

Many options are available for loading data into Azure SQL. Some overlap with what's available on-premises.

This unit calls out a few.

## Bulk Copy Program

Bulk Copy Program (bcp) is a common tool for connecting to Azure SQL from on-premises, and for connecting to Azure SQL from an Azure virtual machine. You can then use it to move data into Azure SQL.

## Bulk insert

Bulk insert operations are similar to what's available in SQL Server on-premises, but instead of loading data from a file (or multiple files) on your machine, you load data from Azure Blob storage. The next exercise will walk through an example of how.

## SSIS packages

In Azure SQL, you can use packages to connect with SQL Server Integration Services (SSIS) on-premises. You can host an SSIS database in Azure SQL Database or Azure SQL Managed Instance. Additionally, you can use the Azure-SSIS Integration Runtime for SSIS packages with tools like Azure Data Factory.

## Other options

Other interesting options include using technologies like Spark or Azure Data Factory to load data into Azure SQL Database or Azure SQL Managed Instance. Though not directly related to "loading," it's possible to create a database by using an existing database for a copy or doing an import of a BACPAC file. In Azure SQL Managed Instance, you can use T-SQL commands to restore a database natively from a URL.

## Considerations for loading data

The biggest difference between loading data on-premises and loading data into Azure SQL is that the data you want to load needs to be hosted in Azure. The data can't be hosted in files on-premises. You can store the on-premises files and file systems in Azure Blob storage as an alternative. This strategy will also increase your efficiency in loading your files in and setting up ETL jobs.

Also keep in mind that minimal logging isn't supported, so you're always running in full recovery mode. Because of full recovery mode and limits around log throughput, you might be affected by log governance as you're loading data. Techniques like using batches and appropriately sizing them become important during a bulk load. You'll get more tips in the following exercise. Loading into a clustered columnstore index might help in avoiding transaction log limits, depending on your scenario.

## Knowledge check

1. Which Azure SQL deployment options support natively restoring a database? \*

- ☐ SQL Server on an Azure VM
- ☐ Azure SQL Managed Instance
- ☐ Azure SQL Database
- ☒ SQL Server on an Azure VM and Azure SQL Managed Instance

✓ In an Azure VM, it's just SQL Server, so it's supported. In Azure SQL Managed Instance, native restore from a URL is supported.

2. What isn't an example of how you can more efficiently load data into Azure SQL Database and Azure SQL Managed Instance? \*

- ☐ Use clustered columnstore indexes
- ☐ Use batching with appropriately sized batches
- ☒ Use simple recovery mode

✓ This capability isn't currently supported in Azure SQL Database or Azure SQL Managed Instance. Full recovery mode is required to meet the SLA.

☐ All of the above will help in Azure SQL Database and Azure SQL Managed Instance

Next unit: Exercise - Load data into Azure SQL Database

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆