# The Absolute Beginner’s Guide to Windows Azure SQL Database

In virtually every *CODE* article I’ve written since 2004, the proverbial 99% of the content has been based on things I’ve done in production. Well, every technical person has to “go back to school” at some point-and my return to school has been in the last month as I finally took a look at SQL Azure. This article represents my findings as I recently finished “kindergarten” and am now ready for the trials of first grade!

## What I Set Out to Do

I’ll try not to repeat this too many times, but until a month ago, I had not touched Azure. I’ve been reading about it for over a year, and recently received advice that right now is a good time for someone to start looking at it if they haven’t already.

If you have been working on Windows Azure SQL Database and if you’ve built reports using this tool, I seriously doubt you’ll gain anything from this article. This article is purely for those who have not touched Azure and would like a quick roadmap on how to get started. At this stage, all I’ve done is register a free trial Azure account, create a test database, and create a test SSRS report “in the cloud” against that database.

I’m not going to use my standard “Baker’s Dozen” format for this article. I didn’t want to spend any time trying to fit the steps into 13 items.

Before I begin, I want to give a ton of credit to Microsoft’s Peter Laudati who wrote an extended blog entry on Windows Azure that helped me greatly: <http://blogs.msdn.com/b/peterlau/archive/2012/02/28/get-started-with-sql-azure-resources.aspx>.

So if you’re an absolute beginner to Azure SQL Databases, here we go!

### Step 1: Sign Up for a Free Azure Account

First, you need to sign up for a free Azure trial. You can go to <http://www.windowsazure.com/en-us/> and sign up for a free trial. Note that even with a free trial, you’ll either need to provide a credit card (for proof of identity) or associate the account with an MSDN subscription. Note that you’ll need to associate your Windows Live ID with an instance of an Azure trial.

You can run a trial without being charged—if you want to see the latest Azure prices (which have fallen in the last two years) you can go to this site and access the Azure pricing calculator: <http://www.windowsazure.com/en-us/pricing/calculator/>.

### Step 2: Accessing the New Azure Management Portal and Creating a New Azure SQL Database

Once you create your trial account, you can return and launch the Azure Management Portal (**Figure 1**) by going to [http://manage.windowsazure.com](http://manage.windowsazure.com/). Once in the portal, you can create a new empty SQL Azure database by clicking the NEW link at the bottom of the portal. The portal will walk you through **Figures 2–4** to define the database, the authentication, collation sequence, etc.

Figure 1: The new Azure Management Portal dashboard.

Figure 2: New dialog to create a SQL Database.

Figure 3: Dialog to create a new Database (SampleAzure).

Figure 4: Basic authentication for the sample database (make sure to keep checkbox checked on).

After you go through the basic steps to create a new database, you’ll return to the Management Portal with the database server and empty database name (**Figure 5**). SQL Database auto-generates a unique, ten-character alphanumeric server name. You cannot change the name of your SQL Database server.

Figure 5: Management Portal, with new empty database and server name.

SQL Database auto-generates a unique, ten-character alphanumeric server name. You cannot change the name of your SQL Database server.

Eventually you’ll want to access this database from the “outside” – in the main configuration area of the portal, there’s a link to access the connection strings (**Figure 6**).

Figure 6: Available dialog of Connection Strings.

### Step 3: Accessing the SQL Azure Database Remotely

The next big step is to access the empty database from SQL Management Studio. You can use either SQL Server 2012 or SQL Server 2008R1 (Service Pack 1). From there, we can use the SSMS designer to create tables and other database objects. **Figure 7** shows the SSMS Connection Dialog, with the server name from the connection strings back in **Figure 6**.

Unfortunately, the connection doesn’t work (see the error message in **Figure 8**). The reason is because you’ll need to create an IP firewall back in the Azure Management Portal dashboard.

After you create a SQL Azure database, you can’t connect to it back in your local instance of Management Studio until you define an IP firewall rule back in the Azure Management Portal dashboard.

Figure 7: Attempting to connect to the Azure database server from local instance of Management Studio.

Figure 8: First attempt to connect failed – must define an IP firewall rule!

### Step 4: Defining an IP Firewall Rule

From Step 3, in order to access SQL Azure databases, you’ll need to define an IP firewall rule. Back in the Azure Management Portal, you can go to the Azure Database and access the IP Firewall Rules dialog (**Figure 9**) in the Configure Database area. In that dialog, you can click on the link to “Add to Allowed IP Addresses.”

Figure 9: Back in the Management Portal, must set IP firewall rule!

Once you do that and then save (at the bottom of the screen), you can then go back to Management Studio, access the database, and then create any database objects you need. Obviously, the database objects you create are stored back in the Azure database, on the server associated with the connection string.

### Step 5: Creating a Reporting Instance

The next step was to create a simple SSRS project, point to the Azure database, and then deploy the SSRS project to an SSRS Azure instance.

Here was where I had some hair-pulling moments. As it turns out, the Azure Management Portal does not contain any functionality to define a SQL Azure Reporting instance. Fortunately, I found this URL (<http://blogs.msdn.com/b/psssql/archive/2012/11/15/ack-where-the-heck-did-azure-reporting-and-data-sync-go.aspx>), which explains that the dialog to create a reporting instance is in the **OLD** version of the management portal, not the new one.

As it turns out, the new Azure Management Portal does not contain any functionality to define a SQL Azure Reporting instance. The dialog to create a reporting instance is in the OLD version of the management portal, not the new one.

So in order to access the old portal, you can click in the upper right corner of the new portal (near the login option), and you’ll find an option (**Figure 10**) to access the old portal.

This will launch the old version of the portal (**Figure 11**), where you can create a new instance of a SQL Azure reporting service URL subscription and set the administrator name and password (**Figures 12–13**). Once you complete these steps, the old portal (**Figure 14**) shows the Web service URL for the new instance.

Figure 10: In order to create a SQL Azure Reporting implementation, must revert to the “Previous portal.”

Figure 11: The “Previous portal,” to create a new Reporting instance in Azure.

Figure 12: Create a new SQL Reporting Server.

Figure 13: Set Reporting Administrator.

Figure 14: New Reporting Instance, with Web server URL.

Finally, you can create a regular SSRS project with a shared data source that points to the SQL Azure database (**Figures 15–16**). Note that you cannot use Windows Authentication in the data source credentials – you must use the name you provided when you initially defined the Azure database. Next, you can create reports against the shared data source, and then deploy to the Web service URL back in **Figure 14** (by going to the SSRS project properties and providing the Web service URL, as shown in **Figure 17**).

Figure 15: New SSRS project, with Shared Data Source pointing to Azure DB.

Figure 16: Shared Data Source Credentials (Windows Authentication not supported)

Figure 17: Project properties.

Once you deploy the SSRS project, you can access the report using the Web service URL. You can also incorporate the URL into your own custom ASP.NET project.

## Final Thoughts

I have scratched the proverbial “surface of the surface” of Windows Azure and SQL Databases in Azure. I’m building an actual demo project over the next few months and will follow up with a more detailed article. However, I hope I’ve helped people with the absolute basics in getting started with Azure.

Kevin S. Goff

Kgoff@KevinSGoff.NET

Kevin S. Goff has been a SQL Server MVP since 2010, and was a C# MVP from 2005-2009. Kevin has worked in the industry since 1987.