**Create a build pipeline with Azure Pipelines**

**Learning Objectives**

Set up a continuous integration (CI) pipeline that automates the process of building your application.

## Learning objectives

In this module, you will:

* Create a build pipeline in Azure Pipelines
* Map the manual build steps to automated build tasks
* Publish your builds so others can access them
* Use templates to build multiple configurations

# Introduction

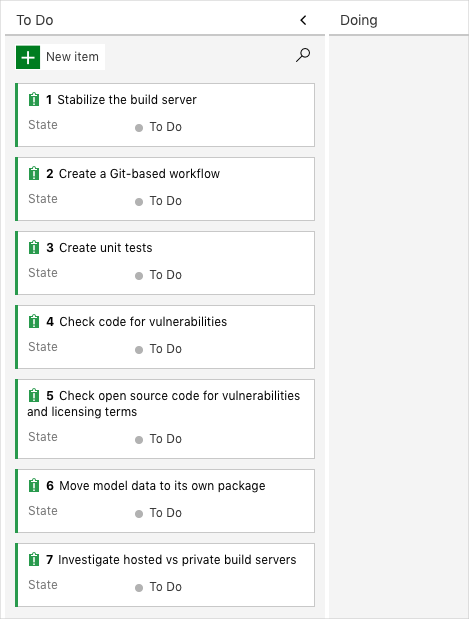
* 2 minutes

In the [Evolve your DevOps practices](https://docs.microsoft.com/en-us/learn/paths/evolve-your-devops-practices) learning path, you helped the Tailspin team start their DevOps journey by evaluating their current processes and technologies and then planning their initial set of tasks on Azure Boards.

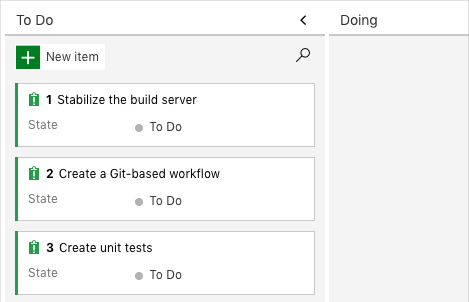
In this module, you help the team with their first task: setting up a continuous integration (CI) pipeline for their application.

The Tailspin web team is being pulled in many directions. Andy, the lead dev, is running from one meeting to another and never has any time. Amita, the QA person, is helping a tester on another team. Tim, who's in ops, is upgrading his servers and hasn't been seen for days. Mara's dream of a true DevOps team is stalled (again).

But she isn't giving up. She remembers that she and the team used Azure Boards to create a backlog of issues with the build process:



She also remembers that they picked three of the problems to fix within the next two weeks. (Of course, everyone else has forgotten about them.)



Mara decides to take an initial pass on the biggest issue, **Stabilize the build server**, herself. She won't try to fix everything. Instead, she's going to see if she can use Microsoft Azure Pipelines to replicate the current build process. She's convinced that Azure Pipelines provides enough benefits to improve the process. If she's right, she'll show her version to the team to see if she can make them more enthusiastic.

## Learning objectives

In this module, you will:

* Create a build pipeline in Azure Pipelines
* Map the manual build steps to automated build tasks
* Publish your builds so others can access them
* Use templates to build multiple configurations

## Prerequisites

The modules in this learning path and previous learning path form a progression.

To follow the progression from the beginning, be sure to first complete the [Evolve your DevOps practices](https://docs.microsoft.com/en-us/learn/paths/evolve-your-devops-practices) learning path.

If you want to start with this learning path, you need to set up a development environment on your Windows, macOS, or Linux system. You need:

* An [Azure DevOps organization](https://dev.azure.com/)
* A [GitHub](https://github.com/join" \t "az-portal) account
* [Visual Studio Code](https://code.visualstudio.com/)
* [.NET Core 3.1 SDK](https://dotnet.microsoft.com/download/dotnet-core/3.1)
* [Git](https://git-scm.com/downloads)

If your company's policy doesn't allow you to create an Azure DevOps organization, you can create your own personal account. You can get started with Azure DevOps for free.

This environment lets you complete the exercises in this and future modules. You can also use it to apply your new skills to your own projects.

**Note**

Keep in mind that you can use Azure DevOps to build and deploy almost any kind of application written in any language. In this module, you'll be working with a .NET Core application written in C#.

You don't need to be an expert in .NET or C# to complete this module. You can apply the patterns you learn here to your own projects that use your favorite programming languages and frameworks.

## Meet the team

You met the Space Game web team at Tailspin Toys in previous modules. As a refresher, here's who you'll work with in this module:



Andy is the development lead.



Amita is in QA.



Mara just joined as a developer and reports to Andy.

Mara has prior experience with DevOps and is helping the team adopt a more streamlined process by using Microsoft Azure DevOps.