**Summary**

* 1 minute

Great work! In this module, you updated a release pipeline to build, publish, and deploy the team's multi-container solution to Azure Kubernetes Service. The team is now well on their way to implementing a broader microservices strategy that will offer increased agility in their development and operations.

While this module specifically focused on orchestrating a collection of Docker containers using Kubernetes, the fundamentals covered apply across a wide range of build and deployment targets. You can also see how Azure Pipelines can scale to support even the most sophisticated scenarios.

**Learn more**

In this module, you worked with a microservices solution using containers. However, there are many ways to implement microservice solutions in Azure using a wide variety of technologies. To learn more, see [Building microservices on Azure](https://docs.microsoft.com/en-us/azure/architecture/microservices/).

The microservices solution in this module had been refactored from a larger monolithic application. To learn more about migrating larger applications into smaller services, see [Decompose a monolithic application into a microservices architecture](https://docs.microsoft.com/en-us/learn/modules/microservices-architecture/).

The microservices solution in this module used a variety of platforms and services, including Docker, Azure Container Registry, and Kubernetes. To learn more about these and related container-centric technologies in Azure, such as Azure Container Instances and App Service, see the [Administer containers in Azure learning path](https://docs.microsoft.com/en-us/learn/paths/administer-containers-in-azure/).

**Learning path summary**

Congratulations. You've completed the final module in the *Deploy applications with Azure DevOps* learning path. The main focus of this learning path is deploying applications by using Azure Pipelines.

In this learning path, you accomplished a lot, including:

* Building a basic release pipeline that deploys a web application to App Service.
* Expanding your basic release pipeline to a multistage pipeline that deploys to various development, test, and staging environments.
* Running both functional and nonfunctional tests in the pipeline.
* Implementing a blue-green deployment that updates the application and requires minimal downtime.
* Extending pipelines to add support for different deployment targets, such as Azure Functions and Kubernetes.

You can apply what you've learned to the applications and services that you work on.

**Continue the journey**

This learning path focused on deploying applications by using Azure Pipelines. You know that you can use the Azure portal, the Azure CLI, or Azure PowerShell to provision resources on Azure. But how will you scale out your deployments to meet global demand?

In [Provision infrastructure in Azure Pipelines](https://docs.microsoft.com/en-us/learn/modules/provision-infrastructure-azure-pipelines/), you'll help the Tailspin team answer this question. There, you learn how to use *infrastructure as code* to describe and automatically provision the infrastructure that you need for your application.

For more self-paced, hands-on learning related to Azure DevOps, also check out [Azure DevOps Labs](https://www.azuredevopslabs.com/).