**Summary**

Nice job! Your pipeline is becoming a useful part of your process. You and the Tailspin team now have a pipeline that includes consideration for database schema changes. You used Azure SQL Database here, but you could have used any database technology.

In this module, you learned what to consider when you add a database to a web app and ultimately to Azure Pipelines. You learned:

* How to access an Azure SQL Database from Azure App Service.
* How to add database schema changes to Azure Pipelines approvals.
* How to make sure that an approval is required only when the schema changes.

The team successfully designed and built a pipeline that manages the database changes as well as the application changes.

Although the team is improving their release cadence, there's room for more improvements. For example, Amita from QA must manually test and approve builds before the team can present new features to management. In the next module, you'll work with the Tailspin team to automate additional testing so that changes can move through the pipeline even faster.

**Learn more**

The team decided to use Azure SQL Database as the data storage technology. The data storage technology you use depends on the type of data you store. Check out these resources to choose a data storage technology for your projects:

* [Choose a data storage approach in Azure](https://docs.microsoft.com/en-us/learn/modules/choose-storage-approach-in-azure/)
* [SQL Server Data Tools](https://docs.microsoft.com/en-us/sql/ssdt/sql-server-data-tools?view=sql-server-2017)

In this module, you worked with ASP.NET Core, SQL Server Data Tools, and Azure SQL Database. You passed Azure Pipelines variables between stages and configured App Service. Use the following resources to learn more:

* [Safe storage of app secrets in development in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/security/app-secrets?view=aspnetcore-3.0&tabs=windows)
* [The Azure SQL Database Deployment task](https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/deploy/sql-azure-dacpac-deployment?view=azure-devops)
* [Configure an App Service app in the Azure portal](https://docs.microsoft.com/en-us/azure/app-service/configure-common)
* [Pass a variable from stage to stage in Azure DevOps](http://donovanbrown.com/post/Passing-variables-from-stage-to-stage-in-Azure-DevOps-release) (This link leads to an external website.)
* [REST APIs for Azure DevOps Services](https://docs.microsoft.com/en-us/rest/api/azure/devops/search/?view=azure-devops-rest-5.1)

This module mentions the following Microsoft Learn modules:

* [Develop and configure an ASP.NET application that queries an Azure SQL database](https://docs.microsoft.com/en-us/learn/modules/develop-app-that-queries-azure-sql/)
* [Choose a data storage approach in Azure](https://docs.microsoft.com/en-us/learn/modules/choose-storage-approach-in-azure/)

You learned to use *dacpac* and *bacpac* files in this module. For more information about those files and data-tier applications, see [Data-tier applications](https://docs.microsoft.com/en-us/sql/relational-databases/data-tier-applications/data-tier-applications?view=sql-server-ver15).

You needed a blob storage container to hold your *bacpac* file. These Microsoft Learn modules provide more information about working with storage on Azure:

* [Create an Azure Storage account](https://docs.microsoft.com/en-us/learn/modules/create-azure-storage-account/)
* [Secure your Azure Storage account](https://docs.microsoft.com/en-us/learn/modules/secure-azure-storage-account/)