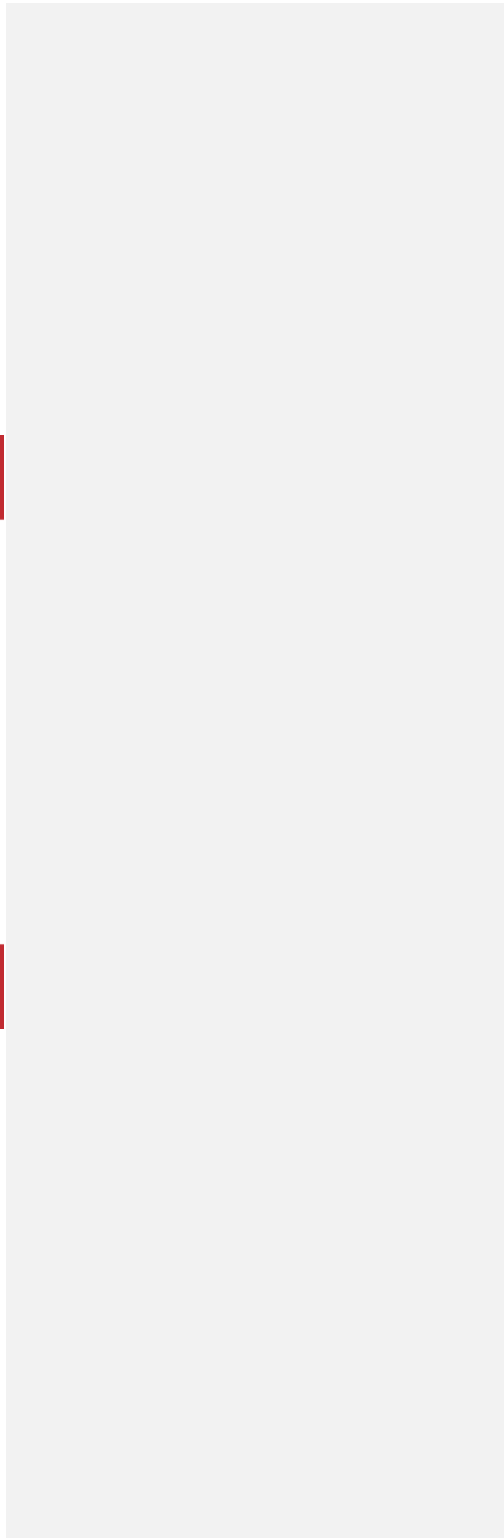
Four solid red squares are arranged in a 2x2 grid, framing the central text.

Exploring Manual and Automatic Database Backup Using Microsoft Azure Storage



Contents

Predictable, efficient and flexible data backups – certainty of availability.....	3
Provisioning a Windows Azure Storage Account	4
Using Microsoft SQL Server Backup to Azure tool....	8
Backup to Azure storage using T-SQL.....	19
Backing up your database to a URL	21
Managed backup process to Windows Azure	25
Azure account clean-up steps	28
Terms of use.....	29

Predictable,
efficient and
flexible data
backups –
certainty of
availability

Estimated time to complete lab is 60 minutes

Being able to configure and run backups in an efficient manner, knowing that the data will then always be available to you.

The need for business to feel confident in their backup processes is core to any organization. Knowing they have security and ready access to these point-in-time snapshots to core business environments and data is essential when planning for disaster recovery and business continuity situations. Traditionally, effective backup procedures have been costly to implement and have required a detailed amount of planning, scheduling and resource to maintain. With the advent of cloud services there have been vast improvements in the ways organizations can back-up their core business data, safe in the knowledge that it is highly available on highly-redundant systems. The Microsoft Windows Azure platform and the enhancements made to SQL Server 2014 provide an easy, cost effective and low-maintenance way to take advantage of these cloud features.

Connect to SQLONE computer

1. Click on **SQLONE** button on right side of the screen to connect to the **SQLONE** computer. If you see the following in the lower right corner of the screen, you can jump to step 5 below to set your screen resolution.



2. Click **Send Ctrl-Alt-Del** for **SQLONE** computer and then click **Switch user**.
3. Click **Send Ctrl-Alt-Del** for **SQLONE** computer again and then click **Other user**.

Provisioning a Windows Azure Storage Account

4. Log on to **SQLONE** computer as **labuser** with password **pass@word1**


Note, if you have a monitor that supports a larger screen resolution than 1024 x 768, you can change the screen resolution for the lab to go as high as 1920 x 1080. By going to a higher screen resolution, it will be easier to use SQL Server Management Studio.

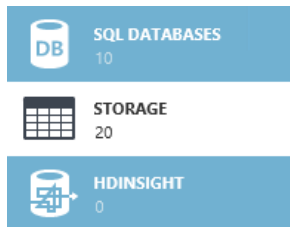
5. Right click on the desktop and click on **Screen resolution**.
6. Select **1366 x 786** (a good minimum screen size for using SSMS) and click **OK**.
7. Click **Keep Changes**.
8. Resize the client **holLaunchPad Online** window for the lab to fit your screen resolution.

Create the storage account:

1. If you are not already in Azure Management Portal, open Internet Explorer from the start screen and browse to <https://manage.windowsazure.com/> then sign in using your Azure account.

Azure Storage is a fully-distributed data storage mechanism. Data is stored on independent nodes over multiple domains which significantly reduces the potential for data corruption and data loss. This provides confidence that your data stored in Azure will always be accessible. When combined with Geo-replication, where your information is also synchronized with a separate set of servers in a different part of the country, redundancy against natural disasters is also applied to your data store, minimizing risk to your business-as-usual operations.

1. Click on **STORAGE**  from the blue navigation pane on the left



2. At the bottom left of the screen, click + **New**
3. Select **Data Services, Storage** and click **Quick Create**

4. For URL, use the first 7 characters of your Microsoft ID used for accessing the Azure account followed by **sqldbexport**. For example, **hdidemosqldbexport**

The storage account name must be unique within Azure, so you need a way to make a meaningful unique name.

5. In the **Location/Affinity Group** field select **South Central US**
6. **If asked select <<Your subscription identifier>>** as the subscription
7. Choose the **Locally Redundant** option in the **Replication** field.

Commented [A1]: Not available selected West US instead

Geo-redundancy enables Azure storage to store the data in two geographic locations. This ensures higher levels of redundancy to negate the effects of a natural disaster causing an outage to the server farm and increases the users' confidence in the durability of their data. There are extra fees for this option. For the purposes of this walk-through, Geo-redundancy is not required.

8. Click on **Create Storage Account**
NOTE: this may take a couple of minutes to complete

The screenshot shows a 'QUICK CREATE' dialog box with a dark background. On the left, there is a lightning bolt icon and the text 'QUICK CREATE'. On the right, there are three input fields: 'URL' with the value 'hdidemosqldbexport', 'LOCATION/AFFINITY GROUP' with a dropdown menu showing 'South Central US', and 'REPLICATION' with a dropdown menu showing 'Locally Redundant'. At the bottom right, there is a button labeled 'CREATE STORAGE ACCOUNT' with a checkmark icon.

9. Once the job completes, your storage account will be ready to use

Retrieve storage account key used for connecting to Azure with Azure Storage Explorer and SSMS

1. Select your newly created storage account and click on **Manage Access Keys** at the base of the page


Manage Access Keys

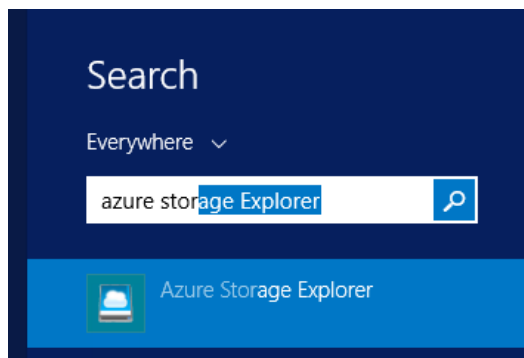
When you regenerate your storage access keys, you need to update any virtual machines, media services, or applications that access this storage account to use the new keys. [Learn more.](#)

STORAGE ACCOUNT NAME
hdidemosqldbexport

PRIMARY ACCESS KEY
qegDB3ipSBUA3yjgC8VRVkJQe6Z5SUC7F5f regenerate

SECONDARY ACCESS KEY
XDwIJOZgqmLbZGJwtfY1neloiZvyOXxuHEi regenerate

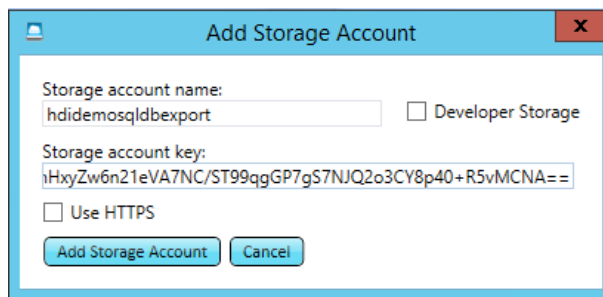
2. Click the **Copy** button  next to the **Primary Access Key**, **allow the webpage to access your clipboard if asked**, and click the **checkmark** button to close the **Manage Access Keys** window.
3. Open the **Azure Storage Explorer** from the Windows start screen



Azure Storage Explorer is one of multiple free tools approved by Microsoft to assist with the easy loading of data from your network or

other sources into your Azure Storage space. These tools allow you to easily create, modify and delete storage containers, blobs and the data within them from the desktop without having to log directly into your Azure portal.

4. On the welcome screen click **Continue**
5. Click the **Add Account** button
6. In the **Storage Account Name** field type the name of the storage account you entered in step 6 (this is visible in the **Manage Access Keys** popup.)
7. In the **Storage Account Key** field, paste the account key you copied to your clipboard by using the 'CTRL+V' command and click **Add Storage Account** button.



The screenshot shows a dialog box titled "Add Storage Account". It has a standard Windows-style title bar with a close button (X) in the top right corner. The dialog contains the following elements:

- A text input field labeled "Storage account name:" containing the text "hdidemosqldbexport".
- A checkbox labeled "Developer Storage" which is currently unchecked.
- A text input field labeled "Storage account key:" containing a long alphanumeric string: "iHxyZw6n21eVA7NC/ST99qgGP7gS7NJQ2o3CY8p40+R5vMCNA==".
- A checkbox labeled "Use HTTPS" which is currently unchecked.
- Two buttons at the bottom: "Add Storage Account" and "Cancel".

*If the copy and paste does not work, you may need to return to the **Azure Management** web page and manually select the contents of the **Primary Access Key** field, right click it and select **Copy**, and return to the **Add Storage Account** screen, right click the **Storage Account Key** field and select **Paste***

8. A message box explaining that the process for the first time creation of an account will appear. Click **OK**
9. Leave the **Azure Storage Explorer** window open.

Using Microsoft SQL Server Backup to Azure tool

In conjunction with in-the-box Cloud Backup in SQL Server 2014, Microsoft SQL Server Backup to Windows Azure Tool that enables a single cloud backup strategy across all versions of SQL Server. This reduces your CAPEX and OPEX by shifting moving from on premises storage or regionally hosted off site back-ups to secure and cost-effective storage in the Windows Azure Blob Storage Service. The advantages of this platform are that any information stored can be easily geo-replicated, is inherently highly redundant and highly available. This will allow you to ensure that any databases from previous versions of SQL Server that were not natively able to be backup up to the cloud can now be moved easily.

You are now going to test the various methods on a database from SQL Server 2012 – **2012Database** hosted on SQLONE.

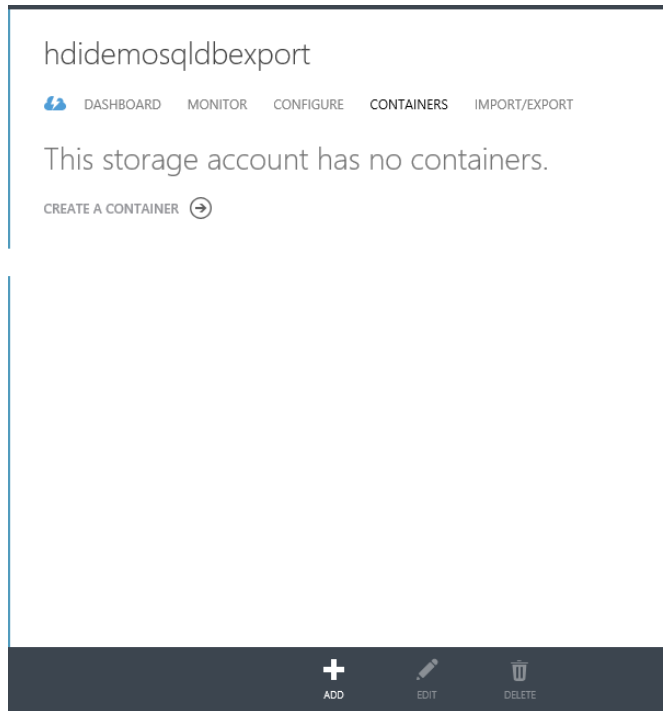
Backing up using Microsoft SQL Server Backup to Microsoft Windows Azure Tool

NOTE: There are a number of options in this scenario. All options are independent and as many or few as you want can be completed – the end result of all is the database ends up on Azure in a restorable form.

Microsoft SQL Server Backup to Windows Azure Tool enables a single cloud backup strategy across all versions of SQL Server. This tool also supports Encryption and Compress of the backups. You could download the tool from <http://www.microsoft.com/en-us/download/details.aspx?id=40740> and install it on your computer.

Create a container in the Storage Account:

1. Go to your Azure Portal in the browser
2. Click on **STORAGE** in the blue navigation pane
3. Select the storage account you created
4. Click **CONTAINERS** to navigate to the CONTAINERS panel
5. Click the **CREATE A CONTAINER** link or the **ADD** button at the bottom of the page.



6. Type **sql-backups** for the **NAME** of the container and leave **ACCESS** set to Private.

New container

NAME

sql-backups

ACCESS ?

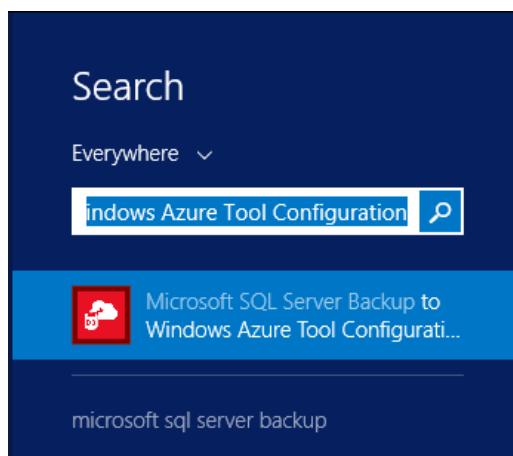
Private

Private access means only users with credentials can read or write to this container. Even if your container is not for secure information, preventing undesired uploads and downloads will save money.

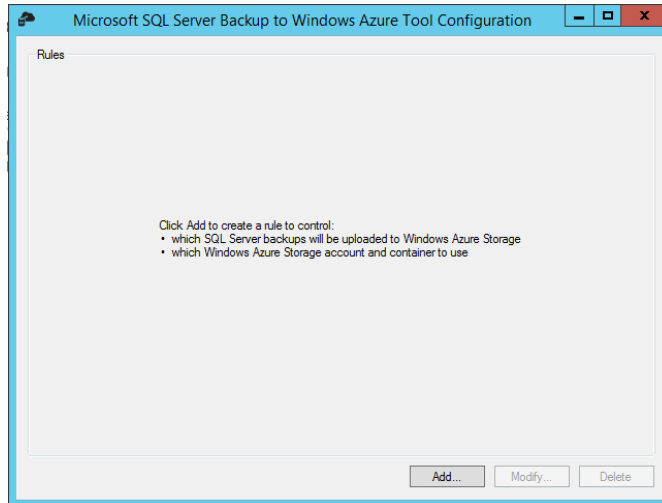
7. Click on the check mark to create the container

Creating a backup rule

1. Open **Microsoft SQL Server Backup to Microsoft Windows Azure Tool Configuration** from the **Start Screen** (under **More Apps/Microsoft SQL Server Back...** section. Note, this is not the same program as **Windows Azure Backup**.)



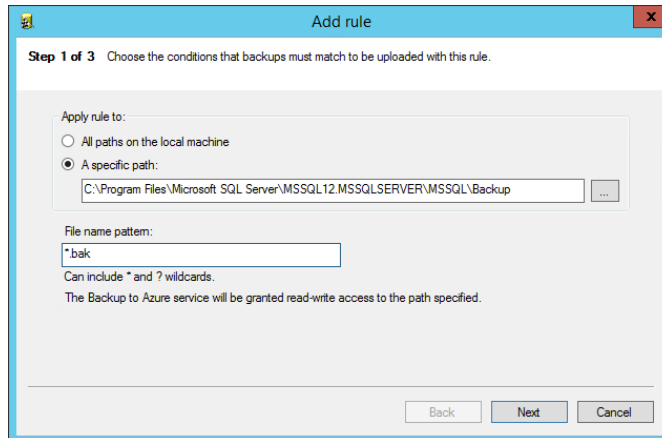
2. Click **Yes** to allow the program to make changes to the computer



3. Clicking on **Add...** opens the **Add Rule** wizard

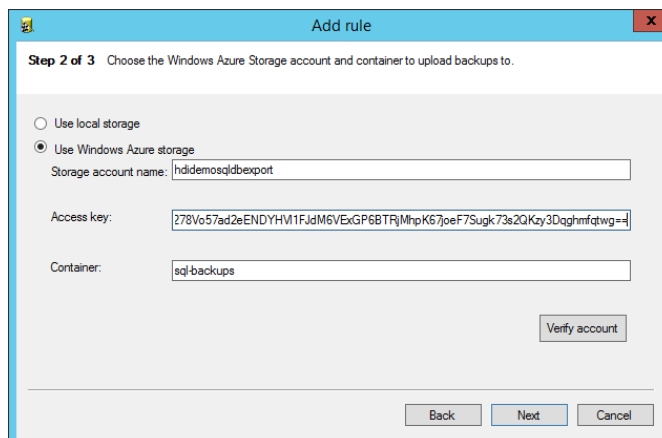
On the Choose the conditions that backups must match to be uploaded with this rule page:

4. In the **Apply rule to** section, select **A specific path** option and click on ellipses (...) to browse and select **C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup**
5. Enter ***.bak** as the **File name pattern** and click on **Next**

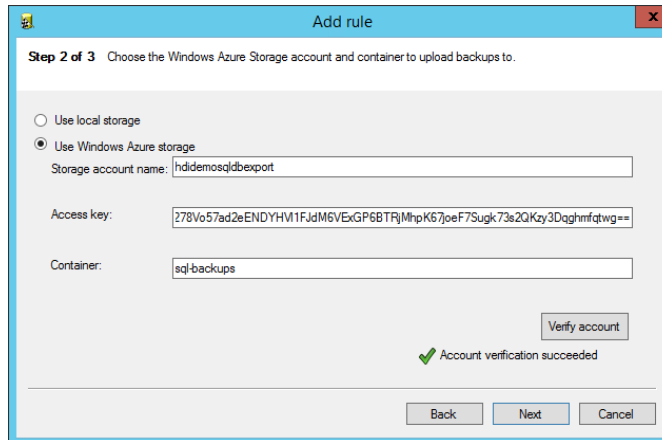


On the Choose the Windows Azure Storage account and container to upload backups to page:

6. Select **Use Windows Azure store** option and provide your Storage account name, Access key and the name of the container **sql-backups** to be used for placing the backup (get the access key by clicking on **Manage Access Keys** on the storage account page for your storage account in the Azure Management portal)



7. Click on **Verify account** to test the availability of the storage account and container

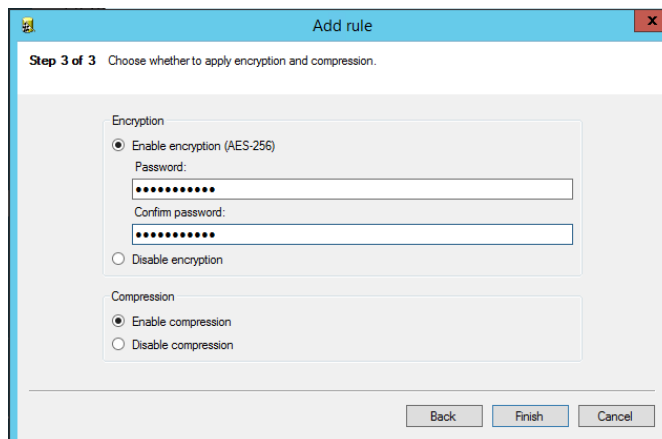


8. Once the Account Verification succeeds, click on **Next**

On the Choose whether to apply encryption and compression page:

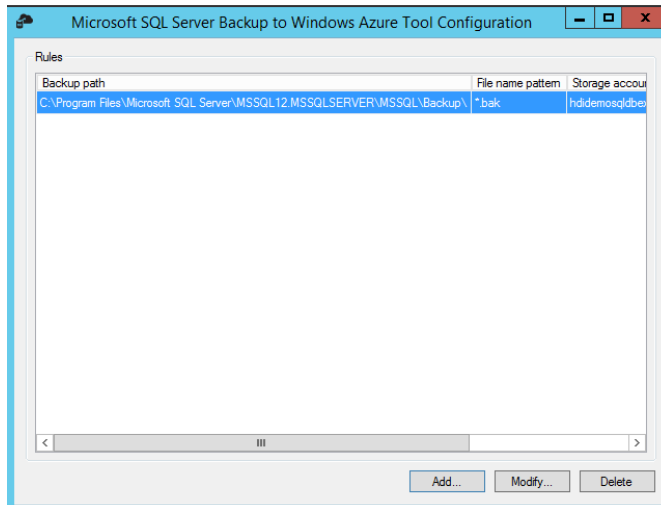
9. Choose **Enable encryption (AES-256)** option to enable encryption of the backups and enter an encryption password as **Pass@word12**

10. Choose **Enable compression** option to compress the backup



11. Click on **Finish** to exit the **Add Rule** wizard

12. New rule added is listed in the Configuration window



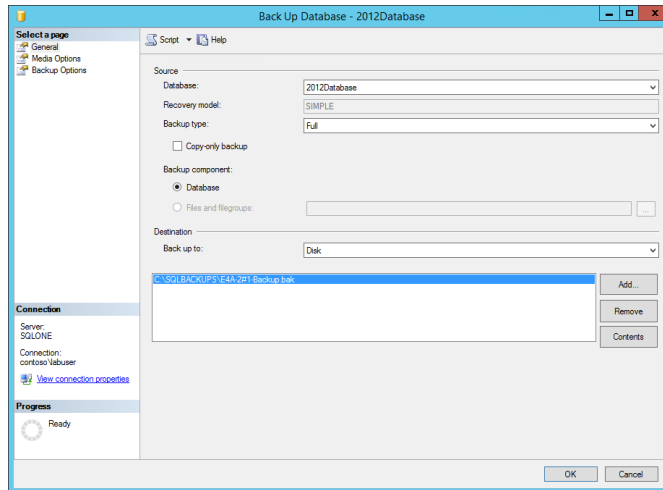
*NOTE: To add more rules, click on **Add...** button again. You could select any rule and click on **Modify** button to modify any of the options defined in the earlier steps. Select any rule and click on **Delete** to remove the rule.*

13. Close **Microsoft SQL Server Backup to Microsoft Windows Azure Tool Configuration** window

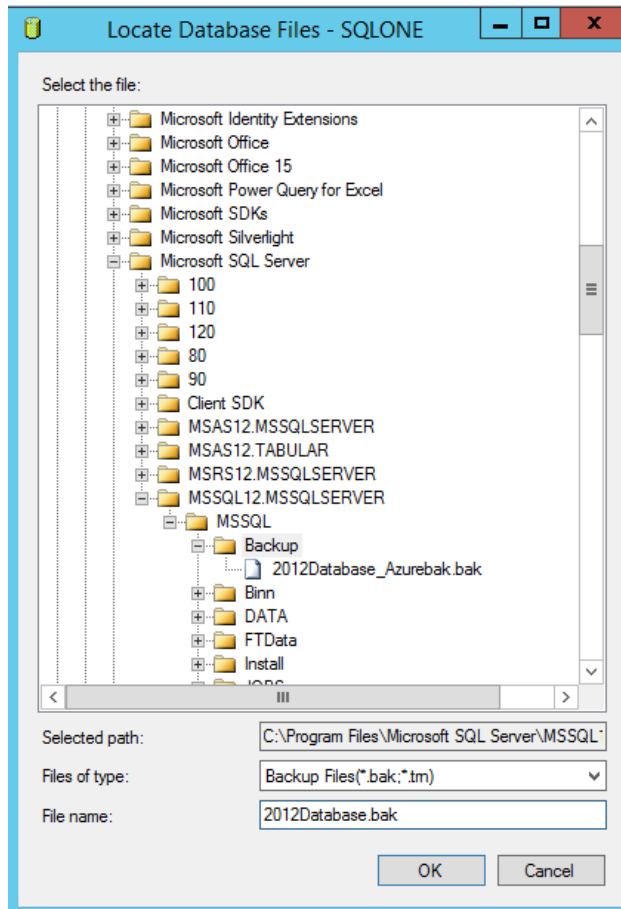
Create a backup to kick off the rule

Create a backup of **2012Database** called **2012Database.bak** in the location specified in Step 1 above. To create a backup:

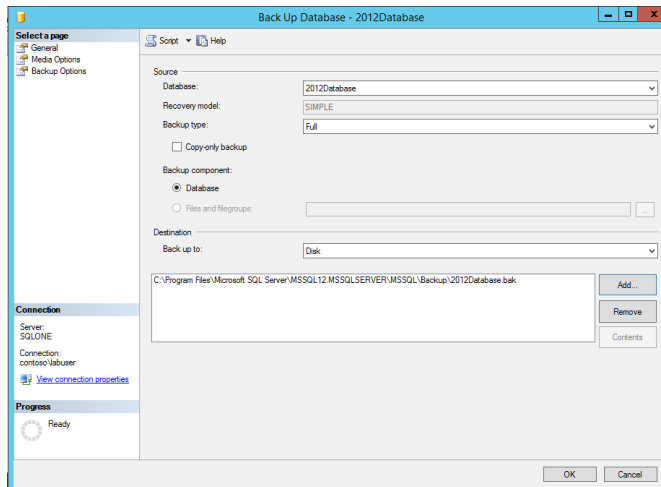
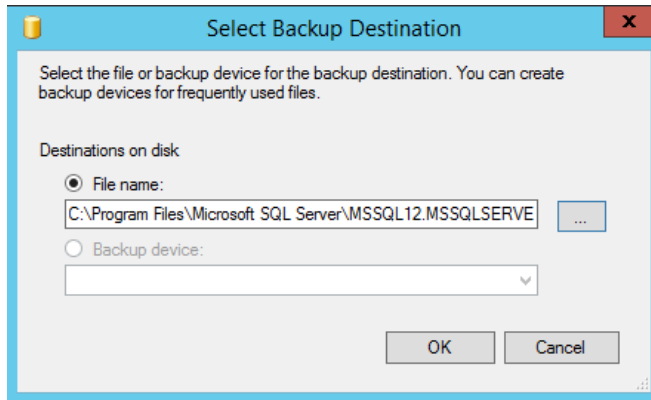
1. Open **SQL Server Management Studio**
2. Connect to **SQLONE** Database Engine server using Windows Authentication
3. Expand the **Databases** node in **Object Explorer** then right-click on the **2012Database** and select **Tasks > Back Up...**
4. In this case, the default backup destination will not trigger the rule, so we need to replace it by clicking **Remove** then **Add...**,



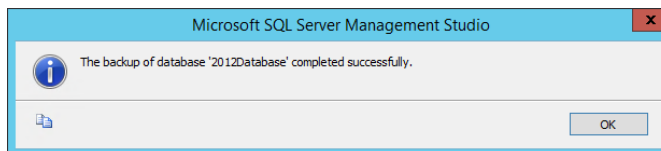
5. Click on the ellipsis (...) and navigate to **C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup** then enter **2012Database.bak** in the File name and



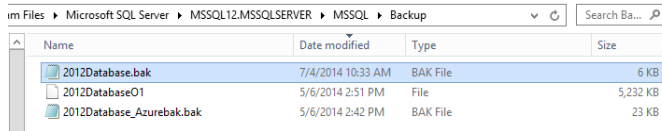
6. Click **OK** three times



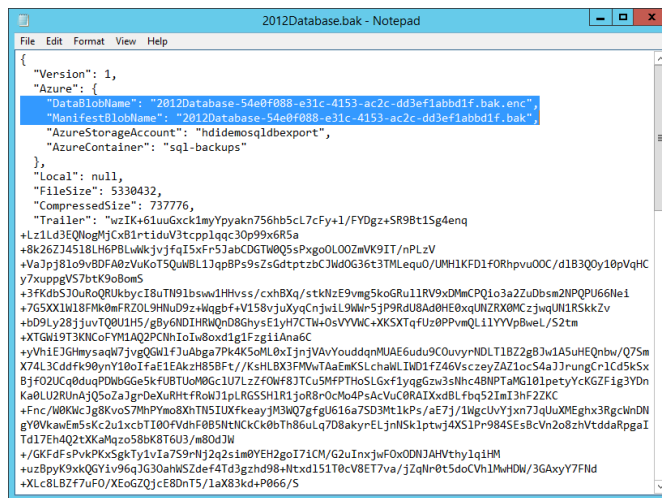
7. Click **OK** in the success dialog.



8. Open File Explorer, browse to **C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\Backup**



9. Right-click on the new backup file, and open with Notepad (you may have to choose Notepad from among the apps on the machine.)
10. Note the **DataBlobName** and the **ManifestBlobName**.



*NOTE: this file does not contain the backup data, it is instead the metadata pointing to the backup files in the cloud. However, restoring the database still only requires pointing at this .bak file and **Microsoft SQL Server Backup to Microsoft Windows Azure Tool** to be installed.*

11. Go to the Windows Azure Management portal at <https://manage.windowsazure.com/> using Internet Explorer and navigate to STORAGE from the left hand side menu
12. Click on the storage account selected as part of Step 2 above
13. Click on **CONTAINERS** from the top menu
14. Select the container **sql-backups**

Backup to Azure storage using T-SQL

NAME	URL	LAST MODIFIED	SIZE
2012Database-54a0f068-e31c-4153-ac2c-0d5ef1atbd1f0ak	https://hdidemossqlexport.blob.core.	7/4/2014 10:33:39 AM	5.46 KB
2012Database-54a0f068-e31c-4153-ac2c-0d5ef1atbd1f0akenc	https://hdidemossqlexport.blob.core.	7/4/2014 10:33:39 AM	720.5 KB

Note that there are 2 backup files created – the **DataBlobName** and the **ManifestBlobName** from the local .bak file.

Following the simple 3 steps in the **Microsoft SQL Server Backup to Microsoft Windows Azure Tool**, you are now able to back up your database (from previous versions of SQL Server) to Windows Azure storage and utilize the advantages of having it on cloud.

You would use this option if you wanted to use reproducible SQL scripts for backing up databases to Azure.

1. Open **SQL Server Management Studio** and connect to the **SQLONE** database engine server using Windows Authentication
2. Go to **File** menu and click on **Open** and select **File...** option
3. In the File open dialog box, browse to **C:\SQLSCRIPTS\E4** location, select **E4C-1#1-Create Credentials.sql** script file and click on **Open**.

```
E4C-1#1-Create Cre...ntoso\labuser (57) X
1  |-- Create Azure Credentials
2  CREATE CREDENTIAL [<INSERT YOUR CREDENTIAL NAME>]
3     WITH IDENTITY = '<INSERT YOUR STORAGE NAME>'
4     ,SECRET = '<INSERT YOUR STORAGE ACCESS KEY>'
5  GO
```

4. To get the storage name and storage access account key, go to Azure Management Portal, click on **STORAGE** in the navigation bar, select the storage account you created earlier and click on **Manage Access Keys** in the grey options pane at the window bottom. Copy the **STORAGE ACCOUNT NAME** by clicking the copy icon next to it then use this to replace **<INSERT YOUR STORAGE NAME>** in the Management Studio query box. Likewise, copy the **PRIMARY ACCESS KEY** from Azure Manage Access Keys and use this as to replace **<INSERT YOUR STORAGE ACCESS KEY>** in the Management Studio query. Replace **<INSERT YOUR CREDENTIAL NAME>** with **E4C1_Credential**

```
E4C-1#1-Create Cre...ntoso\labuser (57))* X
1  -- Create Azure Credentials
2  CREATE CREDENTIAL [E4C1_Credential]
3     WITH IDENTITY = 'hdidemosqldbexport'
4     ,SECRET = 'QfeRM9cMX0z7X4bz7X278Vo57ad2eENDYHh'
5  GO
```

5. In Management Studio, click **Execute** to run the query.
6. Go to **File** menu and click **Open** and select **File...** option.
7. In the File open dialog box, browse to **C:\SQLSCRIPTS\E4** location, select **E4C-1#2-Create Database Backup.sql** script file and click on **Open**
8. Edit the script to change the BACKUP DATABASE name to **[2012Database]**.
9. Edit the script to include details related to your database and Azure Storage.
 - a. **<Insert your storage>** is the name of the storage account (must be the same storage account as used when creating the credential in the previous query),
 - b. the **<Insert your storage blob>** becomes the name of a blob (container) **sql-backus** in the storage account that you wish to store the backup in,
 - c. your Azure database name is what you want the database file to be called on Azure (e.g., **2012Database** unless you prefer to call it something else), and
 - d. your credential is the name of the credential you just created **E4C1_Credential**

```
E4C-1#2-Create Dat...ntoso\labuser (58))* X E4C-1#1-Create Cre...ntoso\labuser (57))*
1  -- Create Database Backup Script
2  BACKUP DATABASE [2012Database]
3     TO URL = 'http://hdidemosqldbexport.blob.core.windows.net/sql-backups/2012Database.bak'
4     WITH CREDENTIAL = 'E4C1_Credential',
5     STATS=5;
6  GO
7
```

10. Click on **Execute** to run the query.

NOTE: this script will take 1-5 minutes to run, depending on your Internet connection and geographic location (if you are close to the storage account location – North Europe in this case – the script will run faster.) You can continue with other scenarios while this is finishing.

NOTE: Windows Azure backup only supports backups of up to 1TB. Also, when using the script with SQL Server 2012, make sure you are running it on SQL Server 2012 SP1 CU2 (11.0.3339.0).

Verify the backup on Azure

1. Go to your Azure Management Portal.
2. Click STORAGE
3. Click on your storage account name
4. Click CONTAINERS
5. Click on the container name **sql-backups**



NAME	URL	LAST MODIFIED	SIZE
sql-backups			
2012Database-54e0f088-e31c-4153-ac2c-d53e11abb0f1.bak	https://hdidemossqlobesportblob.com/	7/4/2014 10:33:39 AM	5.46 KB
2012Database-54e0f088-e31c-4153-ac2c-d53e11abb0f1bak.enc	https://hdidemossqlobesportblob.com/	7/4/2014 10:33:39 AM	720.5 KB
2012Database.bak	https://hdidemossqlobesportblob.com/	7/4/2014 10:28:01 AM	5.69 MB

You are now able to ensure that previous supported versions of SQL Server that you use (SQL Server 2012) which didn't have connectivity features allowing them to directly backup to Azure Storage now benefit from have secure back-ups hosted in Windows Azure allowing them to benefit from high levels of redundancy and availability. This decreases the level of risk to your organization in a disaster recovery situation and improves business continuity.

Backing up
your
database to
a URL

Richard wants to look for ways to simplify the process of providing a direct backup to a managed, highly available location with redundancy built in. In doing this he knows that his backups are always available, and saves the headaches with all the normal backup procedures that organizations can encounter when trying to provide secure and robust backups. Richard is aware that using SQL Server 2014 and Windows Azure, he can achieve this without any intermediate steps. It can either be achieved through a wizard or as a Transact SQL coded function. Backing up to his blob store means Richard takes advantage of all the data redundancy features offered by the Windows Azure Storage service. You will now back up a database onto Azure.

Create Encryption certificate

To create an encrypted backup, Richard will also require an **Encryption certificate** to be created before trying to create the database backups.

1. Switch to **SQL Server Management Studio**

2. Go to **File** menu and click on **Open** and select **File...** option
3. In the File open dialog box, browse to **C:\SQLSCRIPTS\E4** location, select **E4C-2#2-Create Encryption Certificate.sql** script file and click on **Open**.
4. Edit the script by replacing sections in angle brackets <> with relevant values.

```

E4C-2#2-Create Enc...ntoso\labuser (53))' × E4C-1#2-Create Dat...ntoso\labuser (58))' E4C-1#1-Create Cre...
1 USE master;
2 GO
3 -- Step 02: Creates a database master key
4 IF NOT EXISTS (SELECT * FROM sys.symmetric_keys WHERE symmetric_key_id = 101)
5 | CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Pas@word12';
6 |
7 GO
8 -- Step 03: Create Encryption Certificate
9 | CREATE CERTIFICATE [EncryptionCertE42]
10 | WITH SUBJECT = 'EncryptionCertE42 certificate';
11 |
12 GO

```

The password must meet the Windows Requirements to be a valid password, the certificate name is **EncryptionCertE42**, and the subject is a textual description for yourself.

5. Click on **Execute**

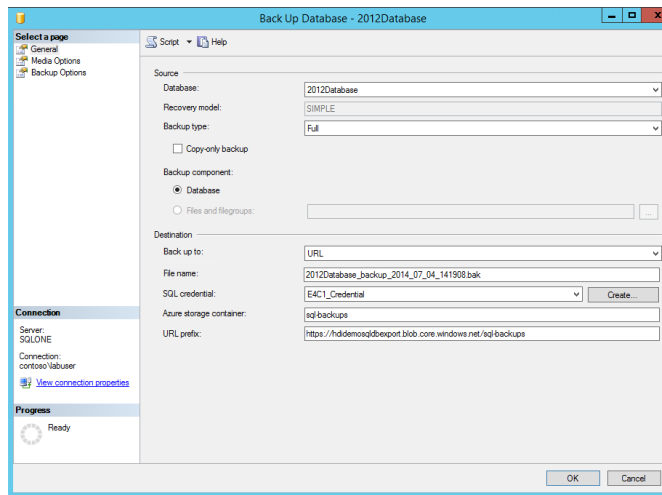
You now a Credential and an Encryption certificate to be used for creating your secure database backups. You could use wither the SSMS UI or a T-SQL query to create the backup. In this example, you will use the SSMS UI.

Creating a Backup using Management Studio Wizard

In the prior back up example where you backed up the database using T-SQL, the backup was not encrypted. In this example, you will perform an encrypted backup to Azure URL using the Encryption Certificate you just created.

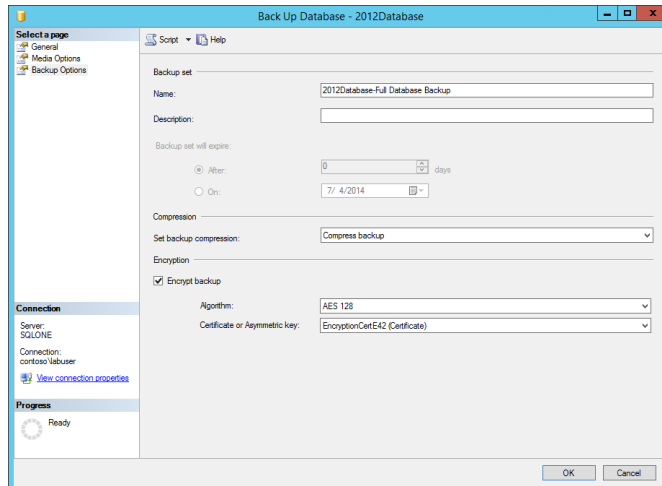
1. Switch to SSMS.
2. Right click on **2012Database** in Object Explorer, select **Tasks** and click on **Back Up...**
3. In the **Destination** section of the General page, Select **URL** from **Back up to** dropdown
4. Enter a name for the backup file (the default is fine.)
5. Select the credential you created **E4C1_Credential** by clicking the drop-down arrow on the **SQL credential** box

6. Enter the name of the **Azure storage container** to store the backup file in as **sql-backups**.
7. **URL prefix** automatically gets updated based on the SQL credentials selected and Azure storage container name provided above

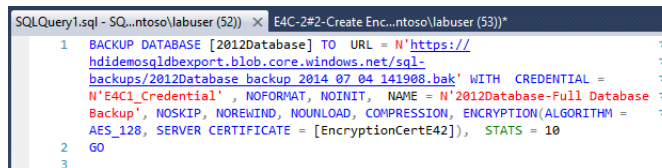


8. In the **Backup Options** tab, change **Set backup compression** to **Compress backup**.
9. Check **Encrypt backup** option
10. Select the **Encryption Algorithm** to be used (default is fine)
11. Also select the Encryption **Certificate or Asymmetric key** to be used as **EncryptionCertE42**.

NOTE: The Certificate or Asymmetric key dropdown will be pre-populated with the Encryption certificate created earlier in this scenario



12. To view the T-SQL that SSMS will use to perform the backup, click the Script button at the top of the dialog. Then, switch to the query editor window that SSMS created.



13. Execute the query.
14. Switch back to the backup dialog and close it.
15. If desired, you can now see the backup file in the Azure container through the Azure Management Portal in Internet Explorer.

You now can quickly and easily establish a backup process that can be automated and allows you to write you backup files directly to an offsite, highly available and inherently highly redundant service ensuring that the backup images will be secure and always available. You are able to do this directly from his management console without having to use any third party tools or intermediate steps. This method will save you time and effort should you face a Disaster Recovery situation or where you need to roll back to a previous version of the database.

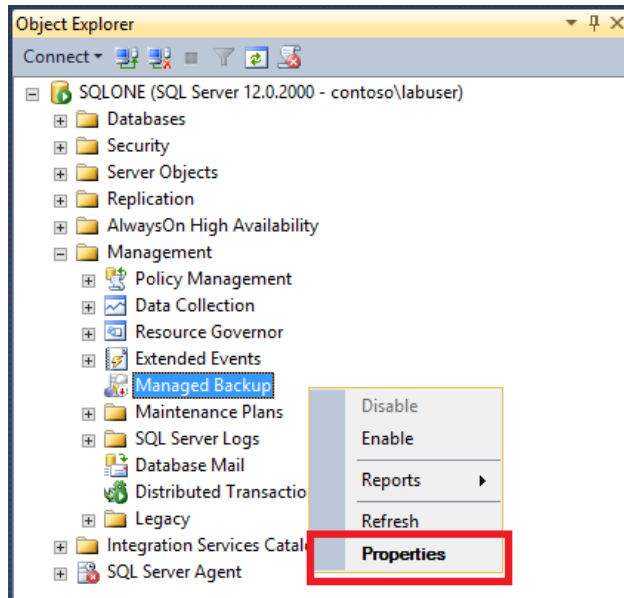
Managed backup process to Windows Azure

With everything he needs now in place, Richard is looking to increase the efficiency with which he can manage the new environment. He is aware that he can use SQL Server Managed Backup to Windows Azure to manage and automate SQL Server backups to the Azure Blob Storage Service. This allows Richard to easily control backup at either the database or instance level and lets him perform point in time restores of his databases. The advantage of pushing these to Windows Azure storage is the limitless amount of space available to him, and the centralized highly available location provided by the Storage services.

Using SQL Server Management Studio to manage backup for a SQL Server instance

1. Ensure **SQL Server 2014 Management Studio** is open
2. Ensure you have a connection open to the **SQLONE** database engine using Windows Authentication (the **Object Explorer** has a heading saying **SQLONE (SQL Server version – contoso\labuser)** where **version** is a string of numbers and decimal points.) If not, click **Connect** in the **Object Explorer** and select **Database Engine...** then give **SQLONE** as the Server name, **Windows Authentication** as the Authentication and click **Connect**.)
3. Go the **Object Explorer** and expand the **Management** node
4. Right click on **Managed Backup** and select **Configure** if present. Otherwise, click **Properties**.

Commented [A2]: Managed Backup section only shows how to set it up, but execution and usage of it not done.

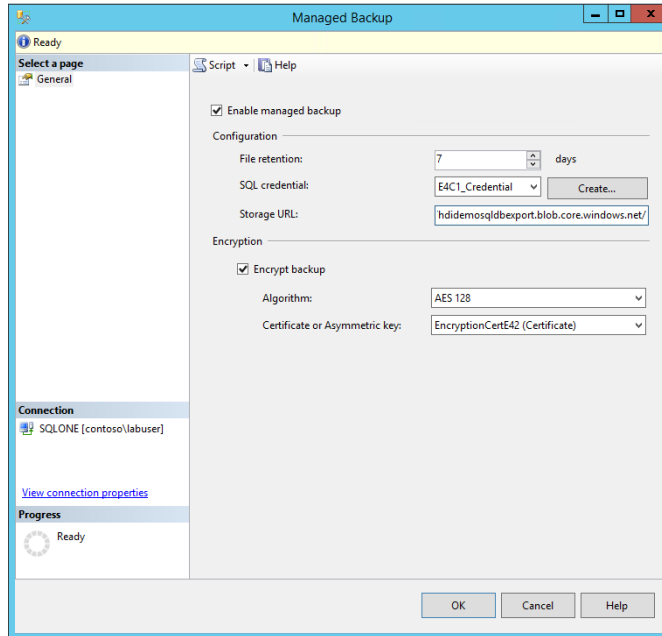


This opens the Managed Backup dialog

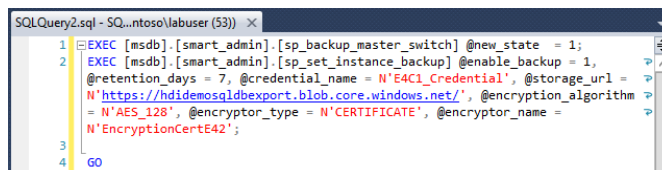
5. Check the **Enable managed backup** option
6. Enter a number between 1 and 30 in File retention box

NOTE: Enter 7, to retain the backup file for 7 days before removing it

7. For the **SQL credential**, select **E4C1_Credential** from the drop-down box.
8. **Storage URL** gets updated based on the **SQL Credential**
9. Select **Encrypt backup** and select **AES 128** as the algorithm and the certificate as the **Certificate or Asymmetric key** as **EncryptionCertE42 (Certificate)**



10. Click the **Script** button at the top of the dialog to create a new query window with the T-SQL needed to setup the **Managed Backup** operation.
11. Click on **Cancel**



12. Review the generated T-SQL in the editor window
13. Right click on **SQL Server Agent** in the **Object Explorer** and click **Start**. Click **Yes** to confirm.
14. Go back to the query window and then **Execute** the query.

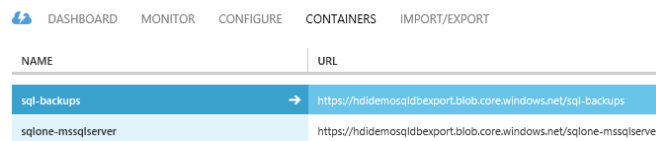
*NOTE: To disable or enable the managed backup, go to the object explorer, expand the **Management** node, and right click on **Managed Backup**. Select **Disable** or **Enable** option to disable or enable the managed backup respectively.*

Verify the new container in your storage account

Managed Backup creates a new container in your account based on the server and instance name.

1. Go to your Azure Management Portal.
2. Click **STORAGE**
3. Click on your storage account name.
4. Click **CONTAINERS**

hdidemosqldbexport



DASHBOARD MONITOR CONFIGURE CONTAINERS IMPORT/EXPORT

NAME	URL
sql-backups	https://hdidemosqldbexport.blob.core.windows.net/sql-backups
sqlone-mssqlserver	https://hdidemosqldbexport.blob.core.windows.net/sqlone-mssqlserver

*In this case, the new container is named **sqlone-mssqlserver**. If you click on the container, you will not see any blobs until the managed backup process gets triggered.*

When database level managed backups are set up, they override instance level managed backup settings. SQL Server Managed Backup to Windows Azure service can also be paused and resumed.

Follow the rollback steps given below based on the option(s) used in each scenario

1. Ensure you are logged in to the **SQLONE** virtual machine as **Contoso\labuser** using the password **pass@word1**
2. Open Windows Azure Management Portal by browsing to <https://manage.windowsazure.com/> using Internet Explorer and entering your Azure credentials
3. Click on **STORAGE** and select the container you used for creating backups into
4. Delete each of the backup files you created during this story in the container by selecting it then clicking **DELETE** in the grey options pane at the window bottom, and confirming your deletion when asked

Azure
account
clean-up
steps

Terms of use

© 2014 Microsoft Corporation. All rights reserved.

By using this Hands-on Lab, you agree to the following terms:

The technology/functionality described in this Hands-on Lab is provided by Microsoft Corporation in a “sandbox” testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the Hands-on Lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this Hands-on Lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION IS EXPRESSLY PROHIBITED.

THIS HANDS-ONLAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCTIONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

FEEDBACK. If you give feedback about the technology features, functionality and/or concepts described in this Hands-on Lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB , INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR

REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER

This lab contains only a portion of new features and enhancements in Microsoft SQL Server 2014. Some of the features might change in future releases of the product. In this lab, you will learn about some, but not all, new features.