Configure CI/CD Pipeline in Azure DevOps

Lab 1: Creating the project

Lab 2: Continuous Integration - Build

Lab 3: Create an Azure Web App

Lab 4: Continuous Deployment – Release

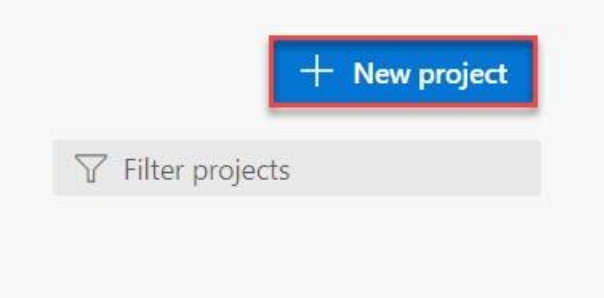
Lab 1: Creating the project

Step 1: Open **Azure DevOps**

https://dev.azure.com

Step 2: Create New Azure DevOps project

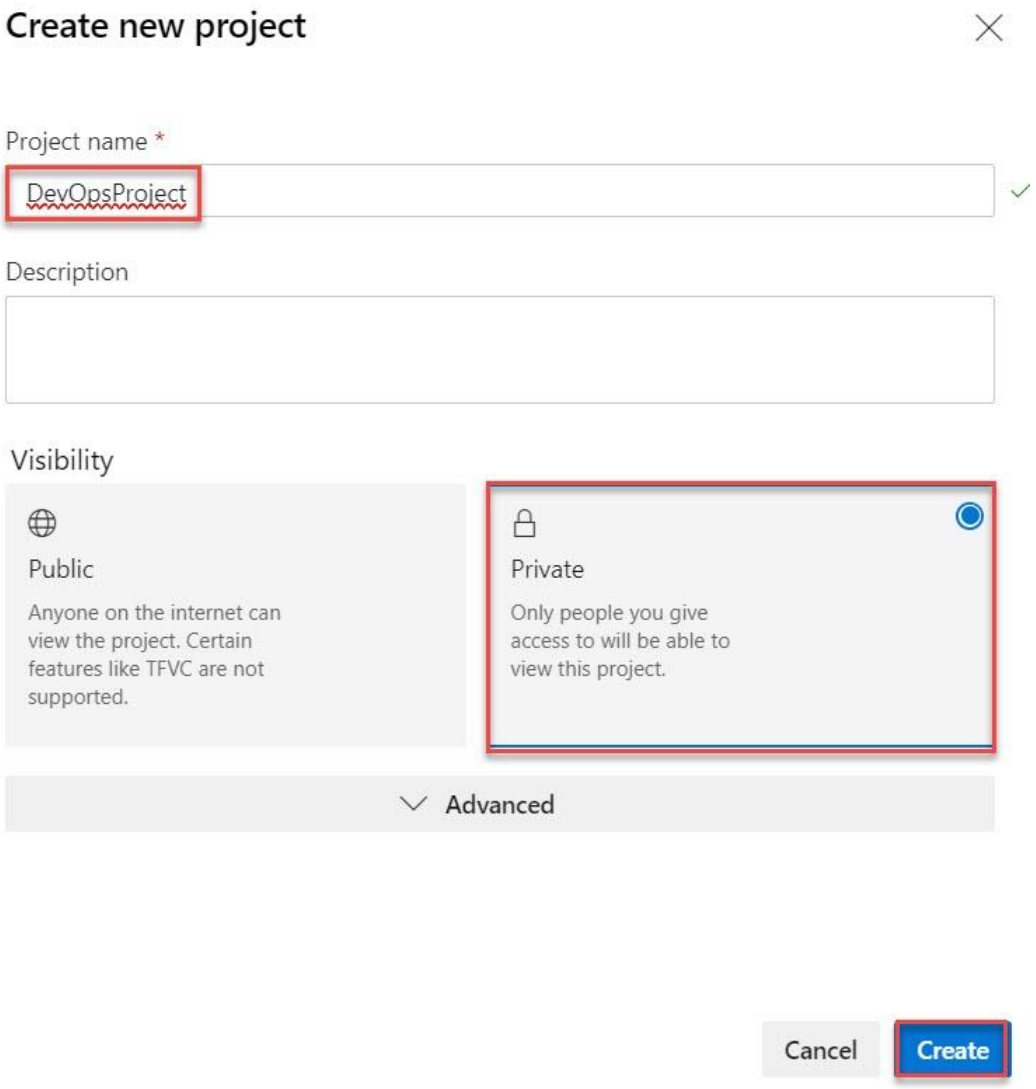
+ New Project or + Create Project



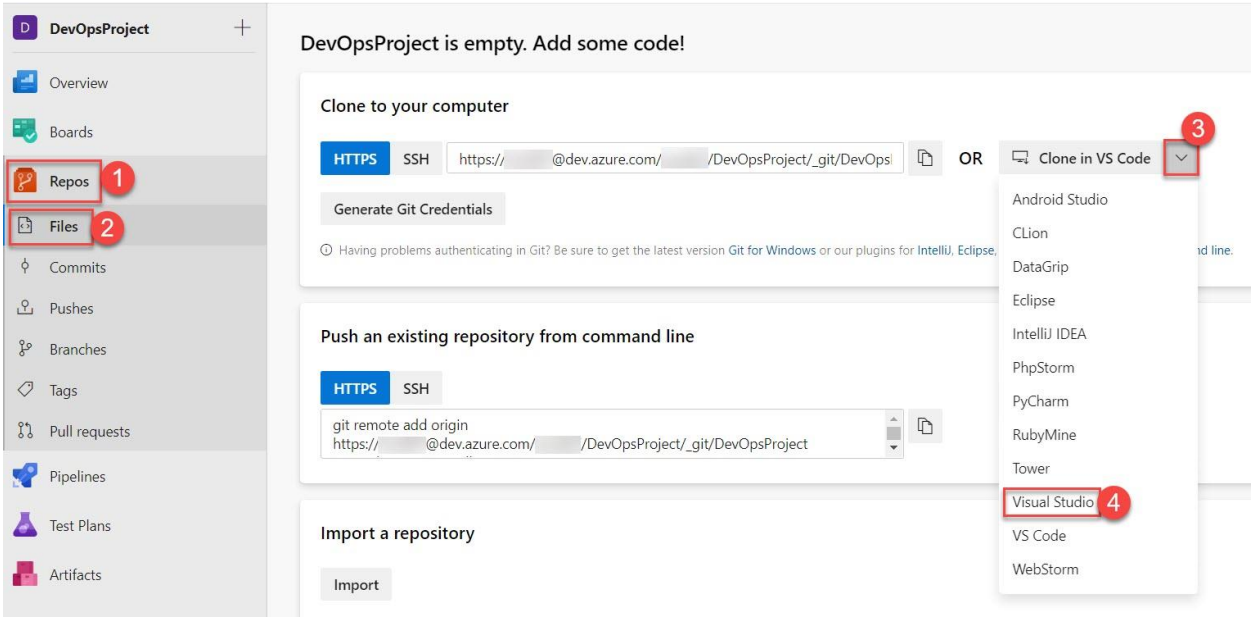
Project Name: **DevOpsProject**

Visibility: **Private**

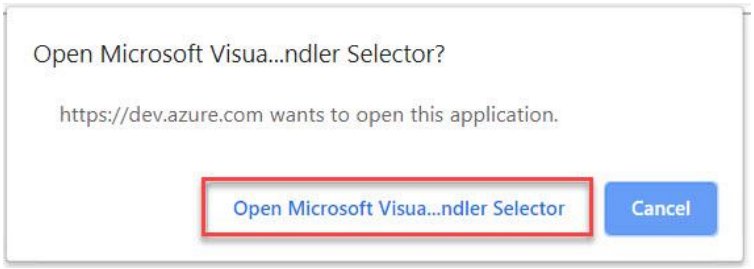
Click on **+ Create project**



Step 3: First Select **Repos** -> **Files** -> Select **Visual Studio**

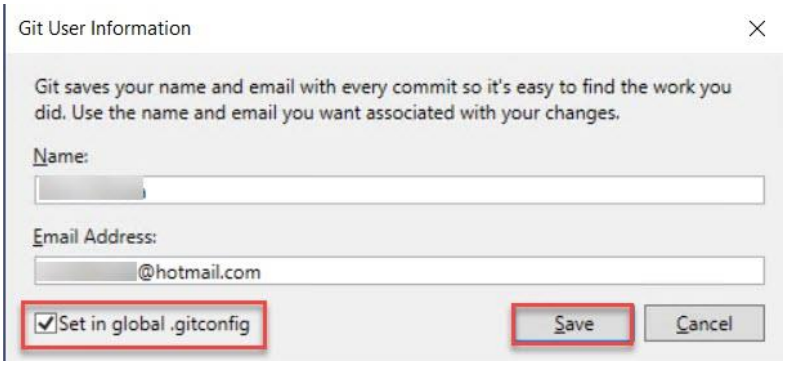


Step 4: Select **Open Microsoft Visual Studio Handler Selector** option

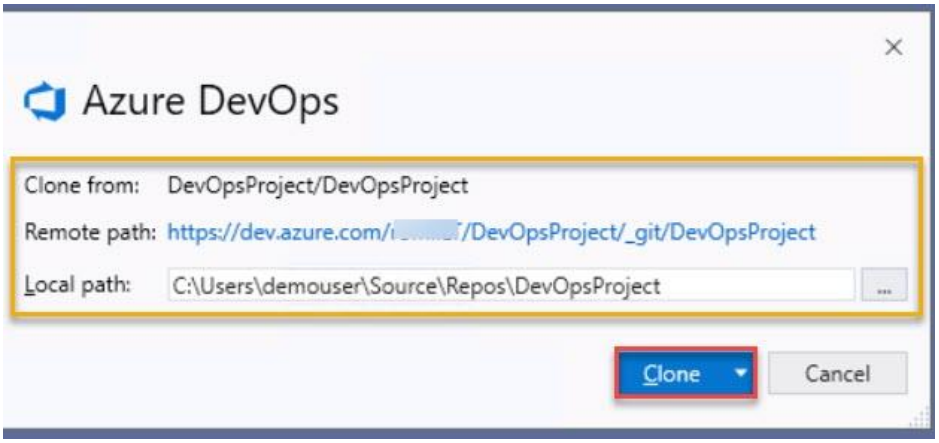


Note: if it will prompt sign in dialog so enter credentials

Go with default settings & click on **Save** button

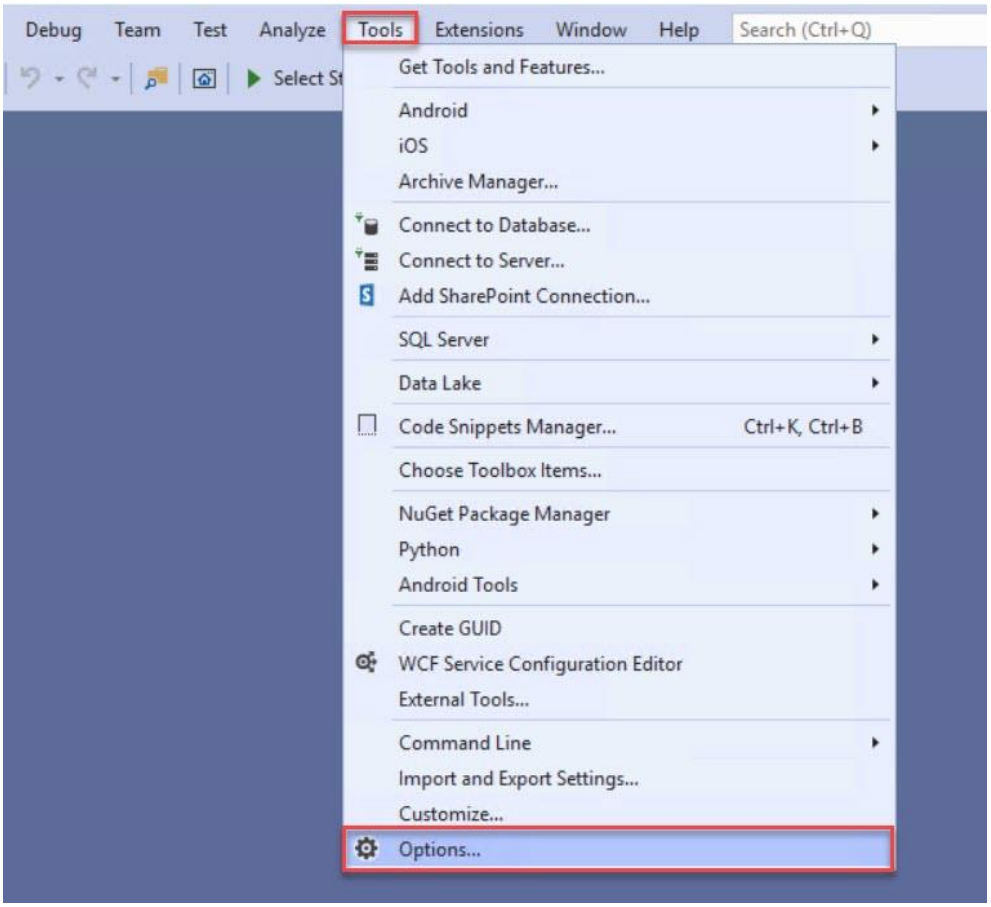


Step 5: Automatic Clone dialog will open and click on **Clone** button



Change Git experience

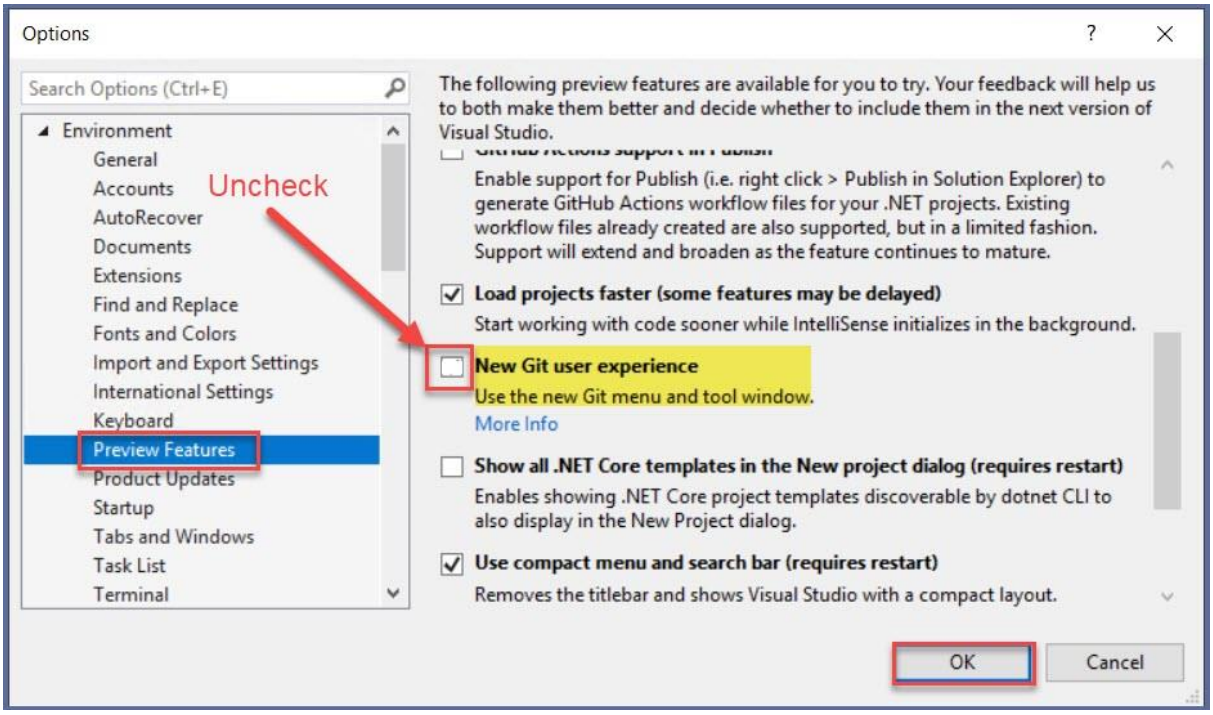
Tools -> Options…



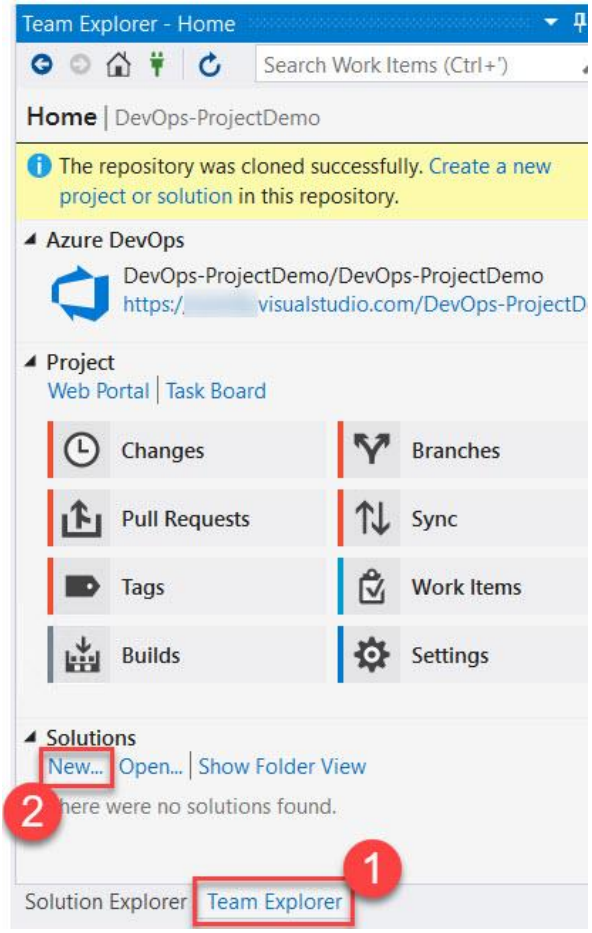
Select **Preview Features**

Uncheck New Git use experience

Click on **OK** button



Step 6: Select **Team Explorer** tab and Create New Project by selecting **New** option



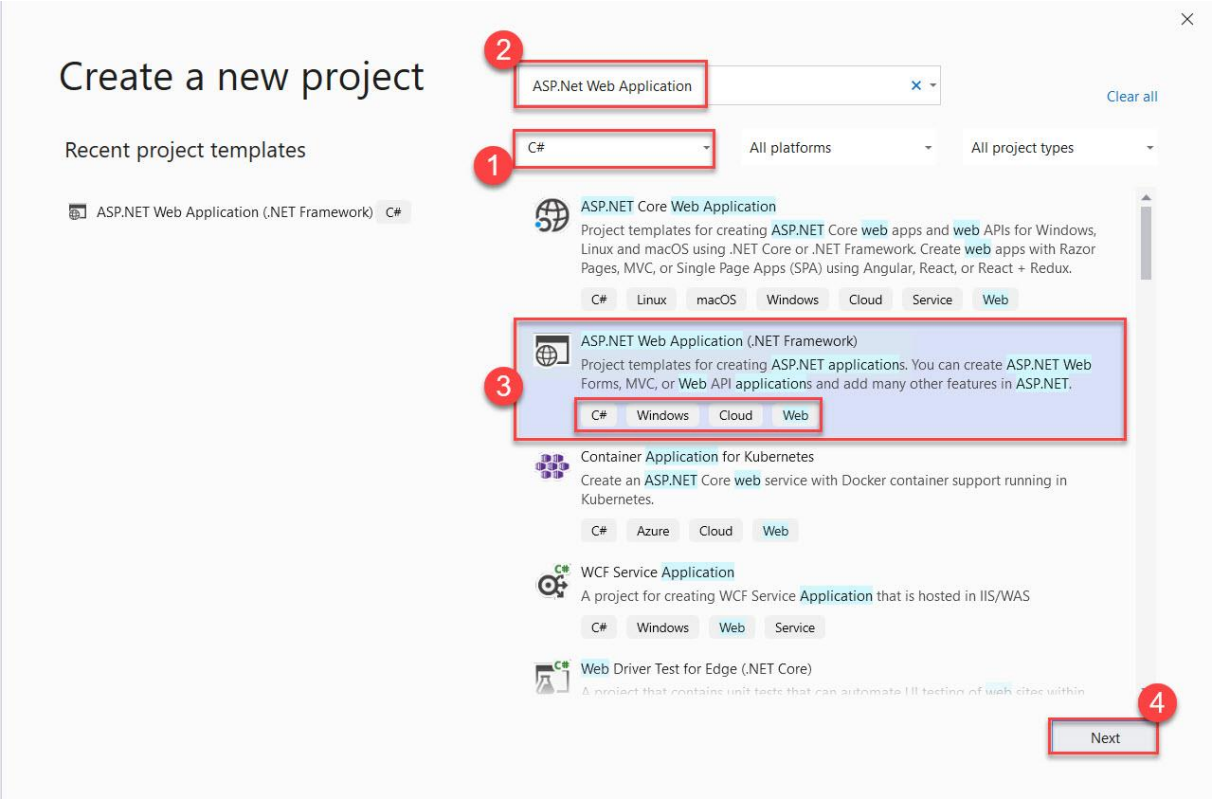
Step 7: Create New Project

Language: **C#**

Search for **ASP.Net Web Application**

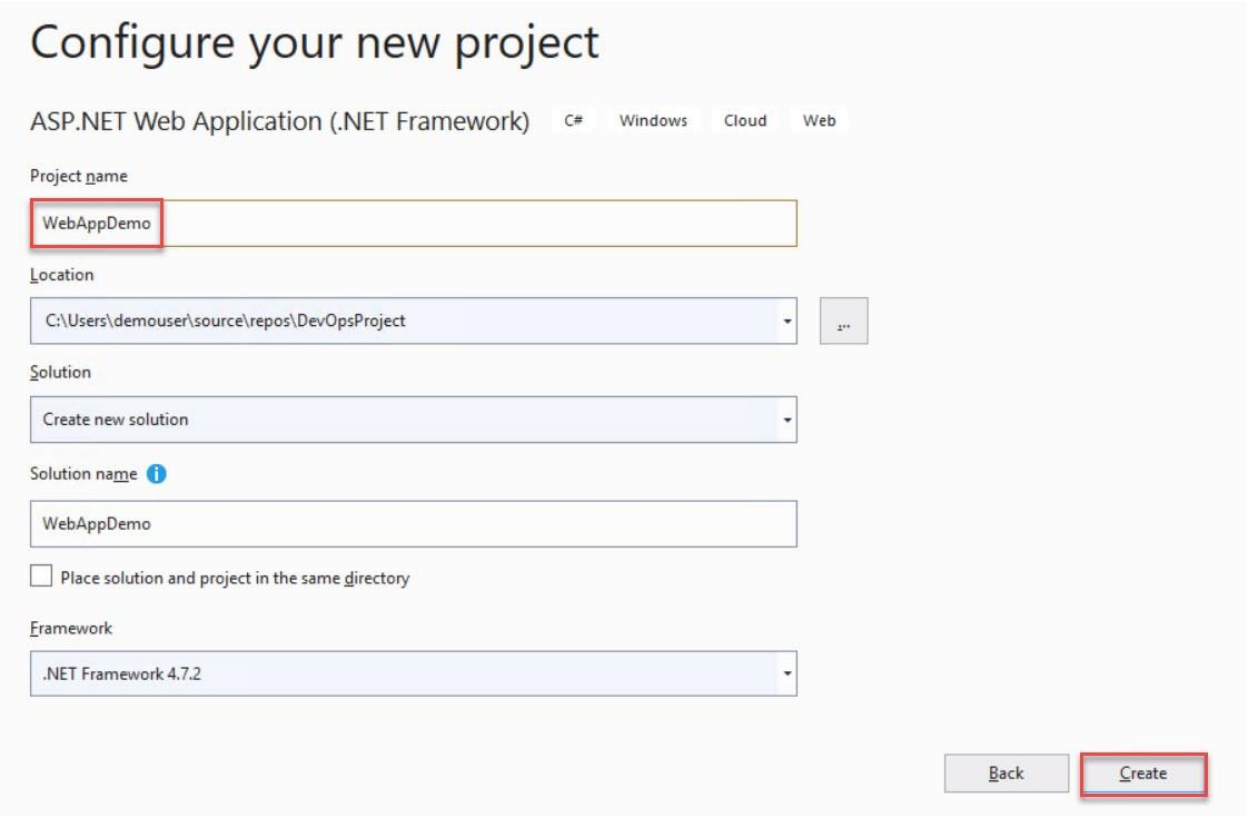
Select **ASP.NET Web Application (.NET Framework)**

Click on **Next** button



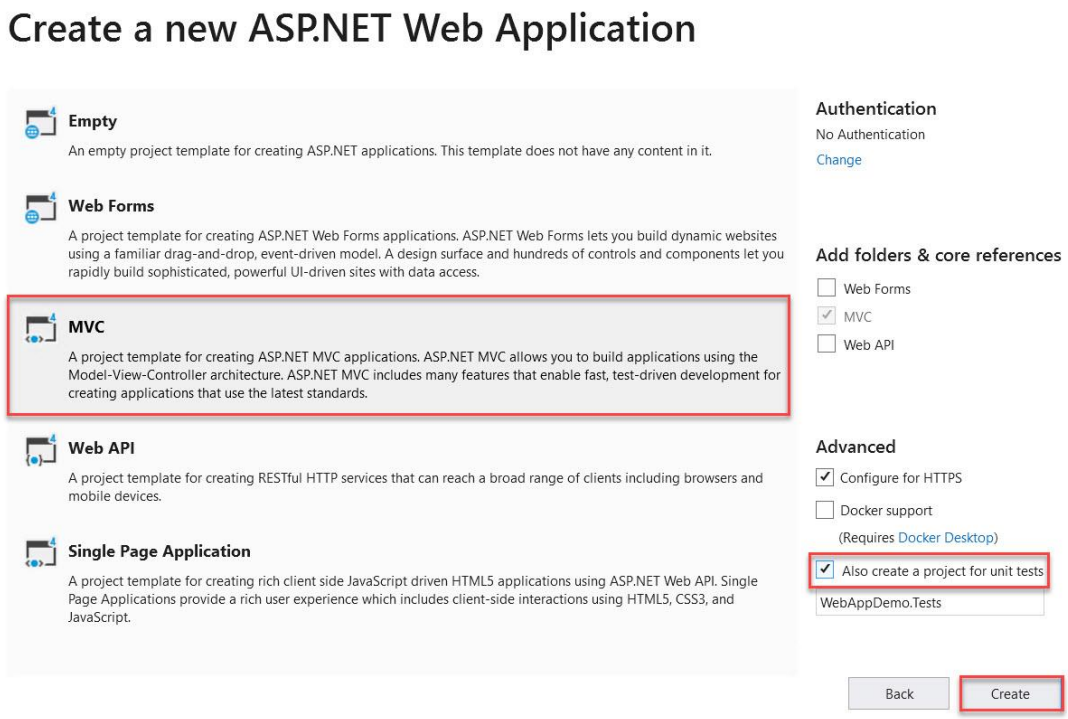
Step 8: Enter Project Name: **WebAppDemo**

Click on **Create** button

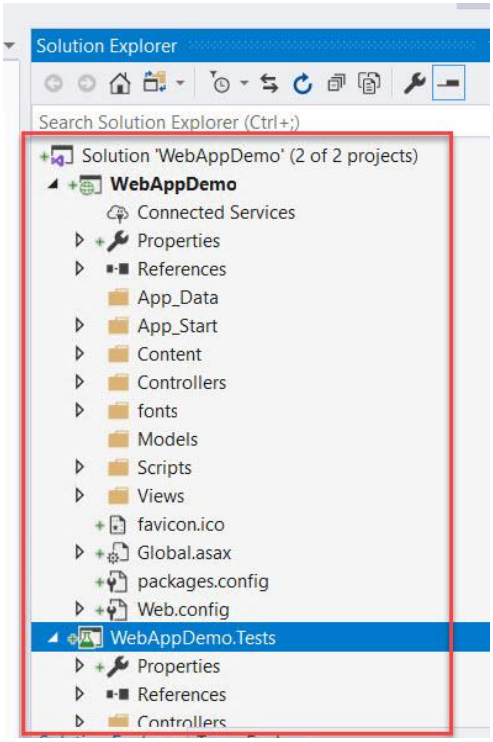


Step 9: Choose **MVC** Template and Add **Test Project**

Click on **Create** button

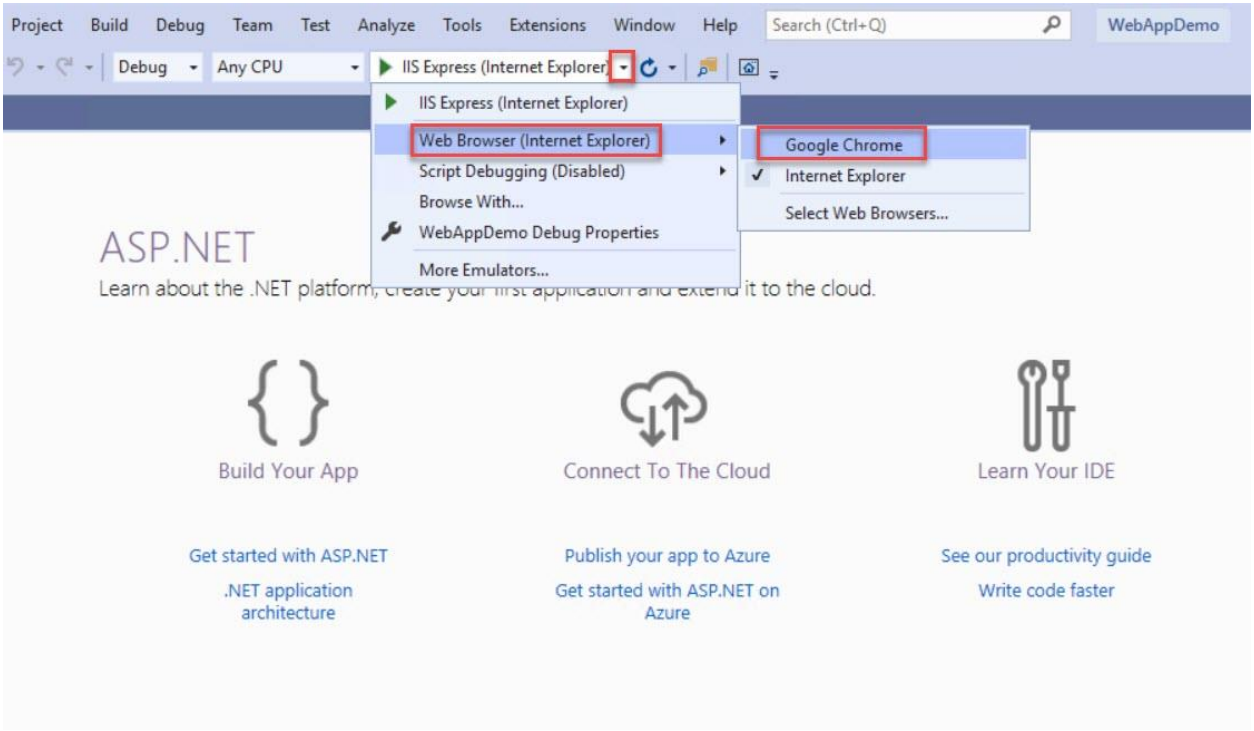


Sample Project loaded as below:

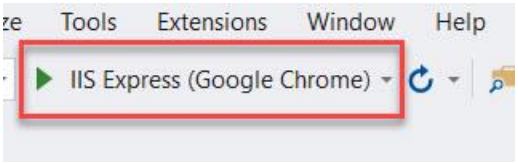


Step 10: Below **Tools** & **Extensions** Menu (header) **IIS Express button** available

Now time to Run project locally. Change Browser if you require.

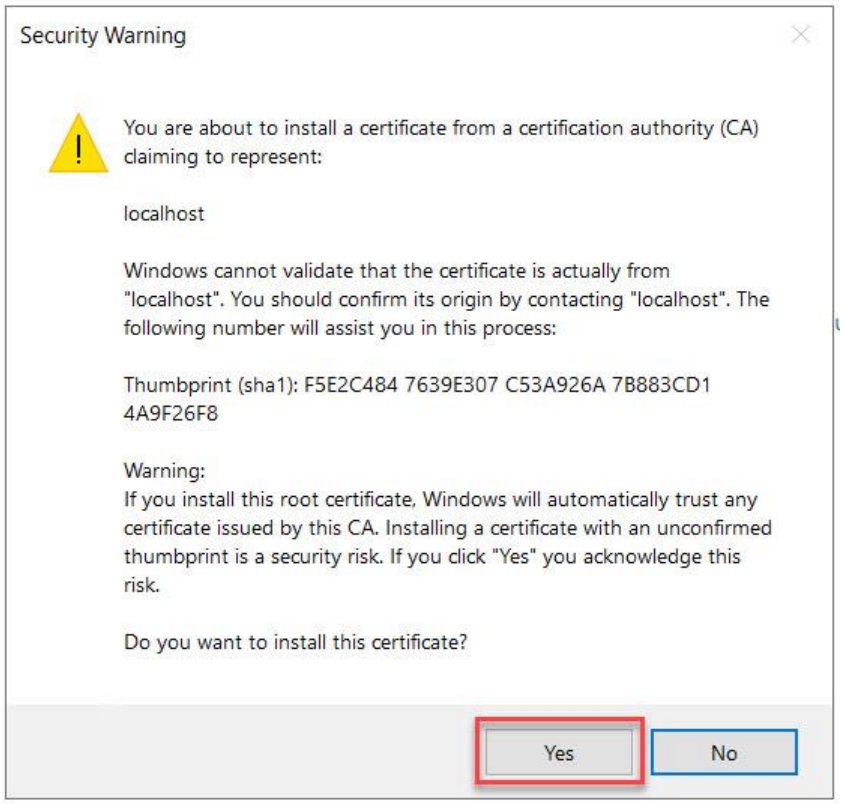


Click on **IIS Express (Google Chrome)** button

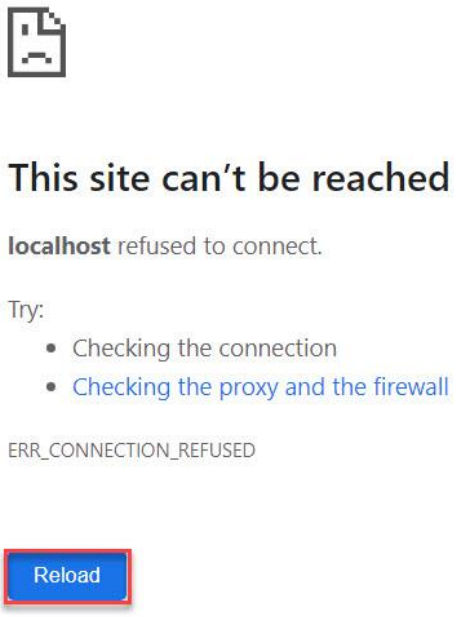


Click on **Yes** button

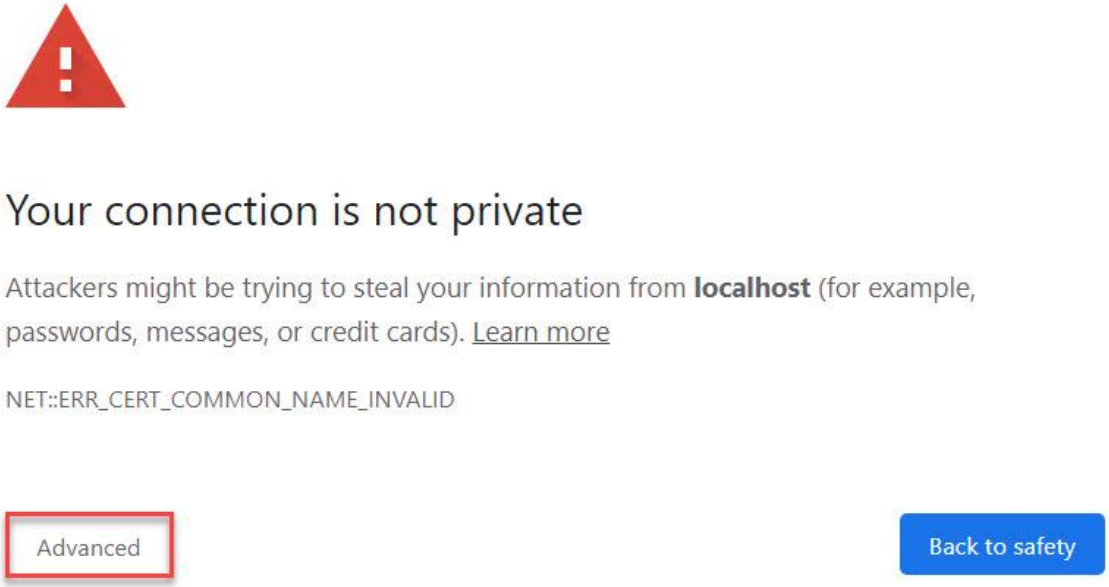
Click on **Yes** button



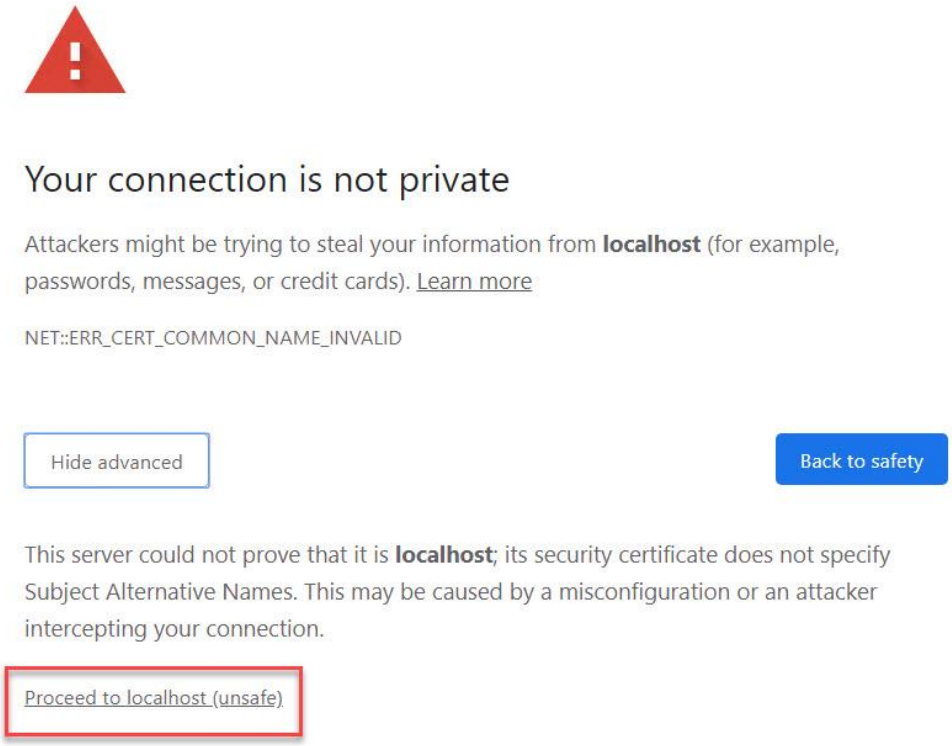
Click on **Reload** button



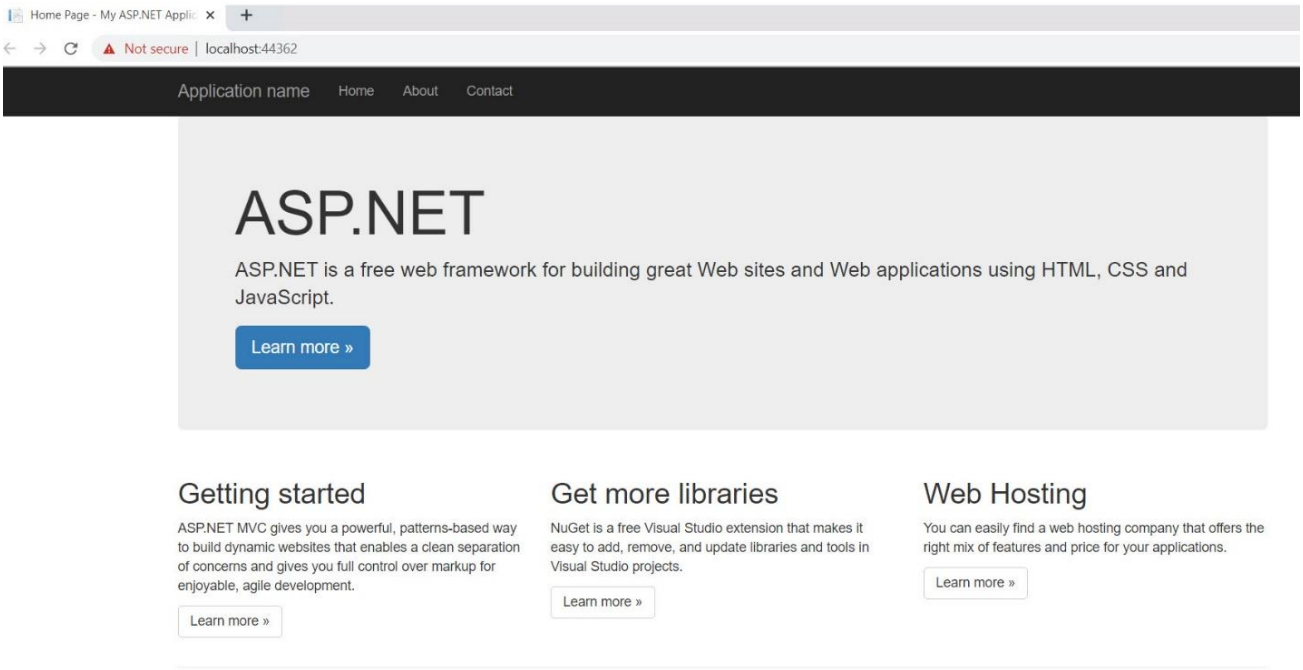
Click on **Advanced** button



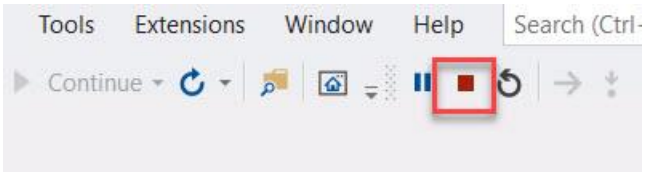
Click on **Process to localhost (unsafe)** option



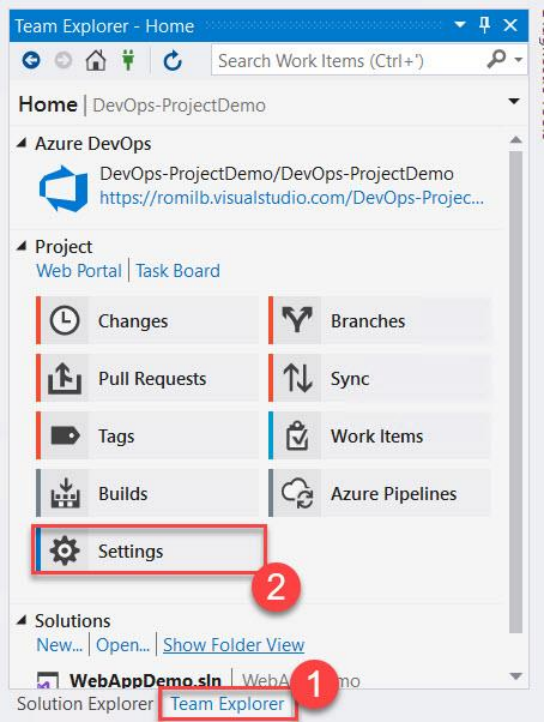
Sample MVC Application will run as below:



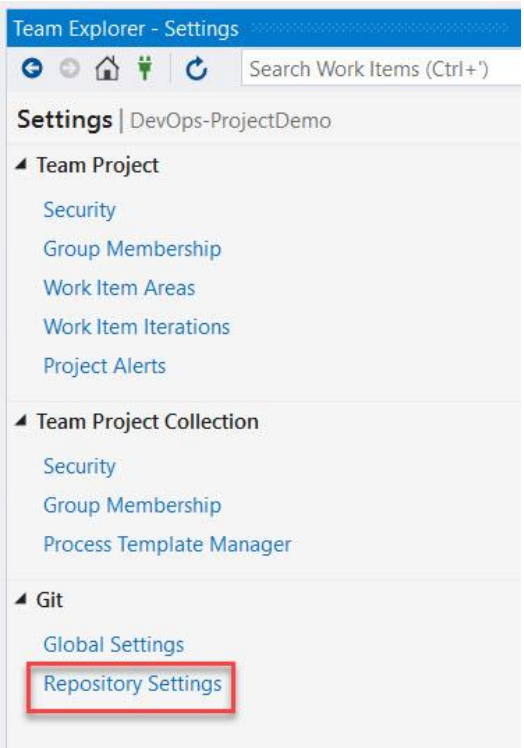
Step 11: Now stop the running project



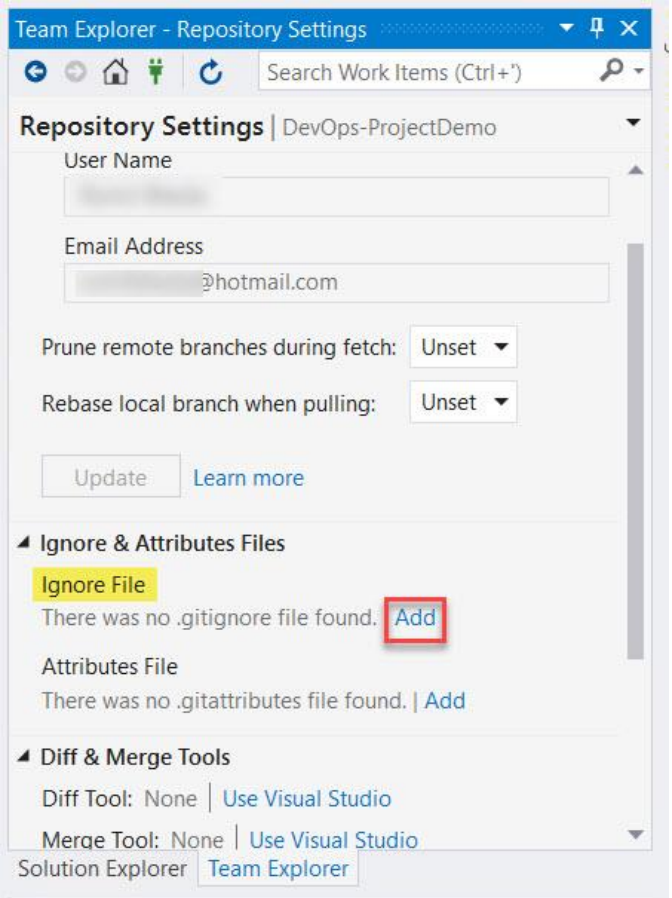
Step 12: Once again, Navigate to **Team Explorer** tab and select **Settings** option



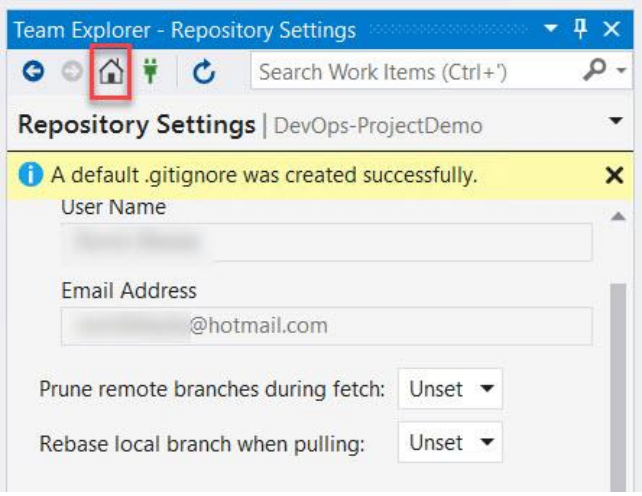
Step 13: Select **Repository Settings** under **Git** option



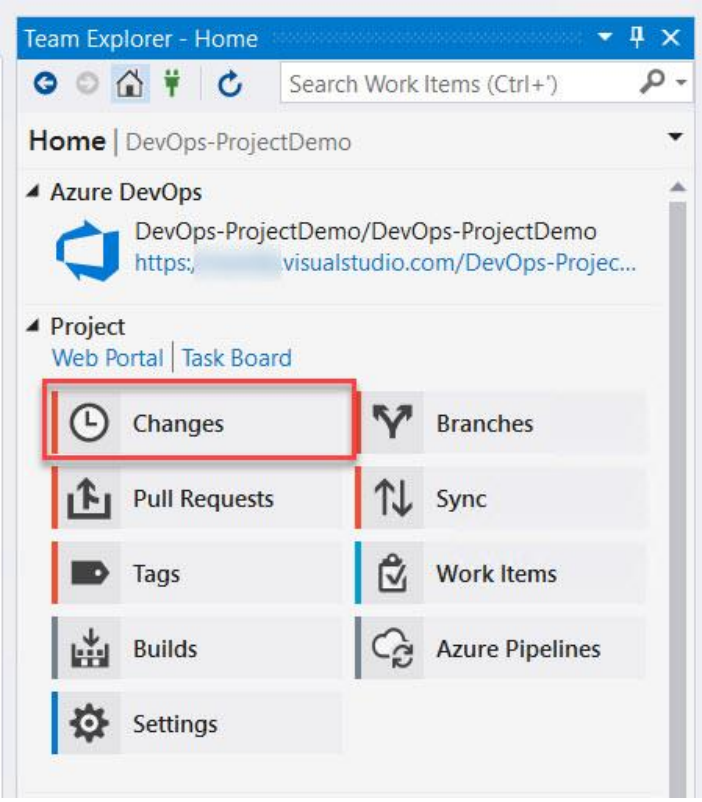
Step 14: Click on **Add** option under **Ignore File**



Step 15: Click on **Home** option

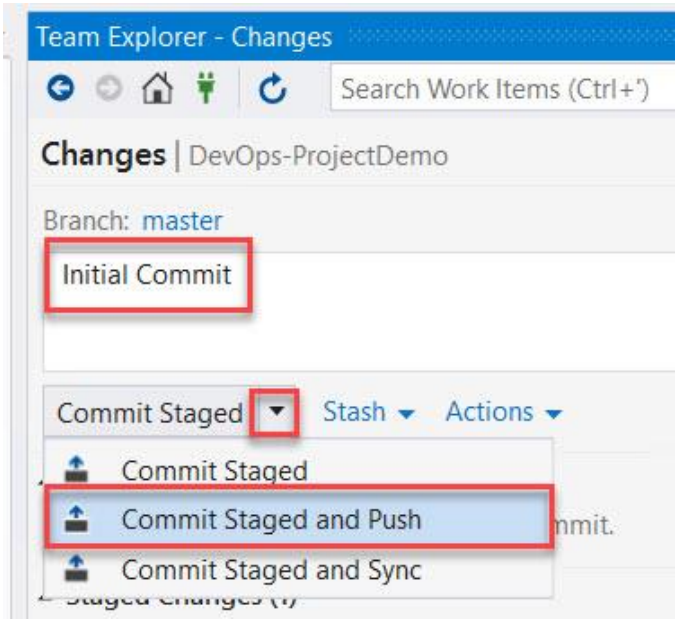


Step 16: Click on **Changes** option

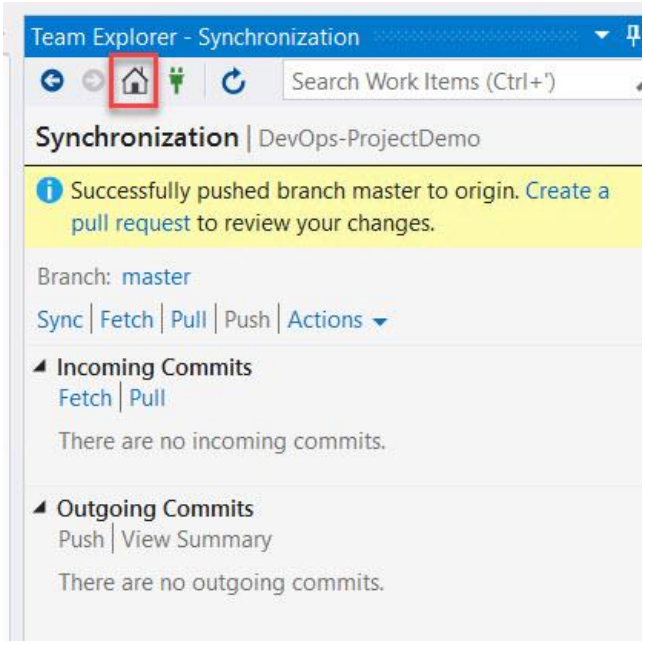


Step 17: Enter message: **Initial Commit**

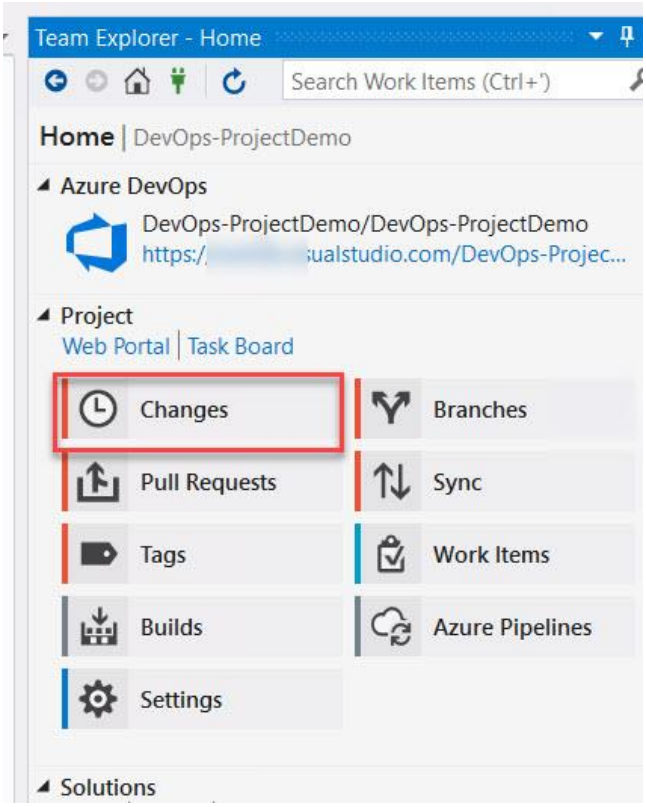
Choose **Commit Staged and Push** option



Step 18: Click on **Home** option

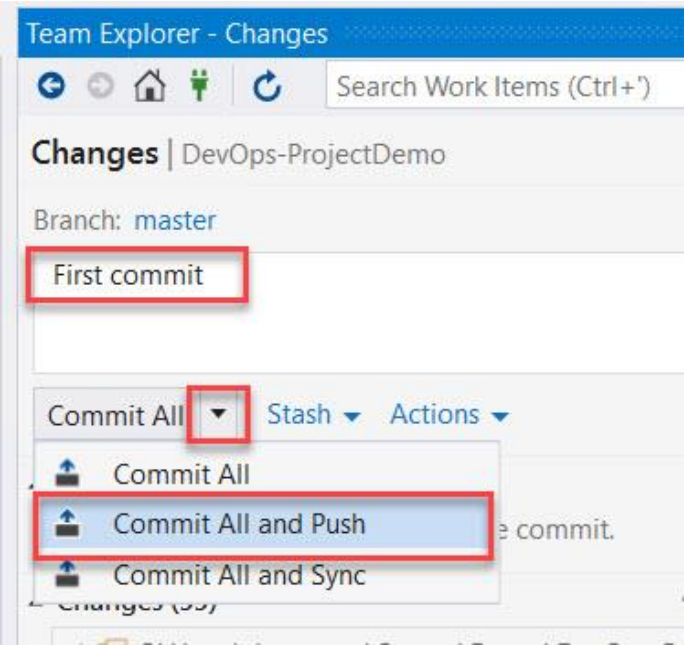


Step 19: Once again, Select **Changes** option

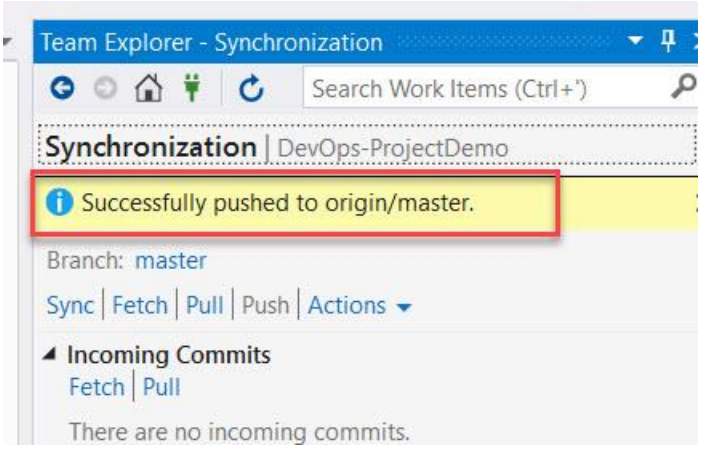


Step 20: Commit Message: **First Commit**

Choose **Commit All and Push** option

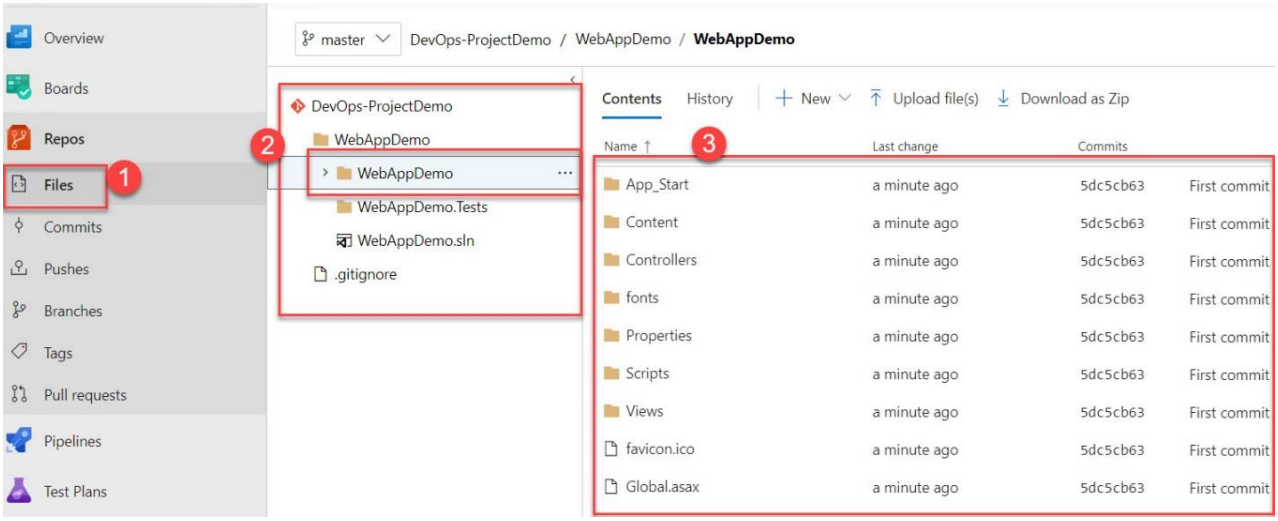


Check the status "**Successfully pushed to origin/master**"



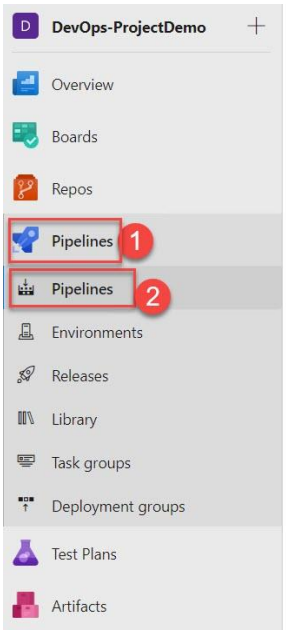
Step 21: Navigate to **Azure DevOps Project**

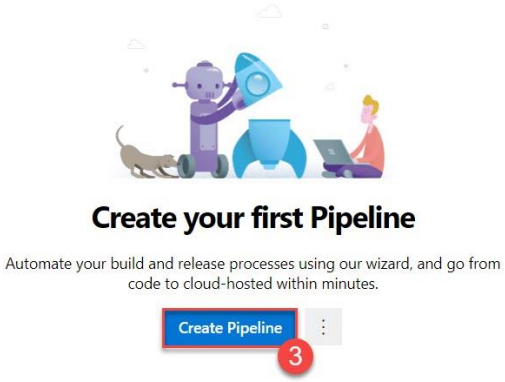
Click on **Files** and check the Project available here.



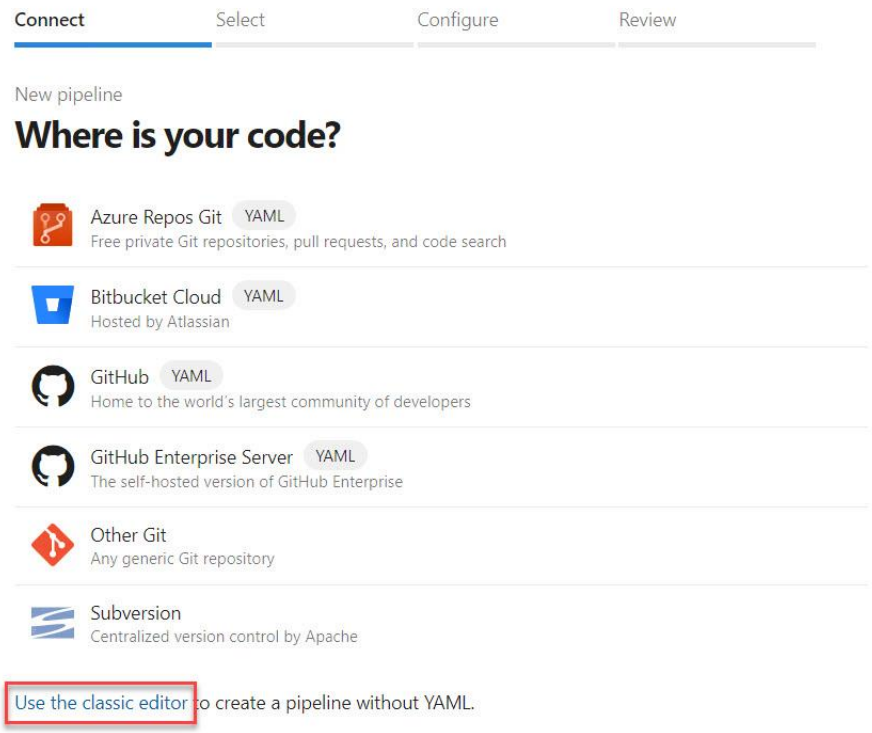
Lab 2: Continuous Integration

Step 22: **Create Pipeline** under **Pipelines** option

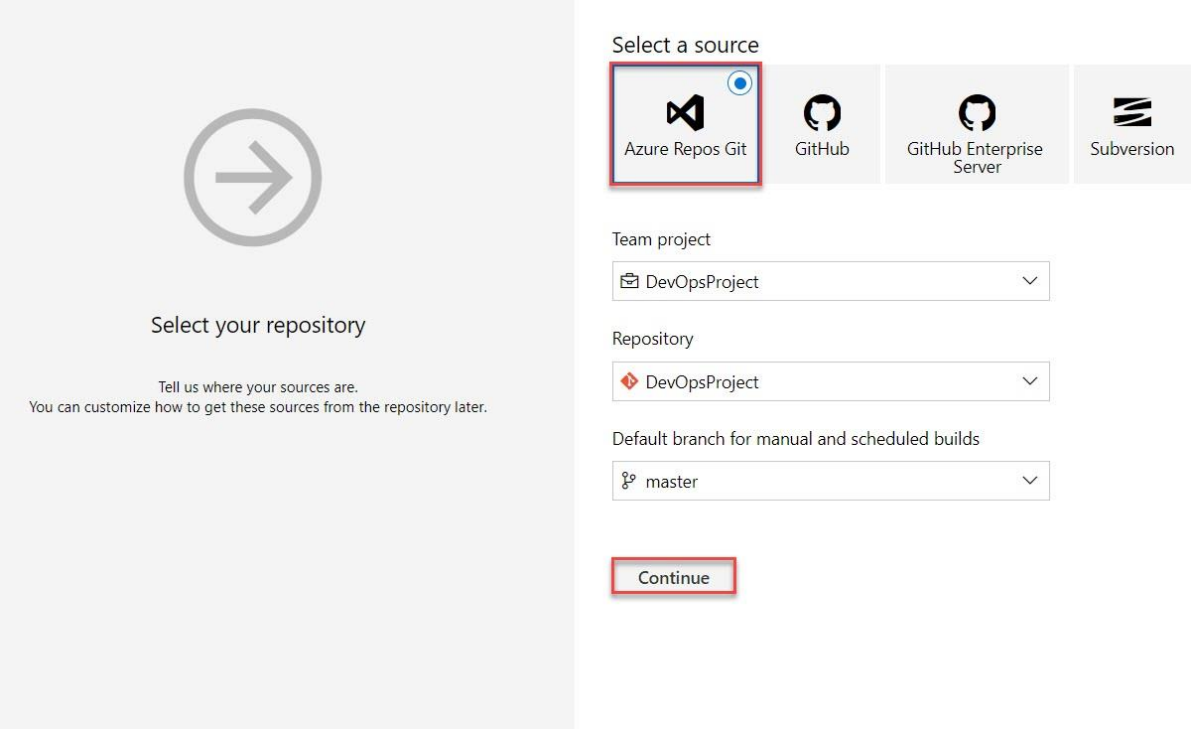




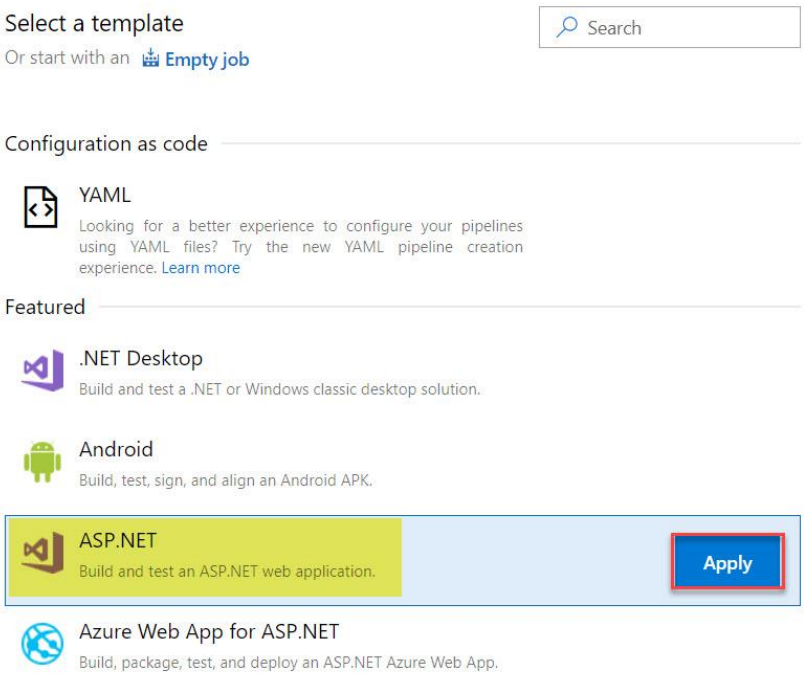
Step 23: Click on **Use the classic editor** option



Step 24: Select **Azure Repos Git** as a Source and Click on **Continue** button.



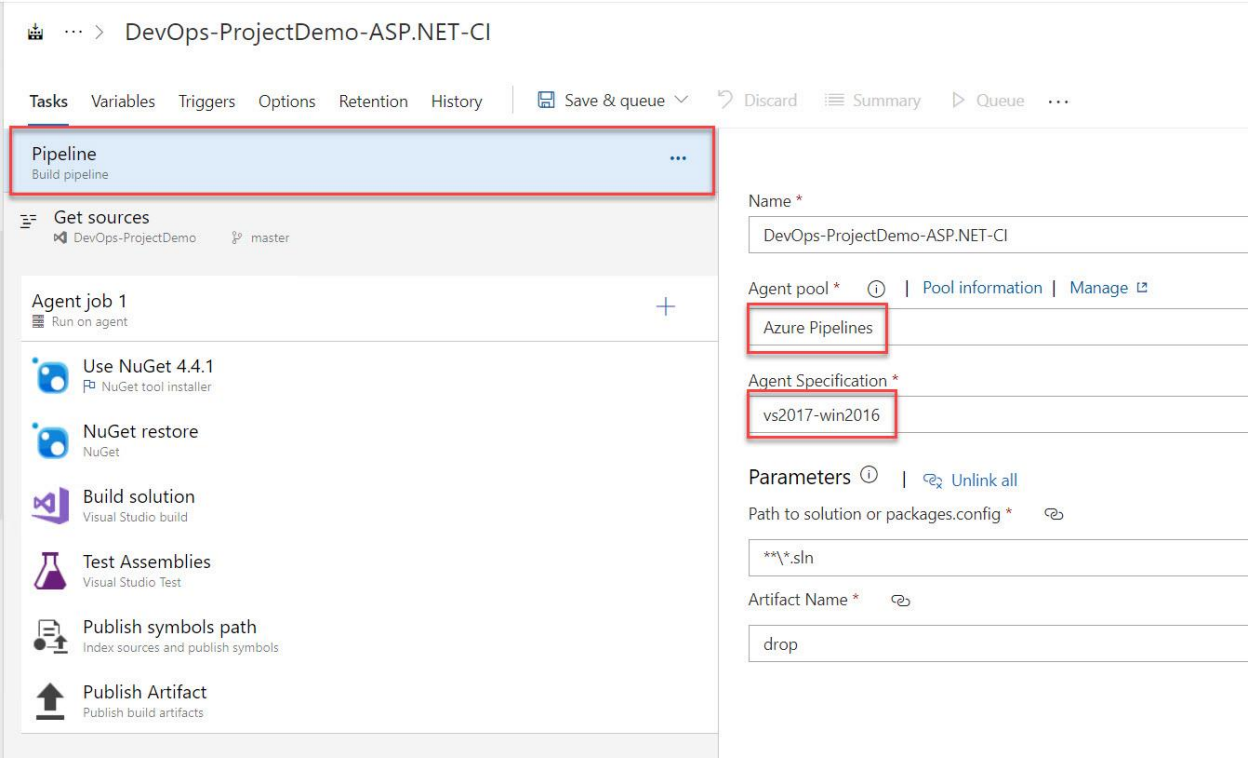
Step 25: Choose template according to your project. Ex. **ASP.NET**



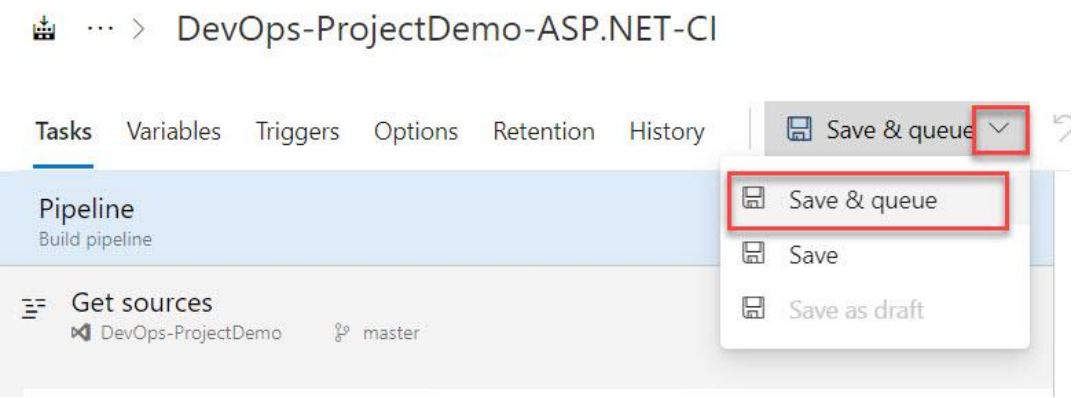
Step 26: Select Pipeline

Agent pool: **Azure Pipelines**

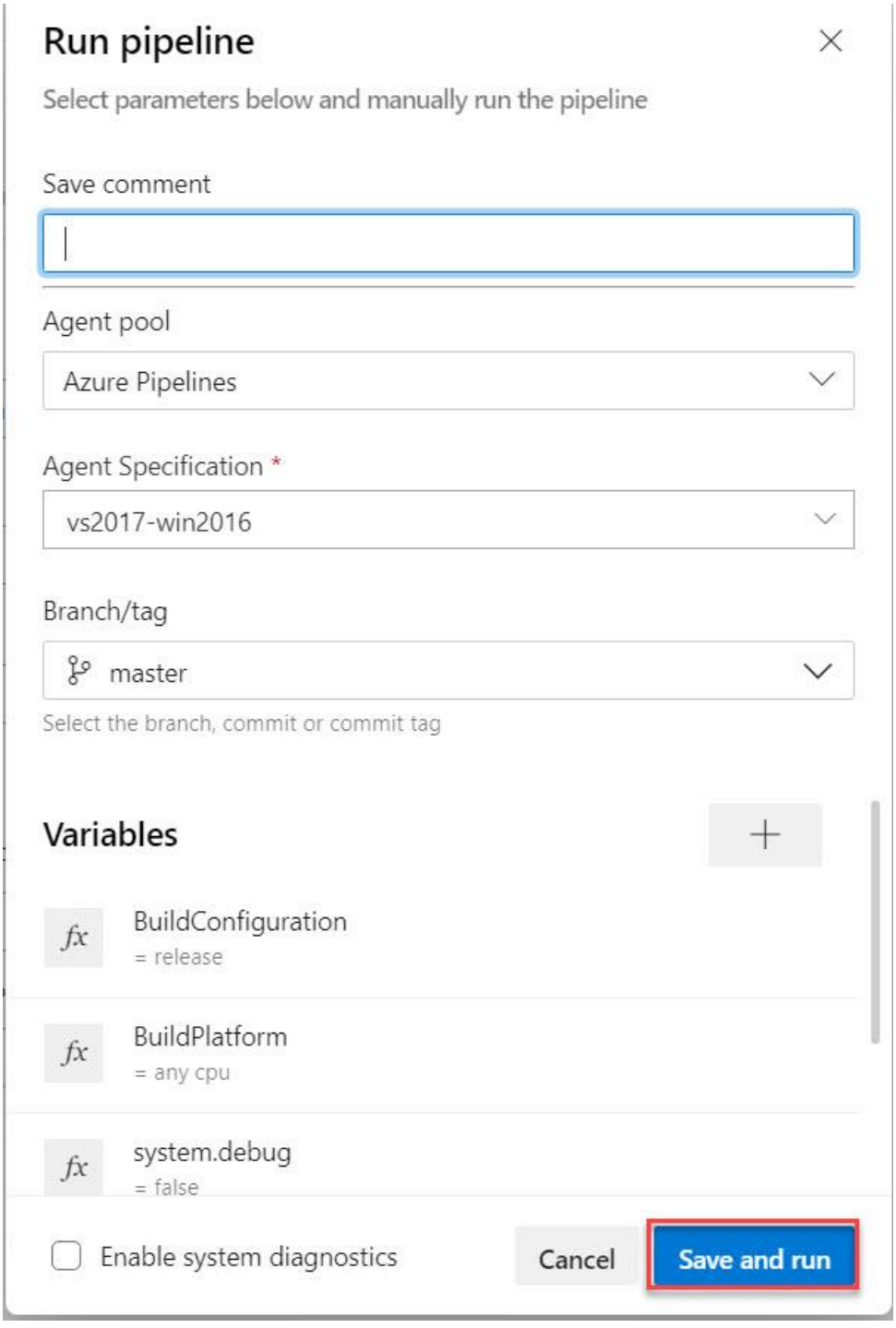
Agent Specification: **vs2017-win2016**



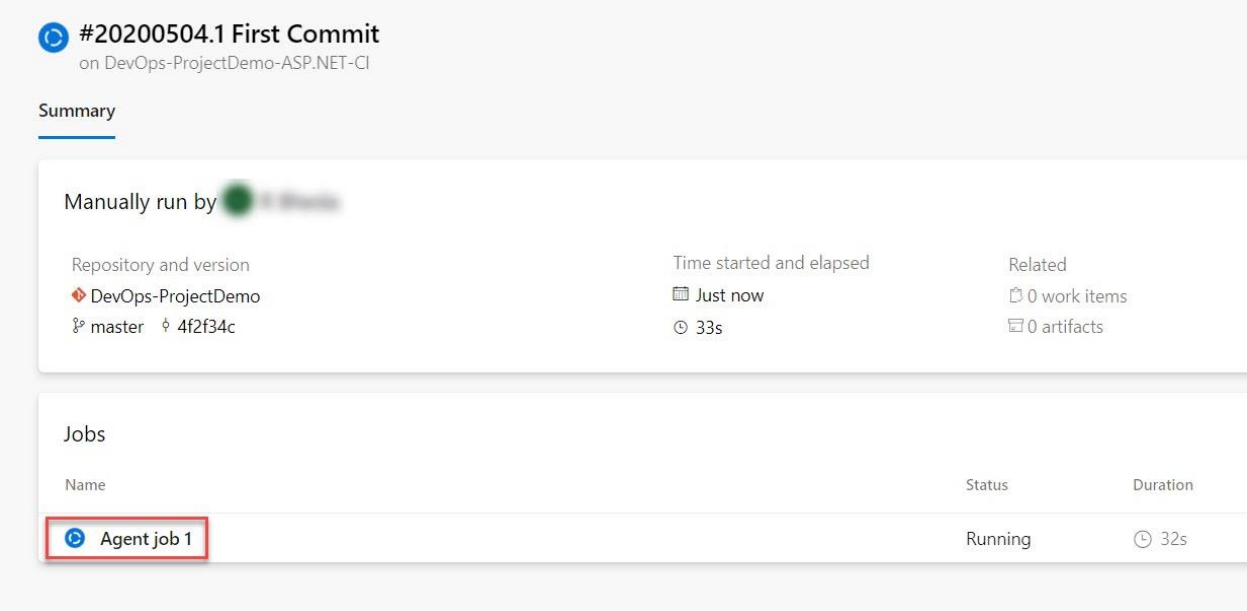
Step 27: Select **Save & queue** button



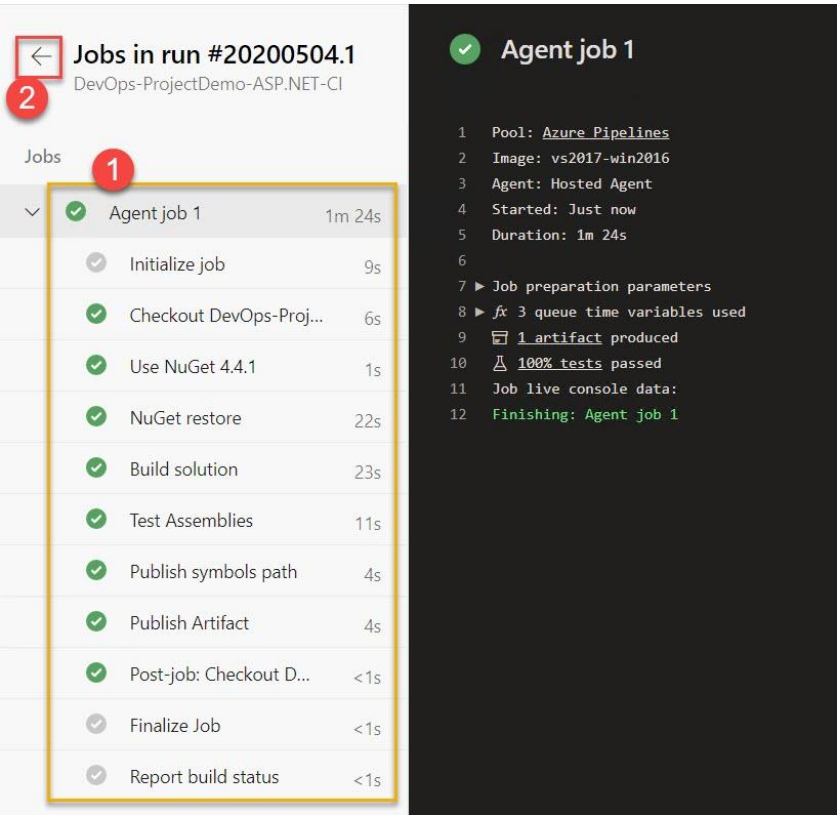
Step 28: Click on **Save and Run** button



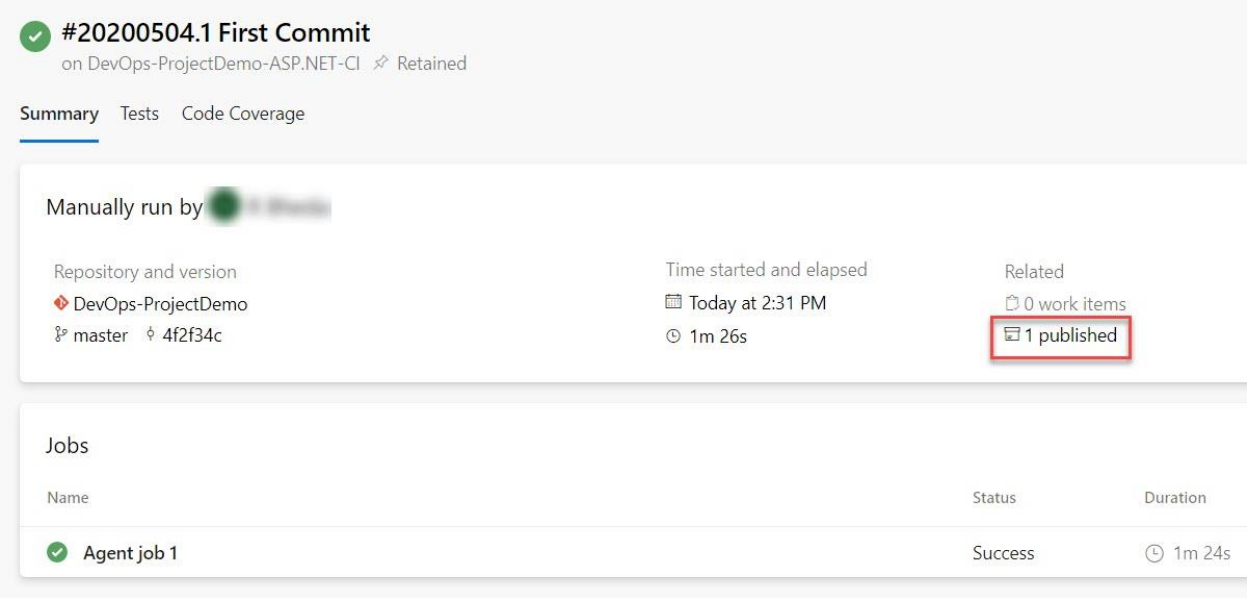
Click on **Agent job 1**



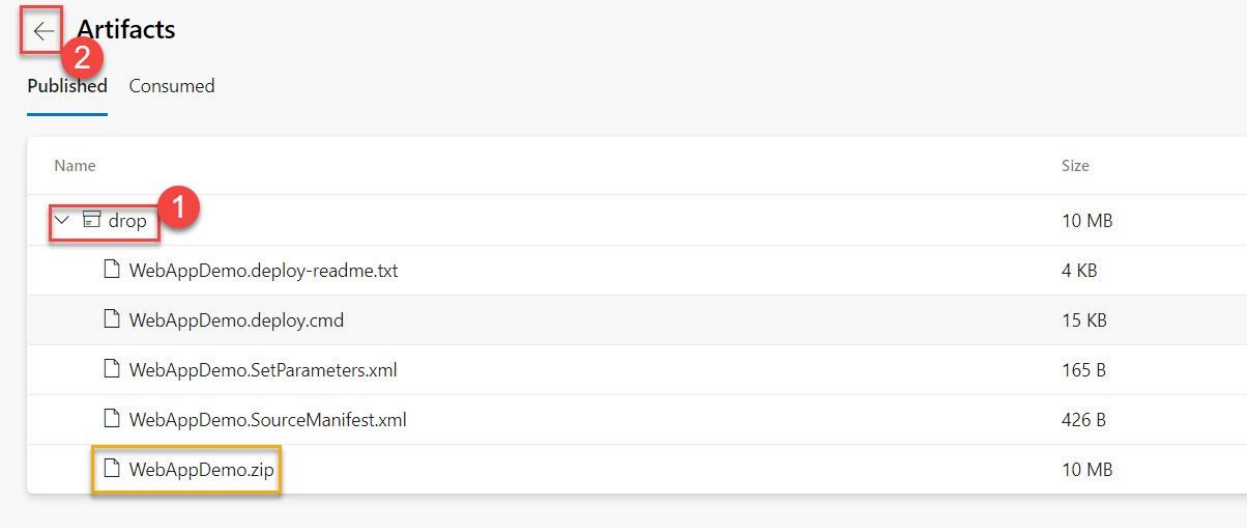
Wait few couple of minutes to complete build pipeline. If completed click on **back arrow** option



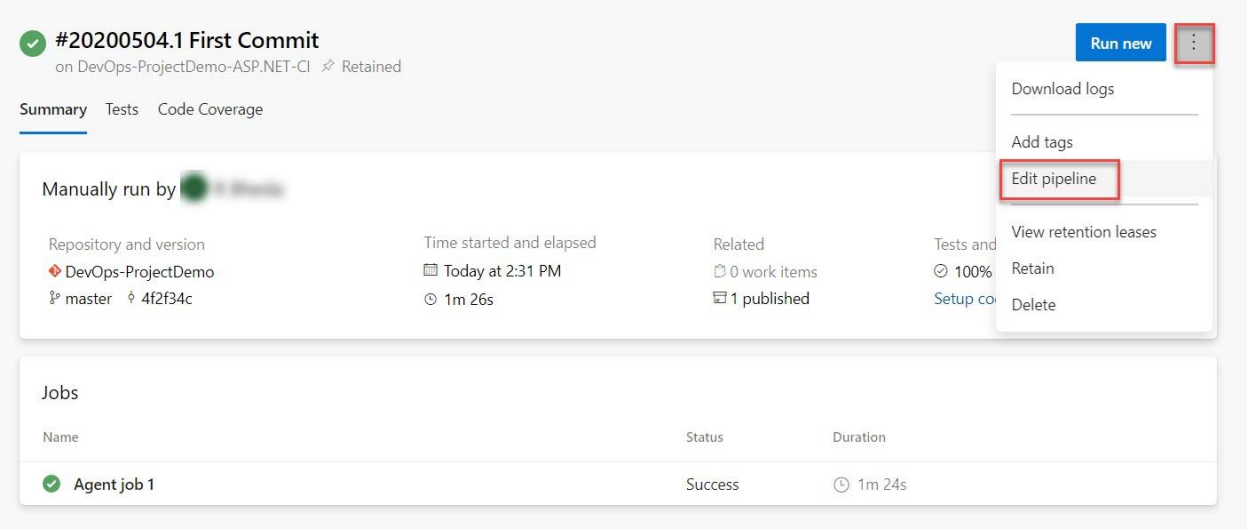
Step 29: Click on **1 published** to check artifact



Expand drop folder and you can see **WebAppDemo.zip** file available

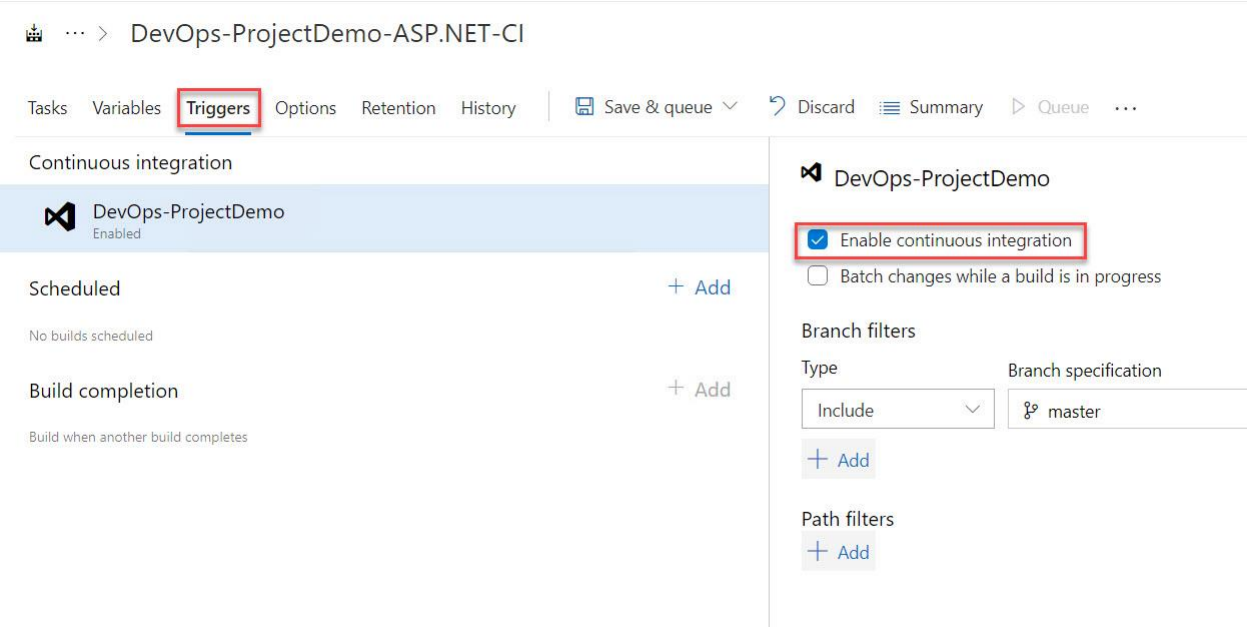


Step 30: Select **Edit pipeline** option



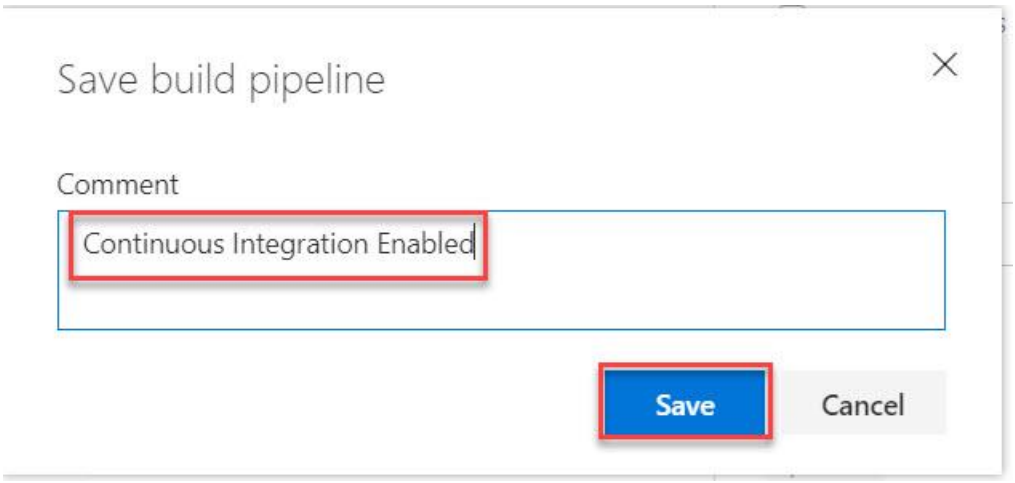
Step 31: Select **Triggers** tab

Enable Continuous integration option



Step 32: Now select **Save** button

Comment: **Continuous Integration Enabled**

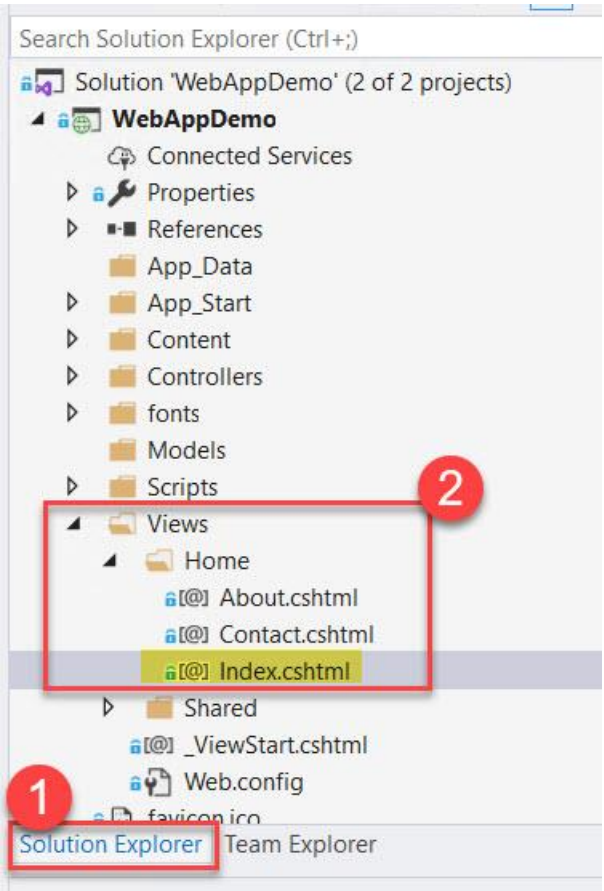


Step 33: **Navigate to Visual Studio Project**

Select **Solution Explorer** tab

Open **Index.cshtml** page

Views/Home/Index.cshtml



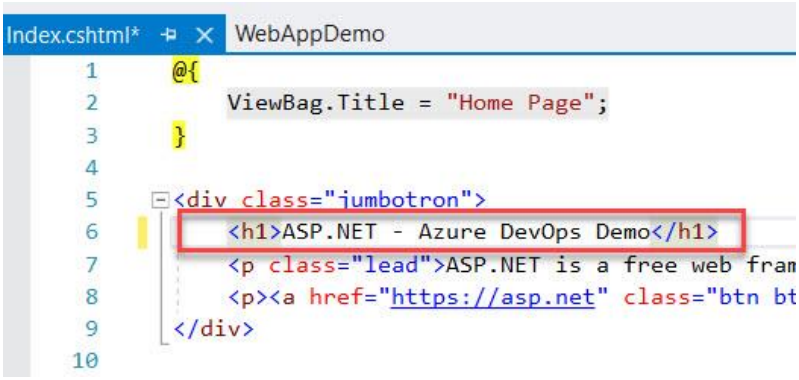
Step 34: **Update text as below:**

From:

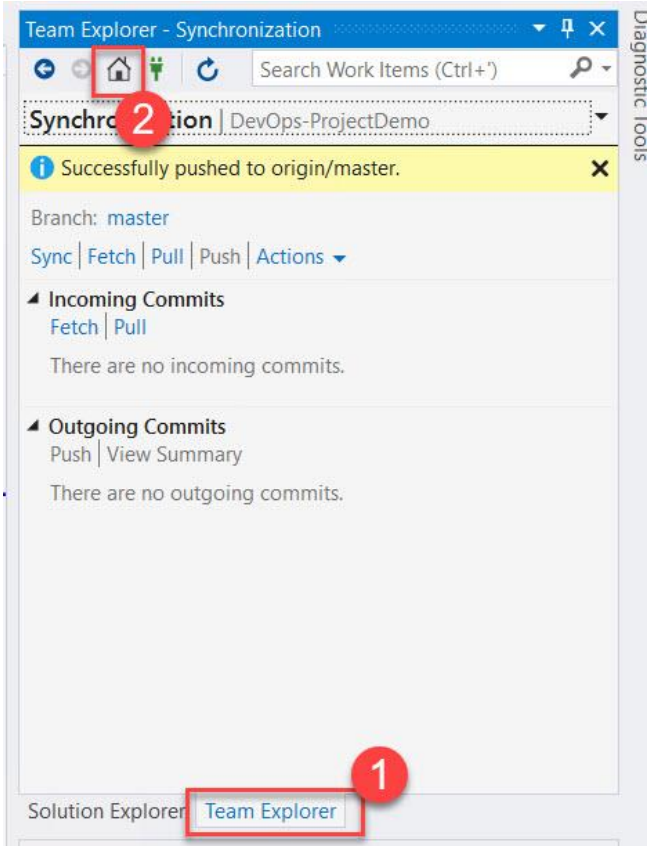
```
<h1>ASP.NET</h1>
```

To:

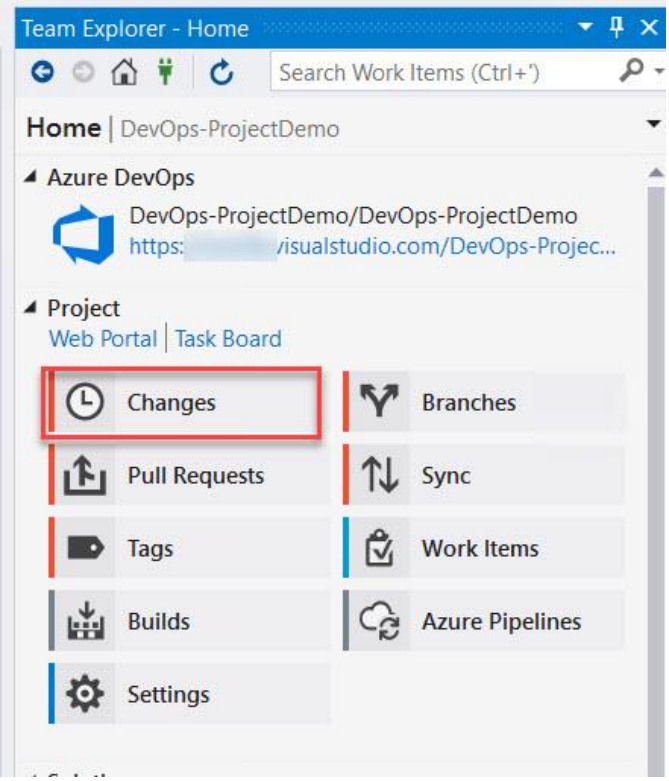
```
<h1>ASP.NET - Azure DevOps Demo</h1>
```



Step 35: Select **Team Explorer** tab and Select **Home** option

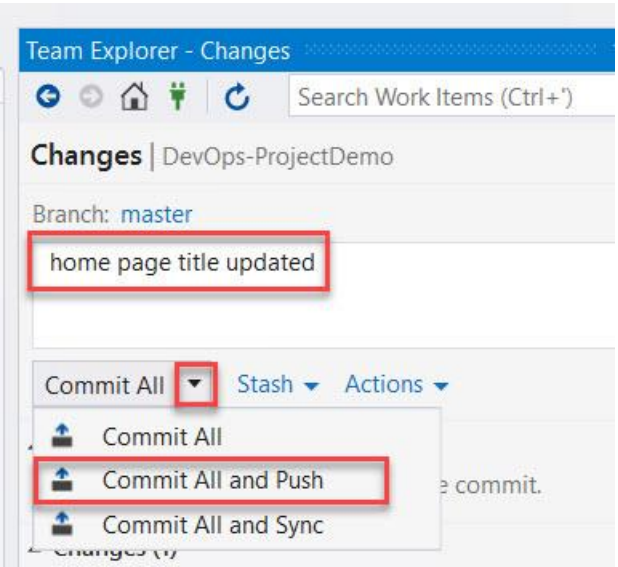


Step 36: Select **Changes** option

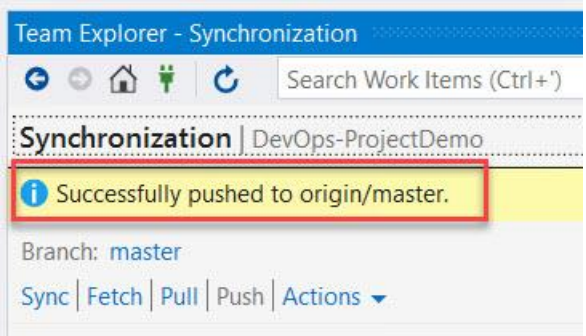


Step 37: Message: **Home page title updated**

Choose **Commit All and Push** option



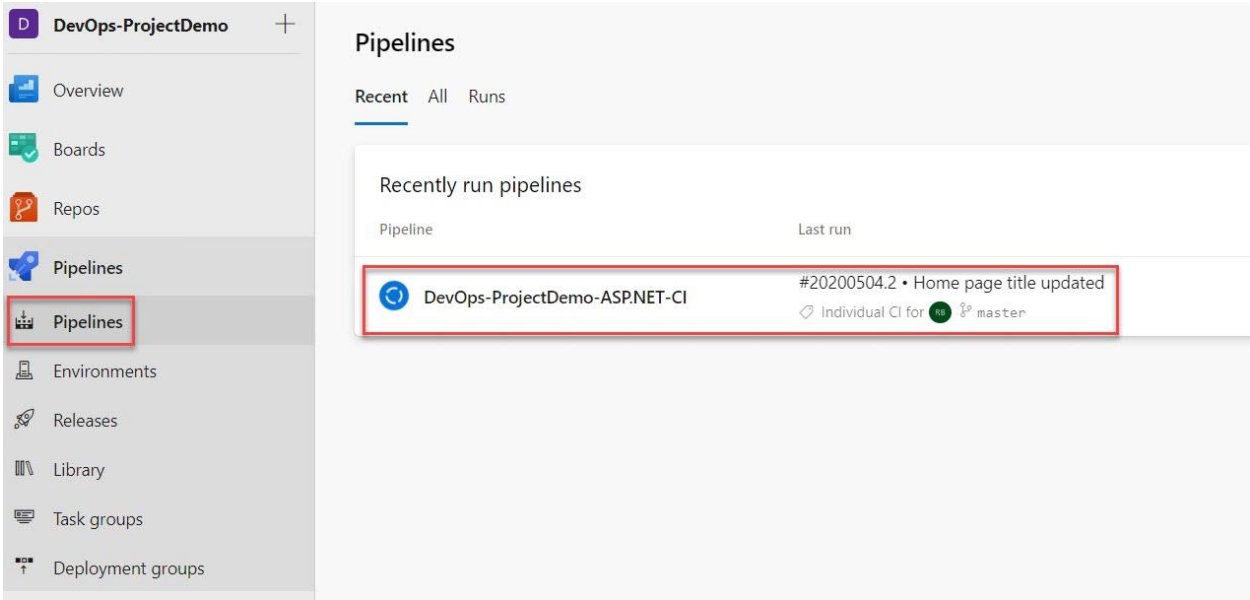
Successfully pushed to origin/master message will display.



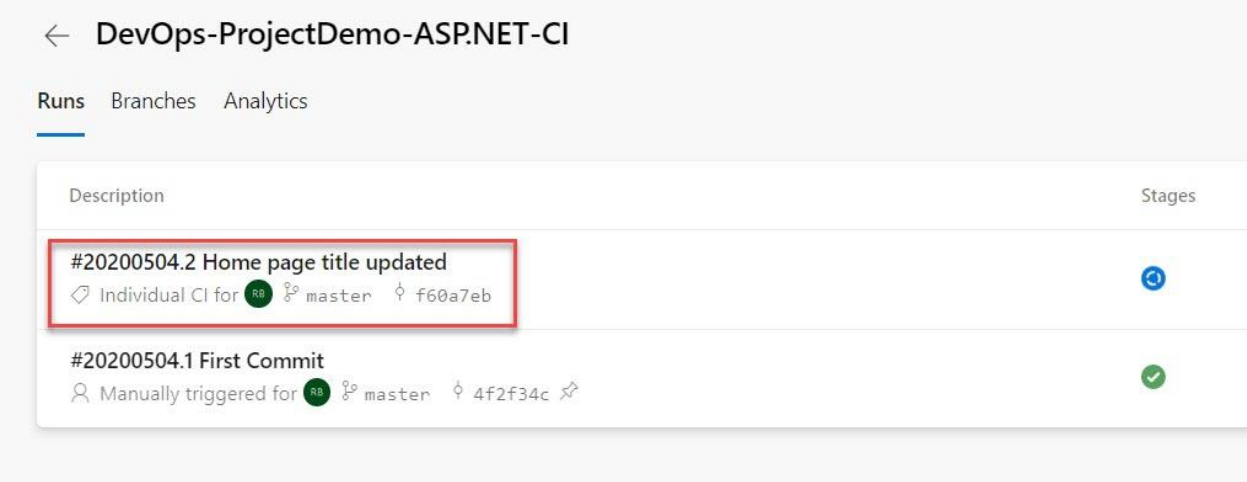
Step 38: Navigate to **Azure DevOps Portal**

Select **Pipelines**

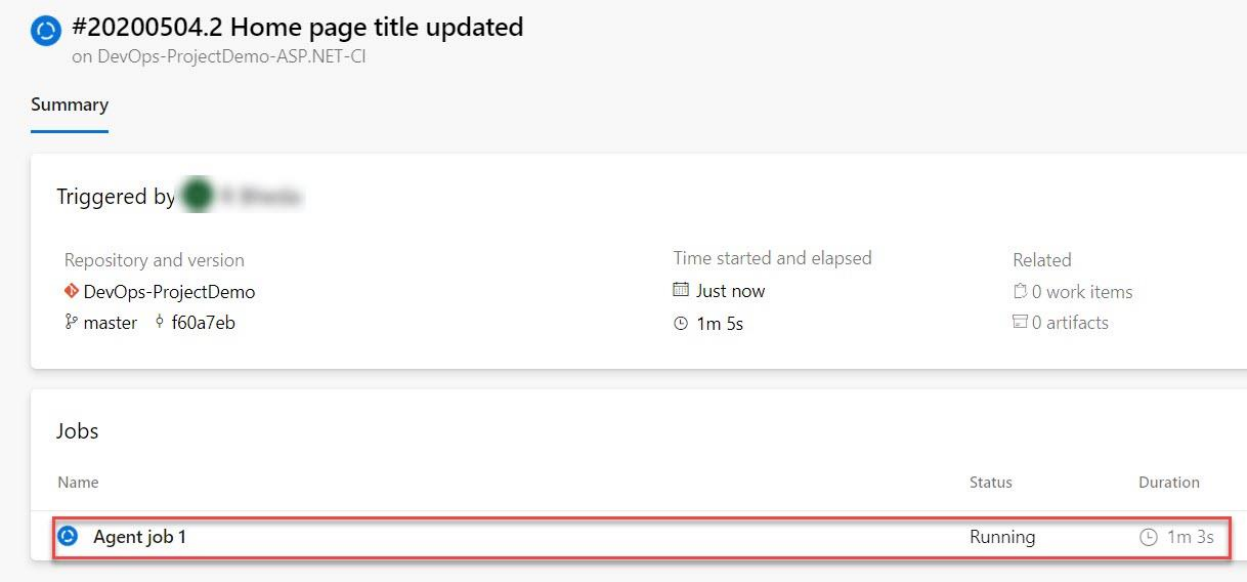
Click on running Pipeline



Select recent build Ex. **2020XXXX.2 Home page title updated**

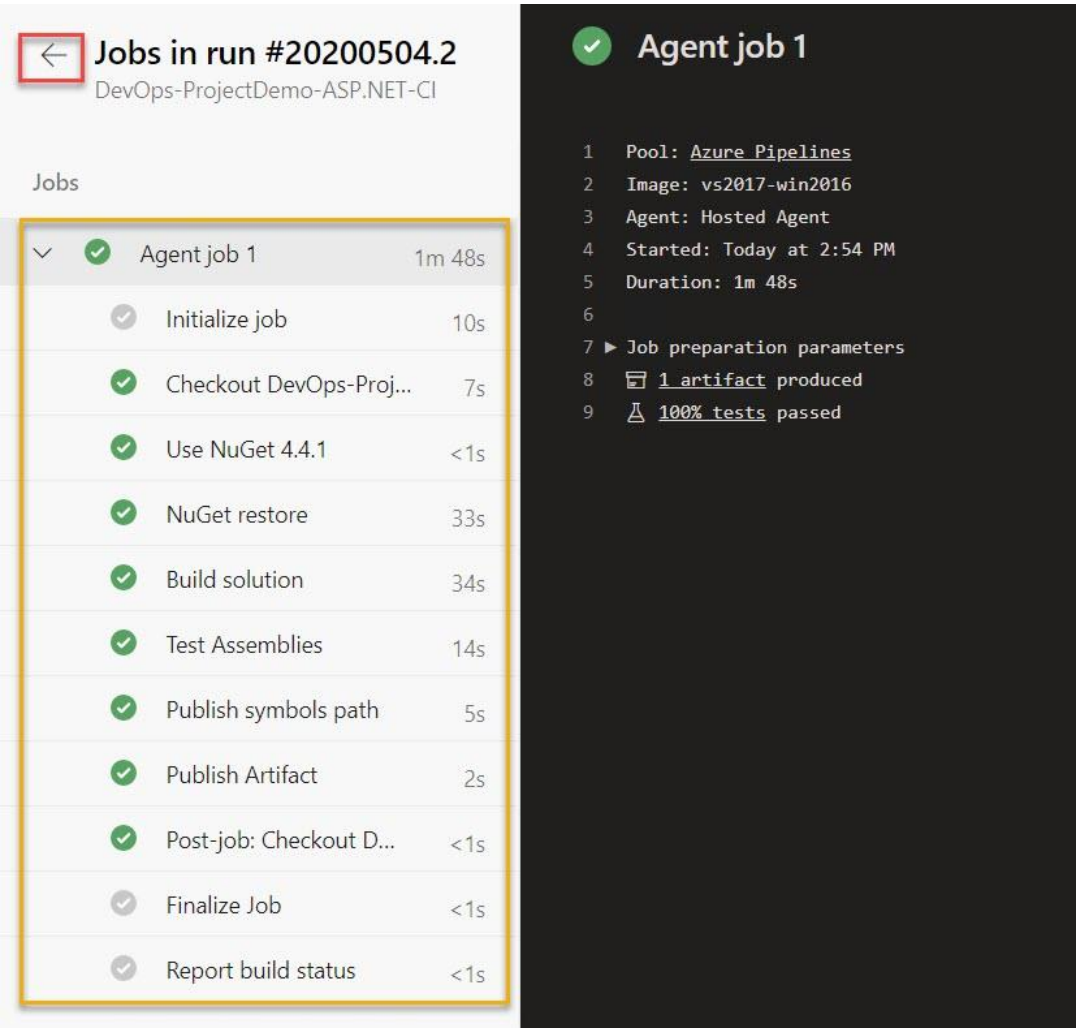


Click on **Agent job 1**



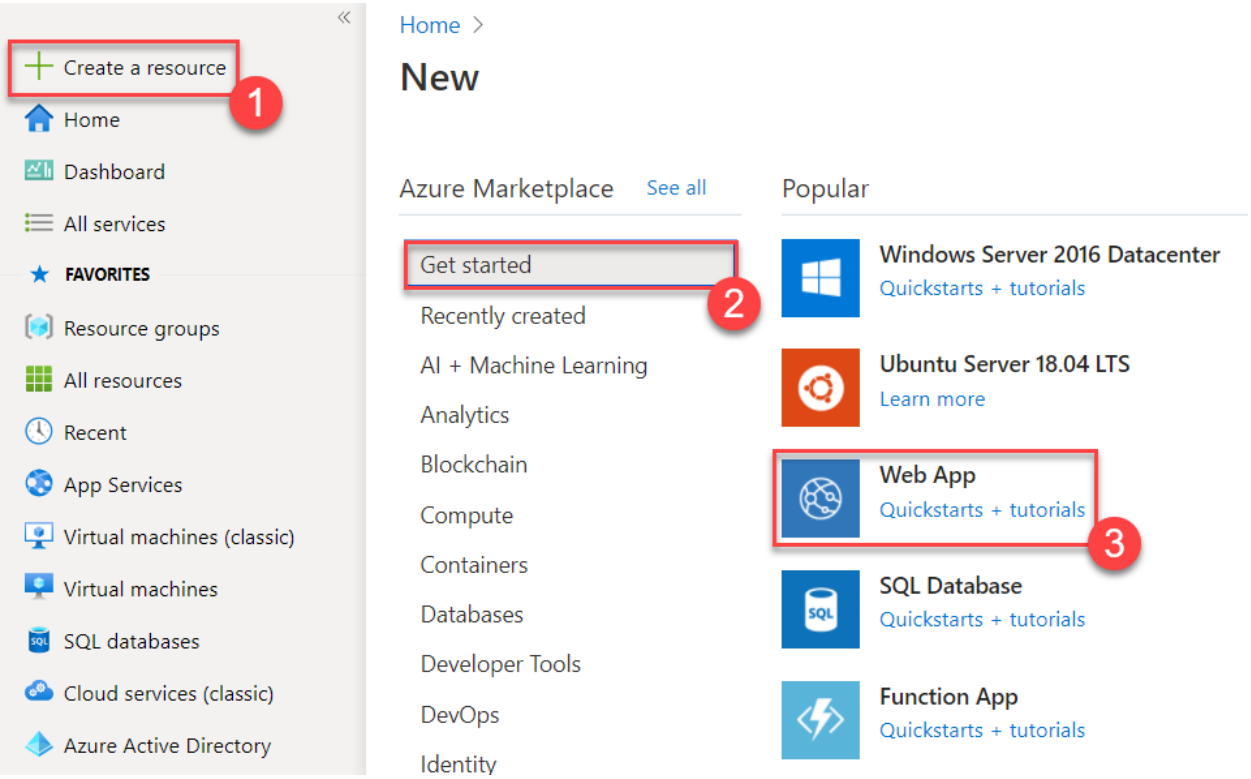
Wait for couple minutes to complete build.

If build completed click on **back arrow** button



Lab 3: Create an Azure Web App

Step 39: **+ Create a resource -> Get started -> Web App**



Step 40: Create Web App

Subscription: **Choose active Subscription**

Resource Group: **Create New Resource Group** Ex. **WebAppDevOpsLabRG**

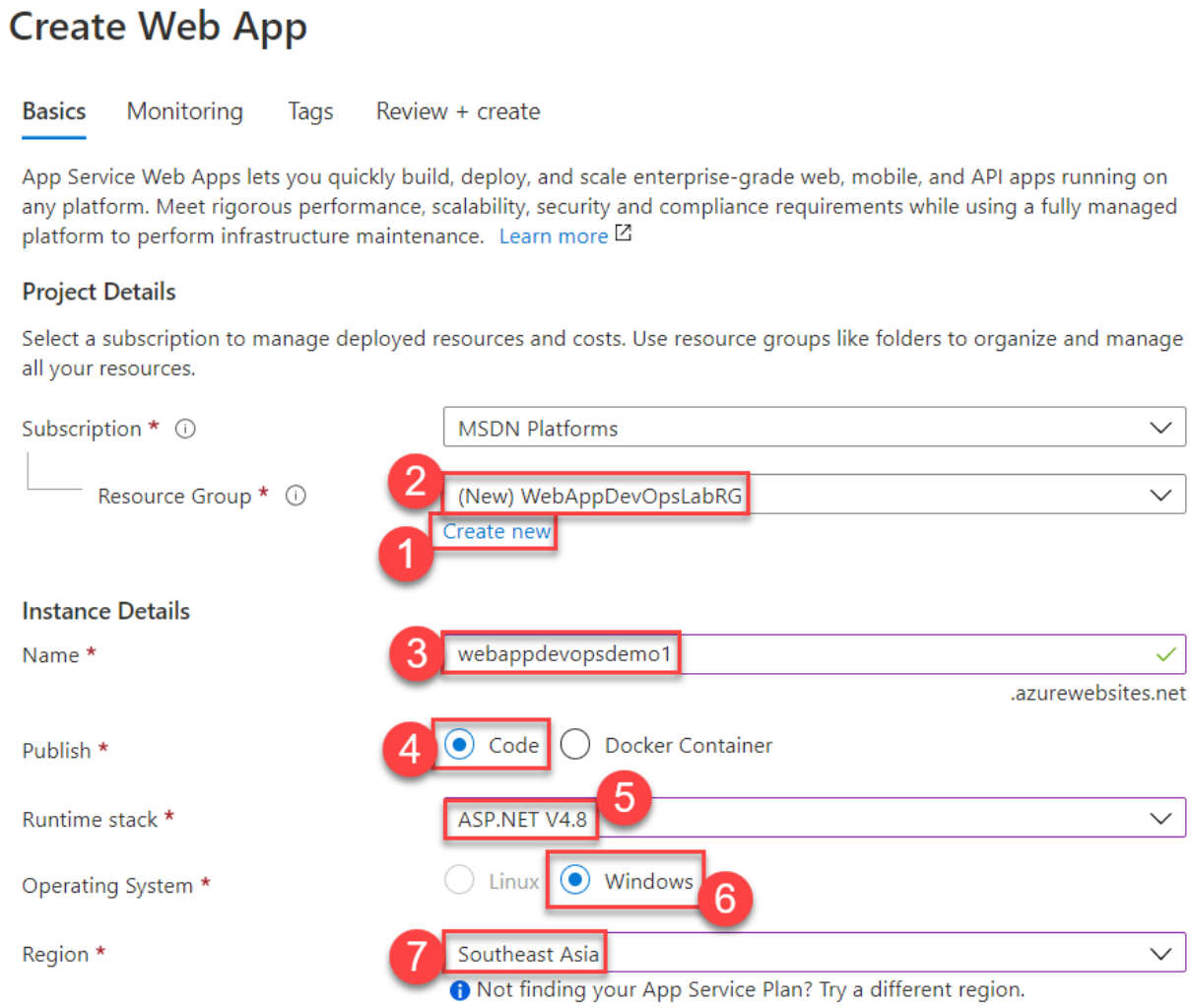
Name: **Enter unique name of web app**

Publish: **Code**

Runtime stack: **ASP.NET v4.8**

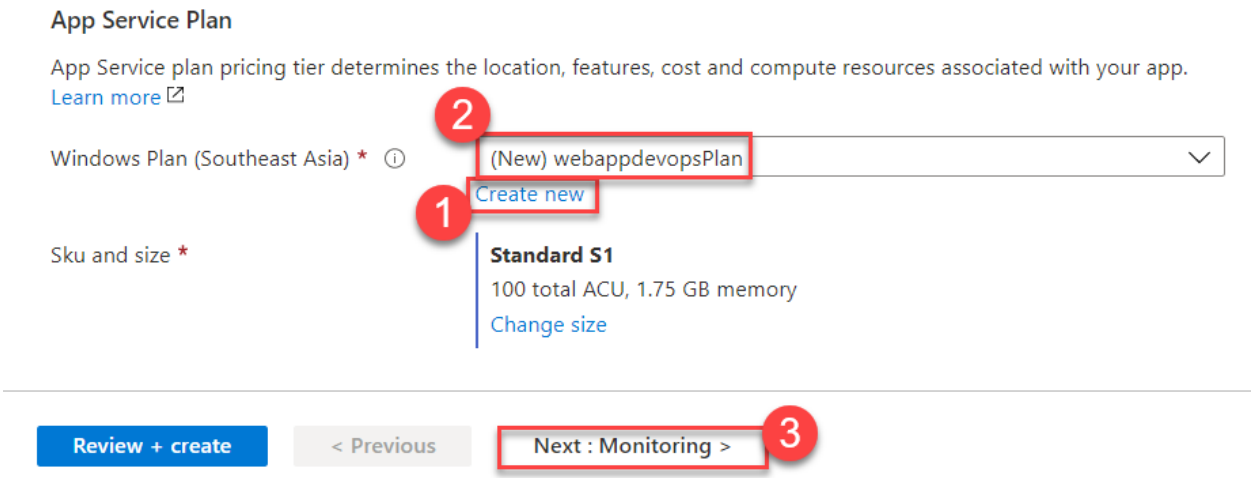
Operating System: **Windows**

Region**: Choose nearest region**



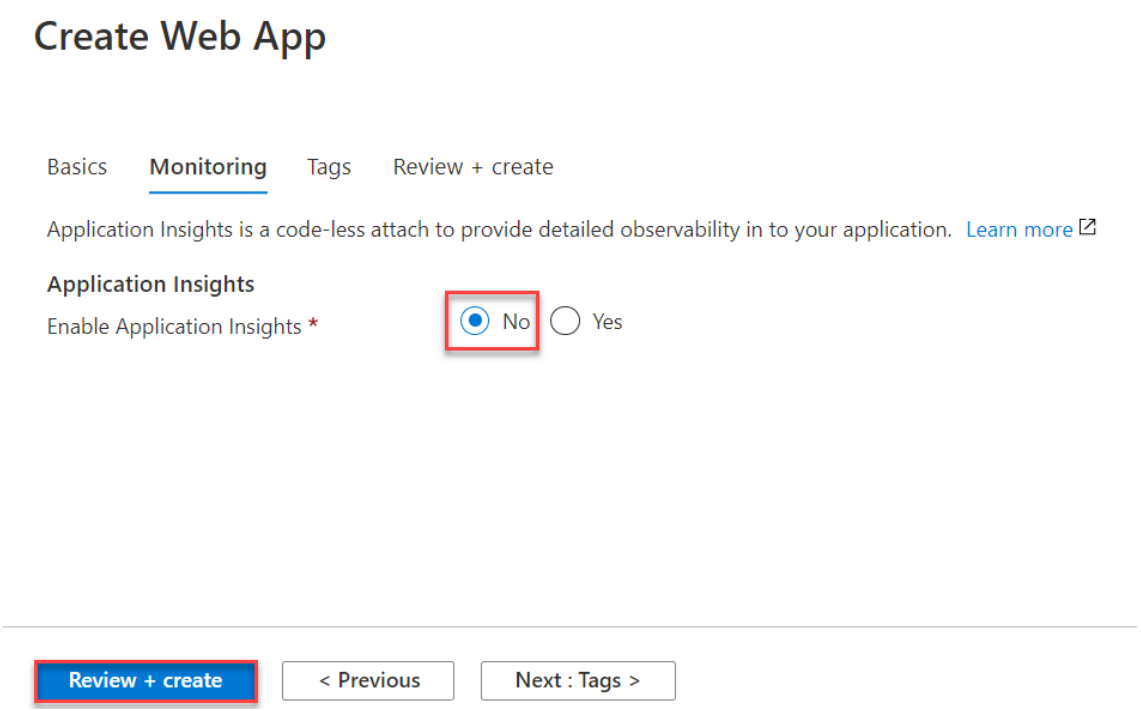
Plan: **Create New Plan** Ex. **webappdevopsPlan**

Click on **Next: Monitoring >** button

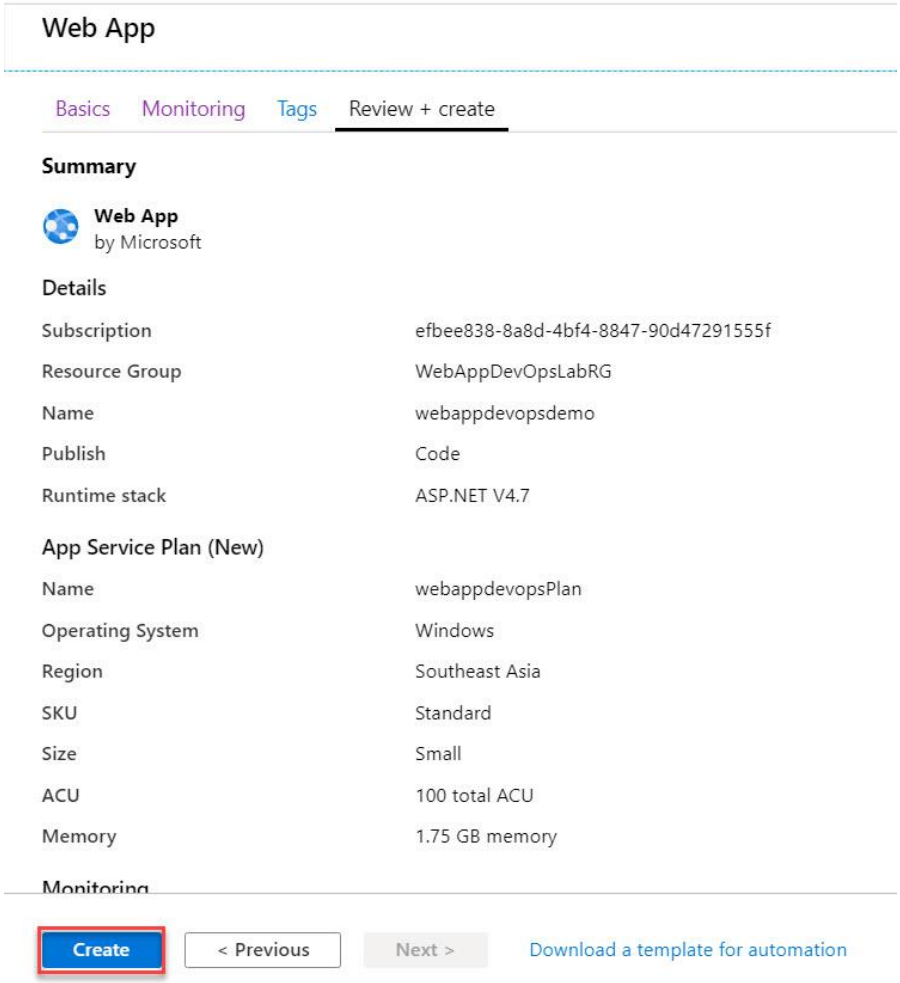


Step 41: Application insights: **No**

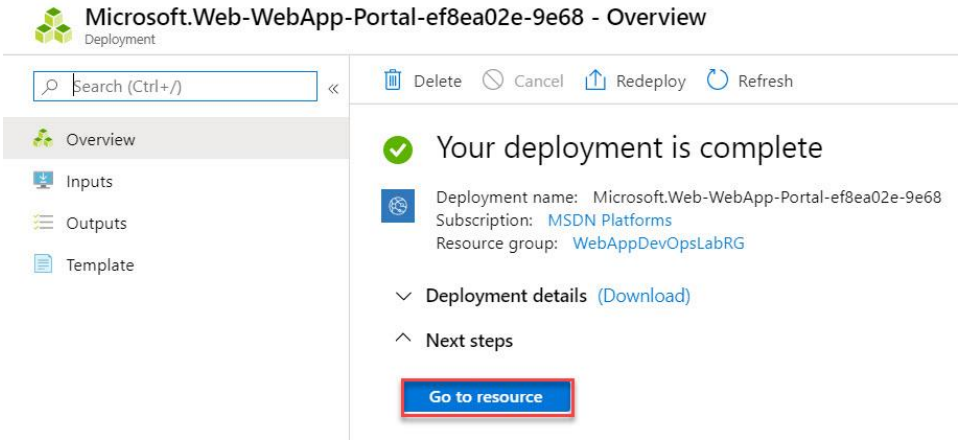
Click on **Review + Create** button



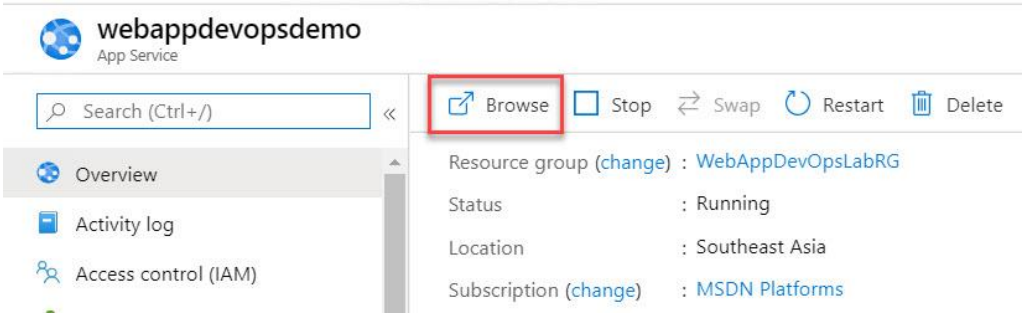
Step 42: Click on **Create** button



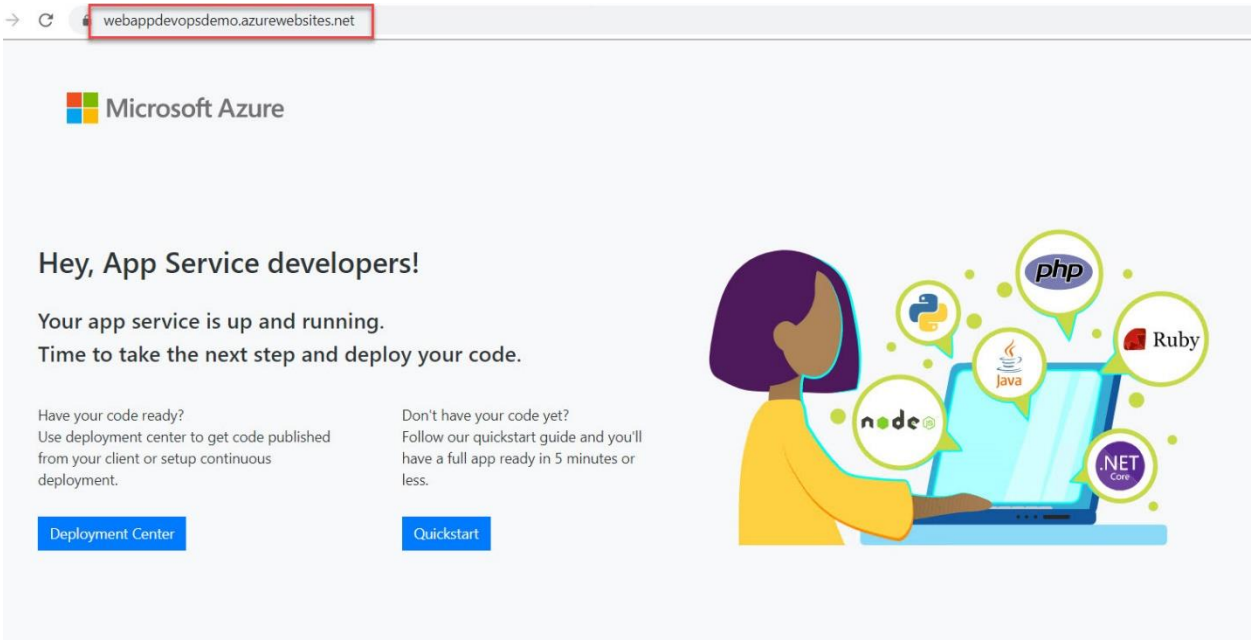
Click on **Go to resource** button



Step 43: Click on **Browse** button



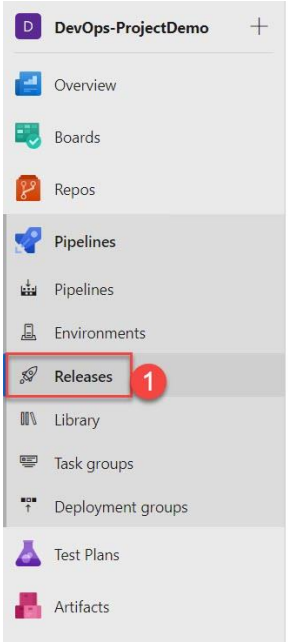
Sample Web App Page will open

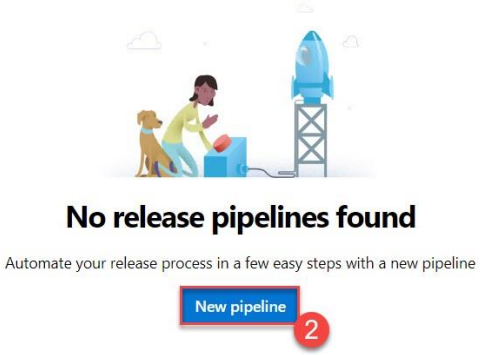


Lab 4: Continuous Deployment

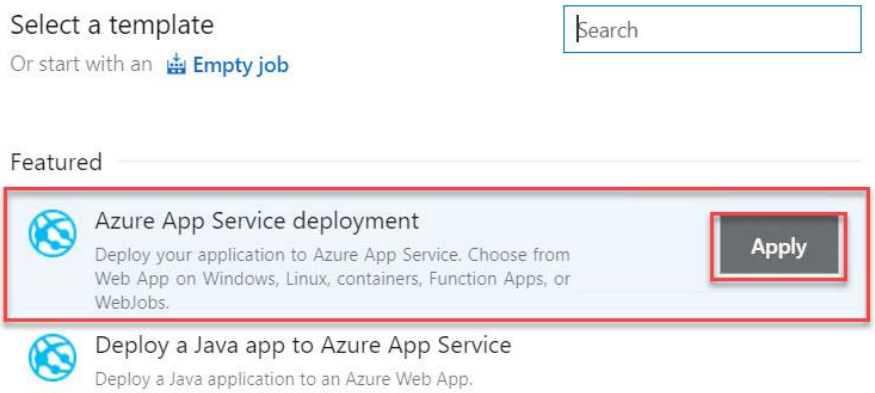
Step 44: Navigate to **Azure DevOps Portal**

Click on **Releases** and Create **New Pipeline**



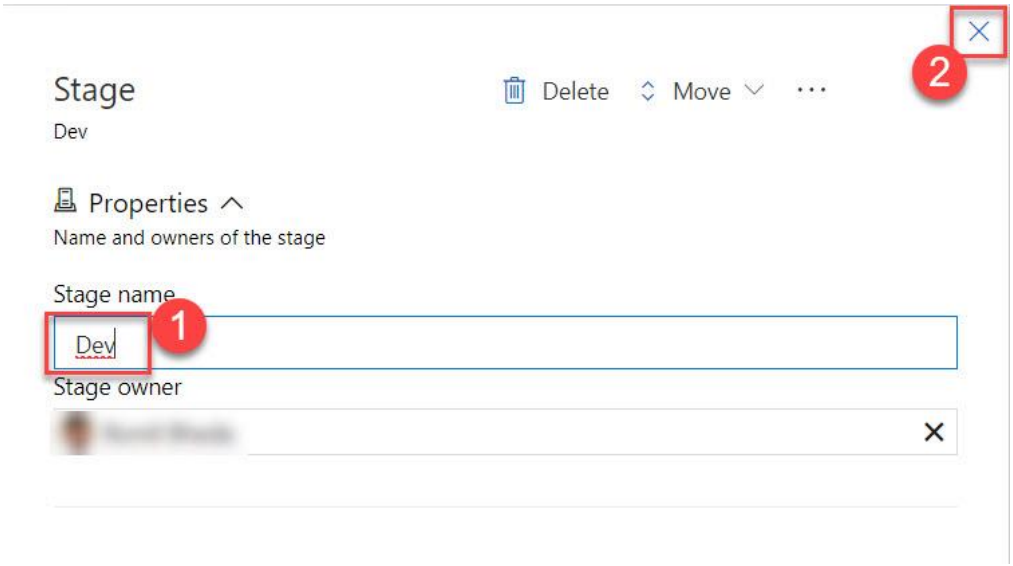


Step 45: Select **Azure App Service deployment** option

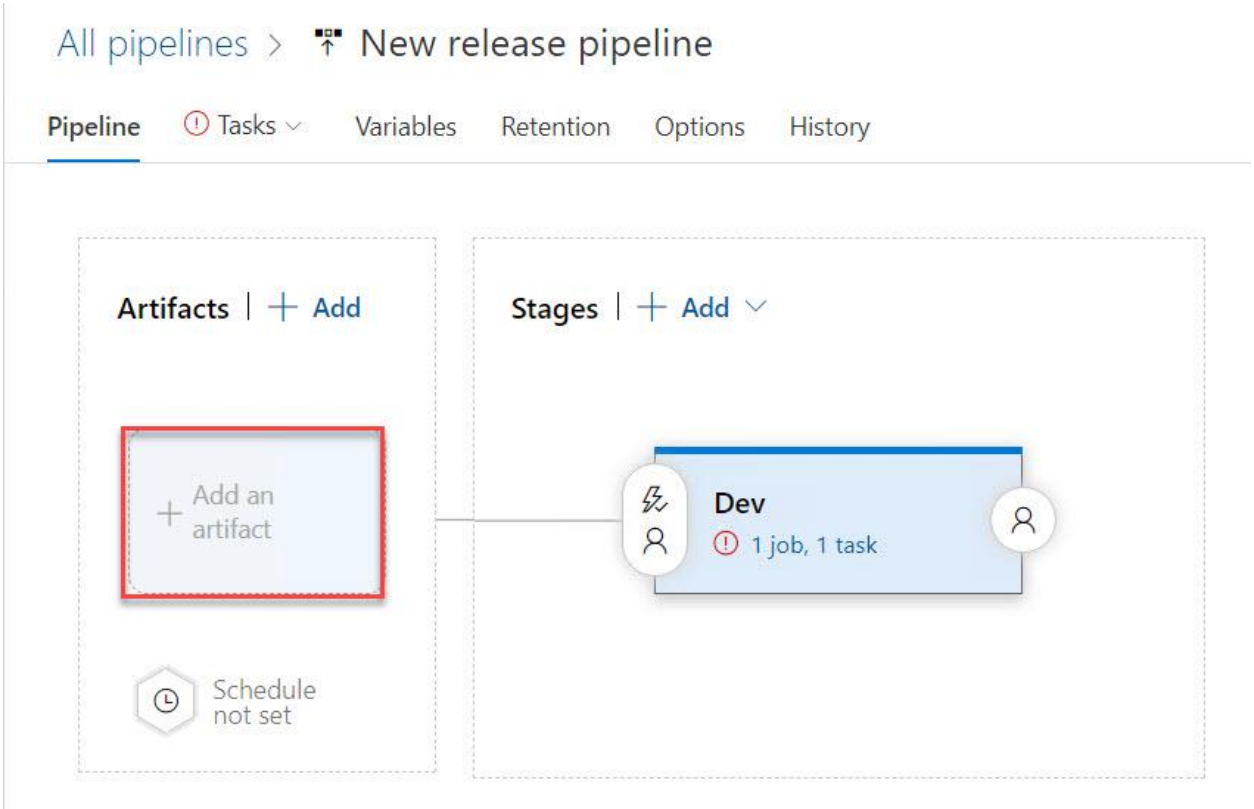


Stage Name: **Dev**

Click on **Close** button



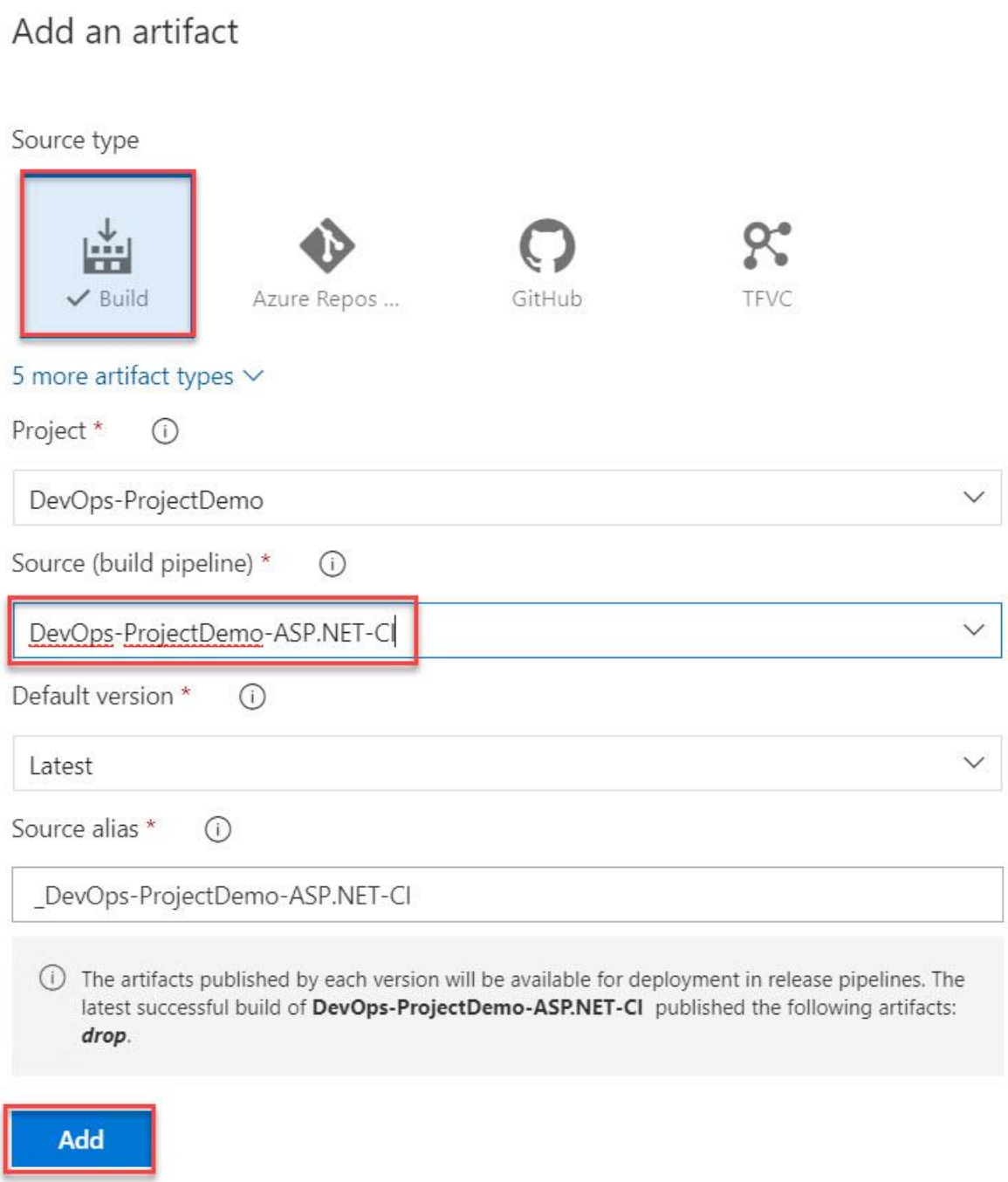
Step 46: Click on **+ Add an artifact** option



Step 47: Select Source type: **Build**

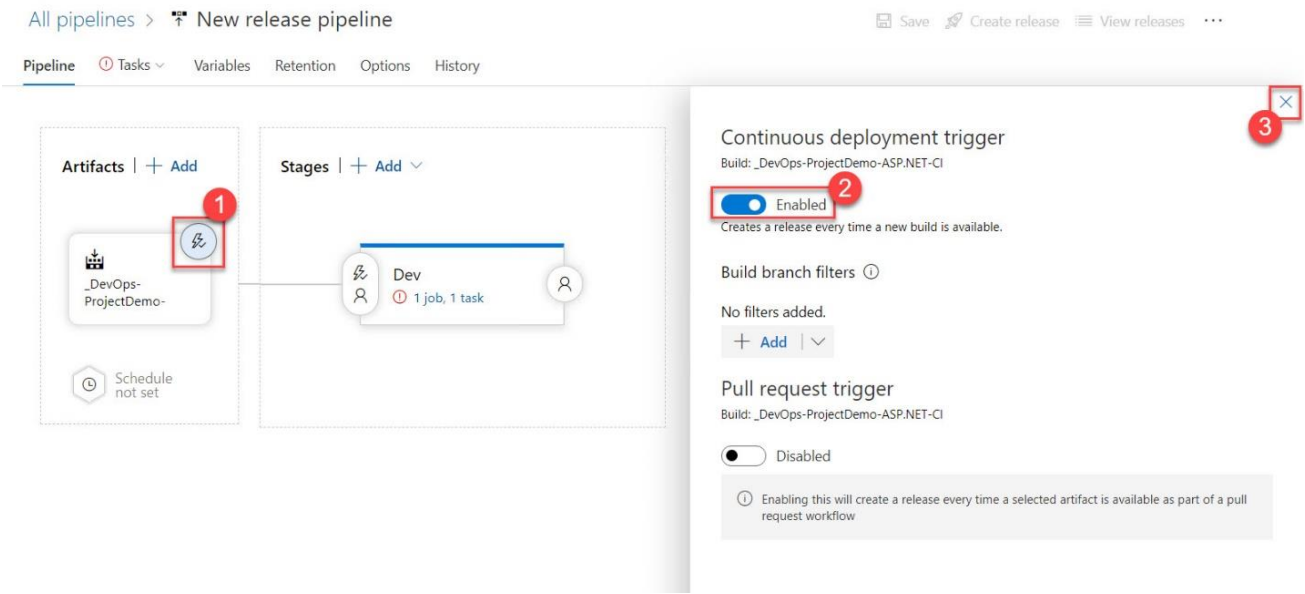
Source (build pipeline): **Choose from drop-down option** Ex. **ProjectName-ASP.NET-CI**

Click on **Add** button

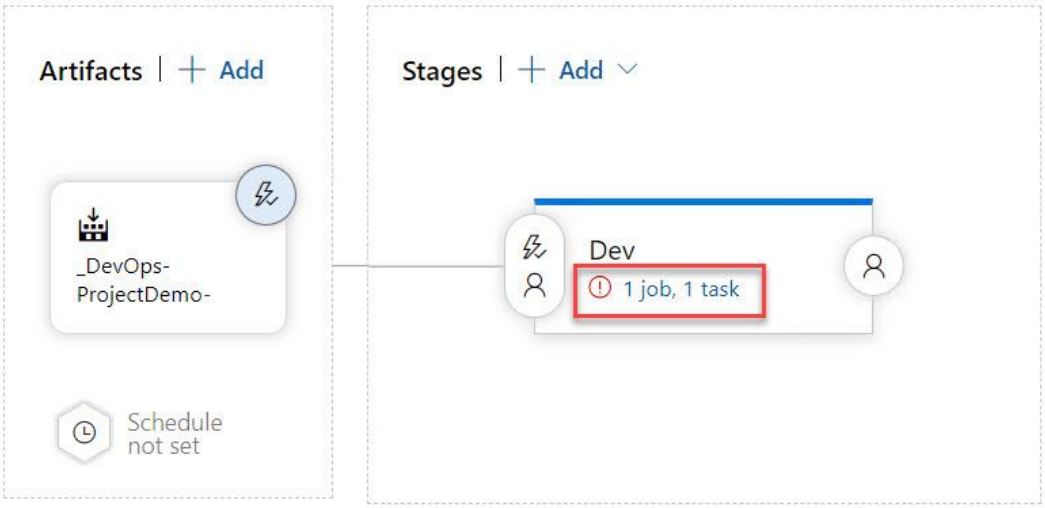


Step 48: Enable Deployment Trigger option

Click on **Thunder** icon and **Enable** option

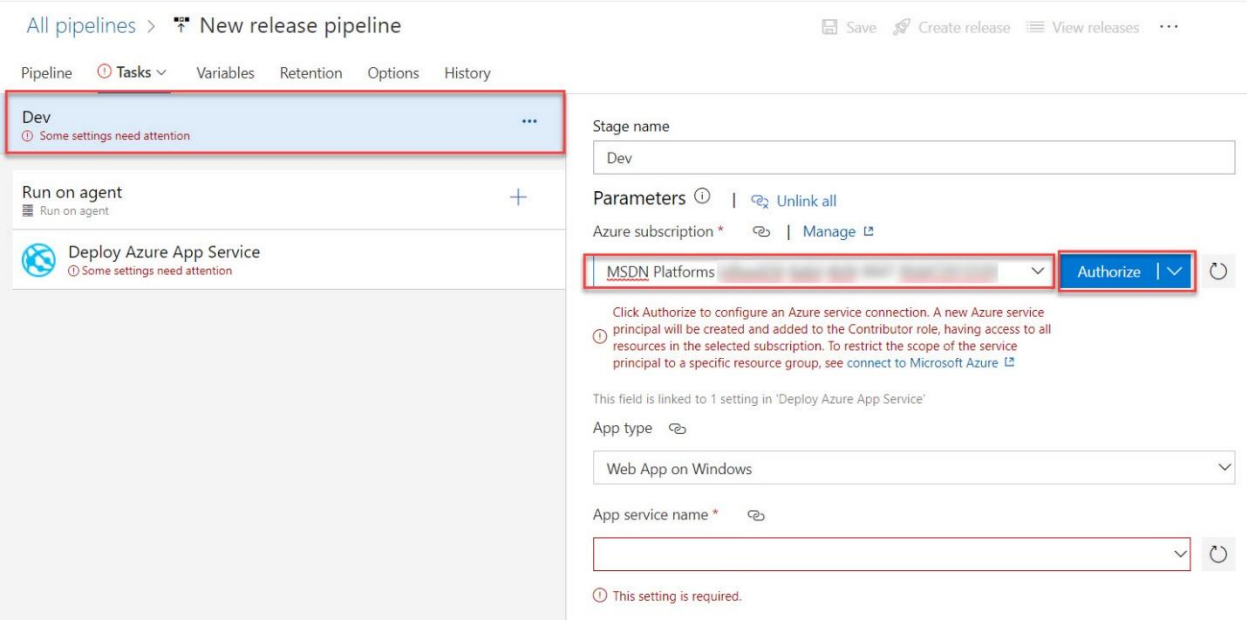


Step 49: Click on **1 job, 1 task** under **Dev** stage

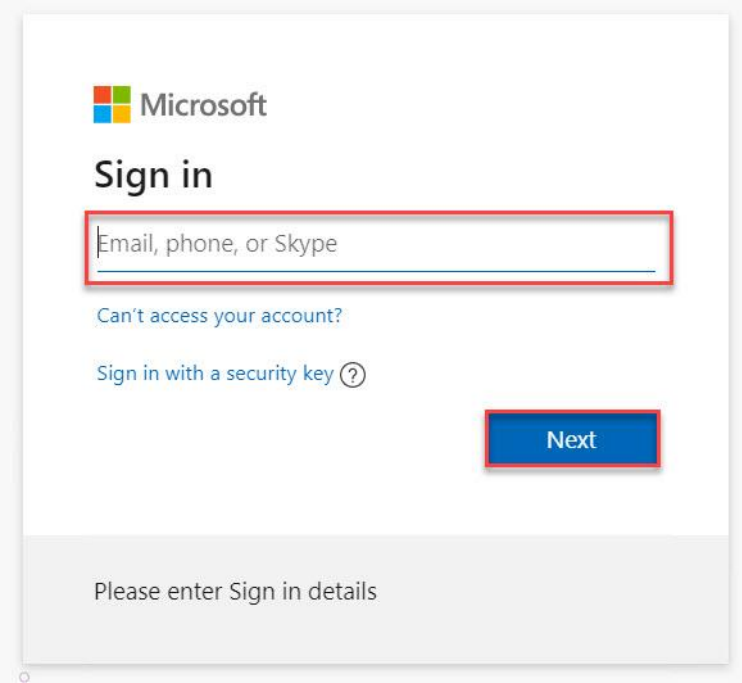


Step 50: Select **Dev**

Azure Subscription: **Choose Subscription** and click on **Authorize** button

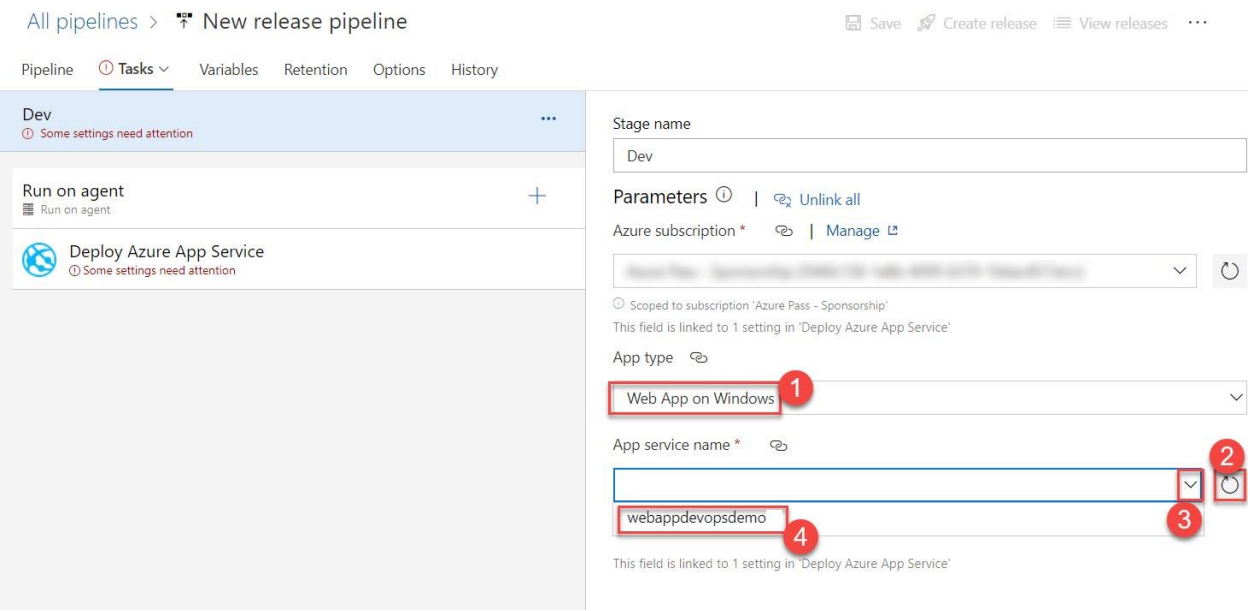


Enter Microsoft Azure Credentials



Step 51: App type: Select **Web App on Windows**

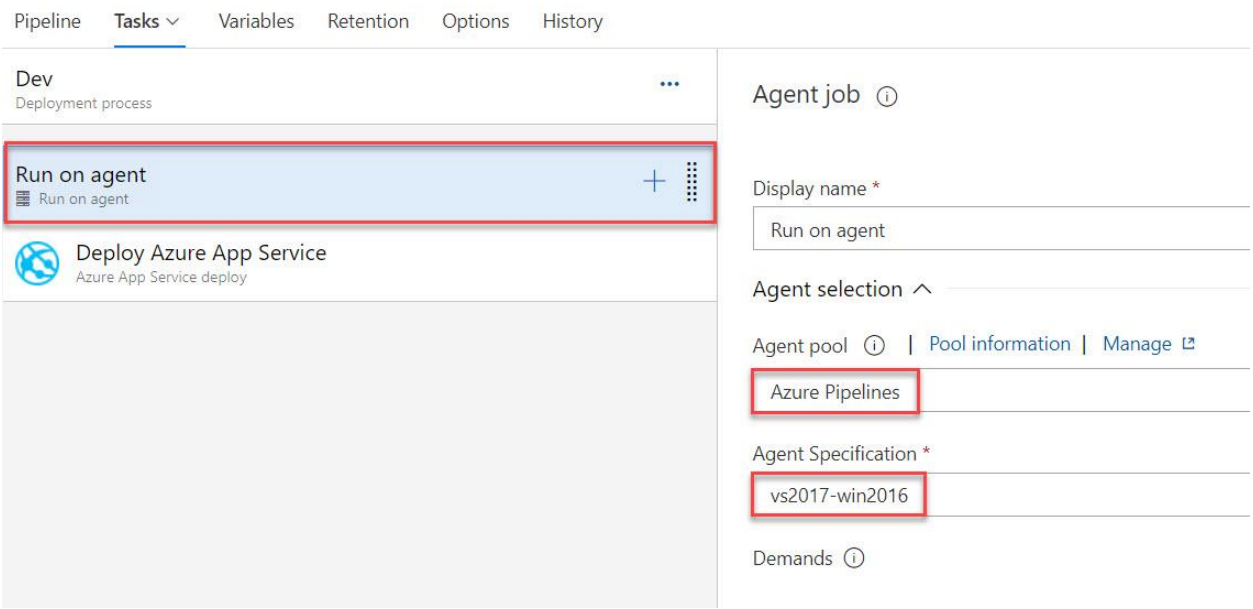
App Service name: Choose **Web App** (created from Lab 3)



Step 52: Select **Run on agent**

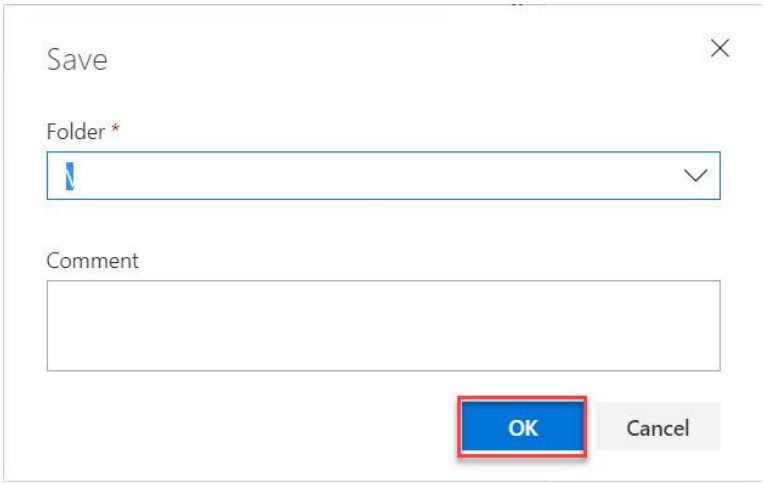
Agent pool: **Azure Pipelines**

Agent Specification: **vs2017-win2016**



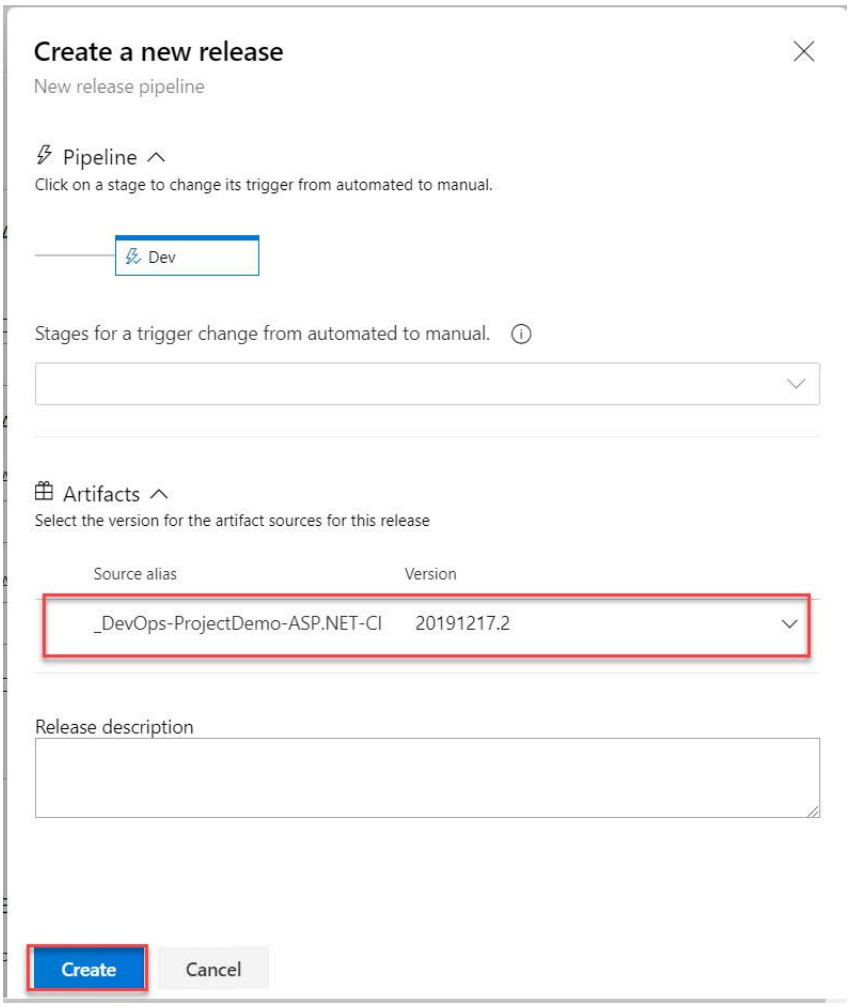
Step 53: Click on **Save** button

Click on **OK** button

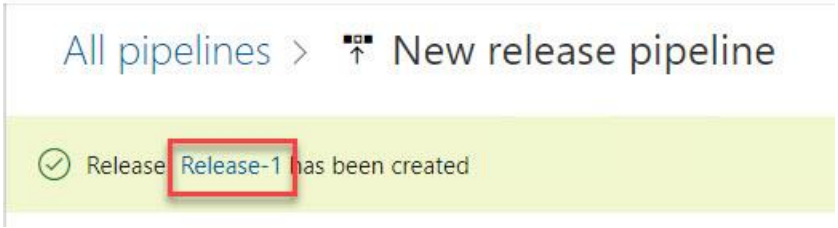


Step 54: Click on **Create release** button

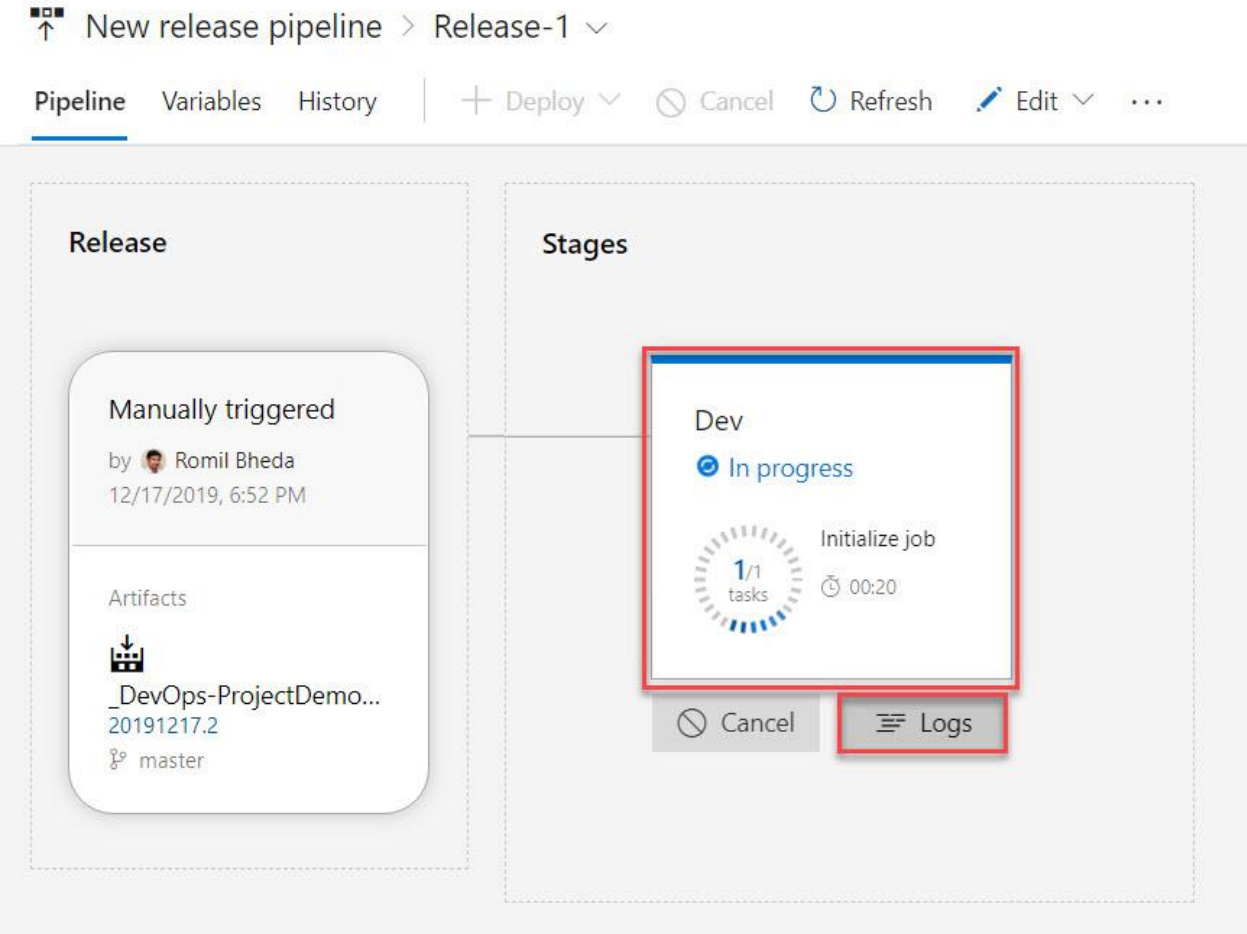
Verify Artifacts and Click on **Create** button



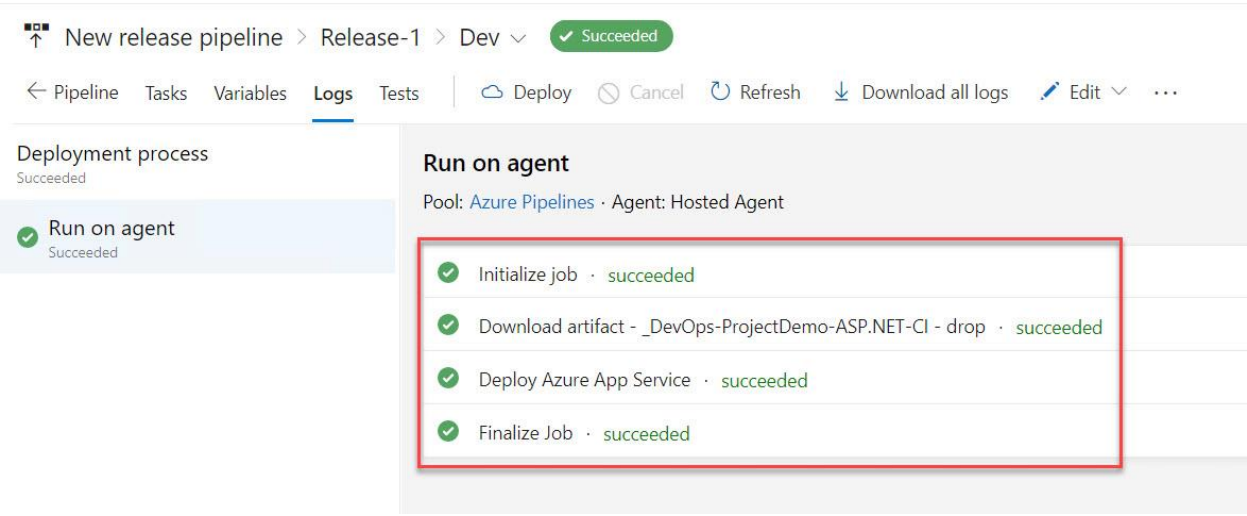
Click on **Release-1**



Release pipeline started. To check status, click on **Logs** button

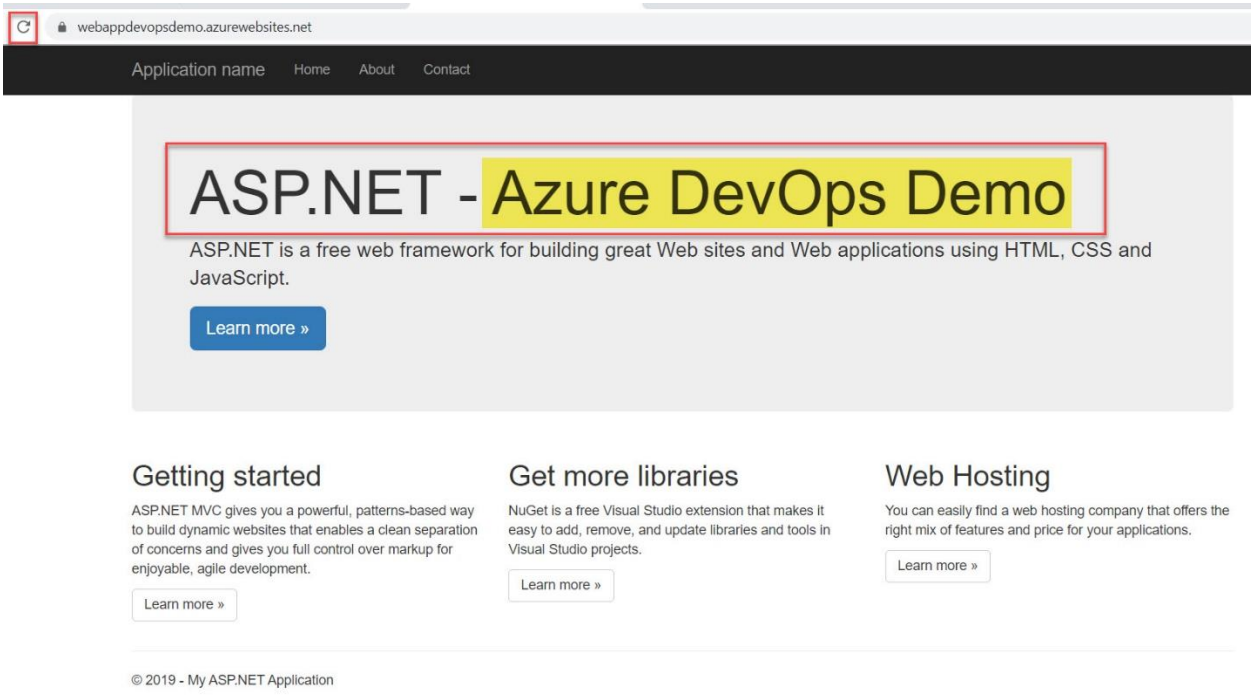


Wait for couple of duration to complete job



Step 55: Navigate to Web App and click on **Refresh** button.

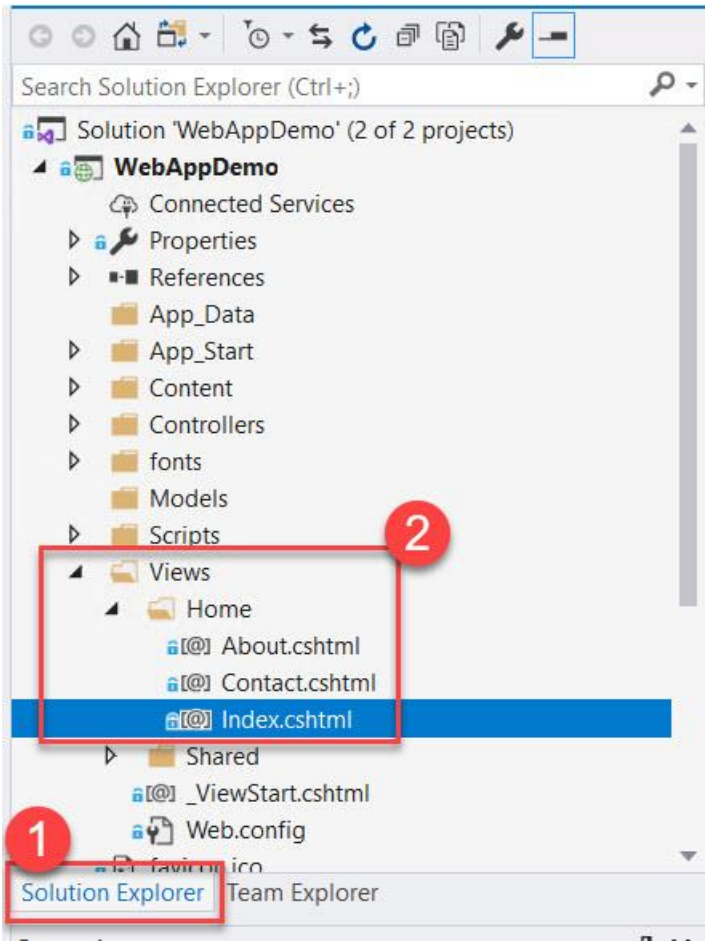
You can see Web App Content updated.



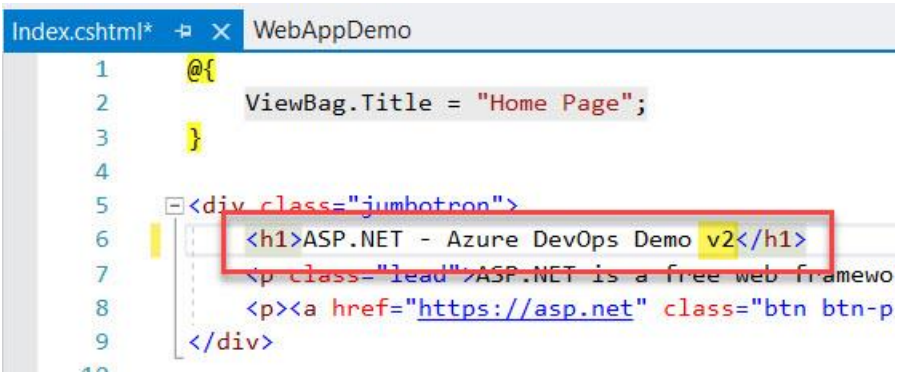
Step 56: **Now to check complete flow of CI/CD Pipeline**

Navigate to **Visual Studio Sample project**

Edit **Index.cshtml** page

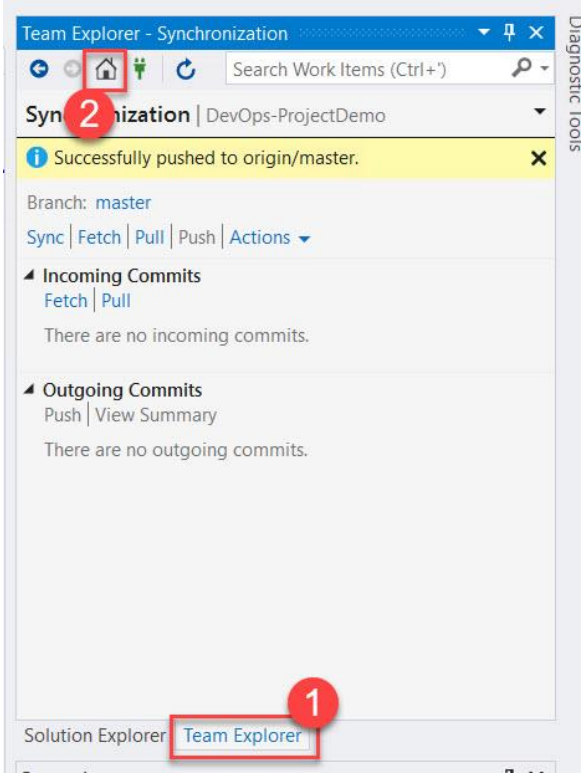


Step 57: Update text as below:

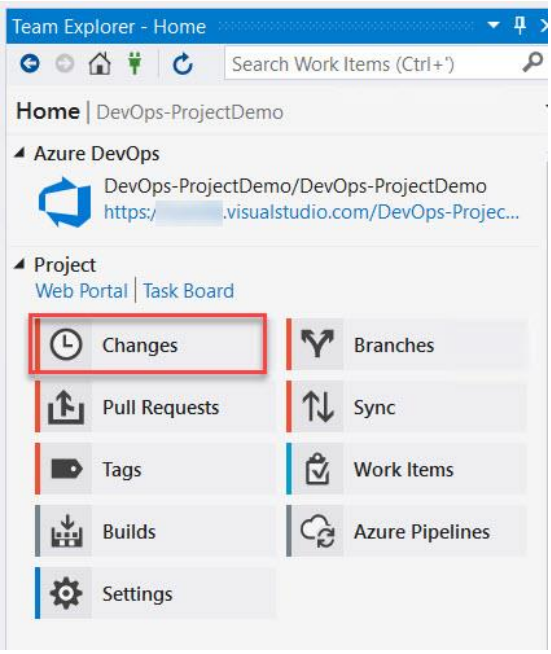


Step 58: Select **Team Explorer** tab

Click on **Home** option

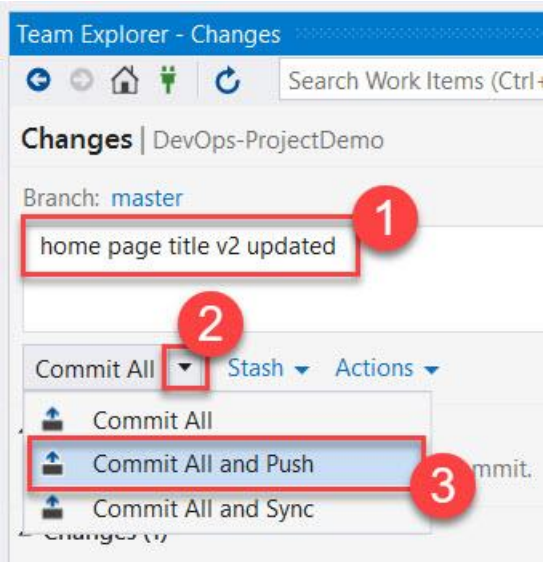


Step 59: Click on **Changes** option

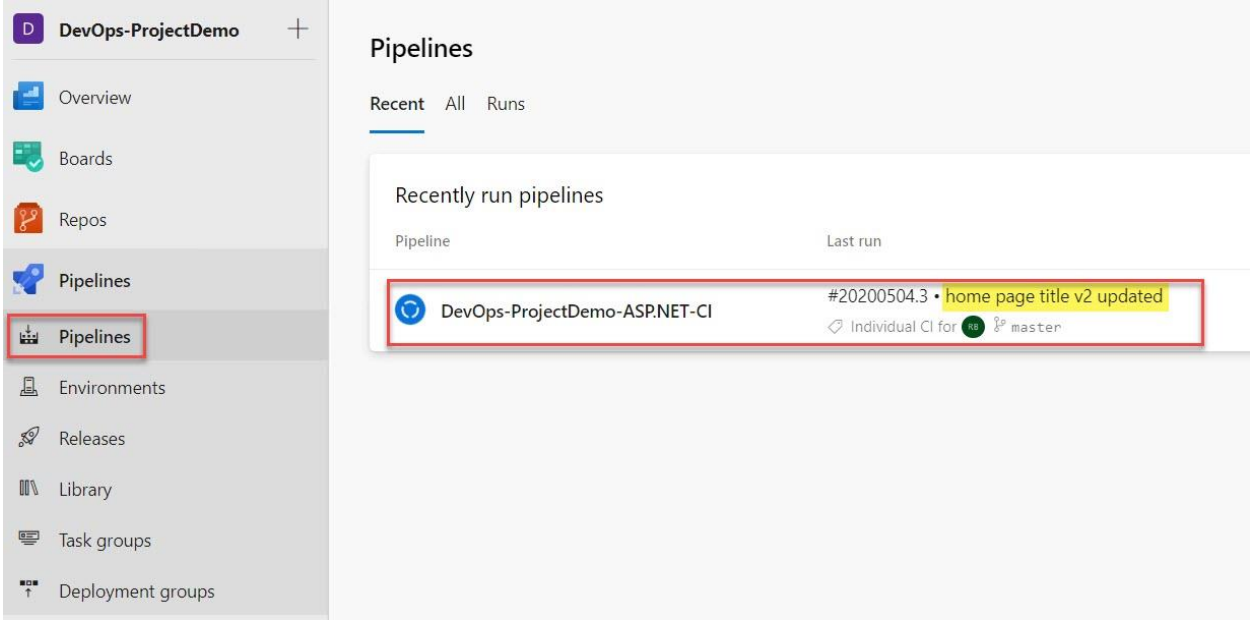


Step 60: Message: **home page title v2 updated**

Choose **Commit All and Push** option



Step 61: Automatic **Build Pipeline** started



If Build Pipeline completed, Release Pipeline will start.

Step 62: Once again **refresh** web app and you can see changes