Create your first pipeline

04/23/2019 • 19 minutes to read • Contributors

In this article

Azure Pipelines | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

This is a step-by-step guide to using Azure Pipelines to build a GitHub repository.

Prerequisites

- A GitHub account, where you can create a repository. If you don't have one, you can create one for free.
- An Azure DevOps organization. If you don't have one, you can create one for free. (An Azure DevOps organization is different from your GitHub organization. Give them the same name if you want alignment between them.)

  If your team already has one, then make sure you're an administrator of the Azure DevOps project that you want to use.

Get the sample code

You can use Azure Pipelines to build an app written in any language. Select a sample repository of your choice from the following languages and fork it into your own GitHub user account:

Programming language	Repository with a sample app
Docker	https://github.com/MicrosoftDocs/pipelines-dotnet-core-docker
.NET Core	https://github.com/MicrosoftDocs/pipelines-dotnet-core
Go	https://github.com/MicrosoftDocs/pipelines-go

Programming language	Repository with a sample app
Java	https://github.com/MicrosoftDocs/pipelines-java
Node.js	https://github.com/MicrosoftDocs/pipelines-javascript
Python	https://github.com/MicrosoftDocs/pipelines-python-django

You should now have a sample app in your GitHub account.

Get your first build

1. Sign in to your Azure DevOps organization and navigate to your project.
2. In your project, navigate to the **Pipelines** page. Then choose the action to create a new pipeline.
3. Walk through the steps of the wizard by first selecting **GitHub** as the location of your source code.
4. You might be redirected to GitHub to sign in. If so, enter your GitHub credentials.
5. When the list of repositories appears, select your desired sample app repository.
6. Azure Pipelines will analyze your repository.
    - Azure Pipelines recommends a starter template based on the code in your repository. If you see multiple recommendations, pick the first one. Select **Save and run**, then select **Commit directly to the master branch**, and then choose **Save and run** again.
    - If your repository already contains an `azure-pipelines.yml` file, then that file will be used, and you'll see a **Run** button. Select **Run**.
7. A new build is started. Wait for the build to finish.

Add a CI status badge to your repository

Many developers like to show that they're keeping their code quality high by displaying a status badge in their repo.

To copy the status badge to your clipboard:

1. In Azure Pipelines, go to the **Pipelines** page to view the list of pipelines. Select the pipeline you created in the previous section.
2. In the context menu for the pipeline, select **Status badge**.
3. Copy the sample Markdown from the status badge panel.

Now with the badge Markdown in your clipboard, take the following steps in GitHub:

1. Go to the list of files and select `Readme.md`. Select the pencil icon to edit.
2. Paste the status badge Markdown at the beginning of the file.
3. Commit the change to the `master` branch.
4. Notice that the status badge appears in the description of your repository.

Because you just changed the `Readme.md` file in this repository, Azure Pipelines automatically builds your code, according to the configuration in the `azure-pipelines.yml` file at the root of your repository. Back in Azure Pipelines, observe that a new build appears. Each time you make an edit, Azure Pipelines starts a new build.

Next steps

You've just learned the basics of using Azure Pipelines. Now you're ready to further configure your pipeline to run tests, publish test results, create container images, or even deploy the app to a cloud service. Follow a track for the language of your choice:

- .NET Core
- Docker
- Go
- Java
- Node.js
- Python

To adjust the timeout of your job, see Timeouts.

To run your pipeline in a container, see Container jobs.

For details about building GitHub repositories, see Build GitHub repositories.

To learn what else you can do in YAML pipelines, see YAML schema reference.