

Azure SQL Database – Elastic Database Pools

Level-300 Demonstration

Script

Version 1.0

9/3/2015

Prepared for: Darmadi Komo

Prepared by: Prowess Consulting

|  |  |
| --- | --- |
| **Form Factor** | Demo script (Microsoft Word) |
| **Target Audience** | TDMs and database administrators |
| **Goals/Objectives** | Increase confidence among Microsoft SQL Server administrators that Azure SQL Database is easy to configure and can still meet the needs of their production databases. |

Overview

This demonstration explores elastic database pools in Microsoft Azure SQL Database in the context of a software-as-a-service (SaaS) provider, Wingtip Tickets, which provides ticketing software to artists and groups. This demonstration centers on the tenant Julie and the Plantes (a fictitious pop-music tenant.

Other Tenants that will be discussed in future labs, will include the following:

* The Archie Boyle Band (a fictitious rock-music tenant)
* Walla Walla Symphony (a fictitious classical-music tenant)

## Demo Architecture

## 

**Figure 1** Overall architecture of demo components

## Dependencies

This demonstration requires running the deployment and configuration PowerShell scripts from the Level-200 demonstration in order to work.

|  |  |  |
| --- | --- | --- |
|  | Section 0: Prerequisite step for Azure SQL Database Elastic Database Pools - Creating a Database |  |
|  | Create a new database in Azure SQL Database.   1. From **portal.azure.com**, at the top left, click **+ New**. 2. Click **Data + Storage**. 3. Click **SQL Database**. |  |
|  | 1. In the **Name** field, type **Customer2**. 2. Under **Server**, select the **Primary Database Server**. (In this example *mbjulieandtheplantesprimary.database.windows.net*) |  |
|  | 1. Click **Create**. 2. If you selected the **Pin to Starboard** check box, the Starboard (homepage) will appear and a tile with your Customer2 database will be added and will reflect the status. |  |
|  | 1. Once the Database is created, you’ll be taken to the database overview automatically. Here you can view graphs that reflect the resource utilization and connection information, in addition to being able to change the pricing tier and other related settings. |  |

|  |  |  |
| --- | --- | --- |
|  | **Section 1: Azure SQL Database Elastic Database Pools** |  |
|  | Scaling up individual databases to higher-performance pricing tiers is not always the most efficient means of meeting increased demand: often the exact demands of a database are not known, and managing the individual capacity of these databases isn’t efficient. This is especially true for SaaS providers such as the one in this lab, which might have hundreds or even thousands of different Tenant databases, each with its own unique mix of performance demands, which rarely coincide. Elastic database pools help solve this by enabling you to set a policy for a group of elastic databases. This way you don’t have to worry about the performance of each database individually or inefficiently overprovision individual databases to meet occasional peak demand. | |
|  | 1. From **portal.azure.com**, click **Browse**. 2. Click **SQL servers**. 3. Select the *Primary Database Server* (in this example *mbjulieandtheplantesprimary.database.windows.net*). 4. Click **Add pool**. | We’ll start by adding a new elastic database pool from the overview screen for our primary database server. |
|  | 1. Accept the Preview terms (if not already accepted). 2. Click **Add Databases**. 3. Select **Customer1** and **Customer2**. | Next, we add the databases we want to the pool. |
|  | 1. Click **Select**. 2. Click **OK**. 3. If you select the **Pin to Starboard** check box, the Starboard (homepage) will appear and a tile with your ElasticPool-1 will be added and will reflect the status. | While creating the pool, we can also opt to add it to our Azure Starboard (our homepage). |
|  | 1. Once created, you’ll be taken to the ElasticPool-1 overview automatically, where you can view graphs that reflect the resource utilization and connection information, in addition to being able to change the pool size, add additional databases, and set other related settings.   It will take a couple of minutes for your databases to be added to the pool. | Once the pool is created, we’re taken automatically to the ElasticPool-1 overview screen. Here you can see graphs that reflect the resource utilization and connection information for the pool. In addition, we can also change the pool size, add additional databases, and set other related settings for the pool from here. |
|  | 1. If you recall, the Customer1 database was previously set to S0 (10 DTUs) and Customer2 was also set to S0 (10 DTUs). The max pool setting that was created by default for this new ElasticPool is 100 DTUs, which means any of the databases in this pool can now spike to 100 DTUs, but their capacities won’t need to be managed individually. 2. To see this in effect, open the LoadGenerator, type in the Primary Database Server name (in this example, *mbjulieandtheplantesprimary.database.windows.net*), select the first **Event** from the list which will correspond to ConcertId 1, and then **Ticket Level**, user name that you previously created, and then change the **Tickets To Purchase** value to 20000 and **Bulk Tickets to Purchase** value to 500. | Both the Customer1 and Customer2 databases were previously set to S0, which gives a maximum of 10 DTUs to each database. The max pool setting for the new elastic pool we created is 100 DTUs, which means that either of the databases in the pool can now spike to 100 DTUs, but we don’t need to manage their capacities individually.  To see this in action, I’m going to crank up the load generator to hammer the databases: a spike in ticket purchases. Notice how many thousands of tickets were purchased per second. |
|  | 1. Click **Start**. 2. Notice how many thousands of tickets were purchased per second. 3. Return to **portal.azure.com** and to the ElasticPool-1 elastic database pool. 4. Click the **Customer1** database. | Now let’s look at the Customer1 database. Notice in the Resource Utilization graph that the max utilization was about 20 percent, yet the number of tickets per second increased over six times. |
|  | 1. Notice also in the ElasticPool-1 Resource Utilization graph that the pool was barely affected by this spike in usage. | But notice in the ElasticPool-1 Resource Utilization graph that the pool was barely affected by this spike in usage. |

|  |  |  |
| --- | --- | --- |
|  | **Section 2: Conclusion** |  |
|  | Azure SQL Server elastic database pools offer an efficient and cost-effective way of providing groups of databases—large or small—with the means to rapidly scale to meet changes in demand. Since this does not require managing the performance characteristics of individual databases, this can be particularly helpful to SaaS providers. That said, there are times when it is most appropriate to manually scale individual databases to higher performance tiers. To see the vertical database scaling in Azure in action, ask your Microsoft sales representative about seeing the Azure SQL Database vertical scaling demonstration. | |