**Lab 7 – Planning and Implementing a High Availability and Disaster Recovery Environment**

**Estimated Time**: 90 minutes

**Pre-requisites**: The Azure SQL Database used by Exercise 1 was created in the lab for Module 3.

**Lab files**: The files for this lab are located in the D:\Labfiles\High Availability folder.

**Lab overview**

The students will execute two main tasks: make Azure SQL Database geo-redundant, and backup to and restore from a URL which uses Azure.

**Lab objectives**

After completing this lab, you will be able to:

* + Enable geo-replication for Azure SQL Database
  + Backup and restore a SQL Server database using a URL

**Scenario**

Now that you have automated day-to-day tasks in the previous lab, as the Senior Data Engineer, you are tasked with improving the availability of both IaaS and PaaS configurations for your database environment. You are tasked with the following objectives:

* + Enable geo-replication for Azure SQL Database to increase availability for a database.
  + Back up a database to a URL in Azure and restore it after a human error has occurred.

**Exercise 1: Enable Geo-Replication for Azure SQL Database**

Estimated Time: 45 minutes

Overview

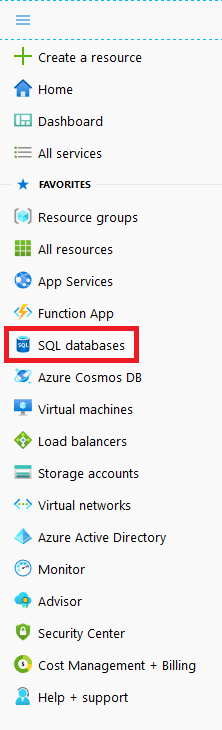
The Students will alter the configuration of the Azure SQL Database created during the lab for Module 3 to make it highly available.

Scenario

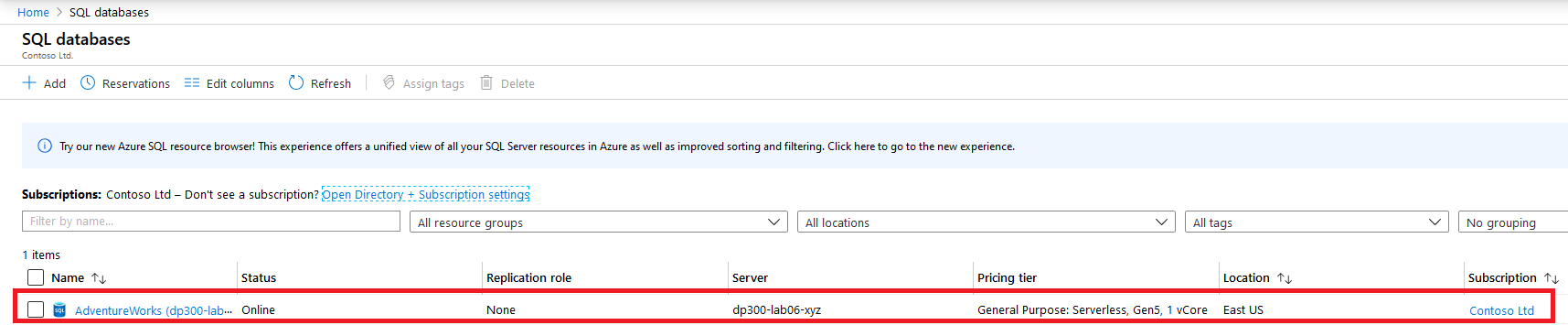
As a DBA within WideWorldImporters you need to know how to enable geo-replication for Azure SQL Database, ensure it is working, and know how to manually fail it over to another region using the portal.

Pre-requisites

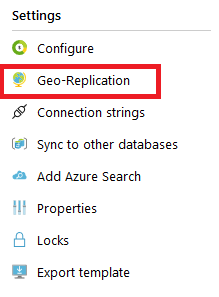
* + Azure account created for the student – must provide a login (e-mail) and password
  + Azure SQL Database server and database pre-created
  + If you are not logged into the Azure portal via a browser window, do so using the Azure credentials provided to you.
  + From the menu, select SQL databases as shown below.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-01.png)

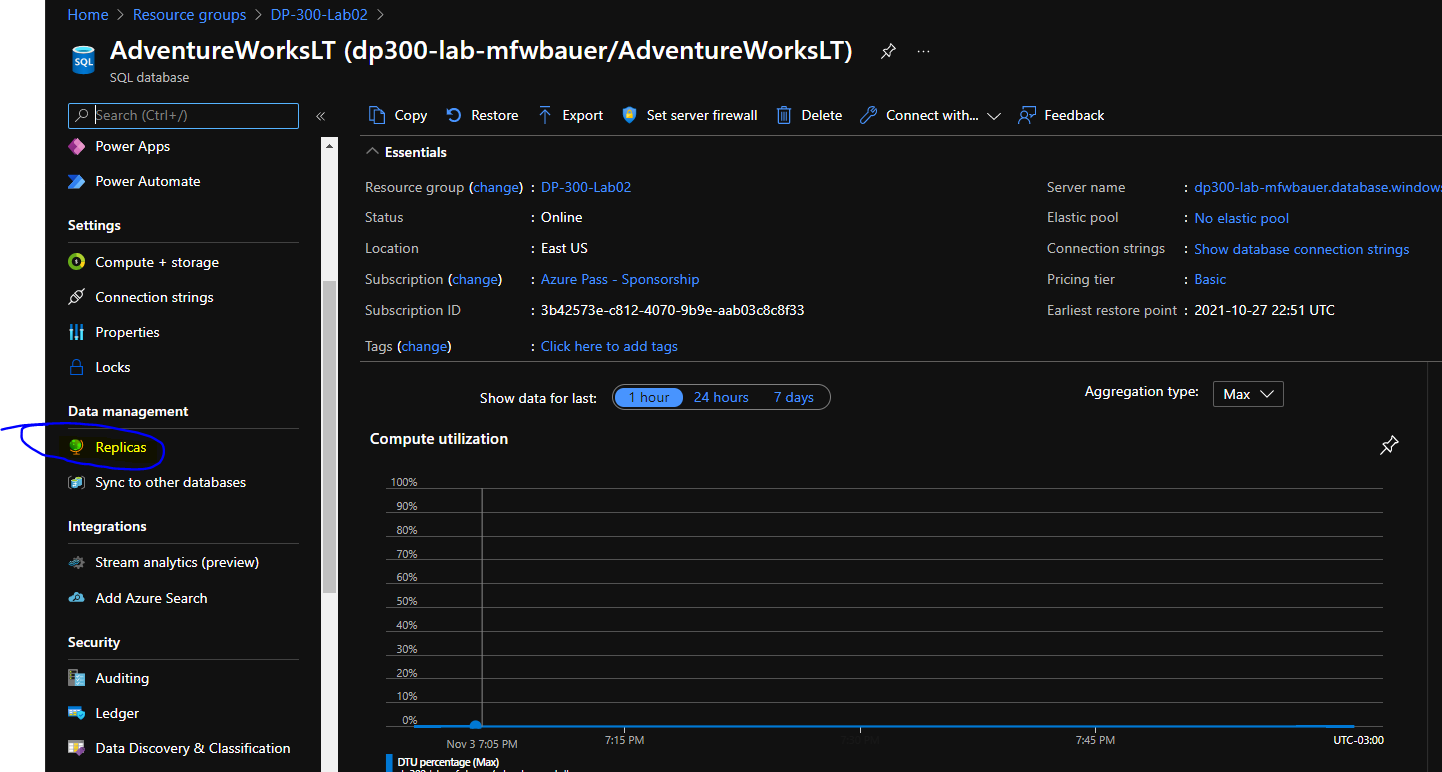
* + Click on the Azure SQL Database that was created in Lab 3. An example is shown below.

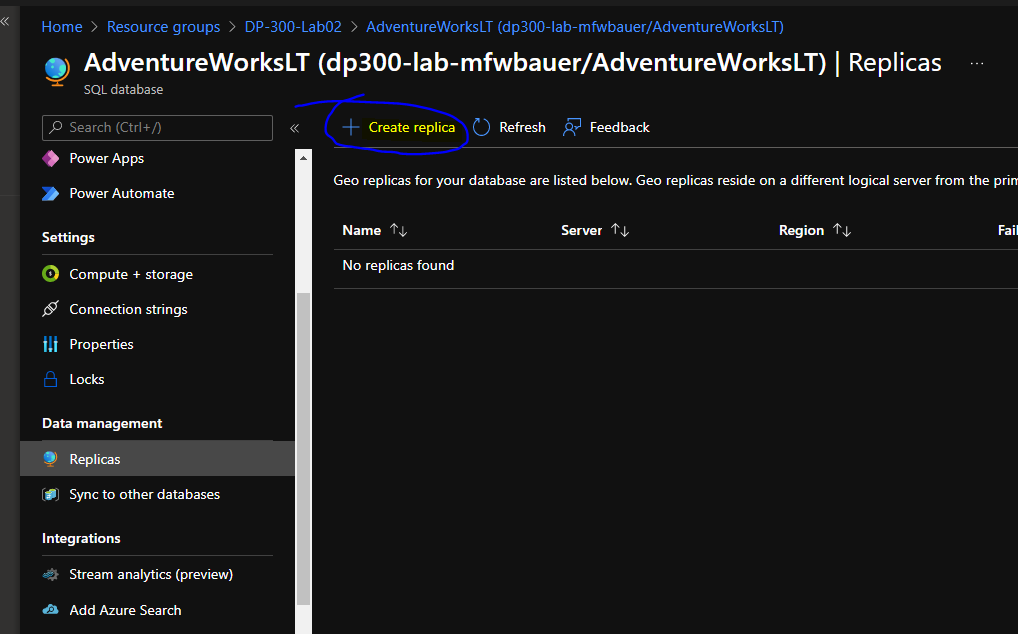
[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-02.png)

* 1. In the blade for the database, under Settings, select Geo-Replication.

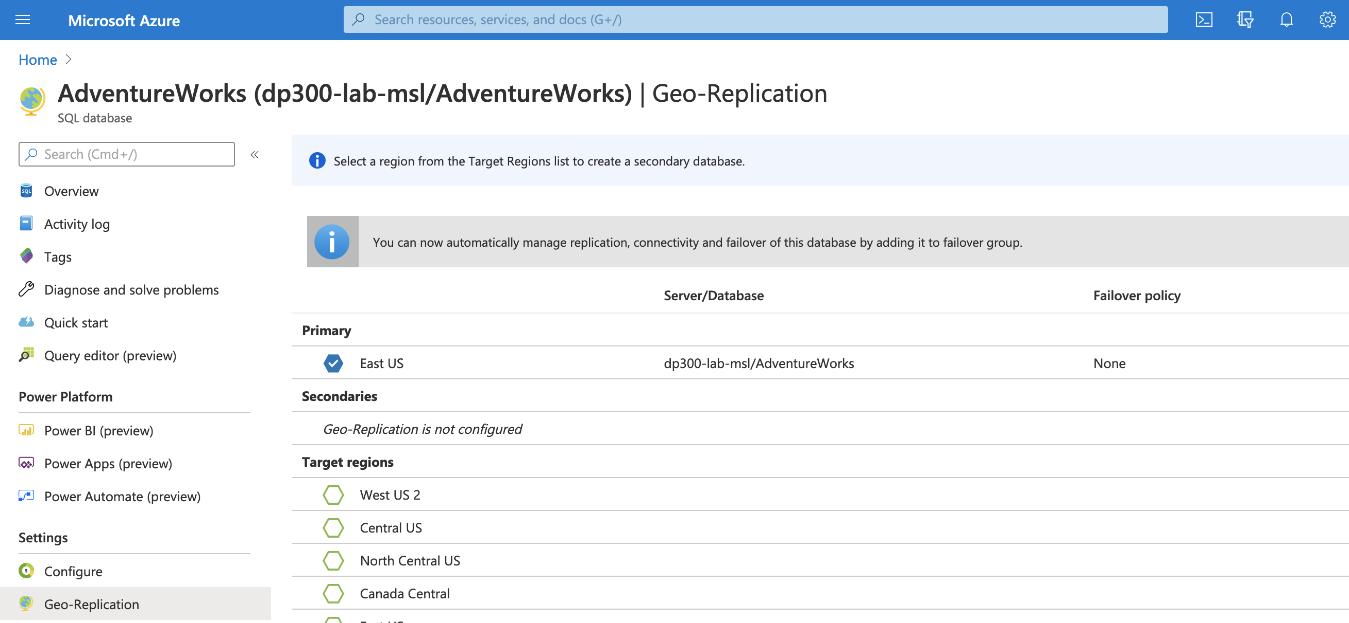
[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-03.png)

Mudou de GEO REPLICATION para REPLICAS, como evidencia abaixo:

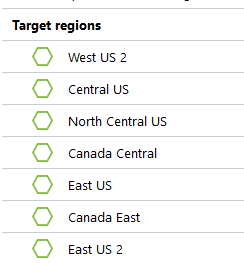




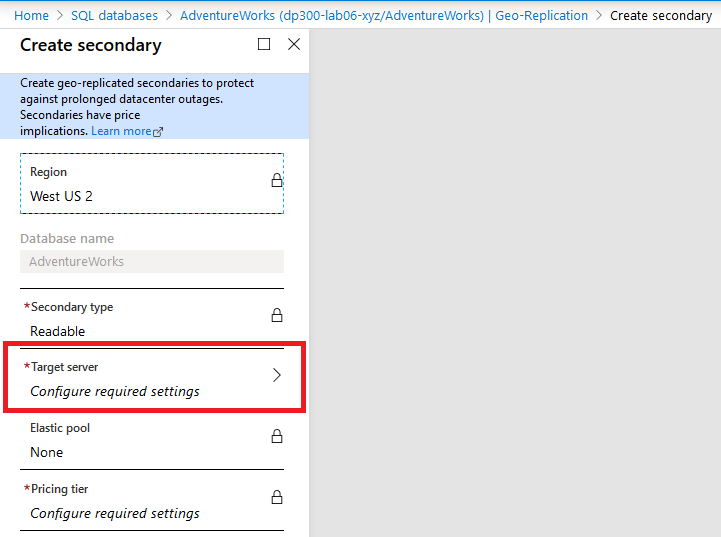
The region where the database is currently configured is shown in a blue hexagon with a white checkmark, as shown below. You will see that geo-replication is not configured.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-04.png)

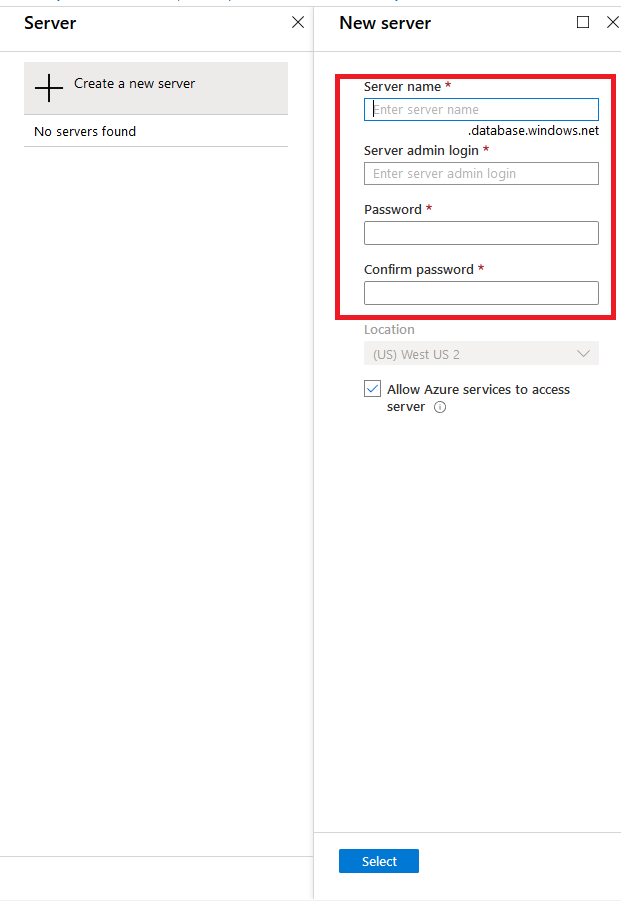
* 1. Select a Target Region. All available regions will have a hexagon with a green outline.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-05.png)

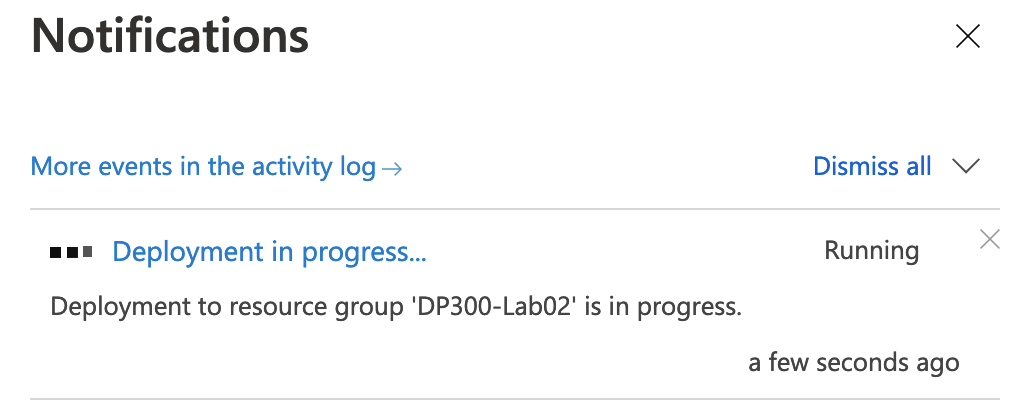
* 1. In this example, West US 2 was selected as can be seen on the Create secondary blade. Select Target server.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-06.png)

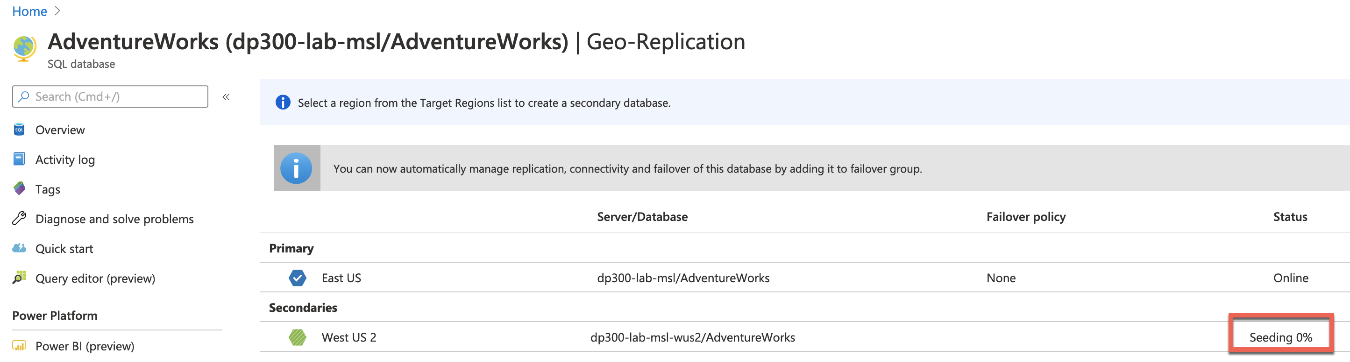
* 1. Select Create a new server. On the New server blade, enter a name of **module7geo** (appended with your initials or a unique suffix), a valid admin login, and a secure password. Make a note of these credentials. Click Select when done.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-07.png)

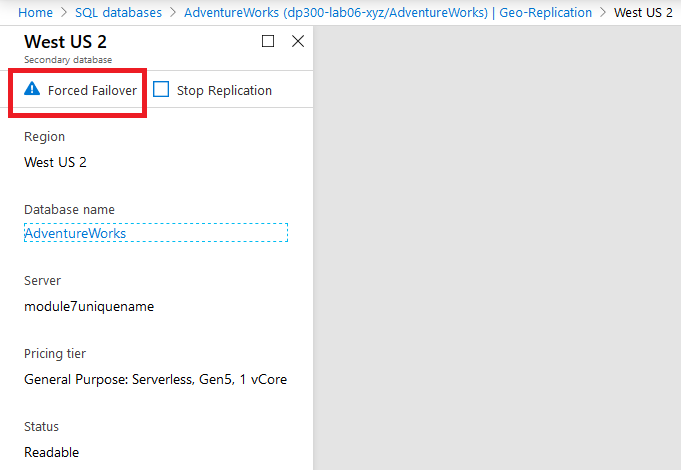
* 1. ]Click **OK** on the Create secondary blade. The secondary server and the database will now be created. To check the status, look under the bell icon at the top of the portal. If successful, it will progress from Deployment in progress to Deployment succeeded.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-08.png)

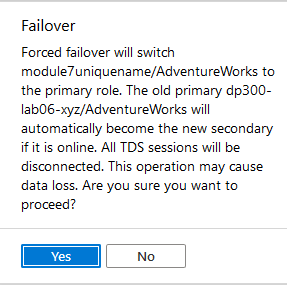
You will also notice that on the Geo-Replicate page, the status column of the Secondary will go from Initializing, to a seeding percentage complete, as shown below, to Readable when the replication in synchronized

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-09.png)

* 1. Now that the Azure SQL Database is configured with geo-replication, you will perform a failover. Select your secondary server (West US 2 as shown in the previous picture but you may have chosen a different region).
  2. On the blade, click Forced Failover.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-10.png)

* 1. When prompted, click Yes.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-11.png)

The status of the primary replica will switch to Pending and the secondary, Failover. The process will take a few minutes. When complete, the roles will switch with the secondary becoming the new primary and the old primary becoming the secondary.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-12.png)

**Exercise 2: Backup to URL and Restore from URL**

Estimated Time: 45 minutes

The tasks for this exercise are as follows:

* + Configure backup to URL
  + Back up WideWorldImporters
  + Restore WideWorldImporters

**Task 1: Configure Backup to URL**

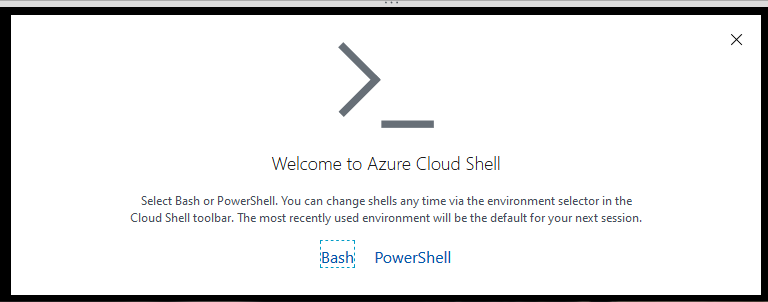
Before backing up a database in SQL Server to Azure, there are some configuration tasks to perform.

**Note:** There are several long strings, including storage account keys and shared access signatures, that are generated and then reused. You should consider opening up a Notepad file within the lab VM to use for holding these strings.

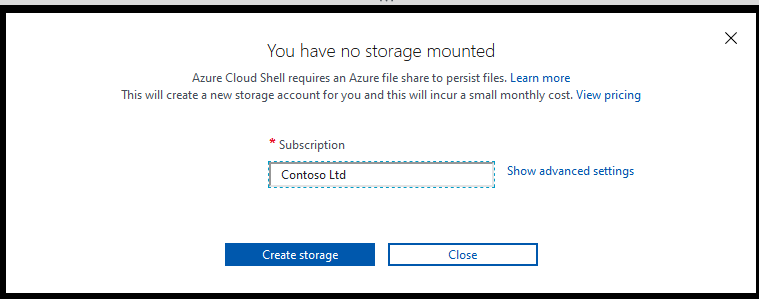
* + Verify that your context is the LON-SQL1 Virtual Machine.
  + Start Edge and log into the Azure portal ([https://portal.azure.com](https://portal.azure.com/)) unless you are are already there.
  + Open a Cloud Shell prompt by selecting the icon shown below in the upper right corner.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-13.png)

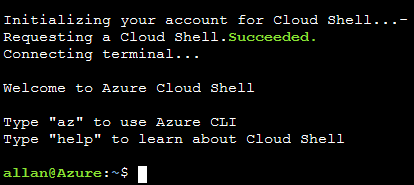
* + At the bottom half of the portal, you may see a message welcoming you to Azure Cloud Shell, if you have not yet used Cloud Shell. Select Bash.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-14.png)

* 1. If you have not previously used Cloud Shell, you must give it storage. Click Create Storage in the dialog below.

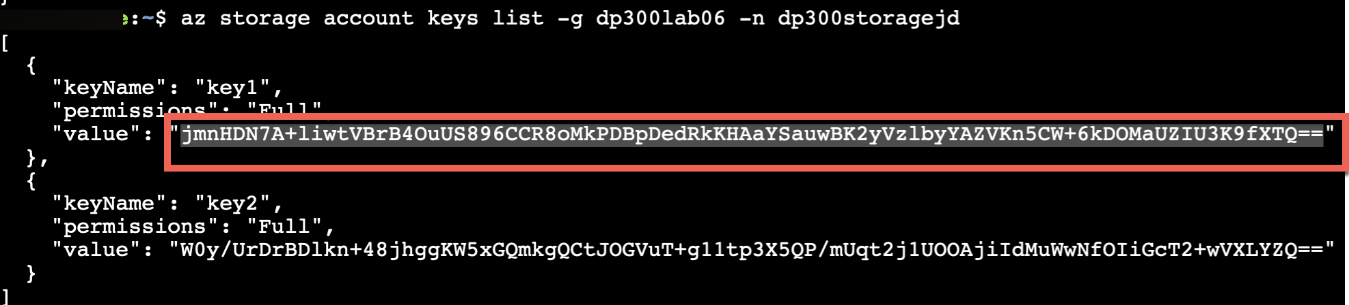
[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-15.png)

* 1. If you have already used Cloud Shell, just make sure the upper level corner of the Cloud shell screen show Bash. You can use the drop down arrow to select either PowerShell or Bash.  
     Once complete, you will see a prompt similar to the one below.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-16.png)

* 1. Create a storage account from the CLI using by copying the following command into cloud shell.  
     az storage account create -n dp300storagelab7mb -g DP-300-Lab7 --kind StorageV2 -l eastus2  
       
     Edit the command so that your storage account name is unique and all lower case with no special characters. You should change dp300storage in the above to a unique name like dp300storagemsl123. The value DP-300-Lab02 is the name of an existing Resource Group. Make sure you use one that has been created in an earlier lab. Change the region if desired. Press Enter to run the command.  
     Next you will get the account keys for your account, which you will use in subsequent steps. Execute the following code in cloud shell, after editting to use the same name (after the -n) and resource group (after the -g) that you used in the previous command.

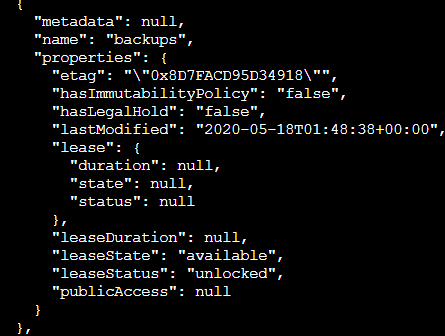
az storage account keys list -g DP-300-Lab02 -n dp300storage  
  
Your account key will be in the results of the above command. Make sure you Copy the returned value for key1 (without the double quotes) as shown here. You might save it in Notepad.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-17.png)

* 1. Backing up a database in SQL Server to a URL uses a storage account and a container within it. You will create a container specifically for backup storage in this step. To do this, execute the following command where dp300storage is the storage account name used when creating the storage account and storage\_key is the key generated above.  
     az storage container create --name "backups" --account-name "dp300storagelab7mb" --account-key "storage\_key" --fail-on-exist  
       
     The output should return true.

[Picture 1](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-18.png)

* 1. To further verify the container backups has been created, execute the following, where **dp300storagelab7** is the storage account name used you created and storage\_key is the key you generated above.  
     az storage container list --account-name "dp300storagelab7mb" --account-key " "  
       
     Part of the output should return something similar to below.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-19.png)

* 1. A shared access signature (SAS) at the container level is required for security. This can be generated via Cloud Shell or PowerShell. Execute the following, where dp300storage is the storage account name you created above, storage\_key is the key generated above, and date\_in\_the\_future is a time later than now. date\_in\_the\_future must be in UTC. An example is 2021-12-31T00:00Z which translates to expiring at October 31, 2020 at midnight:  
     az storage container generate-sas -n "backups" --account-name "dp300storagelab7" --account-key "storage\_key" --permissions "rwdl" --expiry "date\_in\_the\_future" -o tsv  
       
     The output should return something similar to the value shown below which will be used in the next Task. You can copy and save the value in Notepad along with the key you saved earlier.

[Picture 25](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-20.png)

**Task 2: Back Up WideWorldImporters**

Now that the functionality is configured, you can generate a backup file as a blob in Azure.

* 1. Open SQL Server Management Studio and ensure you are connected to LON-SQL1.
  2. Click New Query.
  3. Create the credential that will be used to access storage in the cloud with the following Transact-SQL. Fill in the appropriate values, where **dp300storagelab7** is the storage account name created in Task 1, Step 8 and sas\_token is the value generated in Task 1, Step 10.  
     IF NOT EXISTS   
       
     (SELECT \* FROM sys.credentials   
       
     WHERE name = '<https://dp300storagelab7.blob.core.windows.net/backups>')   
       
     BEGIN  
       
     CREATE CREDENTIAL [<https://dp300storagelab7.blob.core.windows.net/backups>]  
       
     WITH IDENTITY = 'SHARED ACCESS SIGNATURE',  
       
     SECRET = 'se=2021-12-31T00%3A00Z&sp=rwdl&sv=2018-11-09&sr=c&sig=1lnMje4oYwj535YN73eat2LuEuSquz88jOCe1JxDuQQ%3D'  
       
     END;  
       
     GO
  4. Click Execute. This should be successful.
  5. Back up the database WideWorldImporters to Azure with the following command in Transact-SQL:  
     BACKUP DATABASE WideWorldImporters   
       
     TO URL = '<https://dp300storagelab7.blob.core.windows.net/backups/WideWorldImporters.bak>';  
       
     GO   
     where dp300storage is the storage account name used in Task 1.  
     If successful, you should see output similar to this:  
     Processed 1240 pages for database 'WideWorldImporters', file 'WWI\_Primary' on file 1.  
     Processed 53104 pages for database 'WideWorldImporters', file 'WWI\_UserData' on file 1.  
     Processed 3865 pages for database 'WideWorldImporters', file 'WWI\_InMemory\_Data\_1' on file 1.  
     Processed 1468 pages for database 'WideWorldImporters', file 'WWI\_Log' on file 1.  
     BACKUP DATABASE successfully processed 59677 pages in 14.839 seconds (31.419 MB/sec).  
     Completion time: 2020-05-18T08:01:41.6935863+00:00  
     If something is configured wrong, you will see an error message similar to the following:  
     Msg 3201, Level 16, State 1, Line 33  
     ‎Cannot open backup device '<https://dp300storage.blob.core.windows.net/container_name/WideWorldImporters.bak>'. Operating system error 50(The request is not supported.).  
     ‎Msg 3013, Level 16, State 1, Line 33  
     ‎BACKUP DATABASE is terminating abnormally.  
     Check that you did not mistype anything and that everything was created successfully.
  6. To see that the file is actually in Azure, you can use Storage Explorer or Azure Cloud Shell. The syntax in Bash is as follows, where dp300storage is the storage account name used in Task 1, storage\_key is the key used there as well.  
     **az storage blob list -c "backups" --account-name "dp300storagelab7" --account-key "storage\_key"**  
       
     Sample output is shown below.

[](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/images/dp-3300-module-77-lab-21.png)

**Task 3: Restore WideWorldImporters**

This task will show you how to restore a database.

* 1. In SQL Server Management Studio, in a New Query window, execute  
     **USE WideWorldImporters;  
     GO**
  2. Now execute the statement below to return the very first row of the Customers table which has a CustomerID of 1. Note the name of the customer.  
     **SELECT TOP 1 \* FROM Sales.Customers;  
     GO**
  3. Run this command to change the name of that customer.  
     **UPDATE Sales.Customers  
     SET CustomerName = 'This is a human error'  
     WHERE CustomerID = 1;  
     GO**
  4. Re-run Step 2 to verify that the name has been changed. Now imagine if someone had changed thousands or millions of rows without a WHERE clause – or the wrong WHERE clause.
  5. To restore the database to get it back to where it was before the change you made in Step 3, execute the following, where dp300storage is the storage account name used in Task 1.  
     **USE master;  
     GO  
       
     RESTORE DATABASE WideWorldImporters   
     FROM URL = '**[**https://dp300storagelab7.blob.core.windows.net/backups/WideWorldImporters.bak**](https://dp300storagelab7.blob.core.windows.net/backups/WideWorldImporters.bak)**';  
     GO**  
     The output should be similar to this:  
     Processed 1240 pages for database 'WideWorldImporters', file 'WWI\_Primary' on file 1.  
     Processed 53104 pages for database 'WideWorldImporters', file 'WWI\_UserData' on file 1.  
     Processed 1468 pages for database 'WideWorldImporters', file 'WWI\_Log' on file 1.  
     Processed 3865 pages for database 'WideWorldImporters', file 'WWI\_InMemory\_Data\_1' on file 1.  
     RESTORE DATABASE successfully processed 59677 pages in 16.167 seconds (28.838 MB/sec).  
     Completion time: 2020-05-18T08:35:06.6344123+00:00
  6. When the restore of WideWorldImporters is finished, re-run Steps 1 and 2. The data will be back to what it was.

Go

* 1. © 2020 GitHub, Inc.
  2. [Terms](https://github.com/site/terms)
  3. [Privacy](https://github.com/site/privacy)
  4. [Security](https://github.com/security)
  5. [Status](https://githubstatus.com/)
  6. [Help](https://docs.github.com/)
  7. [Contact GitHub](https://github.com/contact)
  8. [Pricing](https://github.com/pricing)
  9. [API](https://docs.github.com/)
  10. [Training](https://services.github.com/)
  11. [Blog](https://github.blog/)
  12. [About](https://github.com/about)rom <[*https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/Labs/DP-300\_07\_lab.md*](https://github.com/MicrosoftLearning/DP-300T00-Administering-Relational-Databases-on-Azure/blob/master/Instructions/Labs/DP-300_07_lab.md)>