# 1. Microsoft Azure SQL Database

Microsoft Azure SQL Database (Azure SQL Database) is a relational database-as-a-service, which falls into the industry category Platform as a Service (PaaS). Azure SQL Database is built on standardized hardware and software that is owned, hosted, and maintained by Microsoft. With SQL Database, you can develop directly on the service using built-in features and functionality.

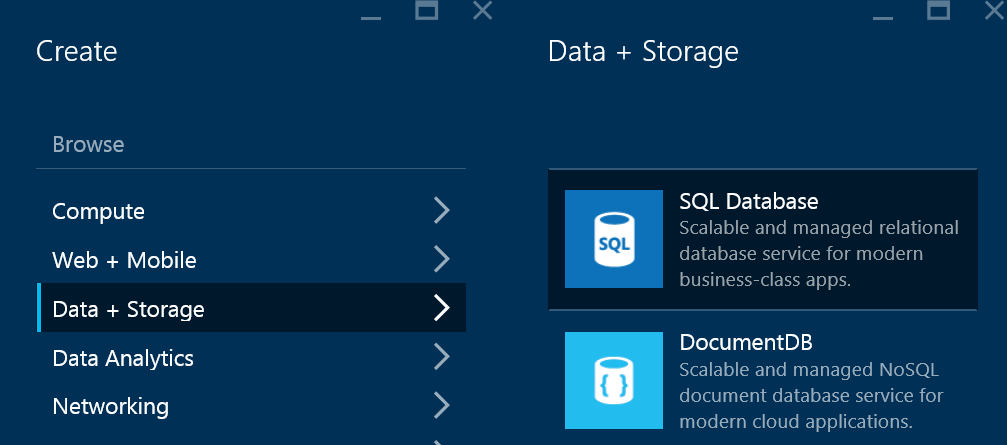
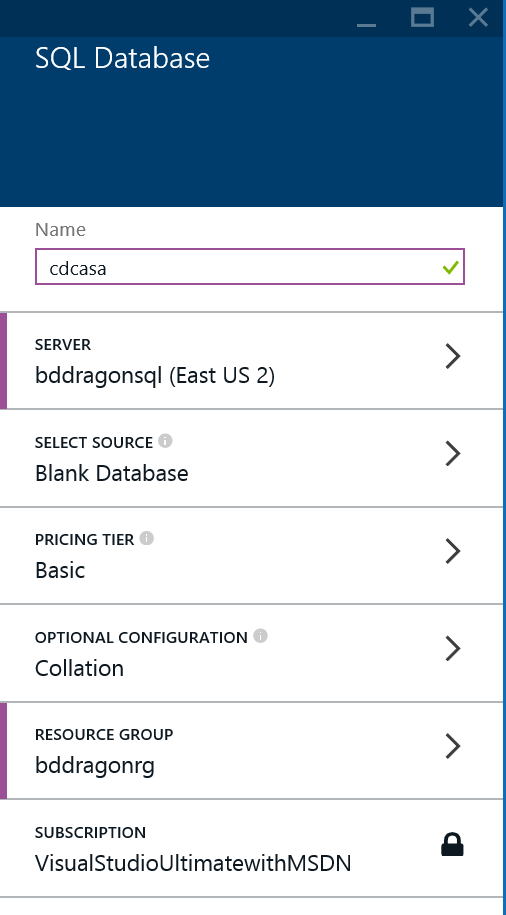
This exercise will demonstrate how to use the Azure portal to create and manage a SQL Azure Database. At the end of this section you will have:

* A new Azure SQL Database and SQL Instance.
* Firewall settings to enable management through Management Studio.
* A new table which will be used in subsequent labs.

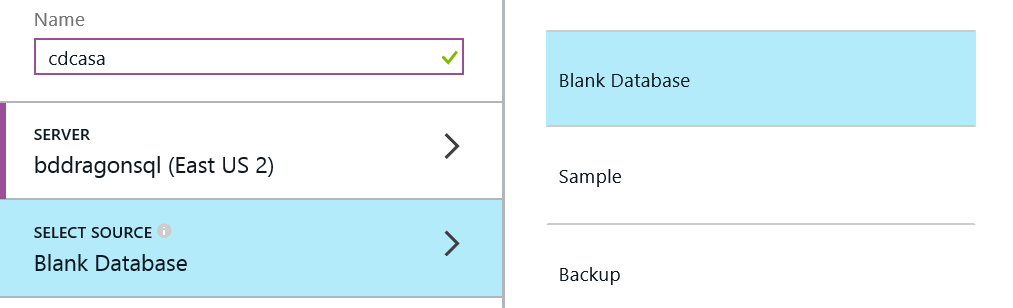
## 1.1 Create a Microsoft Azure SQL Database

The following steps will create an Azure SQL Database that will be used in subsequent labs.

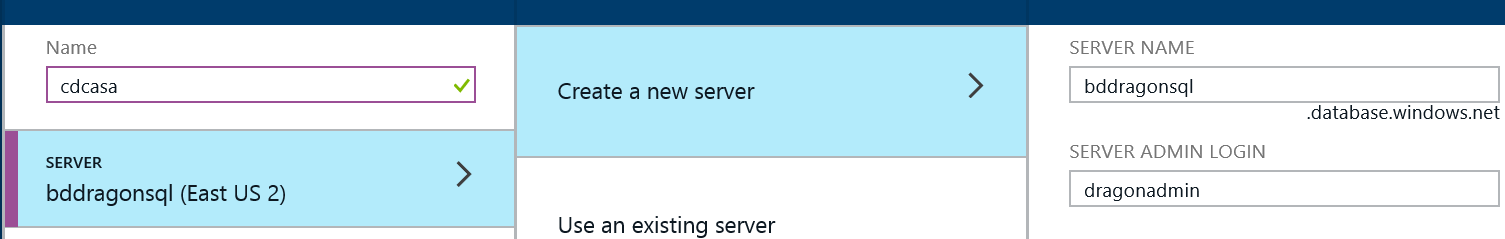
1. Navigate to the Microsoft Azure portal <https://portal.azure.com/>.
2. Click New -> Data + Storage and select the blue square labeled **SQL Database**. Note: DO NOT CHOOSE THE RED SQL SERVER RECTANGLE.

* 
* 

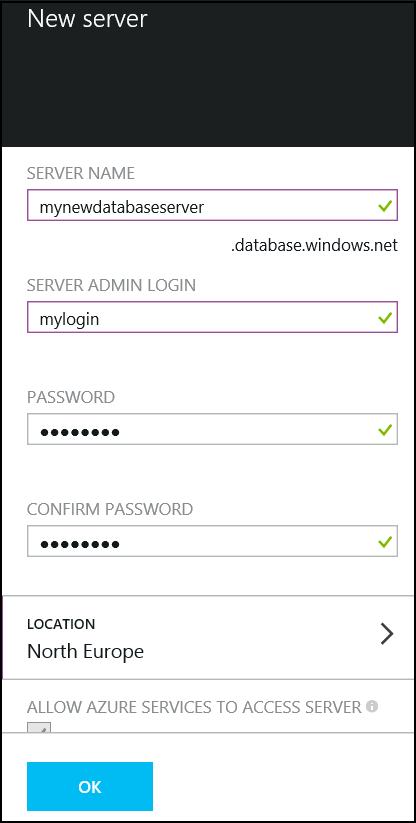
1. Enter the database name **cdcasa**. This database will be used in the Azure Stream Analytics lab.
2. Press the **SELECT SOURCE** button and validate it is configured to **Blank Database**. If needed, close the "Select Source" blade by clicking the X in the upper right corner of the blade.

* 

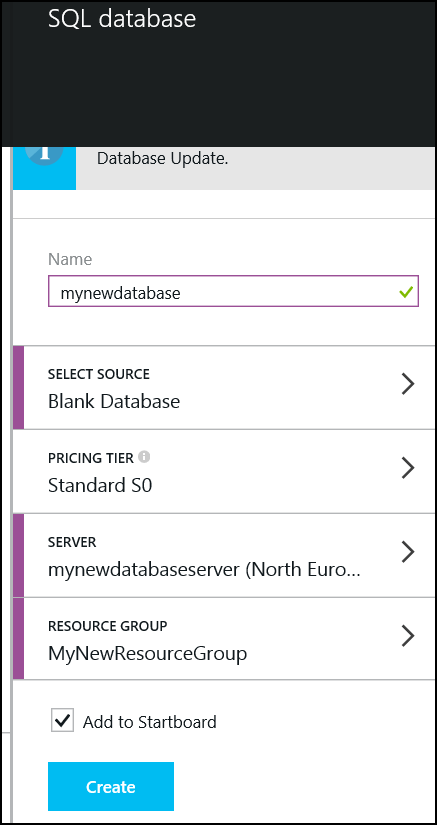
1. Press the **SERVER** button to configure a new server.

* 

1. Click on **Create a new server** to open the New Server blade.
   * Enter a server name, admin login and password.
   * Validate the location is set to the correct location as specified in the course. All Azure services will be created in the same location.
   * Validate the **ALLOW AZURE SERVICES TO ACCESS SERVER** option is checked at the bottom of the blade.
2. Once all the server settings are input, click **OK** in the lower left corner of the New Server blade. This will close the blade and save the changes.

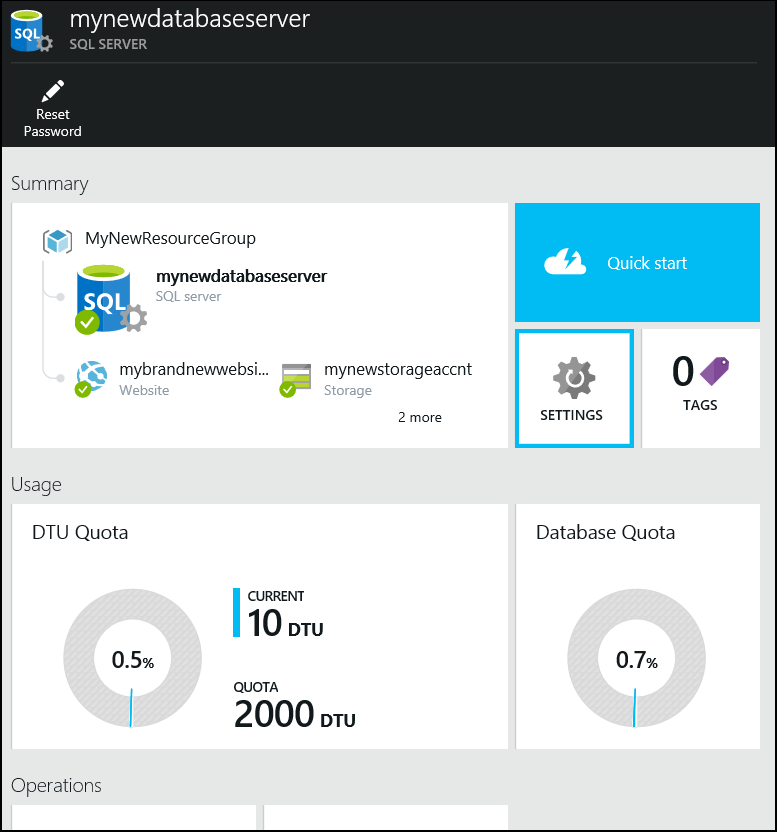
* 

1. Click on **PRICING TIER** to open the Recommended pricing tiers blade. Select **Basic (B)**. Click **Select** in the lower left corner of the blade to accept the change.
2. Click on **RESOURCE GROUP** to open the Resource Group blade. Select the Resource Group to the new one created as part of the Storage Account creation lab.
3. Click Create once all configuration changes are made. The Azure SQL Database will be available in approximately 2-4 minutes.

* 

## 1.2 Configure Firewall Settings for the SQL Server

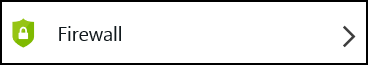
1. Connecting to the database and the server with tools like SQL Server Management Studio will require adding a firewall rule to the server instance. To add a firewall rule to the server instance, choose “Browse” and then choose SQL Servers via the azure preview portal.
2. Click on Browse -> SQL Servers to view the Server instances. Be sure to select the Server instance and not the SQL Server Database. Click on the server you created earlier.
3. The Server Instance blade supports viewing and managing the instance.

* 

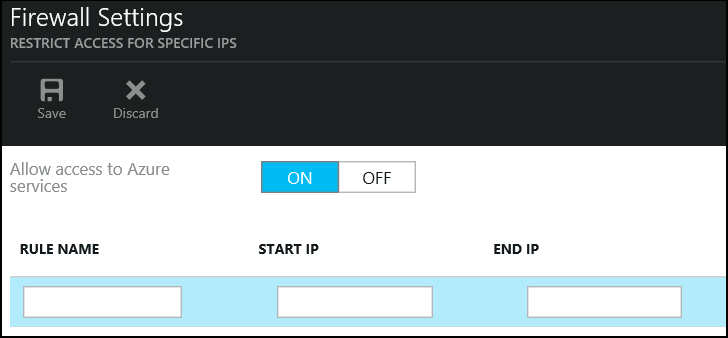
1. Click **Settings** to view and manage the instance settings.

* 

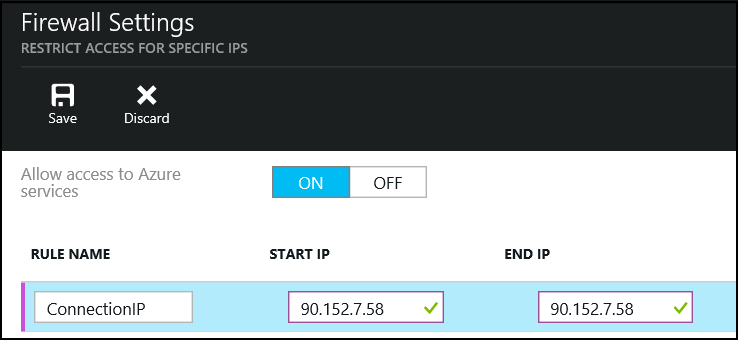
1. Click **Settings** -> **Firewall** to display the firewall rules.

* 

1. The following pane will be displayed.

* 

1. Enter the IP Address range appropriate to access the service. The IP Address for the course virtual machine is found on the desktop wallpaper. Be sure to use the Public IP.
2. Click Save from the upper menu of the blade. Once the changes are complete, a "Success" message will be returned. Click OK. Close the Firewall Settings blade, and close all the blades to return to the Startpage of the portal.

* 

1. You can now connect to the SQL Server instance using SQL Server Management Studio.

## 

## 1.3 Executing Database Queries

<< CHECK INTERNAL AZURE CHANGES ARE PUBLIC>>

# 2. Event Hub

The following lab will demonstrate how to create an Event Hub in Azure and will introduce the Microsoft Azure Management Portal. Many of the labs will connect to Event Hubs as a data source.

Microsoft Azure Service Bus provides a hosted, secure, and widely available infrastructure for widespread communication, large-scale event distribution, naming, and service publishing. Service Bus provides connectivity options including REST endpoints.

Service Bus provides both relayed and brokered messages. The relay service supports direct one-way messaging, request/response messaging, and peer-to-peer messaging. Brokered messaging provides durable, asynchronous messaging components such as Queues, Topics, and Subscriptions.

The Event Hub provides a scale out way to build a high performance and high volume messaging backbone in Azure and is suitable for all types of lines of business that need scale when ingesting data into the Cloud.

## 2.1 Create Event Hub

1. Navigate to the Microsoft Azure management interface <https://manage.windowsazure.com> (NOTE: Event Hubs are only configurable in this interface at this time).
2. Ensure you are on the correct subscription by locating the following button in the upper right hand corner of the screen.

* createHDInsightClusterImg1.png

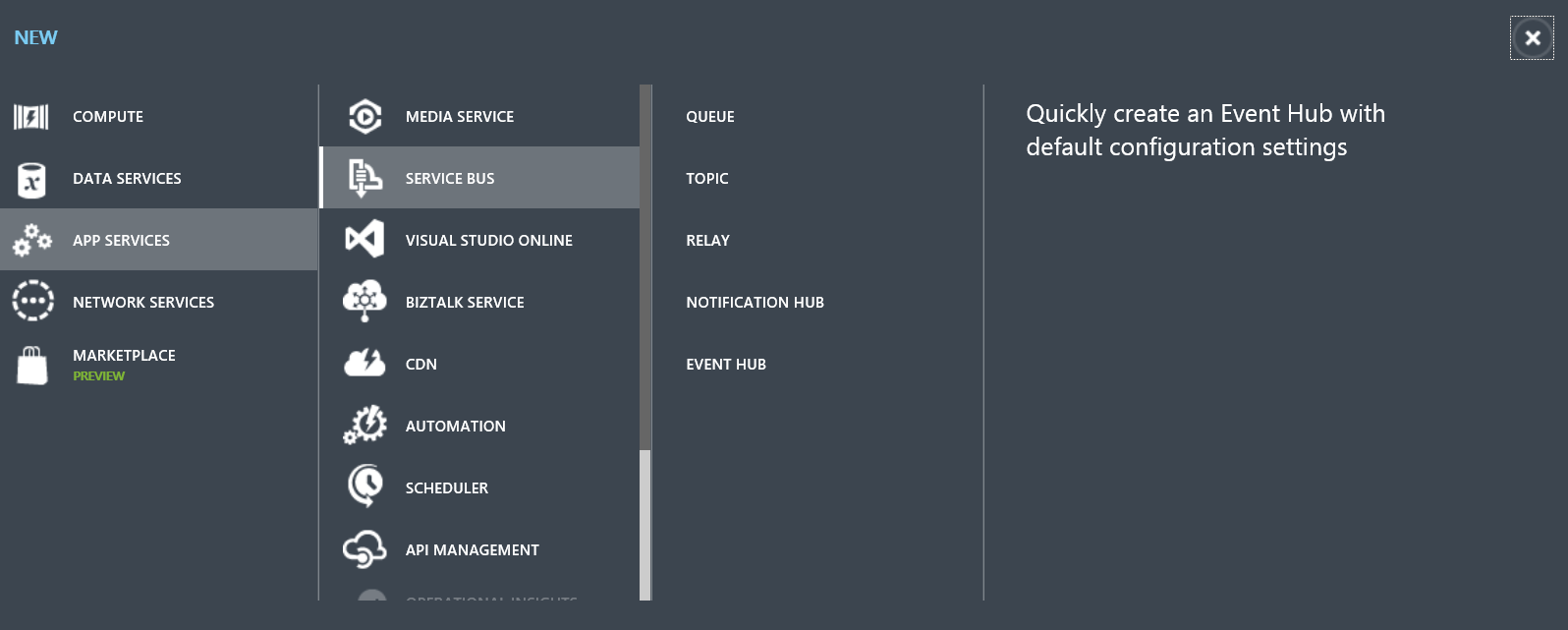
1. Click on the **Service Bus** link on the left hand menu.

* 

1. Click **+NEW** in the bottom left hand corner of the screen.

* 

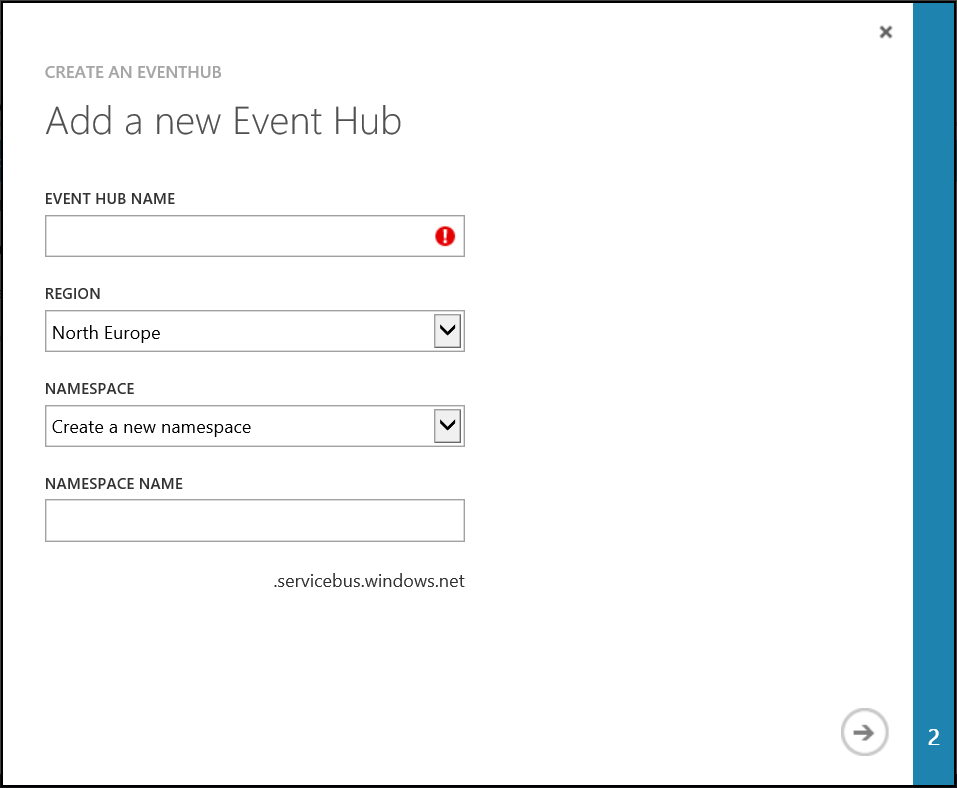
1. From the new menu select App Services -> Service Bus -> Event Hub.

* 
* createEventHubImg3.png

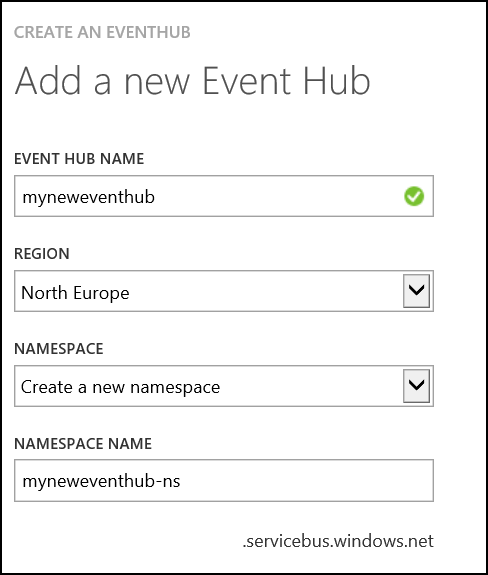
1. Click **Custom Create** on the right of the screen.

* 

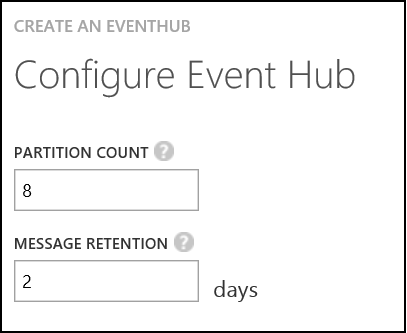
1. The first page of the Event Hub wizard will appear.

* 

1. Enter a name for the Event Hub and also a Service Bus namespace (this must be unique). Click the next (->) arrow on the bottom right of the wizard.

* 

1. Configure the Event Hub.
   * Set partition count to 8. Partitions are a data organization mechanism and are more related to downstream parallelism than to Event Hub throughput. This makes the choice of number of partitions in an Event Hub directly related to the number of concurrent readers you expect to have
   * Set message retention to 2 days.

* Click Finish.
* 

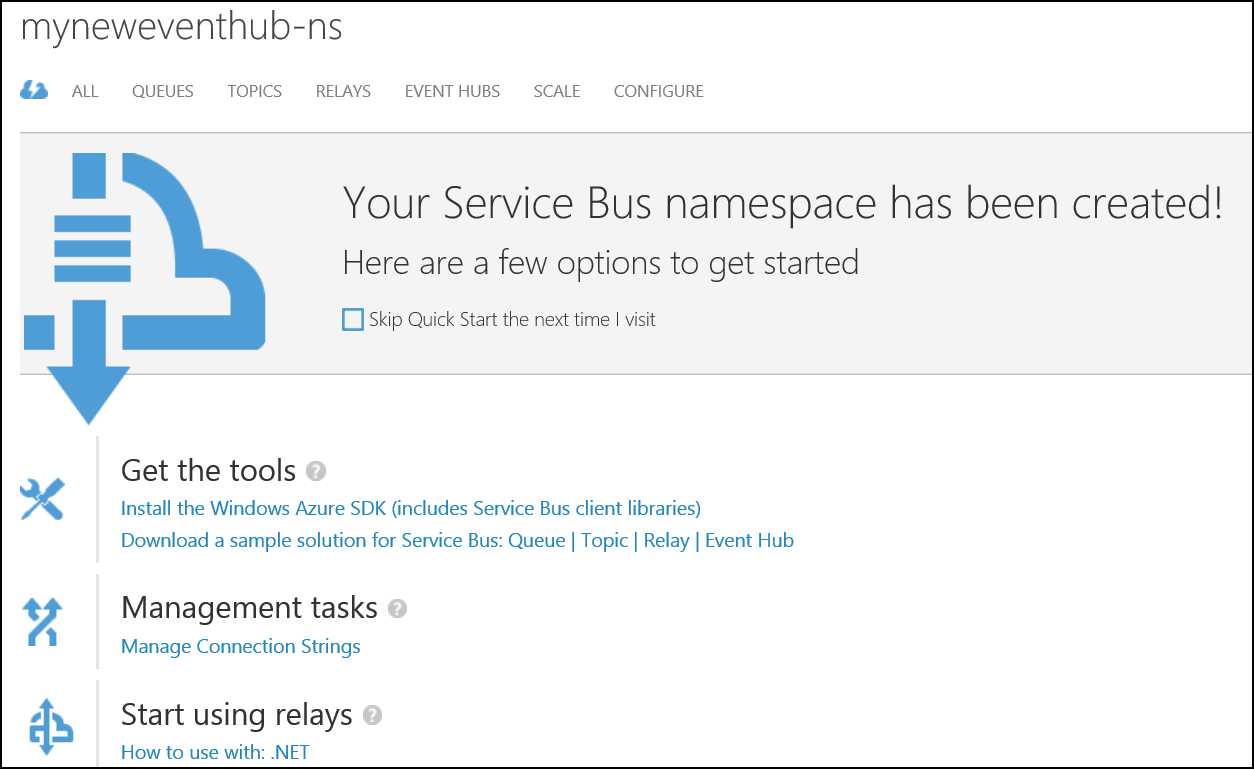
1. Once provisioning is complete the new event hub will appear in the list of available Service Bus entries.

* createEventHubImg8.png

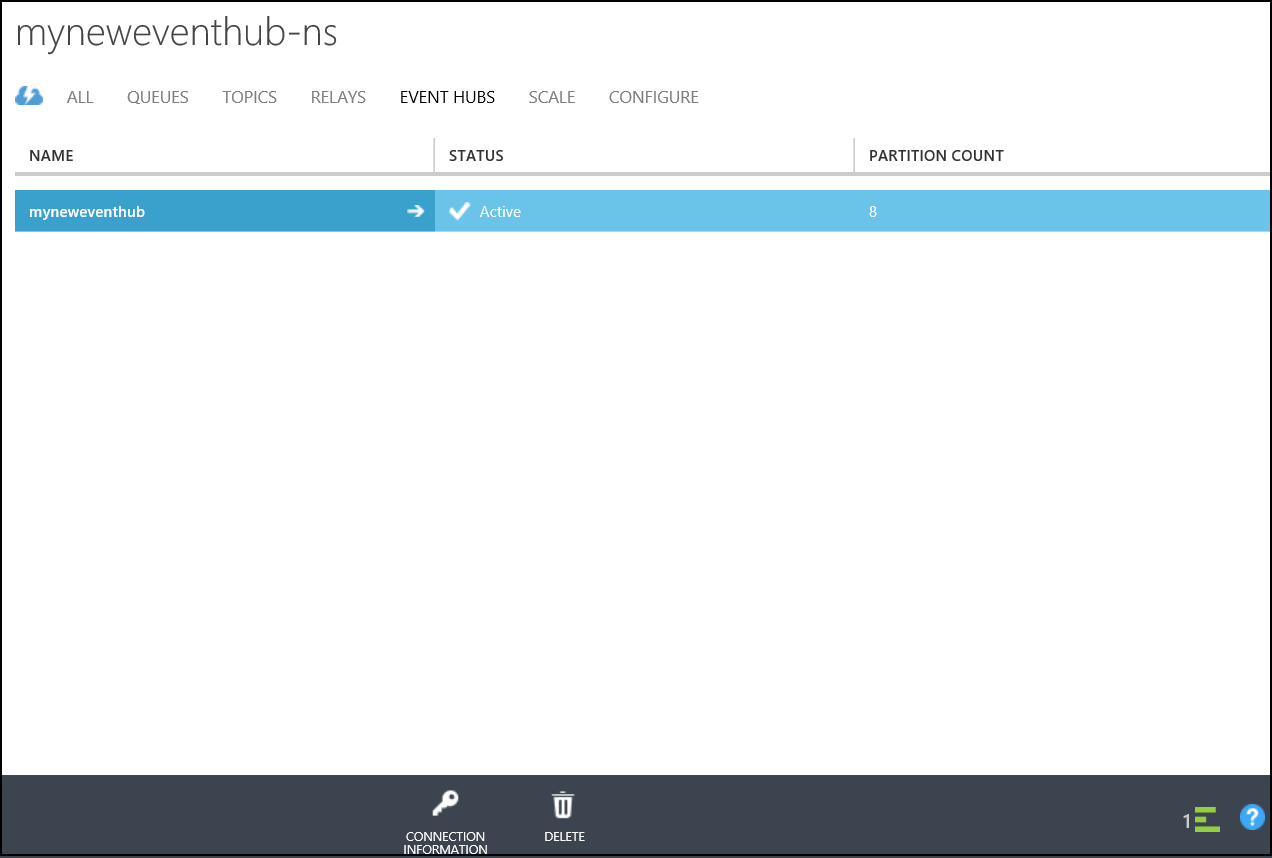
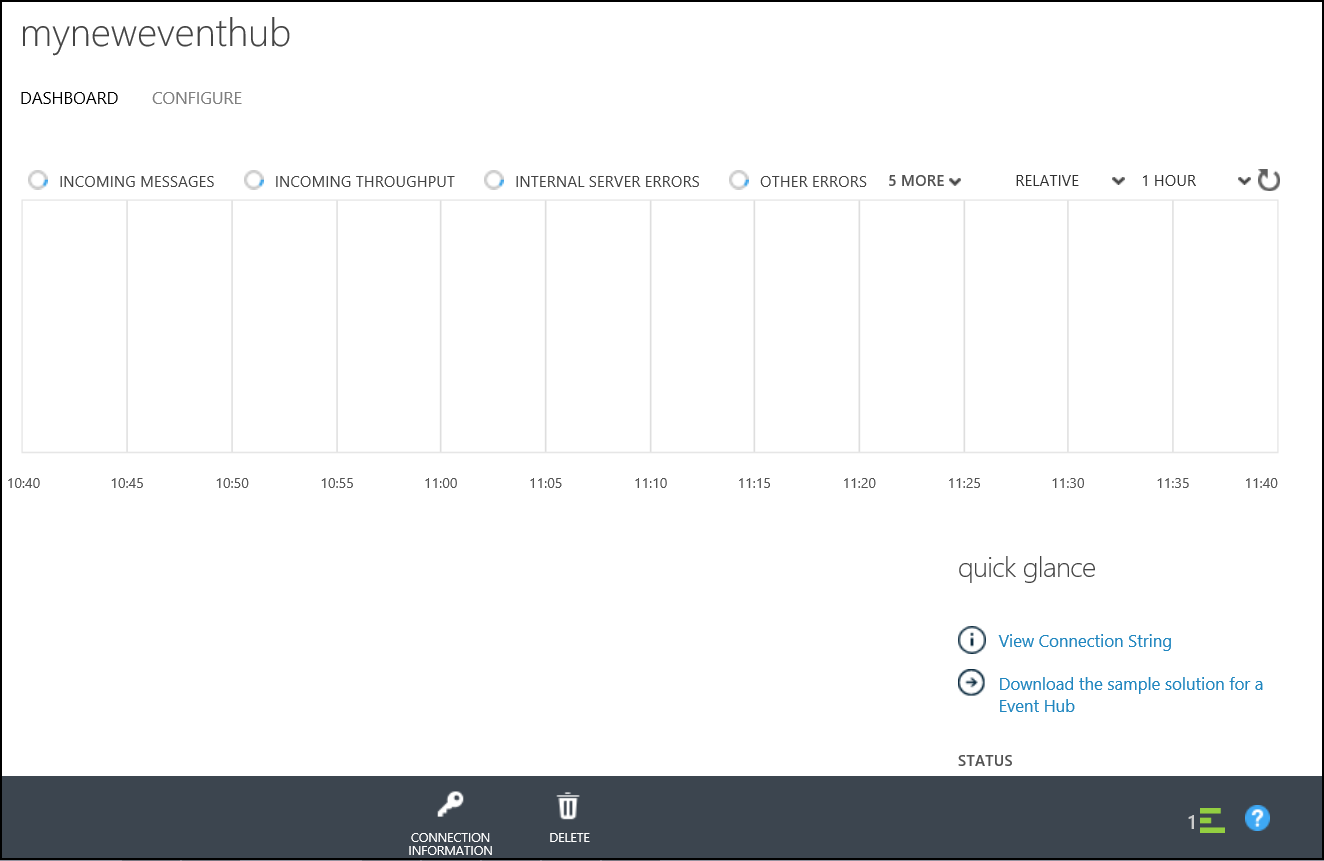
## 2.2 Configure Shared Access Policies

The Event Hub is created, and the next section will configure the new Event Hub to send and receive data.

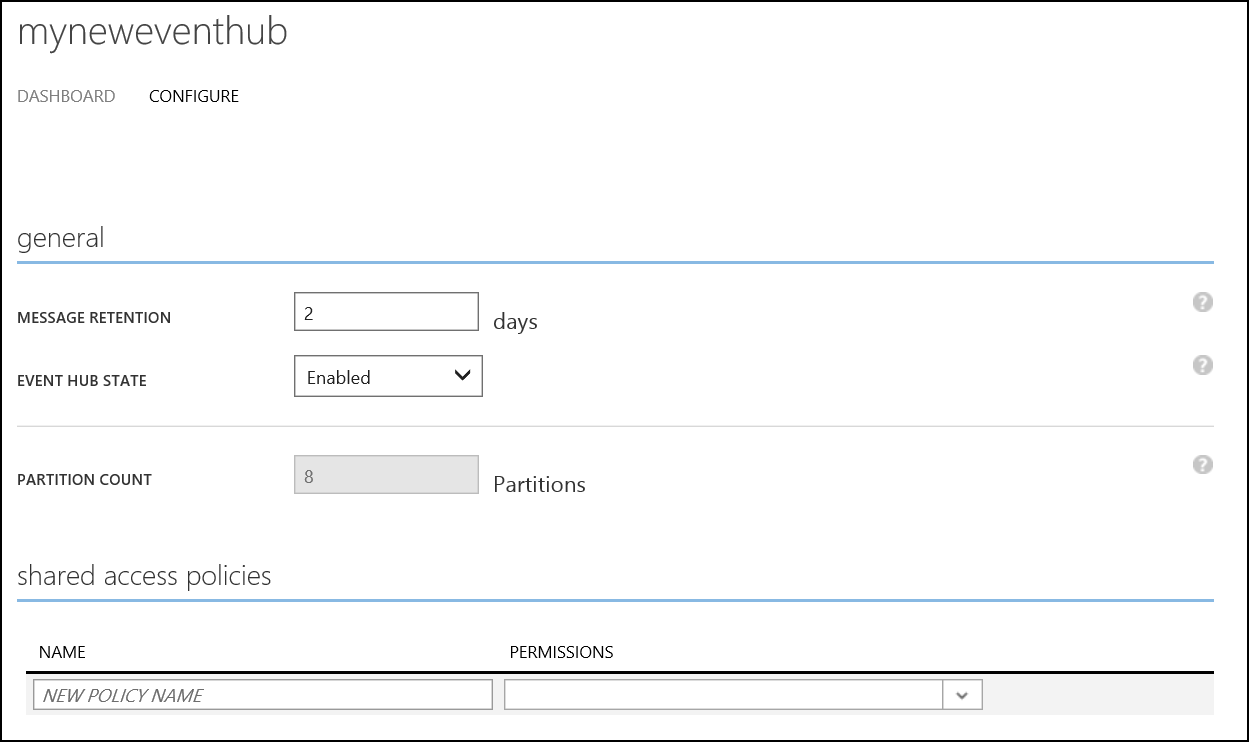
1. Select the new Service Bus entry, this will display the following page.

* 

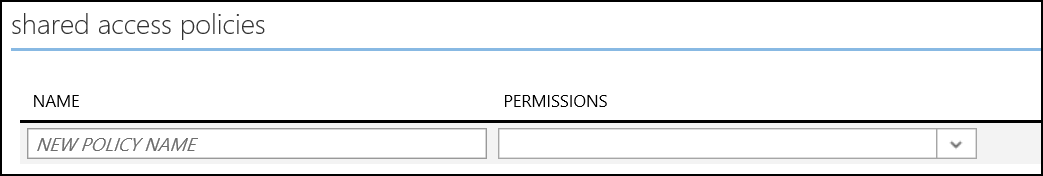
1. Select the Event Hubs tab from the top menu.

* 
* 

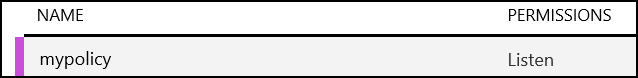
1. Select the configure tab.

* 

1. Locate the shared access policies section at the bottom of the screen.

* 

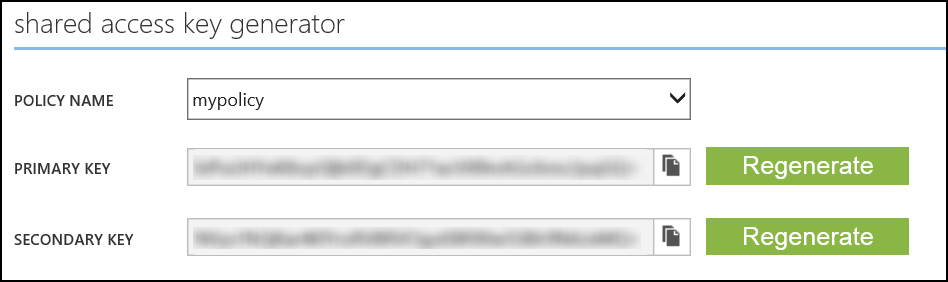
1. Enter **mypolicy** as the new policy name in the name textbox and select **Send** and **Listen** permissions.

* 

1. Click Save at the bottom of the screen.

* createEventHubImg15.png

1. Once the policy creation is complete the bottom of the screen will show the policy details including the keys.

* 

1. The key will be used in several labs during the course. Copy the key and policy name to a file, and save the file to the desktop for later use.

## 3. Device Sender

Device Sender is a small command line program that can be used to send messages to an event hub. It will automatically generate a set of messages over a small timeframe. It simulates 4 device types; Temperature, Energy, Humidity and Light. You should only need to do this once for all subsequent labs as you can re-use this dataset.

1. Open a command prompt. Navigate to the Data Culture Resources and in the IoT Folder find the tools and DeviceSender.exe file

***For example:*** *C:\\Documents\DataCultureSeries\IoT Track\2.CreateResources\tools\DeviceSender*

1. Execute the following command to send messages to the event hub. A copy of this script is available on the OneDrive here:<> or the Github repo here:<> **C:.txt**.
   * <eventHubNamespace> refers to the namespace of the Event Hub created in the previous section.
   * <eventHubName> refers to the name of the Event Hub created in the previous section.
   * <policyName> will be **mypolicy** if the previous directions were followed. If not, enter the name of the policy created in the previous section.
   * <policyKey> is the Event Hub policy key saved to the text file.

* DeviceSender GenerateDataToEventHub -n <eventHubNamespace> -e <eventHubName> -p <policyName> -k <policyKey>

1. The command prompt will show messages sent to the Event Hub. Do not close the prompt as this will stop the messages.