# Build, test, and deploy .NET  MVC 5

**Azure DevOps Services | Azure DevOps Server 2022 - Azure DevOps Server 2019**

Use a pipeline to automatically build and test your .NET mvc projects.

* Set up your build environment with [Microsoft-hosted](https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/hosted?view=azure-devops) or [self-hosted](https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=azure-devops) agents.
* Use the [publish code coverage task](https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/test/publish-code-coverage-results?view=azure-devops) to publish code coverage results.
* Package and deliver your code with the [.NET  mvc task](https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/build/dotnet-core-cli?view=azure-devops) and the [publish build artifacts task](https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/utility/publish-build-artifacts?view=azure-devops).
* Deploy your AWS Windows' server 2019 .

**Create your first pipeline**

**Create a .NET project**

If you don't have a .NET project to work with, create a new one, and upload your code to your GitHub repository or Azure Repos. Start by installing the  .NET 5.0

Create a new .NET

From the same terminal session, run the application locally using the [dotnet run](https://docs.microsoft.com/en-us/dotnet/core/tools/dotnet-run) command from your project directory.

**Upload your code**

Upload your code to new webapp GitHub or Azure Repos:

* [Create a new Git repo in Azure Repos](https://docs.microsoft.com/en-us/azure/devops/repos/git/creatingrepo?view=azure-devops).
* [Create a new GitHub repository](https://docs.github.com/en/get-started/quickstart/create-a-repo).

**Sign in to Azure Pipelines**

Sign-in to [Azure Pipelines](https://azure.microsoft.com/services/devops/pipelines). After you sign in, your browser goes to <https://dev.azure.com/my-organization-name> and displays your Azure DevOps dashboard.

Within your selected organization, create a *project*. If you don't have any projects in your organization, you see a **Create a project to get started** screen. Otherwise, select the **New Project** button in the upper-right corner of the dashboard.

**Create the pipeline**

1. Sign-in to your Azure DevOps organization and go to your project.
2. Go to **Pipelines**, and then select **New pipeline**.Graphical user interface, text, application, email

   Description automatically generated
3. Do the steps of the wizard by first selecting **GitHub** as the location of your source code.

Graphical user interface, text, application, email

Description automatically generated

1. You might be redirected to GitHub to sign in. If so, enter your GitHub credentials.
2. When you see the list of repositories, select your repository.

Graphical user interface, text, application, email

Description automatically generated

1. You might be redirected to GitHub to install the Azure Pipelines app. If so, select **Approve & install**.
2. When the **Configure** tab appears, select **ASP.NET**

Graphical user interface, text, application, email

Description automatically generated

1. Examine your new pipeline to see what the YAML does. When you're ready, select **Save and run**.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Graphical user interface, text, application, chat or text message

   Description automatically generated
2. Commit a new *azure-pipelines.yml* file to your repository. After you're happy with the message, select **Save and run** again.
3. If you want to watch your pipeline in action, select the build job.
4. Because your code appeared to be a good match for the ASP.NET 5.0 template, we automatically created the pipeline for you, generate artifice.

Text

Description automatically generated

1. After the completed build select Release pipeline and, in the Release, pipeline select create new release.

Graphical user interface, application

Description automatically generated

1. Select artifact latest

Graphical user interface, application, PowerPoint

Description automatically generated

15. In the stage select IIS website and SQL database deployment

Deployment Group: Deploy ASP.NET or ASP.NET Core web applications to an IIS Website and SQL database on physical or virtual machines (VM).

**Apply**

Apply IIS website and SQL database deployment template

Graphical user interface, application

Description automatically generated

16. Inside stage one we have stage1, IIs Deployment, IIS Web App manager, IIs web App Deploy, in the Deployment group select the server's name and add port number

Graphical user interface, application, email

Description automatically generated

17.In IIS web app Manager click Enable IIs

Graphical user interface, text, application, email

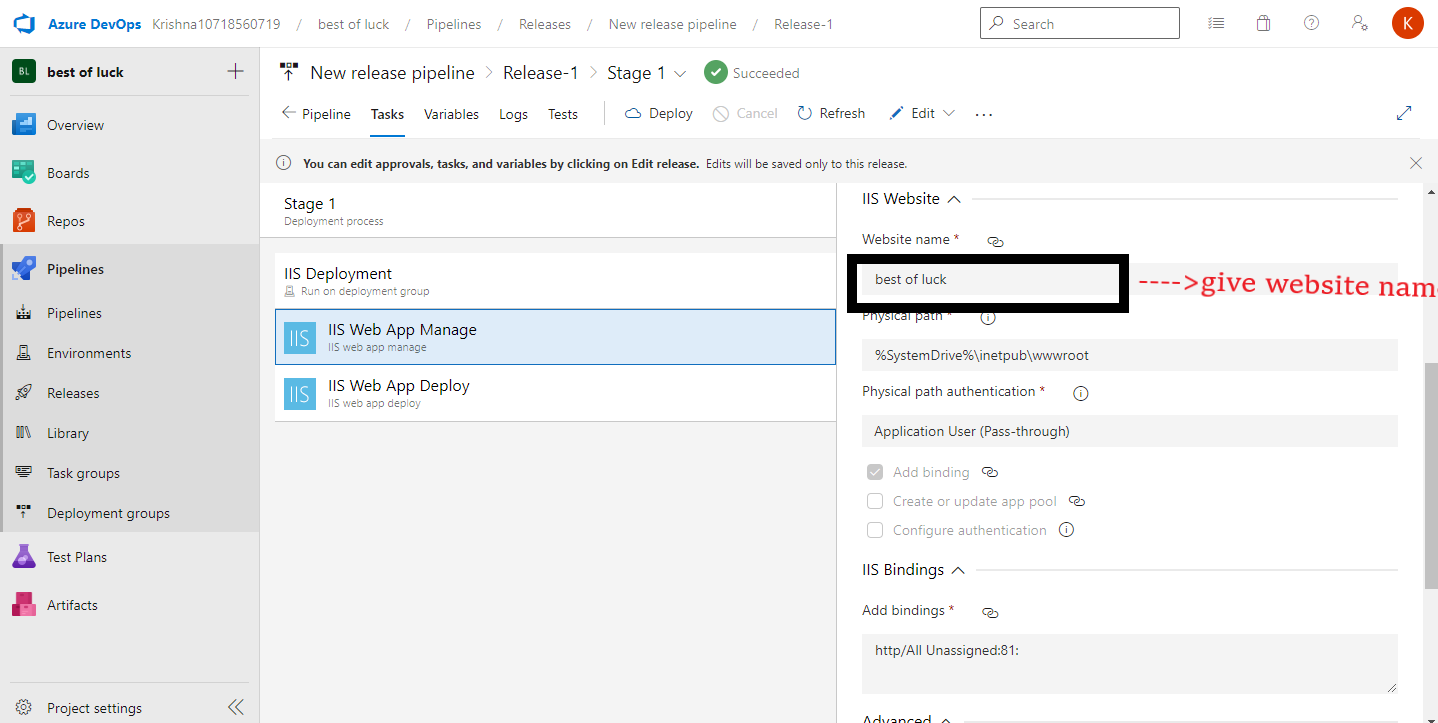
Description automatically generated

18. Select Deployment group server name

Graphical user interface, text, application, email

Description automatically generated

In the IIS web App manager give the website name



Graphical user interface, text, application, email

Description automatically generated

19. Click the release

Graphical user interface, application

Description automatically generated

20. You will see this image click on right mark

Graphical user interface, application

Description automatically generated

21. Click right we will get output

Before Create Release in the AWS cloud lunch ec2 windows 2019 server

Select Deployment group

Graphical user interface, text, application, email

Description automatically generated

sClick new option

Graphical user interface, text, application, email

Description automatically generated  
Give the Deployment group name

Graphical user interface, application

Description automatically generated

Copy the script in the clipboard and past in the windows power sell, the power shell must run as administration

Graphical user interface, text, application

Description automatically generated

After words we have to click deployment online or offline, it should be online only deployment is successful

Graphical user interface, text, application, email

Description automatically generated