

Create a CI/CD pipeline for .NET with the Azure DevOps Project

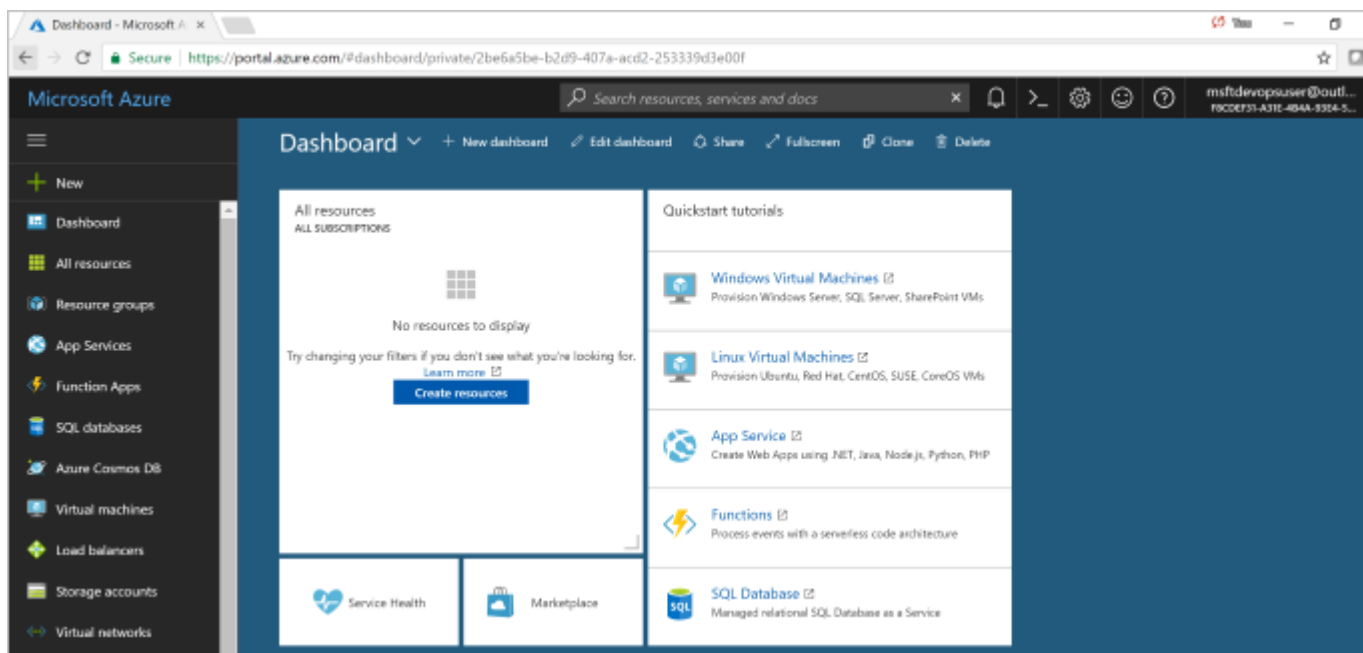
Configure continuous integration (CI) and continuous delivery (CD) for your .NET core or ASP.NET application with The **Azure DevOps Project**. The Azure DevOps project simplifies the initial configuration of a VSTS build and release pipeline.

If you don't have an Azure subscription, you can get one free through [Visual Studio Dev Essentials](#).

Sign in to the Azure portal

The Azure DevOps Project creates a CI/CD pipeline in VSTS. You can create a **new VSTS** account or use an **existing account**. The Azure DevOps Project also creates **Azure resources** in the **Azure subscription** of your choice.

1. Sign into the [Microsoft Azure portal](#).
2. Choose the **Create a resource** icon in the left navigation bar, then search for **DevOps project**. Choose **Create**.










Select a sample application and Azure service

1. Select the **.NET** sample application. The .NET samples include a choice of either the open source ASP.NET framework or the cross-platform .NET Core framework.



Start fresh with a sample application

 .NET  New Web app by using ASP.NET or .NET Core	 Java New Web app by using Spring or JSF
 Static Website New static website by using HTML, CSS and JavaScript	 Node.js New Web app by using Node.js, Express.js or Sails.js
 PHP New Web app by using Laravel or Symfony	 Python New Web app by using Python, Bottle, Django or Flask

or use your application

Previous

Next

2. Select the **.NET Core** application framework. This sample is an ASP.NET Core MVC application. When you're done, choose **Next**.
3. **Web App on Windows** is the default deployment target. Optionally, you can choose Web App on Linux or Web App for Containers. The application framework, which you chose on the previous steps, dictates the type of Azure service deployment target available here. Leave the default service, and then choose **Next**.

Configure VSTS and an Azure subscription

1. Create a **new** free VSTS account or choose an **existing** account. Choose a **name** for your VSTS project. Select your **Azure subscription**, **location**, and choose a **name** for your application. When you're done, choose **Done**.



Runtime



Framework



Service



Create

Almost there!

Ready to deploy an .NET Core web app to a new Web App on Windows.

Visual Studio Team Services

A Continuous Delivery pipeline will be setup in Visual Studio Team Services (VSTS).

* Account



Create new



Use existing

devopstutorial



* Name

SampleTeamProject



Azure

[Change](#)

We will create the following Azure resources and deploy an application.

* Subscription

Pay-As-You-Go



* App Service Location

South Central US



* Name

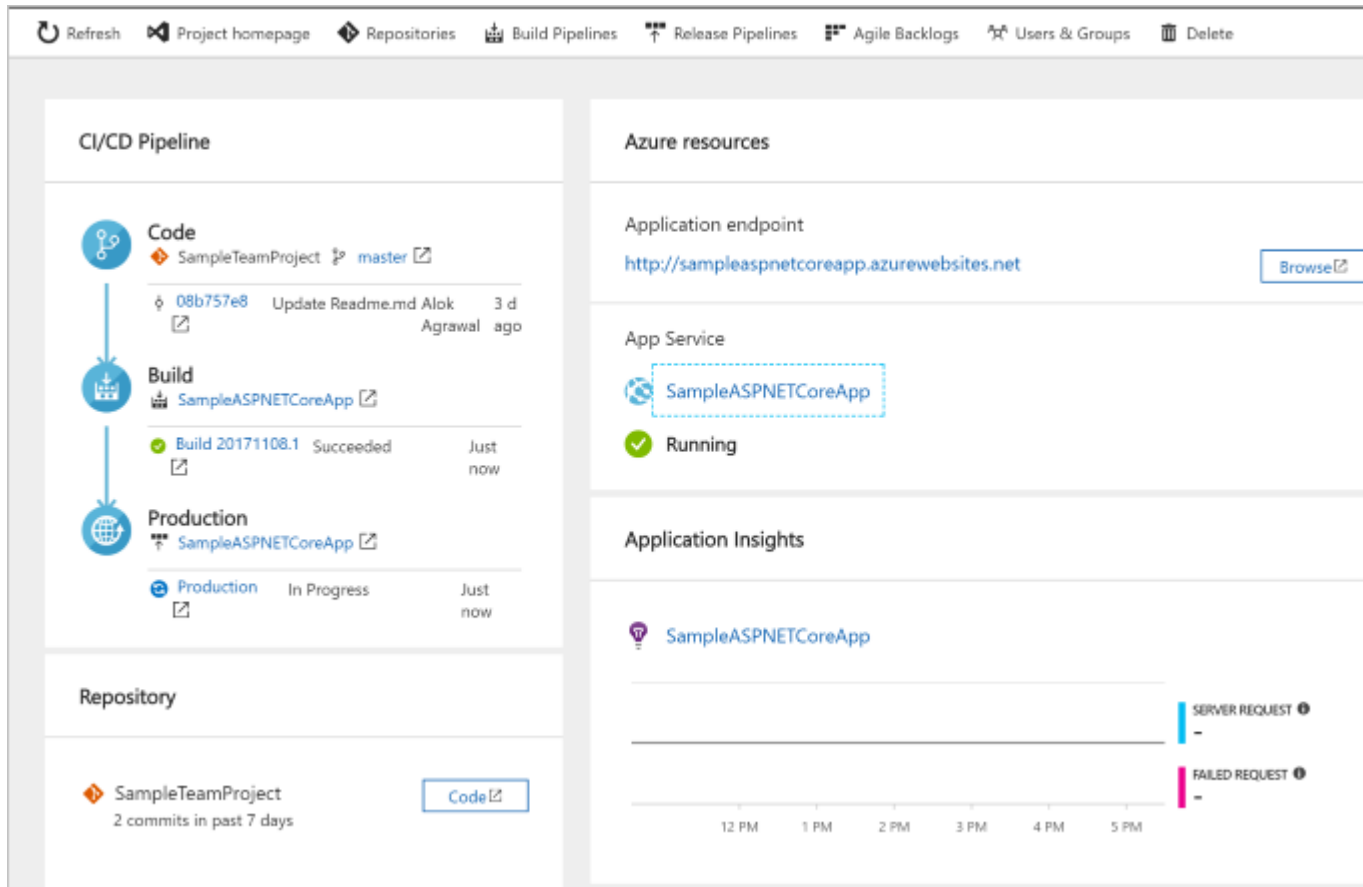
SampleASPNETCoreApp



.azurewebsites.net

Pricing tier: S1 Standard

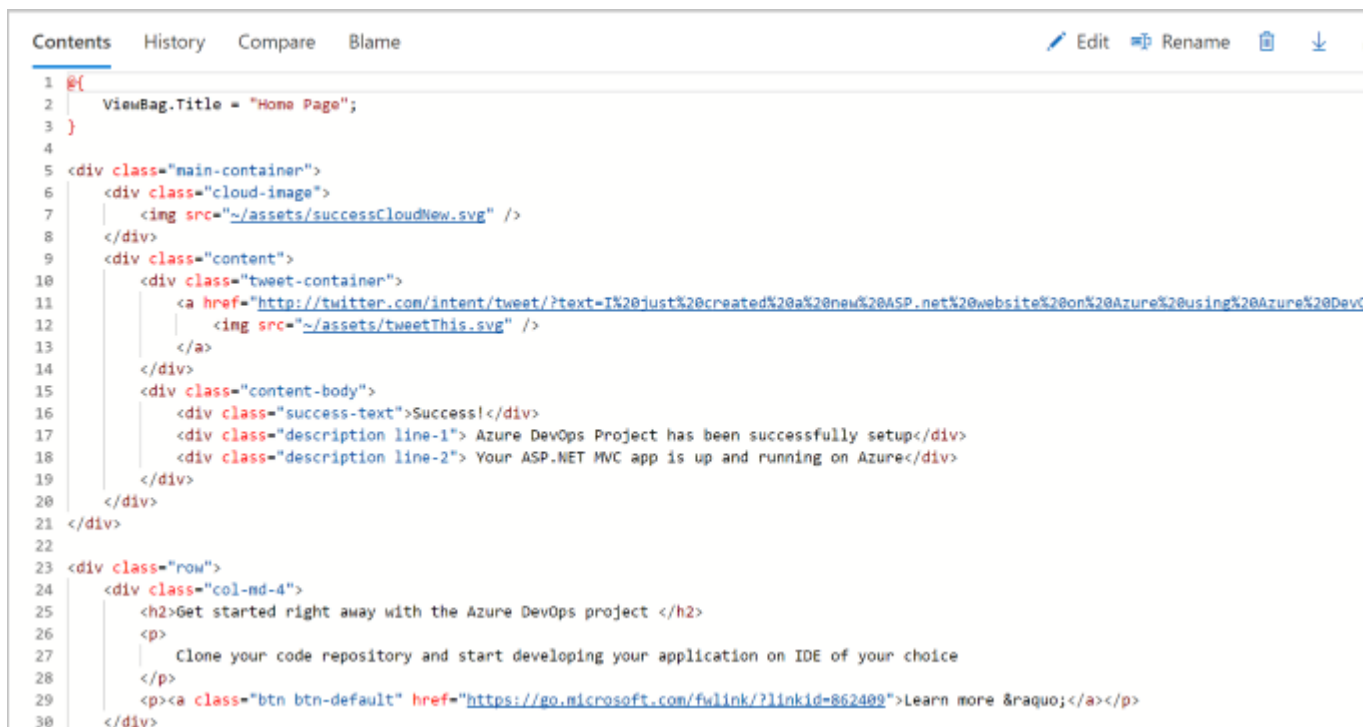
2. In a few minutes, the **project dashboard** loads in the Azure portal. A sample application is set up in a repository in your VSTS account, a build executes, and your application deploys to Azure. This dashboard provides visibility into your **code repository**, **VSTS CI/CD pipeline**, and your **application in Azure**. On the right side of the dashboard, select **Browse** to view your running application.



Commit code changes and execute CI/CD

The Azure DevOps project created a Git repository in your VSTS or GitHub account. Follow the steps below to view the repository and make code changes to your application.

1. On the left-hand side of the DevOps project dashboard, select the link for your **master** branch. This link opens a view to the newly created Git repository.
2. To view the repository clone URL, select **Clone** from the top right of the browser. You can clone your Git repository in your favorite IDE. In the next few steps, you can use the web browser to make and commit code changes directly to the master branch.
3. On the left-hand side of the browser, navigate to the **Views/Home/index.cshtml** file.
4. Select **Edit**, and make a change to the h2 heading. For example, type **Get started right away with the Azure DevOps Project** or make some other change.



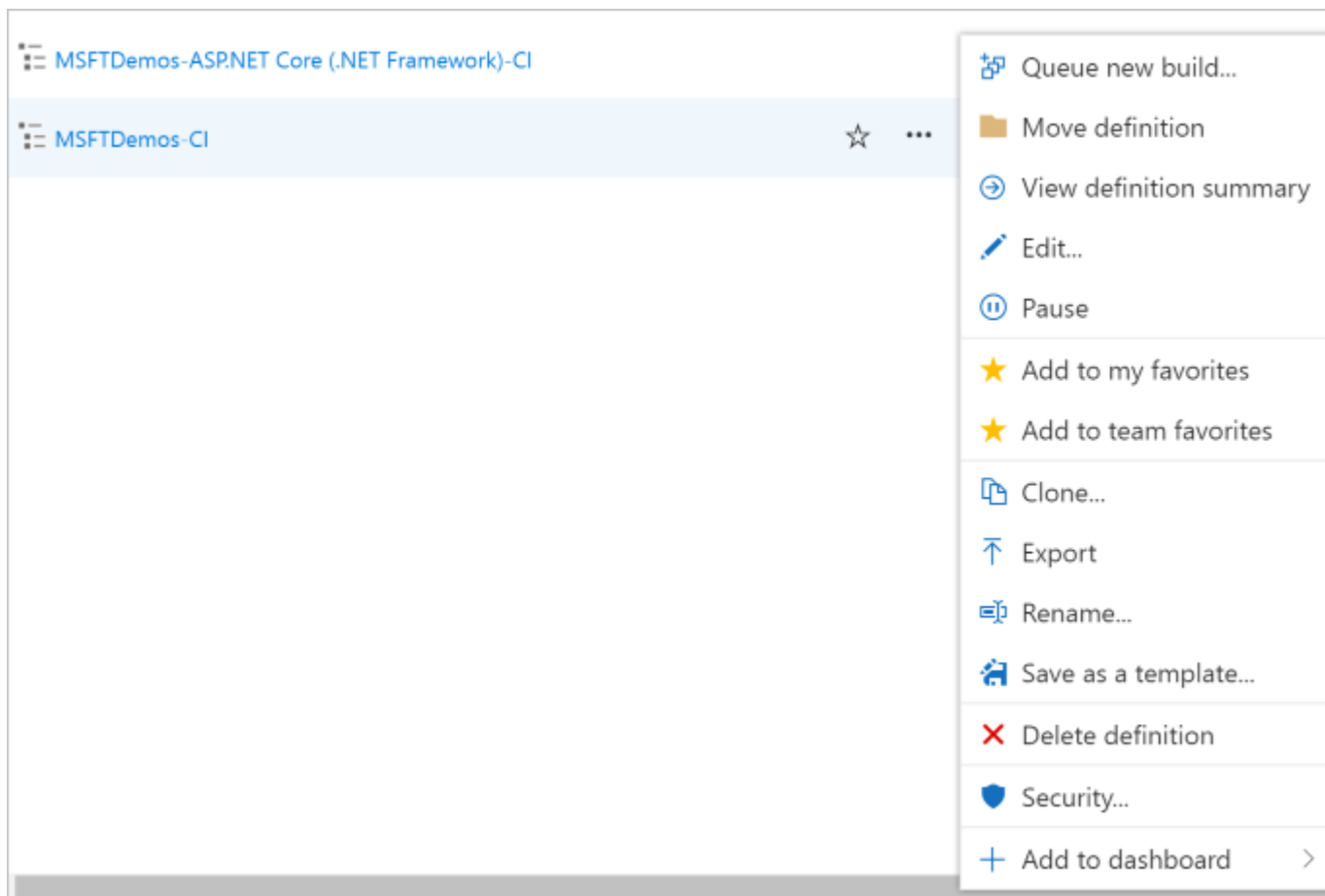
```
Contents History Compare Blame
1 @{
2     ViewBag.Title = "Home Page";
3 }
4
5 <div class="main-container">
6     <div class="cloud-image">
7         
8     </div>
9     <div class="content">
10         <div class="tweet-container">
11             <a href="http://twitter.com/intent/tweet/?text=I%20just%20created%20a%20new%20ASP.net%20website%20on%20Azure%20using%20Azure%20DevOps">
12                 
13             </a>
14         </div>
15         <div class="content-body">
16             <div class="success-text">Success!</div>
17             <div class="description line-1"> Azure DevOps Project has been successfully setup</div>
18             <div class="description line-2"> Your ASP.NET MVC app is up and running on Azure</div>
19         </div>
20     </div>
21 </div>
22
23 <div class="row">
24     <div class="col-md-4">
25         <h2>Get started right away with the Azure DevOps project </h2>
26         <p>
27             Clone your code repository and start developing your application on IDE of your choice
28         </p>
29         <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?linkid=862499">Learn more &raquo;</a></p>
30     </div>
```

5. Choose **Commit**, then save your changes.
6. In your browser, navigate to the **Azure DevOps project dashboard**. You should now see a build is in progress. The changes you just made are automatically built and deployed via a VSTS CI/CD pipeline.

Examine the VSTS CI/CD pipeline

The Azure DevOps project automatically configured a full VSTS CI/CD pipeline in your VSTS account. Explore and customize the pipeline as needed. Follow the steps below to familiarize yourself with the VSTS build and release definitions.

1. Select **Build Pipelines** from the **top** of the Azure DevOps project dashboard. This link opens a browser tab and opens the VSTS build definition for your new project.
2. Select the **ellipsis**. This action opens a menu where you can perform several activities such as queue a new build, pause a build, and edit the build definition.
3. Select **Edit**.



4. From this view, **examine the various tasks** for your build definition. The build performs various tasks such as fetching sources from the Git repository, restoring dependencies, and publishing outputs used for deployments.
5. At the top of the build definition, select the **build definition name**.
6. Change the **name** of your build definition to something more descriptive. Select **Save & queue**, then select **Save**.
7. Under your build definition name, select **History**. You see an audit trail of your recent changes for the build. VSTS keeps track of any changes made to the build definition, and allows you to compare versions.
8. Select **Triggers**. The Azure DevOps project automatically created a CI trigger, and every commit to the repository initiates a new build. You can optionally choose to include or exclude branches from the CI process.
9. Select **Retention**. Based on your scenario, you can specify policies to keep or remove a certain number of builds.
10. Select **Build and Release**, then choose **Releases**. The Azure DevOps project created a VSTS release definition to manage deployments to Azure.
11. On the left-hand side of the browser, select the **ellipsis** next to your release definition, then choose **Edit**.
12. The release definition contains a **pipeline**, which defines the release process. Under **Artifacts**, select **Drop**. The build definition you examined in the previous steps produces the output used for the artifact.

13. To the right-hand side of the **Drop** icon, select the **Continuous deployment trigger**. This release definition has an enabled CD trigger, which executes a deployment every time there is a new build artifact available. Optionally, you can disable the trigger, so your deployments require manual execution.
14. On the left-hand side of the browser, select **Tasks**. The tasks are the activities your deployment process performs. In this example, a task was created to deploy to **Azure App service**.
15. On the right-hand side of the browser, select **View releases**. This view shows a history of releases.
16. Select the **ellipsis** next to one of your releases, and choose **Open**. There are several menus to explore from this view such as a release summary, associated work items, and tests.
17. Select **Commits**. This view shows code commits associated with the specific deployment.
18. Select **Logs**. The logs contain useful information about the deployment process. They can be viewed both during and after deployments.

Clean up resources

When no longer needed, you can delete the Azure App service and related resources created in this quickstart by using the **Delete** function from the Azure DevOps project dashboard.