module

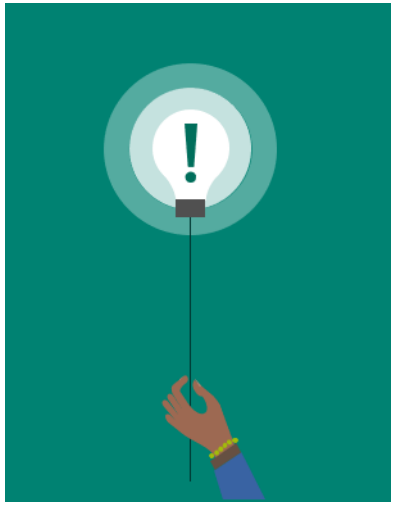
- Lesson 1: Monitoring and Troubleshooting Azure SQL Database
- Lesson 2: Monitoring Query Performance using Query Performance Insight
- Lesson 3: Azure SQL Database Tuning using Automatic Tuning
- Lesson 4: Monitoring Azure SQL Database Performance using Extended Events
- Lesson 5: Configure Alerts through Azure Portal

Lesson 1: Monitoring and Troubleshooting Azure SQL Database

Objectives

After completing this learning, you will be able to:

- Know the various options to monitor and troubleshoot the Azure SQL Database.



Common Issues on Azure SQL Database

Monitoring for Azure SQL Database is scoped at database level.

Here is list of most faced issues:

- Database Connectivity
- High DTU Percentage
- Query Timeouts
- Deadlocks
- Database Storage consumption
- Slow Queries

Tools to Monitor & Troubleshoot Issues

Query Performance Insight

Automatic Tuning

Intelligent Insights

Extended Events

Dynamic Management Views (DMVs)

Azure Database Portal Dashboard

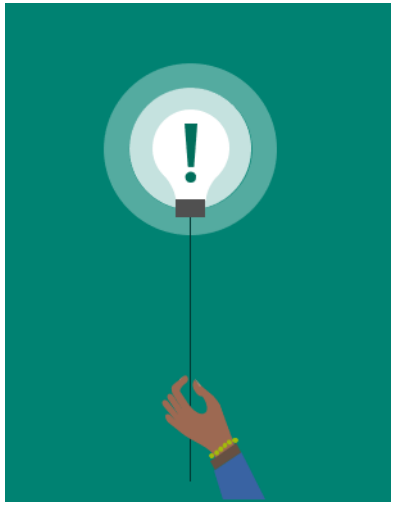
Questions?

Lesson 2: Monitoring Query Performance using Query Performance Insight

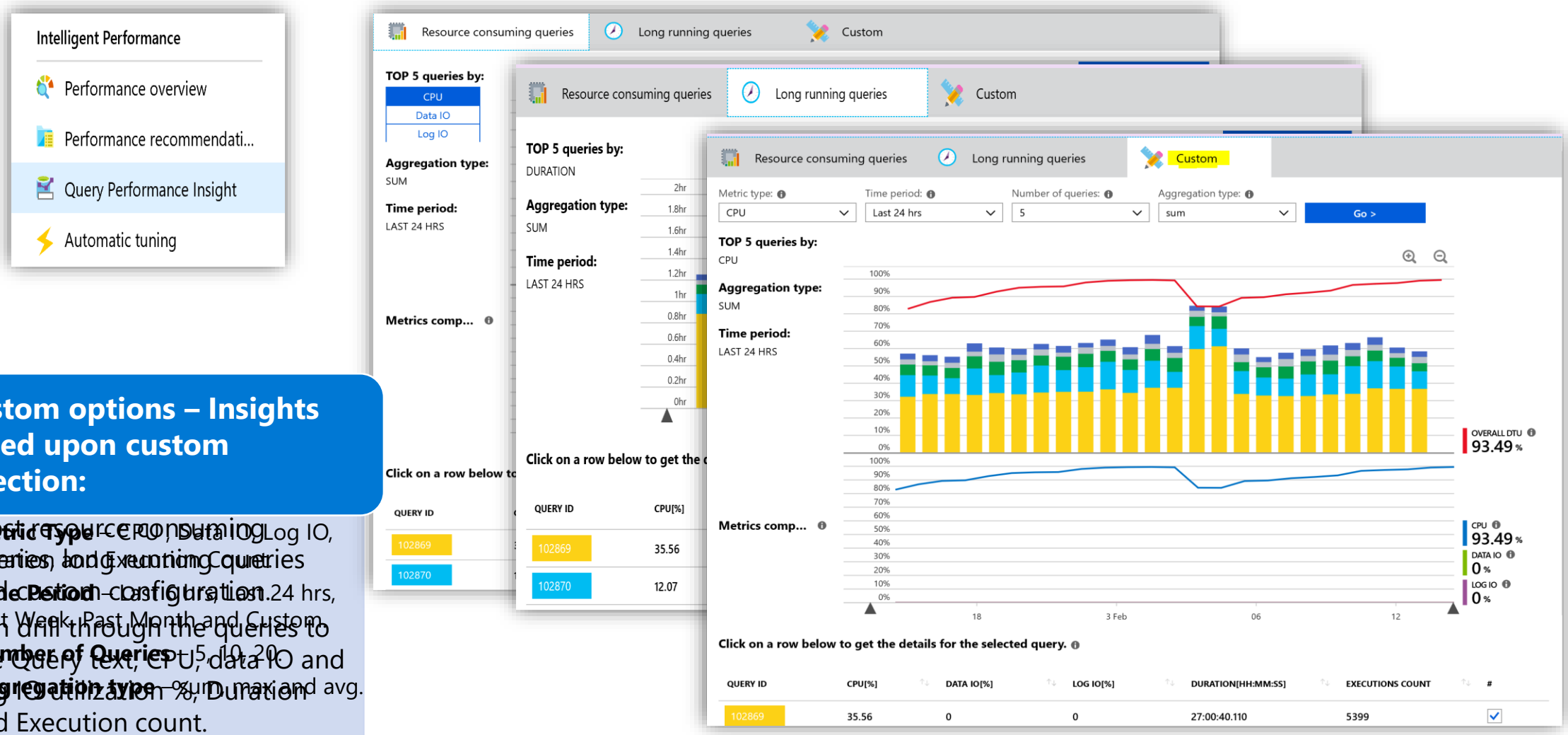
Objectives

After completing this learning, you will be able to:

- Know how to troubleshoot the performance of your queries by using Query Performance Insight.



Query Performance Insight



Custom options – Insights based upon custom selection:

- **Metric Type** – CPU, Data IO, Log IO, Duration and Execution Count
- **Time Period** – Last 6 hrs, Last 24 hrs, Past Week, Past Month and Custom
- **Number of Queries** – 5, 10, 20
- **Aggregation type** – sum, max and avg.

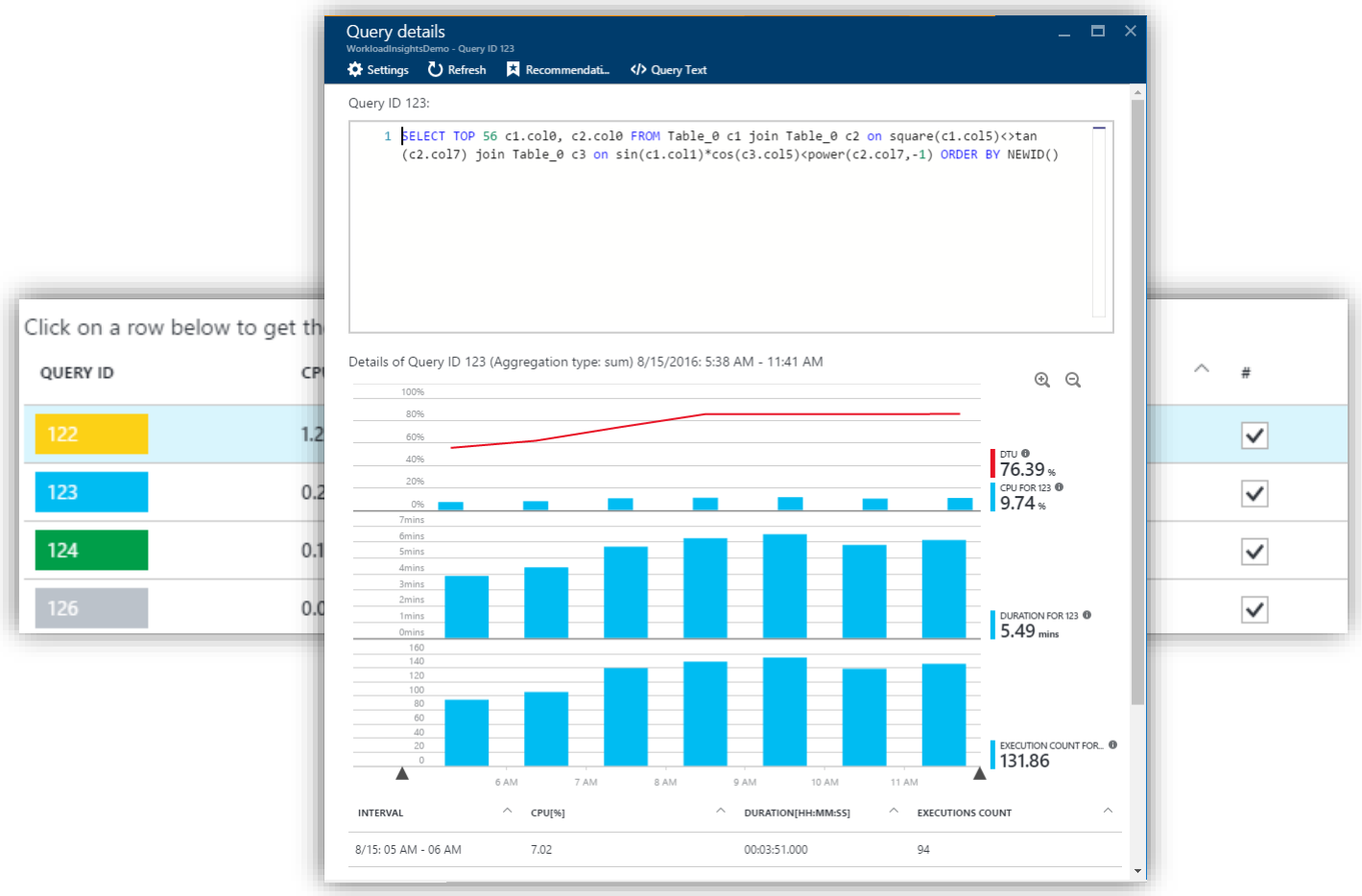
- Identify resource consuming queries and long running queries
- Custom configuration.
- Can drill through the queries to see Query text, CPU, data IO and Log IO utilization %, Duration and Execution count.

Viewing individual query details

Get details for the individual queries

- CPU Consumption
- Duration
- Execution Count

It does not capture DDL queries

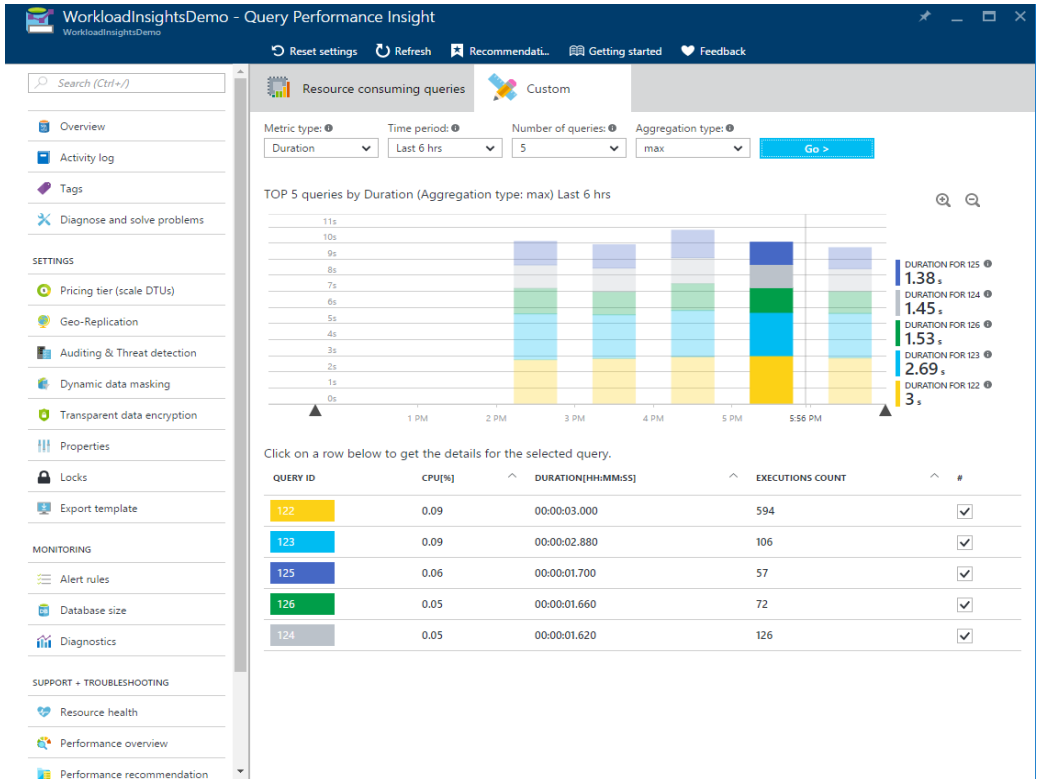


Review top queries per duration

Duration is one of the metrics showing potential bottleneck

Long-running queries has potential for:

- Longer locks
- Blocking other users
- Limiting scalability

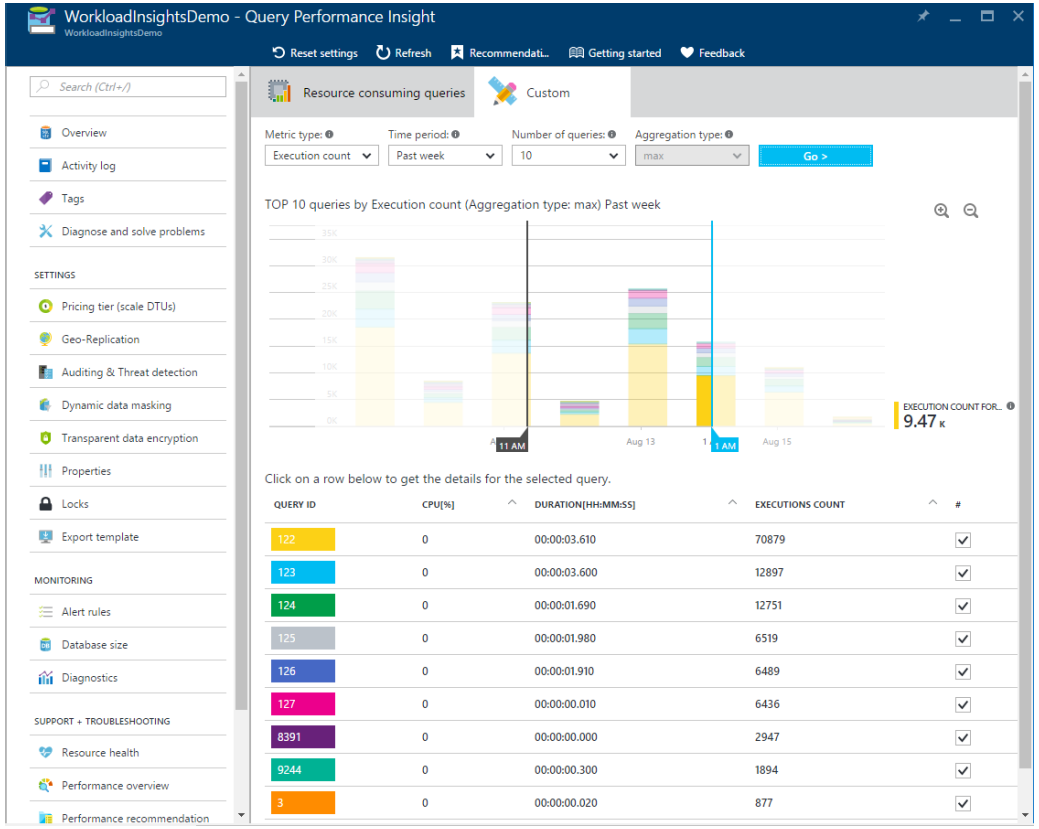


Review top queries per execution count

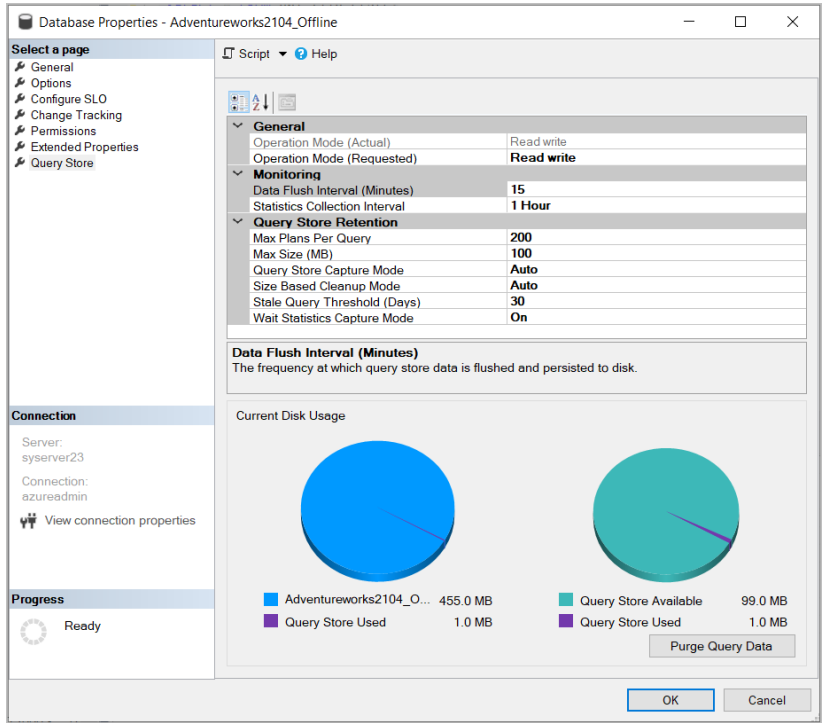
Execution count is one of the metrics showing potential bottleneck

High number og executions has potential for:

- Database performance
- Network latency
- Downstream server latency



Query Store



Retention Policy

- Size based – Auto cleanup when near max size.
- Time based – Default 30 days.
- Max Plans Per Query – Default 200.
- Wait Statistics Capture Mode – Default On.

Capture Policy

- All – Captures all queries.
- Auto – Infrequent queries are ignored.
- None – No queries are captured.
- Custom – Advanced Options

Demonstration

Query Performance Insight

- Analyze the Query Performance Insight output.

Monitoring Query Performance using Query Performance Insights

- Configure the Query Store.
- Analyze the Query Performance Insight.

Questions?

Knowledge Check

What feature should be enabled on your Azure SQL Database before you can use Query Performance Insight?

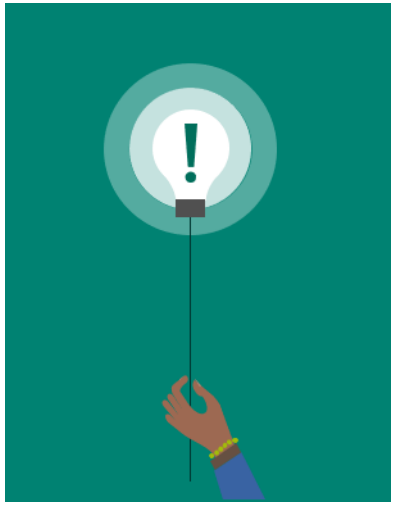
How can you view individual query details?

Lesson 3: Azure SQL Database Tuning using Automatic Tuning

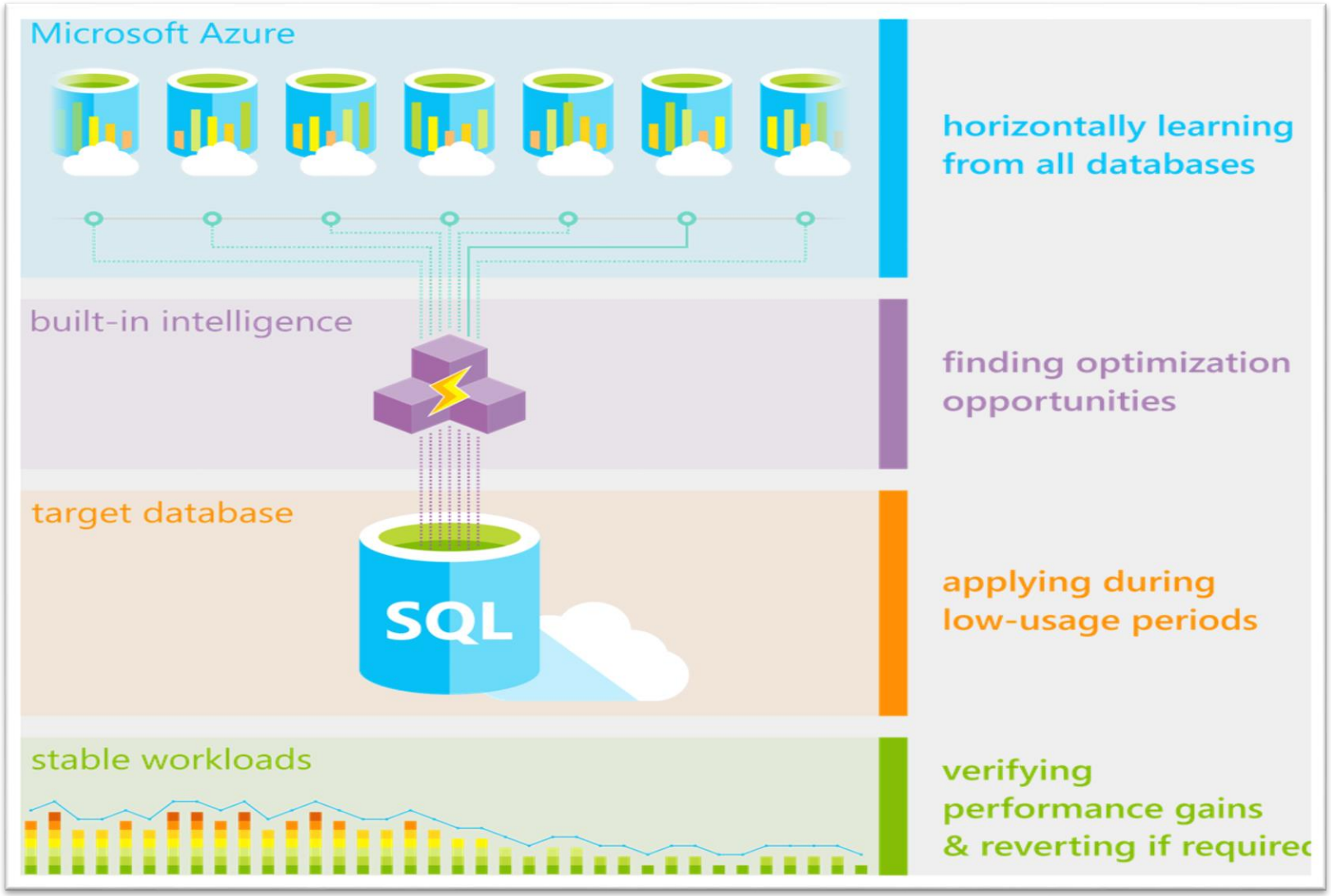
Objectives

After completing this learning, you will be able to:

- Know how Performance Recommendations can help to improve database performance.

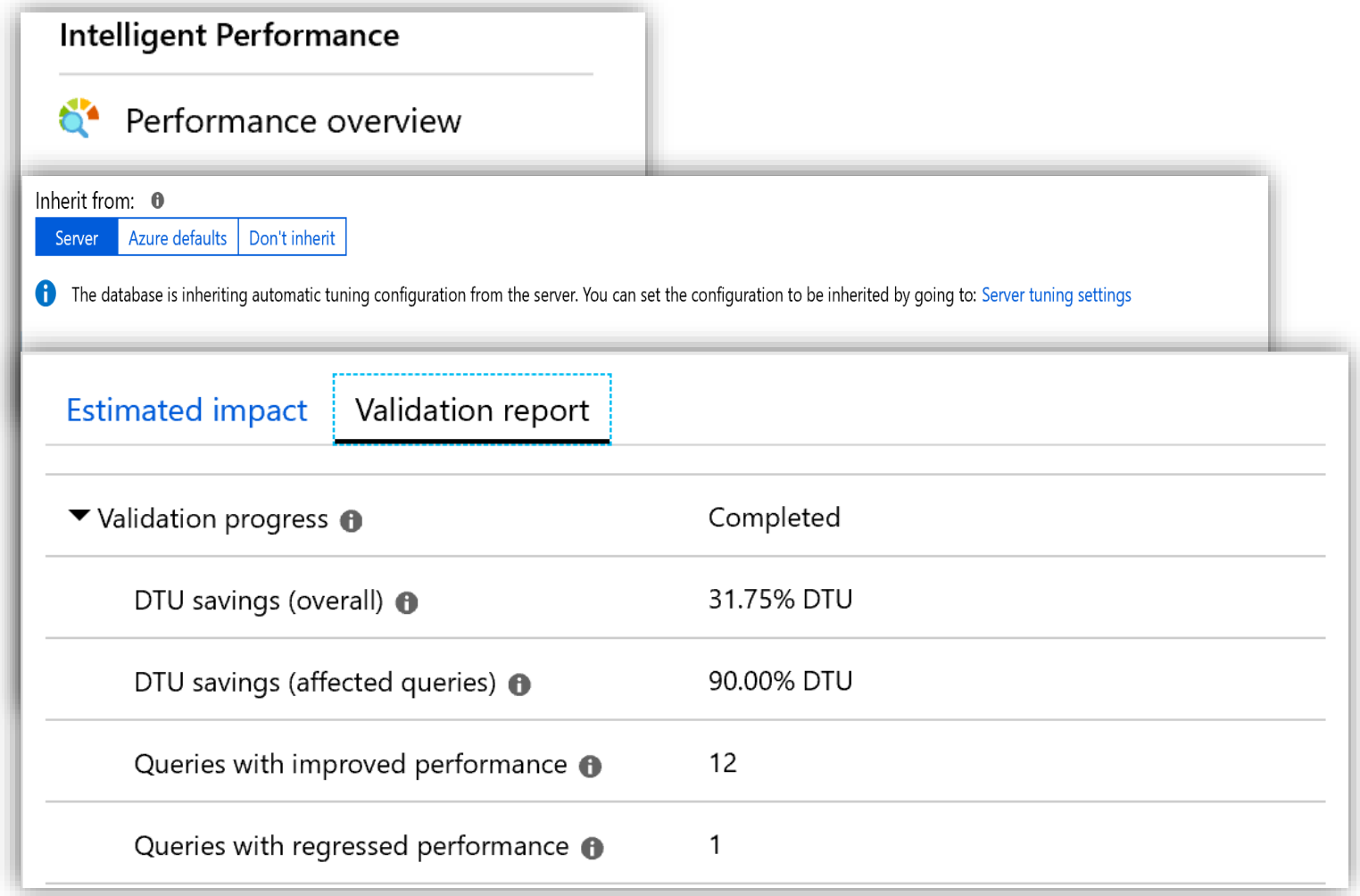


Automatic Tuning



[Performance recommendations for SQL Database]

Intelligent Performance – Automatic Tuning



http://automaticplancorrectiondemo.azurewebsites.net/index.html

Force Last Good Plan:

- Identifies regressed queries due to bad plan and replaces the bad plan with last Good Plan, validates performance improvements and reverts the change if performance does not improve.

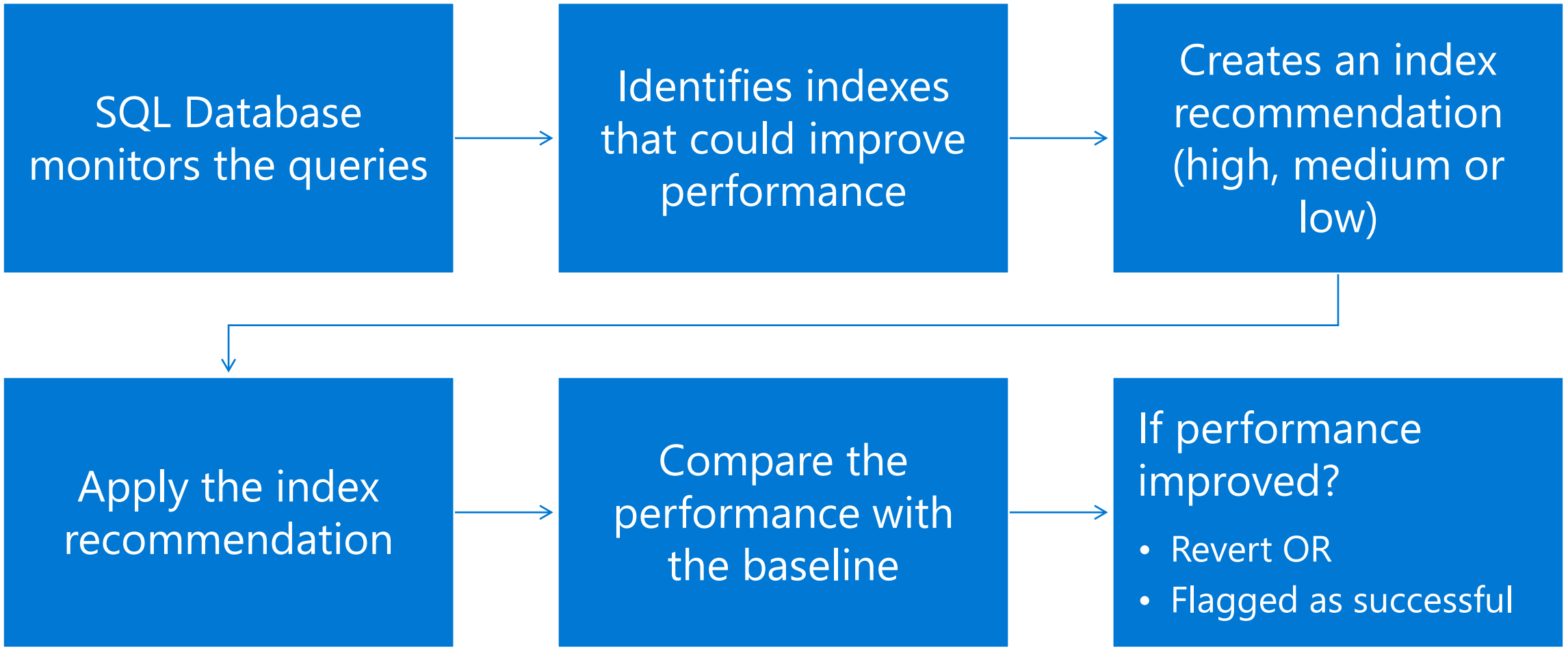
Create Index:

- Identifies and creates Indexes, validates performance improvements and reverts the change if performance degrades.

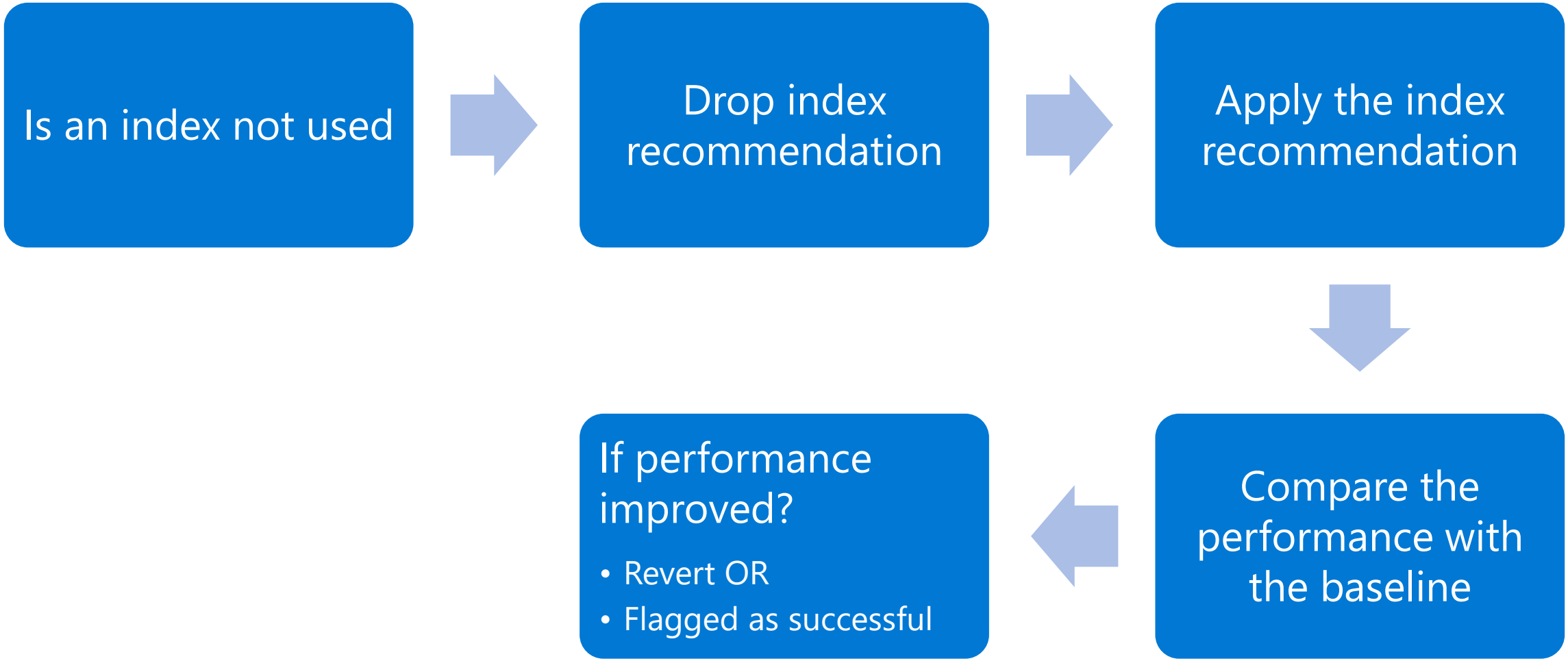
Drop Index:

- Identifies and drops unused Indexes, validates performance improvements and reverts the change if performance degrades.

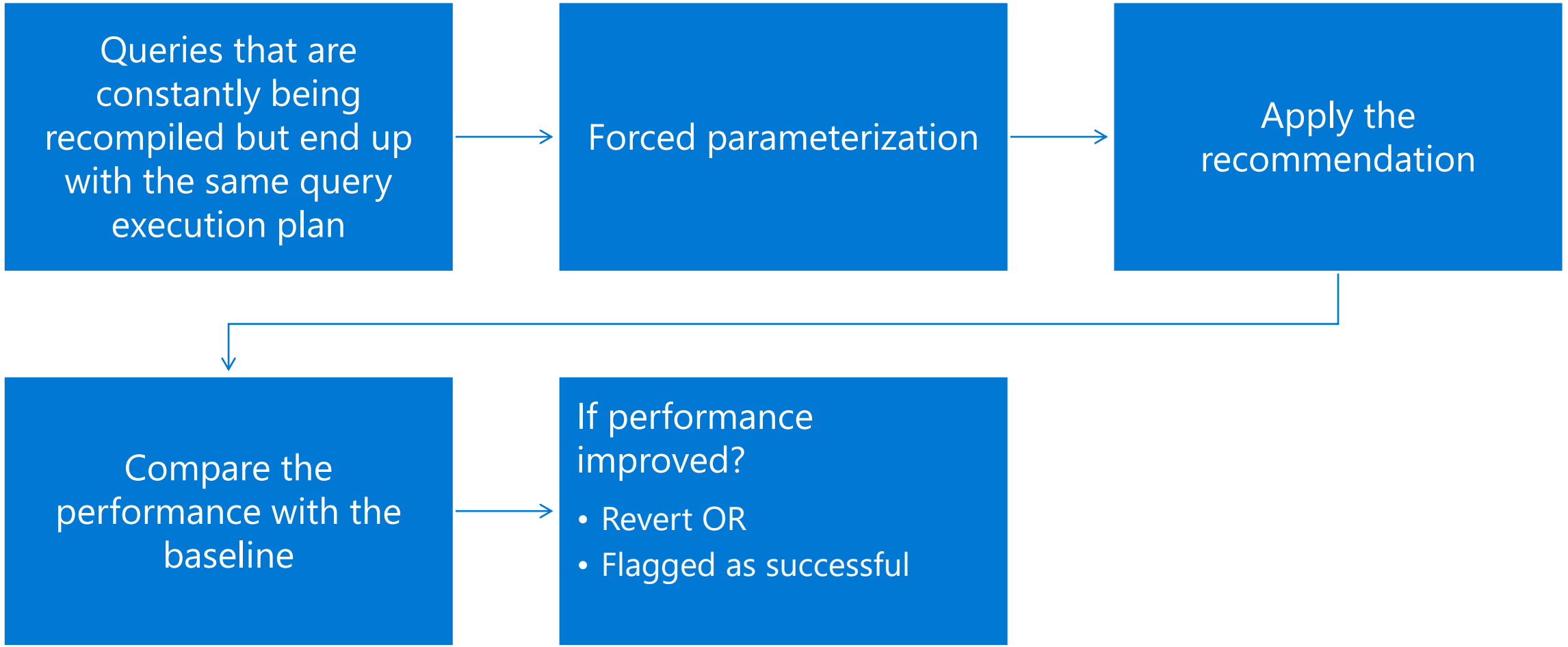
Automatic Tuning – Create Index



Automatic Tuning – Drop Index



Automatic Tuning – Parameterize Queries



Questions?

Knowledge Check

List three types of recommendations from Automatic Tuning.

What could be a reason to disable the automatic tuning option?

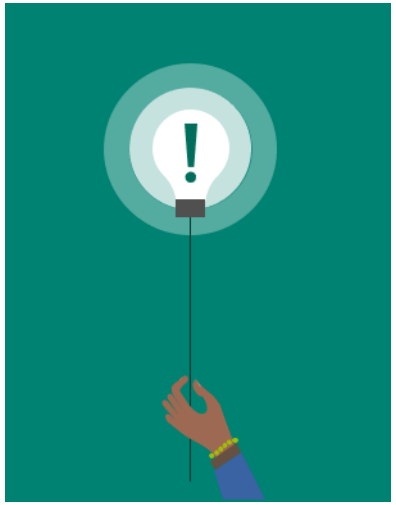
What technology is used for Automatic Tuning?

Lesson 4: Monitoring Azure SQL Database Performance using Extended Events

Objectives

After completing this learning, you will be able to:

- Use extended events for troubleshooting performance issues.



Why XEvent?

SQLTrace and SQLProfiler are deprecated

Faster and scalable

Designed not to cause server problems

More events than SQLTrace ever had

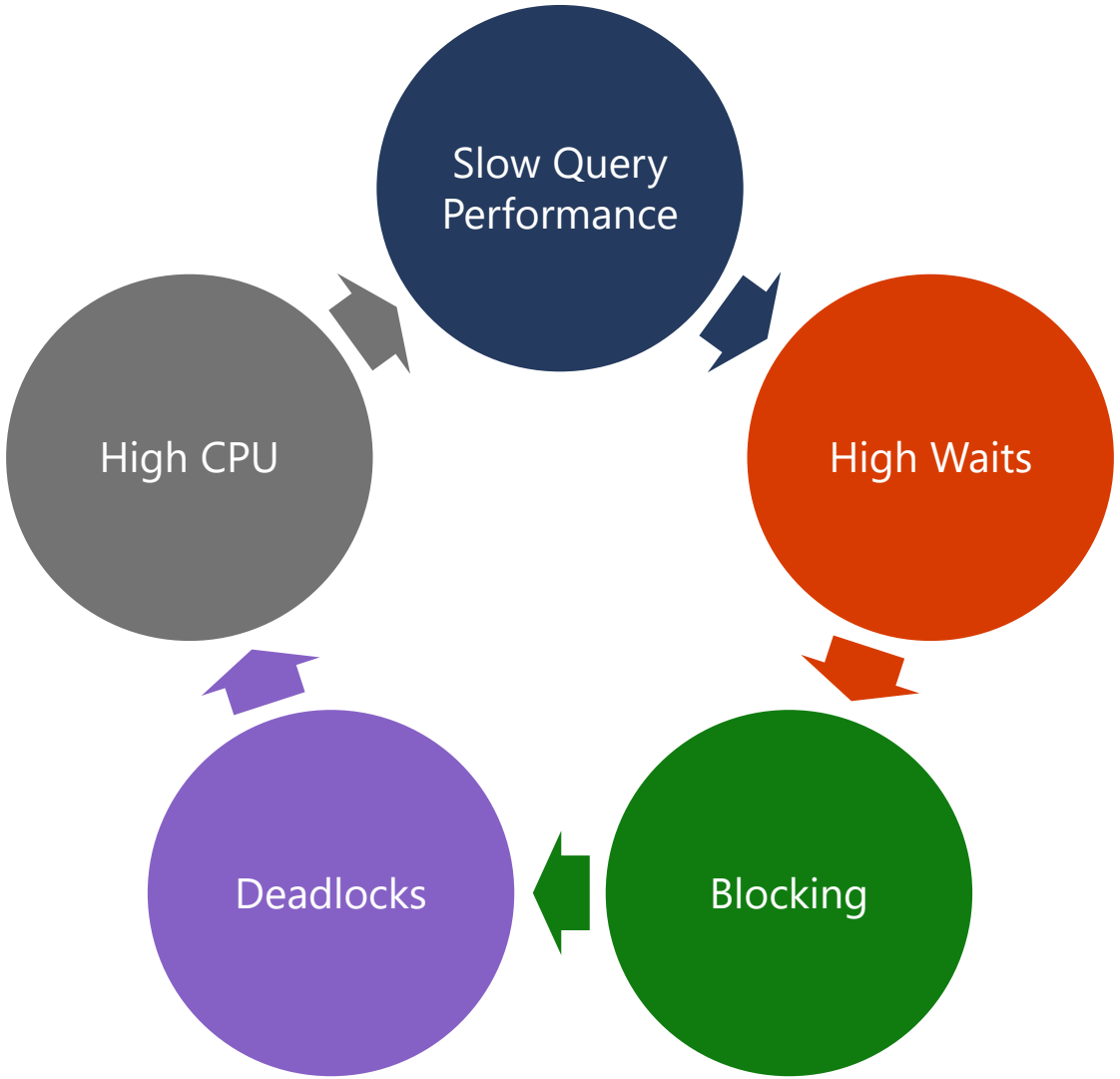
Targets and actions make it powerful

SSMS includes the basic UI tooling

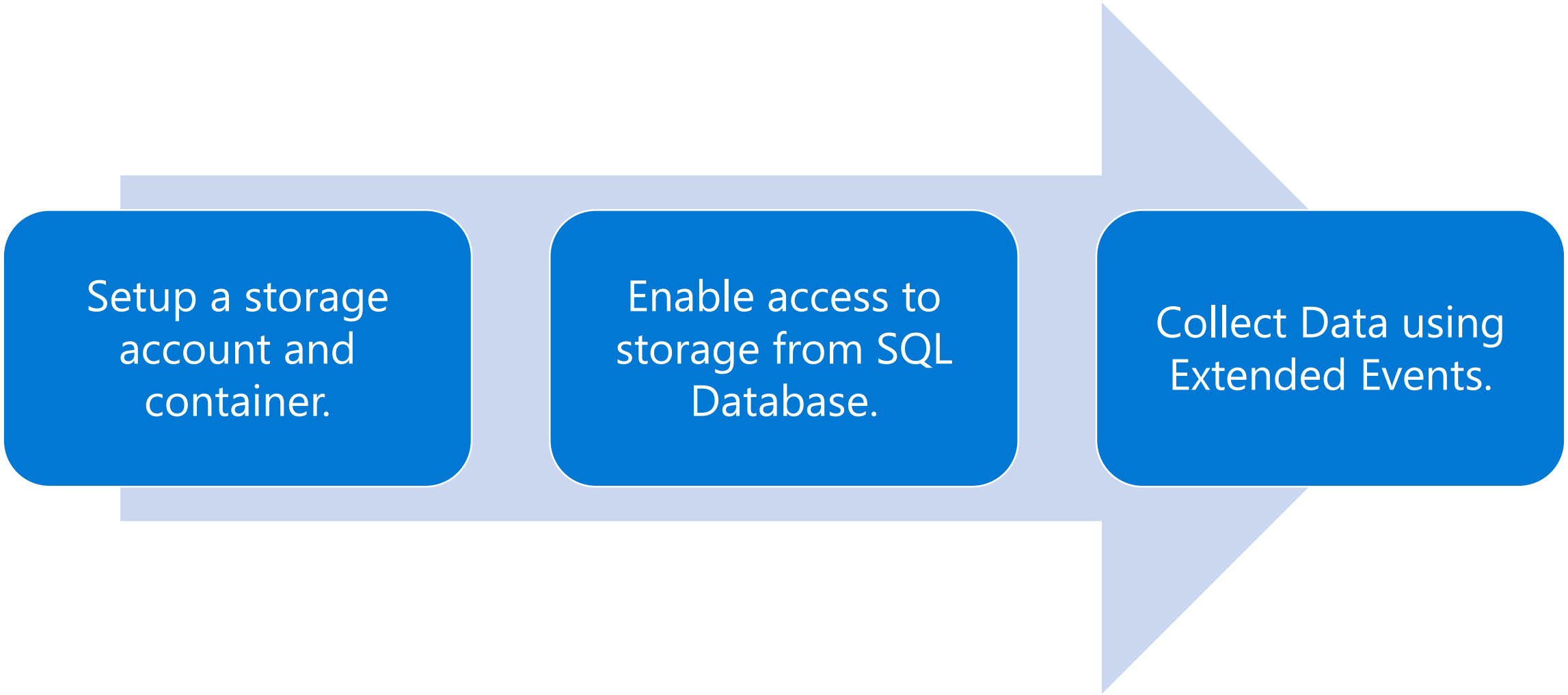
XEvents Objects Explained

Event	• Predefined instrumentation points in the code.
Actions	• Event independent data to add to the collection. • For example: sql_text, create_dump_all_threads
Predicates	• Independent fields for filtering. • For example: database_id, session_id
Targets	• Ring Buffer, Event Counter, Event File
Maps	• Maps “codes” to meaningful names. • For example: wait_type

XEvents Usage Scenarios



How to enable Extended Events



Storage Container Authorizations

Collect Data Using Extended Events

Create a master key specifying a strong password

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

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

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

Define the Extended Event

- Start the event and wait to reproduce the issue.
- Once the issue has been reproduced, stop the event.
- You should see XEL files in the storage container in Azure Storage Explorer.
- You can then download to your laptop/local machine.

Demonstration

Extended Events

- Create Extended Events session using SSMS.
- View Extended Events session.

Monitoring Azure SQL Database Performance using Extended Events

- **Exercise 1:** Monitor Azure SQL Database using Extended Events.

Questions?

Knowledge Check

List three targets for extended events output.

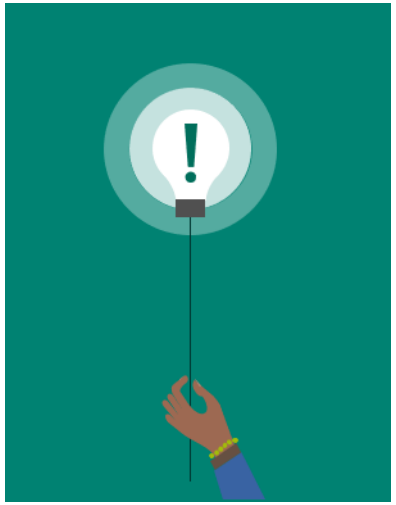
List three problematic scenarios where extended events can help.

Lesson 5: Configure Alerts through Azure Portal

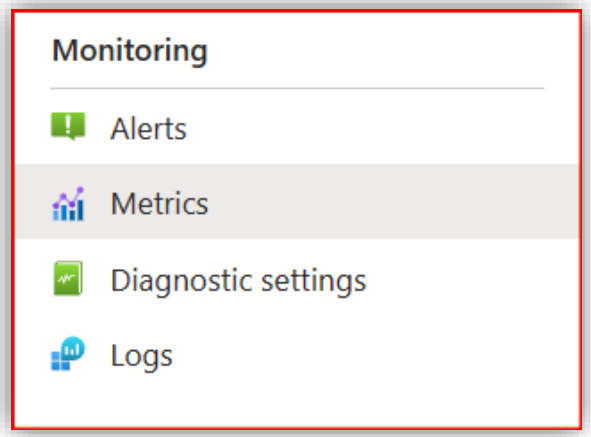
Objectives

After completing this learning, you will be able to:

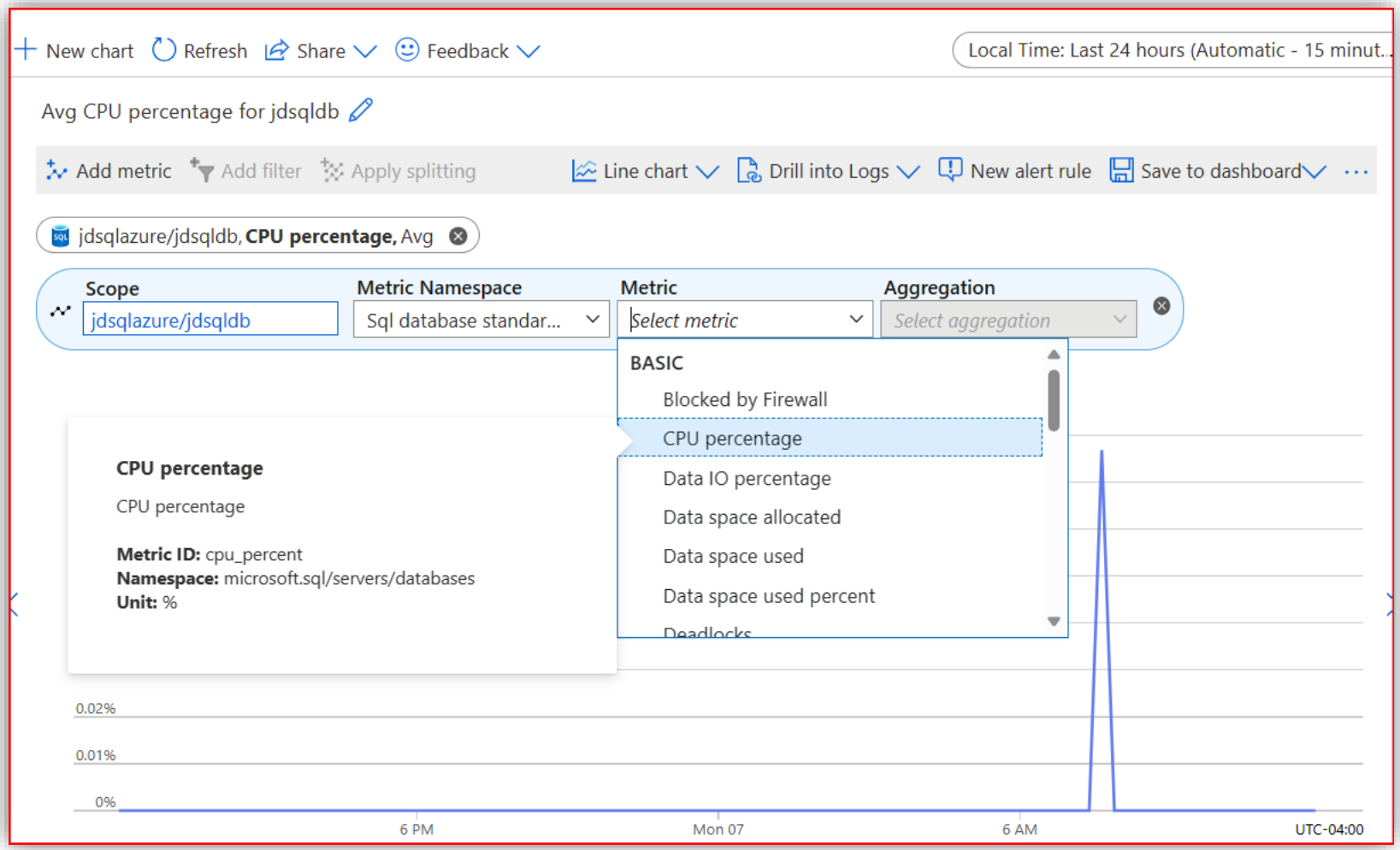
- Configure alerts using Azure Management Portal.



Metrics and Alerts



- Metrics enable you to see if a database is approaching the limits of CPU, memory, IO, or storage resources.
- High DTU, CPU or IO utilization may indicate that your workload needs more resources.



Purpose of Alerts for Azure SQL Database

Database alerts can help to proactively trigger various events related to database connectivity, high DTU usage or deadlocks, etc.

It helps to proactively resolve underlying issues to avoid application outages and improve user experience.

Receiving an alert based on monitoring metrics or events on

Metric values

- The alert triggers when the value of a specified metric crosses a threshold you assigned in either direction. It triggers when the condition is first met and then when that condition is no longer being met.

Activity log events

- An alert can trigger on every event, or, only when a certain number of events occur.

Purpose of Alerts for Azure SQL Database

You can configure an alert to do the following when it triggers:

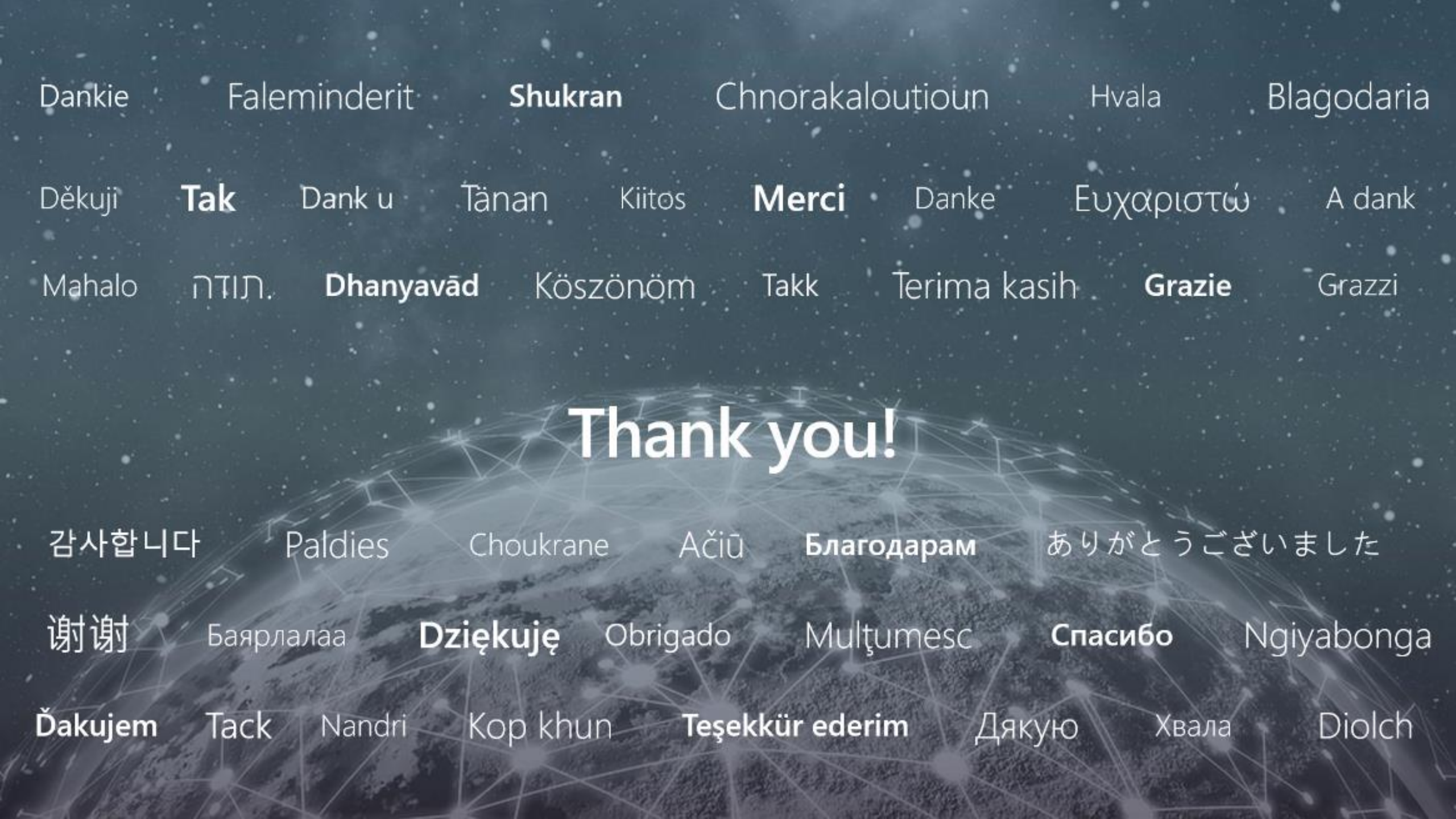
- Send email notifications to the service administrator and co-administrators.
- Send email to additional emails that you specify.
- Call a webhook

You can configure and get information about alert rules using

- Azure portal
- PowerShell
- command-line interface (CLI).
- Azure Monitor REST API.

SQL Database alert values

Metric Name	Aggregation Type	Minimum Alert Time Window
CPU percentage	Average	5 minutes
Data IO percentage	Average	5 minutes
Log IO percentage	Average	5 minutes
DTU percentage	Average	5 minutes
Total database size	Maximum	30 minutes
Successful Connections	Total	10 minutes
Failed Connections	Total	10 minutes
Blocked by Firewall	Total	10 minutes
Deadlocks	Total	10 minutes
Database size percentage	Maximum	30 minutes
In-Memory OLTP storage percent(Preview)	Average	5 minutes
Workers percentage	Average	5 minutes
Sessions percent	Average	5 minutes
DTU limit	Average	5 minutes
DTU used	Average	5 minutes


Dankie
Faleminderit
Shukran
Chnorakaloutioun
Hvala
Blagodaria
Děkuji
Tak
Dank u
Tänan
Kiitos
Merci
Danke
Ευχαριστώ
A dank
Mahalo
תודה.
Dhanyavād
Köszönöm
Takk
Terima kasih
Grazie
Grazzi
Thank you!
감사합니다
Paldies
Choukrane
Ačiū
Благодарам
ありがとうございました
谢谢
Баярлалаа
Dziękuję
Obrigado
Mulţumesc
Спасибо
Ngiyabonga
Ďakujem
Tack
Nandri
Kop khun
Teşekkür ederim
Дякую
Хвала
Diolch

Demonstration

Configure Alerts through Azure Portal

- Configure alerts through Azure Portal.

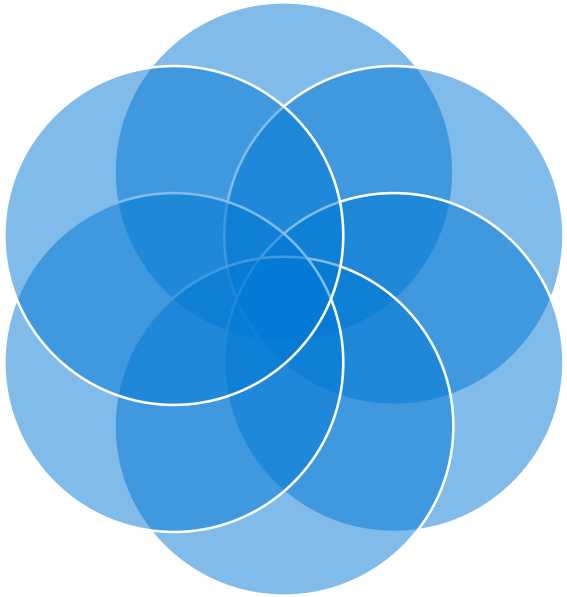
Questions?

Module Summary

Monitoring and Troubleshooting Azure SQL Database

Configure Alerts through Azure Portal

Monitoring Query Performance using Query Performance Insight



Monitoring Azure SQL Database Performance using Extended Events

Azure SQL Database Tuning using Automatic Tuning

Monitoring Azure SQL Database Performance using Intelligent Insights