Learn how to Migrate a SQL Server database to SQL Azure server

Problem

This tip intends to help those Microsoft data professionals that are moving to SQL Azure for the first time. You should use this article as a tutorial and follow it step-by-step. I will also try to share my experiences on the differences that I have found when I was moving from SQL Server to SQL Azure for the first time.

In my first article of this SQL Azure series, I wrote about how to create a SQL Azure database. In this article I am writing about how to migrate regular SQL Server databases to a SQL Azure server.

Solution

SQL Azure does not support BACKUP and RESTORE commands and neither CREATE DATABASE FOR ATTACH, so we cannot use these methods to import a SQL Server database into Microsoft Azure. The Copy Database Wizard also does not work for migrating databases from SQL Server to SQL Azure.

There are some possible methods, but the simpler options use bacpac files. A bacpac file is a compressed file that contains the database schema and the respective data.

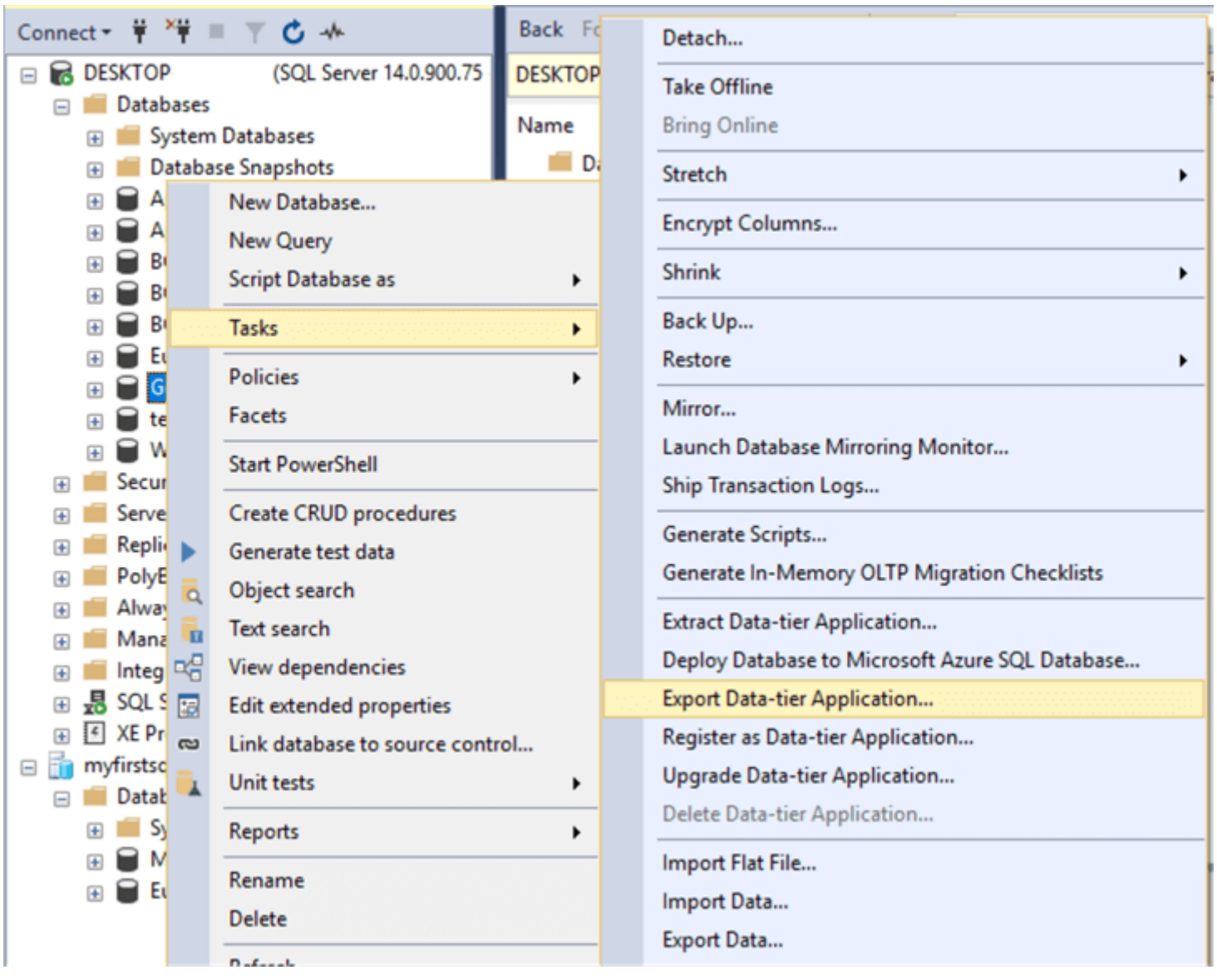
There are alternative methods that use migration tools or other SQL Server features such as Replication. If you want to know those methods, I added a link to the respective MSDN article at the end of this article as I will not explain them in this article.

Before continuing with this article, note that for a successful migration it is expected that everything is working properly after the migration, so be sure that the database that you are migrating is 100% compatible with SQL Azure. If you are not sure, before starting the migration, check in this MSDN article: T-SQL differences between SQL Server and SQL Azure. Also check the supported features in SQL Azure. And do not forget to perform as many tests as you can to assure a clean and error free migration process.

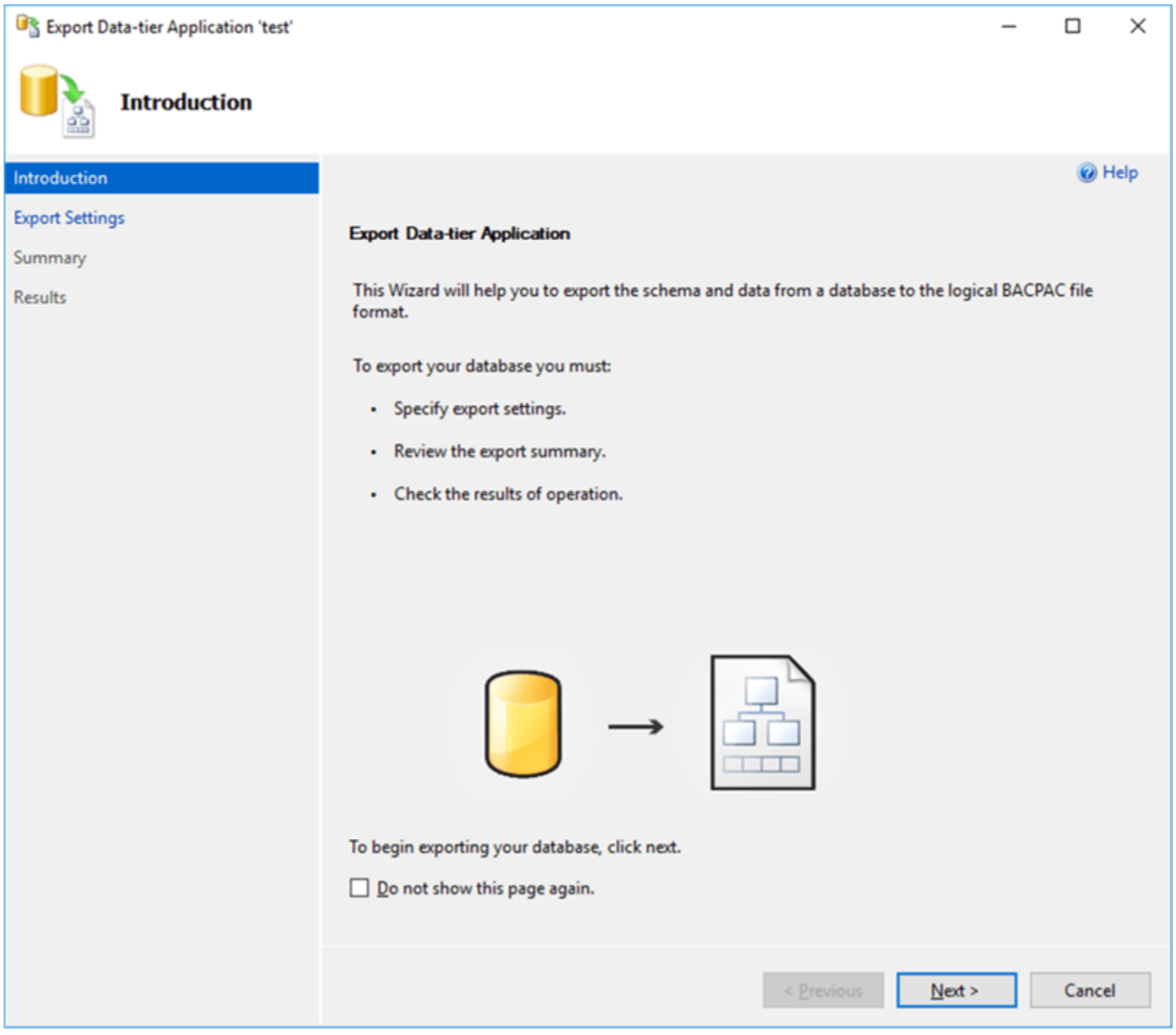
The migration process is basically an export and import of bacpac file. Let us see how to achieve those tasks.

Export to bacpac file in SQL Server Management Studio

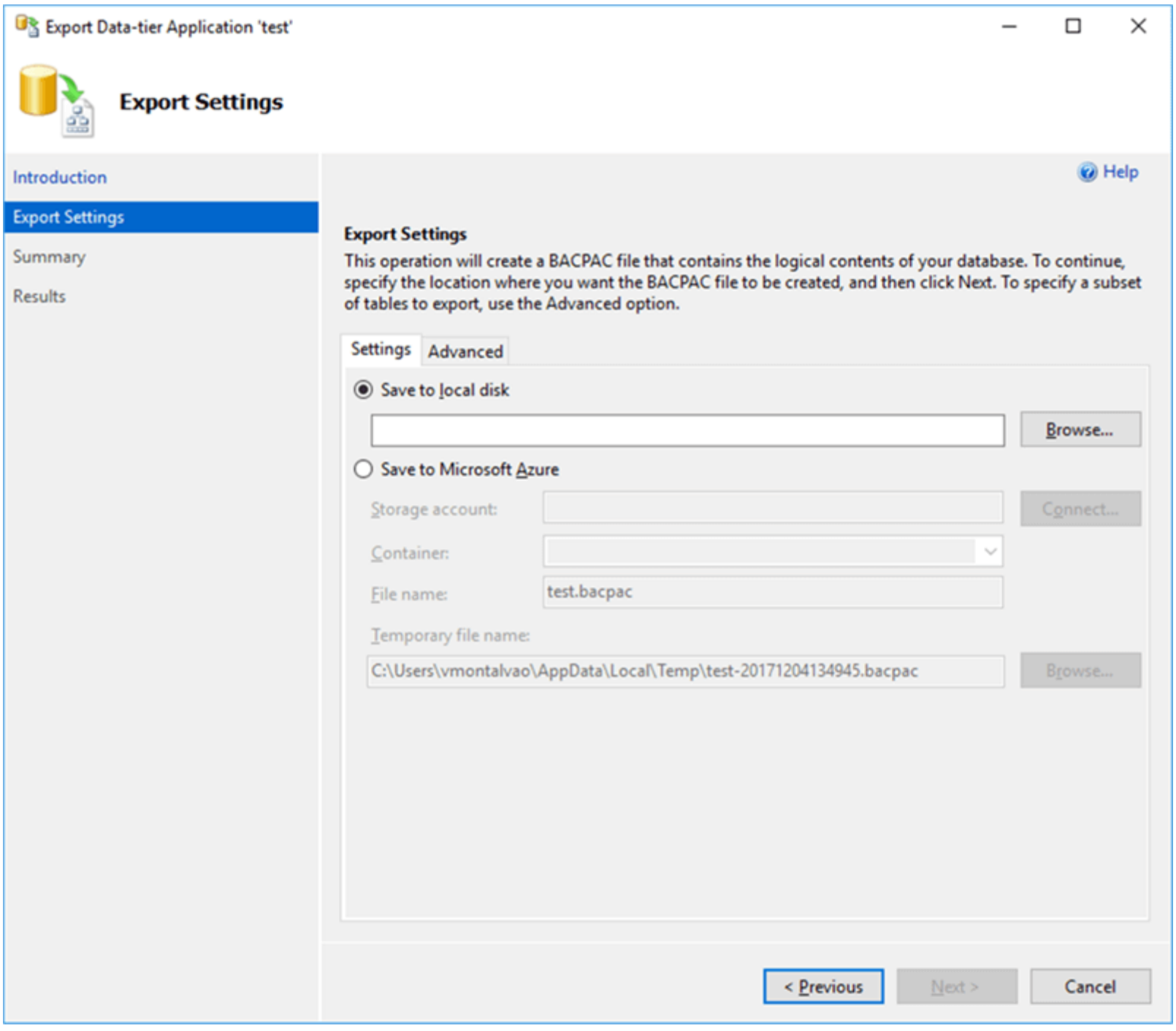
To export a database to a bacpac file in SQL Server Management Studio (SSMS), connect to the desired SQL Server instance and in the Object Explorer pane, right-click on the name of the desired database. In the context-menu select **Tasks > Export Data-tier Application**:



The above option will open the respective wizard:



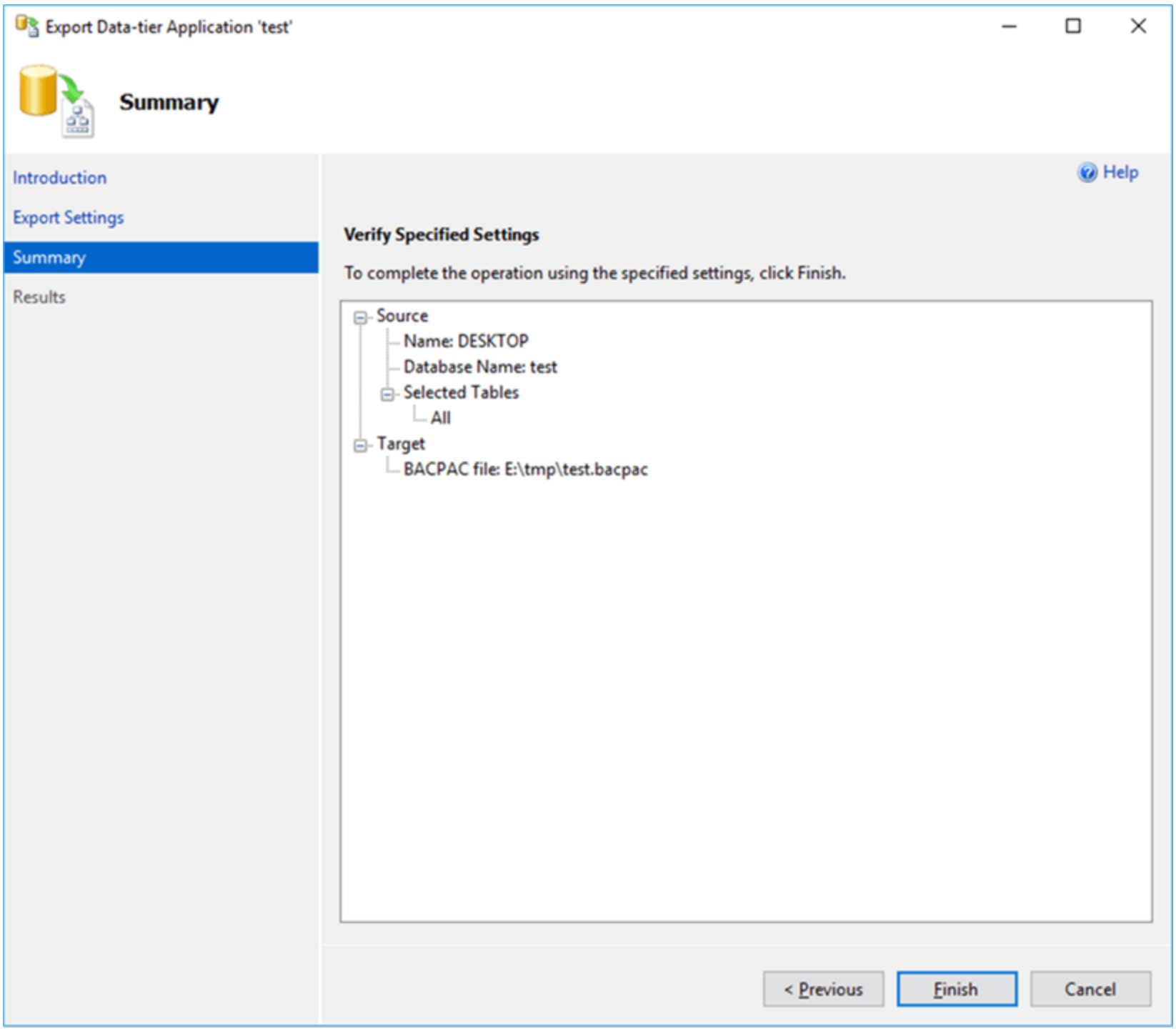
Click **Next >** in the Introduction page to go to the **Export Settings**:



In the **Export Settings** you are requested to provide the location to save the bacpac file. You can save it to the local disk or to an Azure storage account. Since I do not have an Azure storage account I will save the bacpac file to my local disk.

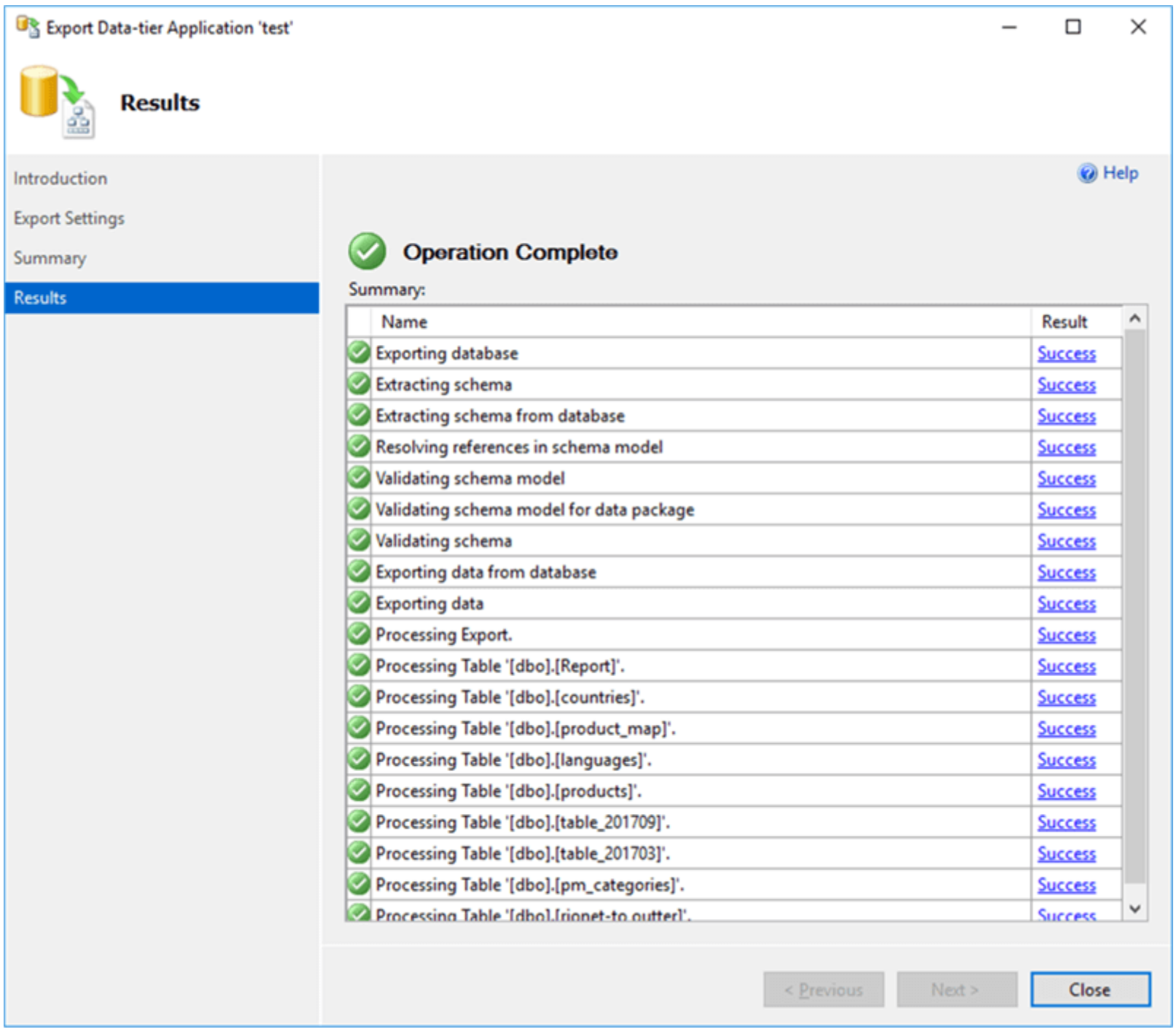
In the Advanced tab you can select which objects will be exported. By default, all objects are selected. Since this is what I want (migrate full database), I will go with the default and have all objects selected.

After providing the location and the bacpac file name, the **Next >** button will be enabled and then you can click on it to proceed. This will present the **Summary** page, where you can verify and confirm the provided settings:



After confirming that all provided information is correct, click on the **Finish** button to start the Export.

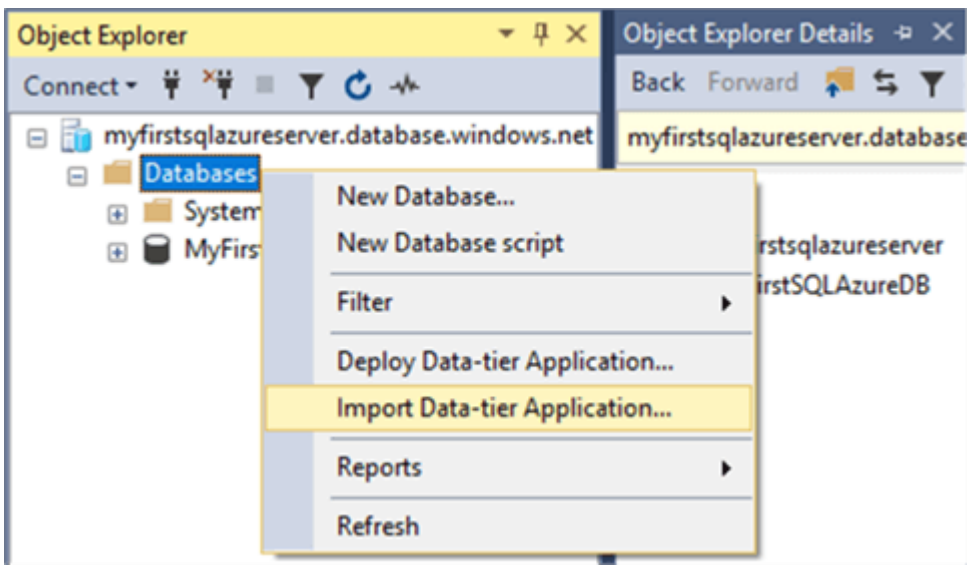
Errors will be shown if the export is unsuccessful and you will need to click on the respective error to find out what went wrong (usually errors occurs when unsupported elements or external references try to be exported). The following screen shows what a successful export looks like:



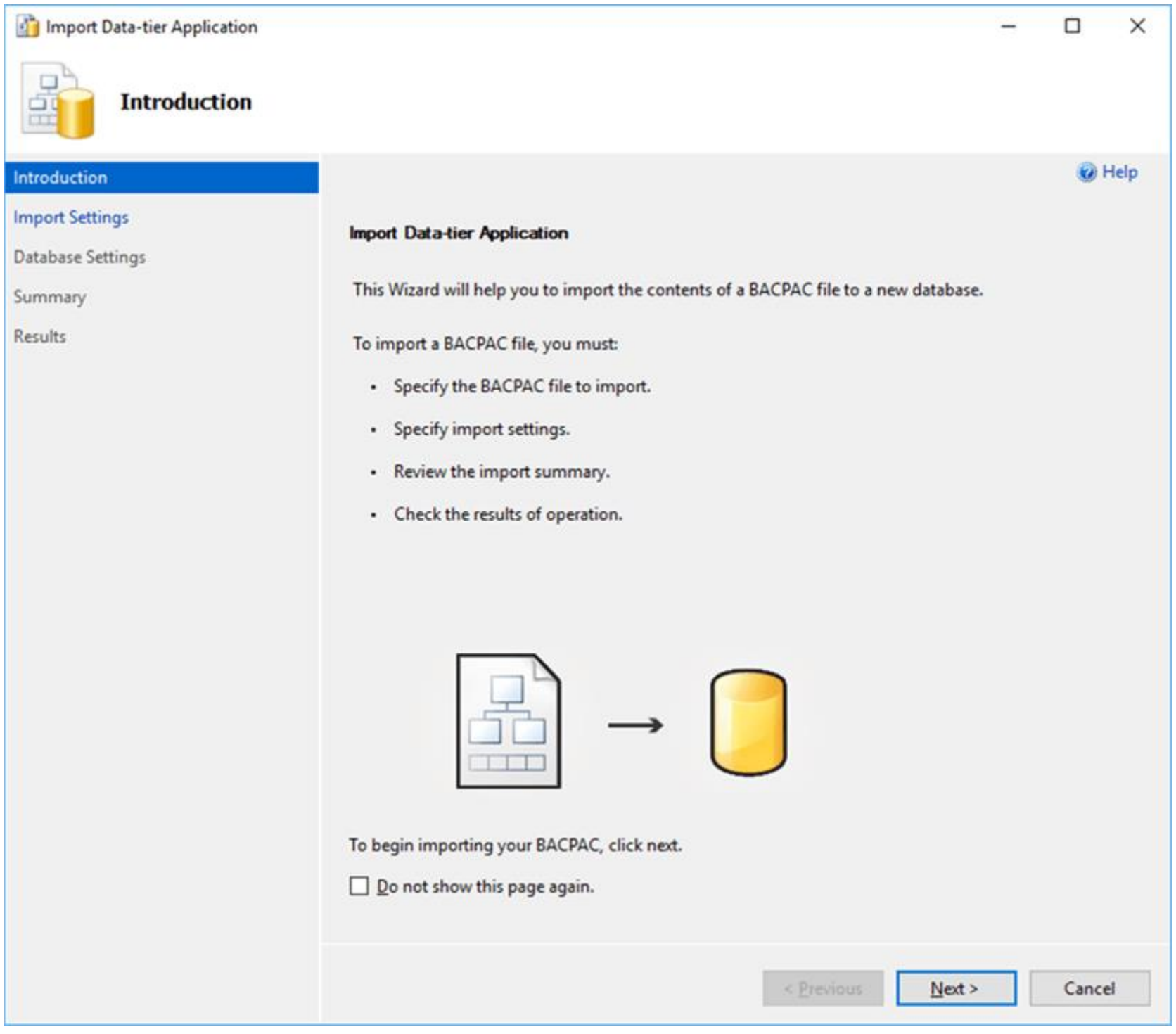
Click on the **Close** button to exit the BACPAC export wizard.

Import a bacpac file to SQL Azure

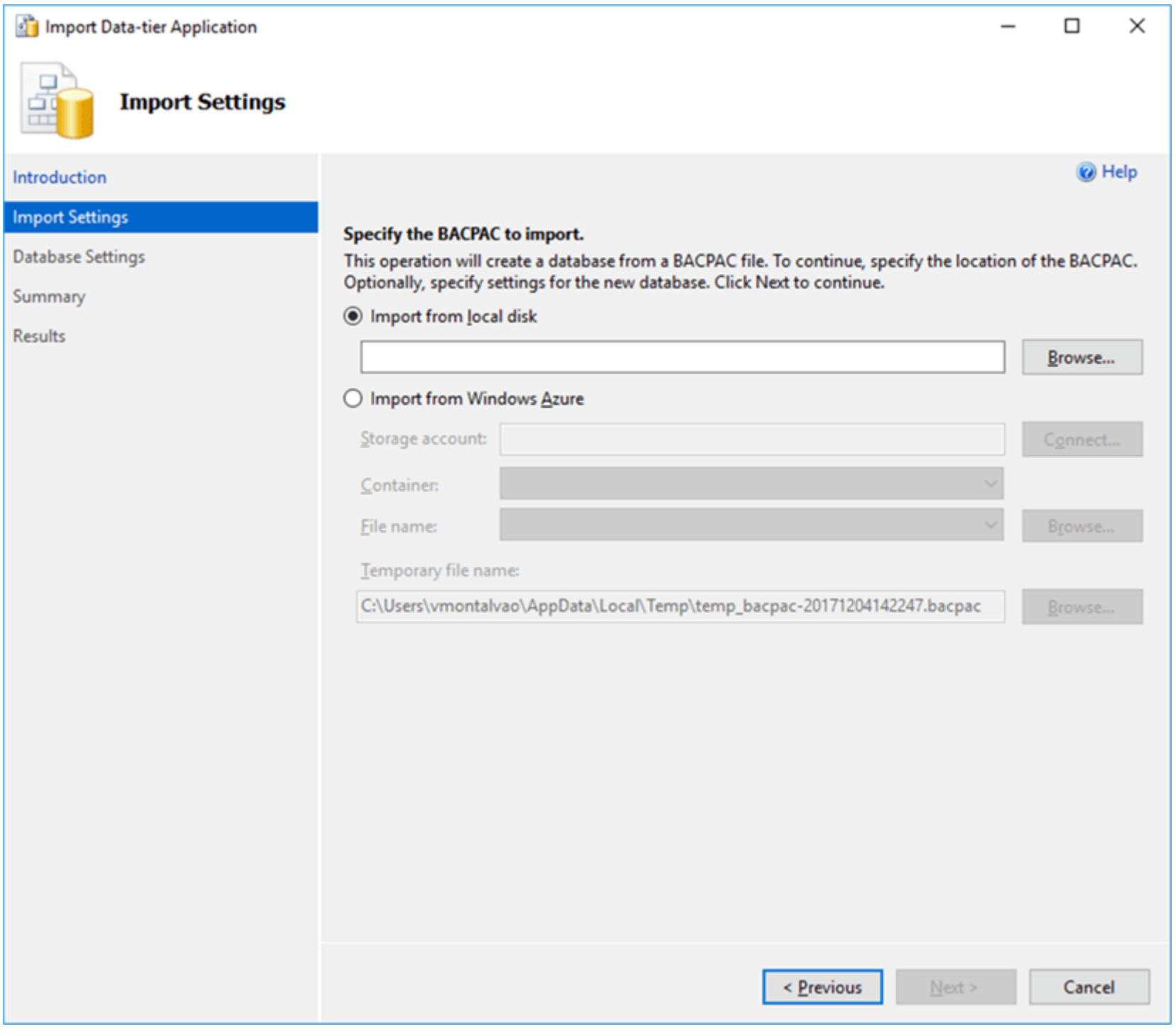
To create a SQL Azure database from a bacpac file in SQL Server Management Studio (SSMS), connect to the desired SQL Azure server and in the Object Explorer pane, right-click on the **Databases** folder. In the context-menu select **Import Data-tier Application**:



The above option will call the Import bacpac wizard and it will start by presenting the **Introduction** page:

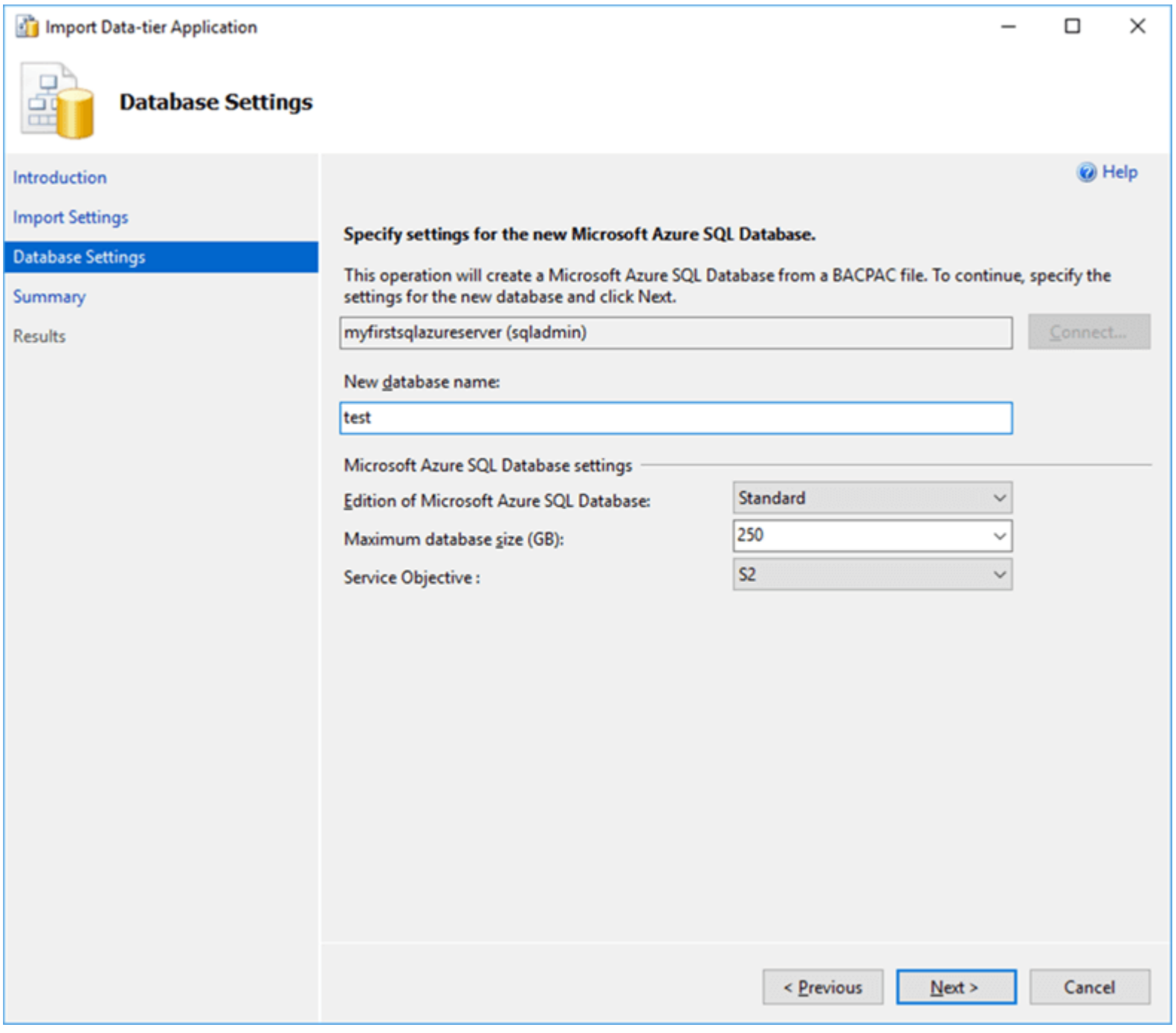


Click on the **Next >** button to provide the **Import Settings**:



In the **Import Settings** you are requested to provide the location of the bacpac file to import from. You can import from an existing bacpac file from the local disk or from an Azure storage account. Since I have exported the bacpac file to my local disk, I will import it from there.

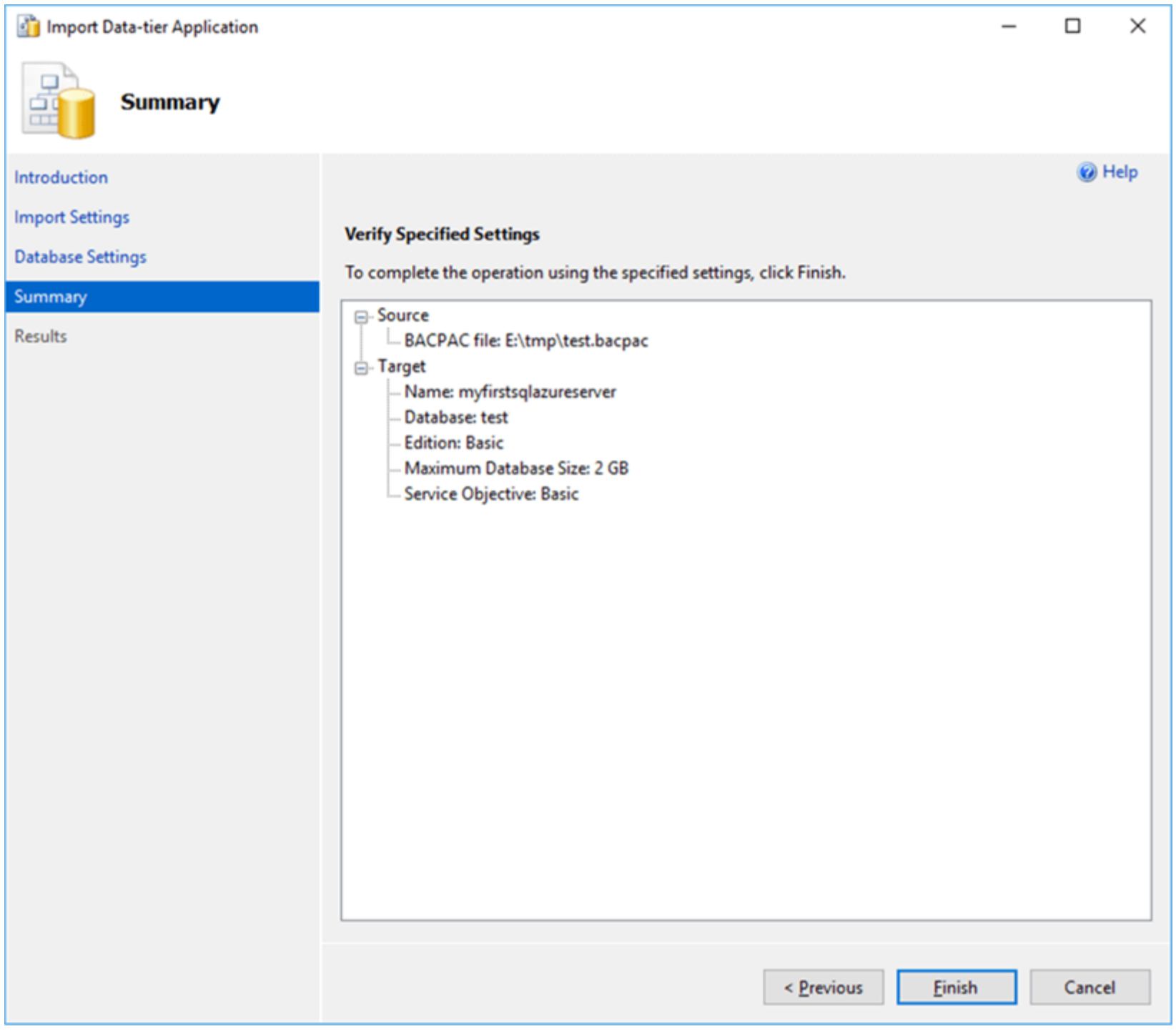
After providing the location and the bacpac file name, the **Next >** button will be enabled and then you can click on it to proceed. This will move to the **Database settings** screen:



In the **Database Settings** page, you are requested to configure the new SQL Azure database by providing the new database name (by default it will present the source database name) and the maximum database size (by default it will be configured with the maximum allowed size for the selected edition).

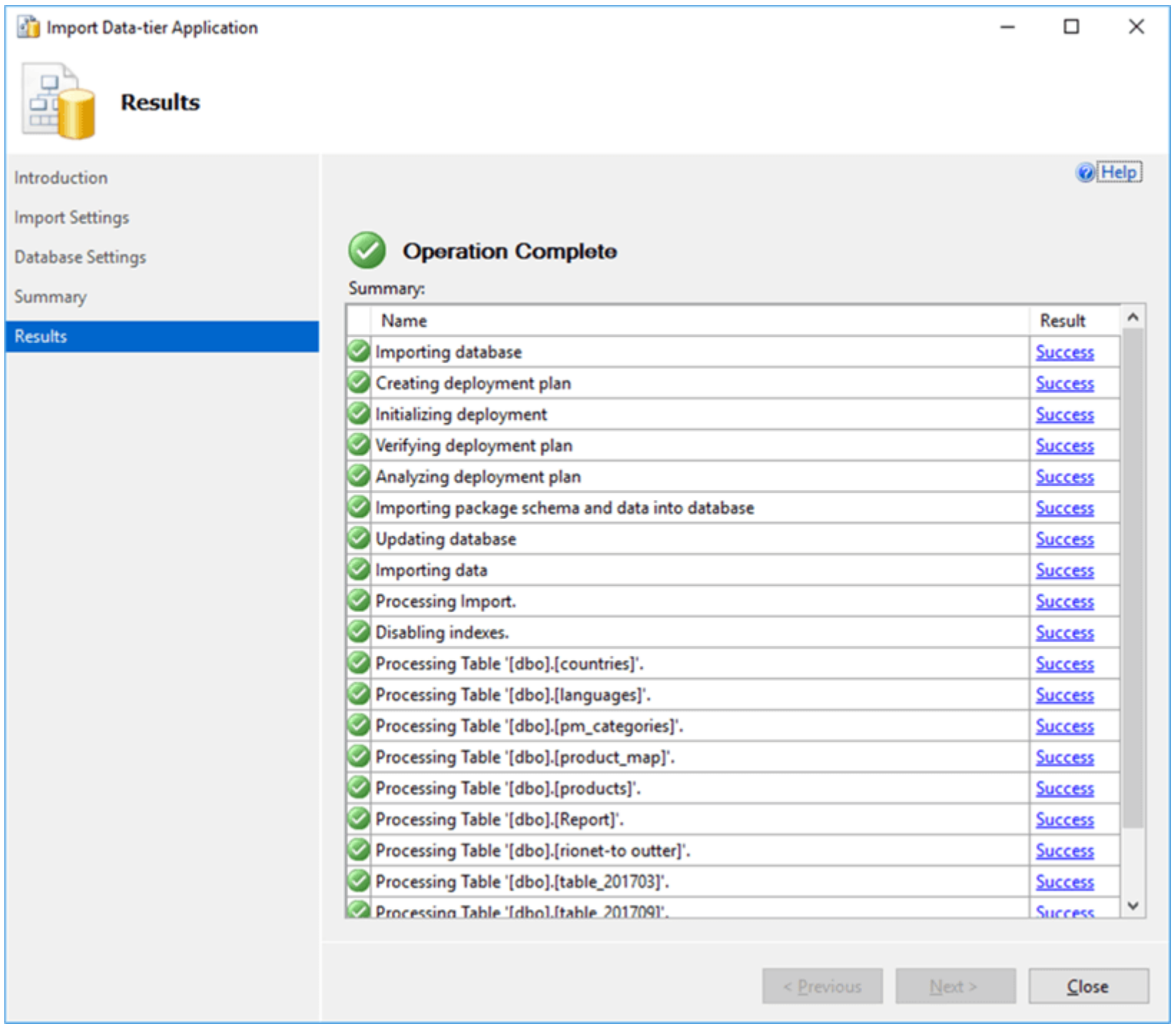
For this case I will keep the original database name, but will change to the Basic edition since the database is very small and I do not need to go for a more expensive edition.

Click on the **Next >** button to proceed to the **Summary** page:



In the **Summary** page, confirm that you provided the correct information. As you can see by the above screenshot, I have changed the edition to Basic. After confirming, click on the **Finish** button to start the Import.

An error will be shown if there is an unsuccessful import and you will need to click on the respective error to find out what went wrong. The following screen shot shows how a successful import should look:

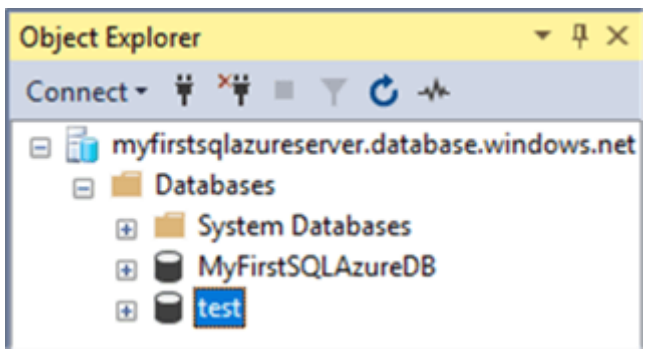


Click on the **Close** button to exit the BACPAC import wizard.

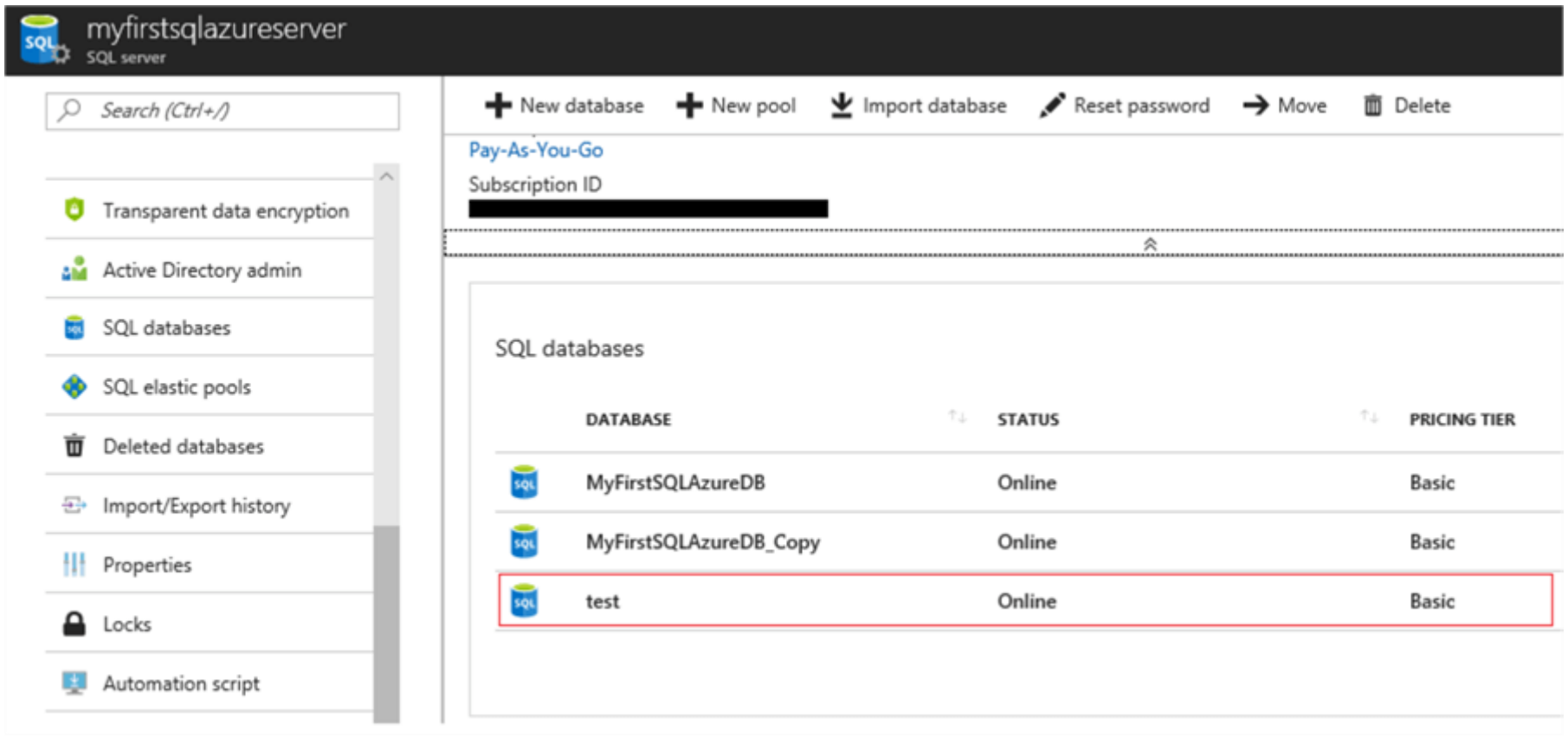
Validate SQL Azure Import

You can confirm that the new database has been created in the SQL Azure server.

In SSMS navigate to the instance:



And in the Microsoft Azure Portal by checking in the respective SQL Azure server:



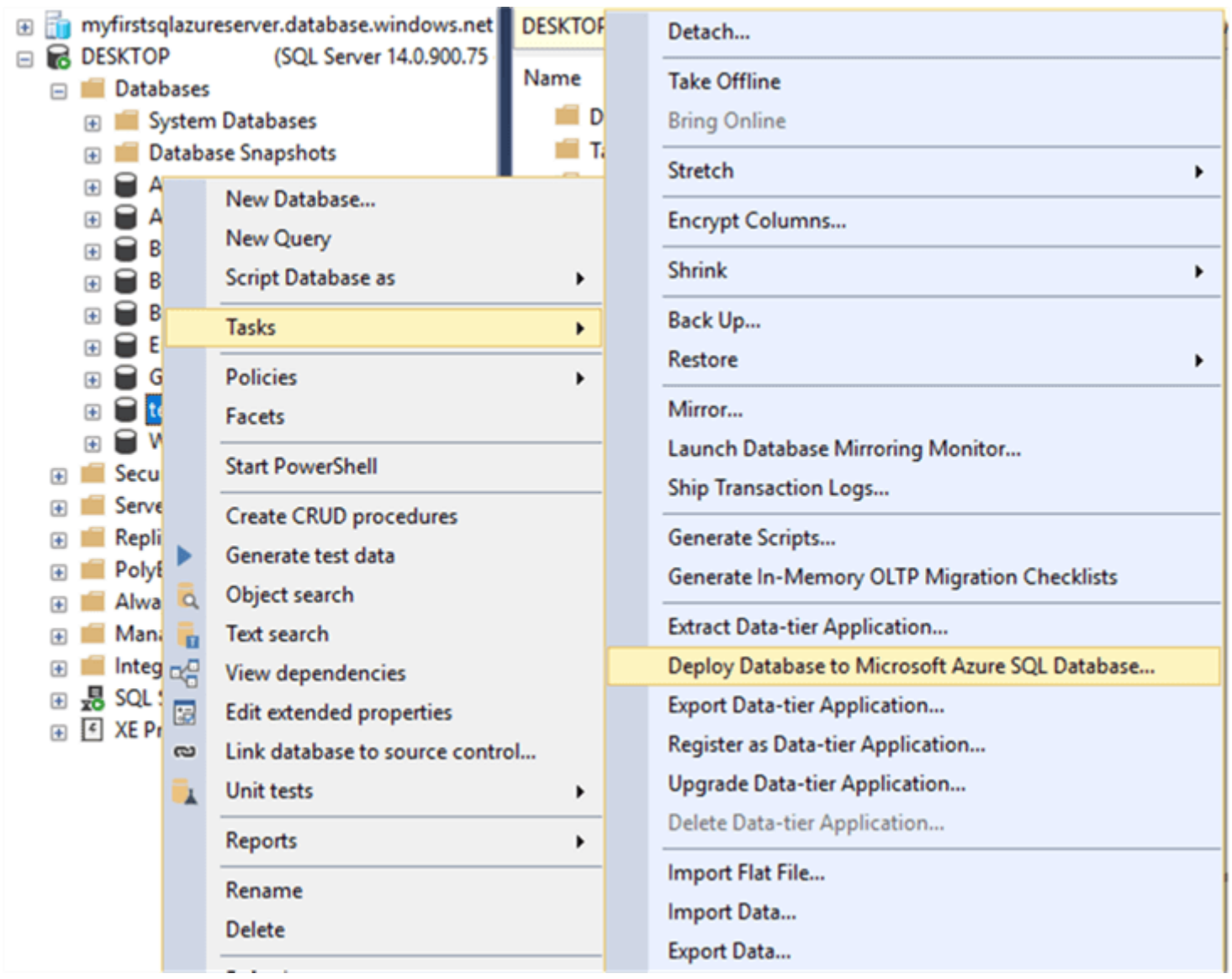
Deploy database to Azure

I have showed you how to migrate a SQL Server database to SQL Azure server by exporting and importing a bacpac file.

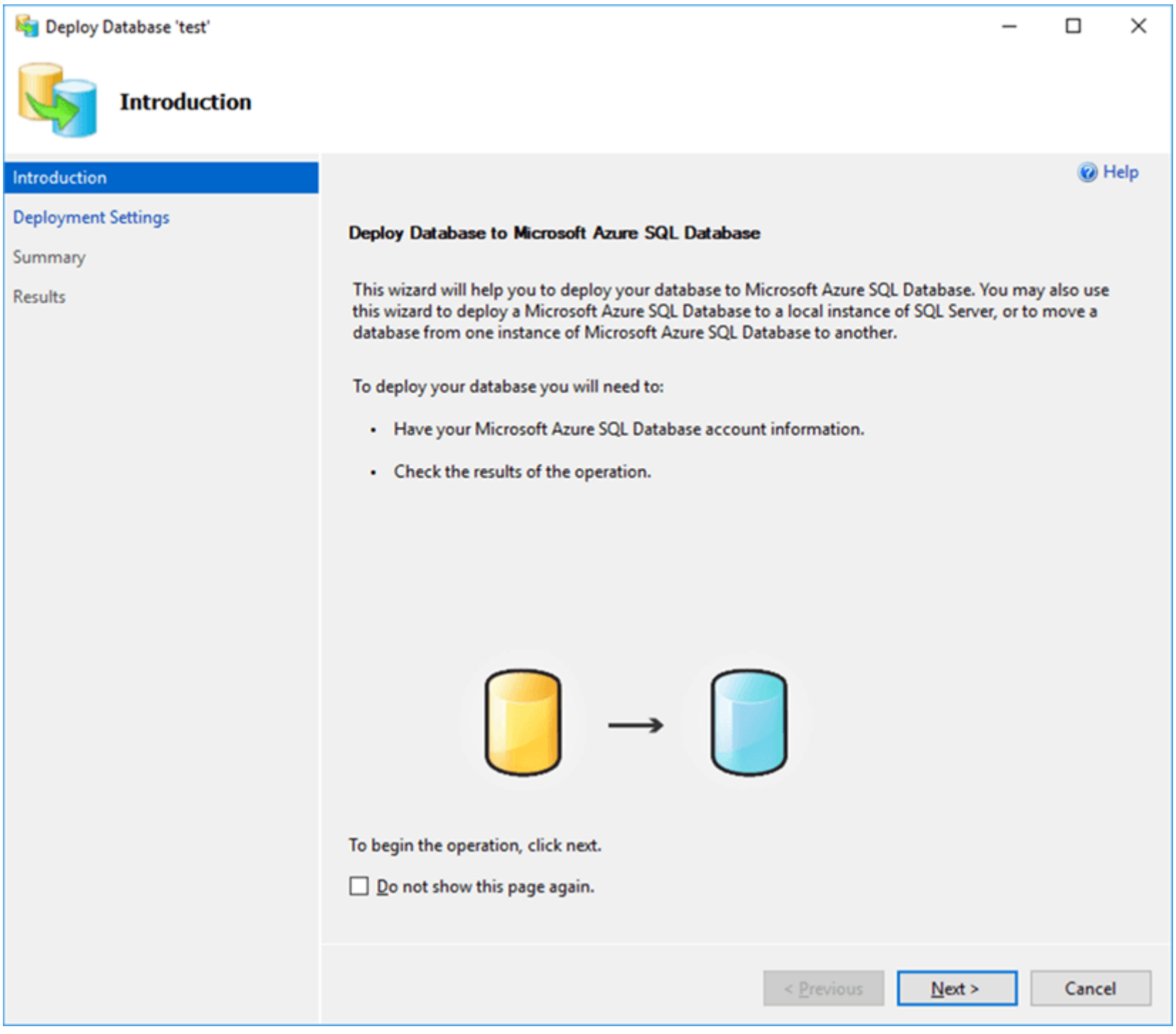
I want to show you each option isolated so you can understand them better and depending on what your needs are you can choose to go for a step-by-step process (for example for later import or to provide the bacpac file to another entity) or go for an immediate migration.

Now I will show you how to do the same but in a single run, i.e. the export and import will happen immediately one after another without human intervention.

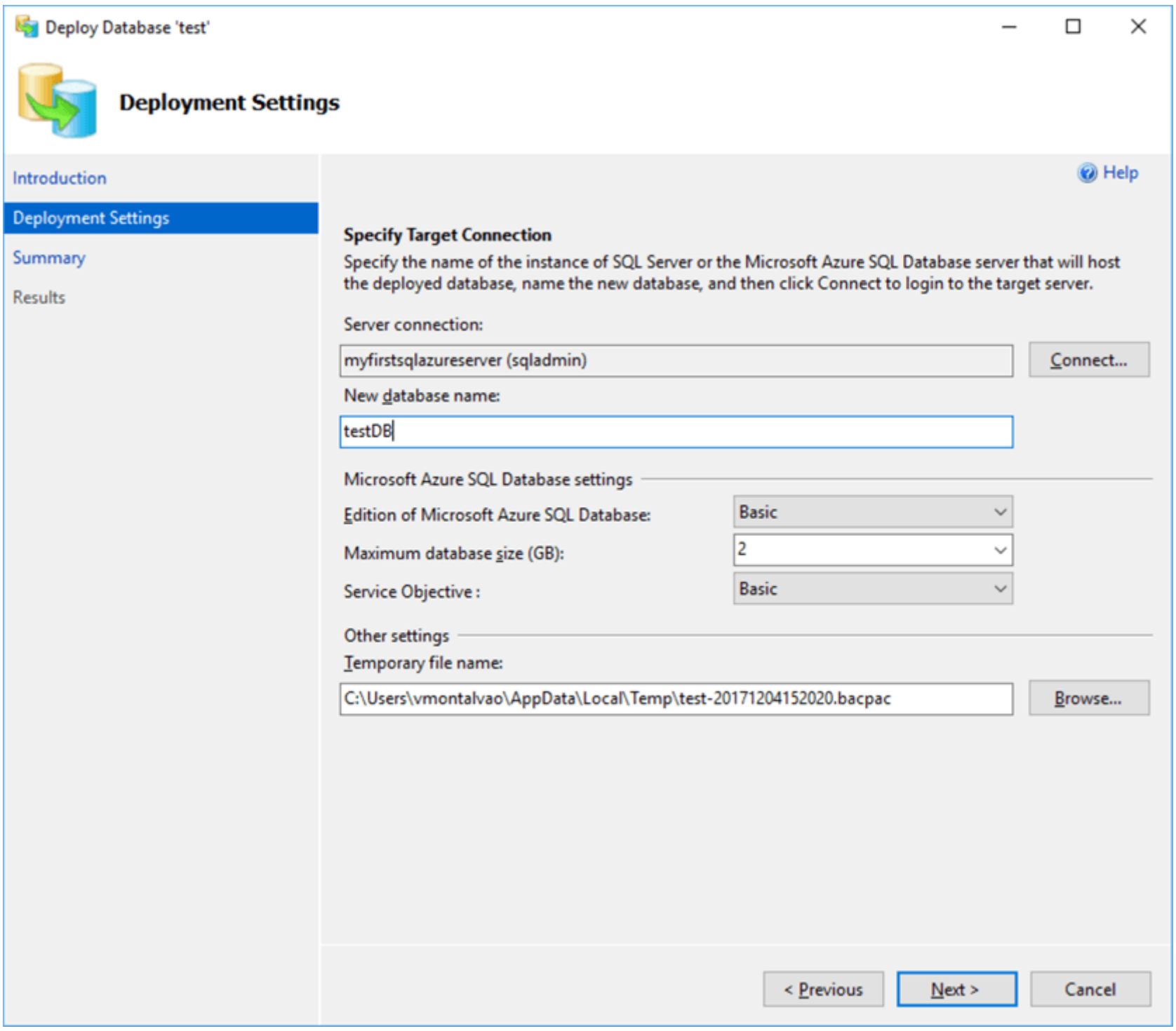
Deploying a SQL Server database to a Microsoft Azure SQL database, can be achieved in SSMS by right-clicking in the database name that we want to migrate and then choose the following options from the context-menu: **Tasks > Deploy Database to Microsoft Azure SQL Database.**



This will start the Deployment wizard by presenting the **Introduction** page:



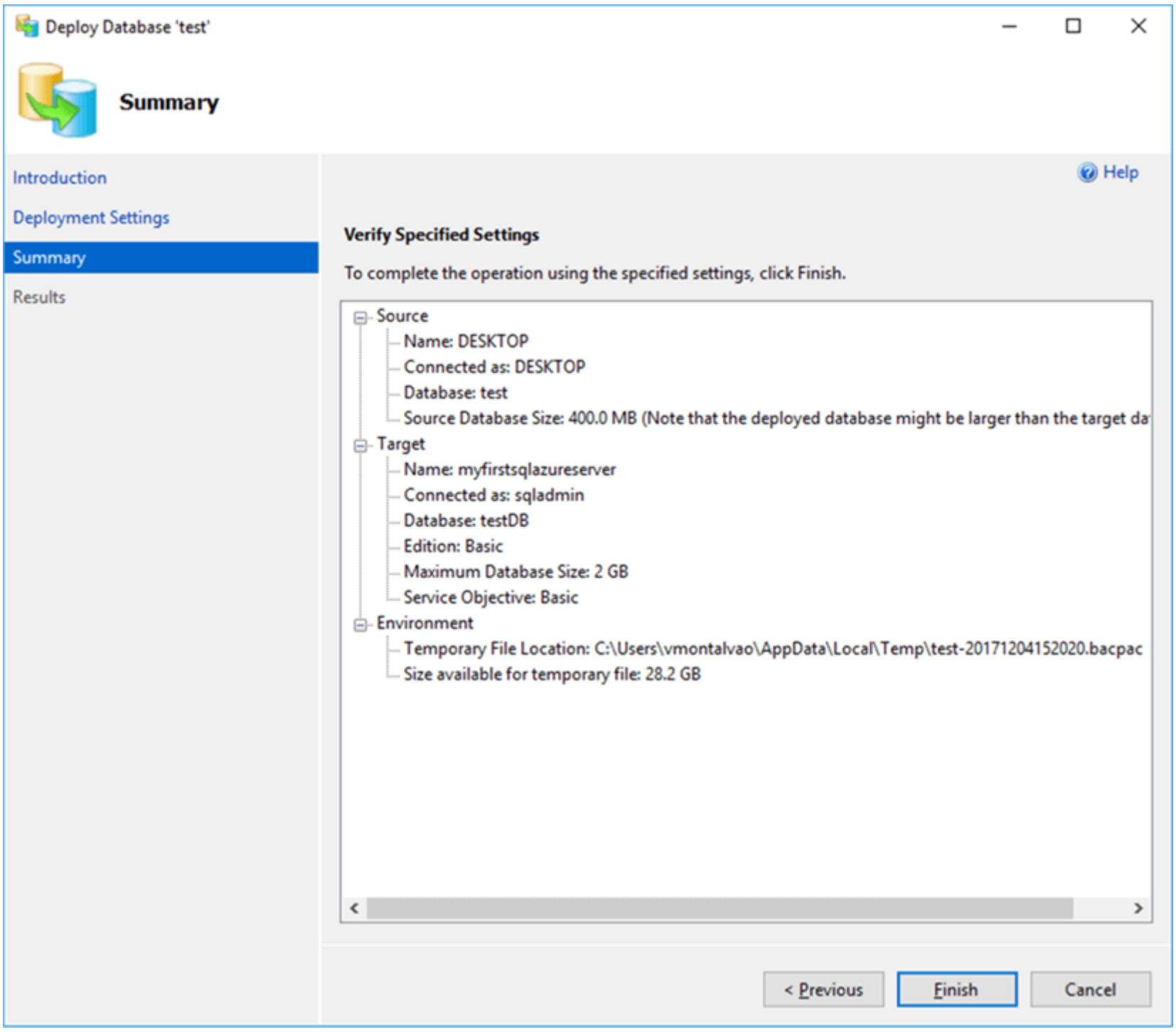
Click on the **Next >** button to provide the **Deployment Settings**:



Click on the **Connect** button to provide the credentials to connect to your SQL Azure server. Then provide the new database name. I will use the same test database, but need to provide a new name since **test** was been used before during the previous import. I will call this new database **testDB** as I cannot use an existing database name.

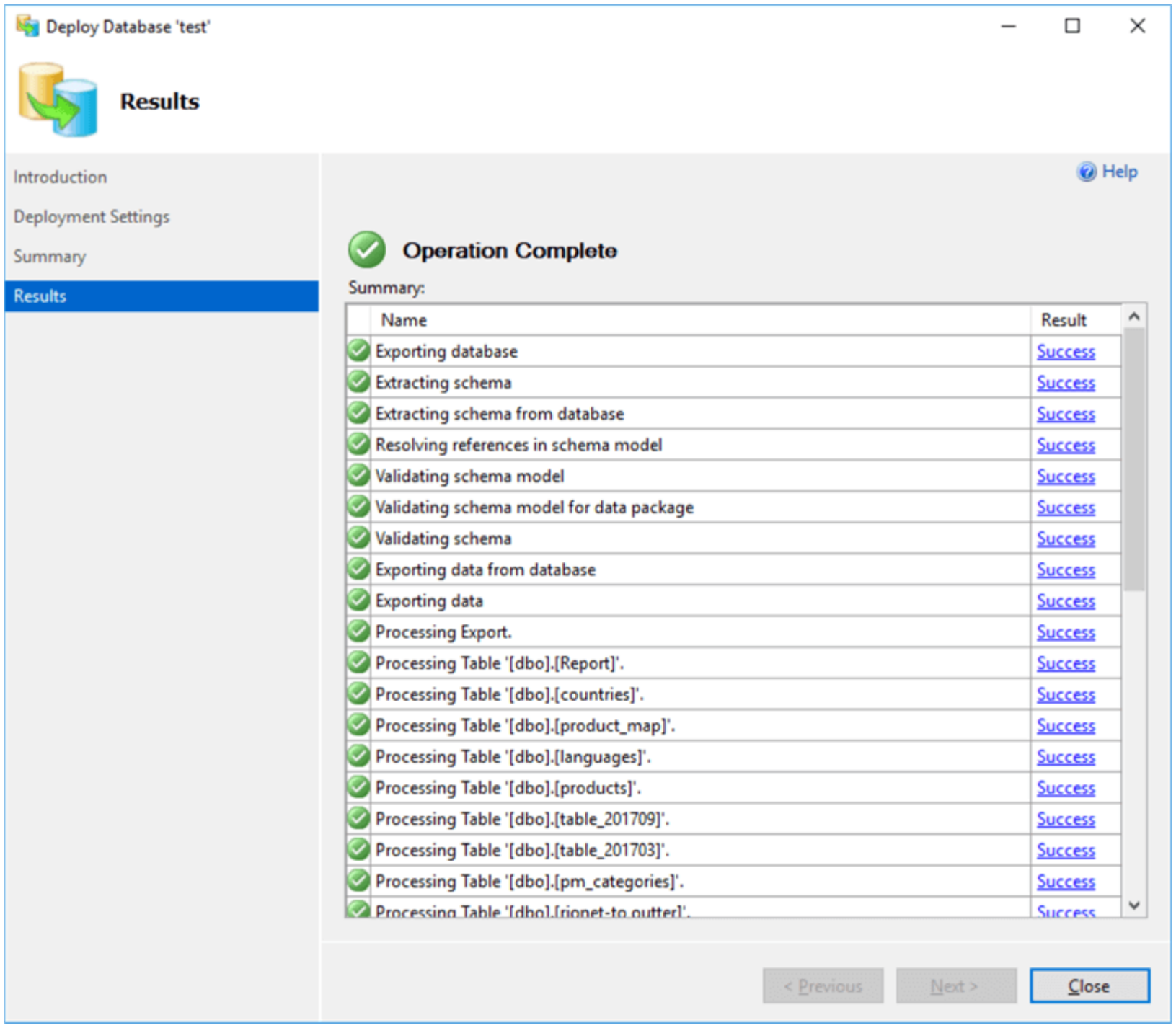
You can use the suggested temporary file name and path or provide a new path and/or file name. In my case I will just accept the suggested one.

Click on the **Next >** button when you have provided all the necessary information for the deployment. Then confirm on the Summary page if the provided information is correct:



If it is all correct, click the **Finish** button to start the deployment of the database.

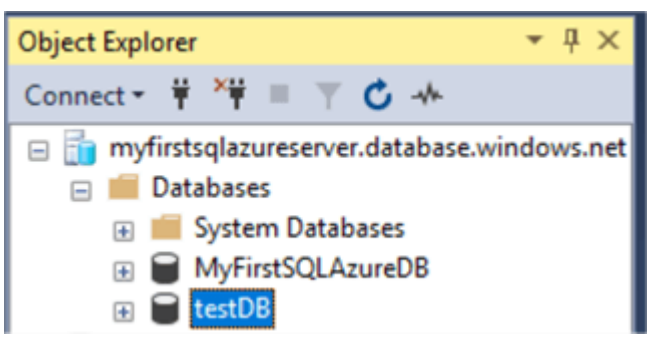
The deployment will show first the Export and then the Import processes. It will stop if any errors occur during these processes. When completed successfully, a screen similar to the following will be shown:



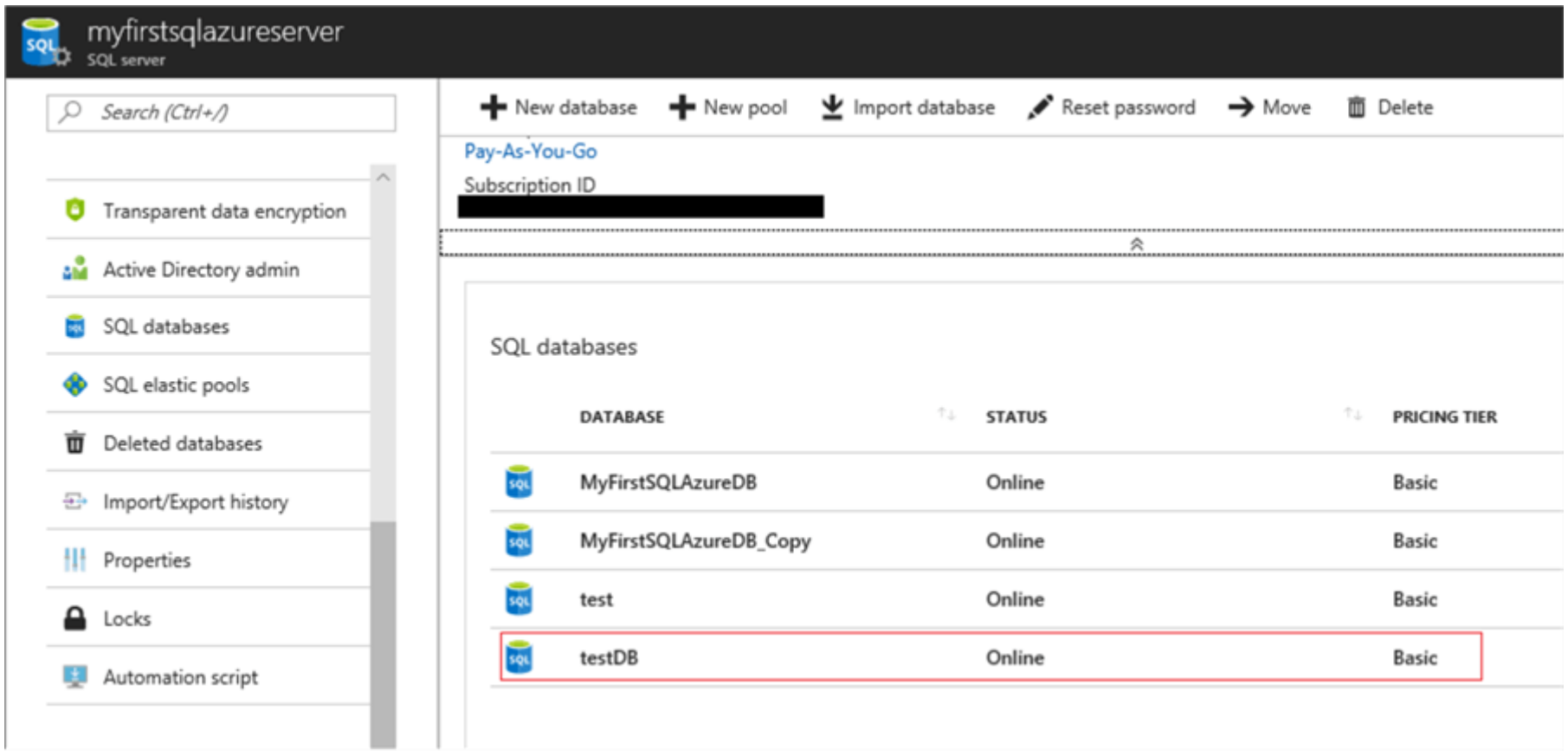
Click on the **Close** button to exit the Deploy database wizard.

Validate the New SQL Azure Database is Online

You can confirm now that the new database has been created in the SQL Azure server. In SSMS navigate to the SQL Azure instance:



And in the Microsoft Azure Portal validate by checking in the respective SQL Azure server:



When talking about the topic of cloud computing services, you must have heard of these concepts, ***Infrastructure as a Service (IaaS)***, ***Platform as a Service (PaaS)***, and ***Software as a Service (SaaS)***. They are the three major service categories provided by cloud providers. It is important to understand what they are and what the difference between them is because they will appear on various occasions where cloud computing services are discussed.

On-premises (Private Cloud)	Infrastructure (as a Service)	Plat (as a S
You Manage	**You Manage**	**Cloud Pr Mana**
Data & Access	**Data & Access**	**Data &**
Applications	**Applications**	**Appli**
Runtime	**Runtime**	**Run**
Operating System	**Operating System**	**Operatin**
Virtual Machine	**Virtual Machine**	**Virtual**
Compute	**Compute**	**Com**

Infrastructure as a Service (IaaS)

As you can see in the image above, IaaS is the most flexible category of cloud services. Instead of buying hardware, with IaaS, you **rent** IT infrastructure servers and virtual machines (VMs), storage, networks, and operating systems from Microsoft on a **pay-as-you-go** basis, and you are responsible for managing the operating systems, data, and applications.

Therefore, **Azure virtual machines** are Infrastructure as a Service (IaaS) and **Azure page blobs** are the backbone of the virtual disks platform for Azure IaaS.

Platform as a Service (PaaS)

As you can see in the image above, a PaaS solution requires less user management and does not provide access to the operating system. That means that the PaaS is a complete development and deployment environment in the cloud and provides a framework that developers can build upon to develop or customize cloud-based applications.

For example, the **Azure Web Apps service** provides an environment for you to host your web applications but you don't have to access the virtual machine and the operating system. And **Azure SQL Database** is a fully managed platform as a

service (PaaS) database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring without user involvement.

Software as a Service (SaaS)

A SaaS solution requires the least management. Microsoft is responsible for managing everything, and you just use the software. SaaS allows you to connect to and use cloud-based apps over the Internet. When you are implementing a SaaS solution, you are responsible for configuring the SaaS solution. Common examples are Outlook email, calendar, and office tools (such as Microsoft Office 365).

Summary

And as a summary, here is a table that shows several examples of Azure services belonging to these three categories.

Examples of Azure Service Typ	
IaaS	PaaS
Azure virtual machines	Azure App Service
Azure Storage accounts	Azure SQL database
	Azure Cosmos DB
	Azure Synapse Analyt

what is BLOB storage in Azure

Blob storage is service which storing the large amounts of unstructured data object like Text, Binary data, image, video, audio and also it store data backup, restoring, and disaster recovery. Blob storage is exposing the data publically and privately. For public usage there is no need of additional configuration. But for private usage we must configure the shared key or shared access signature (SAS)

How a shared access signature works

A shared access signature is a signed URI that points to one or more storage resources. The URI includes a token that contains a special set of query parameters.

When to use shared access signature

Use a SAS to give secure access to resources in your storage account to any client who does not otherwise have permissions to those resources. A common scenario where a SAS is useful is a service where users read and write their own data to your storage account.

What is Azure data lake

Microsoft Azure Data Lake is part of the Microsoft Azure public cloud platform, which includes more than 200 products and cloud services. Azure Data Lake is a cloud platform designed to support big data analytics and it provides unlimited storage for structured, semi-jhjhjhru8 l data. It can be used to store any type of data of any size. Azure Data Lake is built on Azure Blob storage, which is the Microsoft object storage solution for the cloud. The solution Mati features low-cost, tiered storage and high-availability/disaster recovery capabilities.

WHAT IS DEFFRENCE BT SQL AZURE VM AND SQL AZURE DATABASE

In SQL Server Azure VM we need a specific version of sql server or windows but in Azure sql database we don't need a specific version of server or windows.

In sql server Azure VM we need instance level features like Agent jobs, linked servers but in Azure database we don't need instance level features.

In sql Azure VM requires configuration but in Azure database we don't need configuration and manage server or windows.

Great for migrating existing apps but it great for new apps in Azure database.

Deployment models for on perm to Azure

Azure sql server database provides three deployment options:

1. Single database: is a fully managed and isolated database
2. Elastic Pool: is a collection of single database with a shared set of resource
3. Managed instance: is a fully managed instance of the sql server