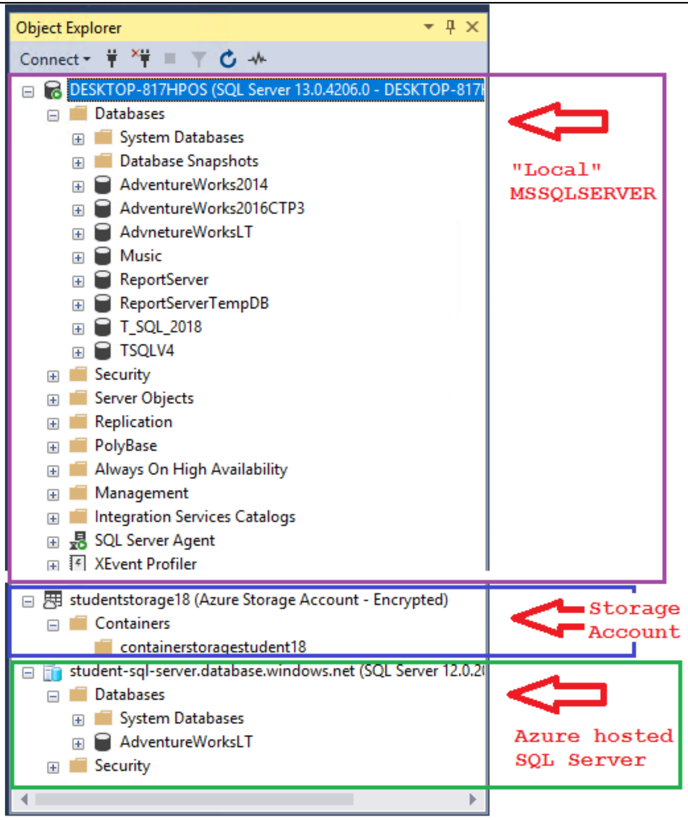
Everything is a "Container"120318

- Container Service Documentation: https://docs.microsoft.com/en-us/azure/container-service/

SSMS & The Azure Portal

Prerequisites…

1. Azure Portal:
   a) Create storage (parent to "containers")
   b) Create containers (inside "storage")
   c) Create SQL Server
2. SSMS:
   a) Connect to Azure:
      1. Database Engine
      2. Storage
   b) Backup database to storage URL.
   c) Restore database from Azure Storage to Azure database



Object Explorer

Connect ▾

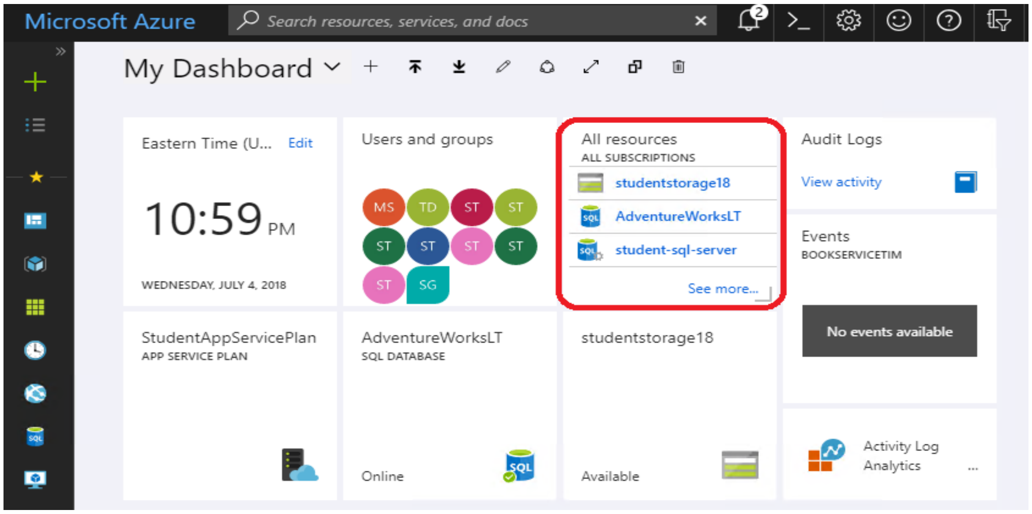
- DESKTOP-817HPOS (SQL Server 13.0.4206.0 - DESKTOP-817I
  - Databases
    - System Databases
    - Database Snapshots
    - AdventureWorks2014
    - AdventureWorks2016CTP3
    - AdvnetureWorksLT
    - Music
    - ReportServer
    - ReportServerTempDB
    - T_SQL_2018
    - TSQLV4
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent
  - XEvent Profiler
- studentstorage18 (Azure Storage Account - Encrypted)
  - Containers
    - containerstoragestudent18
- student-sql-server.database.windows.net (SQL Server 12.0.2
  - Databases
    - System Databases
    - AdventureWorksLT
  - Security

"Local" MSSQLSERVER

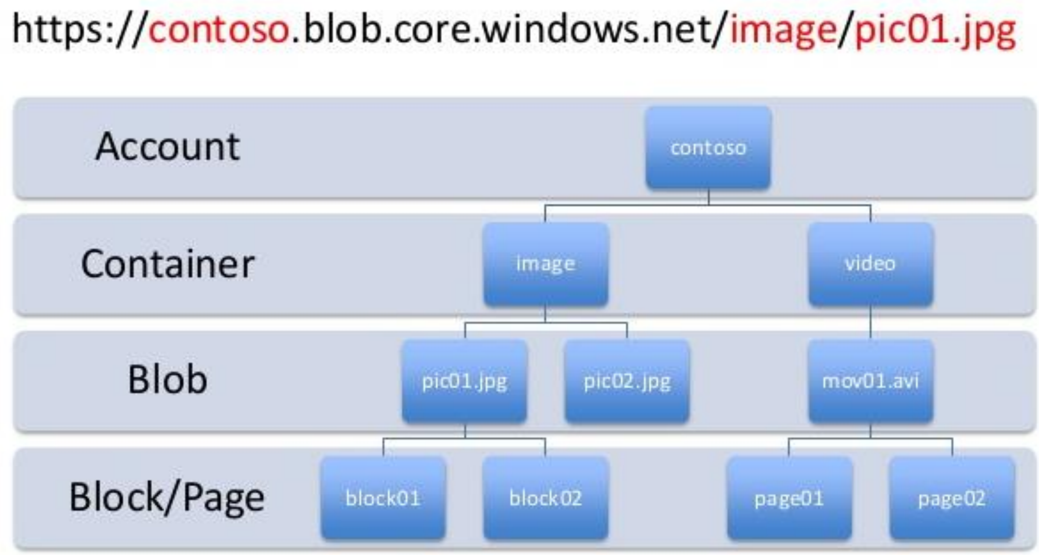
Storage Account

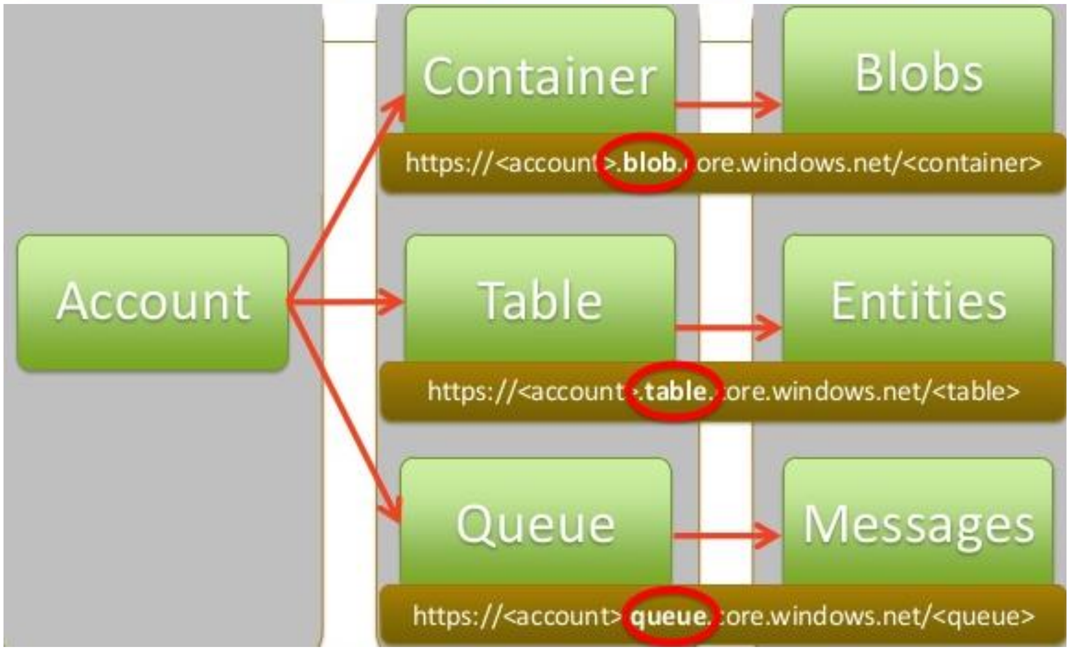
Azure hosted SQL Server

Navigating Azure Portal Dashboard – All resources - *See more*



Note URL: maps to items within storage

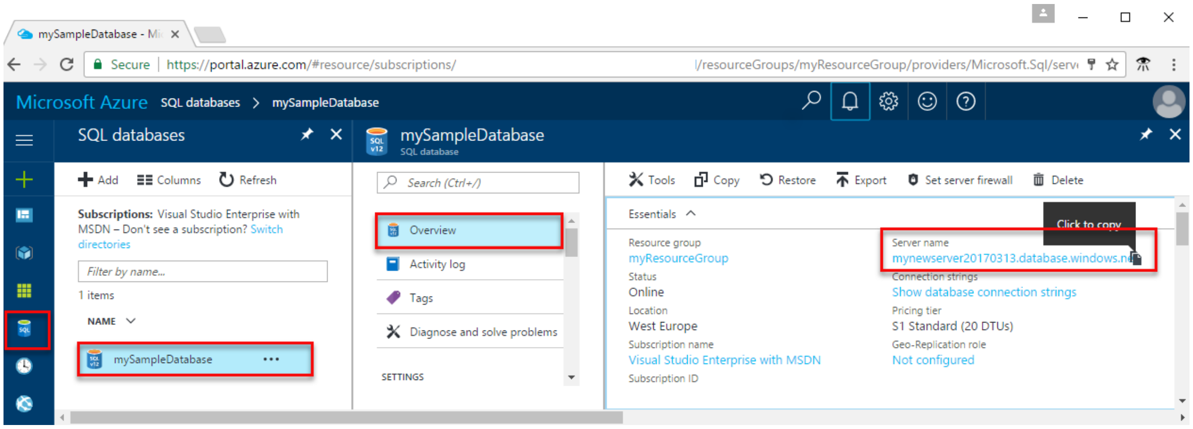



Account
Container
Blobs
https://<account>.blob.core.windows.net/<container>
Table
Entities
https://<account>.table.core.windows.net/<table>
Queue
Messages
https://<account>.queue.core.windows.net/<queue>

SQL server connection information

Get the connection information needed to connect to the Azure SQL database. You will need the fully qualified server name, database name, and login information in the next procedures.

1. Sign in to the Azure portal.
2. Select **SQL Databases** from the left-hand menu and click your database on the **SQL databases** page.
3. On the **Overview** page for your database, review the fully qualified server name as shown in the following image. You can hover over the server name to bring up the **Click to copy** option.



Subscriptions: Visual Studio Enterprise with MSDN – Don't see a subscription? Switch directories

Filter by name...

Resource group: myResourceGroup

Status: Online

Location: West Europe

Subscription name: Visual Studio Enterprise with MSDN

Subscription ID

Server name: mynewserver20170313.database.windows.n

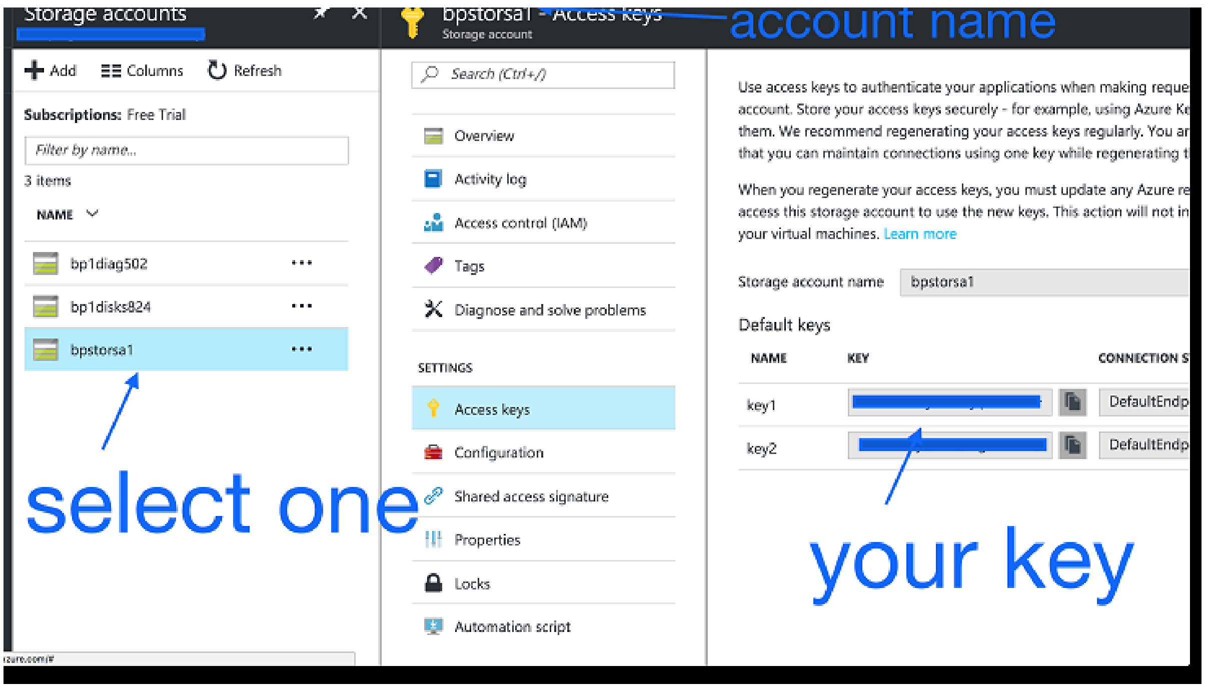
Connection strings: Show database connection strings

Pricing tier: S1 Standard (20 DTUs)

Geo-Replication role: Not configured

4. If you forget your server login information, navigate to the SQL Database server page to view the server admin name. If necessary, reset the password.

Storage Key Location



[redacted]

Subscriptions: Free Trial

Filter by name...

Use access keys to authenticate your applications when making reque account. Store your access keys securely - for example, using Azure Ke them. We recommend regenerating your access keys regularly. You ar that you can maintain connections using one key while regenerating t

When you regenerate your access keys, you must update any Azure re access this storage account to use the new keys. This action will not in your virtual machines. Learn more

Storage account name: bpstorsa1

Default keys

NAME	KEY	CONNECTION S
key1	[redacted]	DefaultEndp
key2	[redacted]	DefaultEndp

Azure Container Instances & Docker vs. Storage Account Containers

Containers are becoming the preferred way to package, deploy, and manage cloud applications.

Azure Container Instances (ACI)

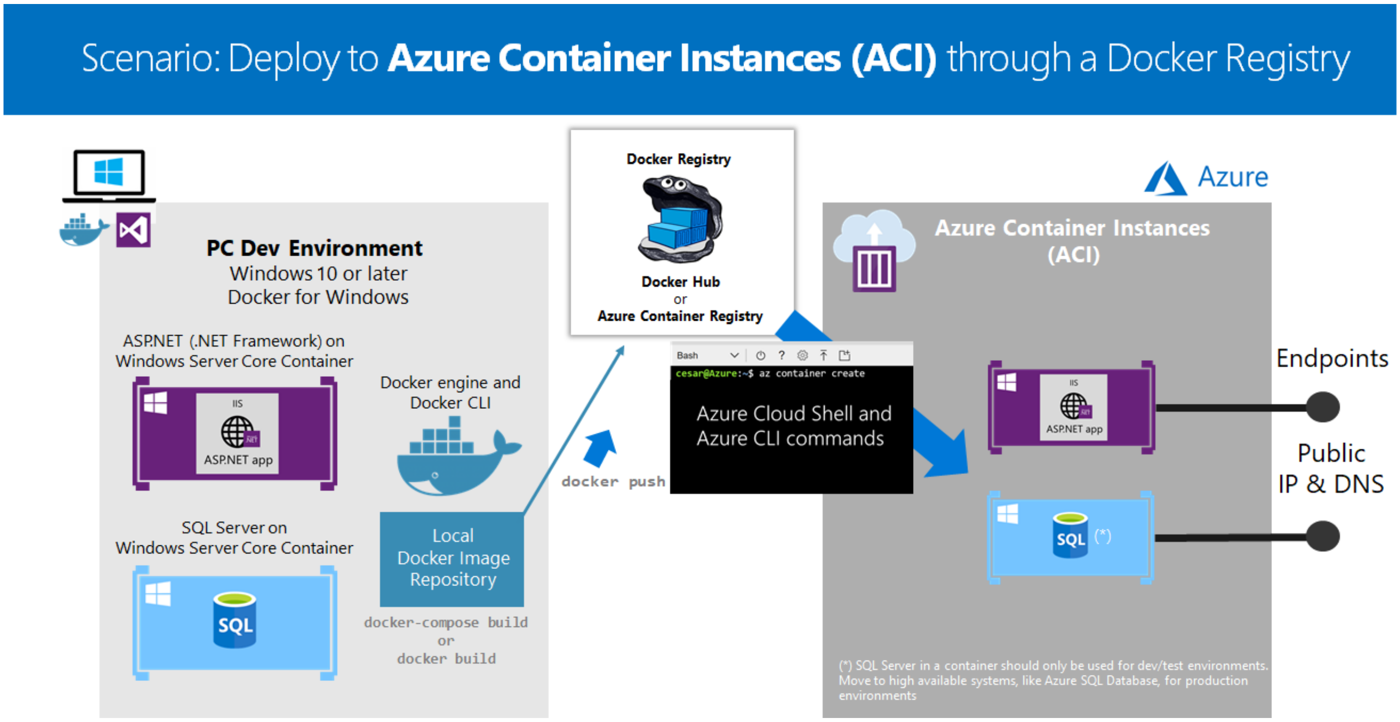
Azure Container Instances (ACI) provide a hosted environment for running containers in Azure. When using ACI, there is no need to manage the underlying compute infrastructure, Azure handles this management for you.

- When running containers in ACI, you are charged by the second for each running container.

With ACI, you can directly deploy a Windows container in Azure and expose it to the internet with a fully qualified domain name (FQDN) in a matter of seconds (Provided that you have the Windows Container image ready in a Docker registry like Docker Hub or Azure Container Registry).

When using the Virtual Kubelet provider for Azure Container Instances, both Linux and Windows containers can be scheduled on a container instance as if it is a standard Kubernetes node.

This configuration allows you to take advantage of both the capabilities of Kubernetes and the management value and cost benefit of container instances.



Scenario: Deploy to **Azure Container Instances (ACI)** through a Docker Registry

PC Dev Environment
Windows 10 or later
Docker for Windows
ASP.NET (.NET Framework) on Windows Server Core Container
IIS
ASP.NET app
SQL Server on Windows Server Core Container
SQL
Docker engine and Docker CLI
Local Docker Image Repository
docker-compose build
or
docker build
Docker Registry
Docker Hub
or
Azure Container Registry
docker push
Bash
Azure Cloud Shell and Azure CLI commands
Azure
Azure Container Instances (ACI)
IIS
ASP.NET app
SQL (*)
Endpoints
Public IP & DNS

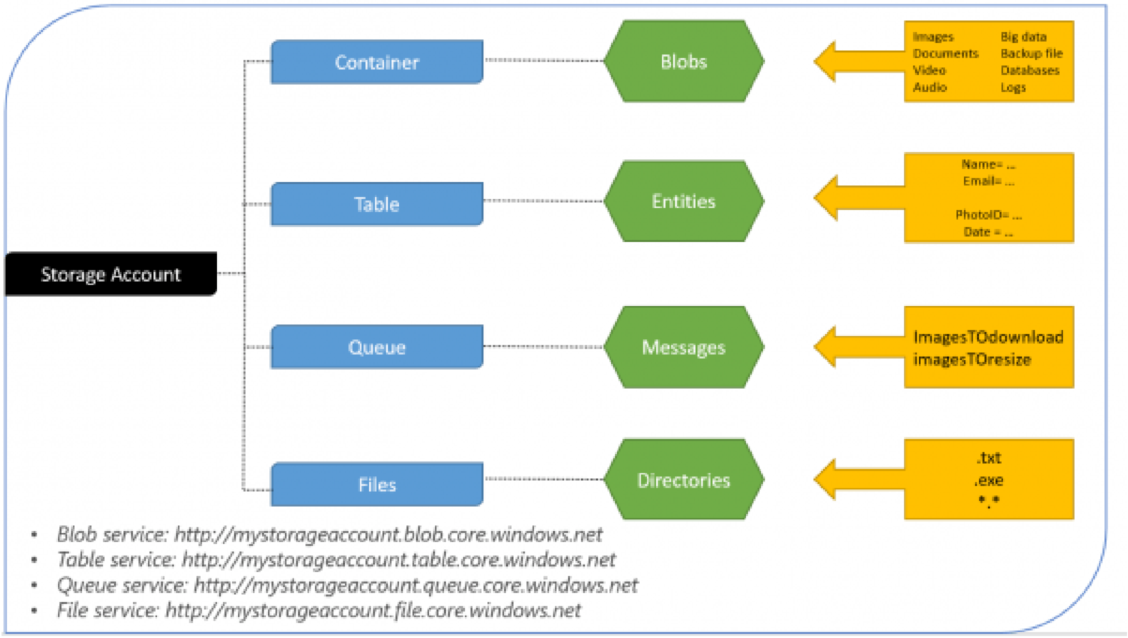
```
cesar@Azure:~$ az container create
```

(*) SQL Server in a container should only be used for dev/test environments. Move to high available systems, like Azure SQL Database, for production environments

Storage Account Container

Once you have an Azure subscription, the next thing needed is to create a storage account. This is a secured account that allows you to access different data abstraction tools, such as the Blob, Queue, Table and File services.

Every single object held within the Azure Storage system has a unique endpoint specified with a URL.



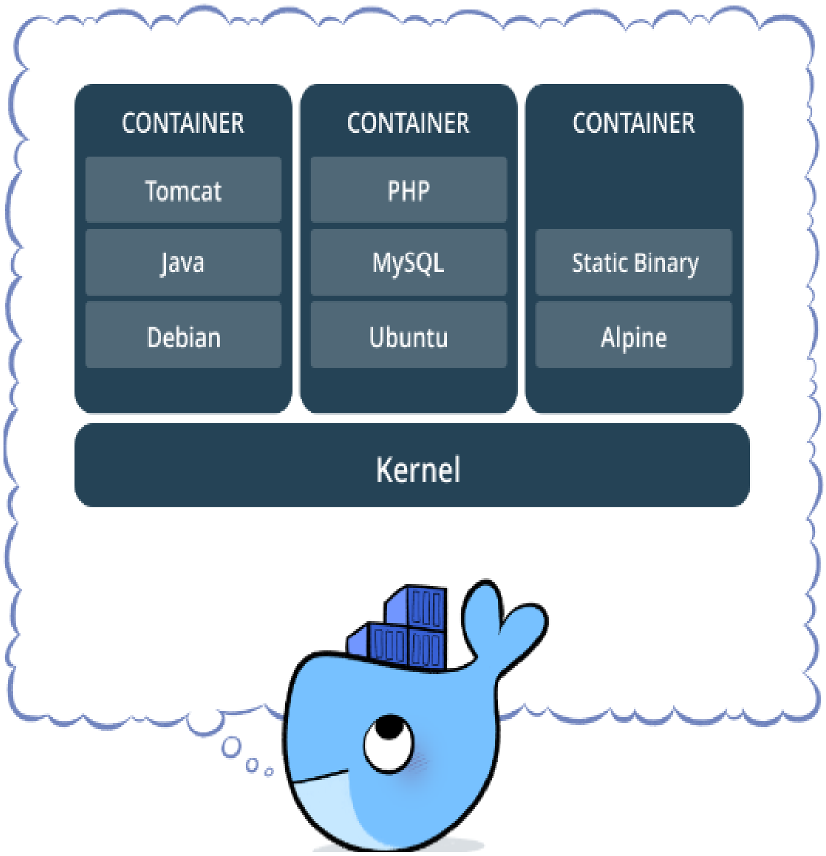
- *Blob service: http://mystorageaccount.blob.core.windows.net*
- *Table service: http://mystorageaccount.table.core.windows.net*
- *Queue service: http://mystorageaccount.queue.core.windows.net*
- *File service: http://mystorageaccount.file.core.windows.net*

Storage account architecture

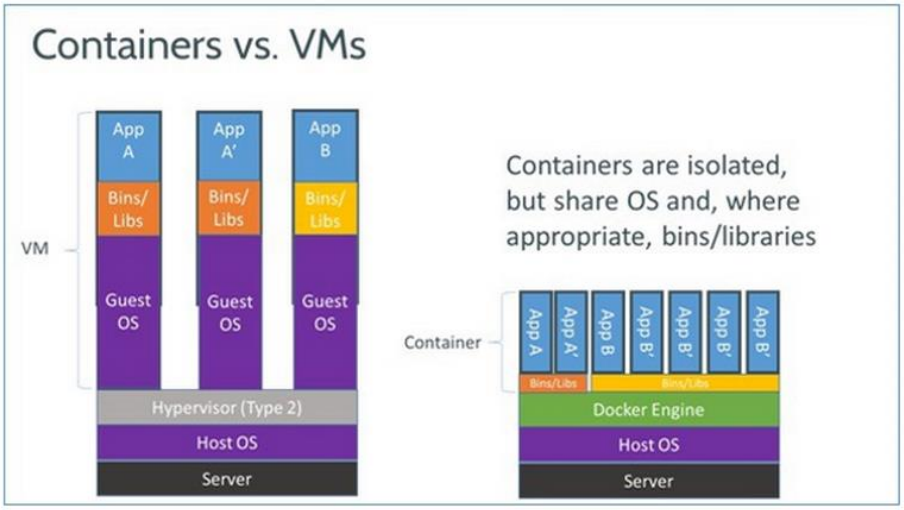
As shown in the graphic above, all access to Azure Storage, both services and data, is done through the storage account.

Docker containers

- Docker enables developers to easily pack, ship, and run any application as a lightweight, portable, self-sufficient container.
- Can run virtually anywhere with an image that is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.
- Available for both Linux and Windows based apps, containerized software will always run the same, regardless of the environment.
- Containers isolate software from its surroundings, for example differences between development and staging environments and help reduce conflicts between teams running different software on the same infrastructure



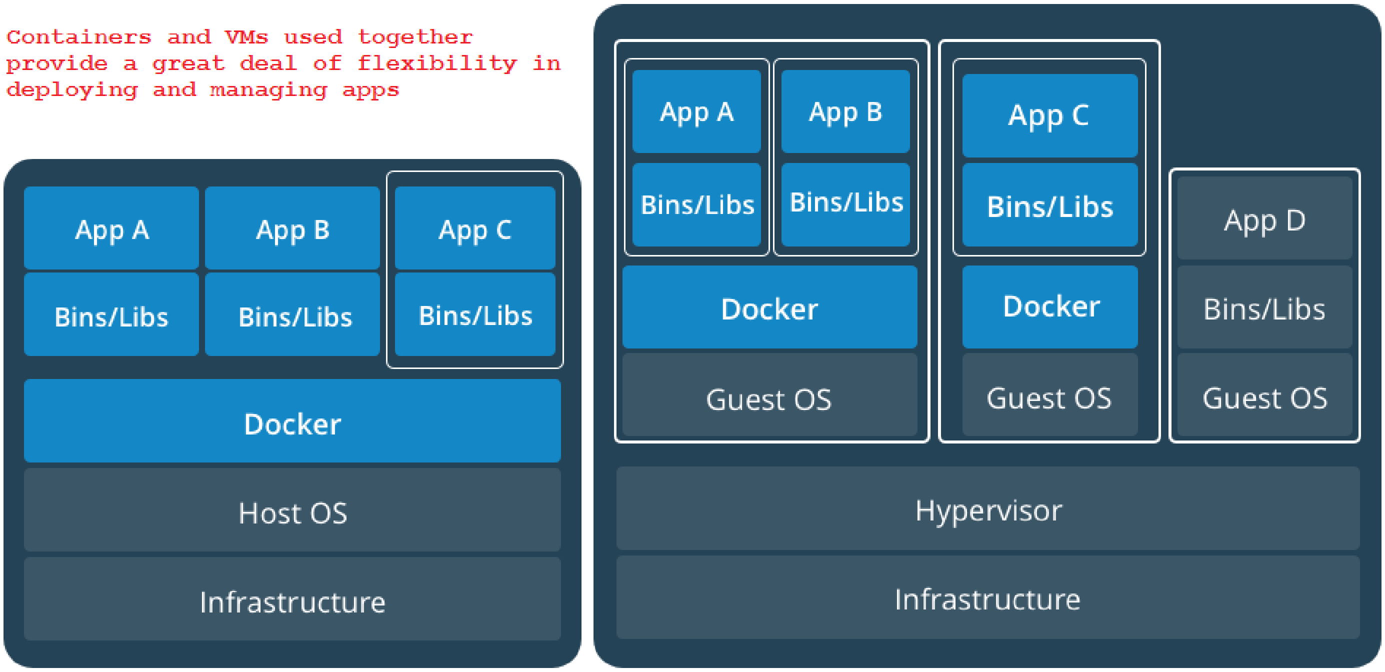
Containers & Virtual Machines


Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

Note Docker containers do not use a guest OS unlike Virtual Machines…

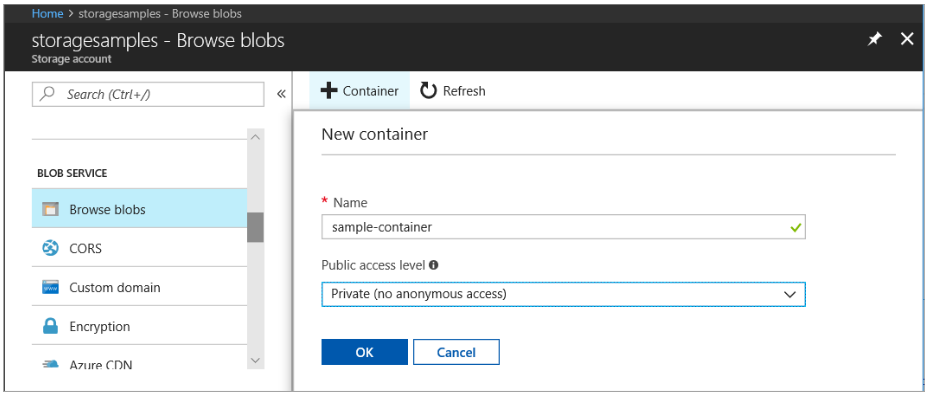
Containers and VMs used together provide a great deal of flexibility in deploying and managing apps



Azure Storage Container

To create a container in the Azure portal, select the storage account (must exist first) then follow these steps:

1. Navigate to your new storage account in the Azure portal.
2. In the left menu for the storage account, scroll to the **Blob Service** section, then select **Browse Blobs**.
3. Click the **Add Container** button.
4. Enter a name for your new container. The container name must be lowercase, must start with a letter or number and can contain only letters, numbers, and the dash (-) character. See [Naming and Referencing Containers, Blobs, and Metadata]() for more information about container and blob names.
5. Set the level of public access to the container. The default level is **Private (no anonymous access)**.
6. Click **OK** to create the container.



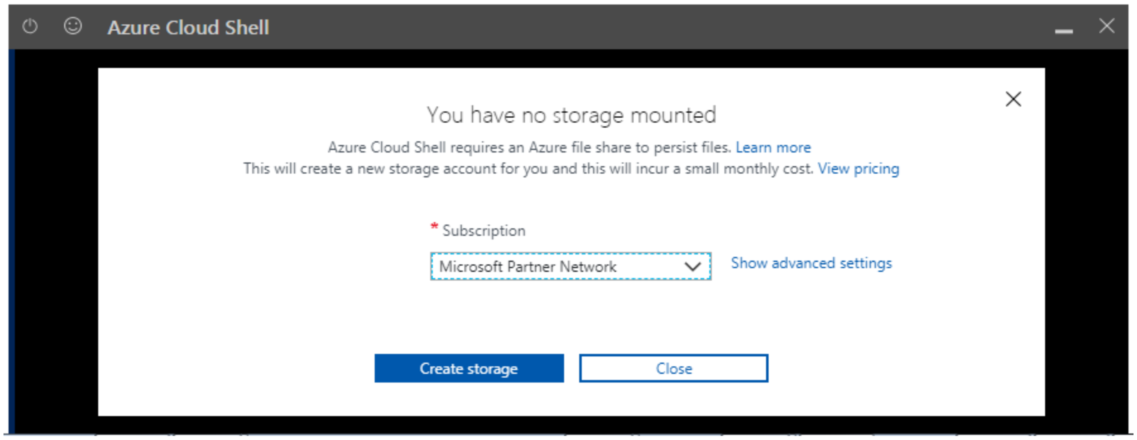
New container

* Name: sample-container

Public access level: Private (no anonymous access)

Or from Get-AzureStorageContainer: [https://docs.microsoft.com/en-us/powershell/module/azure.storage/get-azurestoragecontainer?view=azurermps-6.4.0]()

Click a "Try It" button, login, use the Create Container link....

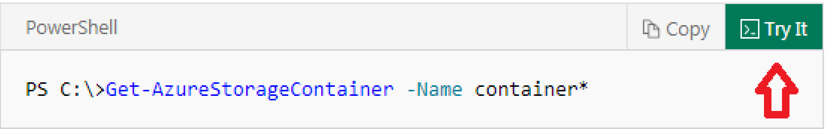


You have no storage mounted

Azure Cloud Shell requires an Azure file share to persist files. Learn more
This will create a new storage account for you and this will incur a small monthly cost. View pricing

* Subscription: Microsoft Partner Network

Example 1: Get Azure Storage blob by name



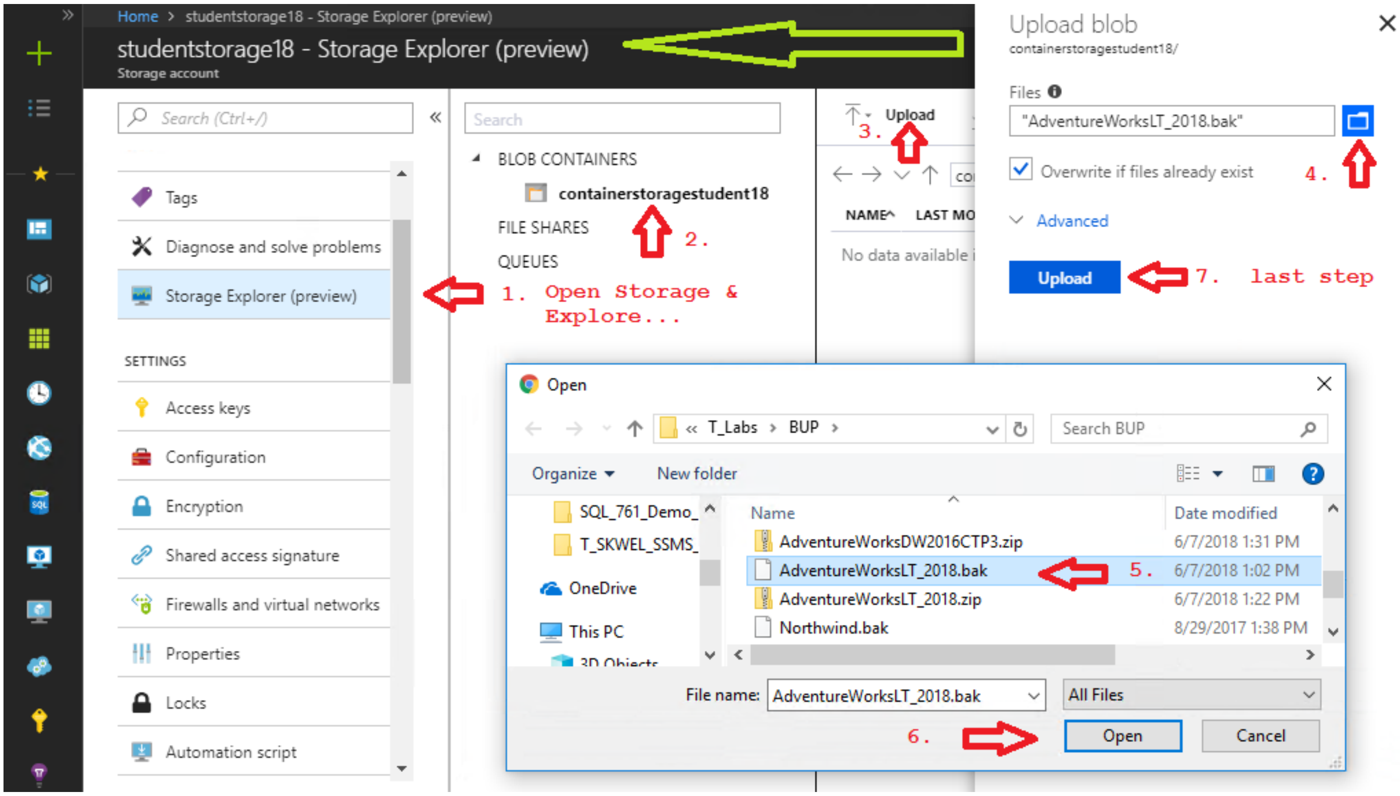
```
PS C:\>Get-AzureStorageContainer -Name container*
```

Upload a block blob

Block blobs consist of blocks of data assembled to make a blob. Most scenarios using Blob storage employ block blobs. Block blobs are ideal for storing text and binary data in the cloud, like files, images, and videos. This QuickStart shows how to work with block blobs.

To upload a block blob to your new container in the Azure portal, follow these steps:

1. In the Azure portal, navigate to the container you created in the previous section.
2. Select the container to show a list of blobs it contains. In this case, since you created a new container, it won't yet contain any blobs.
3. Click the **Upload** button to upload a blob to the container.
4. Browse your local file system to find a file to upload as a block blob, and click **Upload**.



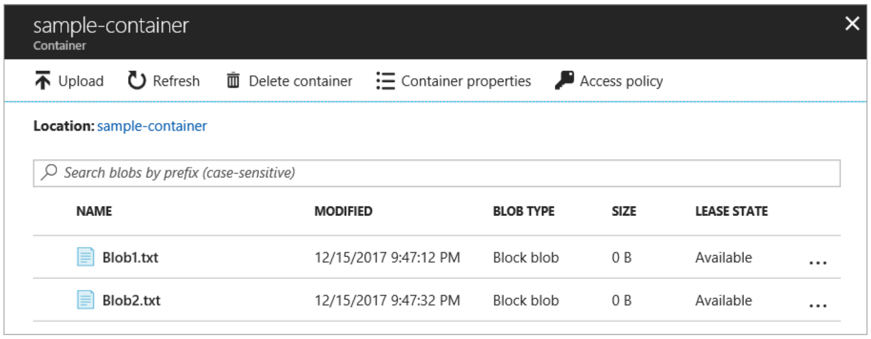
1. Open Storage & Explore...
2.
3.
4.
5.
6.
7. last step

Upload blob
containerstoragestudent18/
Files: "AdventureWorksLT_2018.bak"
☑ Overwrite if files already exist

File name: AdventureWorksLT_2018.bak
All Files

Name	Date modified
AdventureWorksDW2016CTP3.zip	6/7/2018 1:31 PM
AdventureWorksLT_2018.bak	6/7/2018 1:02 PM
AdventureWorksLT_2018.zip	6/7/2018 1:22 PM
Northwind.bak	8/29/2017 1:38 PM

5. Upload as many blobs as you like in this way. You'll see that the new blobs are now listed within the container.



Location: sample-container

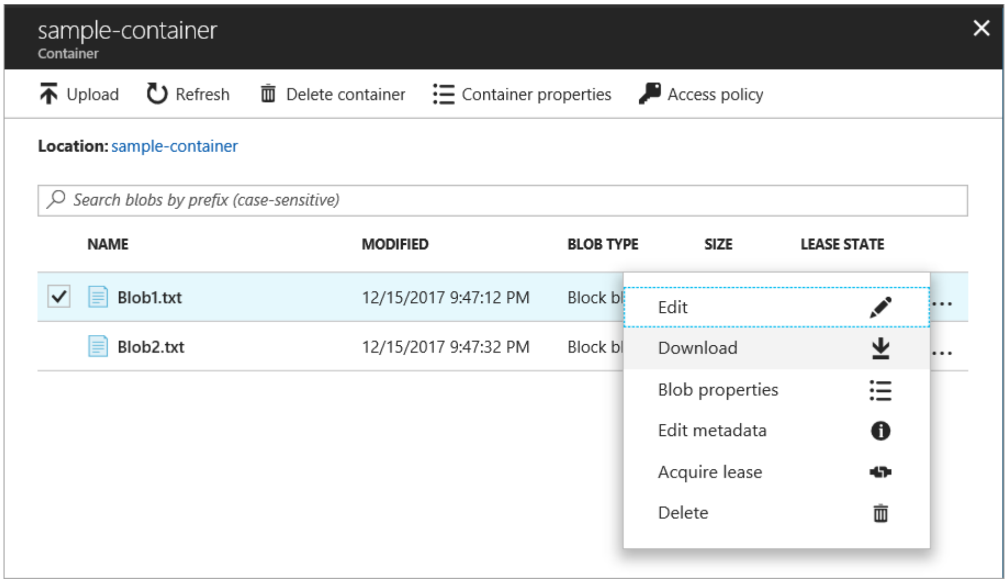
Search blobs by prefix (case-sensitive)

NAME	MODIFIED	BLOB TYPE	SIZE	LEASE STATE	
Blob1.txt	12/15/2017 9:47:12 PM	Block blob	0 B	Available	...
Blob2.txt	12/15/2017 9:47:32 PM	Block blob	0 B	Available	...

Download a block blob

You can download a block blob to display in the browser or save to your local file system. To download a block blob, follow these steps:

1. Navigate to the list of blobs that you uploaded in the previous section.
2. Select the blob to download.
3. Right-click the **More** button (**...**), and select **Download**.

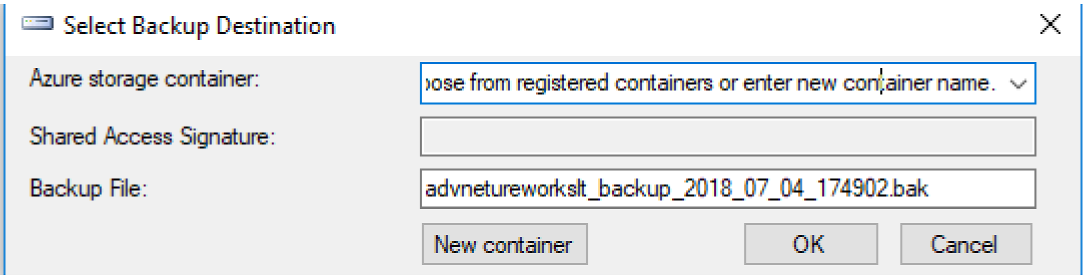


Location: sample-container

Search blobs by prefix (case-sensitive)

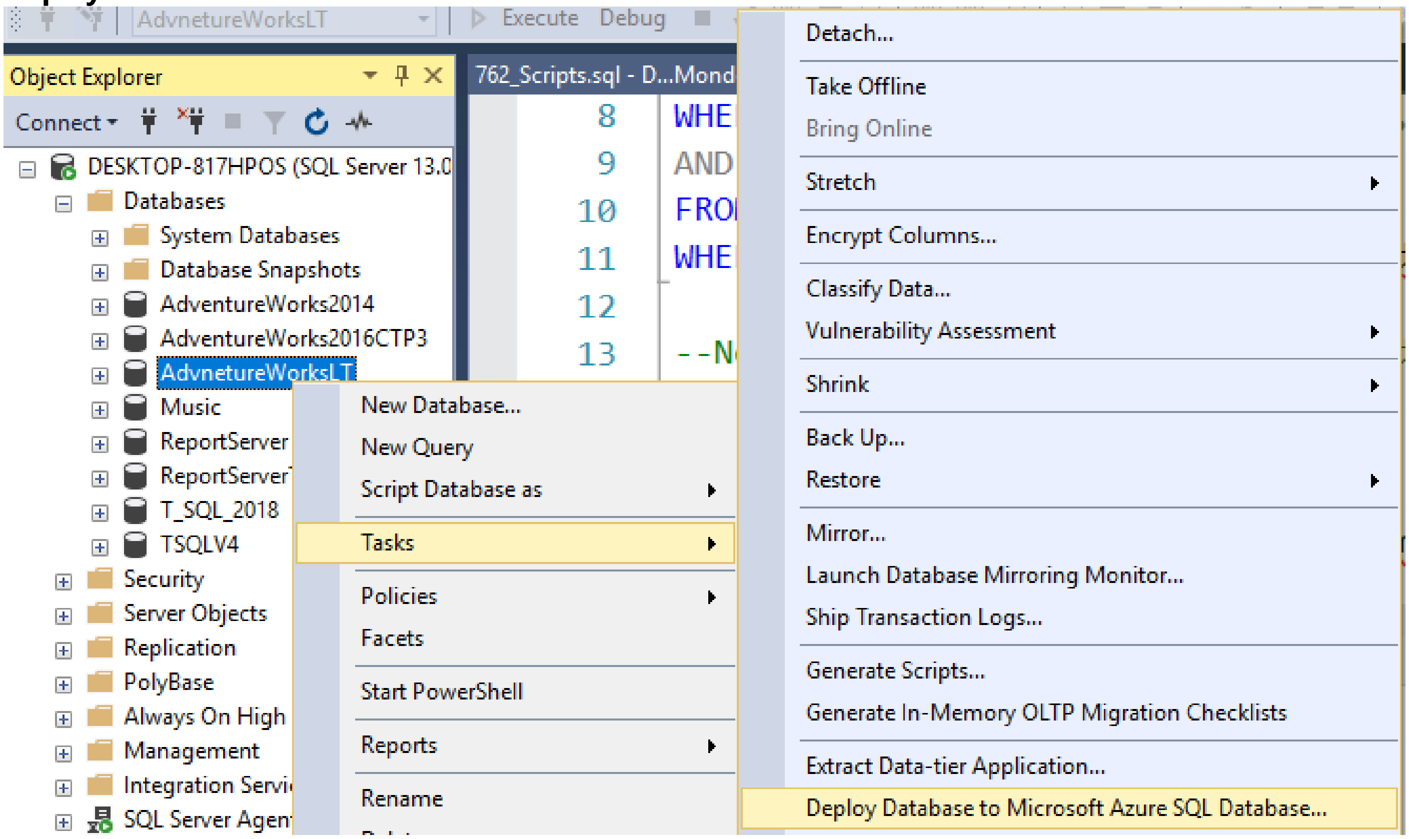
	NAME	MODIFIED	BLOB TYPE	SIZE	LEASE STATE	
☑	Blob1.txt	12/15/2017 9:47:12 PM	Block bl			...
	Blob2.txt	12/15/2017 9:47:32 PM	Block bl			...

Use this container name in tasks like backup…



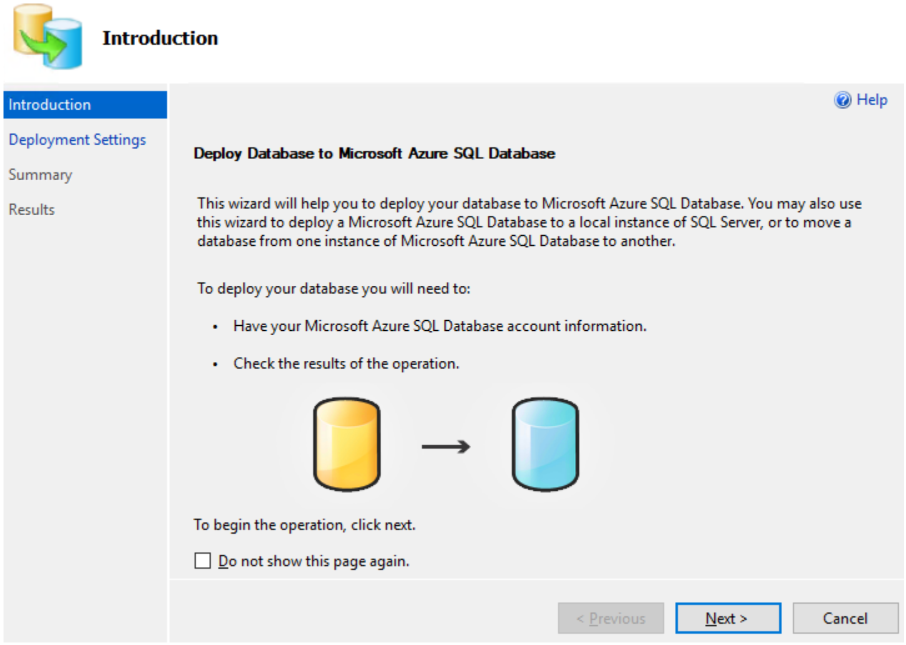
Azure storage container:	›ose from registered containers or enter new container name.
Shared Access Signature:	
Backup File:	advnetureworkslt_backup_2018_07_04_174902.bak

Deploy Database to Azure from SSMS



Follow the wizard

This deployment process creates the database on Azure and moves data with indexes etc…



Deploy Database to Microsoft Azure SQL Database

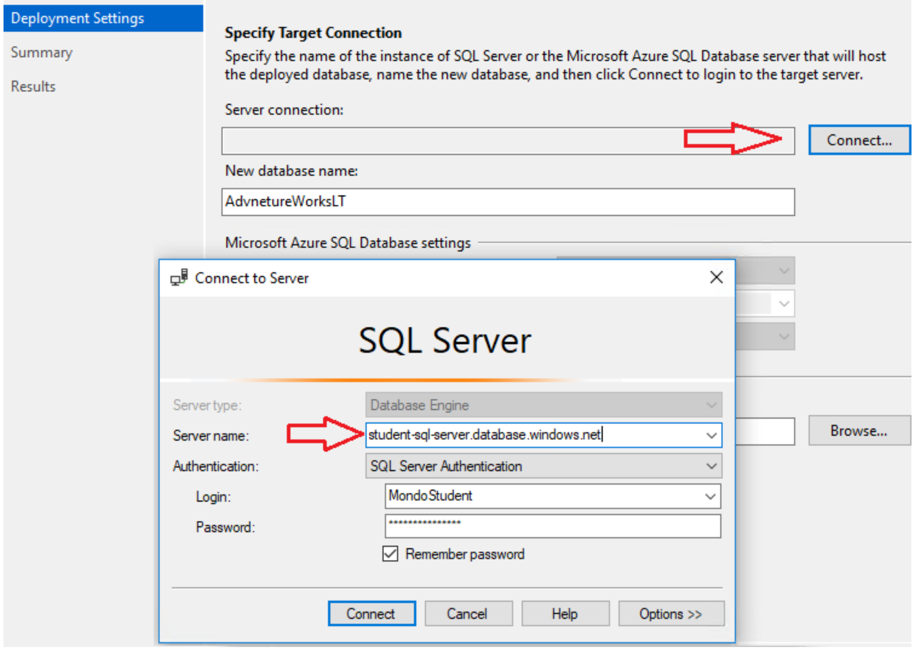
This wizard will help you to deploy your database to Microsoft Azure SQL Database. You may also use this wizard to deploy a Microsoft Azure SQL Database to a local instance of SQL Server, or to move a database from one instance of Microsoft Azure SQL Database to another.

To deploy your database you will need to:

- Have your Microsoft Azure SQL Database account information.
- Check the results of the operation.

To begin the operation, click next.

☐ Do not show this page again.



Specify Target Connection

Specify the name of the instance of SQL Server or the Microsoft Azure SQL Database server that will host the deployed database, name the new database, and then click Connect to login to the target server.

Server connection:

New database name: AdvnetureWorksLT

Microsoft Azure SQL Database settings

Server type: Database Engine

Server name: student-sql-server.database.windows.net

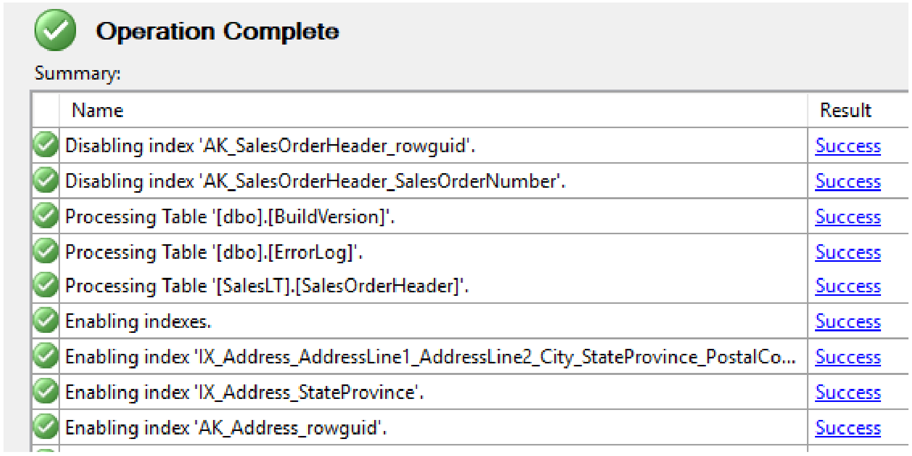
Authentication: SQL Server Authentication

Login: MondoStudent

Password: ***************

☑ Remember password

Results



Operation Complete

Summary:

Name	Result
Disabling index 'AK_SalesOrderHeader_rowguid'.	Success
Disabling index 'AK_SalesOrderHeader_SalesOrderNumber'.	Success
Processing Table '[dbo].[BuildVersion]'.	Success
Processing Table '[dbo].[ErrorLog]'.	Success
Processing Table '[SalesLT].[SalesOrderHeader]'.	Success
Enabling indexes.	Success
Enabling index 'IX_Address_AddressLine1_AddressLine2_City_StateProvince_PostalCo...	Success
Enabling index 'IX_Address_StateProvince'.	Success
Enabling index 'AK_Address_rowguid'.	Success

Connect to database

Use SQL Server Management Studio to establish a connection to your Azure SQL Database server.

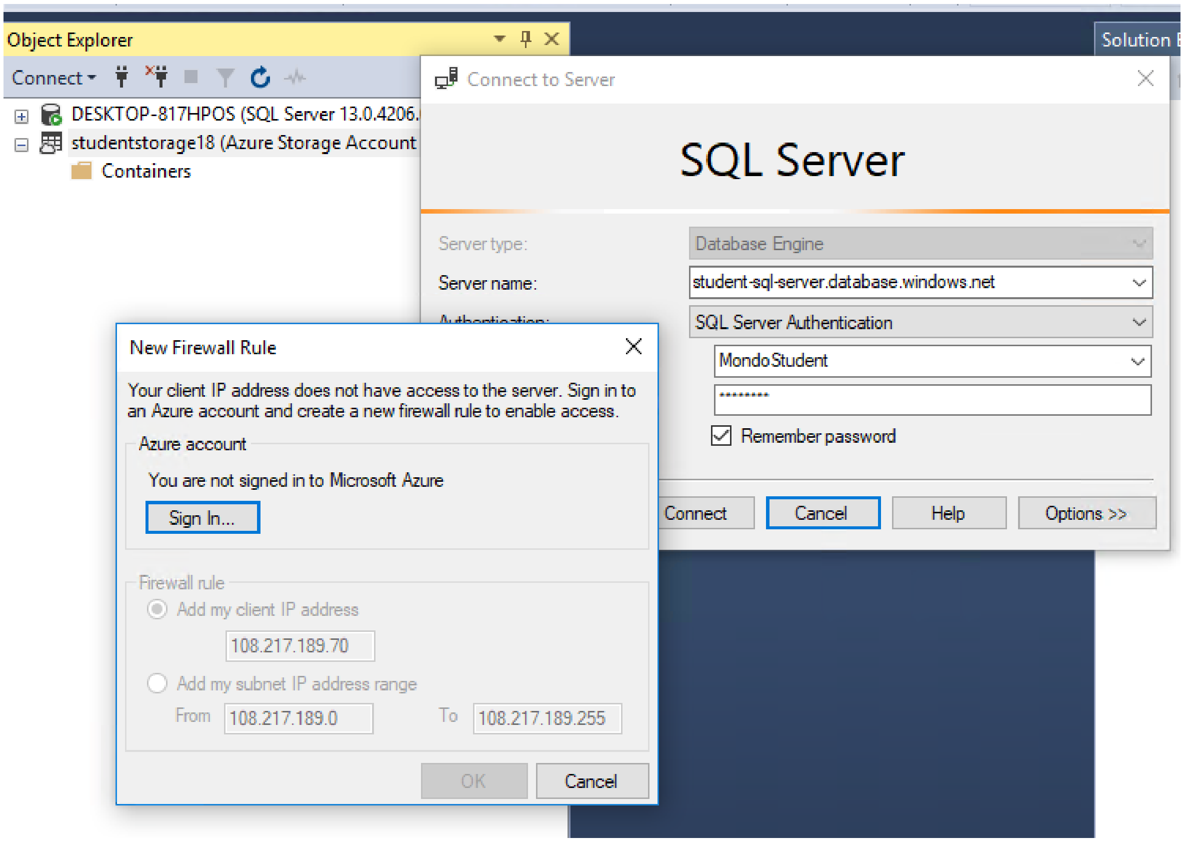
Important

An Azure SQL Database logical server listens on port 1433. If you are attempting to connect to an Azure SQL Database logical server from within a corporate firewall, this port must be open in the corporate firewall for you to successfully connect.

1. Open SQL Server Management Studio.
    a) If prompted; add IP to firewall.

2. In the **Connect to Server** dialog box, enter the following information:

Setting	Suggested value	Description
Server type	Database engine	This value is required.
Server name	The fully qualified server name	The name should be something like this: **mynewserver20170313.database.windows.net**.
Authentication	SQL Server Authentication	SQL Authentication is the only authentication type that we have configured in this tutorial.
Login	The server admin account	This is the account that you specified when you created the server.
Password	The password for your server admin account	This is the password that you specified when you created the server.



Server type: Database Engine

Server name: student-sql-server.database.windows.net

Authentication: SQL Server Authentication

MondoStudent

****

☑ Remember password

New Firewall Rule

Your client IP address does not have access to the server. Sign in to an Azure account and create a new firewall rule to enable access.

Azure account

You are not signed in to Microsoft Azure

Firewall rule

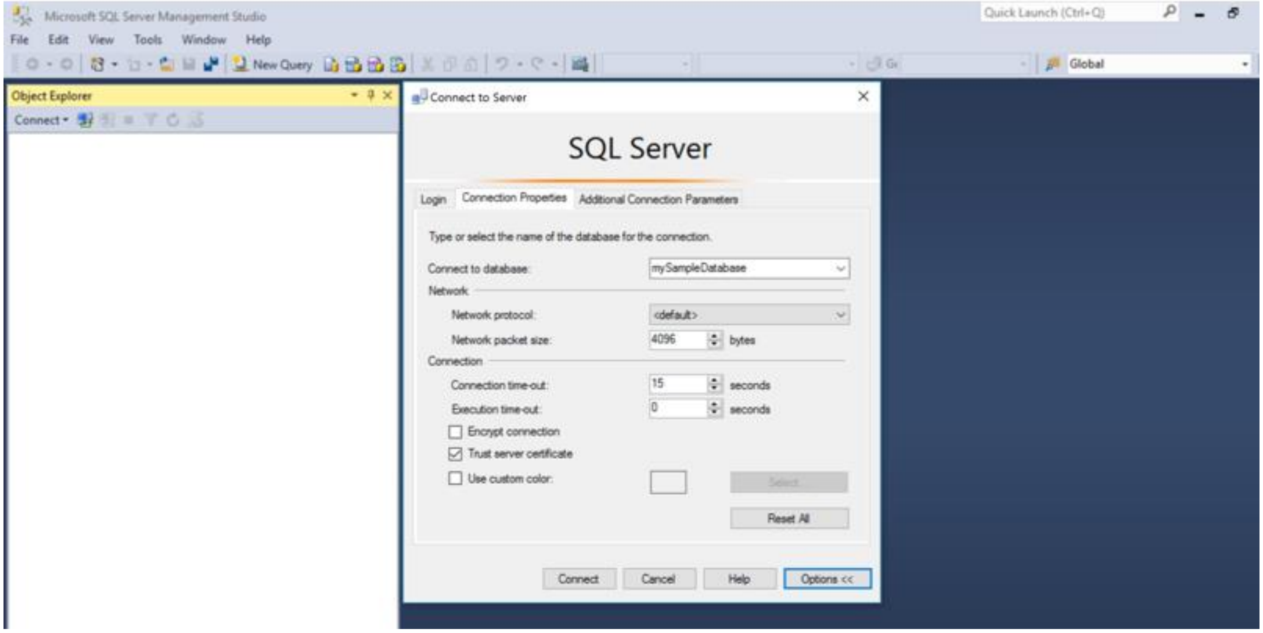
○ Add my client IP address

108.217.189.70

○ Add my subnet IP address range

From 108.217.189.0 To 108.217.189.255

3. Click **Options** in the **Connect to server** dialog box. In the **Connect to database** section, enter **mySampleDatabase** to connect to this database.



Connect to Server

SQL Server

Login | Connection Properties | Additional Connection Parameters

Type or select the name of the database for the connection.

Connect to database: mySampleDatabase

Network

Network protocol: <default>

Network packet size: 4096 bytes

Connection

Connection time-out: 15 seconds

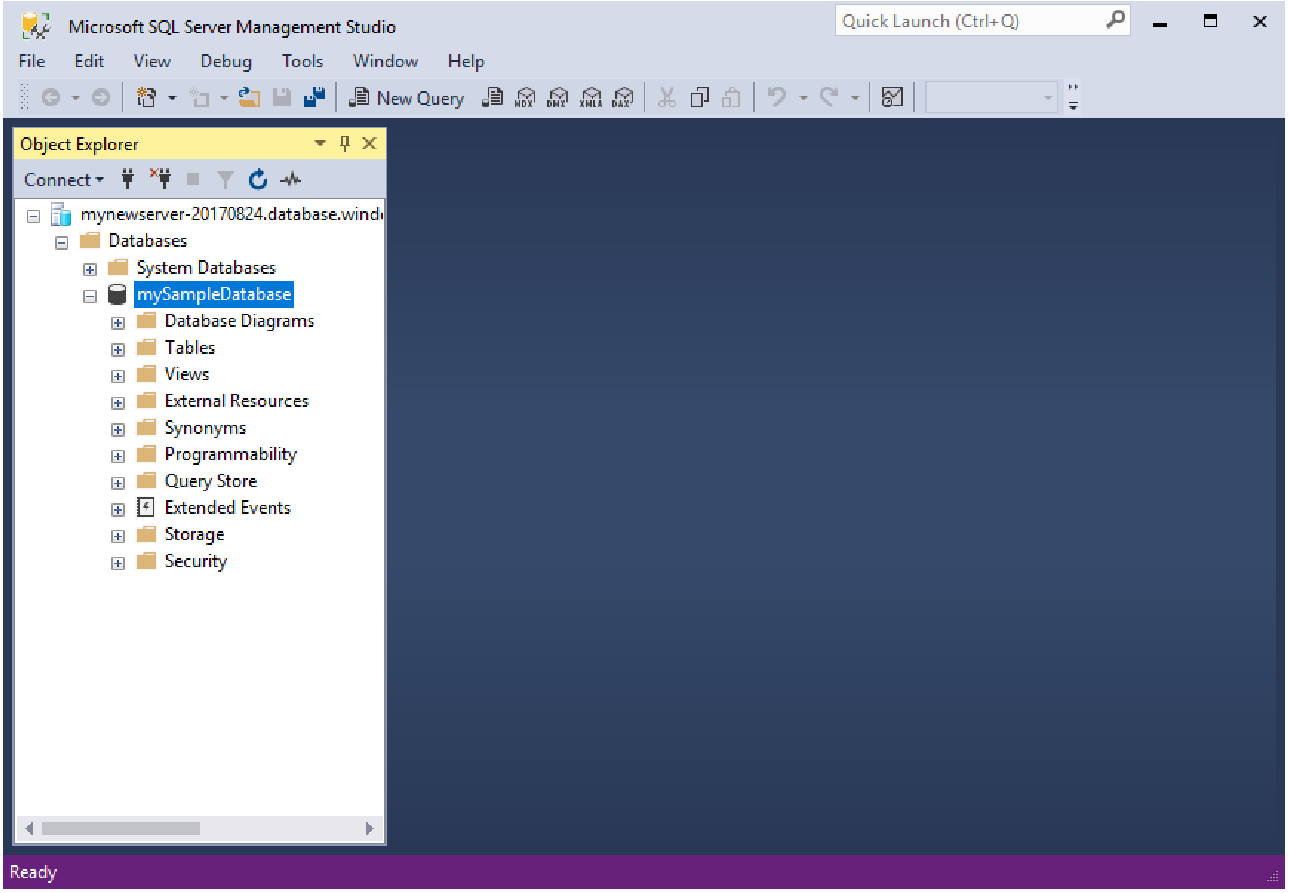
Execution time-out: 0 seconds

☐ Encrypt connection

☑ Trust server certificate

☐ Use custom color:

4. Click **Connect**. The Object Explorer window opens in SSMS.



```
⊟ mynewserver-20170824.database.windo
  ⊟ Databases
    ⊞ System Databases
    ⊟ mySampleDatabase
      ⊞ Database Diagrams
      ⊞ Tables
      ⊞ Views
      ⊞ External Resources
      ⊞ Synonyms
      ⊞ Programmability
      ⊞ Query Store
      ⊞ Extended Events
      ⊞ Storage
      ⊞ Security
```

5. In Object Explorer, expand **Databases** and then expand **mySampleDatabase** to view the objects in the sample database.

Query data

Use the following code to query for the top 20 products by category using the SELECTTransact-SQL statement.

1. In Object Explorer, right-click **mySampleDatabase** and click **New Query**. A blank query window opens that is connected to your database.
2. In the query window, enter the following query:

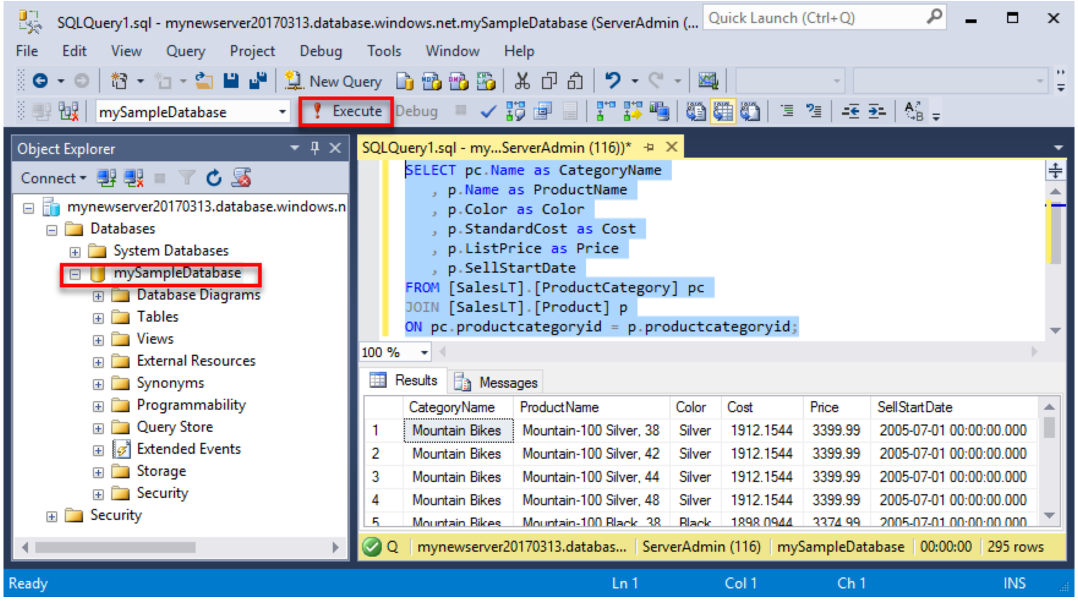
```sql
SELECT pc.Name as CategoryName, p.name as ProductName

FROM [SalesLT].[ProductCategory] pc

JOIN [SalesLT].[Product] p

ON pc.productcategoryid = p.productcategoryid;
```

3. On the toolbar, click **Execute** to retrieve data from the Product and ProductCategory tables.



```sql
SELECT pc.Name as CategoryName
    , p.Name as ProductName
    , p.Color as Color
    , p.StandardCost as Cost
    , p.ListPrice as Price
    , p.SellStartDate
FROM [SalesLT].[ProductCategory] pc
JOIN [SalesLT].[Product] p
ON pc.productcategoryid = p.productcategoryid;
```

	CategoryName	ProductName	Color	Cost	Price	SellStartDate
1	Mountain Bikes	Mountain-100 Silver, 38	Silver	1912.1544	3399.99	2005-07-01 00:00:00.000
2	Mountain Bikes	Mountain-100 Silver, 42	Silver	1912.1544	3399.99	2005-07-01 00:00:00.000
3	Mountain Bikes	Mountain-100 Silver, 44	Silver	1912.1544	3399.99	2005-07-01 00:00:00.000
4	Mountain Bikes	Mountain-100 Silver, 48	Silver	1912.1544	3399.99	2005-07-01 00:00:00.000
5	Mountain Bikes	Mountain-100 Black, 38	Black	1898.0944	3374.99	2005-07-01 00:00:00.000

Q mynewserver20170313.databas... | ServerAdmin (116) | mySampleDatabase | 00:00:00 | 295 rows

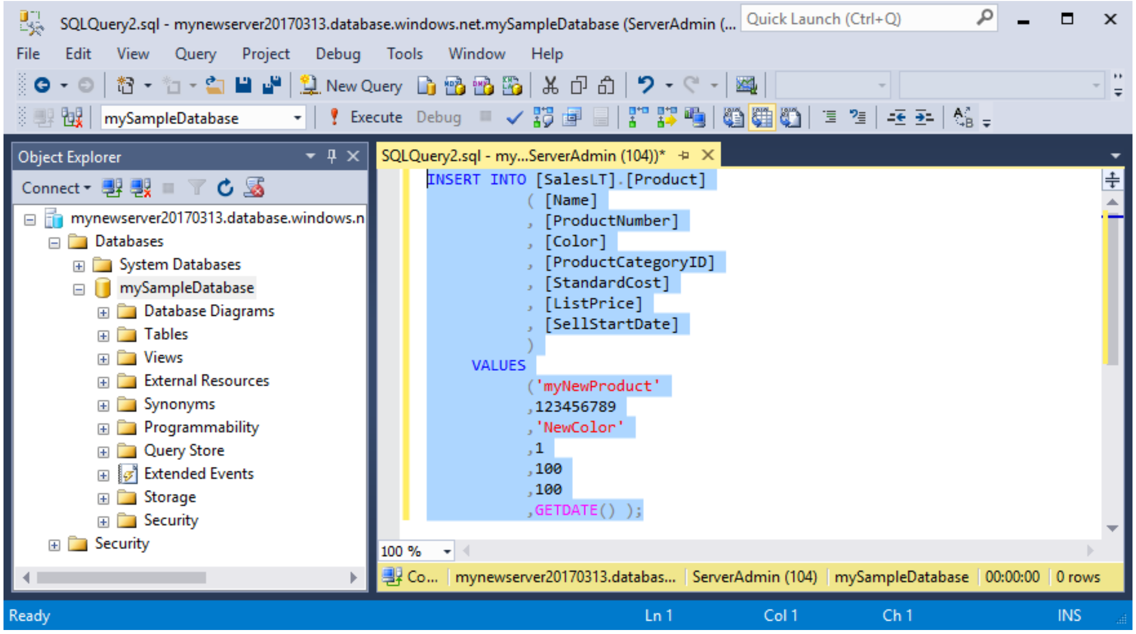
Insert data

Use the following code to insert a new product into the SalesLT.Product table using the INSERT Transact-SQL statement.

1. In the query window, replace the previous query with the following query:

```
INSERT INTO [SalesLT].[Product]
        ( [Name]
        , [ProductNumber]
        , [Color]
        , [ProductCategoryID]
        , [StandardCost]
        , [ListPrice]
        , [SellStartDate]
        )
  VALUES
        ('myNewProduct'
        ,123456789
        ,'NewColor'
        ,1
        ,100
        ,100
        ,GETDATE() );
```

2. On the toolbar, click **Execute** to insert a new row in the Product table.


SQLQuery2.sql - mynewserver20170313.database.windows.net.mySampleDatabase (ServerAdmin (...
Quick Launch (Ctrl+Q)
File Edit View Query Project Debug Tools Window Help
New Query
mySampleDatabase
Execute Debug
Object Explorer
Connect
mynewserver20170313.database.windows.n
Databases
System Databases
mySampleDatabase
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Query Store
Extended Events
Storage
Security
Security
SQLQuery2.sql - my...ServerAdmin (104))*
100 %
Co... mynewserver20170313.databas... ServerAdmin (104) mySampleDatabase 00:00:00 0 rows
Ready Ln 1 Col 1 Ch 1 INS

```
INSERT INTO [SalesLT].[Product]
            ( [Name]
            , [ProductNumber]
            , [Color]
            , [ProductCategoryID]
            , [StandardCost]
            , [ListPrice]
            , [SellStartDate]
            )
     VALUES
            ('myNewProduct'
            ,123456789
            ,'NewColor'
            ,1
            ,100
            ,100
            ,GETDATE() );
```

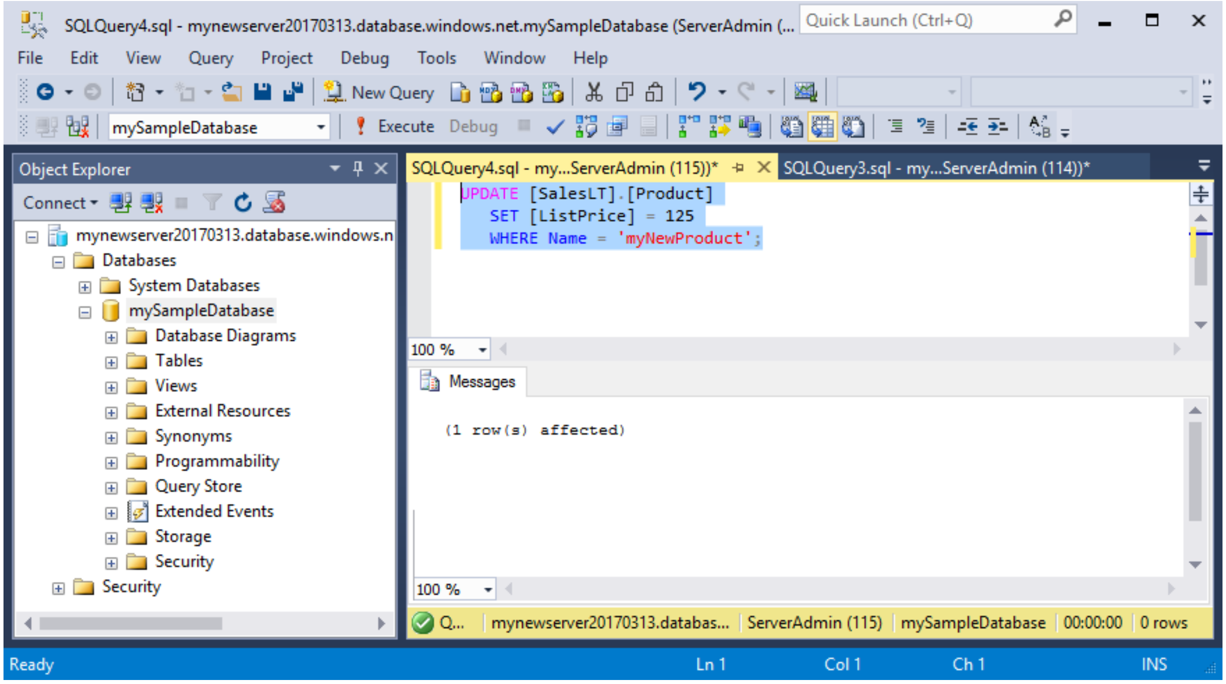
Update data

Use the following code to update the new product that you previously added using the UPDATE Transact-SQL statement.

1. In the query window, replace the previous query with the following query:

```
UPDATE [SalesLT].[Product]
SET [ListPrice] = 125
WHERE Name = 'myNewProduct';
```

2. On the toolbar, click **Execute** to update the specified row in the Product table.



```
UPDATE [SalesLT].[Product]
   SET [ListPrice] = 125
   WHERE Name = 'myNewProduct';
```

```
(1 row(s) affected)
```

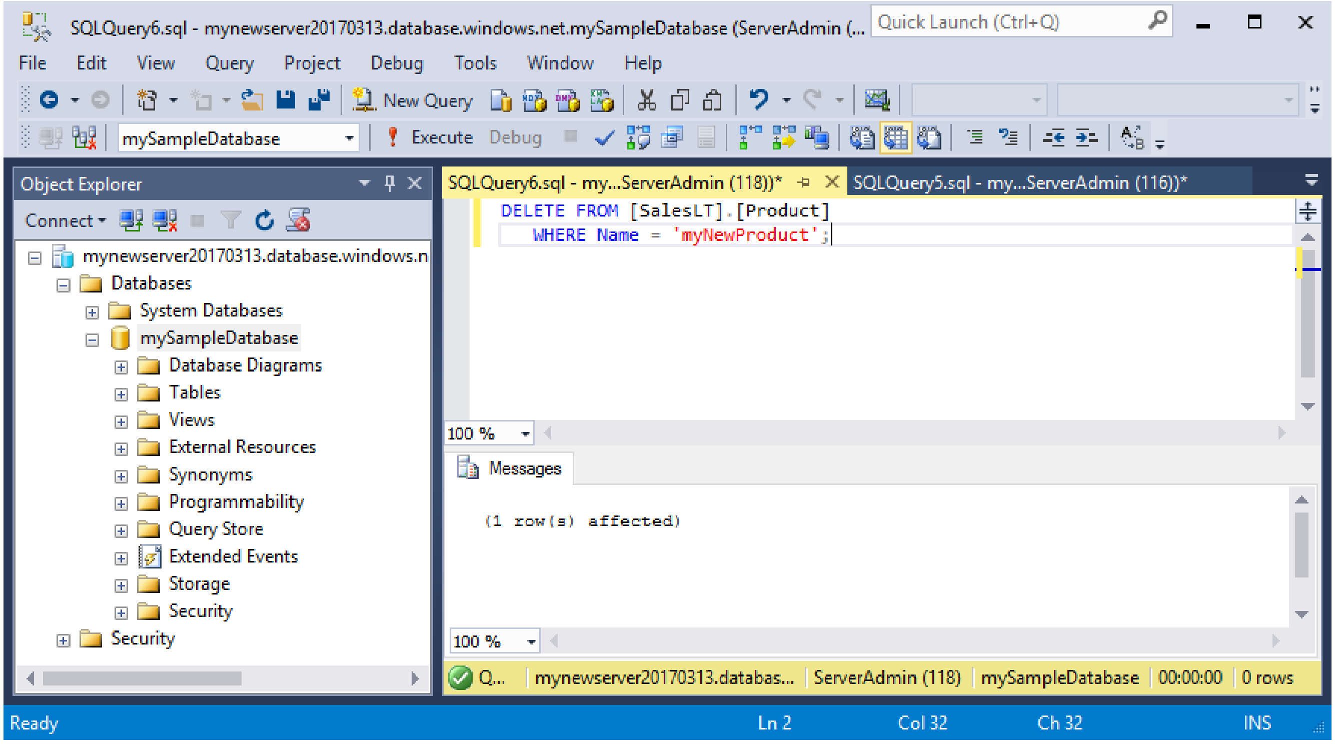
Delete data

Use the following code to delete the new product that you previously added using the DELETE Transact-SQL statement.

1. In the query window, replace the previous query with the following query:

```
DELETE FROM [SalesLT].[Product]
WHERE Name = 'myNewProduct';
```

2. On the toolbar, click **Execute** to delete the specified row in the Product table.



```sql
DELETE FROM [SalesLT].[Product]
   WHERE Name = 'myNewProduct';
```

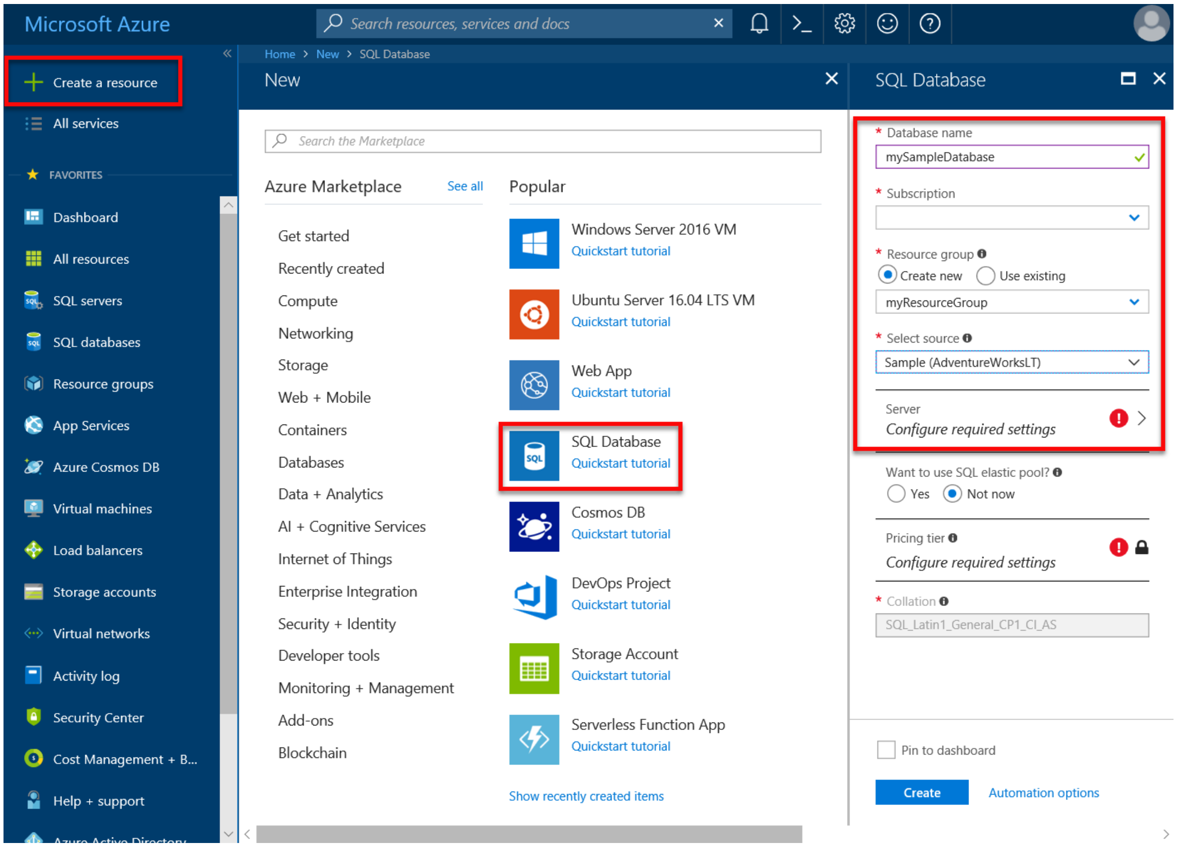
```
(1 row(s) affected)
```

Next steps

- To learn about creating and managing servers and databases with Transact-SQL, see Learn about Azure SQL Database servers and databases.
- For information about SSMS, see Use SQL Server Management Studio.
- To connect and query using the Azure portal, see Connect and query with the Azure portal SQL Query editor.
- To connect and query using Visual Studio Code, see Connect and query with Visual Studio Code.
- To connect and query using .NET, see Connect and query with .NET.
- To connect and query using PHP, see Connect and query with PHP.
- To connect and query using Node.js, see Connect and query with Node.js.
- To connect and query using Java, see Connect and query with Java.
- To connect and query using Python, see Connect and query with Python.
- To connect and query using Ruby, see Connect and query with Ruby.

Create the SQL database containing the Adventure Works LT sample data.

1. Click **Create a resource** in the upper left-hand corner of the Azure portal.
2. Select **Databases** from the **New** page and select **Create** under **SQL Database** on the **New** page.



* Database name: mySampleDatabase

* Subscription:

* Resource group: ● Create new ○ Use existing
myResourceGroup

* Select source: Sample (AdventureWorksLT)

Server: *Configure required settings*

Want to use SQL elastic pool? ○ Yes ● Not now

Pricing tier: *Configure required settings*

* Collation: SQL_Latin1_General_CP1_CI_AS

□ Pin to dashboard

3. Fill out the SQL Database form with the following information, as shown on the preceding image:

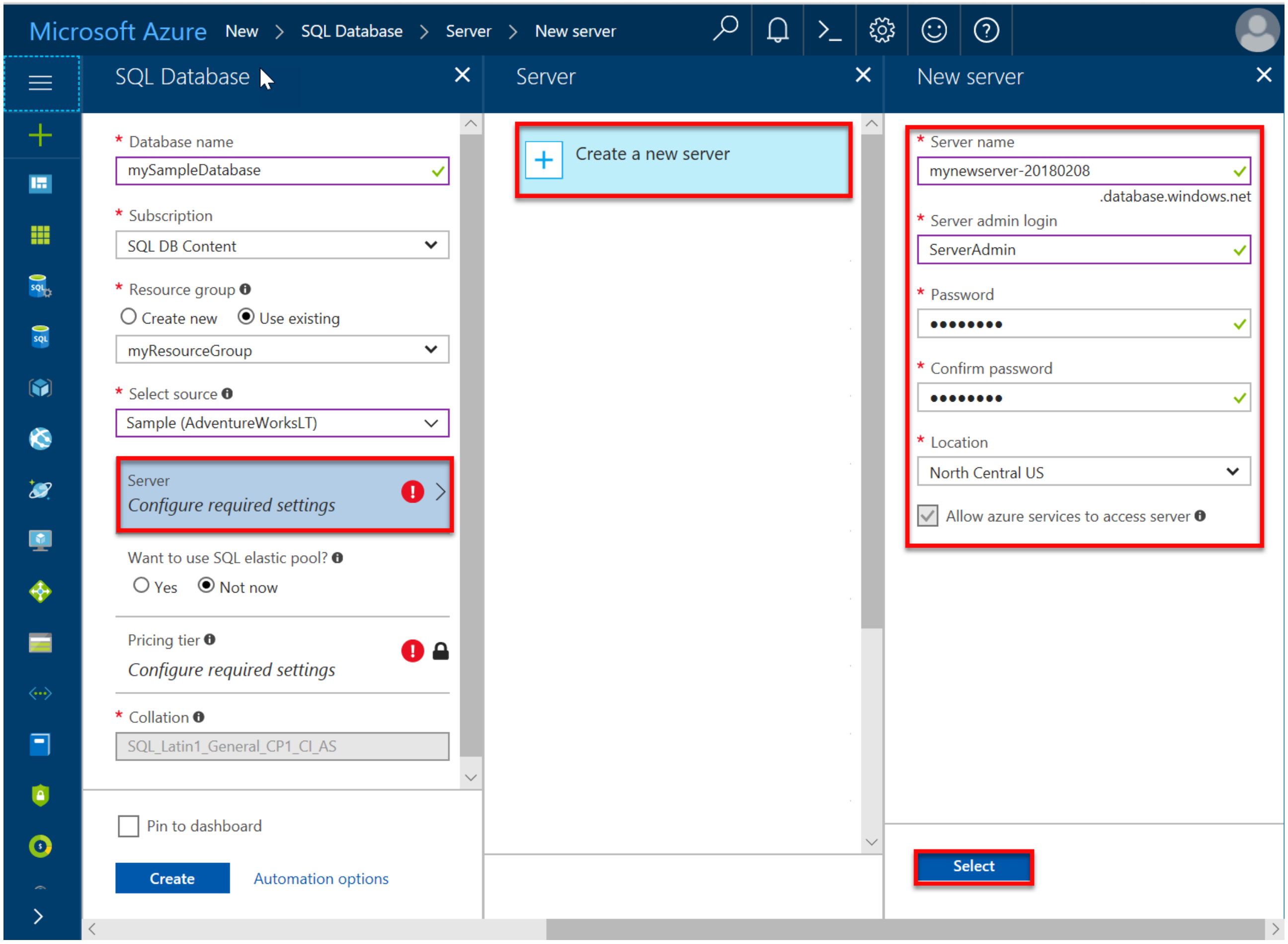
Setting	**Suggested value**	**Description**
Database name	mySampleDatabase	For valid database names, see Database Identifiers.
Subscription	Your subscription	For details about your subscriptions, see Subscriptions.
Resource group	myResourceGroup	For valid resource group names, see Naming rules and restrictions.

Setting	Suggested value	Description
Select source	Sample (AdventureWorksLT)	Loads the AdventureWorksLT schema and data into your new database

4. **Important:** You must select the sample database on this form because it is used in the remainder of this QuickStart.
5. Under **Server**, click **Configure required settings** and fill out the SQL server (logical server) form with the following information, as shown on the following image:

Setting	Suggested value	Description
Server name	Any globally unique name	For valid server names, see Naming rules and restrictions.
Server admin login	Any valid name	For valid login names, see Database Identifiers.
Password	Any valid password	Your password must have at least 8 characters and must contain characters from three of the following categories: upper case characters, lower case characters, numbers, and non-alphanumeric characters.
Subscription	Your subscription	For details about your subscriptions, see Subscriptions.
Resource group	myResourceGroup	For valid resource group names, see Naming rules and restrictions.
Location	Any valid location	For information about regions, see Azure Regions.

6. **Important**
7. The server admin login and password that you specify here are required to log in to the server and its databases later in this QuickStart. Remember or record this information for later use.



SQL Database

- * Database name: mySampleDatabase
- * Subscription: SQL DB Content
- * Resource group: ○ Create new ● Use existing
- myResourceGroup
- * Select source: Sample (AdventureWorksLT)
- Server: *Configure required settings*
- Want to use SQL elastic pool? ○ Yes ● Not now
- Pricing tier: *Configure required settings*
- * Collation: SQL_Latin1_General_CP1_CI_AS
- □ Pin to dashboard

Server

- + Create a new server

New server

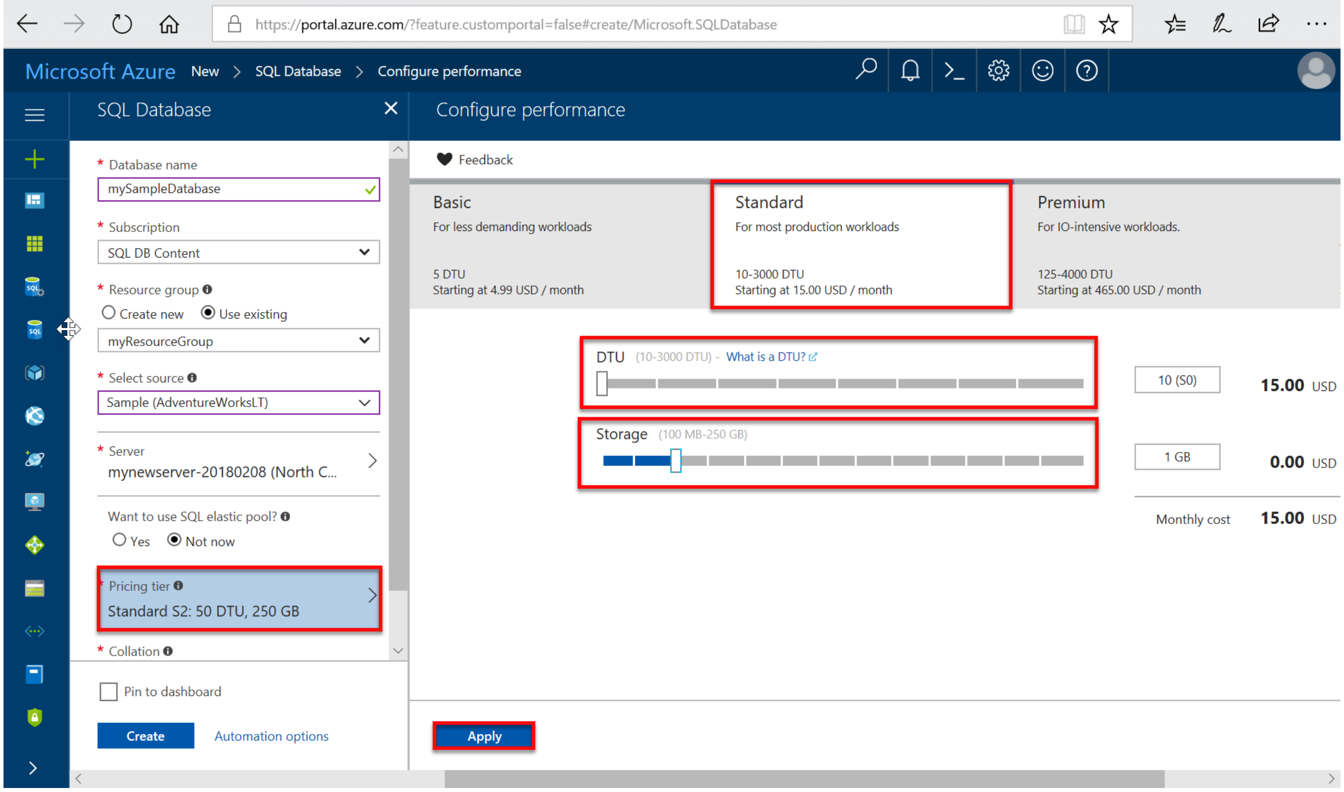
- * Server name: mynewserver-20180208 .database.windows.net
- * Server admin login: ServerAdmin
- * Password: ••••••••
- * Confirm password: ••••••••
- * Location: North Central US
- ☑ Allow azure services to access server

8. When you have completed the form, click **Select**.
9. Click **Pricing tier** to specify the service tier, the number of DTUs, and the amount of storage. Explore the options for the amount of DTUs and storage that is available to you for each service tier.

   Important

   More than 1 TB of storage in the Premium tier is currently available in all regions except the following: UK North, West Central US, UK South2, China East, USDoDCentral, Germany Central, USDoDEast, US Gov Southwest, US Gov South Central, Germany Northeast, China North, US Gov East. In other regions, the storage max in the Premium tier is limited to 1 TB. See **P11-P15 Current Limitations**.

10. For this QuickStart, select the **Standard** service tier and then use the slider to select **10 DTUs (S0)** and **1** GB of storage.



https://portal.azure.com/?feature.customportal=false#create/Microsoft.SQLDatabase

* Database name: mySampleDatabase

* Subscription: SQL DB Content

* Resource group: ○ Create new ● Use existing

myResourceGroup

* Select source: Sample (AdventureWorksLT)

* Server: mynewserver-20180208 (North C...

Want to use SQL elastic pool? ○ Yes ● Not now

* Pricing tier: Standard S2: 50 DTU, 250 GB

* Collation

□ Pin to dashboard

Basic: For less demanding workloads. 5 DTU. Starting at 4.99 USD / month

Standard: For most production workloads. 10-3000 DTU. Starting at 15.00 USD / month

Premium: For IO-intensive workloads. 125-4000 DTU. Starting at 465.00 USD / month

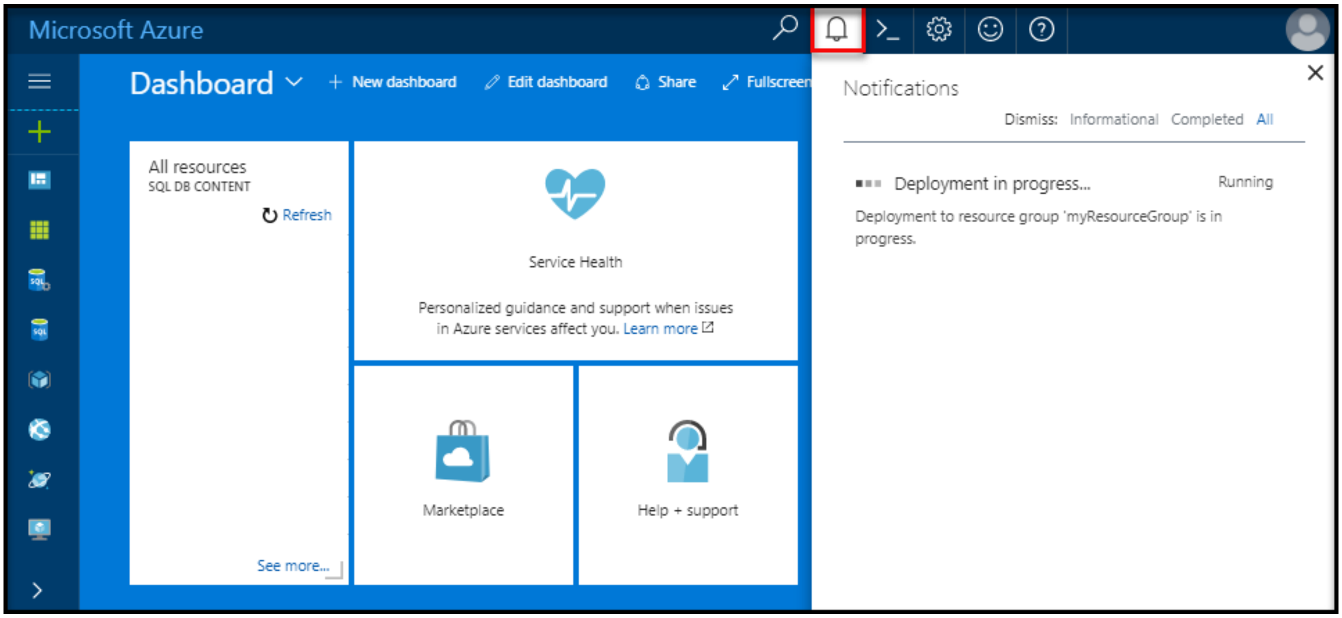
DTU (10-3000 DTU) - What is a DTU?	10 (S0)	15.00 USD
Storage (100 MB-250 GB)	1 GB	0.00 USD
	Monthly cost	15.00 USD

11. Accept the preview terms to use the **Add-on Storage** option.

    Important

    More than 1 TB of storage in the Premium tier is currently available in all regions except the following: West Central US, China East, USDoDCentral, USGov Iowa, Germany Central, USDoDEast, US Gov Southwest, Germany Northeast, China North. In other regions, the storage max in the Premium tier is limited to 1 TB. See **P11-P15 Current Limitations**.

12. After selecting the server tier, the number of DTUs, and the amount of storage, click **Apply**.
13. Now that you have completed the SQL Database form, click **Create** to provision the database. Provisioning takes a few minutes.
14. On the toolbar, click **Notifications** to monitor the deployment process.

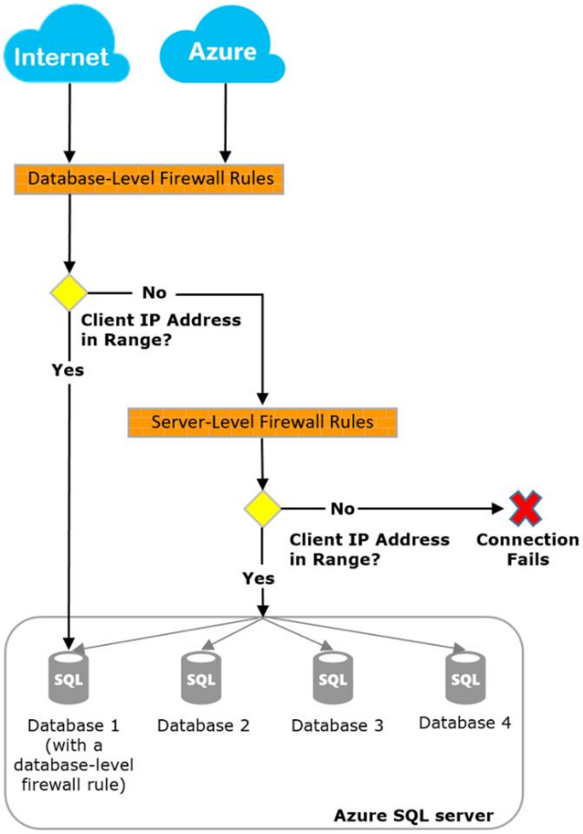


Deployment in progress... Running

Deployment to resource group 'myResourceGroup' is in progress.

Create a server-level firewall rule

The SQL Database service creates a firewall at the server-level that prevents external applications and tools from connecting to the server or any databases on the server unless a firewall rule is created to open the firewall for specific IP addresses. Follow these steps to create a SQL Database server-level firewall rule for your client's IP address and enable external connectivity through the SQL Database firewall for your IP address only.

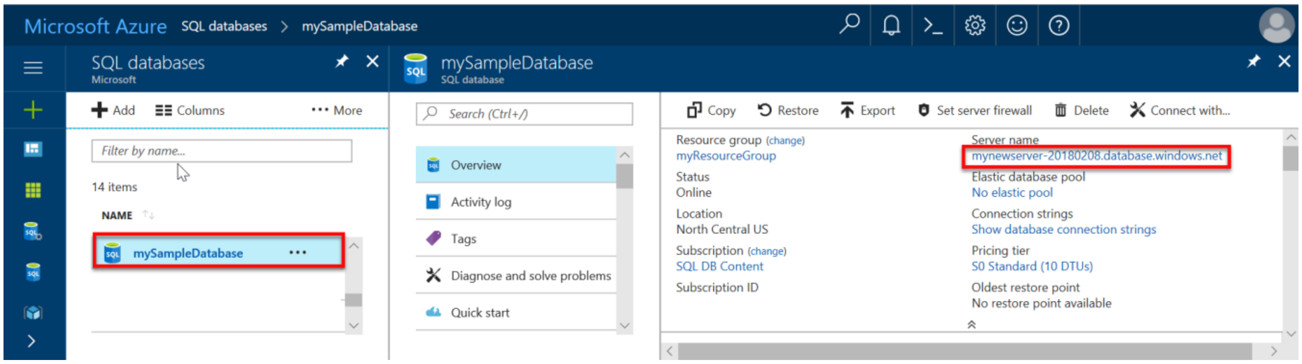
Firewall rules & IP Address range To begin using your Azure SQL server, you must specify one or more server-level firewall rules that enable access to your Azure SQL server. Use the firewall rules to specify which IP address ranges from the Internet are allowed, and whether Azure applications can attempt to connect to your Azure SQL server.	

Internet
Azure
Database-Level Firewall Rules
No
Client IP Address in Range?
Yes
Server-Level Firewall Rules
No
Client IP Address in Range?
Connection Fails
Yes
SQL
Database 1 (with a database-level firewall rule)
Database 2
Database 3
Database 4
Azure SQL server

Note

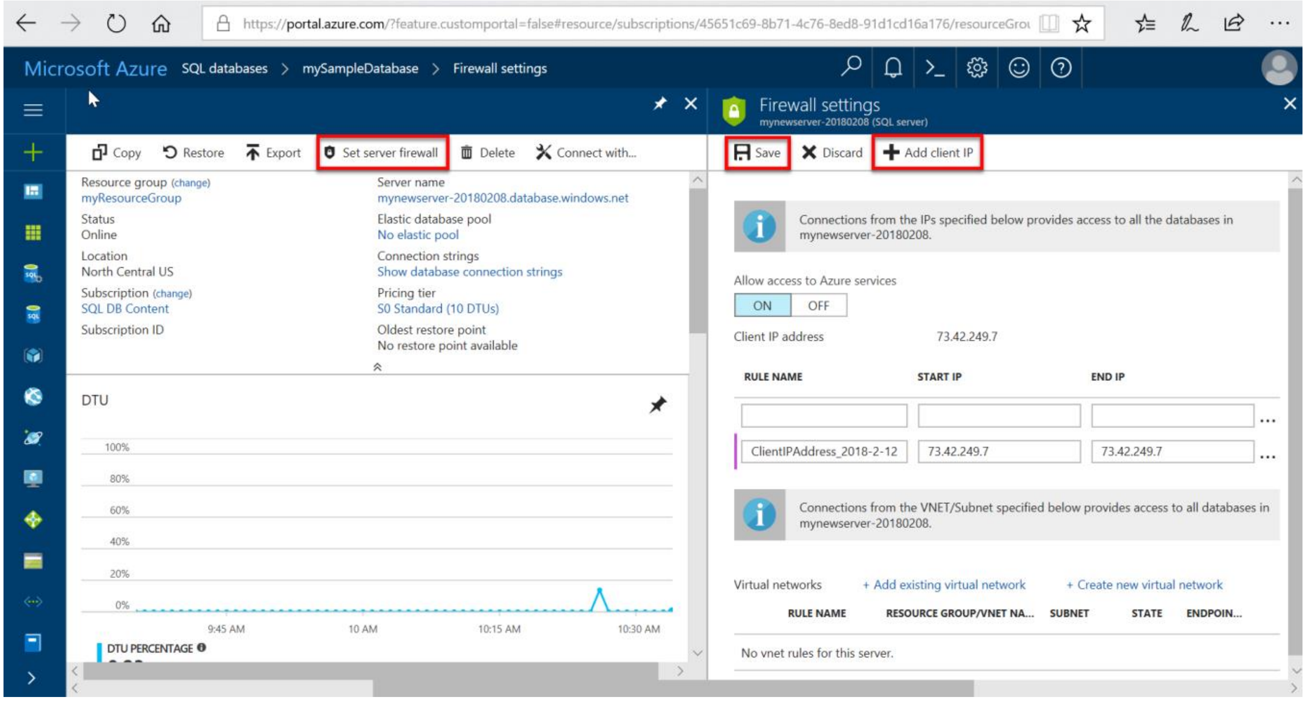
SQL Database communicates over port 1433. If you are trying to connect from within a corporate network, outbound traffic over port 1433 may not be allowed by your network's firewall. If so, you cannot connect to your Azure SQL Database server unless your IT department opens port 1433.

1. After the deployment completes, click **SQL databases** from the left-hand menu and then click **mySampleDatabase** on the **SQL databases** page. The overview page for your database opens, showing you the fully qualified server name (such as **mynewserver-20170824.database.windows.net**) and provides options for further configuration.
2. Copy this fully qualified server name for use to connect to your server and its databases in subsequent QuickStart's.



Resource group (change): myResourceGroup
Status: Online
Location: North Central US
Subscription (change): SQL DB Content
Subscription ID
Server name: mynewserver-20180208.database.windows.net
Elastic database pool: No elastic pool
Connection strings: Show database connection strings
Pricing tier: S0 Standard (10 DTUs)
Oldest restore point: No restore point available

3. Click **Set server firewall** on the toolbar as shown in the previous image. The **Firewall settings** page for the SQL Database server opens.


https://portal.azure.com/?feature.customportal=false#resource/subscriptions/45651c69-8b71-4c76-8ed8-91d1cd16a176/resourceGrou
Microsoft Azure
SQL databases > mySampleDatabase > Firewall settings
Copy
Restore
Export
Set server firewall
Delete
Connect with...
DTU
100%
80%
60%
40%
20%
0%
9:45 AM
10 AM
10:15 AM
10:30 AM
DTU PERCENTAGE
Firewall settings
mynewserver-20180208 (SQL server)
Save
Discard
Add client IP
+ Add existing virtual network
+ Create new virtual network

Resource group (change): myResourceGroup
Status: Online
Location: North Central US
Subscription (change): SQL DB Content
Subscription ID
Server name: mynewserver-20180208.database.windows.net
Elastic database pool: No elastic pool
Connection strings: Show database connection strings
Pricing tier: S0 Standard (10 DTUs)
Oldest restore point: No restore point available

Connections from the IPs specified below provides access to all the databases in mynewserver-20180208.

Allow access to Azure services: ON OFF

Client IP address: 73.42.249.7

RULE NAME	START IP	END IP
ClientIPAddress_2018-2-12	73.42.249.7	73.42.249.7

Connections from the VNET/Subnet specified below provides access to all databases in mynewserver-20180208.

Virtual networks

RULE NAME	RESOURCE GROUP/VNET NA...	SUBNET	STATE	ENDPOIN...

No vnet rules for this server.

4. Click **Add client IP** on the toolbar to add your current IP address to a new firewall rule. A firewall rule can open port 1433 for a single IP address or a range of IP addresses.
5. Click **Save**. A server-level firewall rule is created for your current IP address opening port 1433 on the logical server.
6. Click **OK** and then close the **Firewall settings** page.

You can now connect to the SQL Database server and its databases using SQL Server Management Studio or another tool of your choice from this IP address using the server admin account created previously.

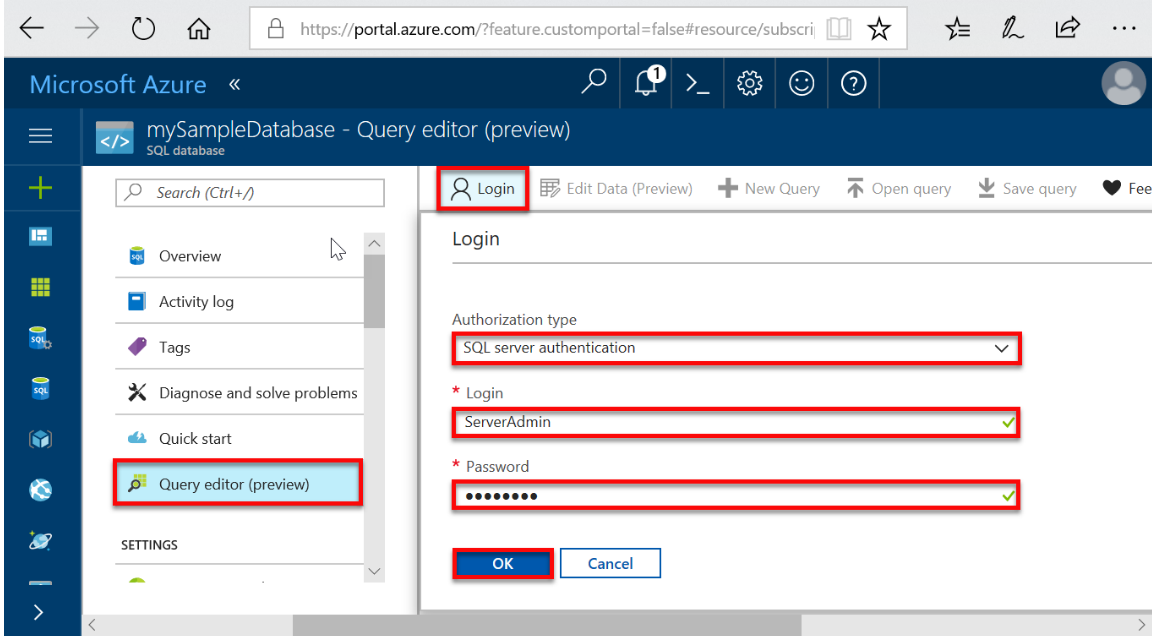
Important

By default, access through the SQL Database firewall is enabled for all Azure services. Click **OFF** on this page to disable for all Azure services.

Query the SQL database

Now that you have created a sample database in Azure, let's use the built-in query tool within the Azure portal to confirm that you can connect to the database and query the data.

1. On the SQL Database page for your database, click **Query editor (preview)** in the left-hand menu and then click **Login**.



Login

Authorization type: SQL server authentication

* Login: ServerAdmin

* Password: ••••••••

2. Select SQL server authentication, provide the required login information, and then click **OK** to log in.
3. After you are authenticated as **ServerAdmin**, type the following query in the query editor pane.

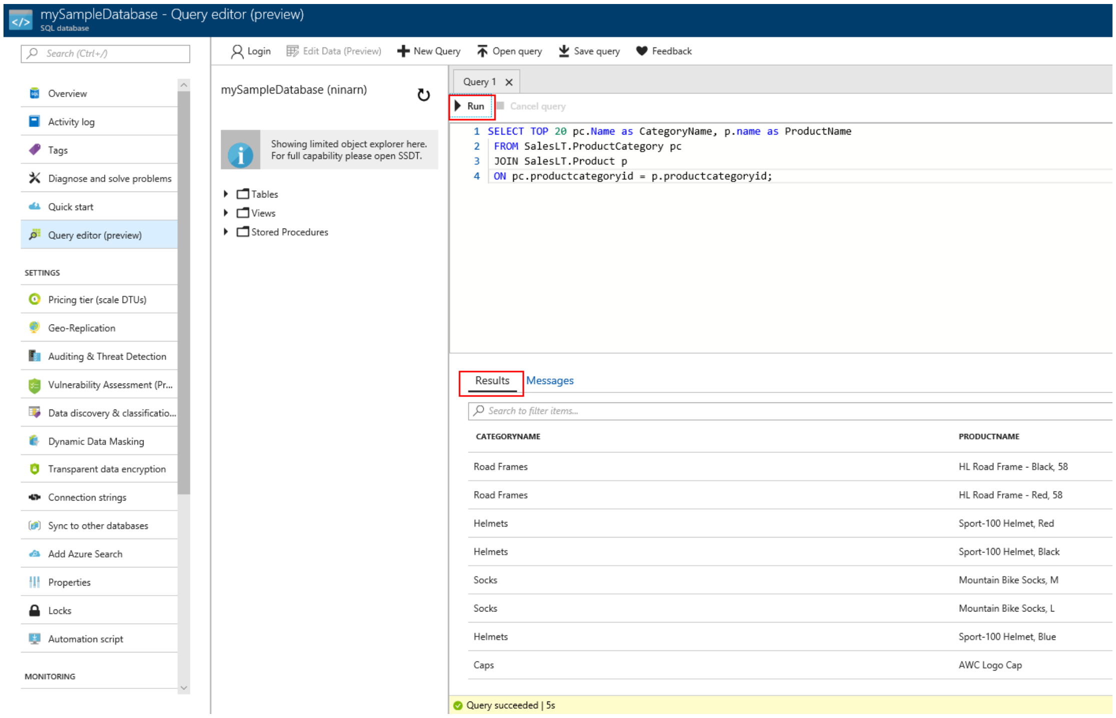
```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
```

```
FROM SalesLT.ProductCategory pc

JOIN SalesLT.Product p

ON pc.productcategoryid = p.productcategoryid;
```

4. Click **Run** and then review the query results in the **Results** pane.



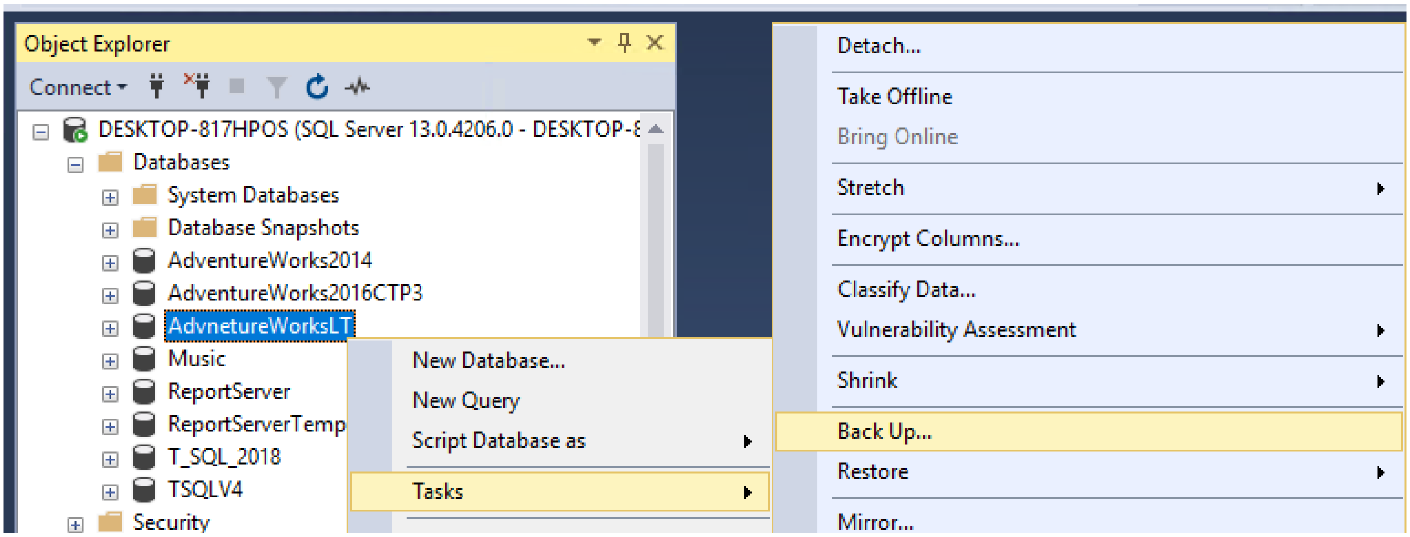
```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
 FROM SalesLT.ProductCategory pc
 JOIN SalesLT.Product p
 ON pc.productcategoryid = p.productcategoryid;
```

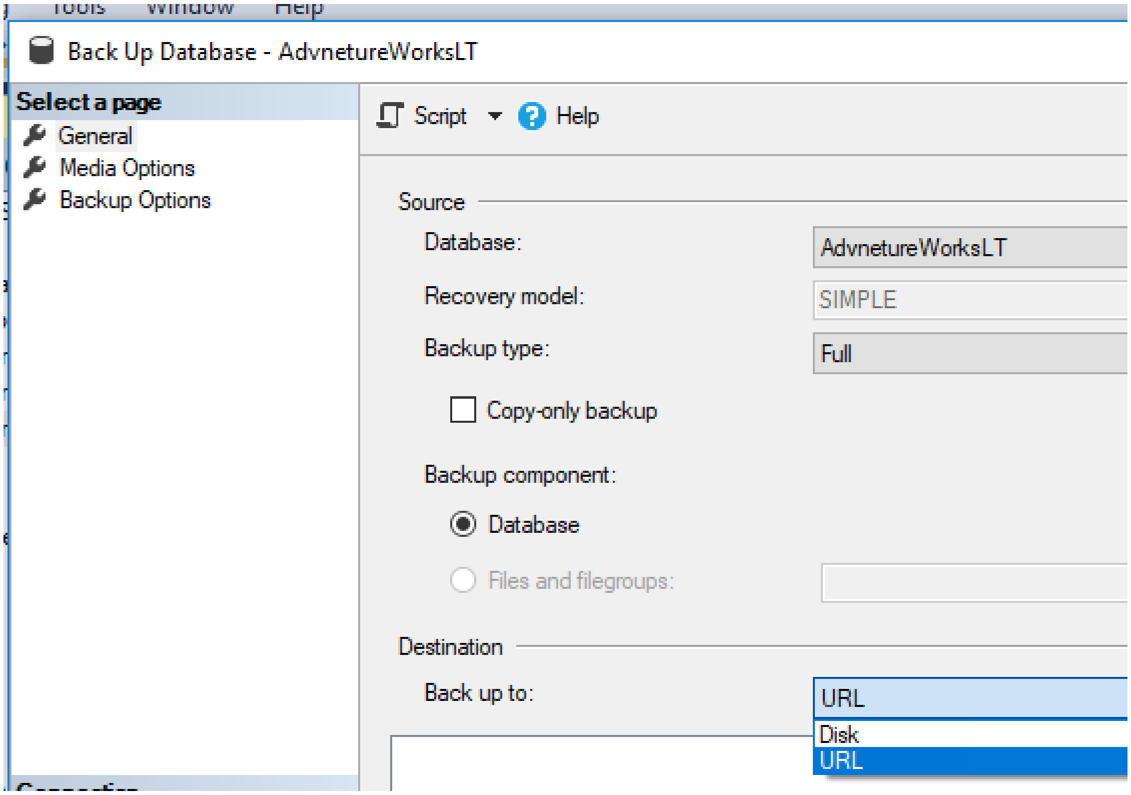
CATEGORYNAME	PRODUCTNAME
Road Frames	HL Road Frame - Black, 58
Road Frames	HL Road Frame - Red, 58
Helmets	Sport-100 Helmet, Red
Helmets	Sport-100 Helmet, Black
Socks	Mountain Bike Socks, M
Socks	Mountain Bike Socks, L
Helmets	Sport-100 Helmet, Blue
Caps	AWC Logo Cap

Query succeeded | 5s

5. Close the **Query editor** page, click **OK** to discard your unsaved edits.

Backup database to Azure Storage: URL Backup





Back Up Database - AdvnetureWorksLT

Source

Database: AdvnetureWorksLT

Recovery model: SIMPLE

Backup type: Full

☐ Copy-only backup

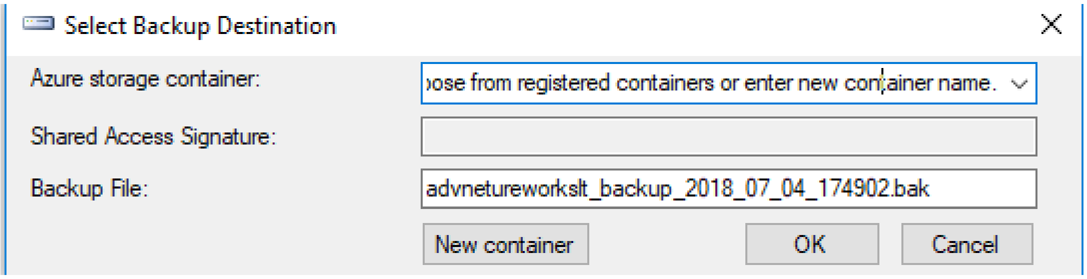
Backup component:

● Database

○ Files and filegroups:

Destination

Back up to: URL



Select Backup Destination

Azure storage container: ›ose from registered containers or enter new container name.

Shared Access Signature:

Backup File: advnetureworkslt_backup_2018_07_04_174902.bak

Save these resources if you want to go to Next steps and learn how to connect and query your database using a number of different methods. If, however, you wish to delete the resources that you created in this quickstart, use the following steps.

1. From the left-hand menu in the Azure portal, click **Resource groups** and then click **myResourceGroup**.
2. On your resource group page, click **Delete**, type **myResourceGroup** in the text box, and then click **Delete**.

Next steps

- Now that you have a database, you can connect and query using one of your favorite tools or languages.
- To learn how to design your first database, create tables, and insert data, see one of these tutorials:
  - Design your first Azure SQL database using SSMS
  - Design an Azure SQL database and connect with C# and ADO.NET

Containers & Azure

1. Web App for Containers:
   a. https://azure.microsoft.com/en-us/services/app-service/containers/
2. Container Service Documentation:
   a. https://docs.microsoft.com/en-us/azure/container-service/
3. Kubernetes (commonly stylized as **K8s**)[3], pronounced "Kū-bər-NəT-ēz –
   a. Is an open-sourcecontainer-orchestration system for automating deployment, scaling and management of containerized applications.
   b. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation.
   c. The goal is to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts.
   d. It works with a range of container tools, including Docker.
   e. https://en.wikipedia.org/wiki/Kubernetes
4. Docker
   a. A computer program that performs operating-system-level virtualization, also known as "containerization".
   b. First released in 2013 and is developed by Docker, Inc.
   c. Is used to run software packages called "containers".
   d. In a typical example use case, one container runs a web server and web application, while a second container runs a database server that is used by the web application.
   e. Containers are isolated from each other and bundle their own tools, libraries and configuration files.
   f. They can communicate with each other through well-defined channels.
   g. https://en.wikipedia.org/wiki/Docker_(software)
   h. Docker Engine 1.12 introduces swarm mode that enables you to create a cluster of one or more Docker Engines called a swarm. A swarm consists of one or more nodes: physical or virtual machines running Docker Engine 1.12 or later in swarm mode.
5. Get Started, Part 1: Orientation and setup – from Docker not Microsoft:

   a. https://docs.docker.com/get-started/
6. Tutorial: Create your first container in Azure Container Instances in the Portal:
   a. https://docs.microsoft.com/en-us/azure/container-instances/container-instances-quickstart-portal
7. Tutorial: Create container for deployment to Azure Container Instances:
   a. https://docs.microsoft.com/en-us/azure/container-instances/container-instances-tutorial-prepare-app
8. Deploy to an Azure Web App for Containers:
   a. https://docs.microsoft.com/en-us/vsts/pipelines/apps/cd/deploy-docker-webapp?view=vsts
9. Create your first Docker container with an ASP.NET web app:
   a. Now that you have created a container image, you can publish it to Azure App Service Linux, also known as Web App for Containers.
   b. This allows you to run your container inside a fully-managed platform provided by Azure.
   c. https://tutorials.visualstudio.com/aspnet-container/publish
10. space

Resources

1. To get started with Azure Storage Services:
   a. https://4sysops.com/archives/azure-storage-services-storage-account/
2. Azure SQL Database and SQL Data Warehouse firewall rules:
   a. https://docs.microsoft.com/en-us/azure/sql-database/sql-database-firewall-configure
3. Create an Azure SQL database in the Azure portal:
   a. https://docs.microsoft.com/en-us/azure/sql-database/sql-database-get-started-portal
4. Azure SQL Database: Use SQL Server Management Studio to connect and query data:
   a. https://docs.microsoft.com/en-us/azure/sql-database/sql-database-connect-query-ssms
5. Restoring From Backups Stored in Microsoft Azure:
   a. https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restoring-from-backups-stored-in-microsoft-azure?view=sql-server-2017
6. List Containers:
   a. https://docs.microsoft.com/en-us/rest/api/storageservices/list-containers2
7. Get-AzureStorageContainer:
   a. https://docs.microsoft.com/en-us/powershell/module/azure.storage/get-azurestoragecontainer?view=azurermps-6.4.0
8. Upload, download, and list blobs using the Azure portal:
   a. https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-portal
9. Deploying the Apps to ACI (Azure Container Instances): (see Also Docker Containers):
   a. https://github.com/dotnet-architecture/eShopModernizing/wiki/05.-Deploying-the-Apps-to-ACI-(Azure-Container-Instances)
10. Azure Container Instances and container orchestrators:
    a. https://docs.microsoft.com/en-us/azure/container-instances/container-instances-orchestrator-relationship
11. Azure Storage security guide:
    a. https://docs.microsoft.com/en-us/azure/storage/common/storage-security-guide
12. Continuous Integration, Delivery, and Deployment:
    a. https://www.digitalocean.com/community/tutorials/an-introduction-to-continuous-integration-delivery-and-deployment

Knock U Out & Slither…

1. KnockoutJs or Knockout
   a. Model–view–viewmodel (MVVM)
   b. MVVM facilitates a separation of development of the graphical user interface – be it via a markup language or GUI code – from development of the business logic or back-end logic (the data model).
   c. The view model of MVVM is a value converter and is responsible for exposing (converting) the data objects from the model in such a way that objects are easily managed and presented.
   d. https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel
   e. https://knockoutjs.com/index.html
2. Python
   a. An interpreted high-level programming language for general-purpose programming.
   b. Created by Guido van Rossum and first released in 1991.
   c. Python has a design philosophy that emphasizes code readability.
   d. Notably using significant whitespace.
   e. The off-side rule if blocks in that language are expressed by their indentation.
   f. The term was coined by Peter J. Landin, after the offside rule in football.
   g. This is contrasted with free-form languages, notably curly-bracket programming languages, where indentation is not meaningful and indent style is only a matter of convention and code formatting.
   h. https://en.wikipedia.org/wiki/Python_(programming_language)
   i. https://www.python.org/
3.