

Source: C# Corner (www.c-sharpcorner.com)

PRINT

Article



Migrating On-Premise SQL Server Database To Azure SQL Database

By [Abdul Rasheed Feroz Khan](#) on **May 17 2016**

Introduction

By this we will be migrating On-Premise SQL Server database to [Azure](#) SQL database using different tools available, like – SQL Database Migration Wizard, SSMS – Deploy database to SQL Azure, Import and Export Wizard.

Here we will also discuss about the features that are not supported in Azure SQL database and ways to identify such features/syntax.

Scenario

1. Migrate on-premise database as-is and find out features not supported in Azure SQL Database. Rectify the problems and migrate again.
2. Migrate database using SQL Database Migration Wizard.
3. Migrate database to a Premium P3 database and compare performance.

Prerequisites

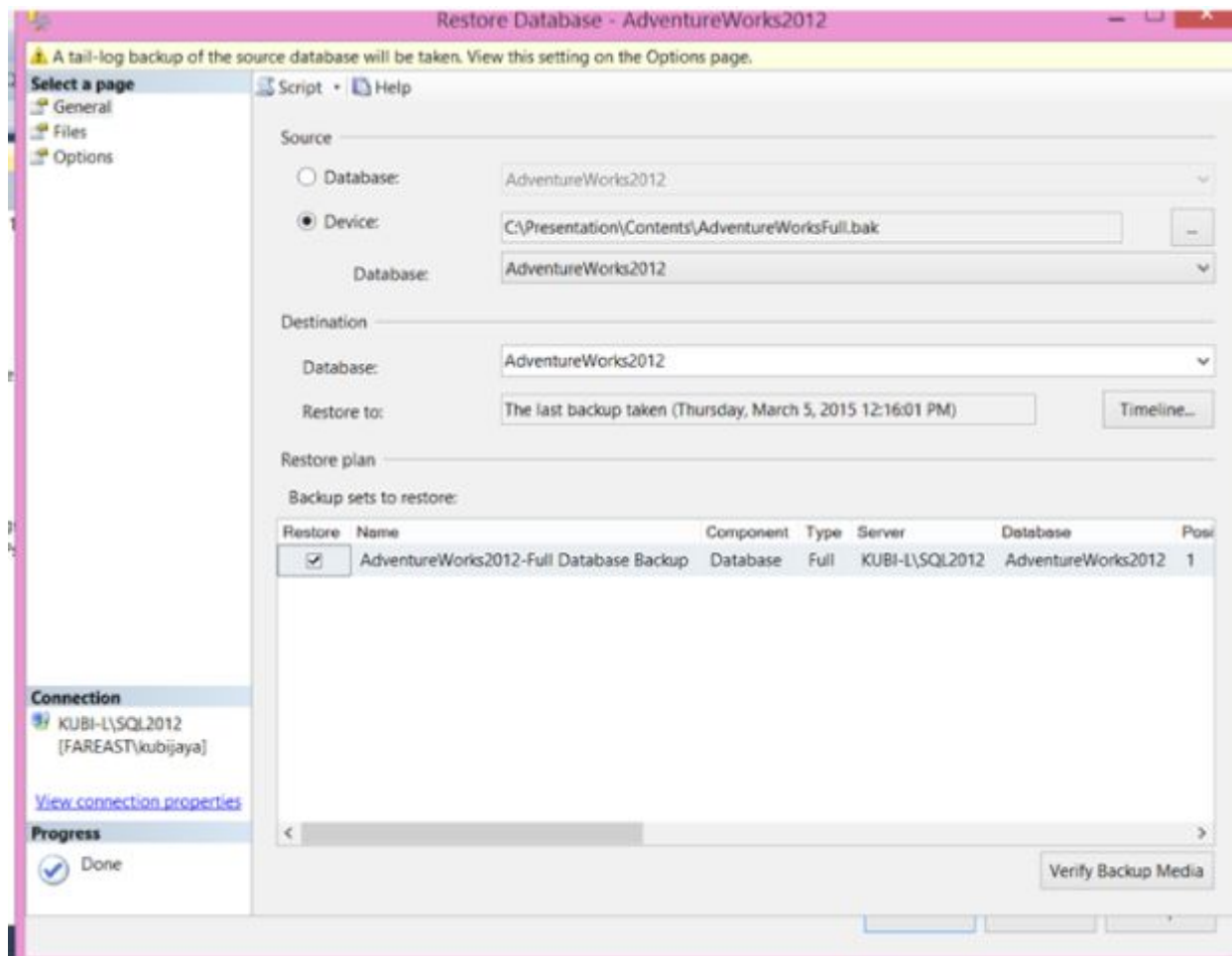
1. Microsoft Azure subscription.
2. OS - Microsoft Windows 8.1 Enterprise.
3. SQL Server 2012 Enterprise Edition.

Microsoft SQL Server 2012 - 11.0.2218.0 (X64)
Enterprise Edition (64-bit) on Windows NT 6.2 <X64> (Build 9200:) (Hypervisor)

4. The following in C:\Contents folder of the machine -
 - a. SQL Database migration Assistant(SQLAzureMW v4.15.4 Release Binary for SQL Server 2012.zip) - can be downloaded [here](#).
 - b. AdventureWorksFull.bak - can be downloaded from <http://bit.do/adventureworks2012>

Step 1

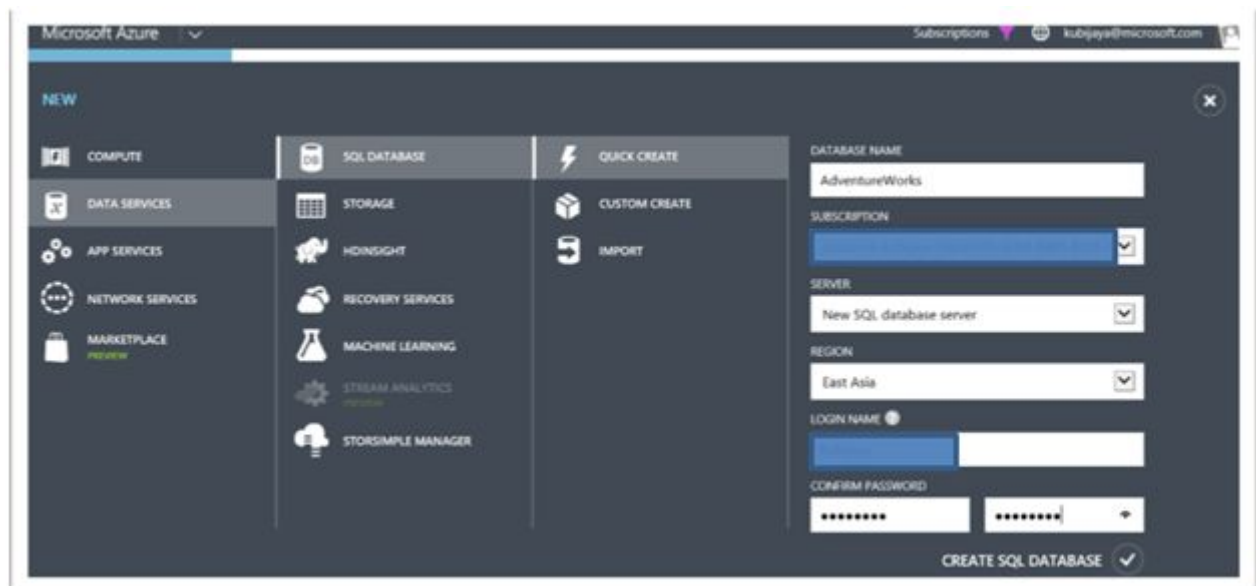
1. Open SQL Server Management Studio and restore AdventureWorksFull.bak placed under c:\contents. This will be the database that we will try to migrate.



- Log on to Azure portal at <https://manage.windowsazure.com/>.
- Create a new Azure SQL Database. Click on **"NEW"** at the left bottom of the screen.



- Click on **"Data Services"** -> SQL Database -> Quick Create -> Fill in the database details.



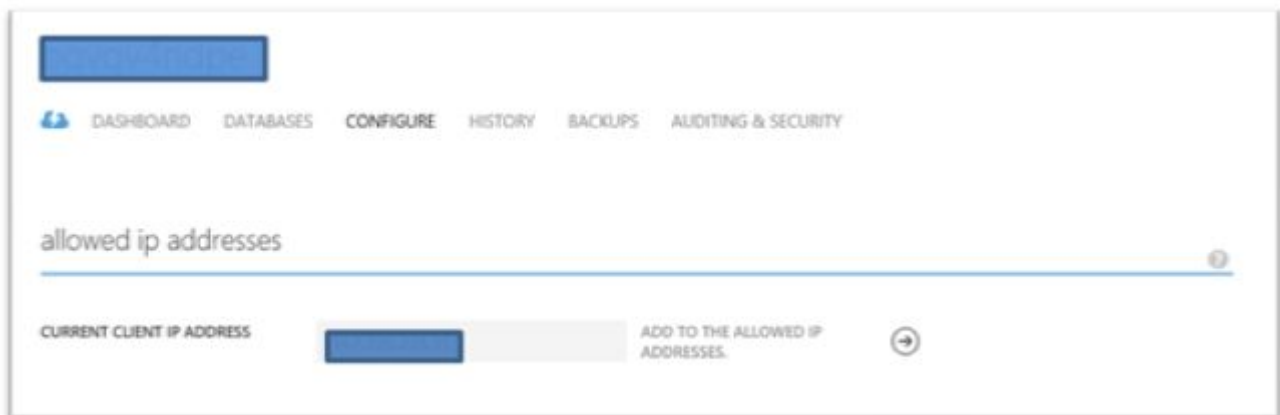
Note: The login name that you give is sort of sa account on Azure SQL Database.

5. This should create your database on a new server –



| NAME | STATUS | LOCATION | SUBSCRIPTION | SERVER | EDITION |
|----------------|----------|-----------|--------------|--------|-------------|
| AdventureWorks | ✓ Online | East Asia | | | Standard S0 |

6. Click on Server ->Configure and add the “**current client IP address**” to the allowed IP addresses.



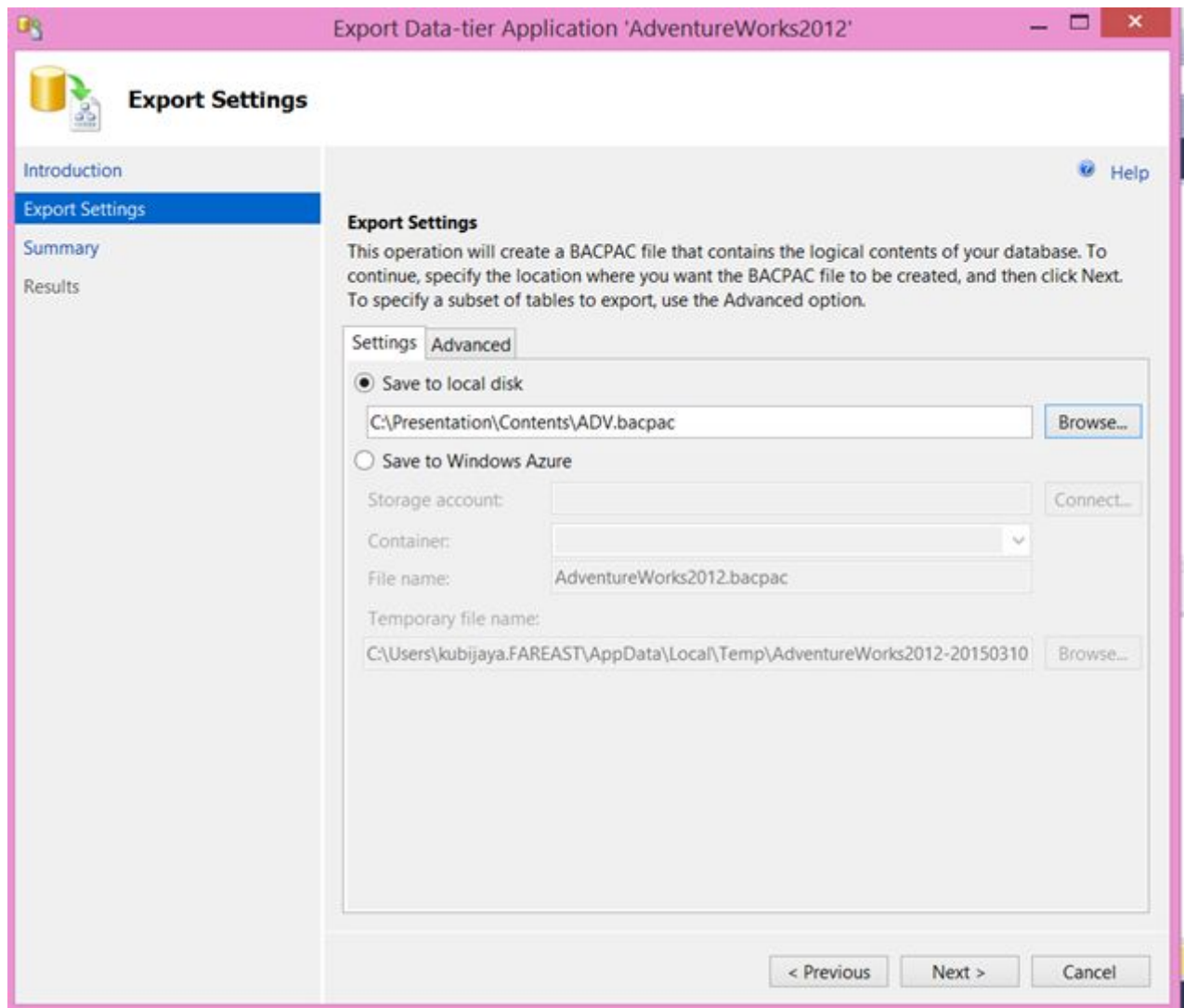
allowed ip addresses

CURRENT CLIENT IP ADDRESS

ADD TO THE ALLOWED IP ADDRESSES.

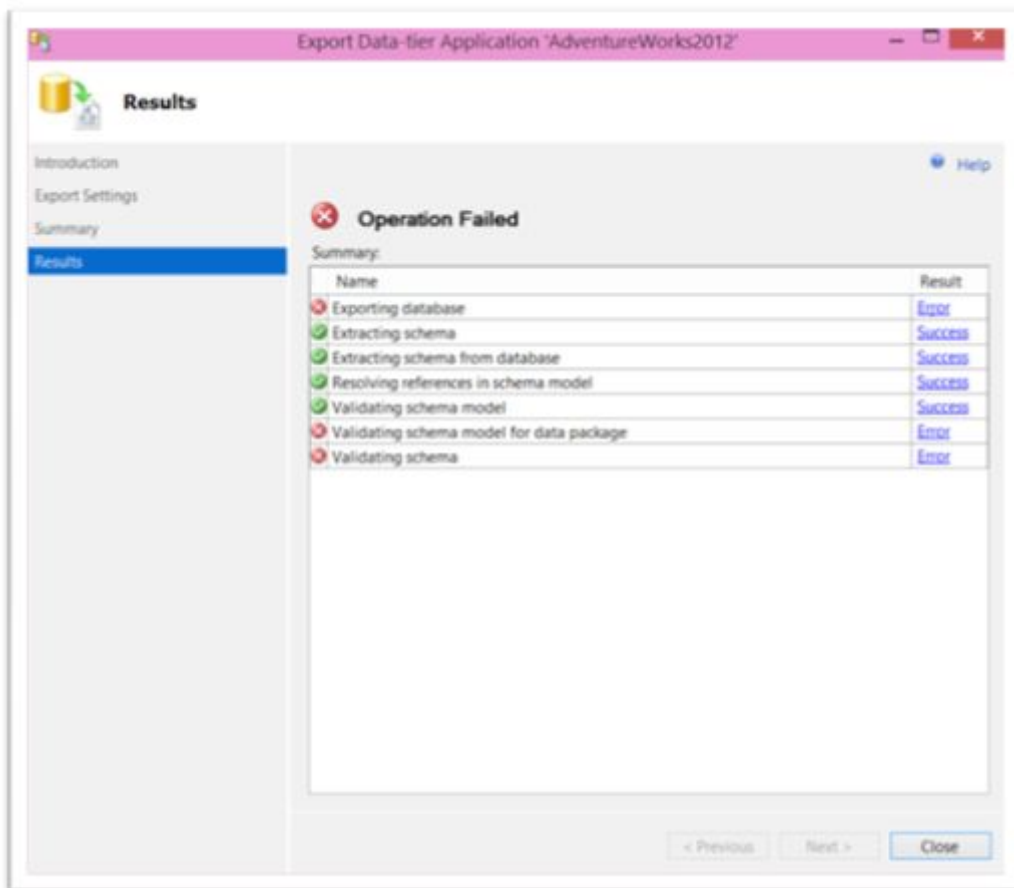
a. Following the steps above will have your Azure environment ready for data migration. Follow the steps below on the on-premise server.

7. On SSMS -> Right click on the database(AdventureWorks2012) -> tasks -> Export Data tier Application. Select “Save to local disk” and specify a location where the bacpacfile will be created.

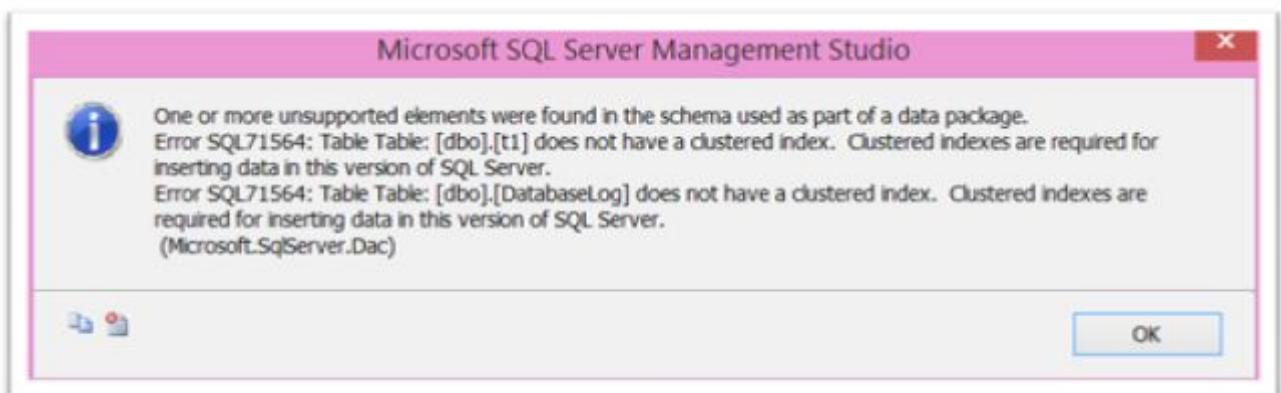


Note: A BACPAC contains both schema and data, but does not support being imported to a database project for schema modification. The primary use of a BACPAC is to move a database from one database service to another (either instances of the Database Engine or Azure SQL Database). A BACPAC can also be used to archive an existing database in an open format. These uses make it a good tool for migrations where the database requires no schema changes. Refer for [details](#).

8. Run through the rest of the tabs and click "Finish".
9. The export would fail as there are objects in the database which are not supported –



10. Click on the error message above, and it displays the following message –



The error message tells us that the two tables (t1 and DatabaseLog) do not have clustered indexes, which is a requirement for the Azure SQL database.

11. To fix the error, connect to your on-premise database and run the following queries –
 1. `CREATECLUSTEREDINDEXind_t1ONT1(name)`
 2. `CREATECLUSTEREDINDEXind_DatabaseLogONDatabaseLog(DatabaseLogID)`
12. We have fixed a problem and we do not know if there are other problems with the database or not. Let's use "**Deploy database to SQL Azure**" wizard this time. Even "Deploy database to SQL Azure" wizard creates a bacpac and then imports it into Azure SQL database.

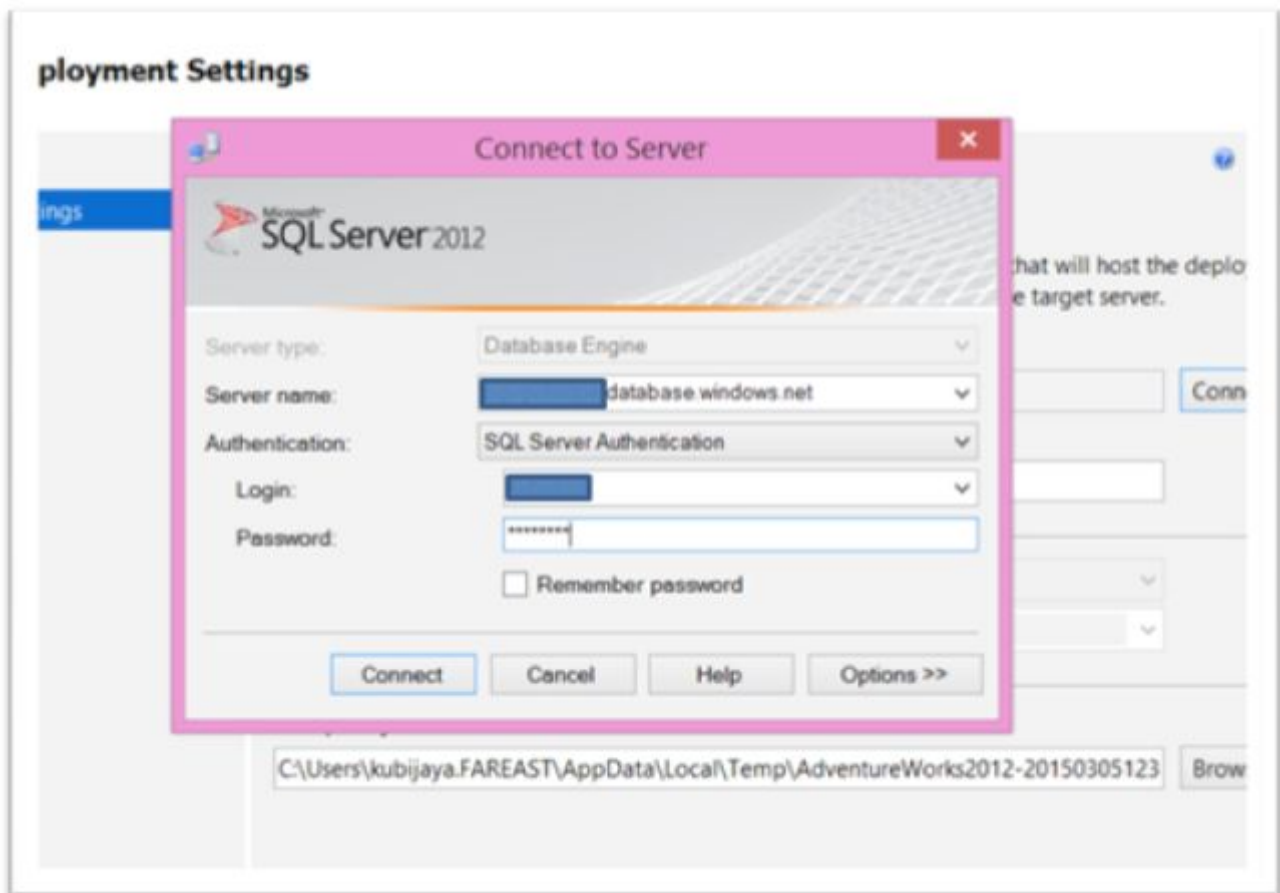
To use the wizard, right click on the database -> tasks -> Deploy database to SQL Azure. This will open up a window. Skip the introduction page and the window will look like this –

The screenshot shows a Windows application window titled "Deploy Database 'AdventureWorks2012'". The window has a sidebar on the left with a tree view containing "Introduction", "Deployment Settings" (which is selected and highlighted in blue), "Summary", and "Results". The main area of the window is titled "Deployment Settings" and contains the following sections:

- Specify Target Connection**: A section with a description: "Specify the name of the instance of SQL Server or the SQL Azure server that will host the deployed database, name the new database, and then click Connect to login to the target server." It includes a "Server connection:" text box and a "Connect..." button.
- New database name:**: A text box containing the text "AdventureWorks2012".
- SQL Azure database settings**: A section with two dropdown menus: "Edition of SQL Azure:" set to "Web" and "Maximum database size (GB):" set to "1".
- Other settings**: A section with a "Temporary file name:" text box containing the path "C:\Users\kubijaya.FAREAST\AppData\Local\Temp\AdventureWorks2012-20150305123" and a "Browse..." button.

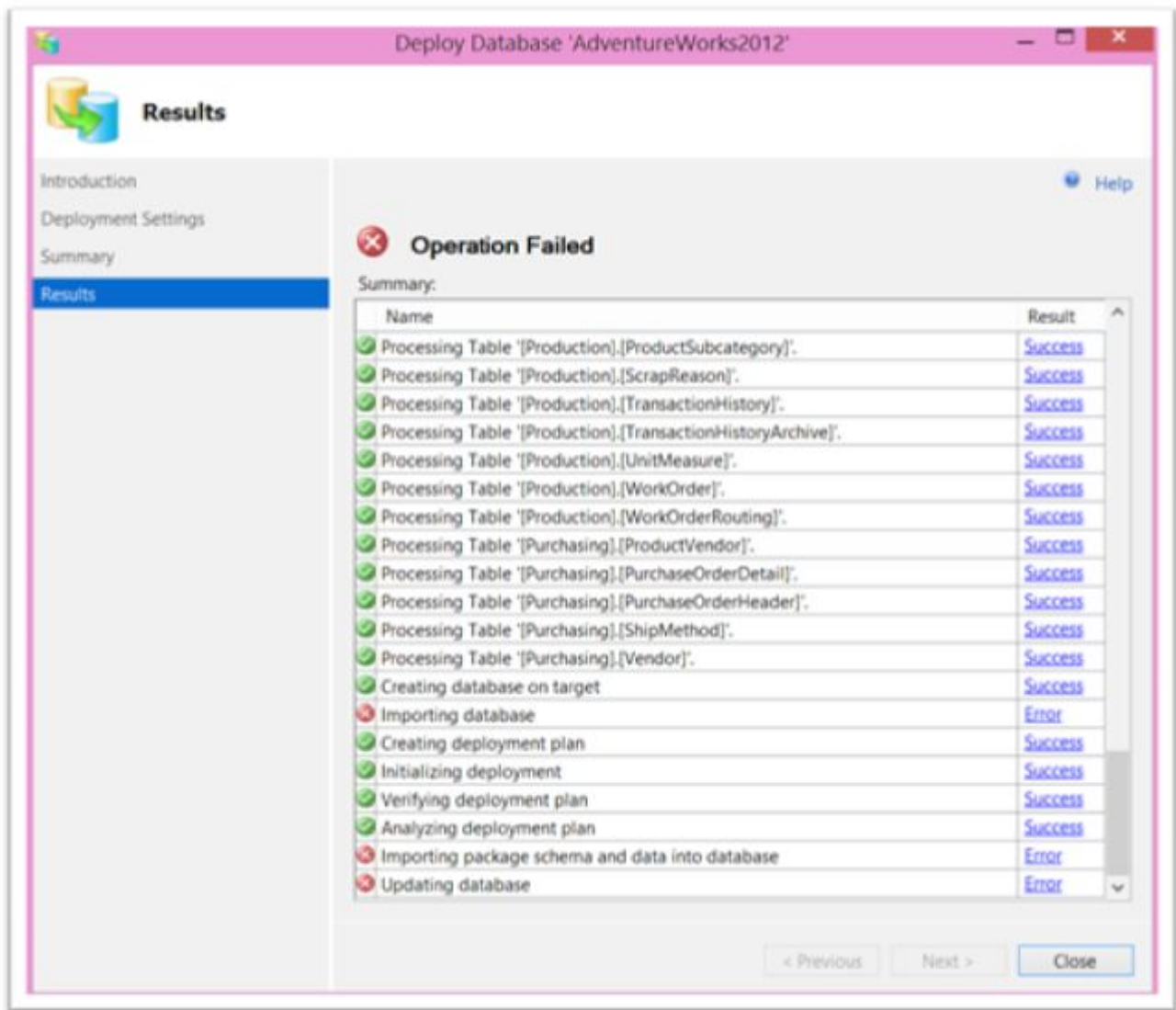
At the bottom right of the window, there are three buttons: "< Previous", "Next >", and "Cancel".

13. Click on **"Connect"** and specify the Azure Database Server that you want your on-premise database to be migrated to –



14. Run through the rest of the steps and click “finish” This will initiate data export.

15. Export then moves forward, but fails (in about 5 minutes) -



16. Click on the error message, and it reports following –



17. From the above error – we understand that there is a Stored Procedure within the on-premise database that is running “DBCC CHECKDB” , which isn’t supported in Azure SQL databases.

Summary

We discovered that the import/export wizard just fails when it finds an object/feature that isn’t supported in Azure SQL Databases. We will use “[SQL Database Migration Wizard](#)” to avoid such failures. This new wizard can read through the schema definition/trace file and report the features not supported in Azure SQL Databases. You can then make necessary changes and move the database to Azure.

Thank you for using C# Corner