**Microsoft Azure Database monitoring playbook**

Contents

[**Monitoring key Concepts Refresher** 1](#_Toc38841167)

[**Why Is This Important?** 1](#_Toc38841168)

[**Importance of Monitoring:** 1](#_Toc38841169)

[**Types of Monitoring Data in Azure:** 2](#_Toc38841170)

[**Diagnostic Log Targets:** 4](#_Toc38841171)

[**Implementing monitoring for Azure SQL database and Azure Synapse Analytics** 4](#_Toc38841172)

[**Benefits of PaaS Database** 4](#_Toc38841173)

[**Azure SQL Database Deployment Types:** 5](#_Toc38841174)

[**Monitoring plan of attack:** 5](#_Toc38841175)

[**SQL Diagnostic Settings:** 6](#_Toc38841176)

[**Monitoring SQL Performance:** 6](#_Toc38841177)

## **Monitoring key Concepts Refresher**

### **Why Is This Important?**

* Data is the foundation of pretty much every important process in any organization.  
  Most corporate data is stored in database services.
* This means that monitoring is critical to ensure the security, health, performance of those database services.
* Benchmarking: When we say that database is slow, it’s very ambiguous. That’s why we need bench marking. With Benchmarking we can show where the performance is today as compared to a known good state.

### **Importance of Monitoring:**

Monitoring is critical for any system component. We monitor everything, but databases really are at the top of most considerations for a system.

If the database is not performing, then nothing else above it, the application, is not going to perform. So we have to make sure we have great insight into our database, and insight comes from monitoring. If there are undetected database problems, then problems will be exhibited throughout the entire application, and it will be very hard to diagnose.

Monitoring provides insight into many aspects of a database:

**Performance:**  This includes created a baseline performance for our DB. We can use the last good metrics we had as something we can compare what we are seeing right now. Also note that baselining is not a one-time thing. It is continuous. As workloads evolve, we are going to constantly see trends and baseline enabled that visibility. We can then adjust according to the trends.

**SLA adherence and overall Service health:** Monitoring tells us overall health in our environment in number of different ways. Things like cosmos DB have additional SLA beyond just availability. Using monitoring we can track those.

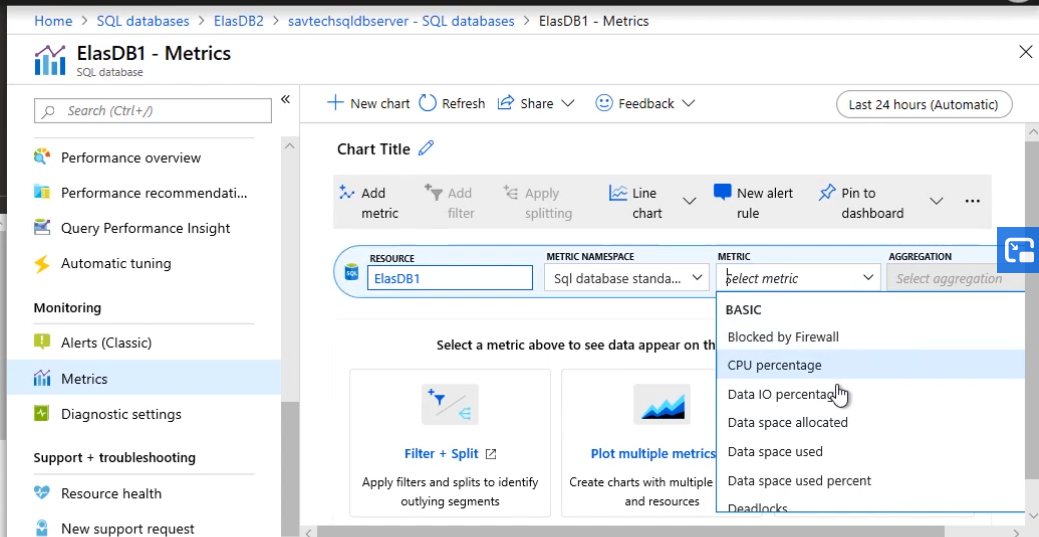
**Spikes and Errors:** Monitoring is going to expose spikes. (Unexpected rises or falls). We need visibility where the spikes and errors are coming from. These are critical to our database.

**Capacity (Storage and Quotas):** Monitoring gives an insight in to whether we are going to hit maximum capacity configured or hit our quotas. This could also be useful to make sure we're really being optimal with our spends. With proper monitoring and addressing the high resource consuming processes/queries we could improve the performance without adding resource. Maybe we could remove resource and therefore save money.

### **Types of Monitoring Data in Azure:**

* Each resource instance, may have different types off monitoring data available. However, those different types of data really fall into three broad categories (it's really two categories of actual monitoring):
  + **Metrics:** These are values we can get about resources. Ex: CPU utilization, DTU utilization. The values available vary depending upon the resource we are monitoring.

Following shows a screen shot of database metrics:



Here we can see that a Db has metrics like CPU percentage, IO, Disk space allocated, DTU Limit, DTU percentage.

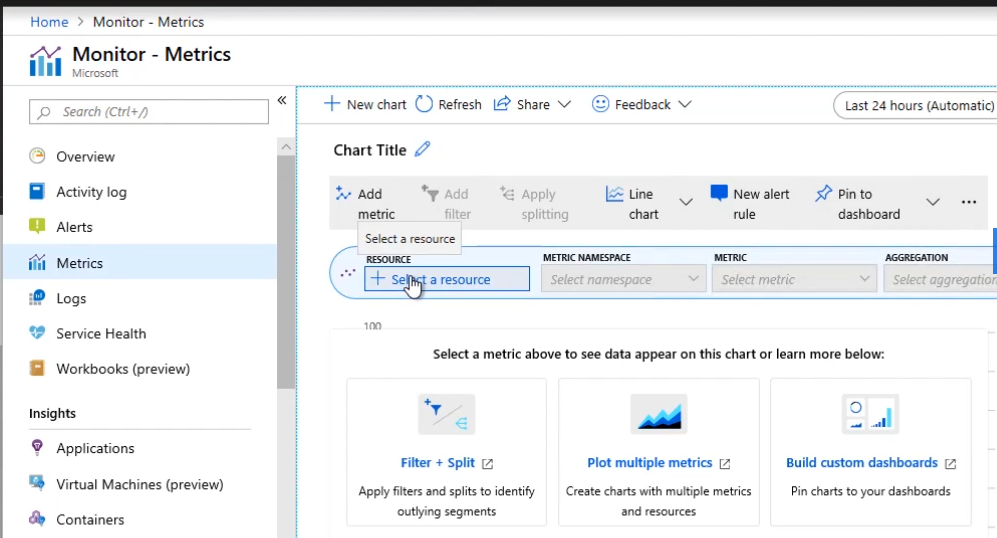
* + **Logs**: These logs are of two types. First ones are generated at the Azure Resource Manager level. These are activity logs that show us information about what has happened at the fabric level to a resource Ex: Resource was created, modified, a new key was generated. Second ones are diagnostics logs. These are generated from inside the resource, and we can send those to various targets.
  + **Alerts (Metrics + Logs):** From those metrics and logs, we can generate alerts based on certain values of the metric or based on some information in a log. These are generated commonly from the Azure subscription/Azure AD tenant, from actual Azure resources or from guest operating systems and guest applications (IaaS scenarios).
* **Azure Monitor:**
  + The goal of Azure Monitor is to really bring together all the different pieces of information we have around Azure related to monitoring of the Azure resources.
  + We can access things like Activity Logs, Metrics, and Logs from Within Azure Monitor.

When we go to “Metrics” tab we can see this is the same view we had in the SQL Database, except this time, there's not a default resource selected.

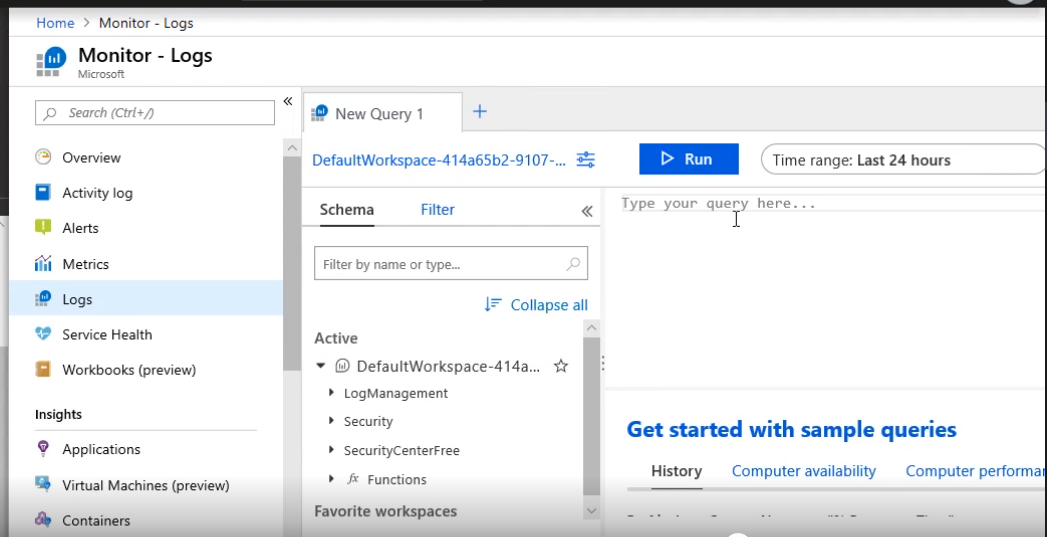
* + Azure Monitor gives us the view of all the metrics across our environment.

Azure Metrics have a very fast pipeline. Typically activity shows up in the metrics within about a minute. So if we wanted to trigger off that through alerting that's going to happen very quickly.

* + Azure Monitor also gives insight into activities and diagnostic logs. The activity log is directly accessed through Azure Monitor. For the diagnostic logs, we can configure for all different resources in the Azure subscription and then we can integrate with Azure Monitor Logs to actually view the content if we have sent them to Log Analytics.
  + Azure Monitor Logs is really the new name for the Log Analytics capabilities of sending data and querying with the Kusto Query Language.



* + We can also generate Alerts. These can be from different sources based on Azure Monitor, Log Analytics, App Insights.
  + Azure Monitor logs takes the Log Analytics capabilities and brings them into Azure Monitor. Following is what we can see under Azure Monitor logs. Any log that we send to our Log Analytics workspace could now be queried from here.



* + Azure also provides monitoring solutions (available in Market Place) that sit on top of Log Analytics and bring a lot more value. They do the analysis and create visualizations for us in a nice dashboards. Monitoring solutions are available for ADF, Databases etc.
  + As mentioned earlier in Azure Monitor, the metrics is a very fast pipeline. We can use this for critical alerts, notifications (typically around a minute). With Log Analytics it is typically longer (5‑minute range)

### **Diagnostic Log Targets:**

* Most Azure resources have a common diagnostics setting. There may be one or more types of logs and one or more types of metrics available depending on the resource type. This diagnostic information can be sent to a combination of three possible targets.
* Metrics are also available through Azure Monitor metrics View. But with logs, there is no native way to interact with those just through the Azure portal. We need to send them to some service that can then consume those logs and give us some way to interact with them.
* For some types of services, we may have to configure diagnostic settings at multiple places. Ex: If our DB is part of elastic pool, we may have to configure diagnostic settings at the DB level and the elastic pool level. The elastic pool also has its own set of diagnostic settings. Also if we had a managed instance database, it has Diagnostic settings as does the managed instance server.
* 3 possible targets are :
  + **Storage Accounts:** Stored in blob. It is useful for cheap, long term storage. We could have other processes read this data and send to some other analysis solution.
  + **Log Analytics:** This allows retention and enables rich analysis. For retention - first 31 days is free and for Data ingestion, 5 GB per month is also free. Above this we have to pay a certain amount. Actual amount may vary based on the region and currency.

It's very common to use Log Analytics for the hot store of data where we want to run that analysis for the most recent data, but then we may move it to blob as we don’t want to keep it there for long term because of the cost.

* + **Event Hub:** Event Hub is publish-subscribe model system. Using this we could send the diagnostic logs to the Event Hub via a publish, and then some external system, typically, a SIEM, will subscribe to that Event Hub and then receive the diagnostic settings, those logs, and potentially, metrics (depending upon whatever we have selected). The 3rd party application could then process and display.

There could be other services. We might have on‑premises solutions, like System Center Operations Manager. Some of these have their own tools that can have management packs, for example, they might interact with Azure Services, they might use APIs, they might pull data from Log Analytics.

* For the configuration, we simply access the Diagnostic settings, select the targets to which we want to send the logs to. In case of log Analytics we also have to select the workspace to which we have to send to. Then we configure what all information we need to send to. For Databases, we have SQLInsights which is a premium service. If we are using Elastic pool or Managed instances then we need to configure diagnostic logs that those levels as well.

## **Implementing monitoring for Azure SQL database and Azure Synapse Analytics**

### **Benefits of PaaS Database**

* Apart from PaaS options, we can install database offerings inside virtual machines. We could manually create a virtual machine and then install database solutions inside that virtual machine. There are templates in the Azure Marketplace that have Azure SQL Database and others pre‑baked into that image, and we can get them up and running very quickly.

However using these approaches we don’t getting all of the cloud benefits available to us. As a customer, we still have massive number of resp: Backups, tuning, maintain the virtual machine etc.

* Using PaaS (DaaS), many responsibilities shift to the provider. Benefits we get using PaaS include :
  + Automated Backups – Backups are part of the service. We can get PITR just by a click of a button.
  + Built-In resiliency options – For Azure SQL Database, we can add a number of readable replicas. We can have technologies like automated fail‑over. For Cosmos DB, we can really replicate data wherever we want. And the manner in which we can interact with that multi‑master model really just depends on the consistency we are configuring.
  + Evergreen
  + Increased scalability and Agility: We can easily scale up and down very easily, and that too instantly.
  + Intelligence capabilities are built into the offering. Ex: Auto-tuning We can get advice for tuning from Azure. Ex: Azure can recommend that we create an index or drop one. We also have options to let Azure do it for us.
  + Advanced Analytics Capabilities: Azure can get learnings from millions of databases hosted in Azure, work out common things it sees, create blueprints around them, and then give us guidance on that. It will perform root cause analysis before we even notice there's a problem.
  + Better Monitoring.

### **Azure SQL Database Deployment Types:**

* **Single Database**: Simplest. It is created against a Server and It has its own sets of assigned resources (in terms of DTU or V-Cores)

There is a concept of logical Server. At SQL server level we mainly adjust the firewall rules.

* **Elastic Pool:** Here we create an elastic pool against a logical Database server. Elastic pool has a set of resources assigned to it and we can create multiple databases within that elastic pool that can share that bucket of resources assigned to the elastic pool. The benefit here is that each database could have varying levels of resource requirement at different times because it's sharing that pool.

Remember that Elastic pool has its own set of diagnostic log settings.

* **Managed Instance:** This is a dedicated instance of SQL Server running in our virtual network. Note that managed instance server is different from the logical servers used in the single and elastic pool. On the managed database instance server, there are instance databases. Resources are provisioned at the managed instance database server level and then all the instance databases running on that server share whatever resources are available.

For Azure SQL data warehouse also we have a logical database server. Same Logical server can have multiple sql databases or data warehouses.

When we are on the logical Server main page, we can see that we can configure firewall rules. However we don’t have any Log or Metrics options.

### **Monitoring plan of attack:**

When we are monitoring we start with the Portal. The portal not only exposes a huge amount of information, but there are intelligence‑backed services that don't just show me the information, they look at patterns, they look at historical trends, they use these analysis technologies to spot problems before we will probably ever do, and then actually perform root cause analysis on them to surface to us.

If we don’t find the answers in the portal, we can move to advanced tools like query Store, Dynamic management views.

### **SQL Diagnostic Settings:**

* For the elastic pool and managed instances, there are diagnostics at the pool/managed instance database server level. For the single database, for database in an elastic pool, and the instance database, there are diagnostics that are enabled per database. Above applies for Data warehouse as well. We have to make sure to hit all levels to ensure we get all the right information.
* Remember that at the logical server level, we don’t have any metrics or diagnostic settings. However we can find them at the Elastic pool level. However note that in the case of the elastic pool, I can just send it metrics. Likewise we would have equivalent settings on the managed database instance server. Beyond this we would have them on individual databases.
* Note that when we configure Diagnostic settings at DB level, we also have option to Send SQLInsights to Azure monitor logs. This has intelligent insights into our DB performance.

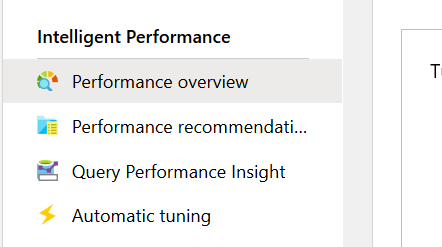
Other Logs that we can capture include “Automatic Tuning”,”Querystore Runtime statistics”, “Query Store wait statistics”, ”errors”, ”timeouts”, ”Blocking”, ”Deadlocks”. For Metrics we have “Basic” metrics that we can choose to send to Azure monitor logs.

Above options are fairly common across all datatypes. The only exception would be for the instance databases that are part of that managed instance SQL deployment that's dedicated just for us. Here we don’t have basic metrics option, Database wait statistics, timeouts, blocks, deadlocks, automatic tuning. However we shall still have Query statistics, Errors, SQL Insights.

* With Azure Data Warehouse the configuration of diagnostic logs is pretty much the same. But the actual data I can send is going to be different. We do have the same Basic metrics. But apart from that under logs we can send DMsWorkers, ExecRequests,RequestSteps,SqlRequests,Waits

### **Monitoring SQL Performance:**

* Azure SQL intelligent Insights is a portal based tools that provides continuous feedback on the SQL performance using artificial intelligence machine learning to help detect and resolve performance challenges. This looks at millions of databases stored in Azure and creates blueprints that can now provide guidance on a per‑database level to spot the problems and give you the solutions.



* Azure SQL Analytics is a monitoring solution built on top of all the data we push to our log Analytics workspace. It provides great visualizations across potentially multiple databases.(not just one). Log data from several DBs can be brought into a common log Analytics workspace and Azure SQL Analytics solution sits on top of that Log Analytics workspace to give us a complete view.
* Another benefit when we send to Log Analytics is all those queries I'm running have no impact on the database itself. All of those analysis‑type operations are running against the separate Log Analytics workspace.
* Another benefit is that we don’t need to go join multiple dynamic views to get information. Intelligent Insights with Azure SQL Analytics solution does this for us. Remember when configuring Logs we had an option to select SQLInsights. This helps us detect all problems that might be plaguing our DB in an easier way.
* Note that for Azure synapse Analytics , we could send diagnostics to Log Analytics, but the solutions like intelligent insights and Azure SQL Analytics are not available with Azure Synapse analytics. It is only for Azure SQL Database. But that does work across single database, it does work for Elastic Pools, and it does work for Managed Instance.

### **Monitoring Query Performance:**

* Most interactions with SQL Database will be via various types of query, and having insight into those queries can provide guidance to tune the queries and the database. We can increase the performance of our entire DB by throwing resources at mu Azure SQL DB instance, but instead of wasting money, it’s better for everyone if we could improve performance by tuning the queries. We can only keep adding resource so far, and that may not even help, depending on exactly the problem.
* Azure SQL Db provides 2 tools to aid with performance of our queries: 1) Query Performance insights 2) SQL Database advisor.
* SQL Database Advisor is actually more than just queries. We can find this in Azure Advisor directly. If we go to Azure Advisor, it gives us guidance on a whole range of things like cost and high availability and performance and security.
* Both SQL Database and Synapse Analytics will actually show up in Azure Advisor if there are things that are applicable.
* Query Performance Insights works only for Azure SQL Database. Not applicable for Azure Synapse Analytics. For the Query Performance Insight, the Query Store has to be enabled on the database. But this is enabled by default for Azure SQL Database, so there nothing we actually have to do to enable this.
* Remember that we can configure diagnostic settings to send query insights as well to Log Analytics workspace.

### **Monitoring Beyond the Portal:**

* Sometimes we might want to move beyond the portal and get deeper insights. Key tools that can be utilized are

**Dynamic Management Views:**

These are available on both Azure SQL DB and on-premises SQL Server Deployments. However exact data we see might differ slightly. To access dynamic management views, we go to SSMS > Master > Views > System Views. Here we can find many dynamic management views. Most of the time dynamic management views will start with dm\_, but not always. Examples of Dynamic management views includes sys.resource\_stats. In this view we get information about various databases and information about CPU perc, memory and IO. In this we get a one record per resource every 5 mins if there is some activity and the data is retained for 14 days. We can also go under UserDB > Views> System Views. Here we can find database level dynamic views Ex: sys.dm\_db\_resource\_stats. Here we get information specific to our database. Here we get record every 15 secs and they are created even if DB is idle. They are retained for 1 hour. Other imp view would be sys.db\_dm\_wait\_stats.

**Query Store:**

This is a fully managed database feature that continuously collects and then can give us insight into the historical information about all of the queries. This is enabled by default. This gives us the ability to find queries that are taking a long time to run, queries that are using the most CPU, the most IO, the most memory, queries that are making us wait for certain types of resources.

The Query Performance Insights that provides nice information about queries and provides hints/steps is actually based on this query store information. However we can still access Query Store directly. To access we can go to SSMS > Server > User DB > Query Store.

This works for both Azure SQL database and Data warehouse.

**xEvents (Extended events):**

This is a light weight tracing technology and it uses few resources. This is something we setup to continoully gather information about something that we specify. This replaces things like SQL Trace or profiler used on on-premises installation. Extended events can be seen under SSMS > Server > UserDB > Extended Events.