SQL Database tutorial: Create a SQL database in minutes using the Azure portal

By Carl Rabeler (https://github.com/CarlRabeler)
Updated: 04/14/2016

Single database

Azure portal

In this tutorial, you'll learn how to use the Aure portal to:

- Create a SQL Database logical server to host SQL databases
- Create a SQL database with no data, with sanple data or with data fron a SQL database backup.
- Create a server-level firewall rule for a single IP address or for a range of IP addresses.

Use these links to perform these same tasks using either C# (../sql-database-get-started-csharp/) or PowerShell (../sql-database-get-started-powershell/).

Connecting to the Microsoft Azure portal with a subscription

To connect to the Microsoft Azure portal, you must have a subscription.

Get a new account

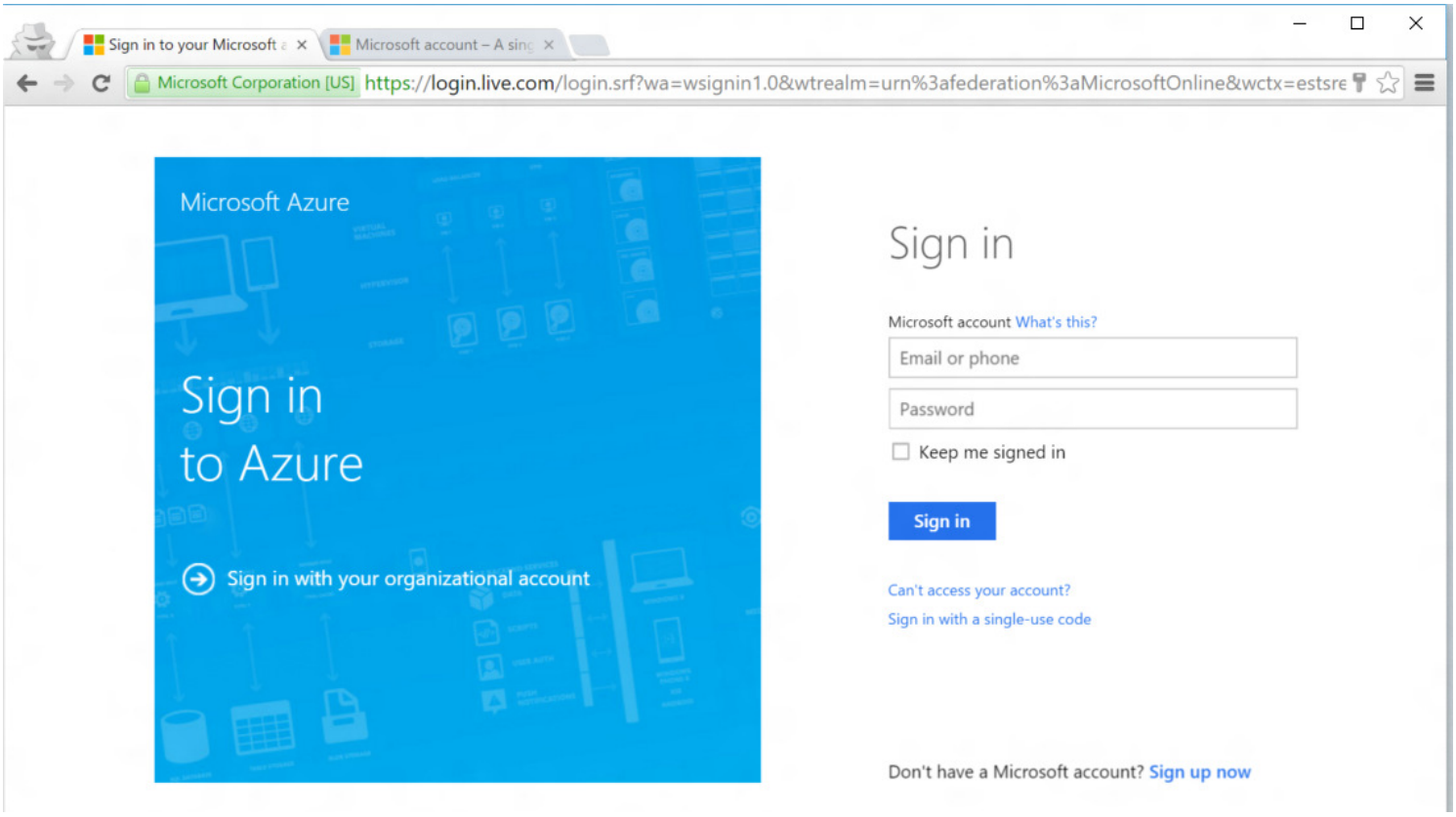
If you do not have an existing Microsoft Azure account, follow one of the links below to obtain an account:

- Get a free account (https://azure.microsoft.com/get-started/)
- Use an MSDN subscription (https://azure.microsoft.com/pricing/member-offers/msdn-benefits/)

Sign in using your existing account

Using your existing subscription (https://account.windowsazure.com/Home/Index), follow the steps below to connect to the Azure portal.

1. Open your browser of choice and connect to the Azure portal (https://portal.azure.com/).
2. Sign in to the Azure portal (https://portal.azure.com/).
3. When the sign in page appears, provide the credentials for your subscription

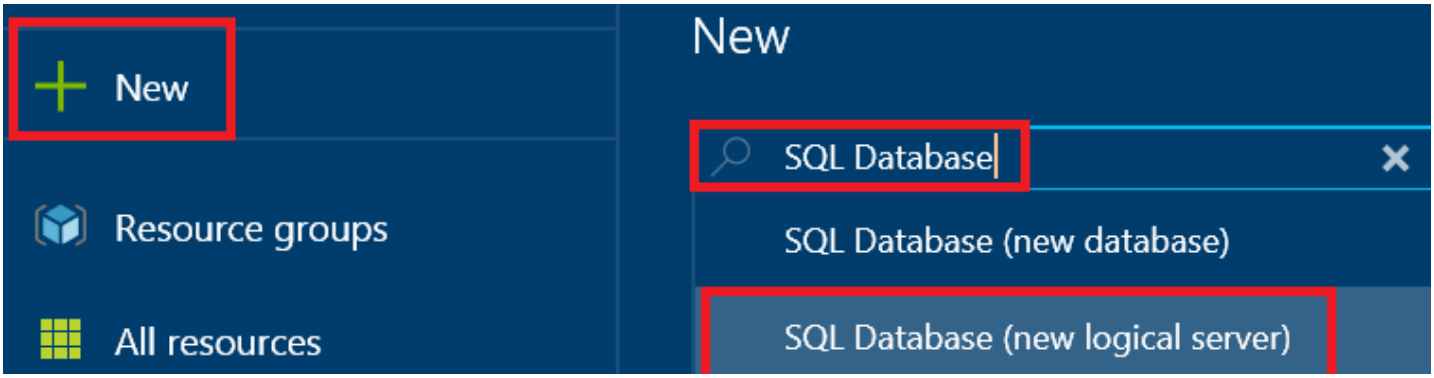


Create an Aure SQL Database logical server

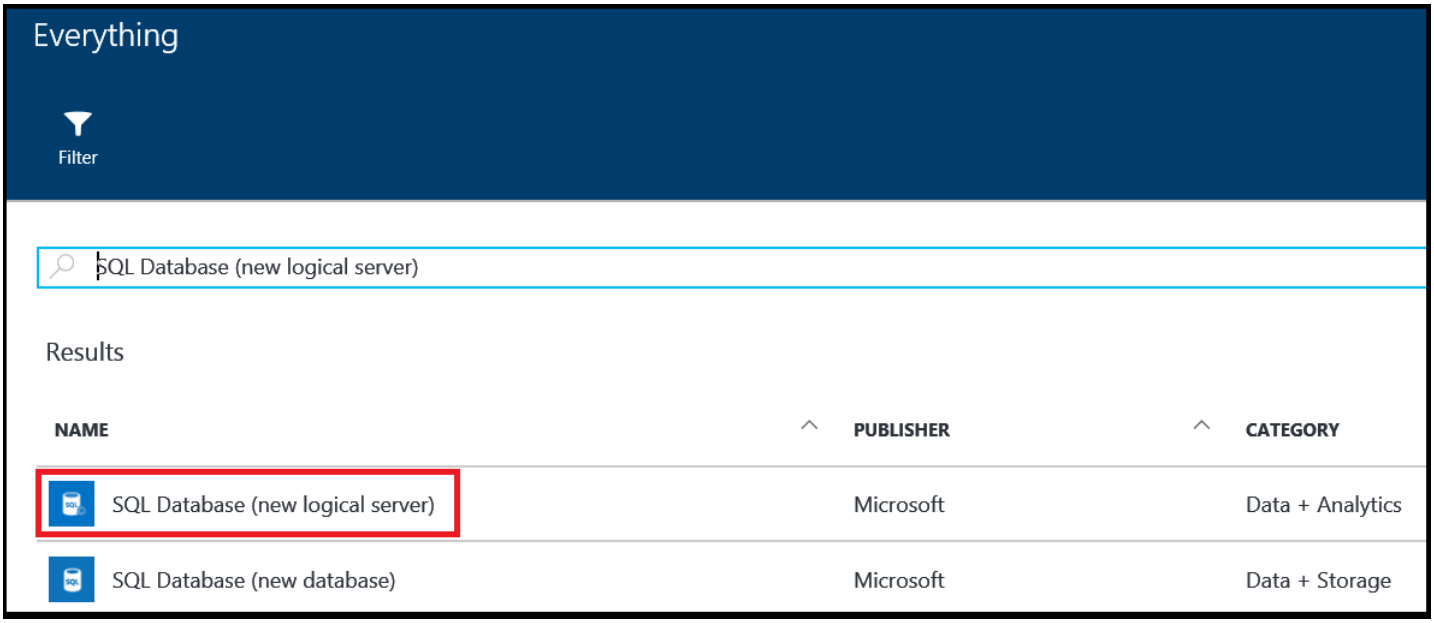
Use the following steps in the Azure portal to create an Azure SQL Database logical server.

1. If not currently connected, connect to the Azure portal (http://portal.azure.com).

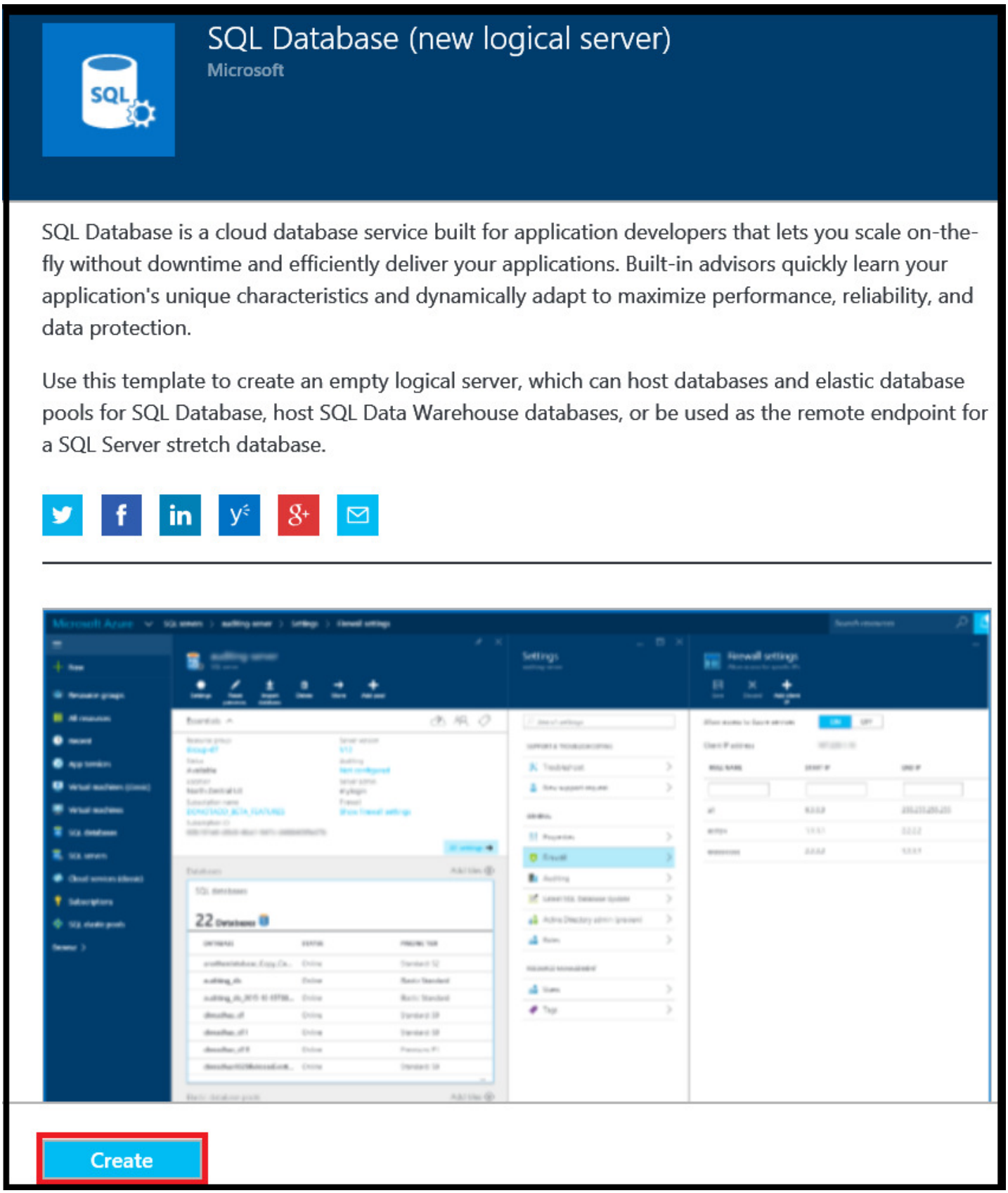
2. Click **New** , type **SQL Database** and then click **SQL Database (new logical server)**



3. Click SQL Database (new logical server).



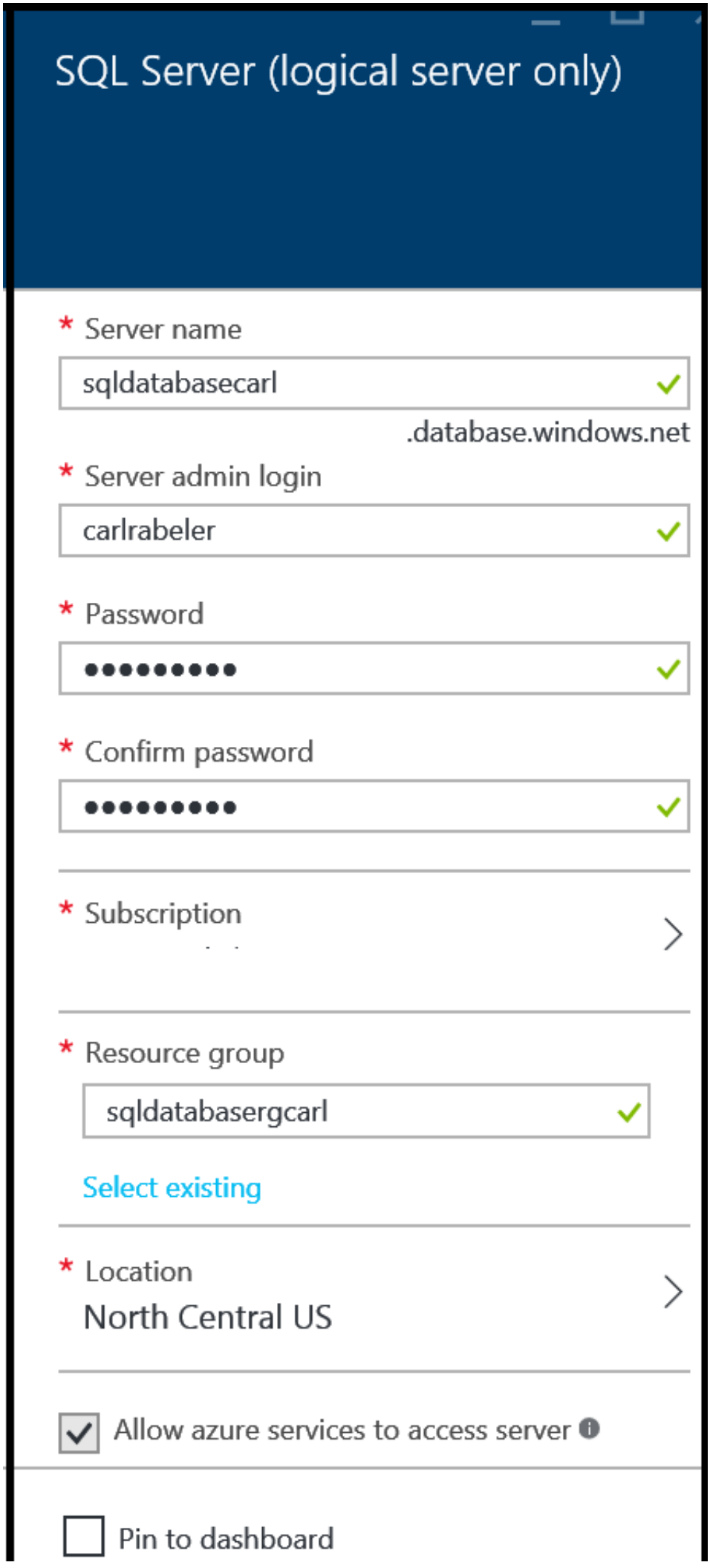
4. Click **Create** to open a template to create an empty logical server that can host single databases and elastic database pools.



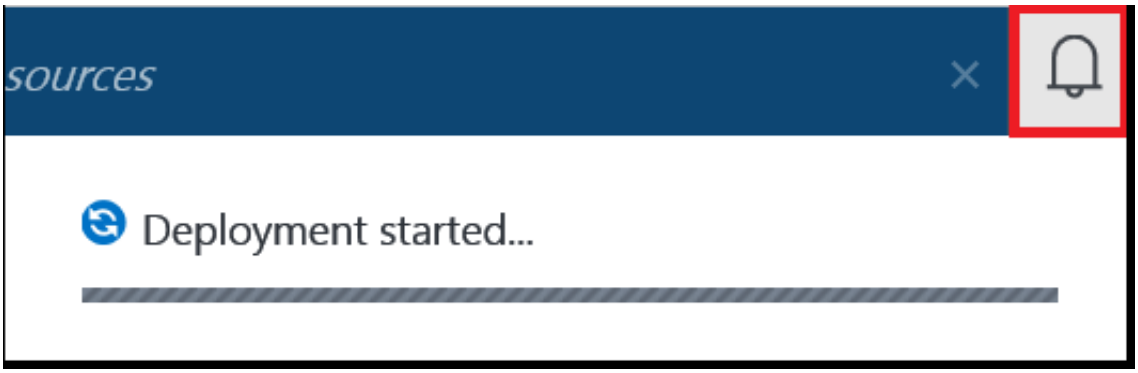
5. Provide the values for the following server properties:

- Server name
- Server admin login
- Password
- Subscription (only if have multiple subscriptions)
- Resource group (new or existing)

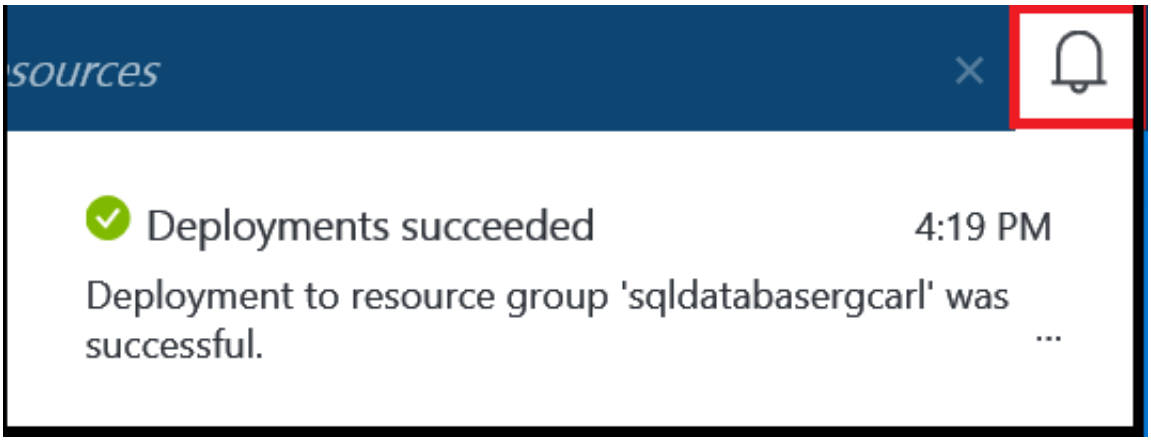
- Location


SQL Server (logical server only)
* Server name
sqldatabasecarl
.database.windows.net
* Server admin login
carlrabeler
* Password
* Confirm password
* Subscription
* Resource group
sqldatabasergcarl
Select existing
* Location
North Central US
Allow azure services to access server
Pin to dashboard

6. Click **Create** and in the notification area, you can see that deployment has started.



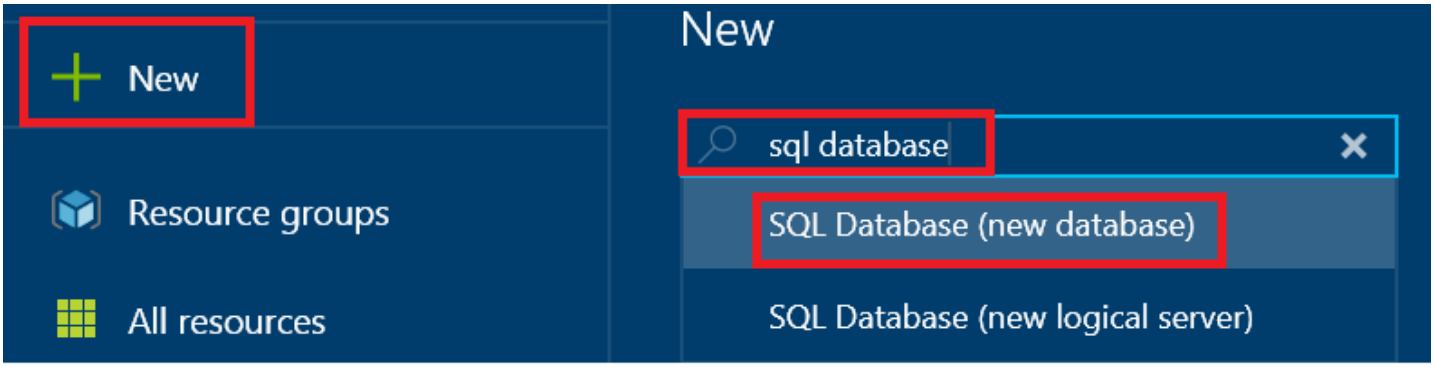
7. Wait for deployment to finish before continuing to the next step.



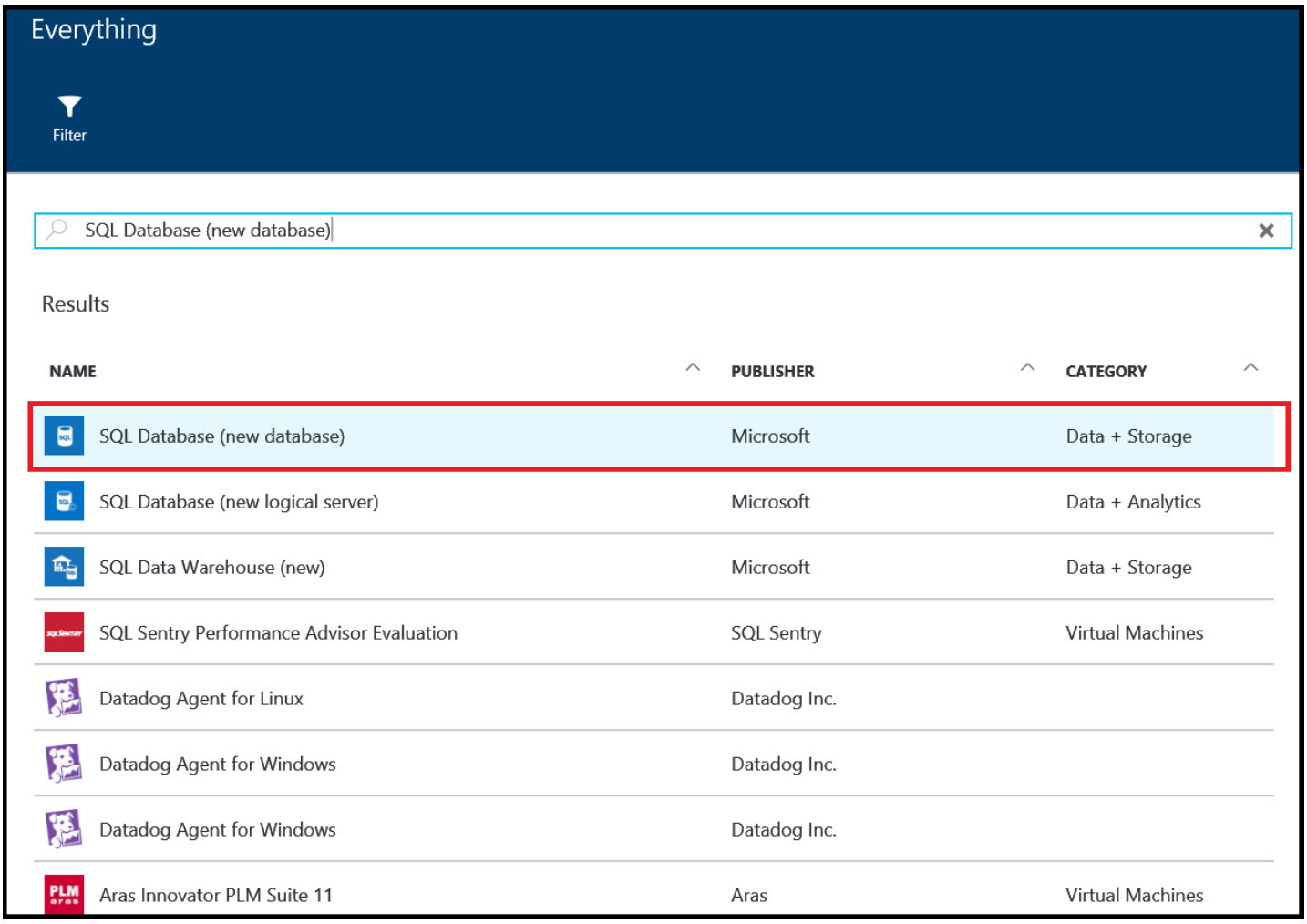
Create a new Aure SQL database

Use the following steps in the Azure portal to create a new Azure SQL database on a new or existing Azure SQL Database logical server.

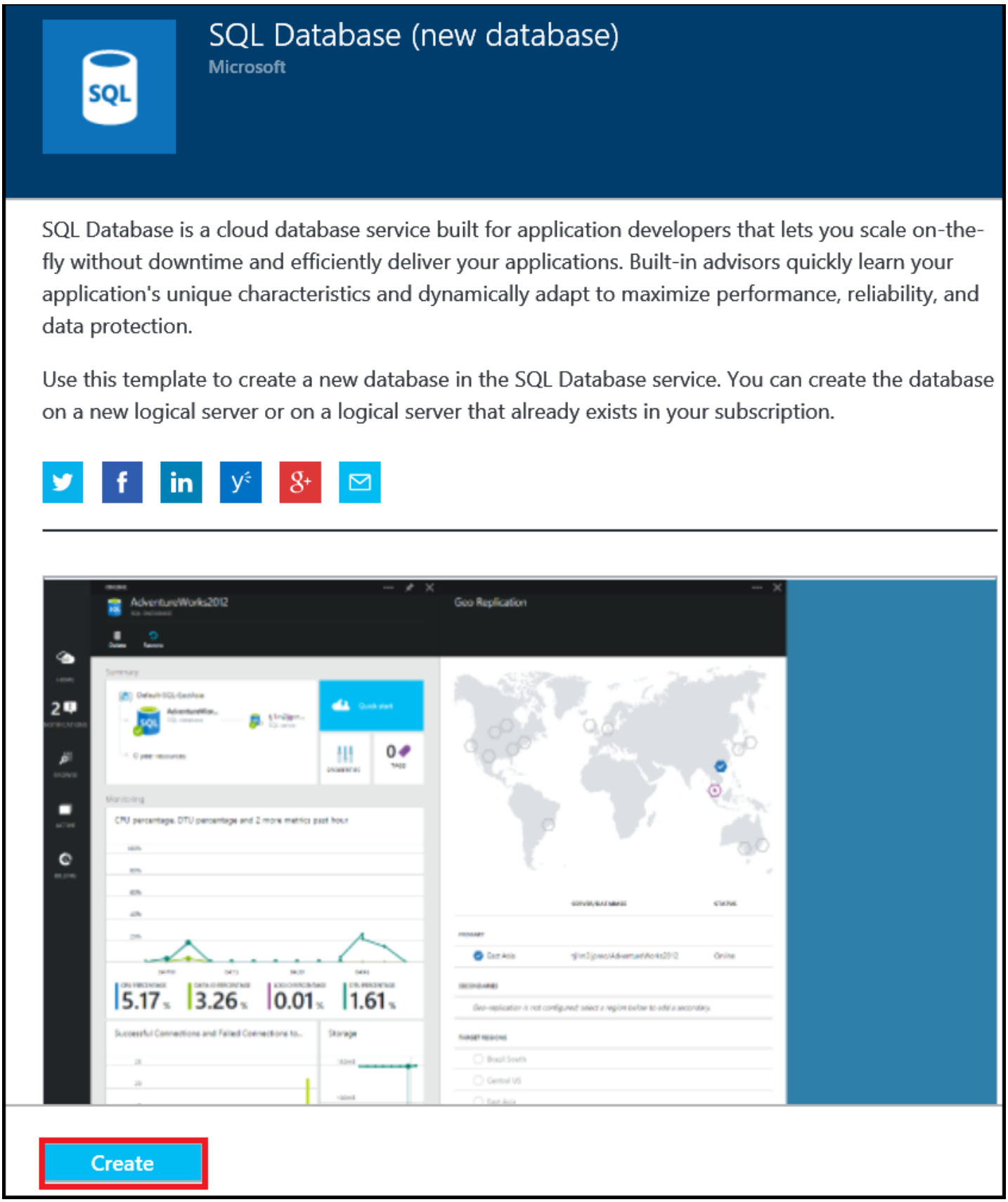
1. If not currently connected, connect to the Azure portal (http://portal.azure.com).
2. Click **New** , type **SQL Database** and then click **SQL Database (new database)**



3. Click SQL Database (new database).



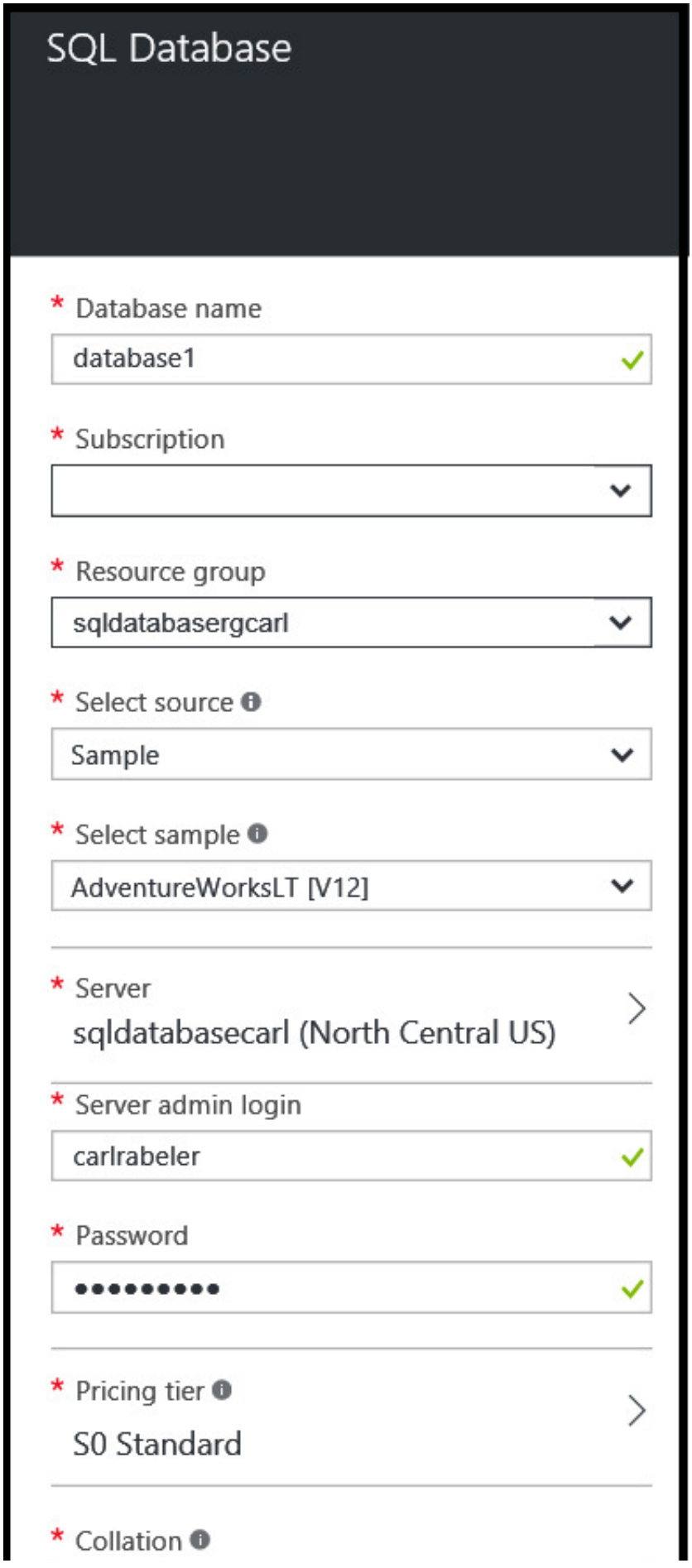
4. Click **Create** to create a new database in the SQL Database service.

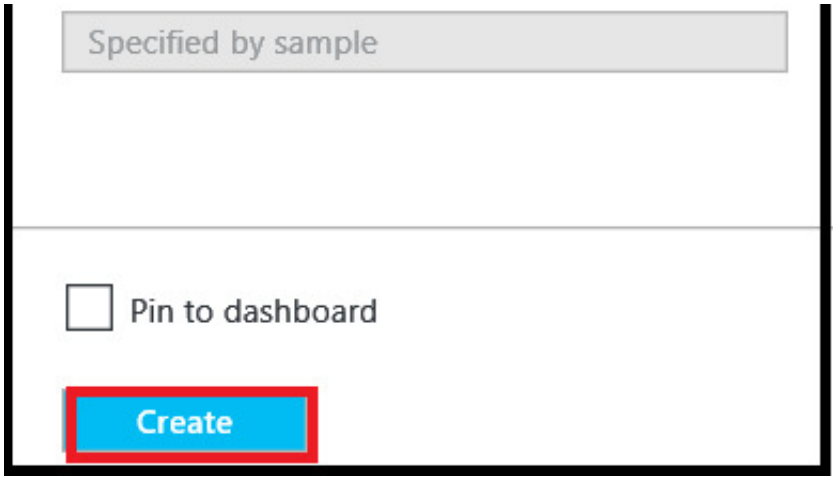


5. Provide the values for the following server properties:

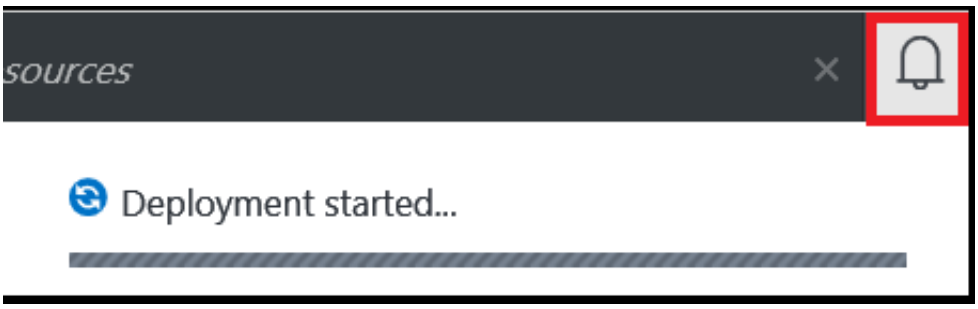
- Database name
- Subscription (only if you have multiple subscriptions)
- Resource group (if just getting started, use the resource group of the logical server)
- Select source (you can choose a blank database, sample data or a database backup)
- Server (a new or existing logical server)

- Server admin password
- Password
- Pricing tier (if just getting started, use the default value S0)
- Collation (only if blank database chosen)

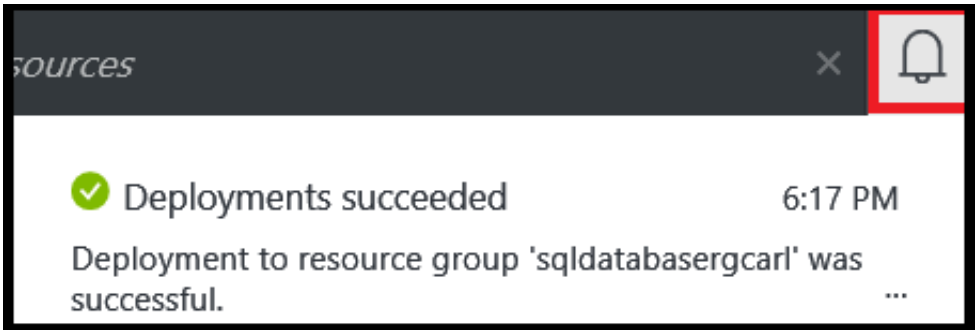




6. Click **Create** and in the notification area, you can see that deployment has started.



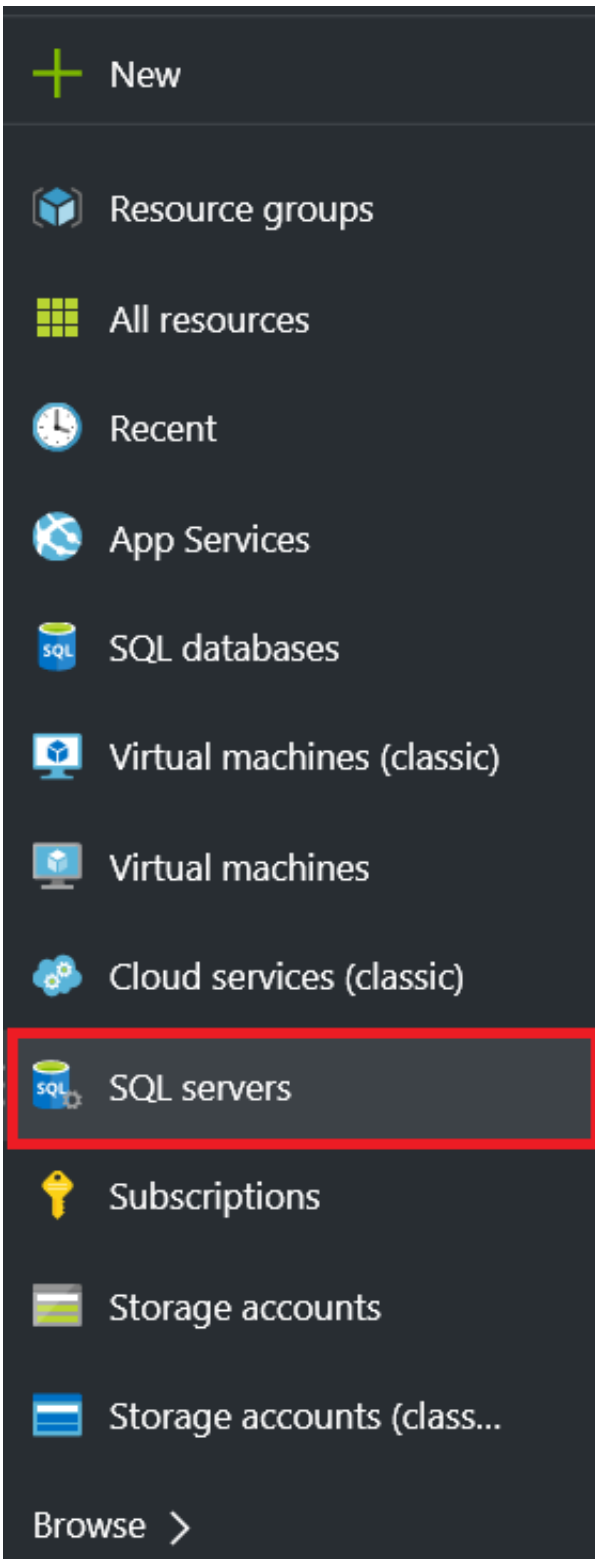
7. Wait for deployment to finish before continuing to the next step.



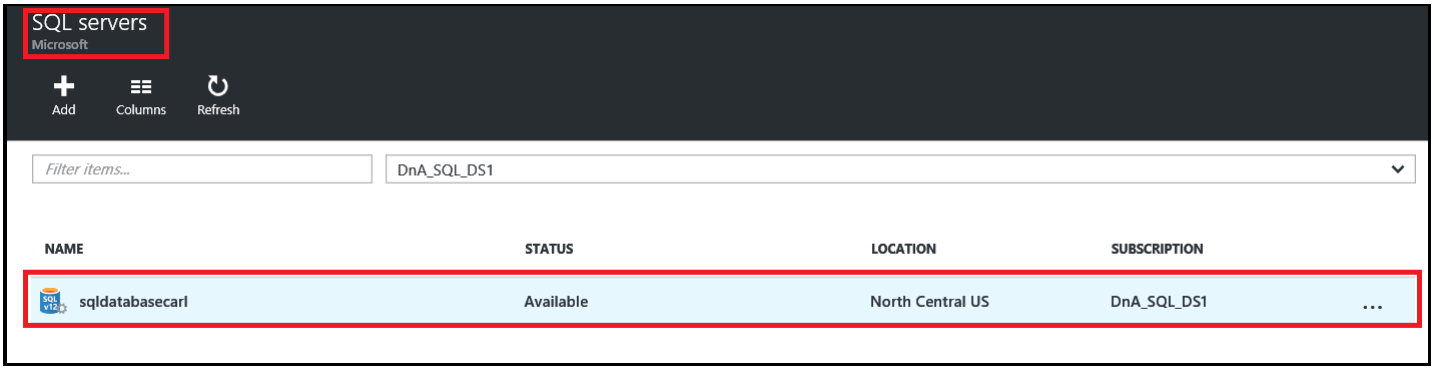
Create a new Aure SQL Database server-level firewall

Use the following steps in the Azure portal to create a server-level firewall rule that allows connections from an individual IP address (your client computer) or an entire IP address range to a SQL Database logical server.

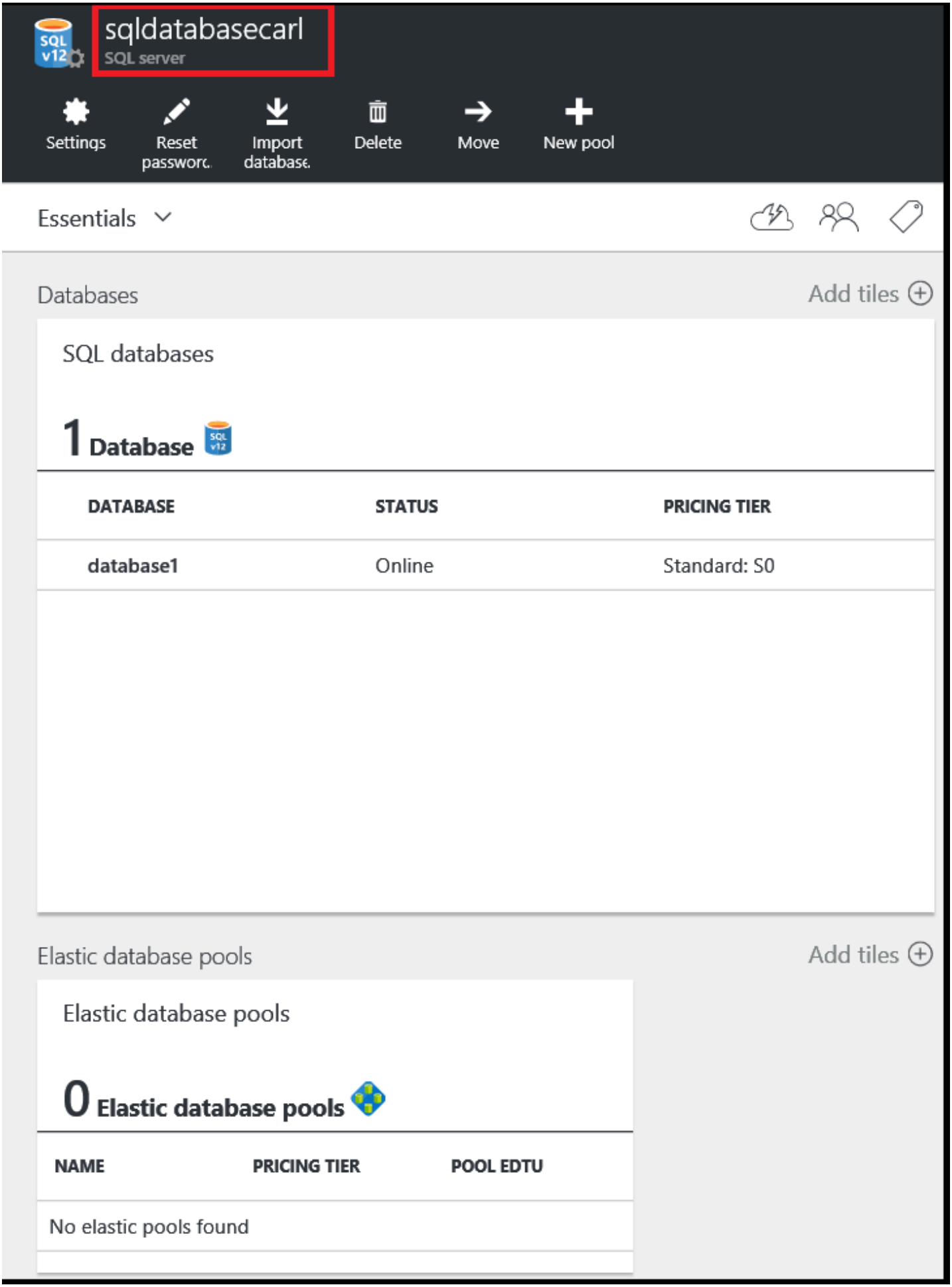
1. If not currently connected, connect to the Azure portal (http://portal.azure.com).
2. In the default blade, click **SQL Server**.



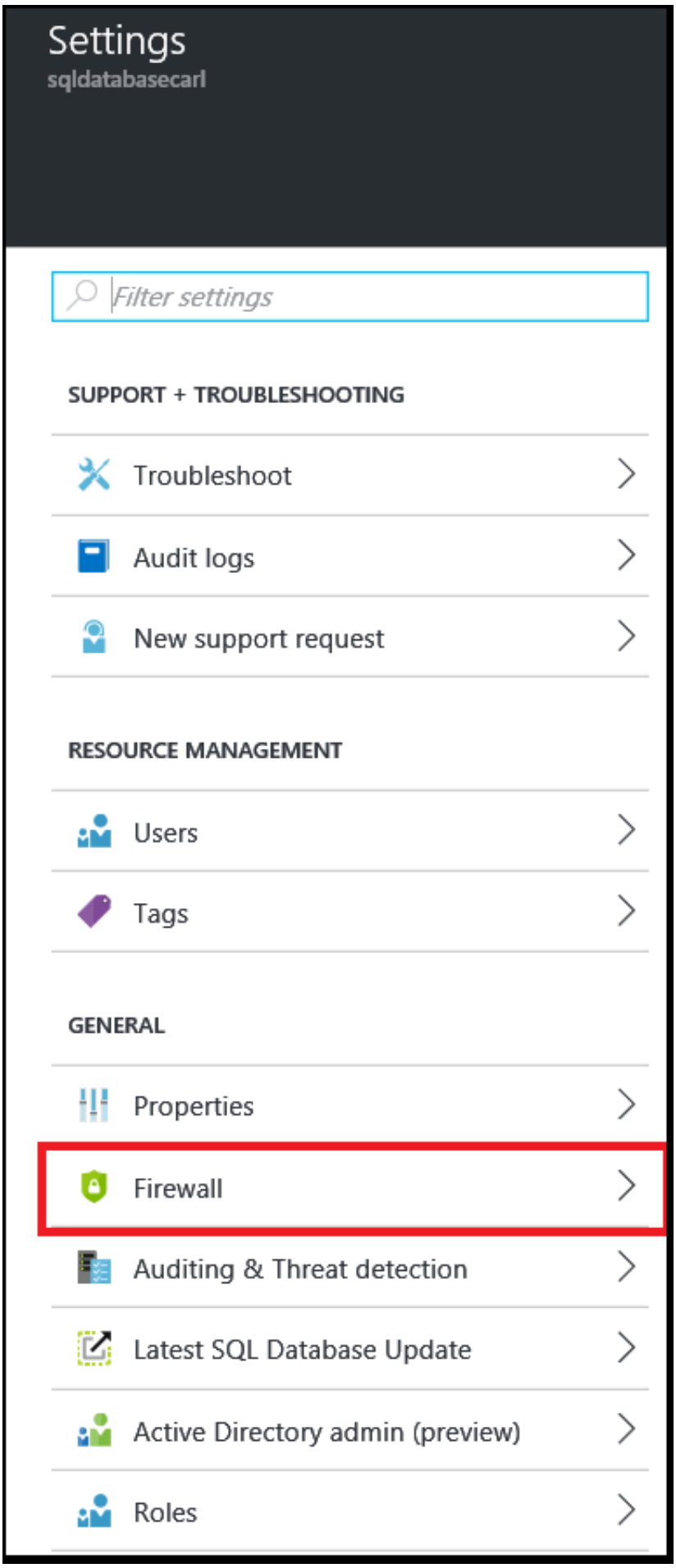
3. In the SQL Server blade, click the SQL Database server on which to create the firewall rule.



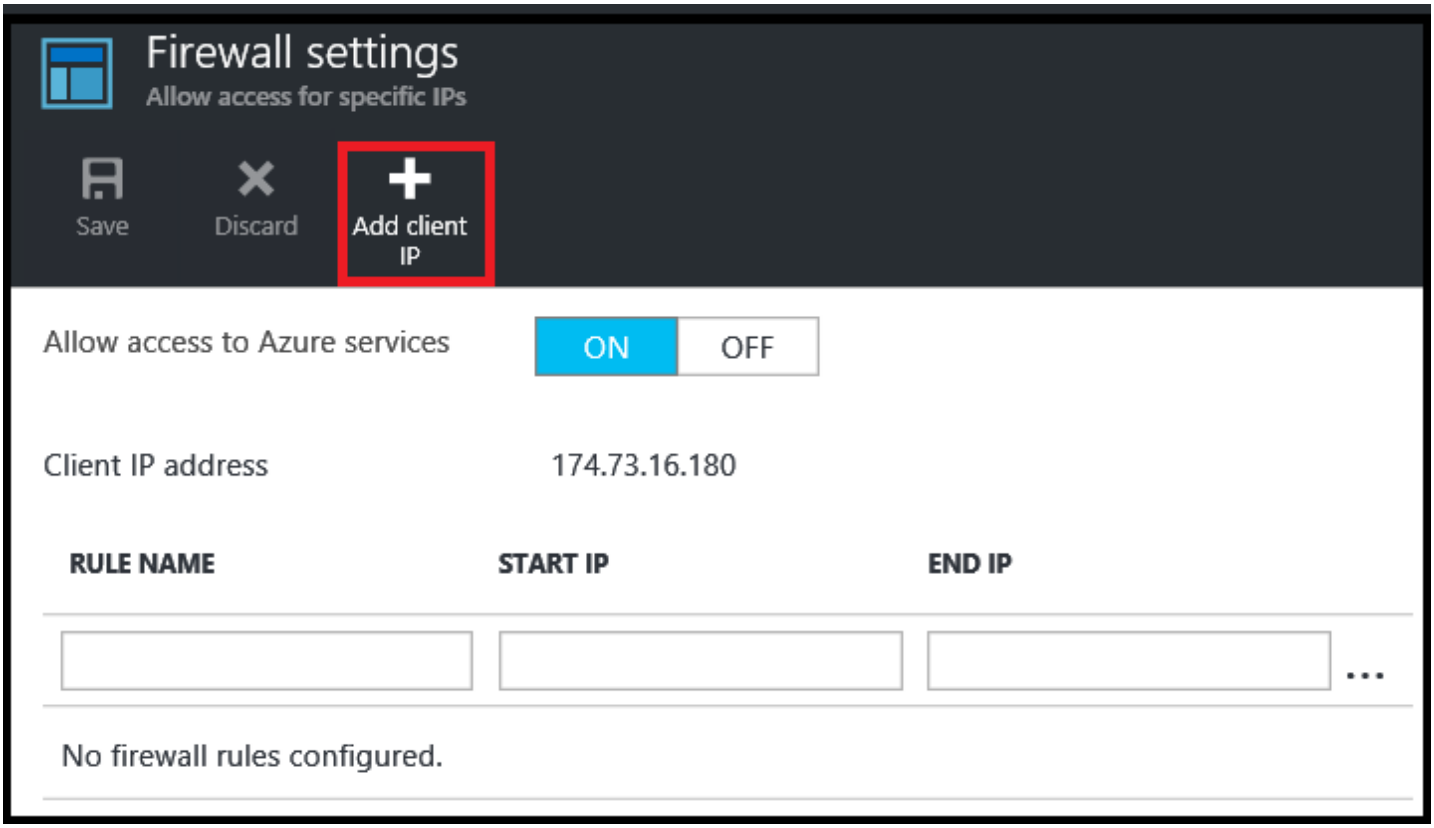
4. Review the properties of your server.



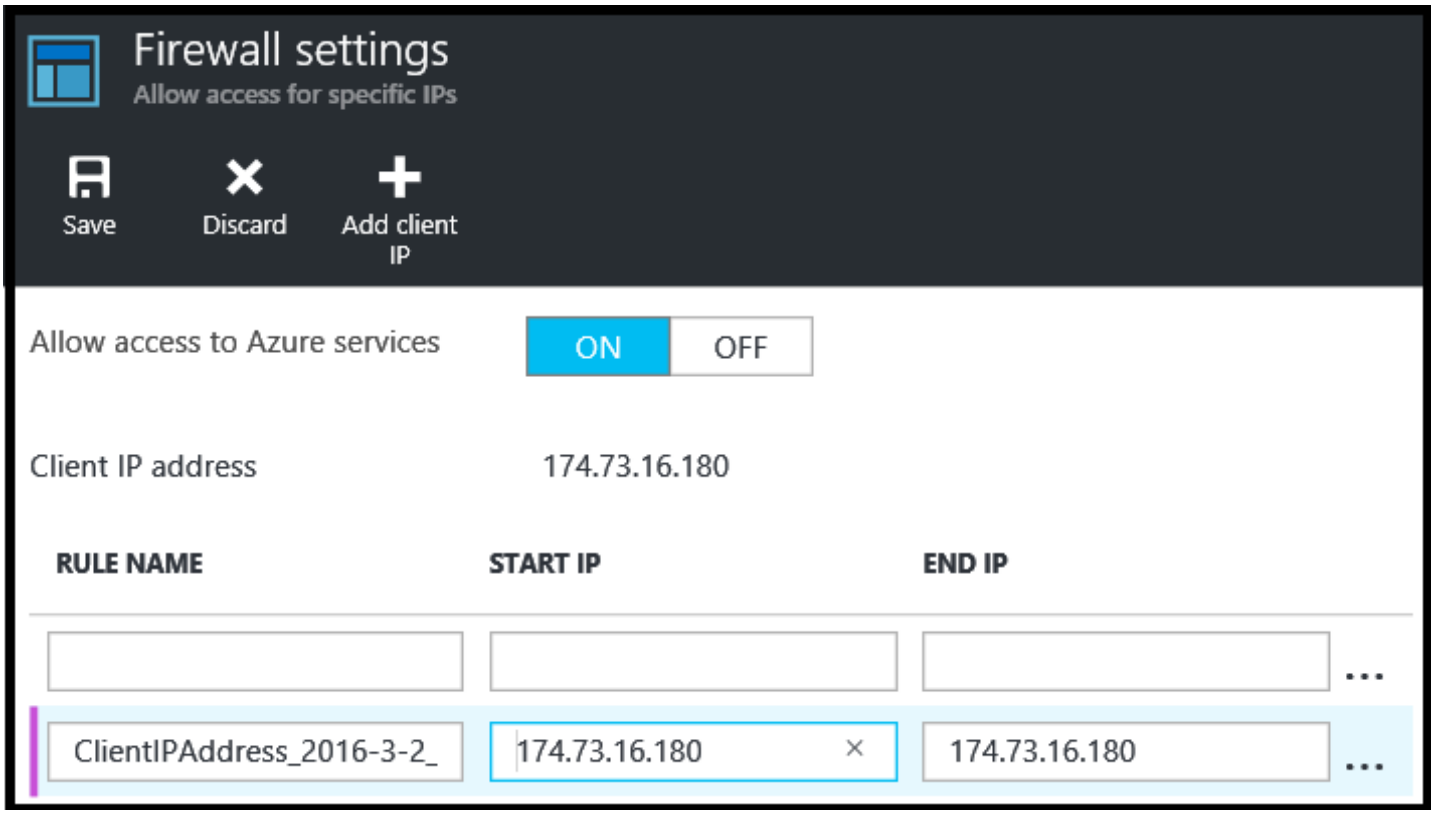
5. In the Settings blade, click **Firewall**.



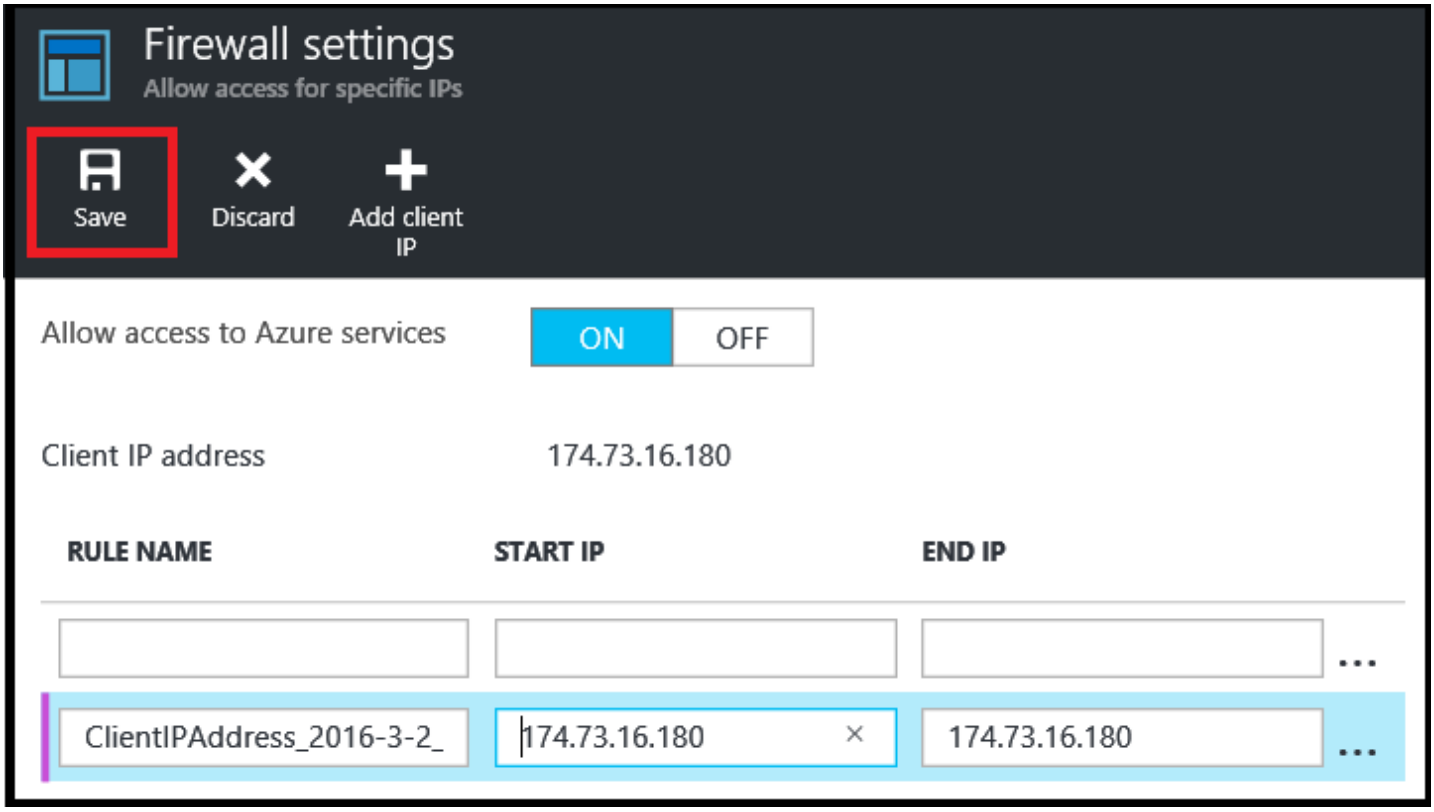
6. Click **Add Client IP** to have Azure create a rule for your client's IP address.



7. Optionally, click the IP address that was added to edit the firewall address to allow access to a range of IP addresses.



8. Click **Save** to create the server-level firewall rule.



Important:

Your Client IP address may change from time to time, and you may not be able to access your server until you create a new firewall rule. You can check your IP address using Bing (http://www.bing.com/search?q=my%20ip%20address), and then add a single IP address or a range of IP addresses. See Manage firewall settings (../sql-database-configure-firewall-settings/#manage-existing-server-level-firewall-rules-through-the-azure-portal) for details.

Next steps

Now that you've completed this SQL Database tutorial and created a database with some sample data, you're ready to explore using your favorite tools.

- If you're familiar with Transact-SQL and SQL Server Management Studio, learn how to Connect and query a SQL database with SSMS (../sql-database-connect-query-ssms/).
- If you know Excel, learn how to Connect to SQL database with Excel (../sql-database-connect-excel/).
- If you're ready to start coding, see Connect and query your SQL database with C# (../sql-database-connect-query/) and Using SQL database from .NET (C#) (../sql-database-develop-dotnet-simple/). See the Quick start code samples to SQL Database (../sql-database-develop-quick-start-client-code-samples/) for Node.js, Python, Ruby, Java, PHP and C++ samples and how-to's in addition to C#.

- If you want to move your on-premises SQL Server databases to Azure, see Migrating a database to Azure SQL Database (../sql-database-cloud-migrate/) to learn more.

Additional resources

SQL Database Overview (../sql-database-technical-overview/)