# Objective:

The purpose of this document is to demonstrate how to **migrate an on premise SQL Server based database to Azure SQL DB**.

# Prerequisites:

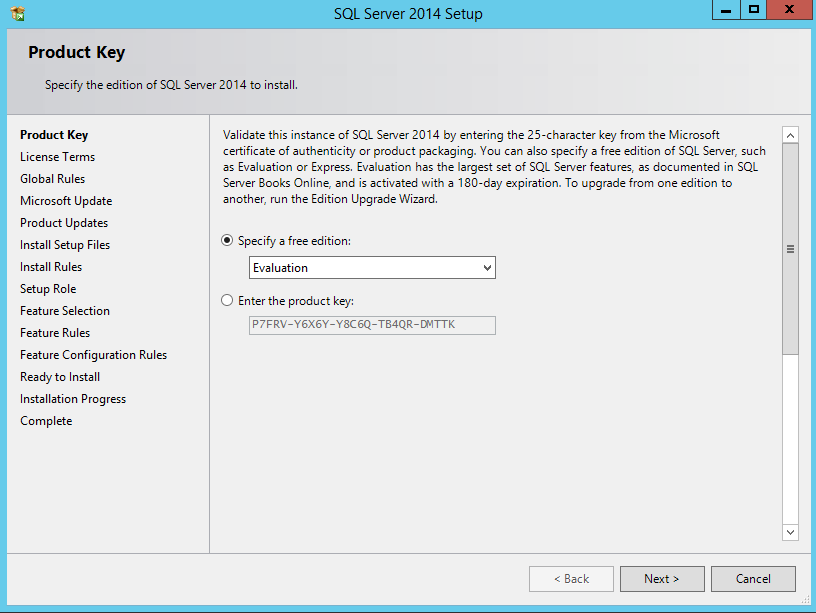
* This demonstration uses an already built SQL Server database named Adventure Works. This database can be downloaded from [here](http://msftdbprodsamples.codeplex.com/releases/view/37304).
  + If you want to test the features of Azure SQL Database V12, use the standard version of the AdventureWorksDB. This database can be downloaded from [here](http://msftdbprodsamples.codeplex.com/releases/view/55330).
* SQL Server 2014 (Standard) has been used in this demo as on premise SQL Server
* Valid Azure Subscription

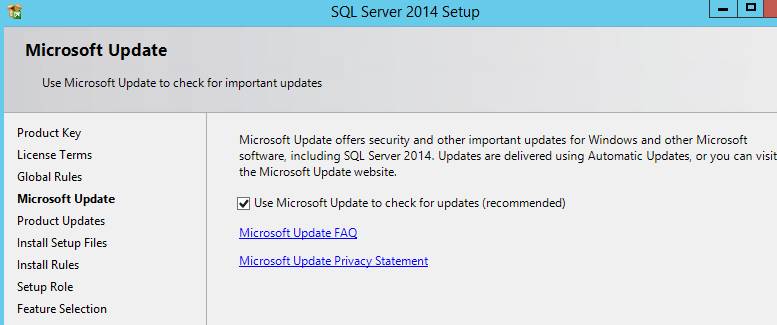
# Steps Overview:

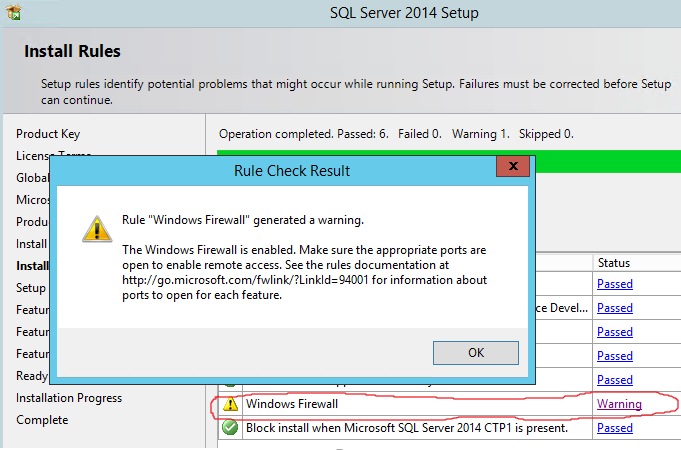
* Setting up On-Premise Database
* Creating a Azure SQL Database from Portal
* Migrating database from on premise to Azure SQL Database

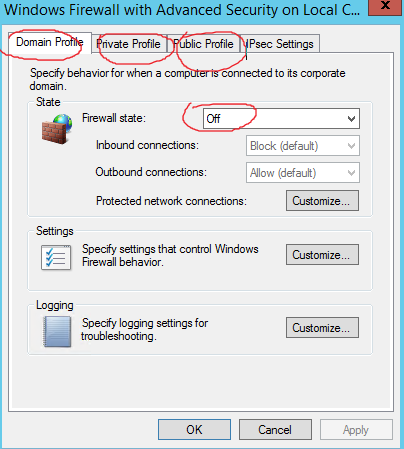
# Setting up On-Premise Database

* Make sure that SQL Server 2014 is installed on a local machine. We’ve used SQL Server 2014 Standard Edition for this demonstration.
* Screenshots of important steps during installation are shown below.

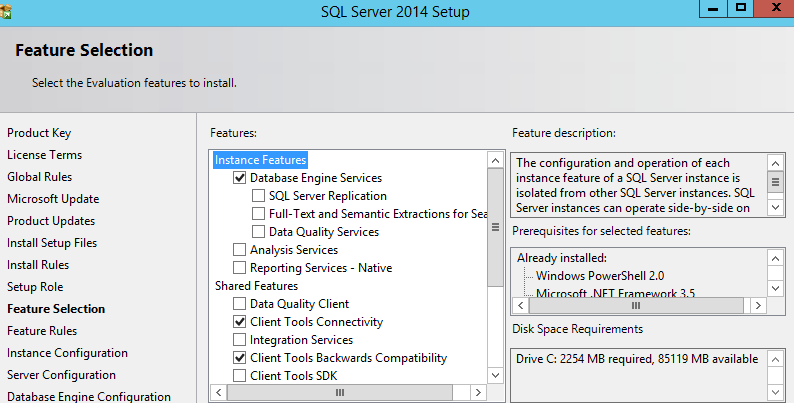


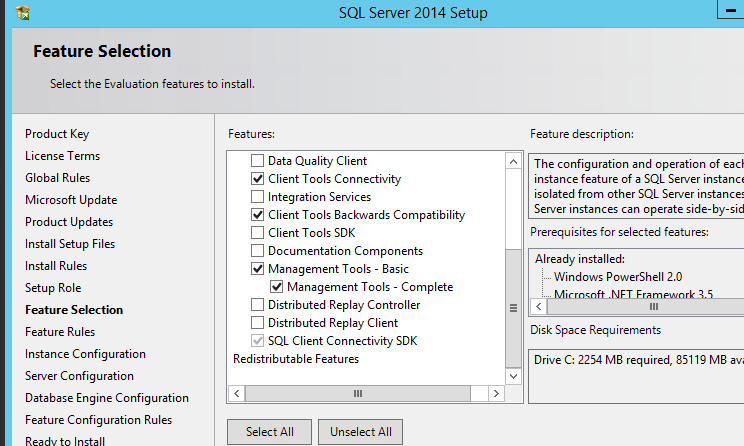


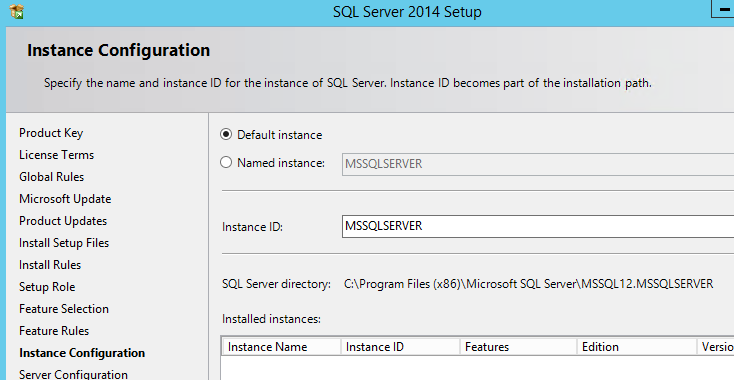


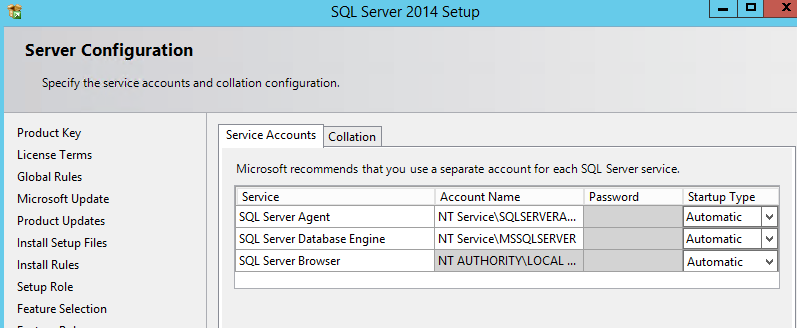


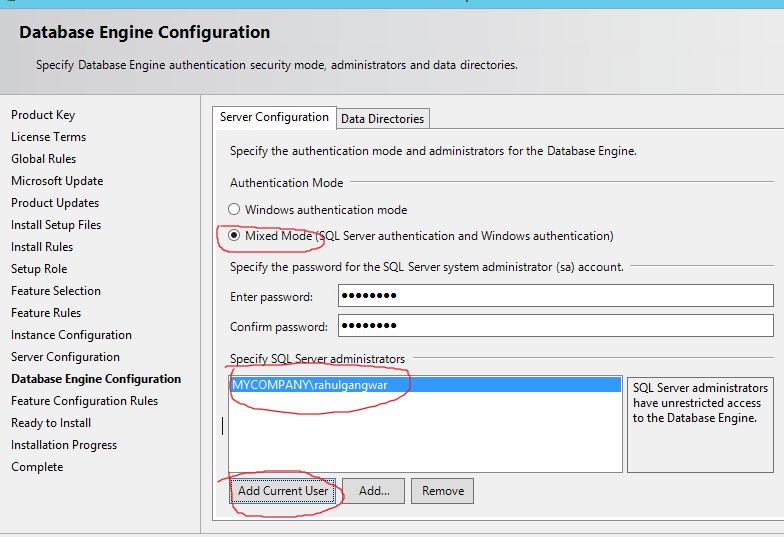
* You can either add exceptions to SQL ports (UDP/TCP) in host firewall or you can temporarily disable it. As shown above, I chose to disable it for demonstration. I’ve disabled it for ALL firewall profiles (Domain, Private & Public).



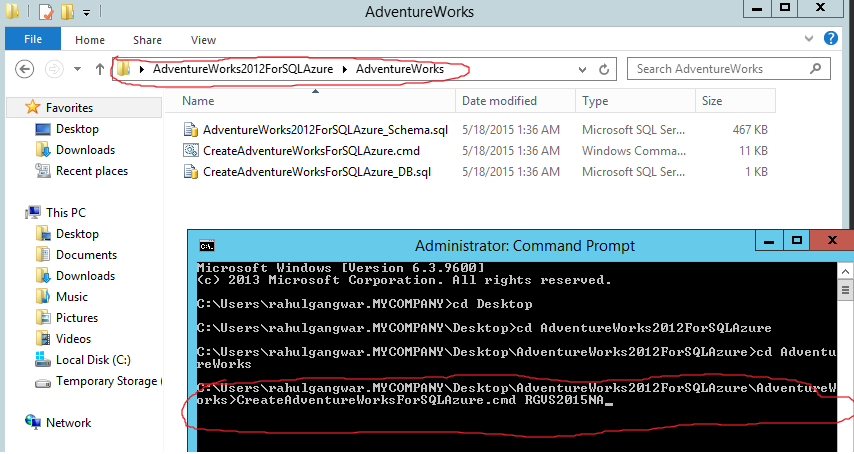




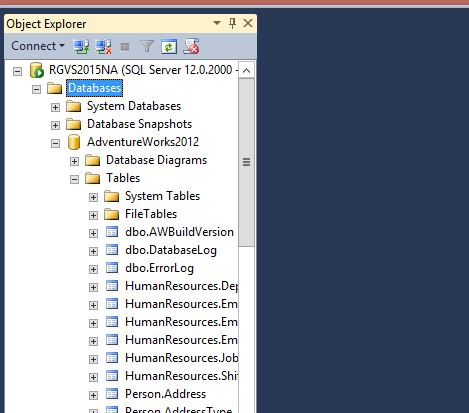




* Download and extract Adventure Works Database on local drive.
* Open command prompt in admin mode and navigate to “AdventureWorks” folder within the extracted folder.
* Run command **CreateAdventureWorksForSQLAzure.cmd** *servername* *username* *password* where *servername* is your local database server name (given all defaults were chosen during installation) and username/password is *<<sa>>* and *<<sa account’s password>>*. Recall that we created this user during installation. Screenshot shown below without username/password.



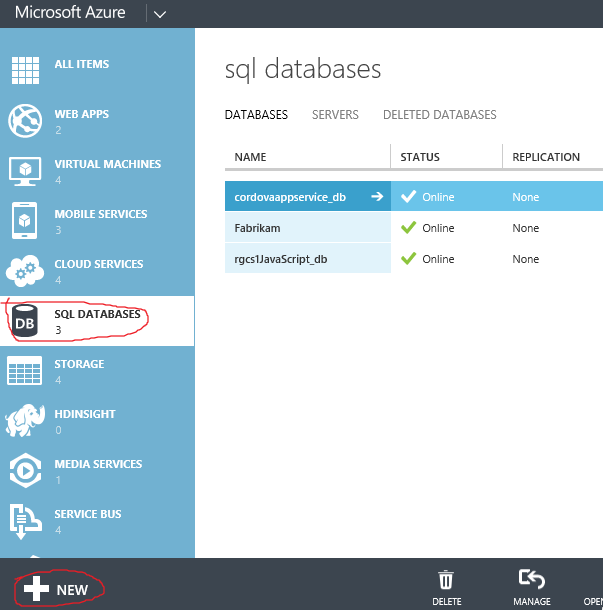
* After this you can verify the database successfully created from Management Studio. Make sure to verify this because no matter what error you get, the command prompt will show installation completion message at the end which is confusing. You need to scroll up to see exact error messages in case database creation failed.



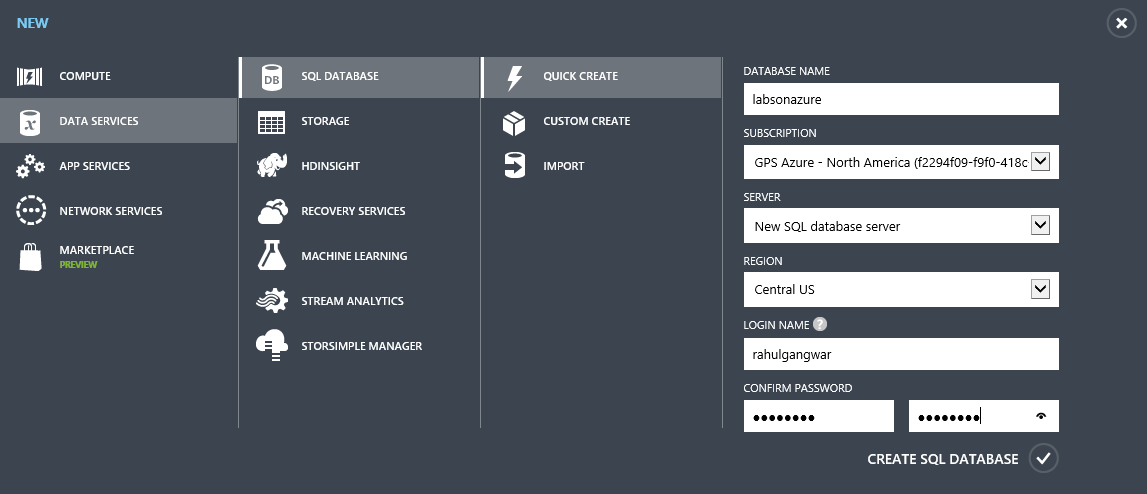
* Congratulations! On premise DB is now successfully setup.

# Creating an Azure SQL Database from Portal

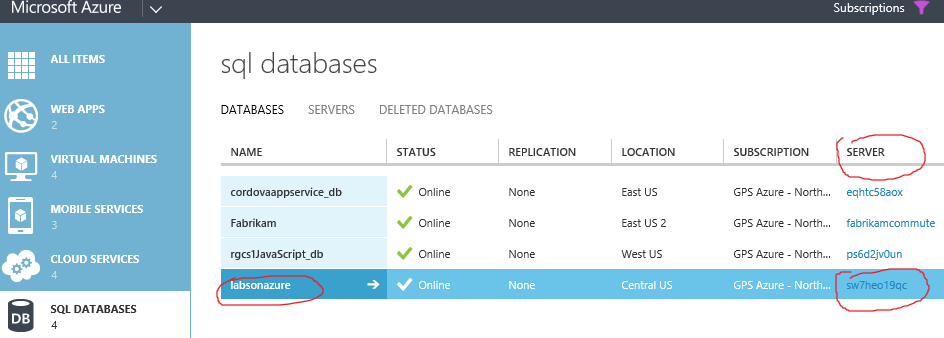
* Navigate to Azure portal (manage.windowsazure.com) and select “SQL Databases” from left. Then create a new database using “New” menu from bottom as shown below.



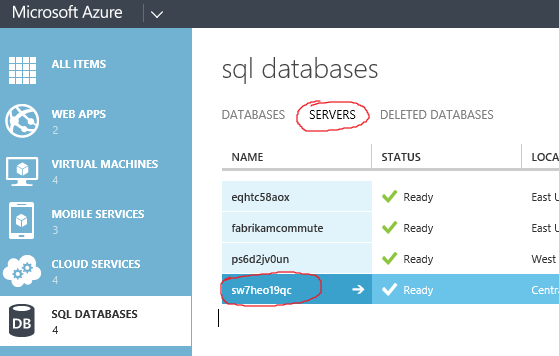
* Fill the details of database that you want to create as shown below:



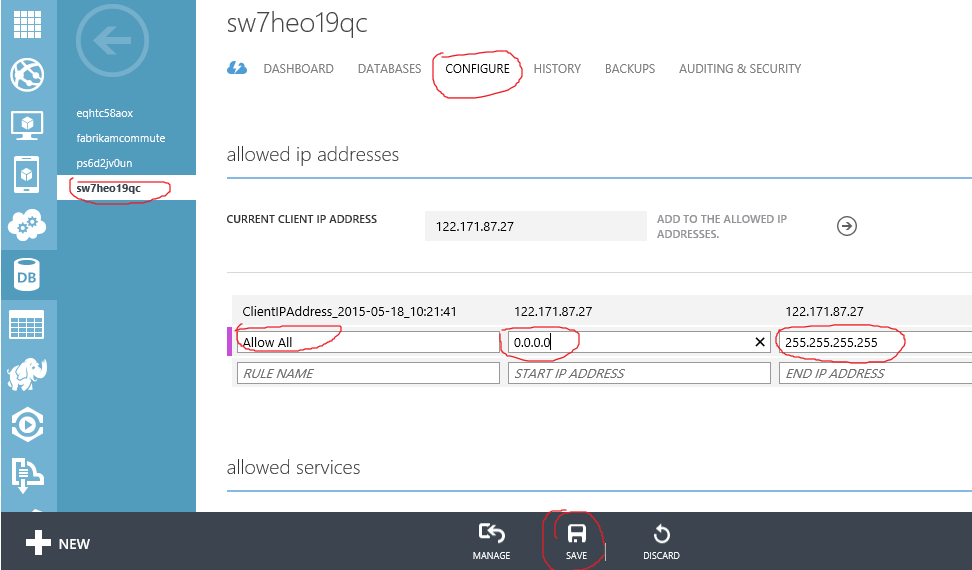
* Configure firewall of SQL Database to allow your client’s IP address. In figure below I’ve allowed ALL IP addresses. This can be done by first checking which is the “server” where your database is instantiated (here “sw7he019qc”) as shown below.



* Then select the server by going to “servers” tab as shown below.



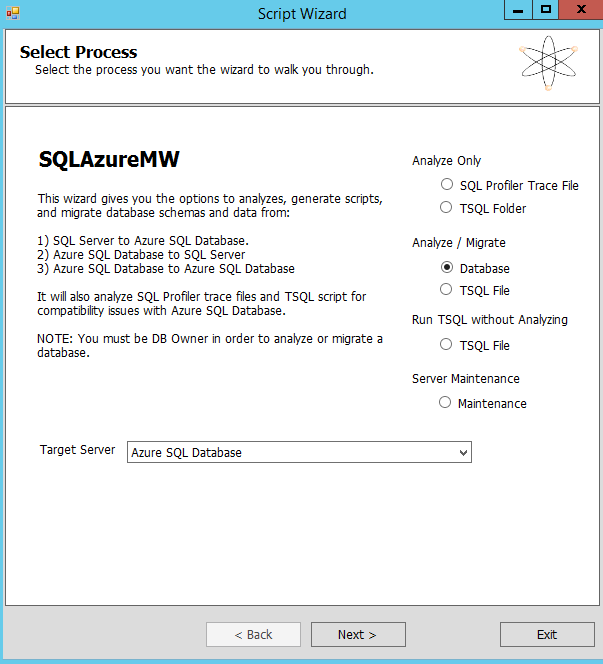
* Then navigate to “Configure” tab on top and then under “allowed ip addresses”, add the IP address of the machine from where you will run the wizard. Here I have added all IP addresses which may not be best practice. Then click on Save as shown below.



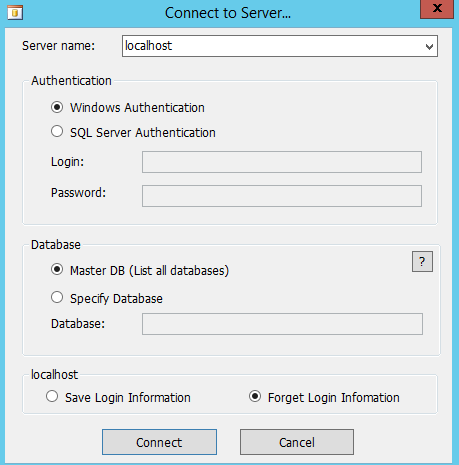
* Congratulations! Your database on Azure is setup and configured.

# Migrating database from on premise to Azure SQL Database

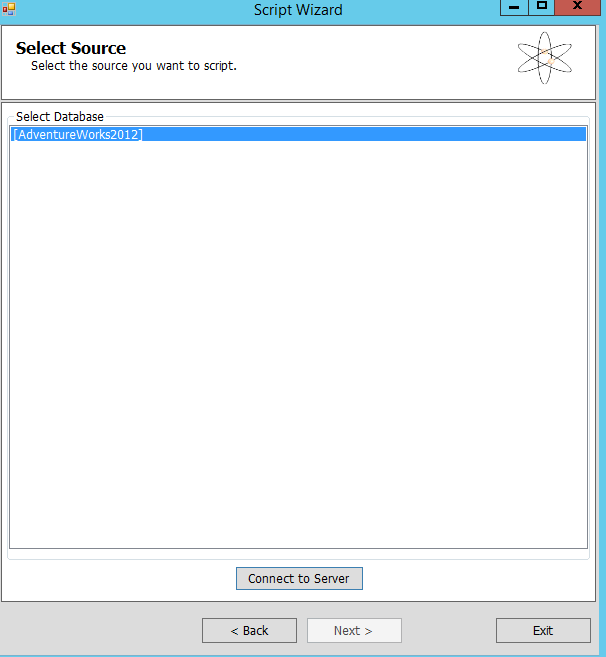
* Download and unzip SQL Server Migration Wizard from [here](http://sqlazuremw.codeplex.com/) and run SQLAzureMW.exe.
* Select the option to “Analyze/Migrate -> Database” and Target Server as “Azure SQL Database” as shown below.

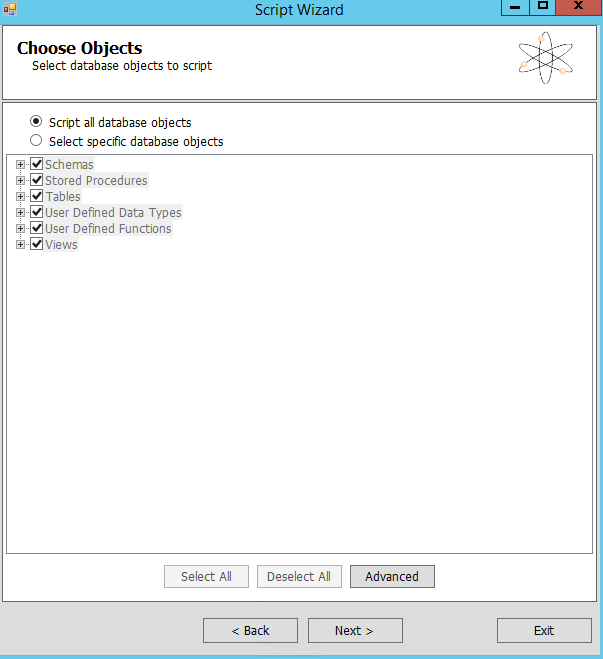


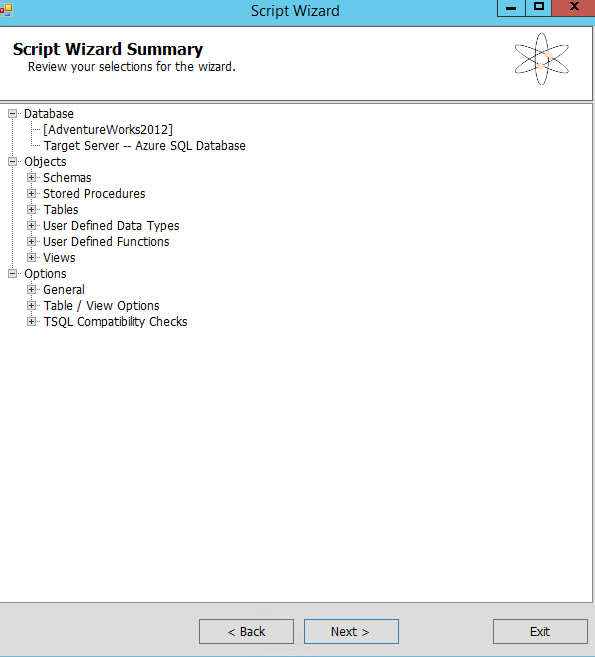
* Connect to your on premise database where Adventure Works reside as shown below. In case you have specific username/password and database instance name, then specify those. We’ve use defaults as they fit our need.

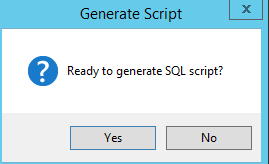


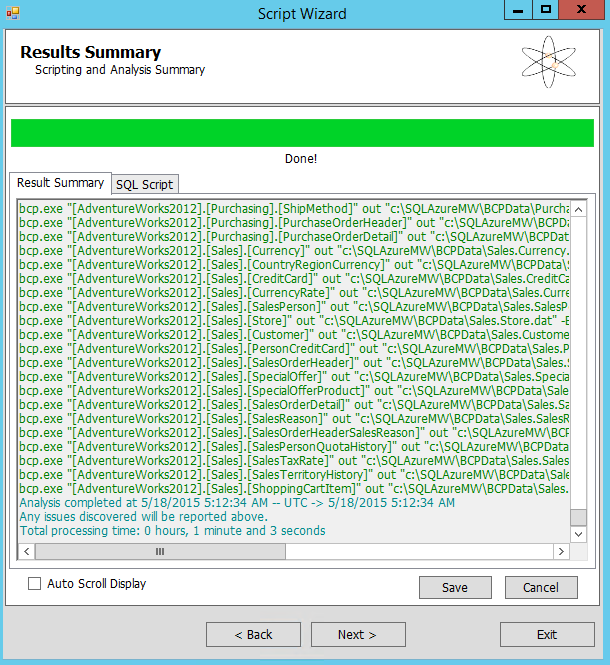
* Once connected, select “AdventureWorks” database and click Next and follow the wizard as shown below.



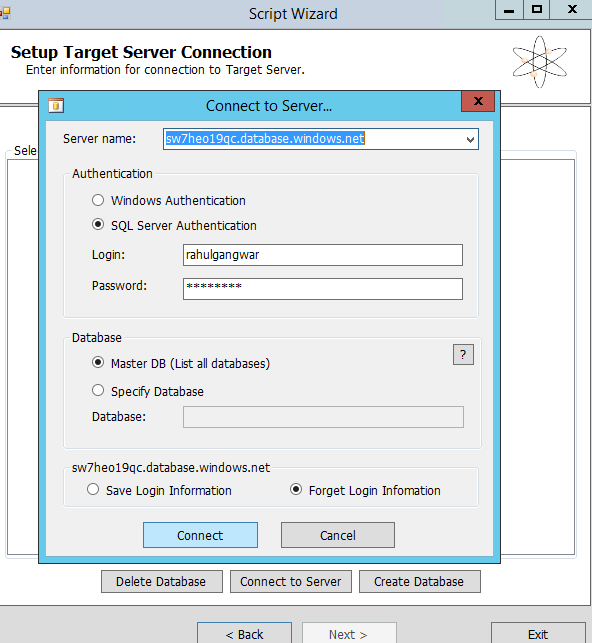


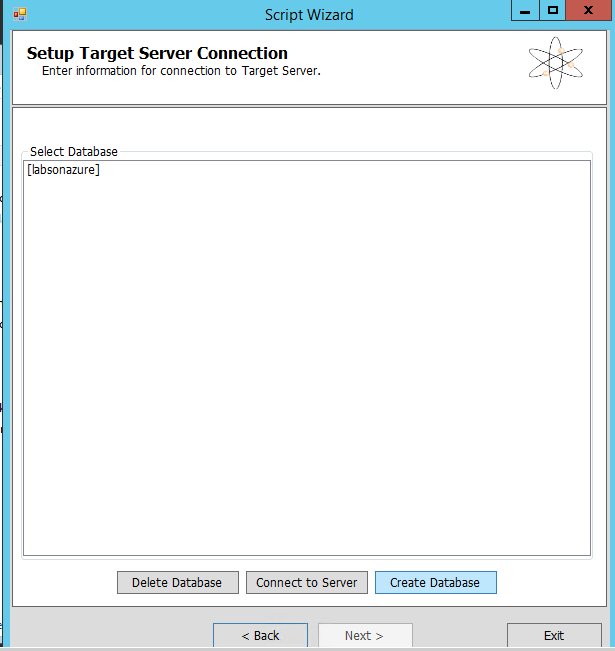




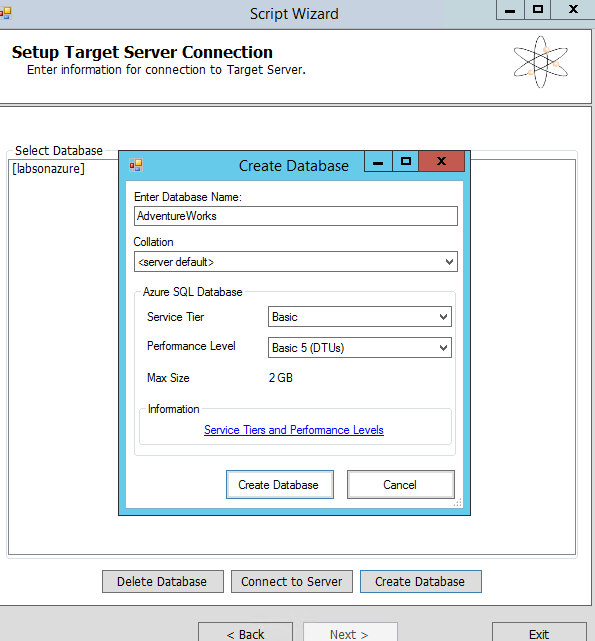


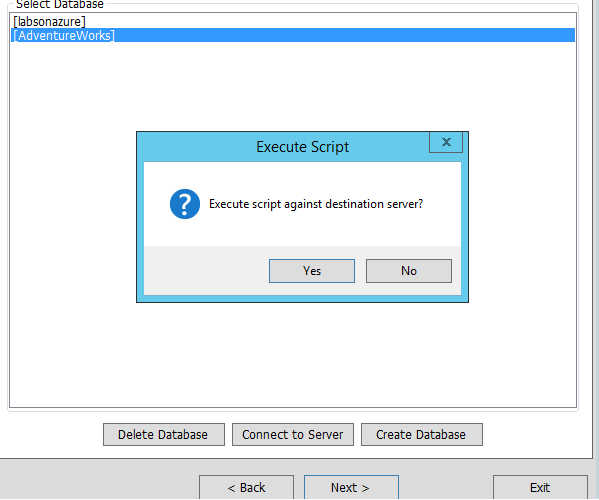
* The above process will analyse your database for migration as well as generate migration script. In below steps, the wizard connects with Azure SQL Database and run these scripts to move data to Azure. In below steps I’ve created a new database called “Adventure Works” on the server that we created earlier. I did not use “Labs on Azure” database that we created from portal although you could use that.
* On target server, clock “connect to server” to connect with SQL Azure DB. Note that I’ve specified the same server name as the one that is hosting “Labs On Azure”.

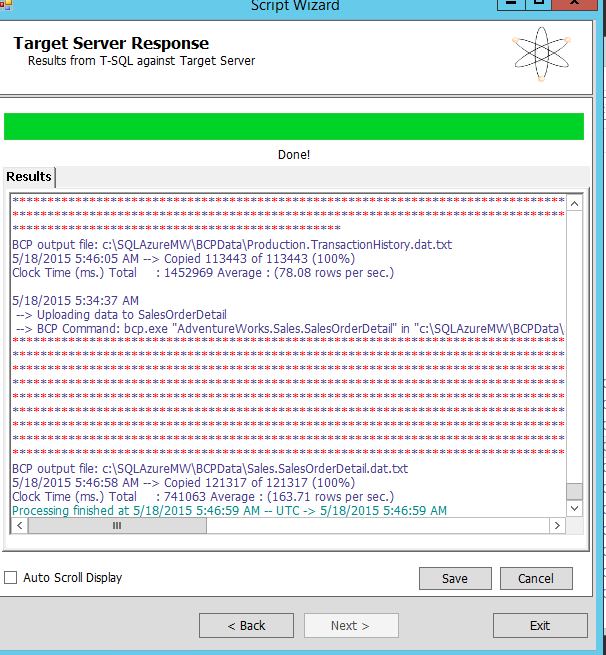




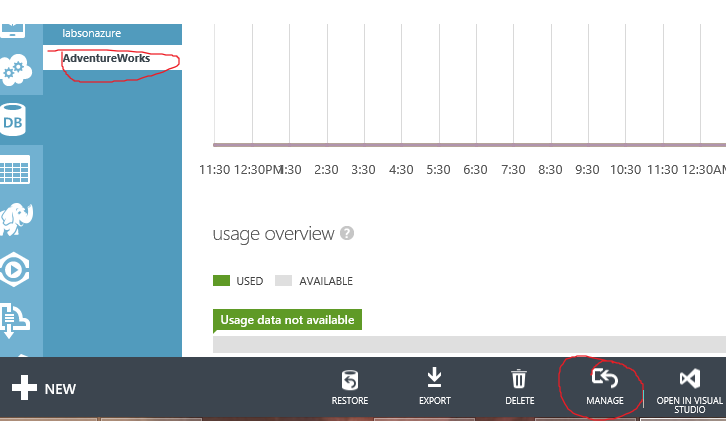
* Below, I’m creating new database although you could use “LabsOnAzure” as well.



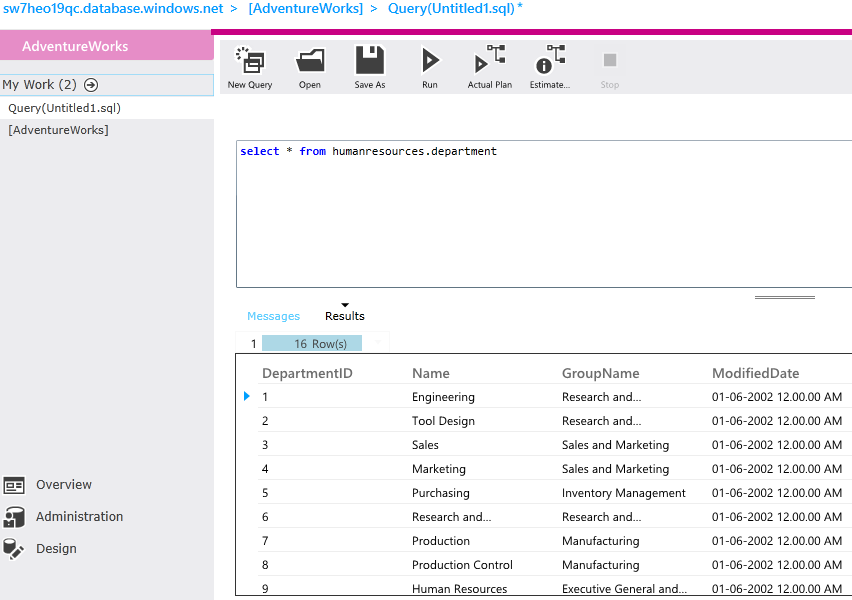




* Now you can verify the database existence by either navigating to Azure Portal or browsing through SQL Management Studio as show below.





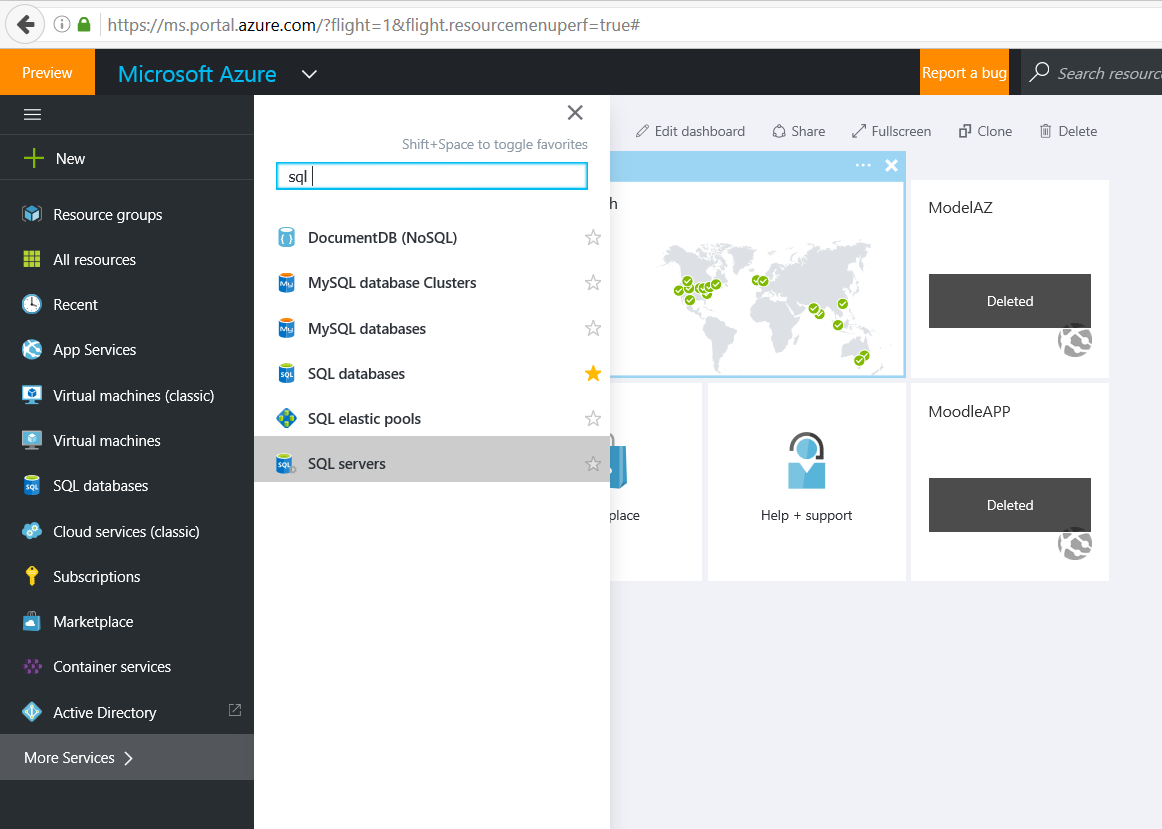


* Congratulations, the database has been migrated from on premise SQL Server to Azure SQL Database.

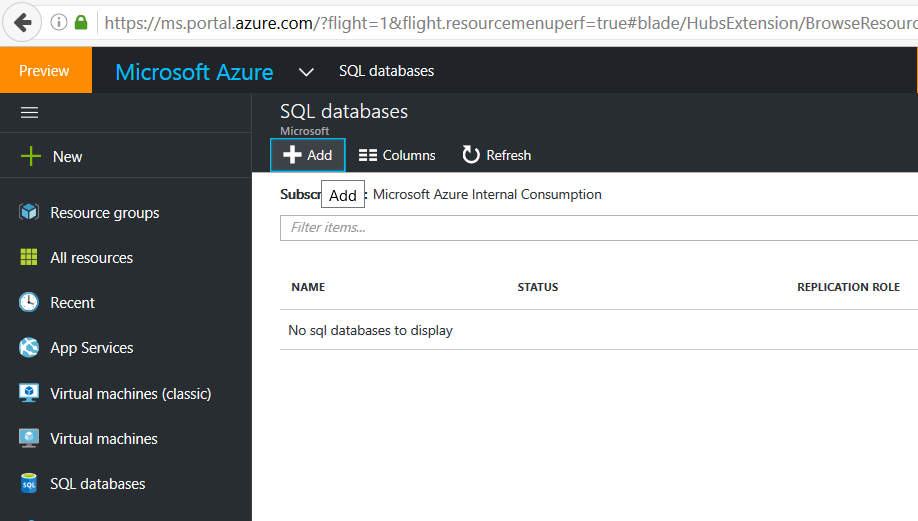
All other techniques of migration are mentioned [here](https://msdn.microsoft.com/en-us/library/azure/ee730904.aspx).

NOTE: SQL on Azure, you can access from the ARM portal as below:

* Login to <http://portal.azure.com>
* Use the same credentials as you use for <http://manage.windowsazure.com>
* Click on “More Services” in the search window, type SQL as shown below:



* Select “SQL Database”, click on “+Add”. Rest of the settings remain same:



Lab Completed!