CHAPTER 7

Understanding Azure Storage and Databases

The computer storage pendulum has swung from people asking, "What would I do with all that storage?" about a decade ago to people now using personal storage media in amounts once only possible for the largest organizations in the world. As you are probably well aware, the average smartphone today has as much storage in it as a PC did in 1997. Data has proliferated in type, size, and audience over the last few years. This data proliferation over the last decade has jokingly been referred to as "datageddon." Data storage and data management have become a real challenge for organizations worldwide, as they struggle to keep up with webcam, voice, e-mail, transaction, and other data feeds that they are required to retain as part of their business's intellectual property or work product.

This chapter discusses the various types of storage available in Microsoft Azure. *Azure Storage* refers to long-term disk storage, not cache or RAM, which is closer to the processor and resides within the hosts running the virtual machines. Another key differentiator is that not all storage is created equal. There are capabilities to be considered, such as drive-access speed and redundancy. Because Azure is created to meet the needs of the enterprise, the type of storage used is enterprise grade as well. Therefore, at a minimum, these are physically 15K spindles and solid-state drives (SSDs) deployed in locally and geographically redundant setups.

Azure Storage is elastic, meaning it can grow and shrink with you, and you're only billed for what you are currently using. Even if a storage account or disk is provisioned at a substantially larger size than what is actually used, only the blocks used are counted for billing purposes.

Azure Storage is available from anywhere there is an Internet connection, like the rest of Azure. Many services already use Azure Storage, such as the Microsoft Windows Server 2012 native backup utility and the Microsoft System Center Data Protection Manager 2012. Azure Storage also uses a system that automatically load-balances your storage based on traffic, just like the most expensive storage area networks (SANs) on the market today. All of these Azure Storage capabilities can be provisioned quickly, and you can have customers up and running using them in about five minutes.

Azure Storage Services: Terms and Concepts

Microsoft's Azure Storage service is available in all Azure datacenters. In general, several types of Azure Storage options are available; they are defined after you create an Azure Storage account, with the exception of databases and StorSimple. The different types of storage are depicted in Figure 7-1.

- *Azure Storage accounts*: An Azure Storage account must be used to create blobs, tables, queues, or file storage services.
- *BLOB (binary large object) storage*: Blobs are generally used to store files accessed via a browser using the HTTP/HTTPS `GET` and `POST` methods.

- *SQL databases*: A SQL database is a type of storage used for structured data. But instead of managing a SQL server, Azure SQL Database is an alternative for consuming the database without having to manage the SQL server that is providing the service. This is different from consuming SQL Server as an IaaS, because in the latter scenario, you still need to manage and maintain the SQL Servers in IaaS.
- *Queues*: As the name implies, a queue provides messaging and communication between different applications. Queue storage provides such a mechanism without having to maintain a separate messaging bus.
- *Tables:* Azure Table Storage is a NoSQL key-value store that is useful for applications that must store large amounts of non-relational data without schemas. This is different from Azure databases in that table storage does not provide any way to represent relationships between data. It is a low-cost way to provide a fault-tolerant store for structured data that may not require the complexities of a relational database. For more information about the differences between Azure Table Storage and Azure SQL Database, see https://msdn.microsoft.com/library/azure/jj553018.aspx.
- *File storage*: File storage provides the ability to access files via standard SMB 2.1.
- *Hybrid storage (StorSimple)*: Microsoft acquired StorSimple, which is a storage solution that deploys on-premises 2U devices that use cloud storage for overflow, backup, and archiving capabilities. The StorSimple device is an ISCSI unit with SSDs and traditional drives for fast, low-latency access to frequently used files.

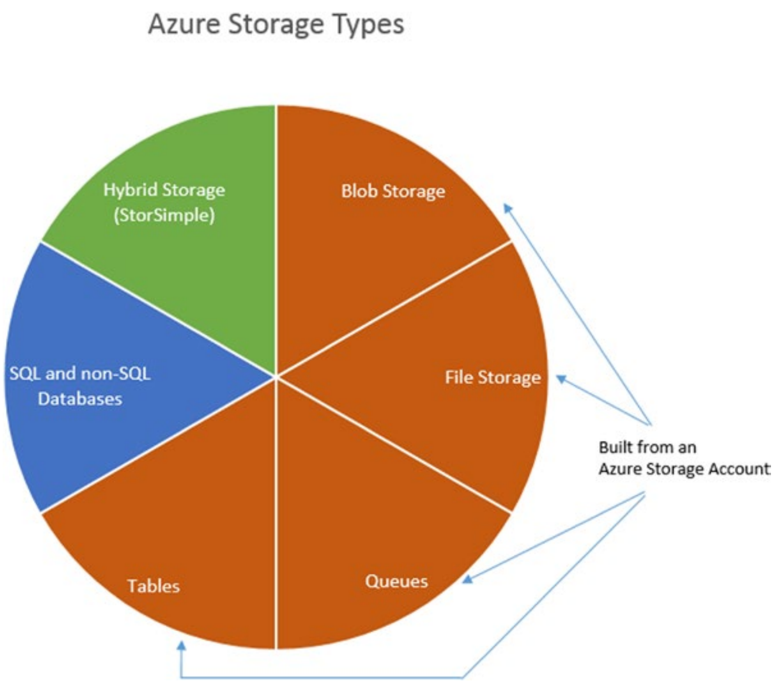

Azure Storage Types
Hybrid Storage (StorSimple)
Blob Storage
SQL and non-SQL Databases
File Storage
Tables
Queues
Built from an Azure Storage Account

Figure 7-1. *Types of Azure storage*

Azure Storage Account

Before going into the details of the different types of storage, it is important to note that with the exception of StorSimple and Azure SQL Database, all storage types are created from an Azure Storage account. The Azure Storage account in turn determines certain characteristics for the storage, such as whether the storage is locally redundant or georedundant, and whether the storage is based on standard hard drives or SSDs.

Premium Storage

Azure Storage is generally provisioned on enterprise-class spinning hard drives. For higher performance, Azure offers Premium Storage. Premium Storage is provisioned on SSDs. As a result, Premium Storage delivers high-performance, low-latency disk support for I/O intensive workloads.

Locally Redundant Storage

Locally redundant storage (LRS), which is the base-level redundancy for storage, ensures that hardware is provisioned on three separate replicas in the same datacenter. This guarantees that no localized hardware failure results from an interruption in data services.

Georedundant Storage

Georedundant storage (GRS) takes LRS storage in one datacenter and replicates it to LRS in another datacenter that is more than 500 miles away. This ensures true georedundancy. Thus if an entire datacenter becomes unavailable, there is no interruption to data services because the second datacenter can continue providing services. Technically, GRS has six replicas of the data (LRS × 2).

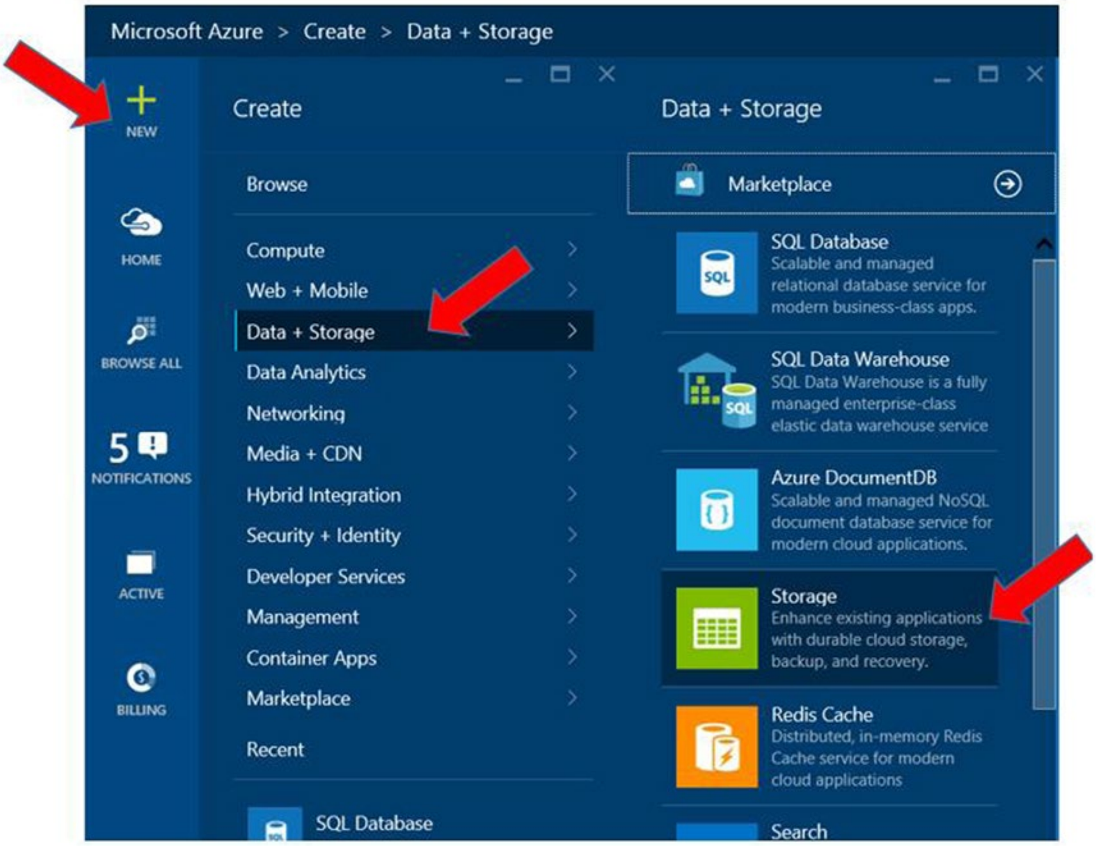
CREATING AN AZURE STORAGE ACCOUNT

The following exercise walks you through the steps required to create an Azure Storage account. As part of this exercise, you see the different Azure Storage accounts and the storage types that each Azure Storage account supports:

■ **Note** Premium Storage is only offered as an LRS option and only supports page blobs, whereas non-Premium Storage provides LRS and GRS options and supports block and page blobs as well as tables and queues.

---

1. Go to the new Azure Management Portal at `https://portal.azure.com`.
2. Click New, select Data + Storage, and then select Storage, as shown in Figure 7-2.



SQL Database
Scalable and managed relational database service for modern business-class apps.

SQL Data Warehouse
SQL Data Warehouse is a fully managed enterprise-class elastic data warehouse service

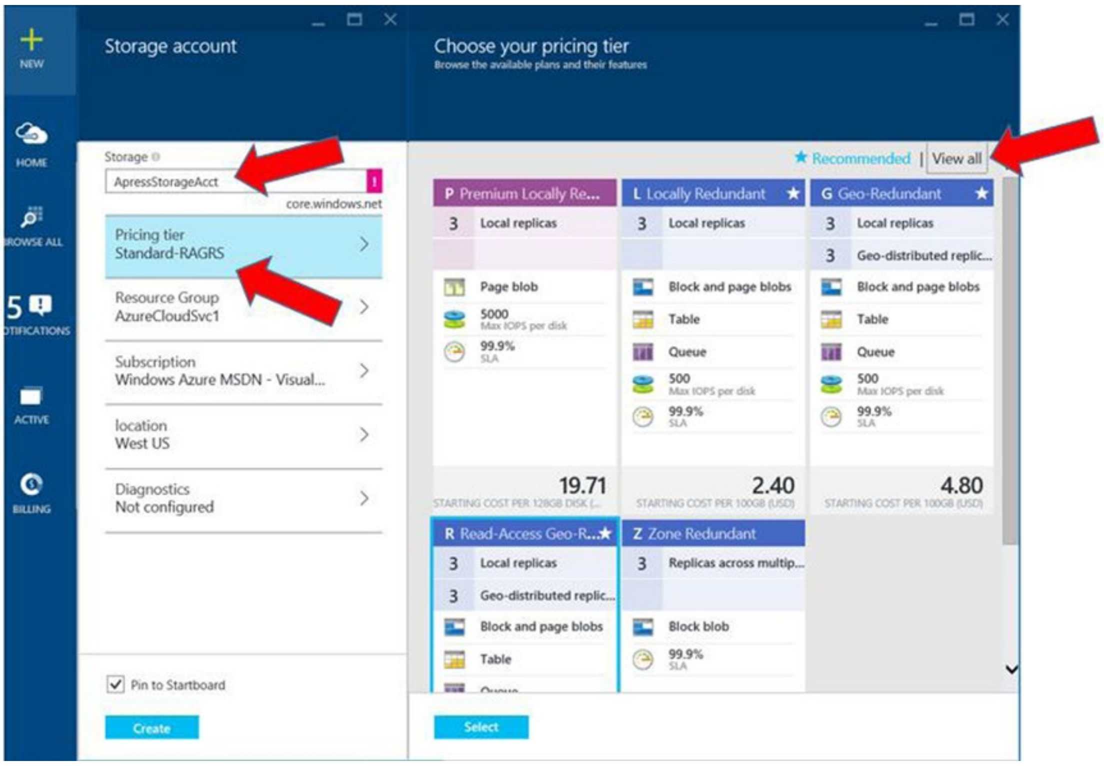
Azure DocumentDB
Scalable and managed NoSQL document database service for modern cloud applications.

Storage
Enhance existing applications with durable cloud storage, backup, and recovery.

Redis Cache
Distributed, in-memory Redis Cache service for modern cloud applications

Figure 7-2. *Creating an Azure Storage account*

3. Provide a name for the Azure Storage account, and then click the Pricing Tier option, as shown in Figure 7-3.



Storage account

Storage
ApressStorageAcct
core.windows.net

Pricing tier
Standard-RAGRS

Resource Group
AzureCloudSvc1

Subscription
Windows Azure MSDN - Visual...

location
West US

Diagnostics
Not configured

☑ Pin to Startboard

Create

Choose your pricing tier
Browse the available plans and their features

★ Recommended | View all

P Premium Locally Re...
3 Local replicas
Page blob
5000 Max IOPS per disk
99.9% SLA
19.71 STARTING COST PER 128GB DISK (...

L Locally Redundant ★
3 Local replicas
Block and page blobs
Table
Queue
500 Max IOPS per disk
99.9% SLA
2.40 STARTING COST PER 100GB (USD)

G Geo-Redundant ★
3 Local replicas
3 Geo-distributed replic...
Block and page blobs
Table
Queue
500 Max IOPS per disk
99.9% SLA
4.80 STARTING COST PER 100GB (USD)

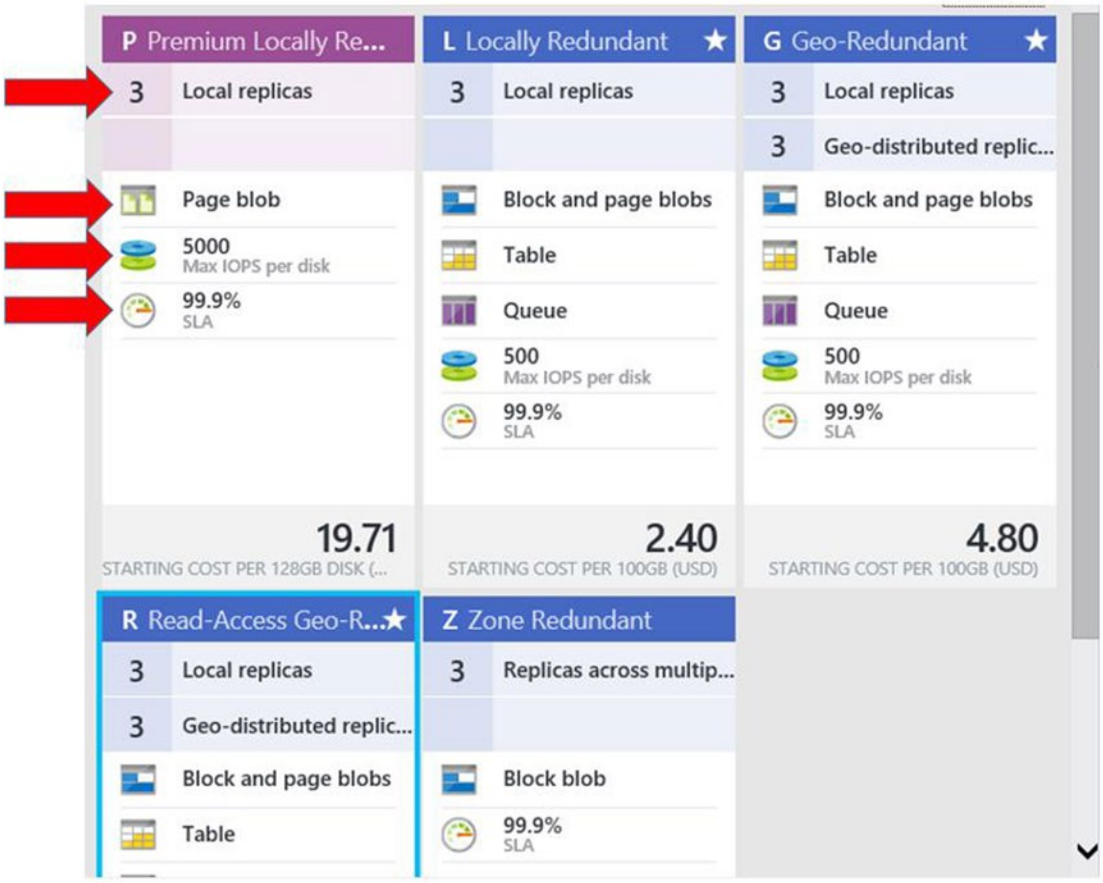
R Read-Access Geo-R...★
3 Local replicas
3 Geo-distributed replic...
Block and page blobs
Table
Queue

Z Zone Redundant
3 Replicas across multip...
Block blob
99.9% SLA

Select

Figure 7-3. *Selecting the Azure Storage account type via the Pricing Tier option*

4. Click View All at upper right to see all available Azure Storage accounts, also shown in Figure 7-3.
5. Click Select to choose the pricing tier. Note the different types of accounts. For example, Premium Storage is only available in LRS and only supports page blobs with a 99.9% SLA, as shown in Figure 7-4. After selecting the type of account, click Create to create the Azure Storage account.



P Premium Locally Re...
3 Local replicas
Page blob
5000 Max IOPS per disk
99.9% SLA
19.71 STARTING COST PER 128GB DISK (...

L Locally Redundant ★
3 Local replicas
Block and page blobs
Table
Queue
500 Max IOPS per disk
99.9% SLA
2.40 STARTING COST PER 100GB (USD)

G Geo-Redundant ★
3 Local replicas
3 Geo-distributed replic...
Block and page blobs
Table
Queue
500 Max IOPS per disk
99.9% SLA
4.80 STARTING COST PER 100GB (USD)

R Read-Access Geo-R...★
3 Local replicas
3 Geo-distributed replic...
Block and page blobs
Table

Z Zone Redundant
3 Replicas across multip...
Block blob
99.9% SLA

***Figure 7-4.** Characteristics of the different Azure Storage accounts*

Now that you have created an Azure Storage account, you can use it to build block or page blobs, tables, queues, and file storage.

Queue Storage

Queues are used primarily for messaging between services and applications. Azure Queue is part of the Azure Storage family of services and uses REST-based protocols such as GET and PUT to transfer messages.

Azure provides two types of queue methods: Azure queues and service bus queues. Although both methods are messaging services, they have different features and capacities in terms of message size, maximum queue size, and message time-to-live (TTL). Application and service developers primarily use Azure queues, which are beyond the scope of this book. For detailed information about Azure queues and service bus queues, and the appropriate use-case scenarios for each, visit `https://azure.microsoft.com/en-us/documentation/articles/service-bus-azure-and-service-bus-queues-compared-contrasted`. You can also find a detailed discussion of Azure Queue storage at `http://azure.microsoft.com/en-us/documentation/articles/storage-dotnet-how-to-use-queues`.

File Storage

Azure File storage offers shared storage for applications using the standard Server Message Block (SMB) 2.1 protocol. Azure VMs and cloud services can share file data across application components via mounted shares, and on-premises applications can access file data in a share via the Azure File storage API. Azure VMs can connect to Azure Storage by simply mounting the file storage as a shared drive via the SMB protocol. Multiple clients can access the file storage via SMB simultaneously.

You can also achieve high-performance, low-latency file storage by building the file storage on the new Azure Premium Storage, which is based on SSD drives.

Blob Storage

Azure Blob storage is designed for storing unstructured data, such as data from backups, rich content such as images and streaming video or audio, text, or binary data. It is designed to be accessed over the HTTP or HTTPS protocol, which makes it ideal for anywhere access over the Internet. Because Azure Blob storage uses the HTTP/HTTPS standards, many third-party solutions use this storage option to offload the data that they generate.

■ **Note** Although Azure Blob storage is usually accessed via a URL using the HTTP/HTTPS protocol, you can also access the contents via a number of APIs/SDKs that the Azure Team has created. These include .NET, Java, PHP, Ruby, and Python. For comprehensive documentation on using these SDKs/APIs, visit `http://azure.microsoft.com/en-us/documentation`.

The following are the core concepts of Azure Blob storage:

- There are two types of blobs:
    - *Block blob*: Comprises data blocks that are identified by their unique block ID. Blocks can be of different sizes, with a maximum of 4 MB per block. There is a maximum size of 200 GB for block blobs. As such, block blobs are ideal for large, efficient uploads; hence their use in backups. Files of up to 64 MB can be written in single-write operations.
    - *Page blob*: As the name implies, essentially a collection of 512-byte page files. A page blob can grow to a maximum size of 1 TB.
- You may define a blob as a block or page blob at the time of creation, but once it has been created, the blob type cannot be changed.
- All blobs reflect committed changes immediately.
- Blobs can be duplicated in a snapshot.
- To limit the risk of a blob being unintentionally overwritten, you can lease it for exclusive write access. When you do so, a lease ID is generated. Henceforth, any write requests that do not include the current or correct lease ID are not allowed to modify the blob's existing contents.
- Management of Azure Blob storage is done through an Azure Storage account, covered in the next section of this chapter.

- Azure Blob storage must reside in *containers*, which are groupings of Azure blobs. Containers can have unlimited blobs; likewise, Azure Storage accounts can contain an unlimited number of containers.
- Blobs are addressable using the following URL format: `http://<storage account>.blob.core.windows.net/<container>/<blob>`.

Figure 7-5 illustrates the concepts and relationships of the different Azure Blob components.

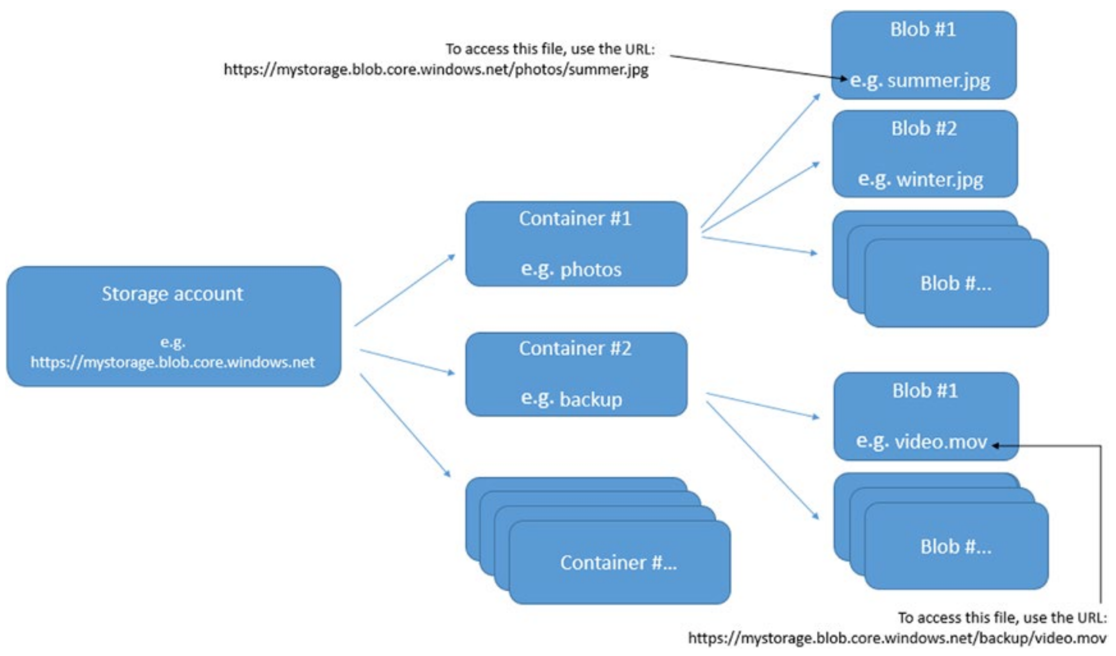

To access this file, use the URL:
https://mystorage.blob.core.windows.net/photos/summer.jpg
Blob #1
e.g. summer.jpg
Blob #2
e.g. winter.jpg
Container #1
e.g. photos
Blob #...
Storage account
e.g.
https://mystorage.blob.core.windows.net
Container #2
e.g. backup
Blob #1
e.g. video.mov
Container #...
Blob #...
To access this file, use the URL:
https://mystorage.blob.core.windows.net/backup/video.mov

***Figure 7-5.** Azure Blob components and access via HTTPS URL*

In addition to accessing Azure Blob storage through the browser, Microsoft also makes available a free Azure Storage Explorer through CodePlex. Azure Storage Explorer is a graphical user interface (GUI) application that provides user-friendly access to Azure Blob storage. You can download the Azure Storage Explorer at `http://azurestorageexplorer.codeplex.com`.

■ **Note** Azure Storage performance metrics, such as IOPS and disk bandwidth, for the various VM sizes are available on the Azure Documentation site: `https://azure.microsoft.com/en-us/documentation/articles/storage-premium-storage-preview-Portal`.

Azure Storage Analytics

It is sometimes helpful to analyze how storage is being used. This is especially useful for optimizing the use of storage and finding opportunities to lower the cost of consumption or for troubleshooting purposes.

Azure Storage Analytics logs the access and actions of users accessing Azure Storage: for example, users storing content in Azure. Azure Storage Analytics then analyzes the logs to provide metrics data for a storage account.

To use Azure Storage Analytics, you must enable it individually for each service. You do so from the Azure Management Portal; for details, see Chapter 13. You can also enable Storage Analytics programmatically via the REST API or the client library.

The aggregated data is stored in a well-known blob (for logging) and in well-known tables (for metrics), which may be accessed using the Azure Table service APIs or through the Azure Management Portal. Storage Analytics has a 20 TB limit on the amount of stored data that is independent of the total limit for your storage account.

Storage Analytics logs detail information about successful and failed requests. This information can be used to monitor individual requests and to diagnose issues with a storage service. Requests are logged on a best-effort basis. Log entries are created only if there is storage service activity. For example, if a storage account has activity in its Blob service but not in its Table or Queue services, only logs pertaining to the Blob service are created.

Azure Storage Analytics is not enabled by default and can only be turned on by the storage account owner. Each write operation performed by Storage Analytics is a billable activity in addition to the amount of storage used to store metrics data.

All logs are stored in block blobs in a container named `$logs`, which is automatically created when Storage Analytics is enabled for a storage account. The `$logs` container is located in the blob namespace of the storage account, such as `http://<accountname>.blob.core.windows.net/$logs`. This container cannot be deleted once Storage Analytics has been enabled, although its contents can be deleted.

■ **Note** At the time of this writing, capacity metrics are only available for the Blob service. Capacity metrics for the Table service and the Queue service will be available in future versions of Storage Analytics.

Azure Import/Export Service

The *Azure Import/Export service* allows you to transfer data to and from Azure by shipping hard disk drives. In scenarios with large amounts of data, shipping it via the Import/Export service may be faster and more cost effective than electronically transferring the data over the network or Internet. The hard disk drives that are shipped to or from Microsoft datacenters are encrypted.

Azure SQL Database

Azure SQL Database is a relational database-as-a-service type of storage for structured data.

■ **Note** Azure SQL Database is based on Microsoft SQL Server database technology. This should not be confused with the similar Azure Table storage offering, which is a NoSQL datastore that allows authenticated requests for structured data. Because the concepts and capabilities in Azure SQL Database and Azure Table are similar, this chapter only covers Azure SQL Database. For information specifically about Azure Table storage, visit `https://azure.microsoft.com/en-us/documentation/articles/storage-dotnet-how-to-use-tables`.

The benefit of using the Azure SQL Database service is the ability to spin up and consume a relational database quickly; it can then easily be replicated for georedundancy without all the infrastructure deployment and management normally required. When you are considering whether to use Azure SQL Database for a solution, it is important to understand the SQL Server features that are *not* available. However, Azure SQL Database should be able to meet the needs of most solutions that require a relational database in the back end.

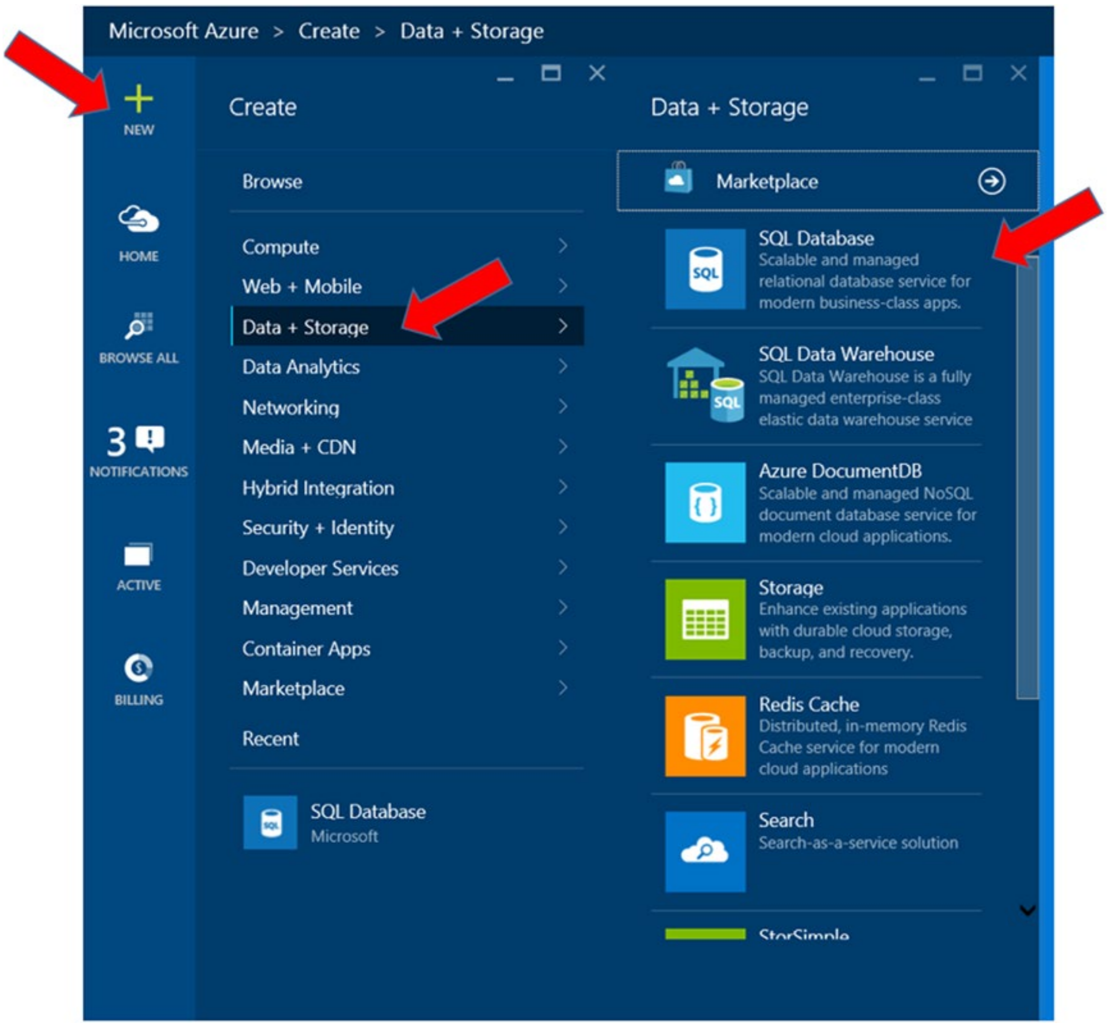
CREATING AND USING AN AZURE SQL DATABASE

The following exercise walks you through the steps required to create and access an Azure SQL database. You also use the new Azure Portal that is currently in preview:

■ **Note** You see in the Portal that you can create other types of databases, such as MySQL. Although this exercise is based on a SQL database, the concepts and steps are similar.

---

1. Go to the new Azure Management Portal at `https://portal.azure.com`.
2. Click New, select Data + Storage, and then select SQL Database, as shown in Figure 7-6.



SQL Database: Scalable and managed relational database service for modern business-class apps.

SQL Data Warehouse: SQL Data Warehouse is a fully managed enterprise-class elastic data warehouse service

Azure DocumentDB: Scalable and managed NoSQL document database service for modern cloud applications.

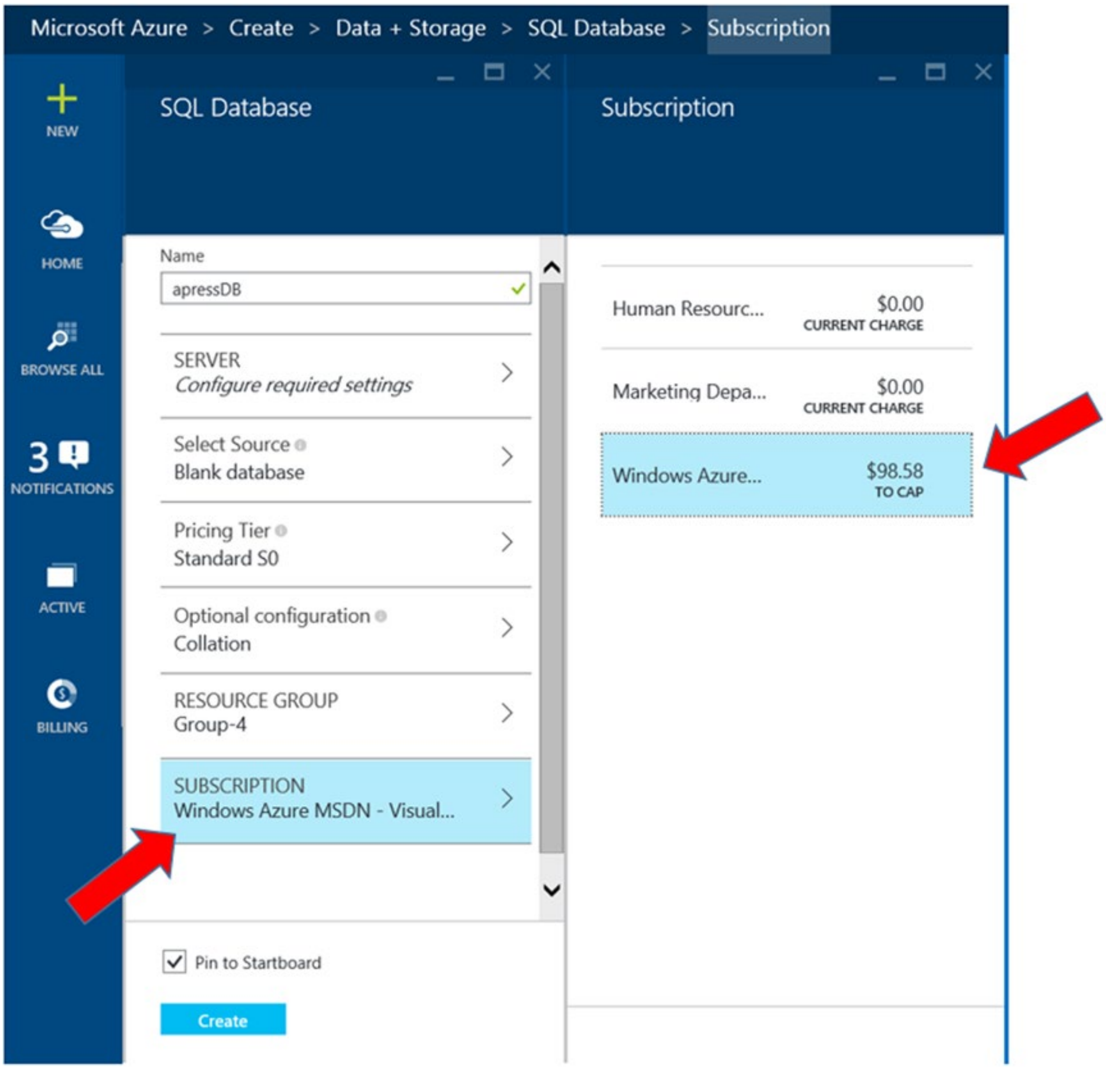
Storage: Enhance existing applications with durable cloud storage, backup, and recovery.

Redis Cache: Distributed, in-memory Redis Cache service for modern cloud applications

Search: Search-as-a-service solution

Figure 7-6. *Create a new Azure SQL database from the Portal*

3. Provide a friendly name for the SQL database. If your organization has multiple subscriptions, it is a good practice to change the subscription with which this SQL database is associated before making any other changes. At the bottom of the list, click Subscription to select the correct subscription for this database, as shown in Figure 7-7.



Name: apressDB

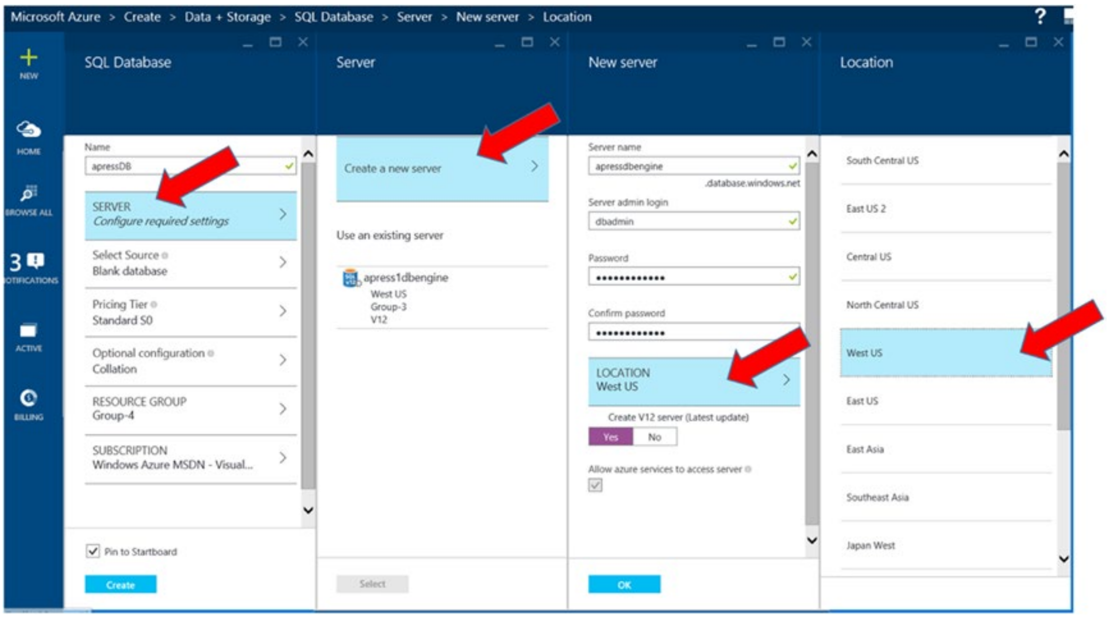
SERVER	Configure required settings
Select Source	Blank database
Pricing Tier	Standard S0
Optional configuration	Collation
RESOURCE GROUP	Group-4
SUBSCRIPTION	Windows Azure MSDN - Visual...

Human Resourc...	$0.00 CURRENT CHARGE
Marketing Depa...	$0.00 CURRENT CHARGE
Windows Azure...	$98.58 TO CAP

Figure 7-7. *Selecting the correct subscription*

4. Select the Pricing Tier menu, and choose the database type you wish to deploy. For this exercise, we selected Basic (5 DTUs), which has a maximum size of 2 GB.

5. Select Server, and then click Create A New Server, as shown in Figure 7-8. Provide information for the server name, server admin login password, and datacenter region where the database should be created. Click OK when you have finished configuring the database server, and then click Create to have Azure create the database.



SQL Database

Name: apressDB

SERVER
Configure required settings

Select Source
Blank database

Pricing Tier
Standard S0

Optional configuration
Collation

RESOURCE GROUP
Group-4

SUBSCRIPTION
Windows Azure MSDN - Visual...

☑ Pin to Startboard

Server

Create a new server

Use an existing server

apress1dbengine
West US
Group-3
V12

New server

Server name: apressdbengine .database.windows.net

Server admin login: dbadmin

Password: ••••••••••••

Confirm password: ••••••••••••

LOCATION
West US

Create V12 server (Latest update)
[✓ Yes] No

Allow azure services to access server ☑

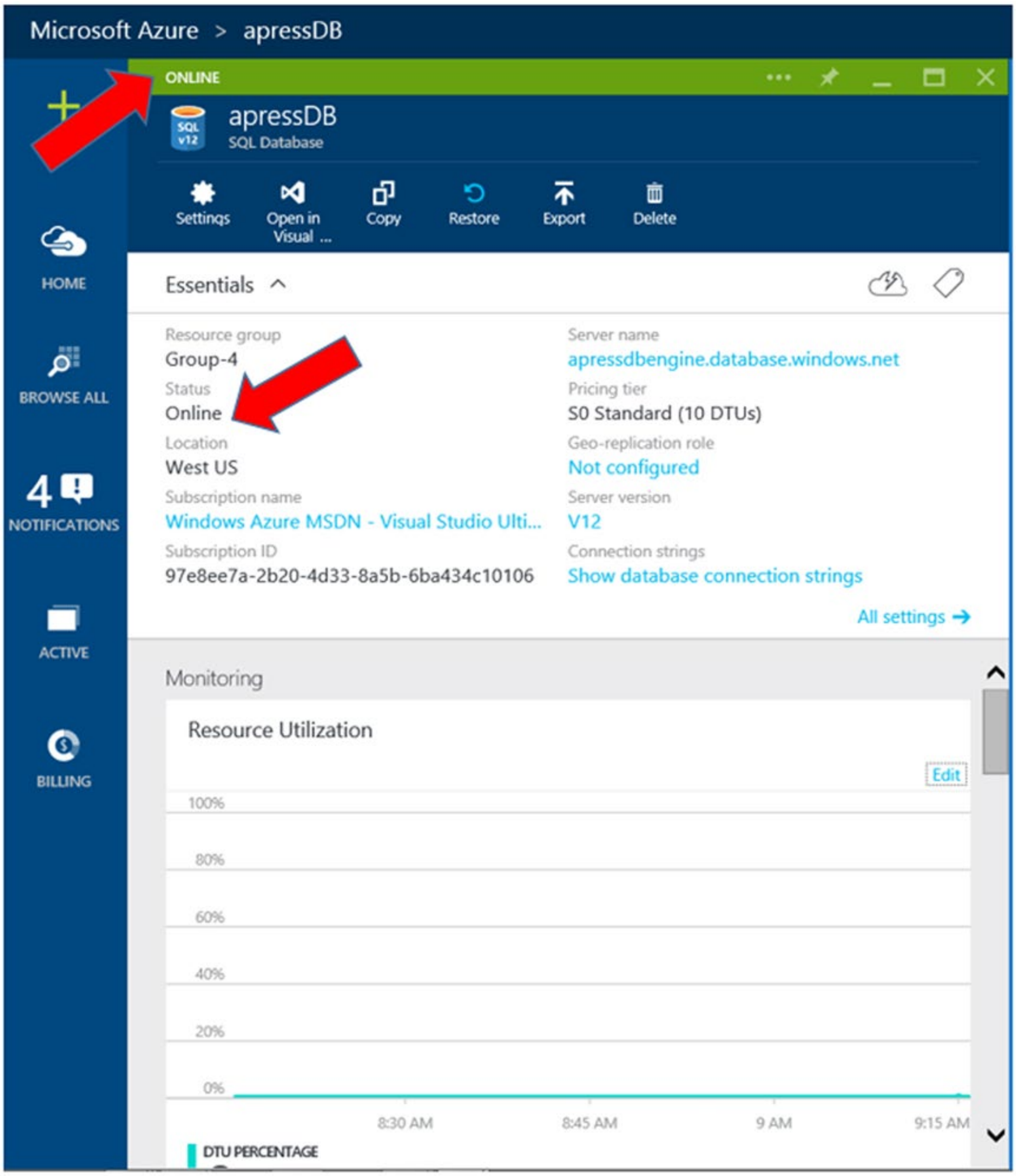
Location

South Central US
East US 2
Central US
North Central US
West US
East US
East Asia
Southeast Asia
Japan West

Figure 7-8. *Configuring the database server*

6. You are returned to the home page of the Portal. The status and progress of creating the new database are displayed on one of the tiles.

7. When the database has been created, you should automatically be directed to the SQL Database configuration screen, as shown in Figure 7-9. You can also get to the configuration screen by selecting the database from the Browse All menu option. Note that the status of the database is Online, and therefore it is ready for use.

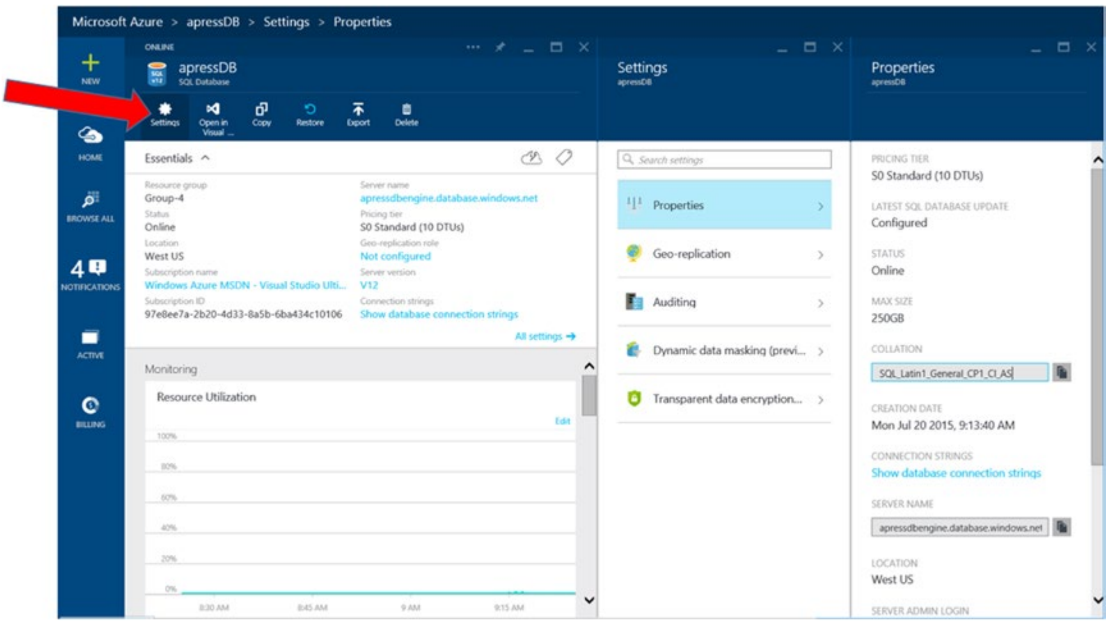


Resource group
Group-4
Status
Online
Location
West US
Subscription name
Windows Azure MSDN - Visual Studio Ulti...
Subscription ID
97e8ee7a-2b20-4d33-8a5b-6ba434c10106

Server name
apressdbengine.database.windows.net
Pricing tier
S0 Standard (10 DTUs)
Geo-replication role
Not configured
Server version
V12
Connection strings
Show database connection strings

Figure 7-9. *Viewing the database properties*

8. As shown in Figure 7-10, click Settings to review and configure additional settings for the database, such as auditing and georeplication. This exercise does not go through all of these settings, because they are self-explanatory.



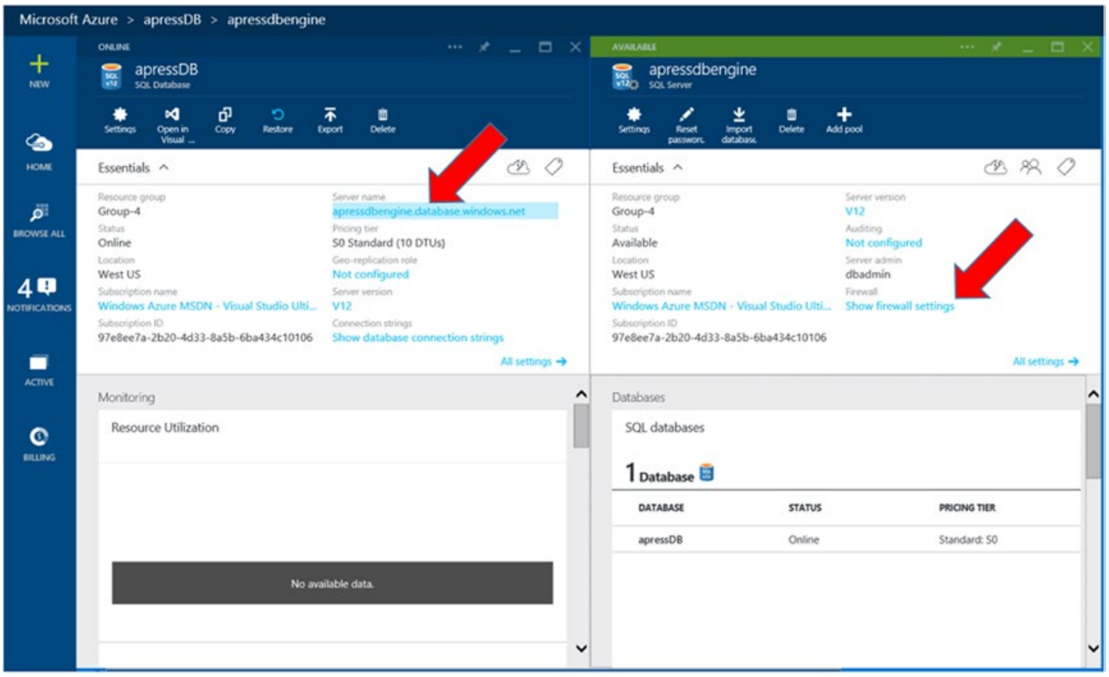
Resource group
Group-4
Status
Online
Location
West US
Subscription name
Windows Azure MSDN - Visual Studio Ulti...
Subscription ID
97e8ee7a-2b20-4d33-8a5b-6ba434c10106

Server name
apressdbengine.database.windows.net
Pricing tier
S0 Standard (10 DTUs)
Geo-replication role
Not configured
Server version
V12
Connection strings
Show database connection strings

PRICING TIER
S0 Standard (10 DTUs)
LATEST SQL DATABASE UPDATE
Configured
STATUS
Online
MAX SIZE
250GB
COLLATION
SQL_Latin1_General_CP1_CI_AS
CREATION DATE
Mon Jul 20 2015, 9:13:40 AM
CONNECTION STRINGS
Show database connection strings
SERVER NAME
apressdbengine.database.windows.net
LOCATION
West US
SERVER ADMIN LOGIN

***Figure 7-10.** Configuring additional database settings*

9. By default, the firewall settings for this database server would not allow you to connect to it directly from your client. In order to configure firewall settings so that you can connect to the database, click the server name and then select Show Firewall Settings, as shown in Figure 7-11.



Resource group	Server name
Group-4	apressdbengine.database.windows.net
Status	Pricing tier
Online	S0 Standard (10 DTUs)
Location	Geo-replication role
West US	Not configured
Subscription name	Server version
Windows Azure MSDN - Visual Studio Ulti...	V12
Subscription ID	Connection strings
97e8ee7a-2b20-4d33-8a5b-6ba434c10106	Show database connection strings

Monitoring

Resource Utilization

No available data.

Resource group	Server version
Group-4	V12
Status	Auditing
Available	Not configured
Location	Server admin
West US	dbadmin
Subscription name	Firewall
Windows Azure MSDN - Visual Studio Ulti...	Show firewall settings
Subscription ID	
97e8ee7a-2b20-4d33-8a5b-6ba434c10106	

Databases

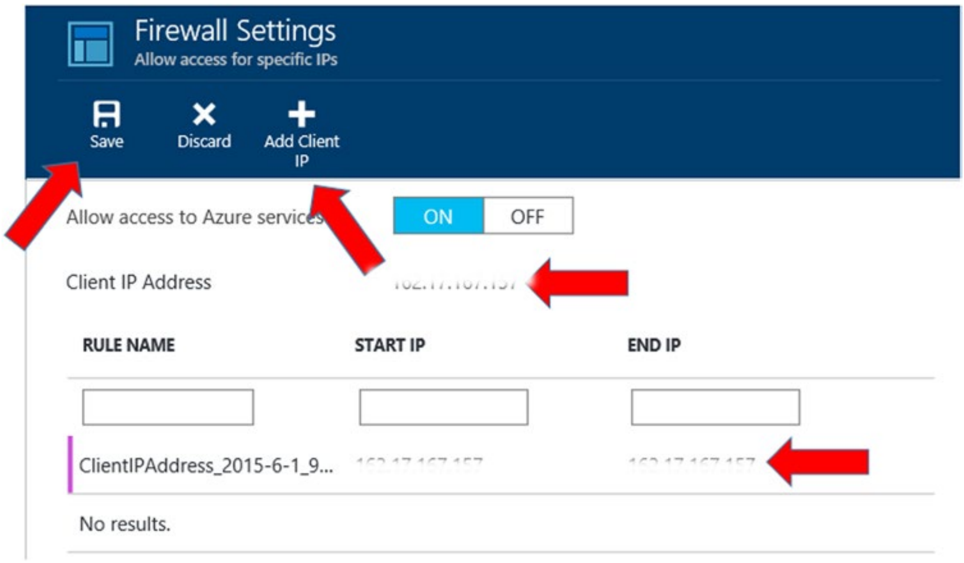
SQL databases

1 Database

DATABASE	STATUS	PRICING TIER
apressDB	Online	Standard: S0

Figure 7-11. *Configuring the database server firewall settings*

10. The client IP address of the machine you are using to access this configuration should be detected. You have the option to add this IP to allow access by clicking Add Client IP; you can also specify ranges of IP addresses. Click Add Client IP, as shown in Figure 7-12. The IP address should be added to the access list.



Allow access to Azure services [✓ ON] OFF

Client IP Address 162.17.167.157

RULE NAME	START IP	END IP
ClientIPAddress_2015-6-1_9...	162.17.167.157	162.17.167.157

No results.

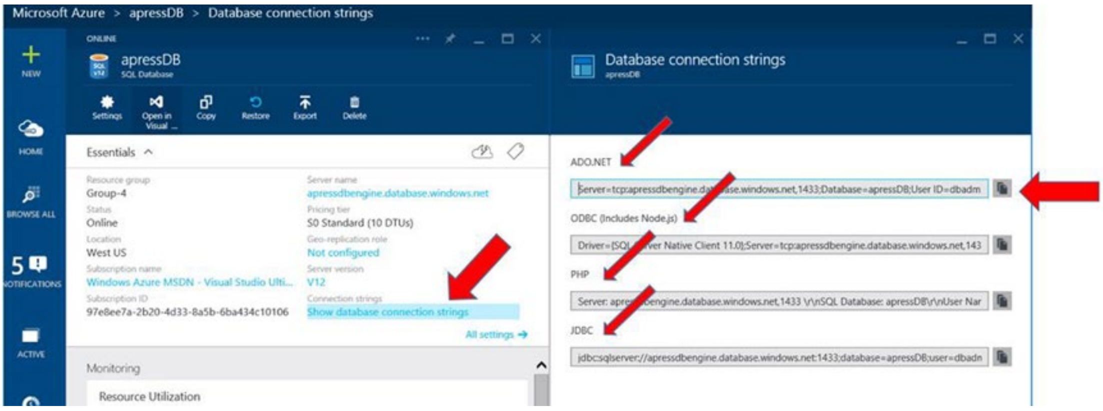
Figure 7-12. *Adding the client IP address to the access list*

11. Click Save to save the settings. The Portal says it may take up to five minutes for the settings to take effect, but it should happen immediately. If you have added the IP address to the access list and saved the settings but are still unable to connect to the database directly, wait a few minutes and then try again.

■ **Note** It is important to save your firewall settings. It is not uncommon for administrators to add IP addresses to the access list and see them appear on the access list, as shown in Figure 7-9, but not save the settings. The firewall settings are not effective until you click Save.

You can now connect to the database to create tables and add data. Like any database, you can access it via a database-management tool like SQL Server Management or Visual Studio, or programmatically via database connection strings. The next part of this exercise show you the available connection strings and then demonstrates how to use Visual Studio to connect to the database:

12. From the SQL Database menu, click Show Database Connection Strings, as shown in Figure 7-13. Notice that there are four connections strings from which to choose, depending on what type of application you are developing. There is also a handy clipboard option that allows you to copy the connection string.

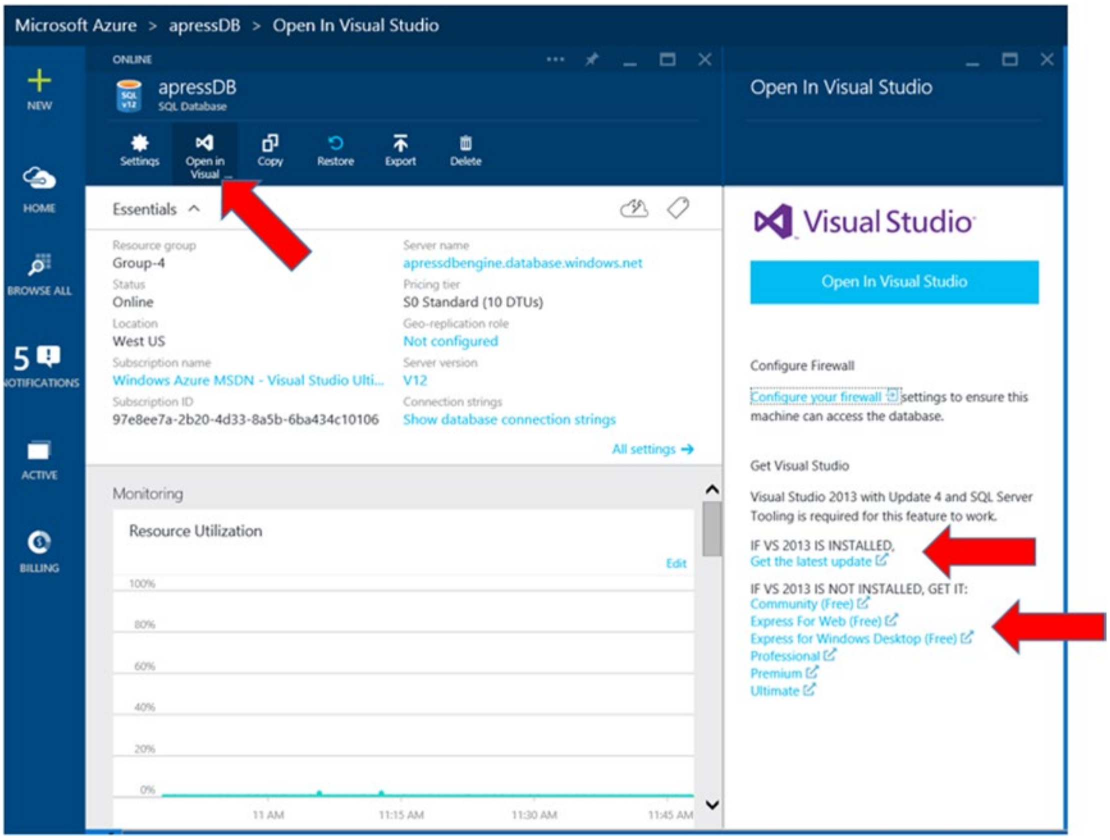


Resource group
Group-4
Status
Online
Location
West US
Subscription name
Windows Azure MSDN - Visual Studio Ulti...
Subscription ID
97e8ee7a-2b20-4d33-8a5b-6ba434c10106
Server name
apressdbengine.database.windows.net
Pricing tier
S0 Standard (10 DTUs)
Geo-replication role
Not configured
Server version
V12
Connection strings
Show database connection strings

ADO.NET
```
Server=tcp:apressdbengine.database.windows.net,1433;Database=apressDB;User ID=dbadm
```
ODBC (Includes Node.js)
```
Driver={SQL Server Native Client 11.0};Server=tcp:apressdbengine.database.windows.net,143
```
PHP
```
Server: apressdbengine.database.windows.net,1433 \r\nSQL Database: apressDB\r\nUser Nar
```
JDBC
```
jdbc:sqlserver://apressdbengine.database.windows.net:1433;database=apressDB;user=dbadn
```

Figure 7-13. *Azure SQL Database connection strings*

13. Click Open In Visual Studio, as shown in Figure 7-14. Then select the option that is most applicable to your scenario. If you already have Visual Studio installed, you may need to get the latest update so that Visual Studio can connect to an Azure SQL database. Otherwise, you will get an error message that says Visual Studio does not recognize the database type. If you do not have Visual Studio, select one of the other options in order to connect to an Azure SQL database.



Resource group
Group-4
Status
Online
Location
West US
Subscription name
Windows Azure MSDN - Visual Studio Ulti...
Subscription ID
97e8ee7a-2b20-4d33-8a5b-6ba434c10106

Server name
apressdbengine.database.windows.net
Pricing tier
S0 Standard (10 DTUs)
Geo-replication role
Not configured
Server version
V12
Connection strings
Show database connection strings

All settings →

Configure Firewall

Configure your firewall settings to ensure this machine can access the database.

Get Visual Studio

Visual Studio 2013 with Update 4 and SQL Server Tooling is required for this feature to work.

IF VS 2013 IS INSTALLED,
Get the latest update

IF VS 2013 IS NOT INSTALLED, GET IT:
Community (Free)
Express For Web (Free)
Express for Windows Desktop (Free)
Professional
Premium
Ultimate

Figure 7-14. *Download Visual Studio updates.*

For this exercise, we downloaded and installed the updates for Visual Studio 2013.

14. Launch Visual Studio.

15. Click Tools from the menu, and select Connect To Database, as shown in Figure 7-15.

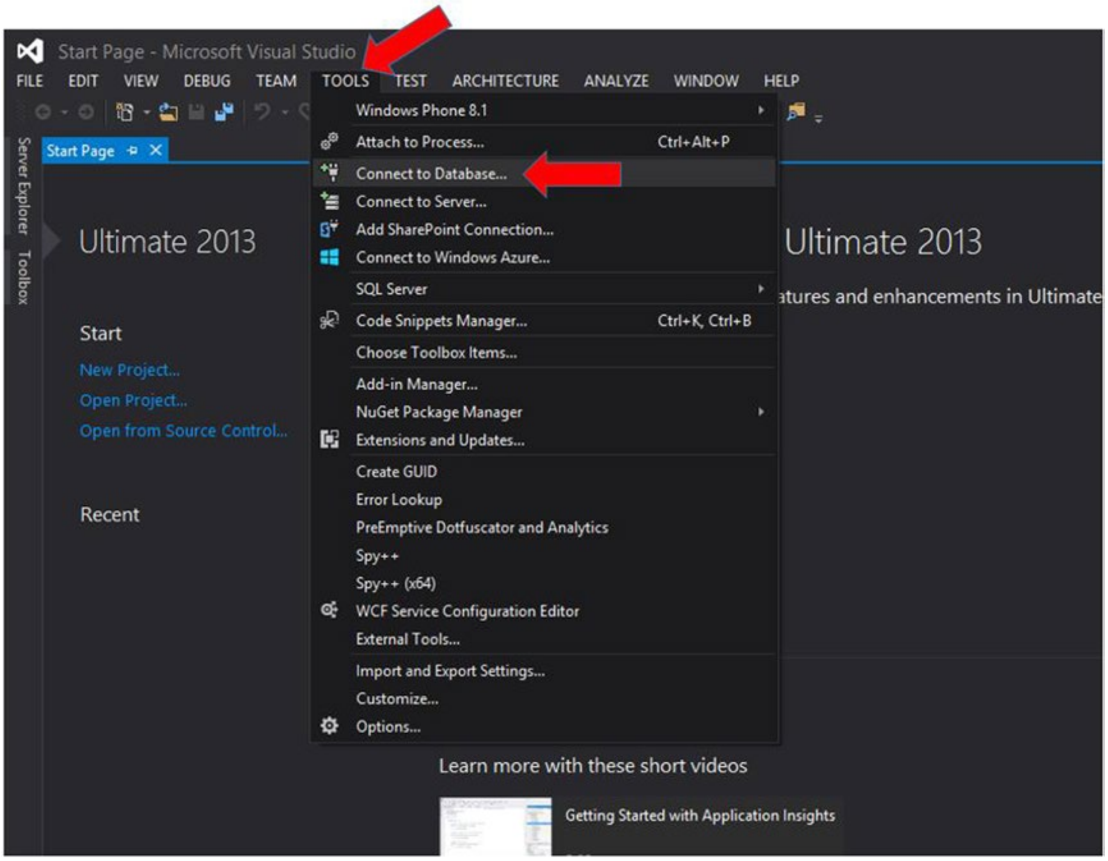
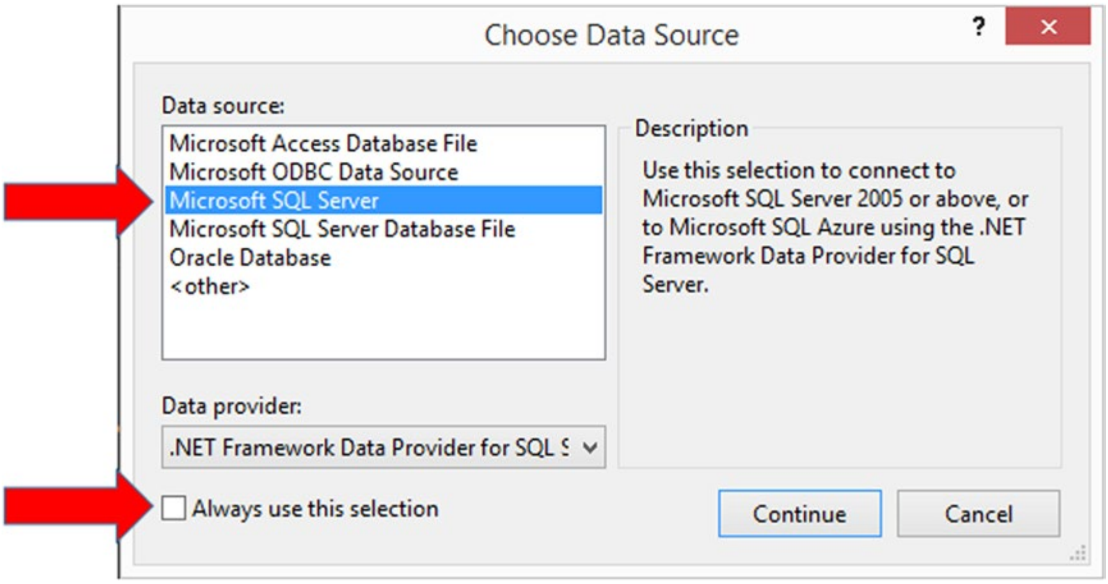

Start Page - Microsoft Visual Studio
FILE EDIT VIEW DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Windows Phone 8.1
Attach to Process... Ctrl+Alt+P
Connect to Database...
Connect to Server...
Add SharePoint Connection...
Connect to Windows Azure...
SQL Server
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
Add-in Manager...
NuGet Package Manager
Extensions and Updates...
Create GUID
Error Lookup
PreEmptive Dotfuscator and Analytics
Spy++
Spy++ (x64)
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...
Start Page
Server Explorer
Toolbox
Ultimate 2013
Start
New Project...
Open Project...
Open from Source Control...
Recent
Ultimate 2013
atures and enhancements in Ultimate
Learn more with these short videos
Getting Started with Application Insights

Figure 7-15. *Connecting to a database from Visual Studio*

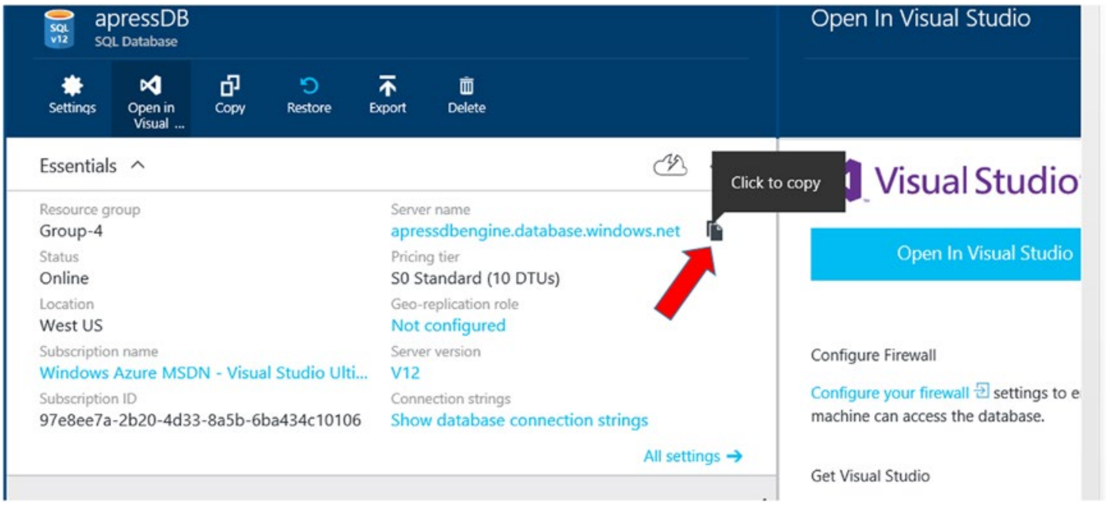
16. Select Microsoft SQL Server as the data source, and uncheck Always Use This Selection, as shown in Figure 7-16. Click Continue.


Choose Data Source
Data source:
Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Database File
Oracle Database
<other>
Description
Data provider:
.NET Framework Data Provider for SQL S
Always use this selection
Continue
Cancel

Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.

Figure 7-16. *Selecting a data source in Visual Studio*

17. From the Azure Management Portal, copy the server name, as shown in Figure 7-17.



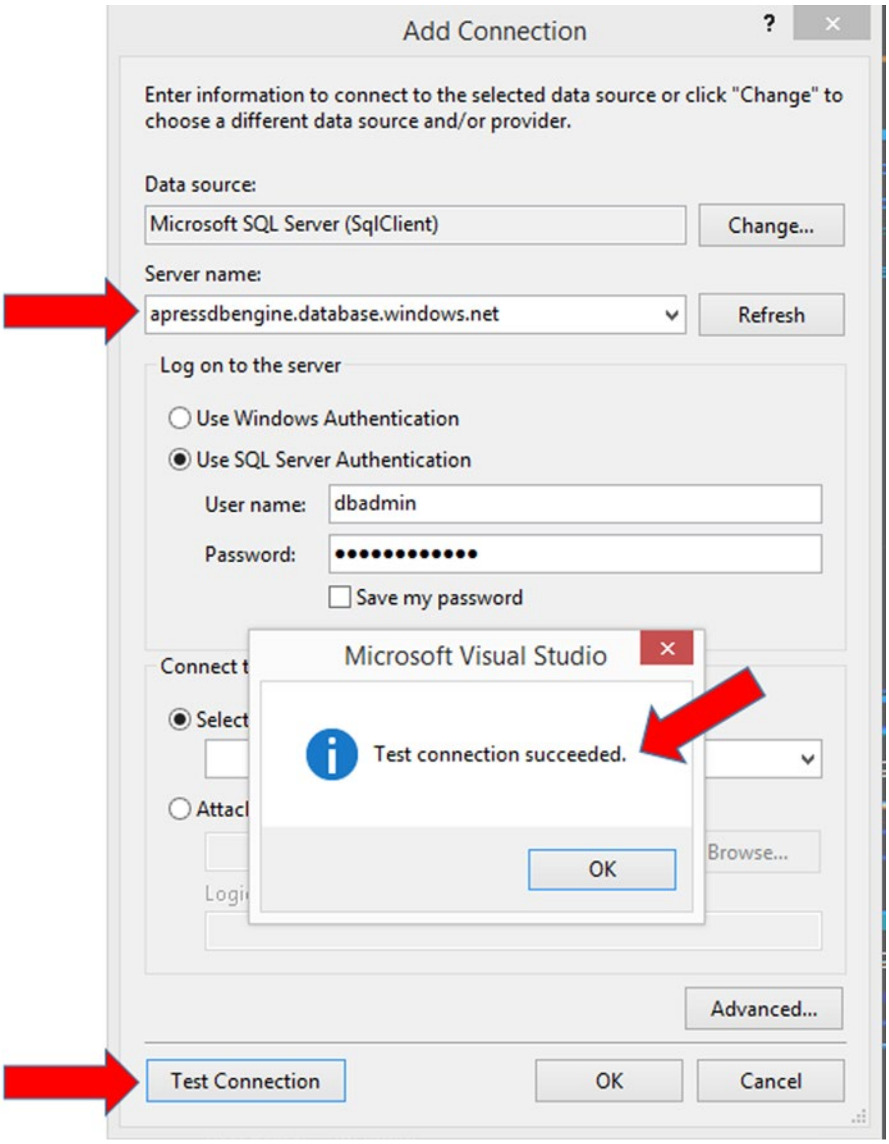
Resource group
Group-4
Status
Online
Location
West US
Subscription name
Windows Azure MSDN - Visual Studio Ulti...
Subscription ID
97e8ee7a-2b20-4d33-8a5b-6ba434c10106

Server name
apressdbengine.database.windows.net
Pricing tier
S0 Standard (10 DTUs)
Geo-replication role
Not configured
Server version
V12
Connection strings
Show database connection strings

Configure your firewall settings to e
machine can access the database.

Figure 7-17. *Copying the Azure SQL database server name*

18. Paste the server name into the Visual Studio Add Connection dialog box, select Use SQL Server Authentication, and provide the required credentials, as shown in Figure 7-18.


Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: apressdbengine.database.windows.net

Log on to the server

◯ Use Windows Authentication

● Use SQL Server Authentication

User name: dbadmin

Password: ●●●●●●●●●●●●

☐ Save my password

Microsoft Visual Studio

Test connection succeeded.

Figure 7-18. *Testing the database connection*

19. Click Test Connection to confirm that a successful connection to the database can be made, as also shown in Figure 7-18.
20. Click OK to close the dialog box indicating that the test connection succeeded.
21. Click OK again to close the Add Connection window.
22. A new database connection is now established. If you expand Data Connections, you should see the Azure SQL database server, as shown in Figure 7-19. Expand it, right-click Tables, and select Add New Table.

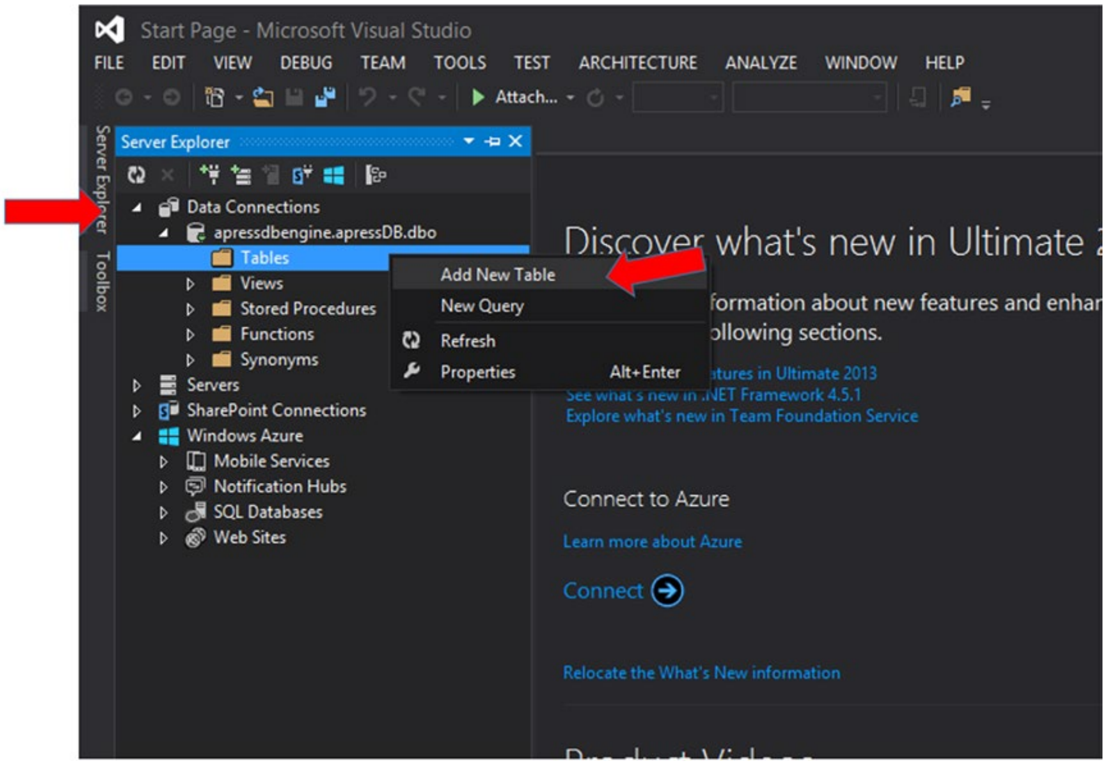


Figure 7-19. *Creating a new table in an Azure SQL database*

This concludes the exercise—managing databases and development are beyond the scope of this chapter. In this exercise, you saw how an Azure SQL database is created and how to access it via Visual Studio.

StorSimple

StorSimple is a hybrid on-premises and cloud storage solution. It consists of a 2U rackmount appliance that has SSDs, hard drives, and the ability to use Azure as a storage source. Figure 7-20 is a schematic of the rear panel of a StorSimple 8100 storage appliance.

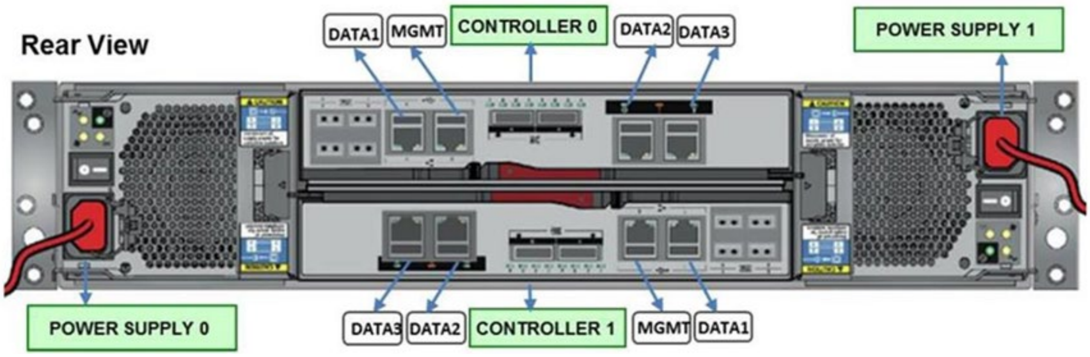


Figure 7-20. *Rear schematic of a StorSimple 8100 series storage appliance*

The idea behind StorSimple is the need for high-speed, low-latency access to frequently used data—hence the local storage on the appliance. Then, as data ages out, the StorSimple device moves the data from higher-speed local SSDs to the traditional hard drives that are also still on the appliance. Finally, once the device has reached 95% of capacity, as the data ages out even more, the content is moved to Azure storage. If the file is accessed after it has been moved to Azure storage, it is “promoted” back into the on-premises local StorSimple SSDs, and the aging process starts over. This entire process is transparent to the administrator and end user. It is depicted in Figure 7-21.

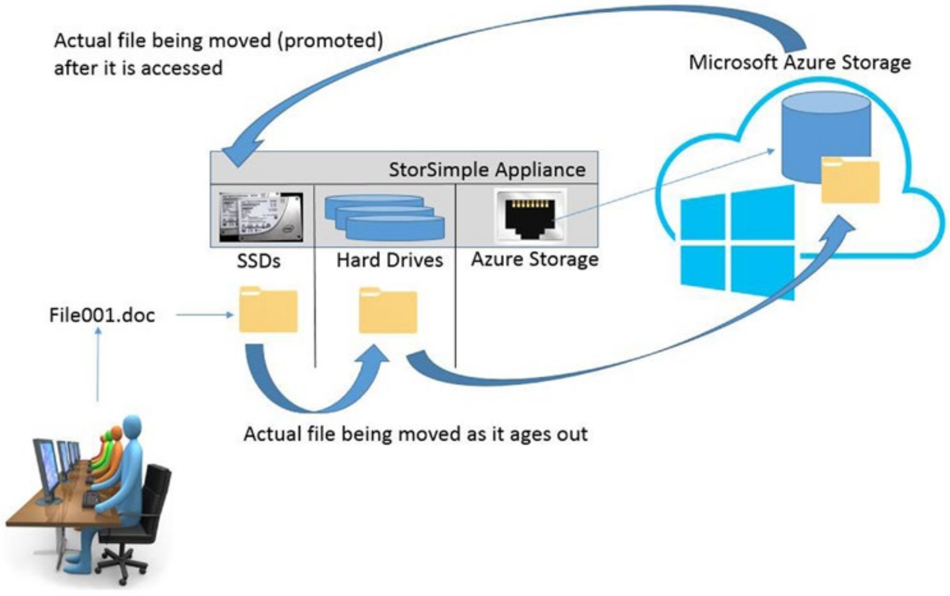
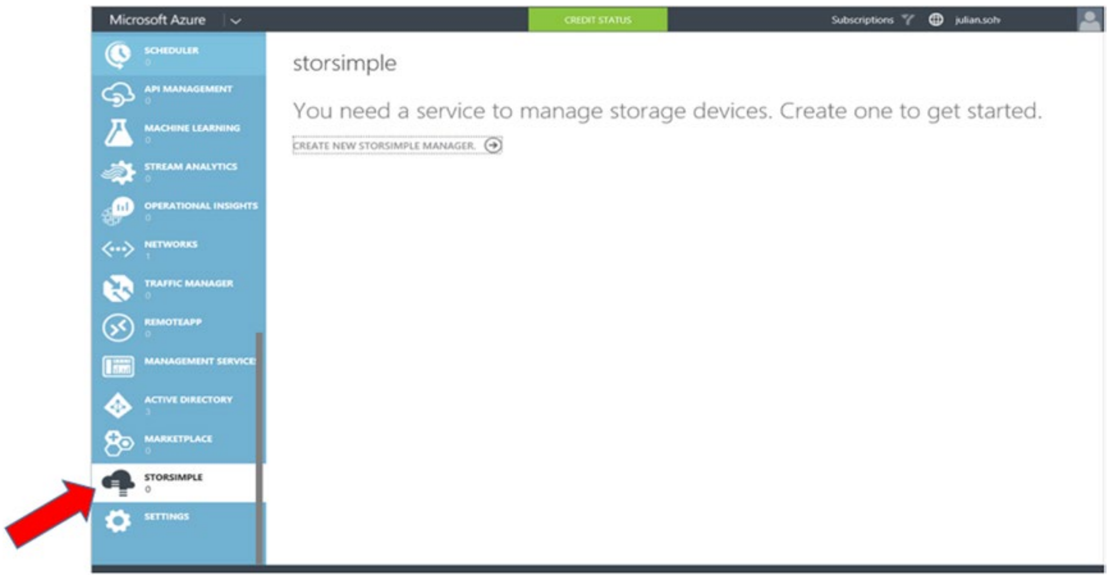

Actual file being moved (promoted) after it is accessed
Microsoft Azure Storage
StorSimple Appliance
SSDs
Hard Drives
Azure Storage
File001.doc
Actual file being moved as it ages out

Figure 7-21. *How StorSimple’s aging process for files works*

A StorSimple-specific storage account known as the *StorSimple Manager* is used to configure StorSimple. The StorSimple Manager is an extension of the Azure Management Portal. At the time of this writing, the new Preview Portal does not yet support the creation of a StorSimple Manager, so you need to use the Azure Portal. When you purchase StorSimple, this option becomes available, as shown in Figure 7-22. Also, at the time of this writing, Microsoft provides an onboarding service for StorSimple customers. As part of the service, Microsoft helps customers configure the StorSimple Manager and the appliance.



storsimple

You need a service to manage storage devices. Create one to get started.

Figure 7-22. *Enabling the StorSimple Manager*

■ **Note** At the time of this writing, the current hardware appliance versions for StorSimple are the 8000 series. The 7000 series of appliances is deprecated and superseded by the 8000 series, although the former can still be used with Azure. In some cases, such as Azure Government (MAG), only the 8000 series appliances are supported. StorSimple 5000 series hardware is no longer supported. For more information about StorSimple, visit `http://azure.microsoft.com/en-us/documentation/services/storsimple`.

StorSimple 8000 appliances provide automatic compression and de-duplication, so there is nothing for an administrator to configure with respect to these capabilities. The appliances have 10 GB Ethernet connections and come in two versions:

- *StorSimple 8100*: Provides 15 TB to 40 TB of storage on the appliance, depending on the level of compression achieved. The SSD on the 8100 is 800 GB in size. The maximum available storage that an 8100 appliance can handle, including Azure storage, is 200 TB.
- *StorSimple 8600*: Provides 40 TB to 100 TB of storage on the appliance, depending on the level of compression achieved. The SSD on the 8600 is 2 TB in size. The maximum available storage that an 8600 appliance can handle, including Azure storage, is 500 TB.

Summary

This chapter introduced the different types of Azure Storage accounts. These accounts determine certain hardware and deployment characteristics, such as LRS versus GRS, and standard HDDs versus high-performance SSDs.

Building on an Azure Storage account, you can define block blob, page blob, or file storage. You can also use storage accounts to create table and queue storage.

You were also introduced to relational databases as a service in the form of Azure SQL Database. But other non-SQL relational databases as a service, like MySQL, are also available.

Finally, you visited StorSimple as a hybrid on-premises and cloud storage solution. StorSimple is a unique solution that takes advantage of the best of on-premises and cloud storage.

The next chapter covers how to extend your network into Azure.