

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](#) or [self-hosted](#) agents.
- Use the [publish code coverage task](#) to publish code coverage results.
- Package and deliver your code with the [.NET mvc task](#) and the [publish build artifacts task](#).
- 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](#) 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](#).
- [Create a new GitHub repository](#).

Sign in to Azure Pipelines

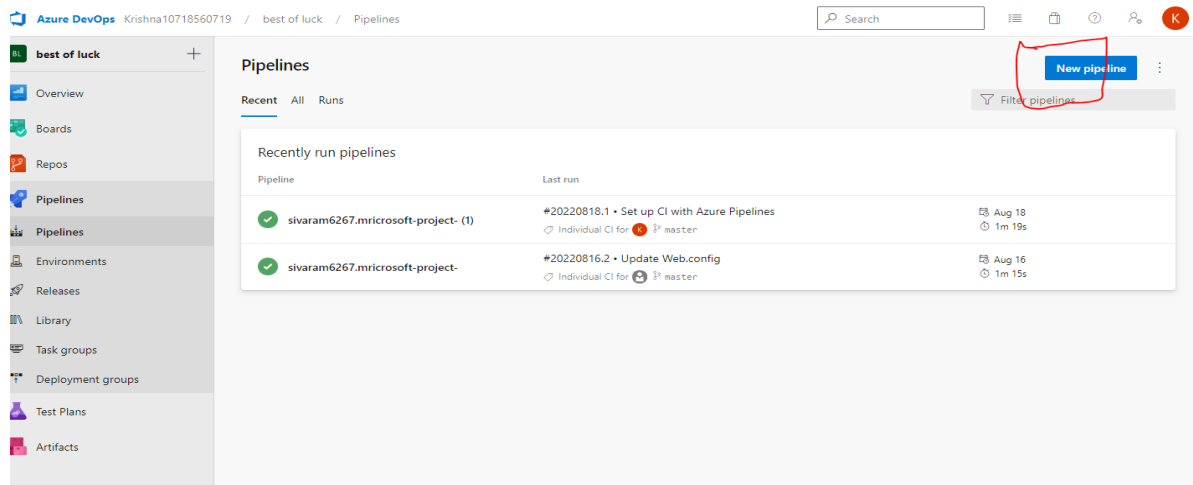
Sign-in to [Azure Pipelines](#). After you sign in, your browser goes to <https://dev.azure.com/my-organization-name> and displays your Azure DevOps dashboard.

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.

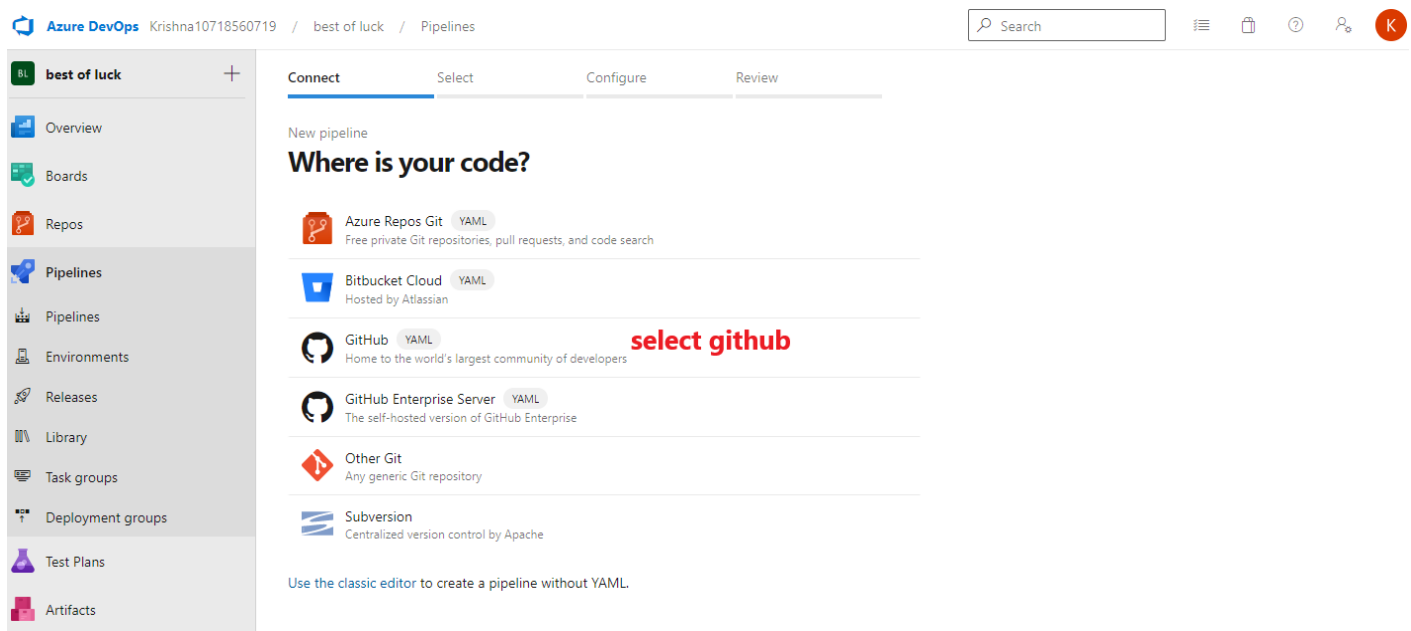
Create the pipeline

1. Sign-in to your Azure DevOps organization and go to your project.

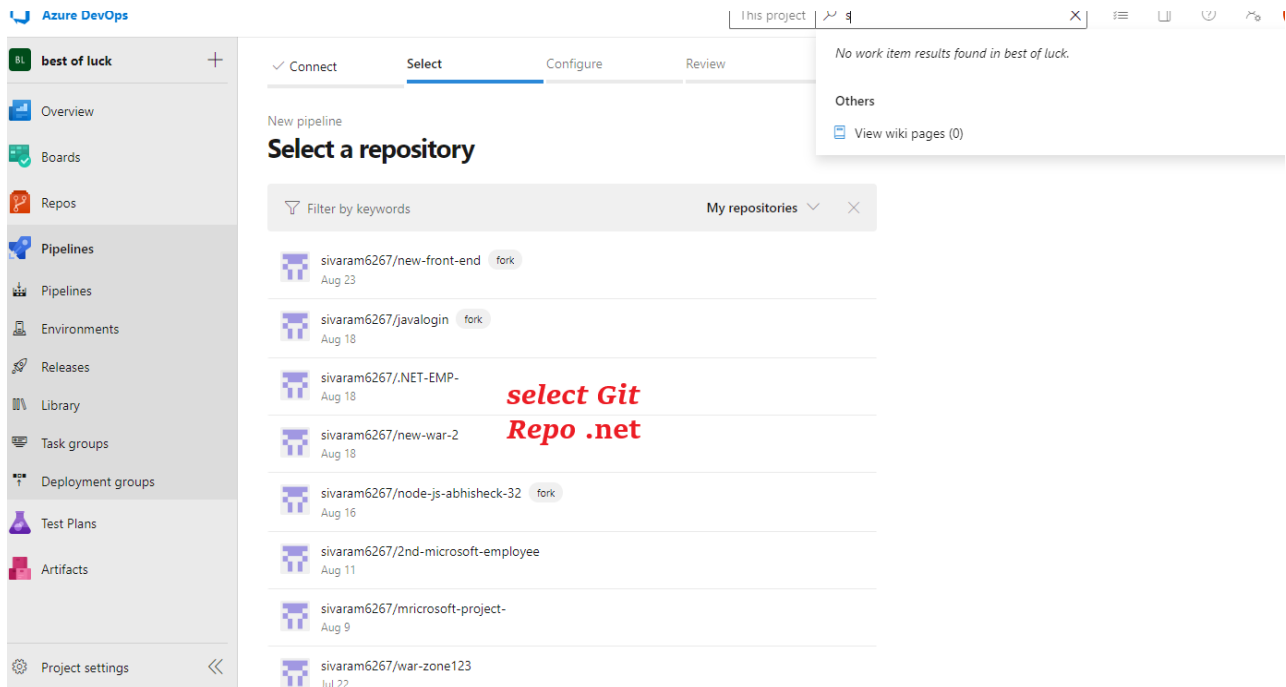
2. Go to **Pipelines**, and then select **New pipeline**.



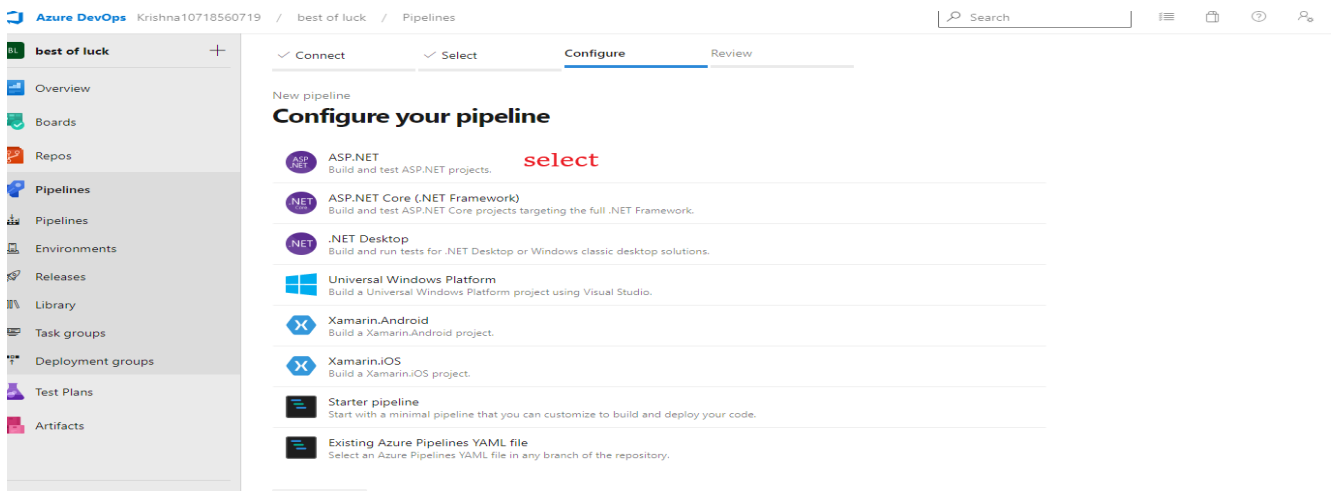
3. Do the steps of the wizard by first selecting **GitHub** as the location of your source code.



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



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



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

Azure DevOps Krishna10718560719 / best of luck / Pipelines

best of luck +

Overview Boards Repos Pipelines Pipelines Environments Releases Library Task groups Deployment groups Test Plans Artifacts Project settings <<

You selected a public repository, but this is not a public project. Go to [project settings](#) to change the visibility of the project. [Learn more](#)

Connect Select Configure **Review**

New pipeline

Review your pipeline YAML

[click The Run](#) Variables Run

azuram6267/.NET-EMP- / azure-pipelines.yml

```
1 # ASP.NET
2 # Build and test ASP.NET projects.
3 # Add steps that publish symbols, save build artifacts, deploy, and more:
4 # https://docs.microsoft.com/azure/devops/pipelines/apps/aspnet/build-aspnet-4
5
6 trigger:
7   - master
8
9 pool:
10  - vmImage: 'windows-latest'
11
12 variables:
13   - solution: '**/*.sln'
14   - buildPlatform: 'Any CPU'
15   - buildConfiguration: 'Release'
16
17 steps:
18   - task: NuGetToolInstaller@1
19
20   - task: NuGetCommand@2
```

Azure DevOps Krishna10718560719 / best of luck / Pipelines

best of luck +

Overview Boards Repos Pipelines Pipelines Environments Releases Library Task groups Deployment groups Test Plans Artifacts Project settings <<

You selected a public repository, but this is not a public project. Go to [project settings](#) to change the visibility of the project. [Learn more](#)

Connect Select Configure **Review**

New pipeline

Review your pipeline YAML

Variables Run

azuram6267/.NET-EMP- / azure-pipelines.yml

```
20 - task: NuGetCommand@2
21   inputs:
22     - restoreSolution: '$(solution)'
23
24   Settings
25   - task: VSBUILD@1
26     inputs:
27       - solution: '$(solution)'
28       - msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true /p:SkipInvalidConfigurations=true /p:Pack
29       - platform: '$(buildPlatform)'
30       - configuration: '$(buildConfiguration)'
31
32   Settings
33   - task: VSTest@2
34     inputs:
35       - platform: '$(buildPlatform)'
36       - configuration: '$(buildConfiguration)'
37
38   Settings
39   - task: PublishBuildArtifacts@1
```

Connect Select Configure **Review**

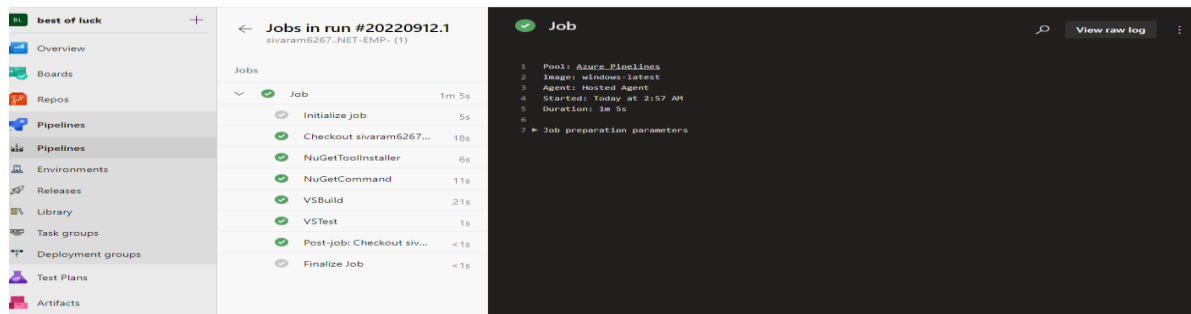
New pipeline

Review your pipeline YAML

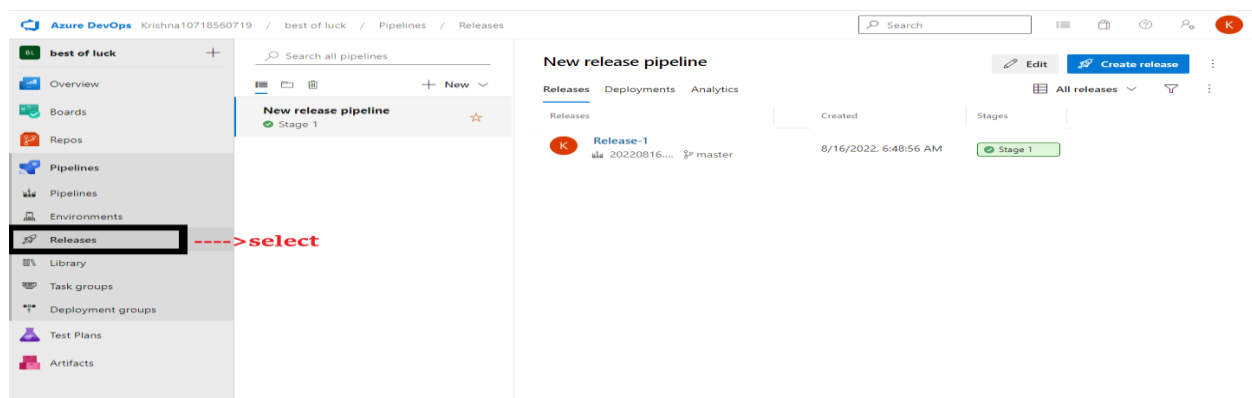
Save and run

9. azure-pipelines.yml

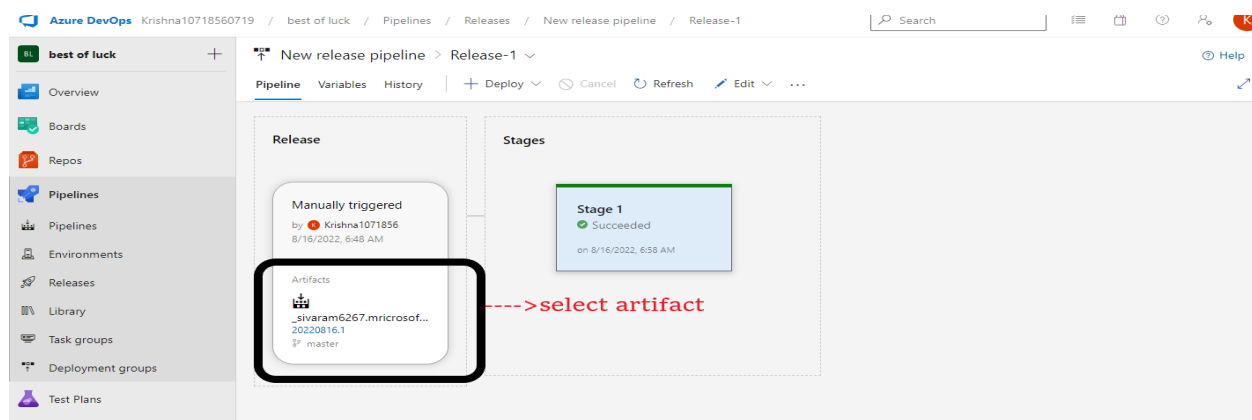
10. Commit a new *azure-pipelines.yml* file to your repository. After you're happy with the message, select **Save and run** again.
11. If you want to watch your pipeline in action, select the build job.
12. Because your code appeared to be a good match for the ASP.NET 5.0 template, we automatically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



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



14. Select artifact latest

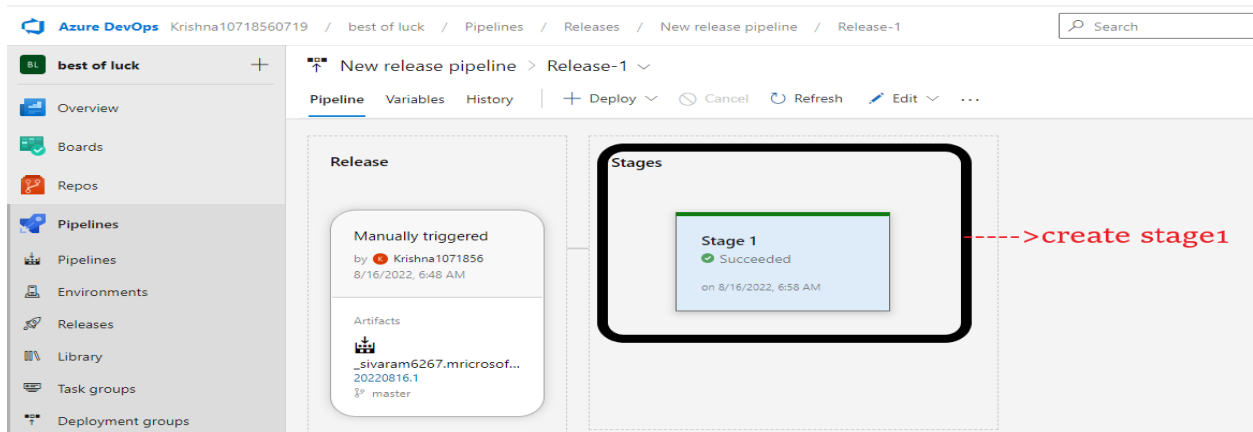


15. In the stage select IIS website and SQL database deployment

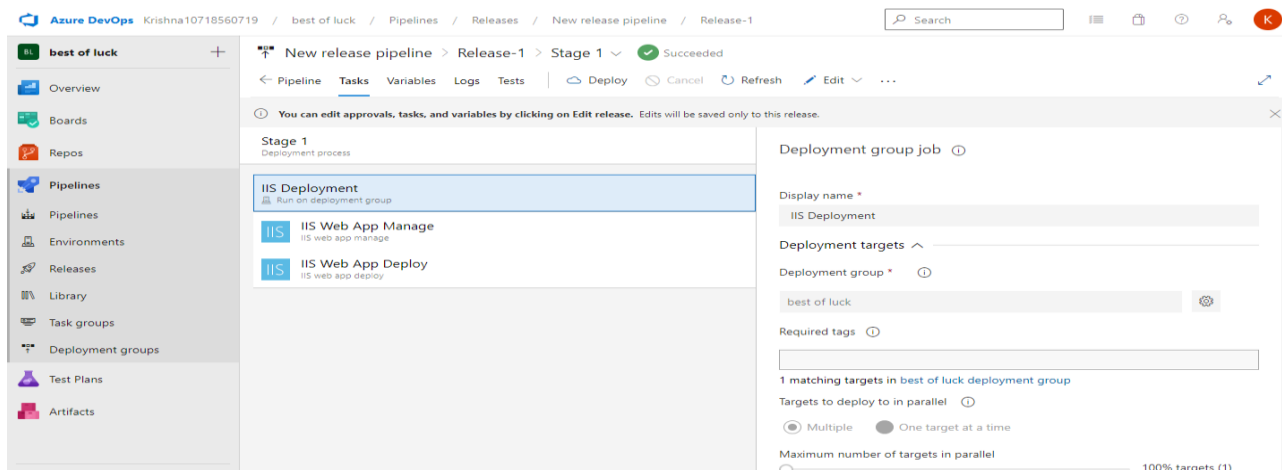
Deployment Group: Deploy ASP.NET or ASP.NET Core web applications to an IIS Website and SQL database on physical or virtual machines (VM).

Apply

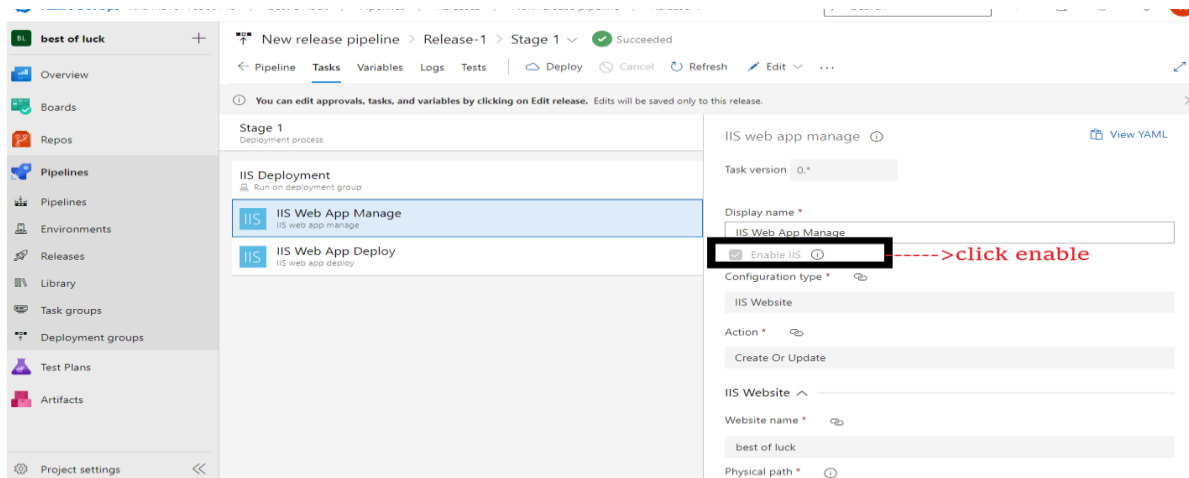
Apply IIS website and SQL database deployment template



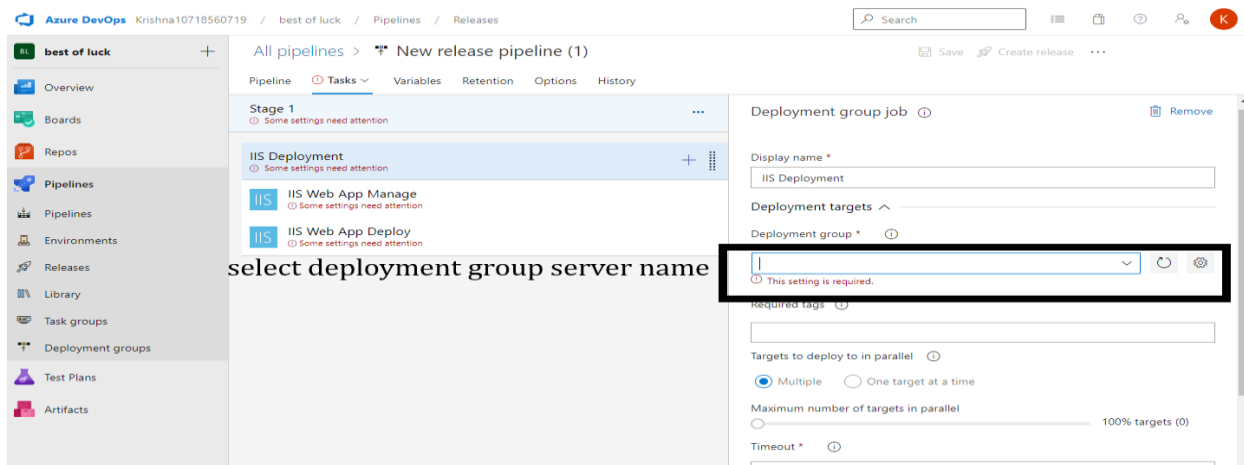
16. Inside stage one we have stage1, IIS Deployment, IIS Web App manager, IIS web App Deploy, in the Deployment group select the server's name and add port number



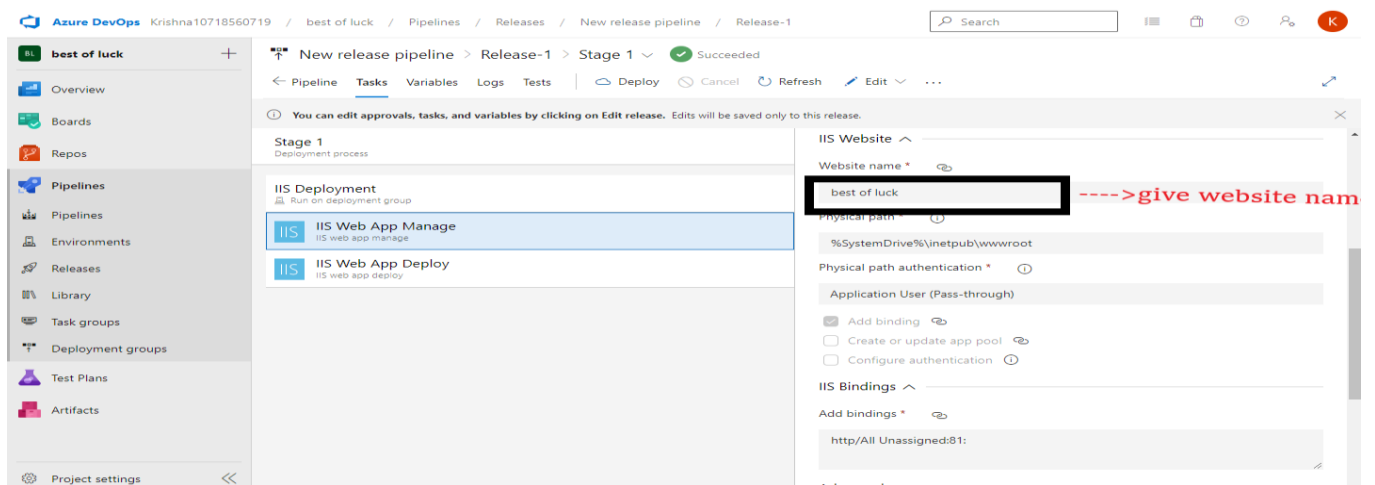
17. In IIS web app Manager click Enable IIS

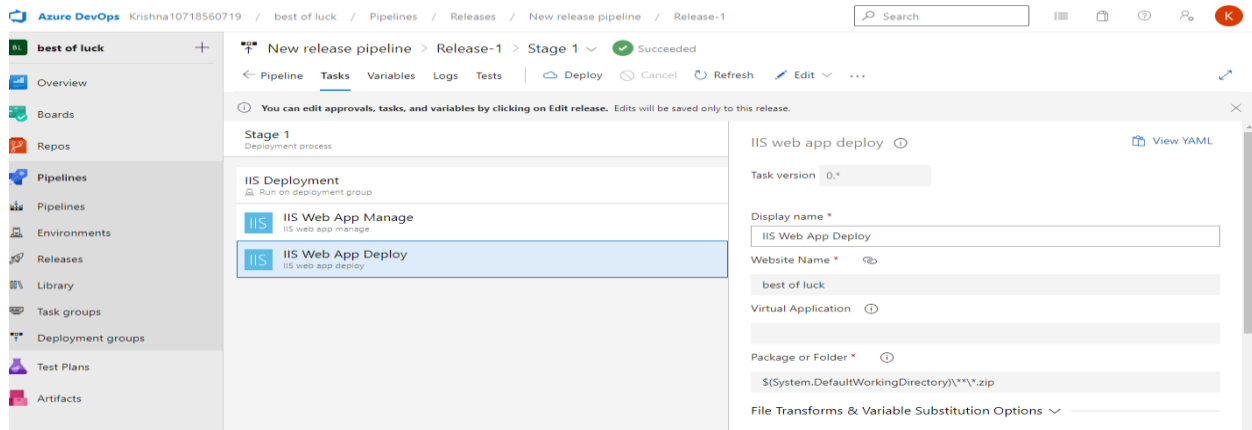


18. Select Deployment group server name

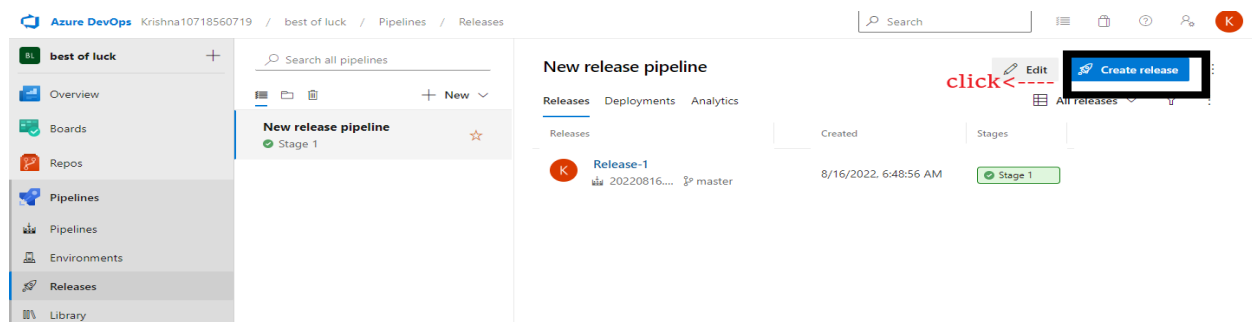


In the IIS web App manager give the website name

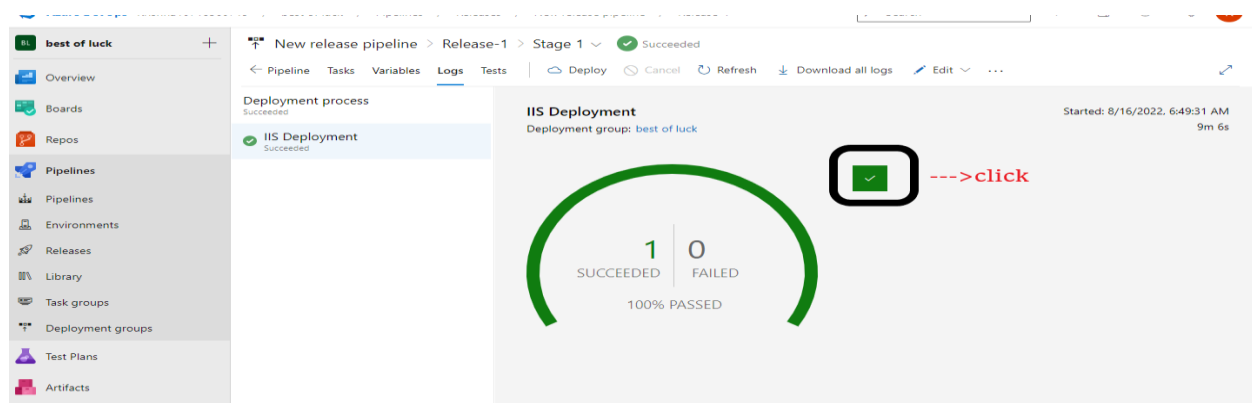




19. Click the release



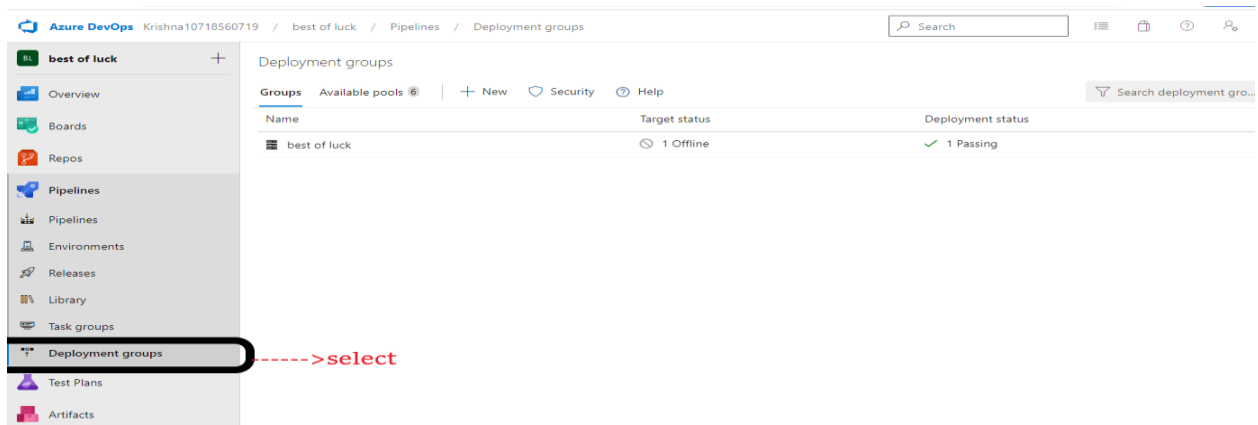
20. You will see this image click on right mark



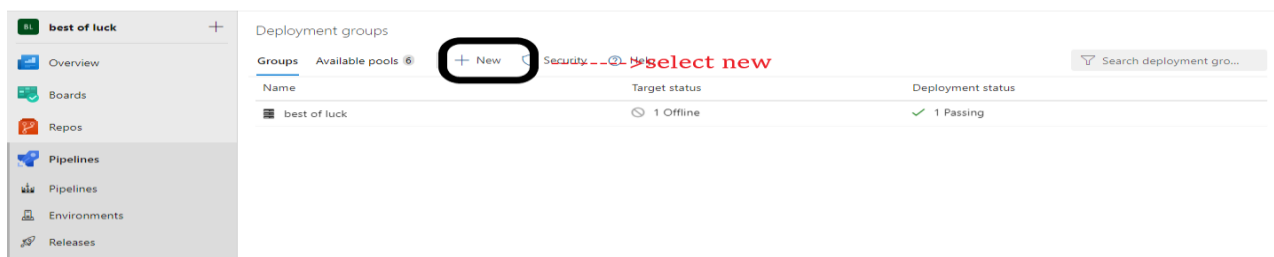
21. Click right we will get output

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

Select Deployment group



Click new option



Give the Deployment group name



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

Run from an administrator PowerShell command prompt in windows sever 2019
 ----->aws lunch ec2 instance
 ----->open powershell
 ----->run power shell paste copy script to the clipboard

```

$errorActionPreference="Stop";if(-NOT ([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole(
[Security.Principal.WindowsBuiltInRole] "Administrator")){ throw "Run command in
an administrator PowerShell prompt";if($PSVersionTable.PSVersion -lt (New-Object
System.Version("3.0"))){ throw "The minimum version of Windows PowerShell that is
required by the script (3.0) does not match the currently running version of
Windows PowerShell." };if(-NOT (Test-Path $env:SystemDrive\azagent')){mkdir
$env:SystemDrive\azagent}; cd $env:SystemDrive\azagent; for($i=1; $i -lt 100;
$i++){($destFolder="A"+$i.ToString());if(-NOT (Test-Path ($destFolder))){mkdir
$destFolder;cd $destFolder;break;}}; $agentZip="$PWD\agent.zip";$defaultProxy=
[System.Net.WebRequest]::DefaultWebProxy;$securityProtocol=@();$securityProtocol+=
[Net.ServicePointManager]::SecurityProtocol;$securityProtocol+=
[Net.SecurityProtocolType]::Tls12;
[Net.ServicePointManager]::SecurityProtocol=$securityProtocol;$WebClient=New-
Object Net.WebClient;
$uri="https://vstsagentpackage.azureedge.net/agent/2.210.0/vsts-agent-win-x64-
2.210.0.zip";if($defaultProxy -and (-not $defaultProxy.IsBypassed($uri)))
{($webClient.Proxy= New-Object
Net.WebProxy($defaultProxy.GetProxy($uri).OriginalString, $true));};
$WebClient.DownloadFile($uri, $agentZip);Add-Type -AssemblyName
System.IO.Compression.FileSystem;
[System.IO.Compression.ZipFile]::ExtractToDirectory( $agentZip,
"$PWD");. \config.cmd --deploymentgroup --deploymentgroupname "production" --agent
$env:COMPUTERNAME --runasservice --work "work" --url
'https://dev.azure.com/Krishna10718560719/' --projectname 'best of luck'; Remove-
Item $agentZip;
  
```

Use a personal access token in the script for authentication

Copy script to the clipboard

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

Groups	Available pools	Target status	Deployment status
best of luck		1 Offline	
production			1 Passing

