Log Analytics

A ‘SaaS in a Box’ Tutorial

# Purpose of this tutorial

This tutorial covers setting up and using Log Analytics with the WTP SaaS app. It builds on the Performance Monitoring and Management tutorial, and shows how to use Log Analytics to augment the monitoring and alerting provided in the Azure portal. Log Analytics is particularly suitable for monitoring and alerting at scale – it supports hundreds of pools and hundreds of thousands of databases. It also provides a single pane of glass monitoring solution, which can integrate monitoring of different applications and Azure services, across different Azure subscriptions if required. This tutorial explores the Azure SQL Analytics solution in Log Analytics, for monitoring SQL Database pools and databases.

# SaaS performance management patterns

See the Performance Monitoring and Management tutorial for a discussion of the SaaS scenarios and patterns and how they affect the requirements on a monitoring solution.

# Monitoring and managing performance with Log Analytics (aka OMS)

The Azure portal provides built-in monitoring and alerting on most resource blades. For SQL Database, monitoring and alerting is available on databases and pools. This built-in monitoring and alerting is resource-specific and convenient for small numbers of resources, but is less well suited to monitoring large installations or for providing a unified view across different resources and subscriptions.

For high-volume scenarios Log Analytics can be used. This is a separate Azure service that provides analytics over emitted diagnostic logs and telemetry gathered in a log analytics workspace, which can collect telemetry from many services and be used to query and set alerts. Log Analytics provides a built-in query language and data visualization tools allowing operational data analytics and visualization. The SQL Analytics solution provides several pre-defined elastic pool and database monitoring and alerting views and queries, and lets you add your own ad-hoc queries and save these as needed. OMS also provides a custom view designer.

Log Analytics workspaces and analytics solutions can be opened both in the Azure portal and in OMS. The Azure portal is the newer access point but may be behind the OMS portal in some areas.

The tutorial walks through installing and using Log Analytics and the Azure SQL Analytics solution and then opening and using the solution in the context of the WTP application.

# Setup

**Download and extract SaaS in a Box Learning Modules and Utilities.zip** to a convenient folder.

**Install and set up the WTP Application** including initializing the catalog. See <link>. Ensure the deployment is complete before starting this tutorial.

**PowerShell ISE** is recommended for executing scripts and following their execution in debug mode.

**PowerShell Tips**

* Open and configure demo- scripts in the PowerShell ISE.
* Use F5 to run the script (using F8 is not advised as the $PSScriptRoot variable is not evaluated when running snippets of a script).
* Use F9 to set a breakpoint to let you trace the script in debug mode to see how it works
* Use F10 to step through the script, F11 to step into a function, and Shift-F11 to step out.

# Walkthrough

## Setup

It’s anticipated that this tutorial will be run after the Performance Management and Monitoring tutorial. Be sure to deploy the batch of tenants, as described in Exercise 1 of that tutorial, and then use some of the load generation scenarios from that tutorial to create interesting loads to monitor in Log Analytics.

1. Update the user configuration file used by all tutorial scripts. Update again if you redeploy the app.
   1. Open ...\Learning Modules\UserConfig.psm1 in **PowerShell ISE**
   2. Modify **$userConfig.ResourceGroupName** to the resource group used for the deployed app.
   3. Modify **$userConfig.Name** to the User name used for the deployed app.
2. Open …\Learning Modules\Provision and Catalog\Demo-PerformanceMonitoringAndManagement.ps1 in PowerShell ISE.Keep this script open as you may want to run several of the load generation scenarios during this tutorial.
3. If not already done, provision the extra tenants to provide a more interesting monitoring context.
   1. Set **$DemoScenario = 1,** **Provision a batch of tenants**
   2. Execute the script using F5.
4. Make sure the load generator is running. If necessary, restart another session.
   1. Set **$DemoScenario = 2,** **Generate a normal intensity load**. You can experiment with different load scenarios.
   2. Execute the script to start the load generation jobs using F5.

## Exercise 1: Installing and configuring Log Analytics and the Azure SQL Analytics solution

Log Analytics is a separate service that needs to be configured. Log Analytics collects log data and telemetry and metrics in a log analytics workspace. A workspace is a resource like others in Azure and must be created. While it doesn’t need to be created in the same resource group as the application(s) it is monitoring, this will often make sense. In the case of the WTP app this works well and allows the workspace to be deleted with the application when the time comes.

1. Open ...\Learning Modules\Performance Monitoring and Management\Log Analytics\Demo-LogAnalytics.ps1 in **PowerShell ISE**
2. **Execute using F5**

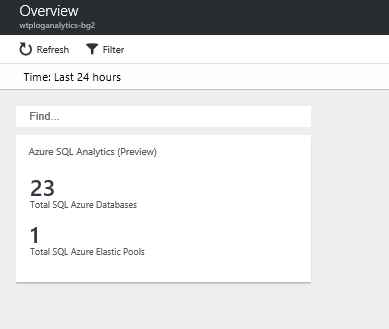
At this point you should be able open either Log Analytics in the Azure portal or the OMS portal. It will take a few minutes for telemetry to be collected in the Log Analytics workspace and to become visible. The longer you leave the system gathering data the more interesting the experience will be. Now’s a good moment to grab a coffee!

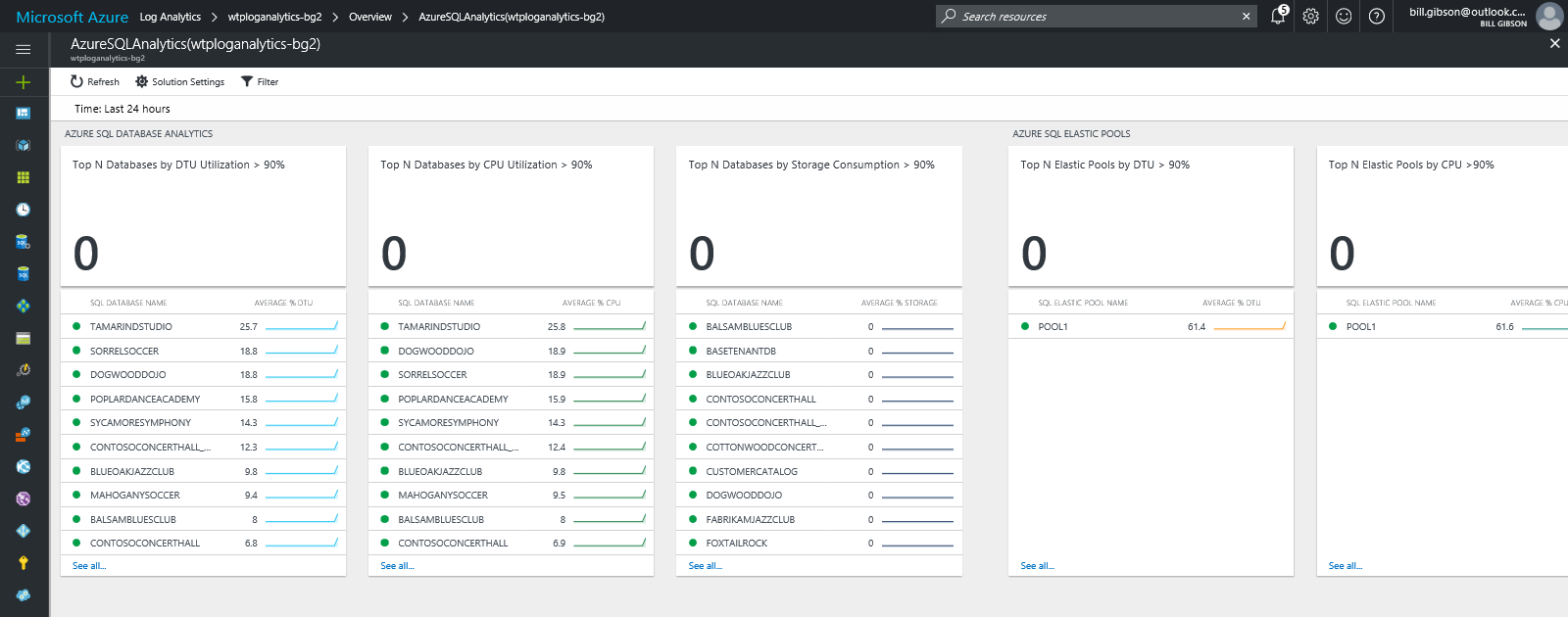
## Exercise 2: Use Log Analytics and the SQL Analytics solution to monitor pools and databases

In this exercise, you will open Log Analytics and the OMS portal to look at the telemetry being gathered for the WTP databases and pools.

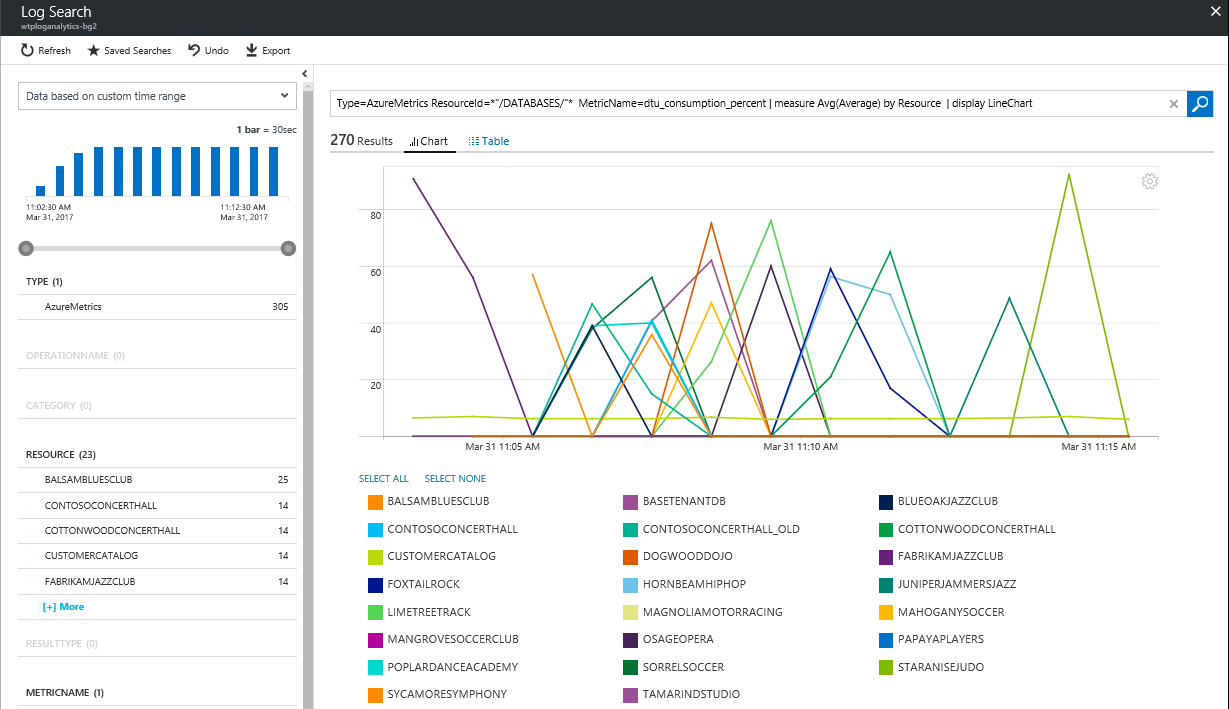
1. Open the Azure portal and locate and click the Log Analaytics option on the menu on the left.
2. Select the workspace named wtploganalytics-<USER>

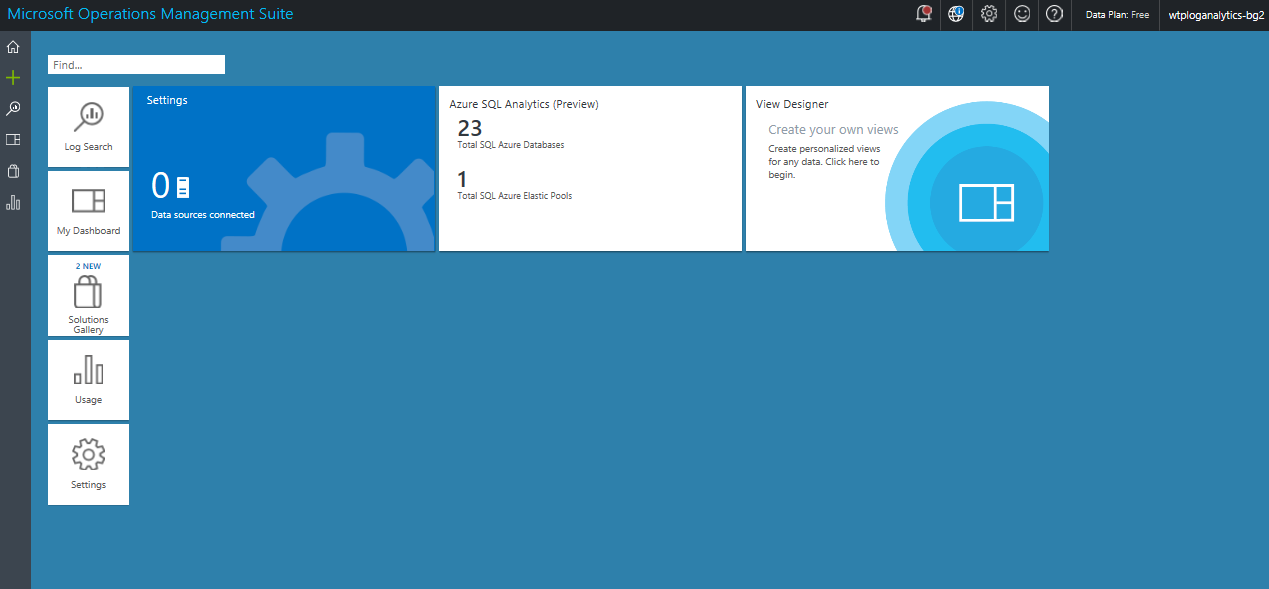


1. Select Overview to open the Azure SQL Analytics solution in the Azure portal, or select OMS Portal to view it in the OMS portal.   
     
   **IMPORTANT**: It may take a couple of minutes before the solution is active. Be patient!   
   
2. Click on the Azure SQL Analytics solution tile to open it.



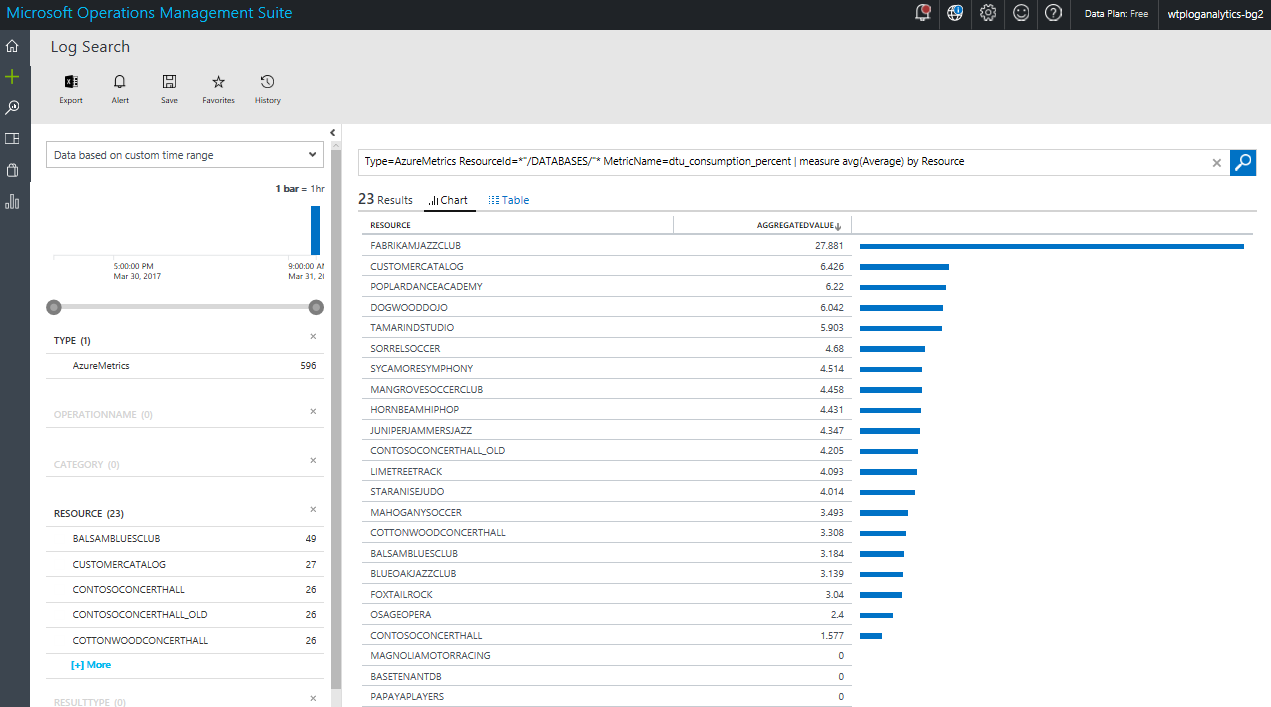
1. The view in the solution blade scrolls sideways, with its own scroll bar at the bottom (refresh the blade if needed).
2. Explore the various views by clicking on them or on individual resources to open a drill-down explorer, where you can use the time-slider in the top left or click on a vertical bar to focus in on a narrower time slice. With this view, you can select/de-select individual databases or pools to focus in on specific resources (see below).

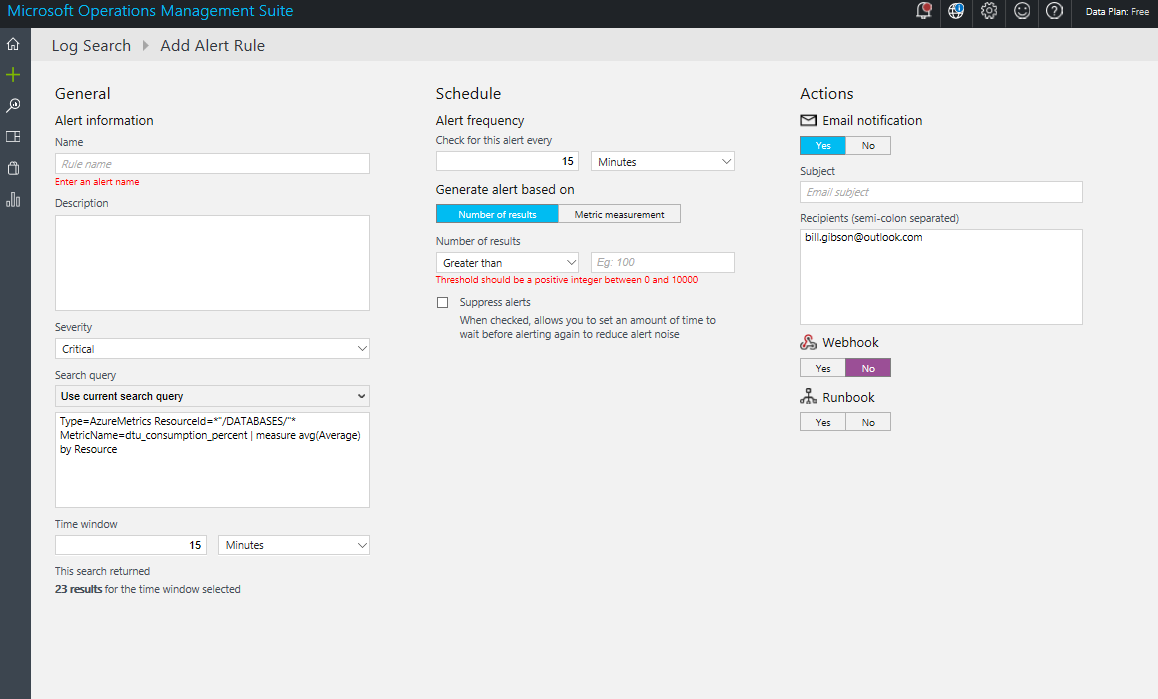


1. Back on the solution blade, if you scroll to the far right you will see some saved queries that you can click on to open and explore. You can experiment with modifying these, and save any interesting queries you produce, which you can then re-open and use with other resources.
2. Back on the Log Analytics workspace blade, select OMS Portal to open the solution there.   
   
3. In the OMS portal, you can configure alerts. Click on the alert portion of the database DTU view.



1. In the Log Search view that appears you will see a bar graph of the metrics being represented.



1. If you click on Alert in the toolbar you will be able to see the alert configuration and can change it.   
   

The monitoring and alerting in Log Analytics and OMS is based on queries over the data in the workspace, unlike the alerting on each resource blade, which is resource-specific. Thus, you can define an alert that looks over all databases, say, rather than defining one per database. Or write an alert that uses a composite query over multiple resource types. Queries are only limited by the data available in the workspace.

Log Analytics for SQL Database is charged for based on the data volume in the workspace. In this tutorial, you created a Free workspace, which is limited to 500MB per day. Once that limit is reached data is not added to the workspace.

# Resources

For more information on using Azure Log Analytics see the following:

<https://blogs.technet.microsoft.com/msoms/tag/msoms/>

<https://blogs.technet.microsoft.com/msoms/2017/02/21/azure-sql-analytics-solution-public-preview/>

<https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-azure-sql>