**CI & CD of ASP.Net Core App using Github, Azure DevOps & Azure**

**Pre-requisites**

* Microsoft Azure Subscription
* Microsoft Azure DevOps Account
* Github Account
* Git - <https://git-scm.com/downloads>
* .Net Core SDK version 2.1.300 or later (if not installed) - <https://dotnet.microsoft.com/download>

Check using this command: **dotnet --version**

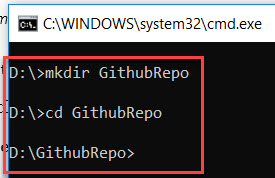
Sample ASP.NET Core Solution (Feed Reader) - <https://github.com/Azure-Samples/simple-feed-reader/>

Step 1: Open Command Prompt and Create one **GithubRepo** Folder

mkdir GithubRepo

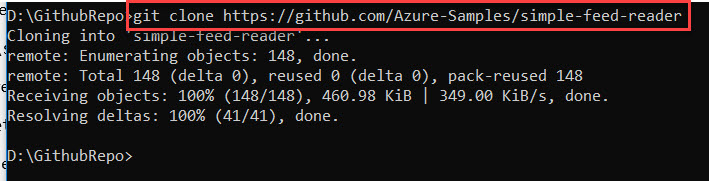
Change Directory

cd GithubRepo



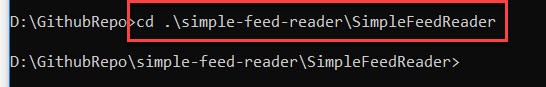
Step 2: Clone the solution from link to local machine

git clone <https://github.com/Azure-Samples/simple-feed-reader>



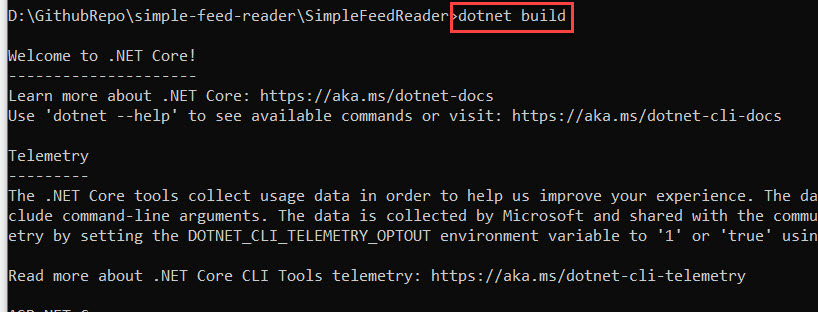
Step 3: Navigate to project folder

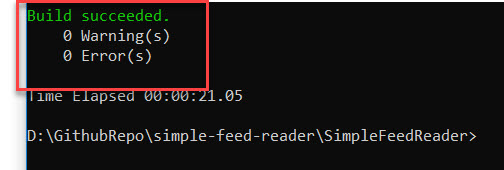
cd simple-feed-reader\SimpleFeedReader



Step 4: Restore the packages

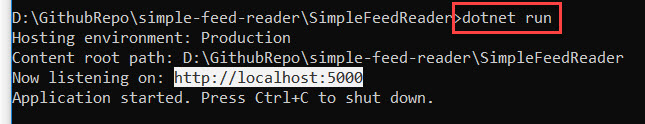
dotnet build



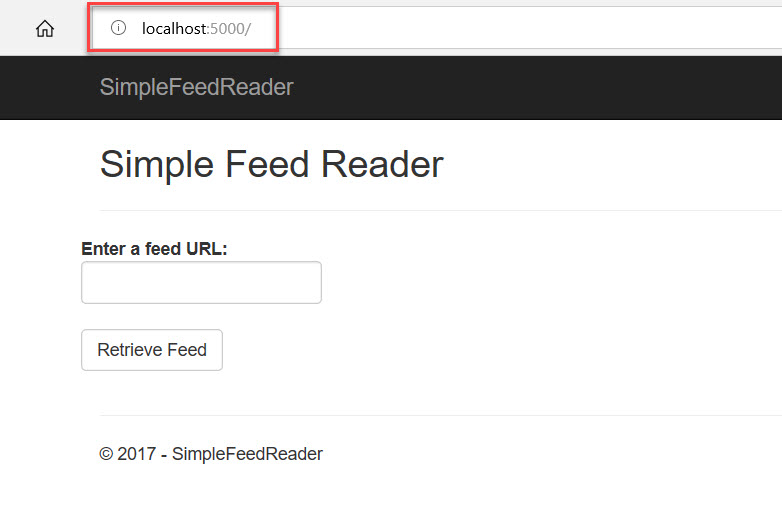


Step 5: Run the project

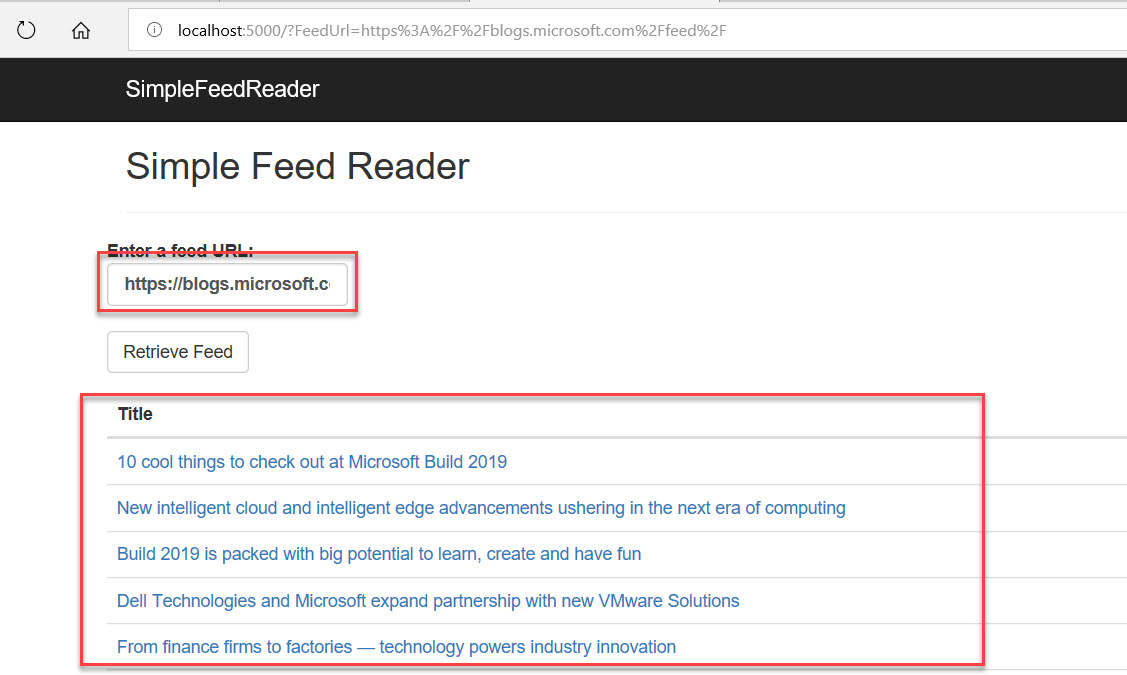
dotnet run



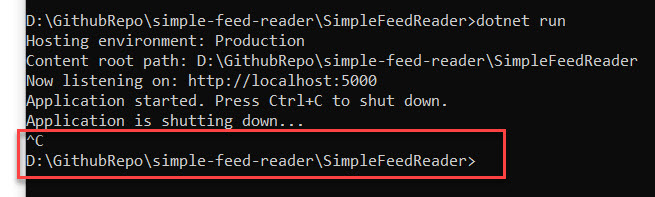
Step 6: Open browser and type **http://localhost:5000**



Step 7: Enter any feed url: <https://blogs.microsoft.com/feed/>



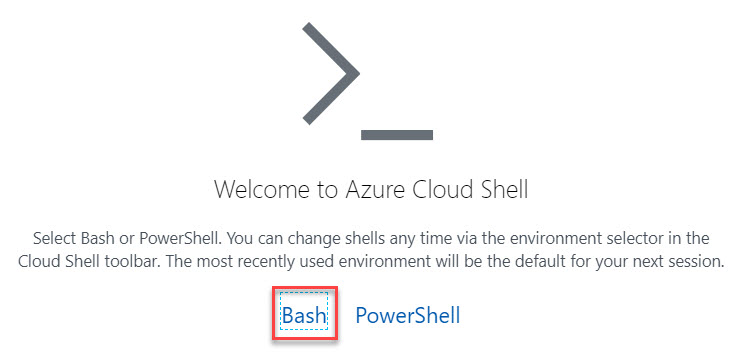
Click on **Ctrl + C** to stop running of application



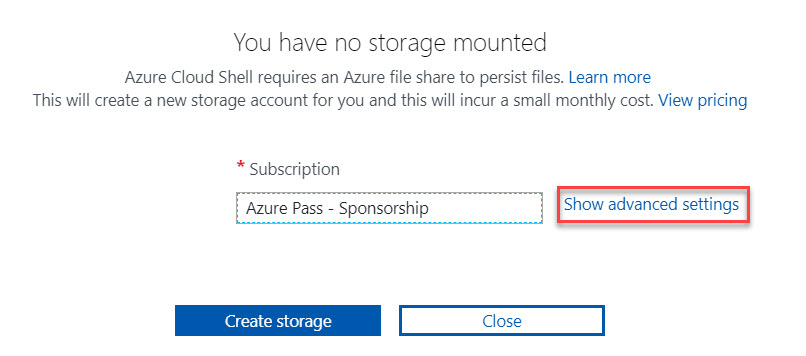
Step 8: Navigate to **Azure Portal** and click on **Cloud Shell**



Step 9: Select **Bash** option



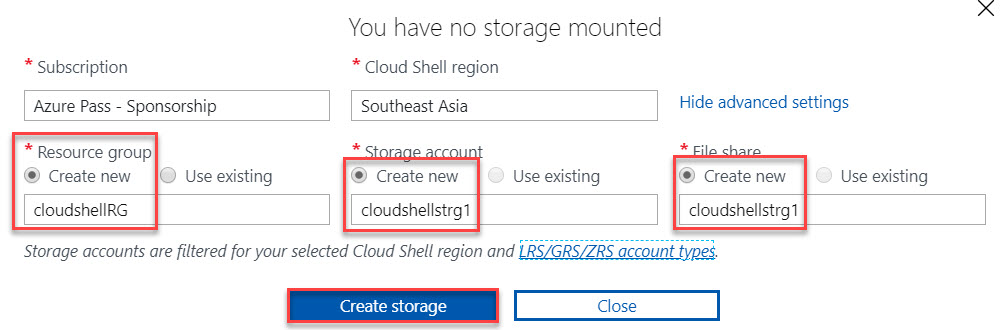
Step 10: Click on **Show advanced settings** option



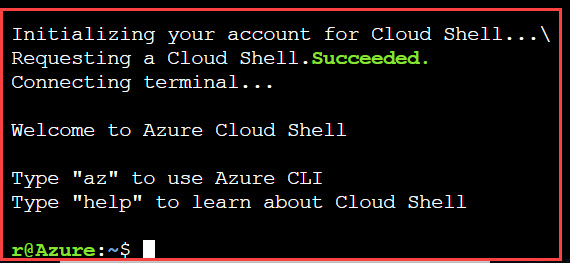
Step 11: Create Storage account

Select Region, Create New Resource Group, Storage account, file share

Click on **Create Storage** button.



Cloud Shell will load as below:



Step 12: declare variable to store web app name & resource group name.

webappname=webapp$RANDOM

rg=AzureDevOpsWebAppRG

Step 13: Create Resource Group

az group create --location southeastasia --name $rg

Step 14: Create app service plan S1 tier

az appservice plan create --name $webappname --resource-group $rg --sku S1

Step 15: Create Web App

az webapp create --name $webappname --resource-group $rg --plan $webappname

Step 16: Set deployment credentials

az webapp deployment user set --user-name devopsdemouser --password Demo@pass123

Step 17: Configure the web app to accept deployments from local Git and display the Git deployment URL. Note: Also copy this URL

echo gitdeployuri: $(az webapp deployment source config-local-git --name $webappname --resource-group $rg --query url --output tsv)

Step 18: Display Web App URL

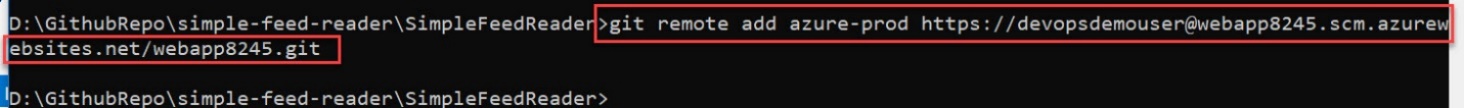
echo Web app URL: <http://$webappname.azurewebsites.net>

Step 19: Navigate to **local system command prompt**

Add Remote URL

git remote add azure-prod GIT\_DEPLOYMENT\_URL

Note: GIT\_DEPLOYMENT\_URL collect from Step 17

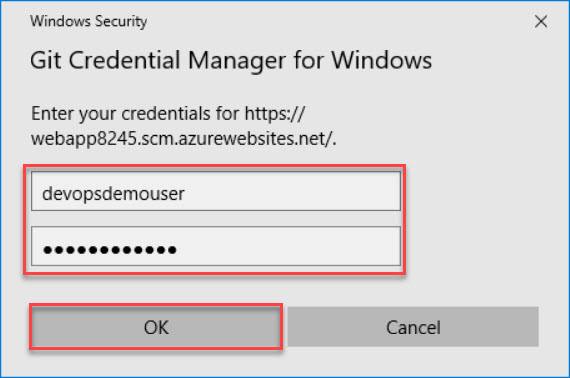


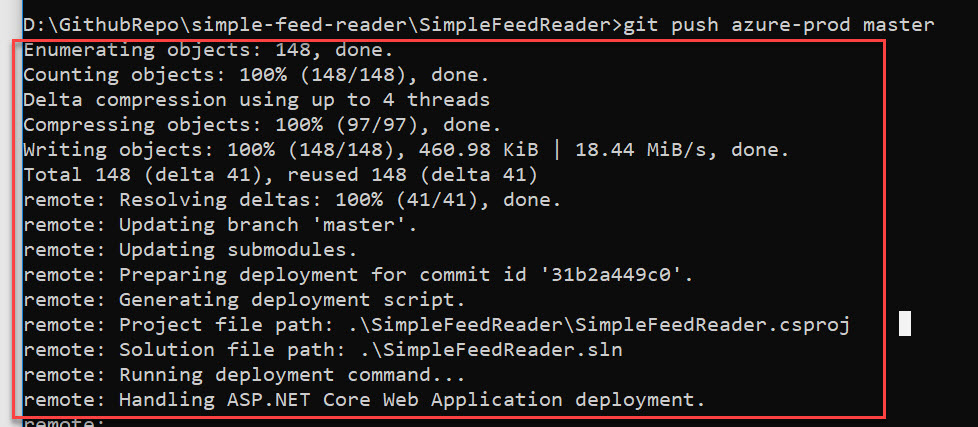
Step 20: Push local master branch to remote branch.

git push azure-prod master

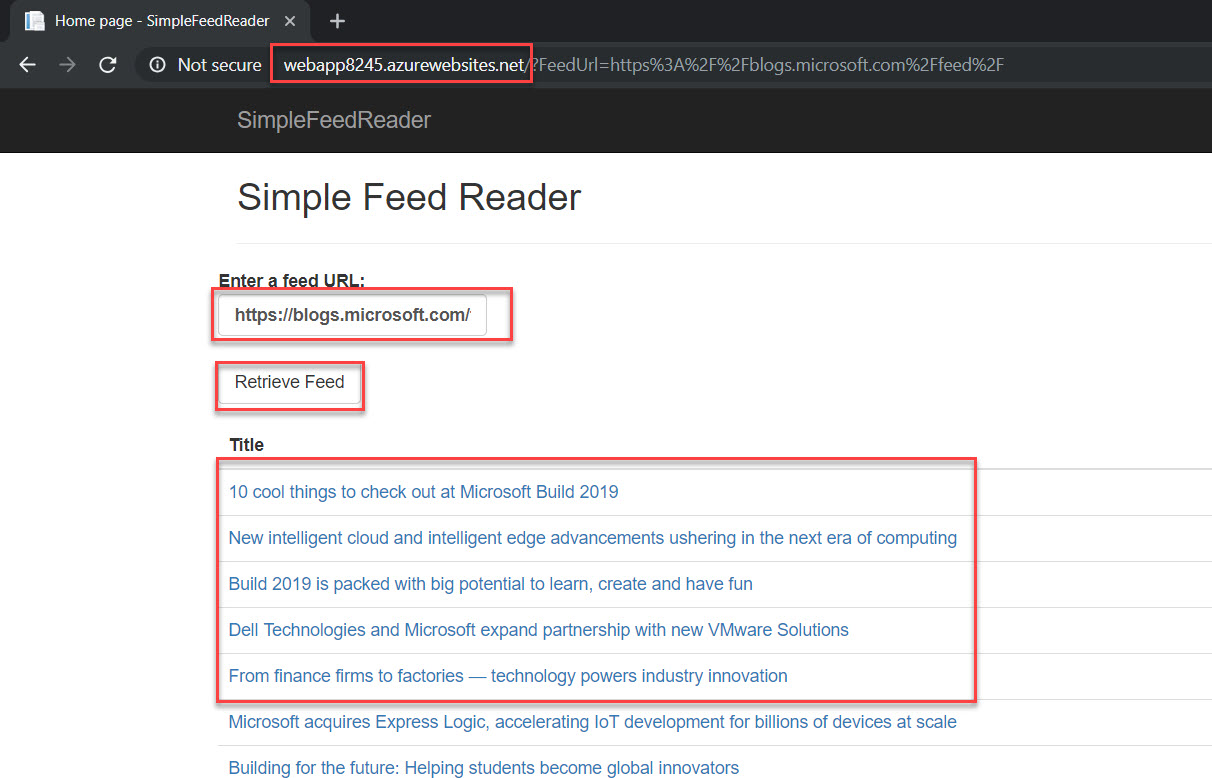


Enter Credentials. Entered in Step 16.

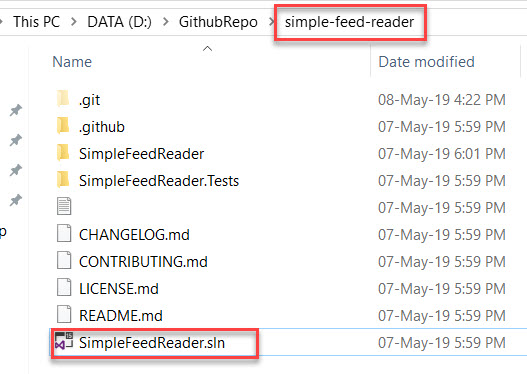




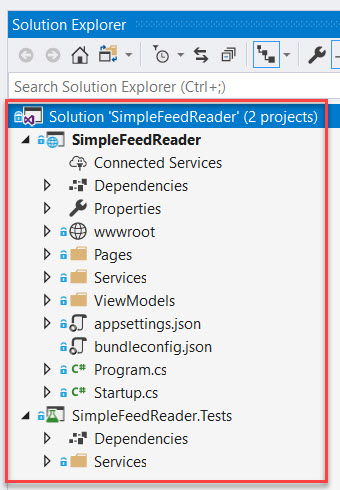
Step 21: Now run web app address. Enter Feed Reader: <https://blogs.microsoft.com/feed/>



Step 22: Navigate to **simple-feed-reader** and double click on **SimpleFeedReader.sln**



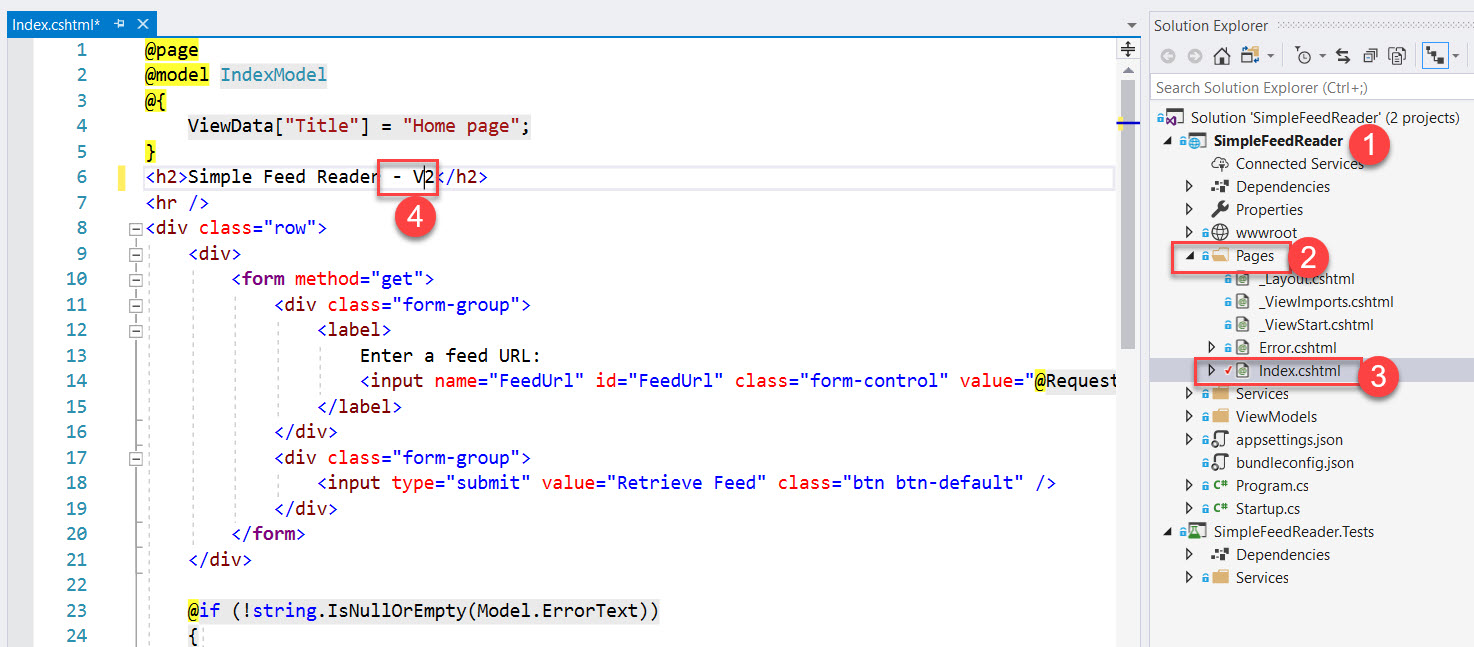
Wait for few seconds to restore packages.



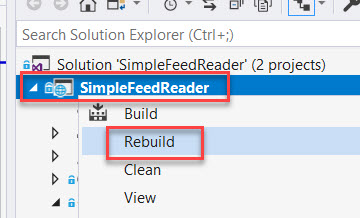
Step 23: Open **index.cshtml**

**SimpleFeedReader -> Pages -> open Index.cshtml**

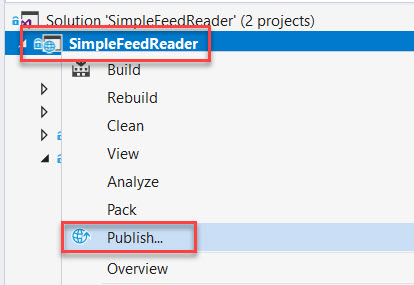
Add – **V2** on line number 6.



Step 24: Right Click on Project and Click on **Rebuild** options



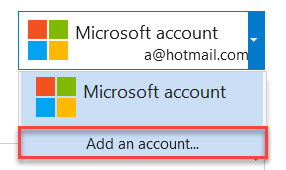
Step 25: Right Click on Project once again and select **Publish…** option



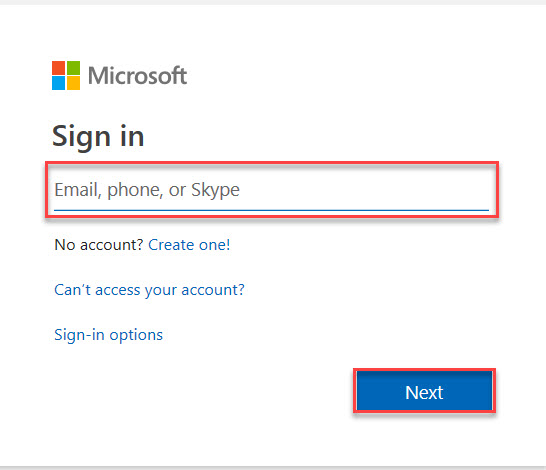
Step 26: Select Existing option and click on Publish button.



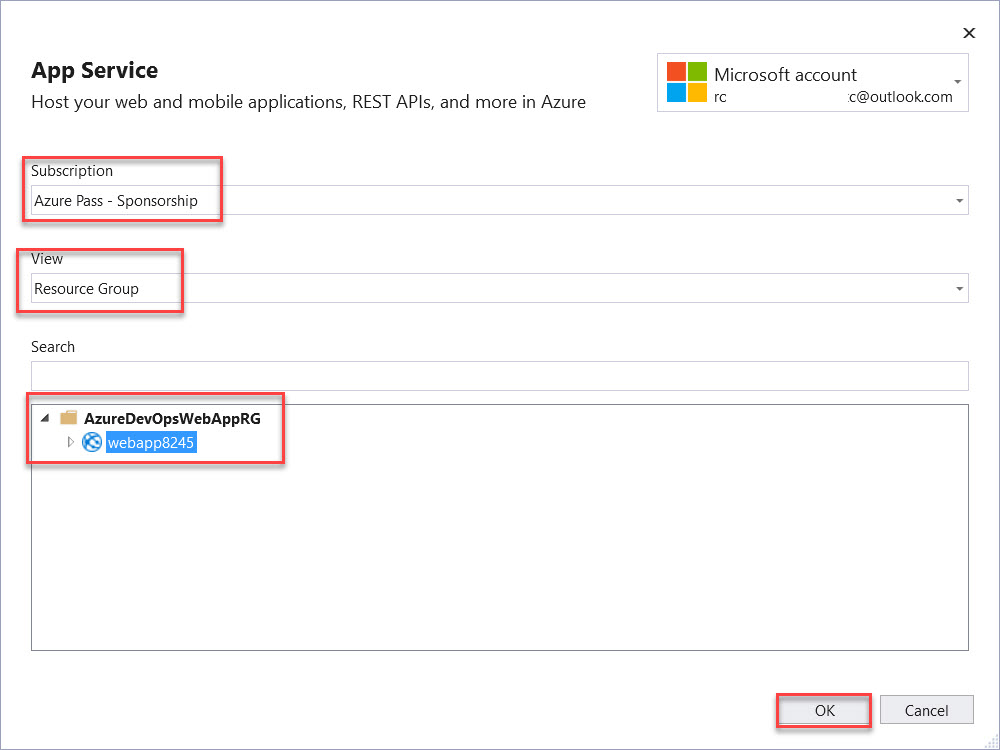
Step 27: If Microsoft Azure account not configured please click on **Add an account…**



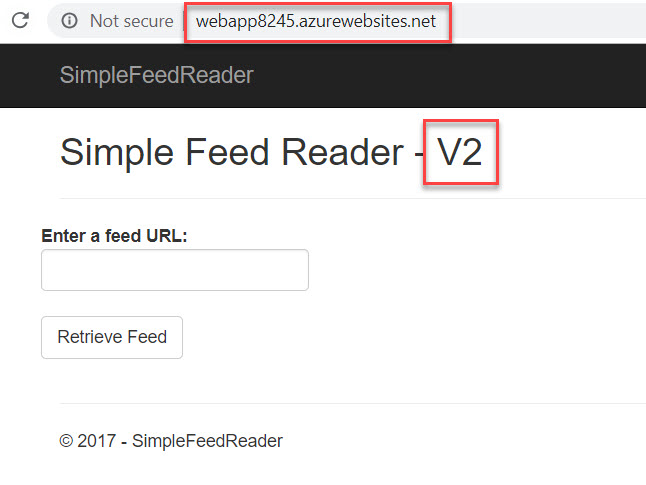
Enter Azure Credentials



Step 28: Select **Subscription -> Resource Group -> Select Web App** from the list and click on **OK** button.



Step 29: Within couple of second it will update with new version



Step 30: Navigate to Azure and Create Deployment Slot with name staging using **Cloud Shell**.

az webapp deployment slot create --name $webappname --resource-group $rg --slot staging

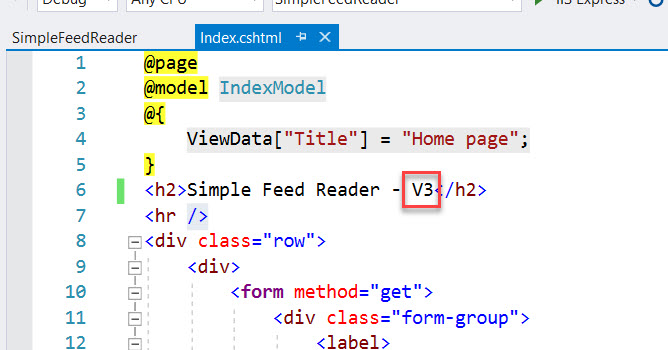
Step 31: Configure the Staging slot to use deployment from local Git and get the staging deployment URL. Note: Copy URL for further steps

echo GitURLstaging: $(az webapp deployment source config-local-git --name $webappname --resource-group $rg --slot staging --query url --output tsv)

Step 32: Get Staging Slot’s URL. Note: Copy URL for further steps.

echo Staging web app URL: <http://$webappname-staging.azurewebsites.net>

Step 33: Now once again modify with version from **V2 to V3**

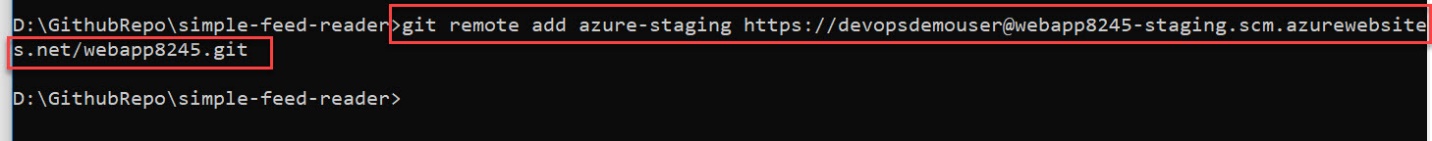


Step 34: Commit the file to local Git

git commit -a -m "upgraded to V3"

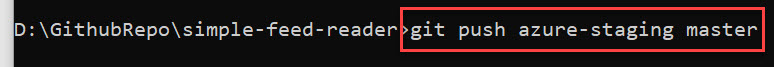
Step 35: Add Remote URL

git remote add azure-staging <Git\_staging\_deployment\_URL>

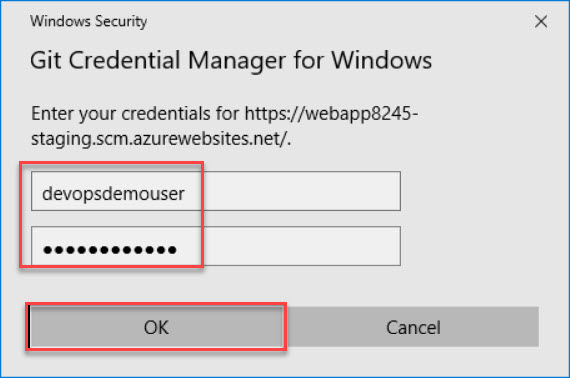


Step 36: Push the local master branch to azure-staging remote master branch

git push azure-staging master

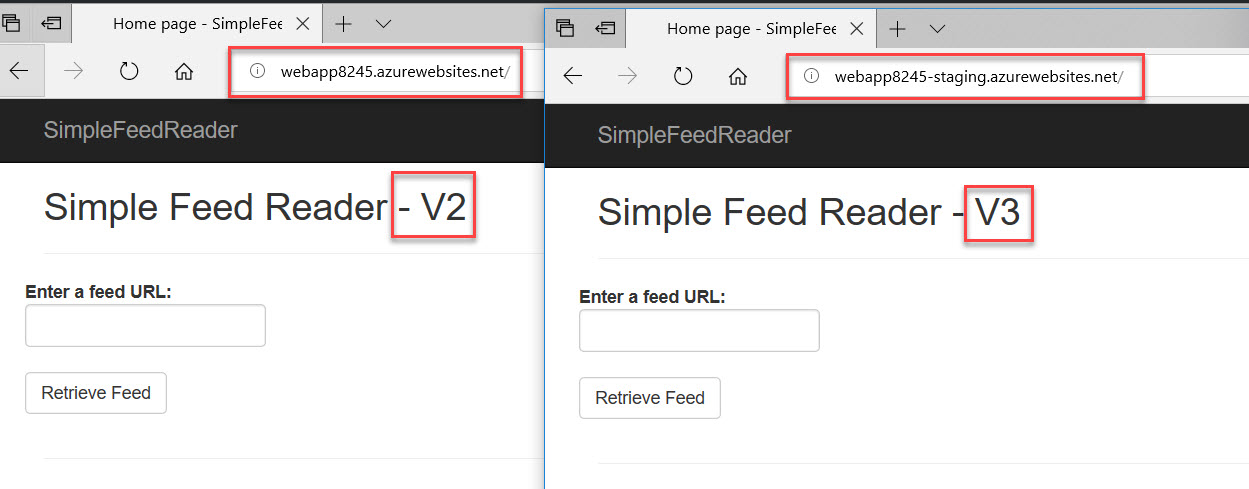


Step 37: Again, enter credentials





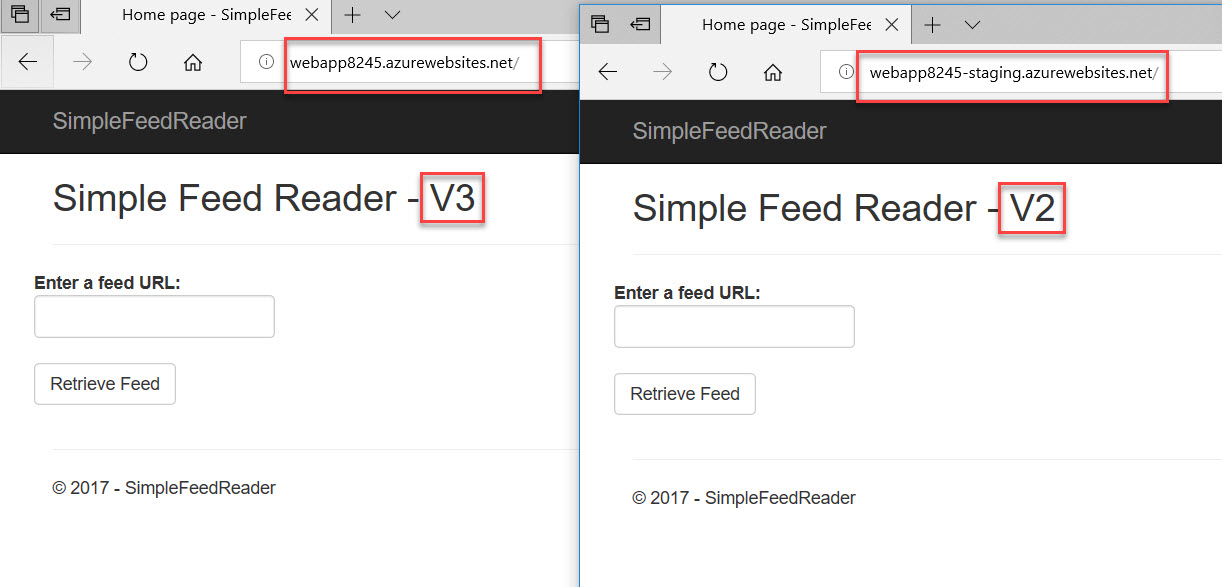
Step 38: Open Web App URL and Staging URL



Step 39: Now swap the content. Back to Azure Portal and type into Cloud Shell

az webapp deployment slot swap --name $webappname --resource-group $rg --slot staging

Step 40: Check both the web app



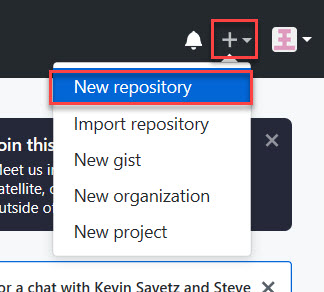
**Continuous Integration & Deployment**

Step 41: Open Github site: <https://github.com>

Note: if you don’t have account please create new one.

Step 42: Create New repository

Click on **+** icon (right-top side) and click on **New repository** option

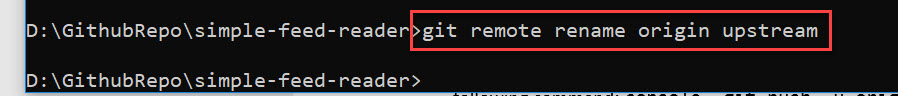


Step 43: Enter repository name: **simple-feed-reader** and click on **Create repository** button



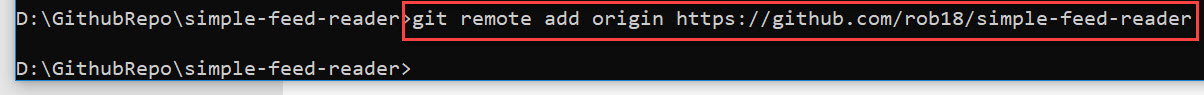
Step 44: Navigate to Command Prompt of machine and run below command to upstream

git remote rename origin upstream



Step 45: Add a new origin remote pointing to your copy of the repository on Github.

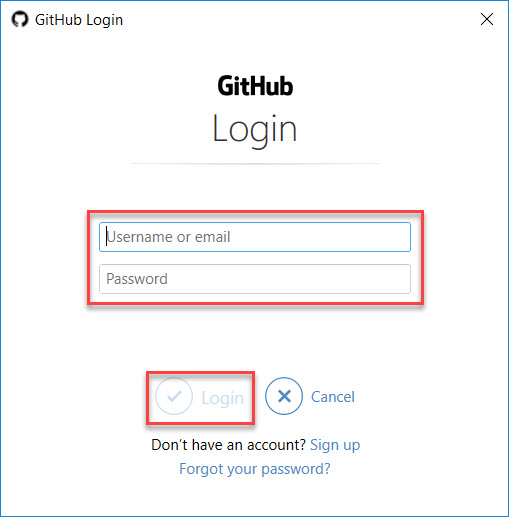
git remote add origin https://github.com/<GitHub\_username>/simple-feed-reader



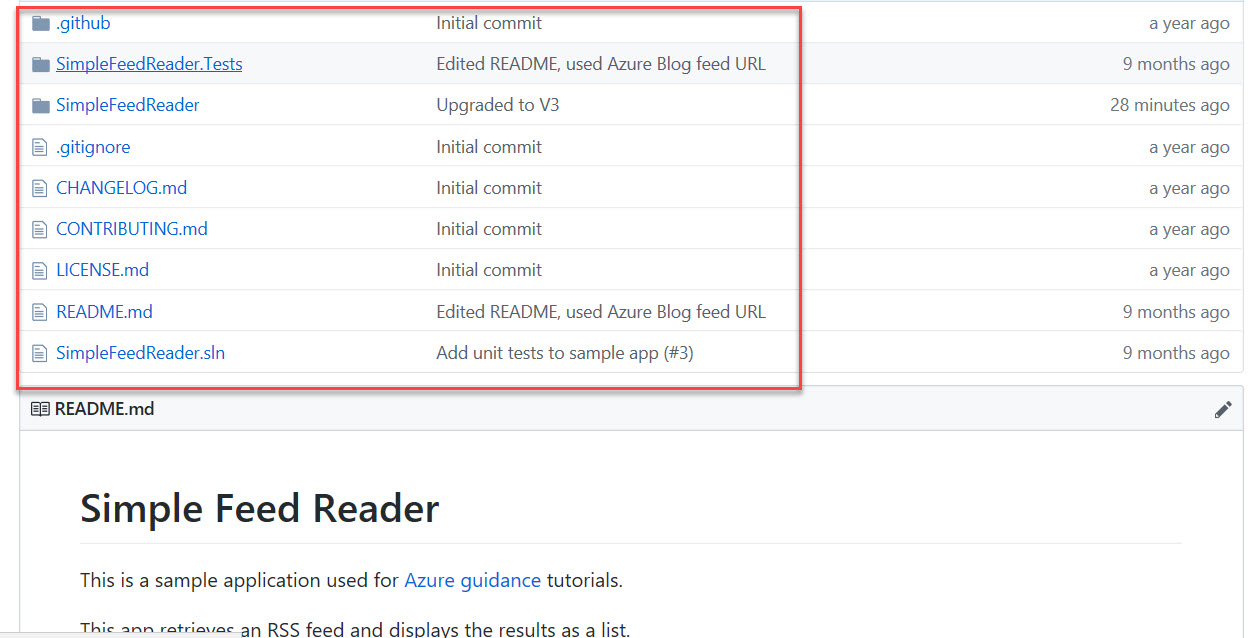
Step 46: Publish your local Git repo to newly created Github Repo.



Enter Github Credentials



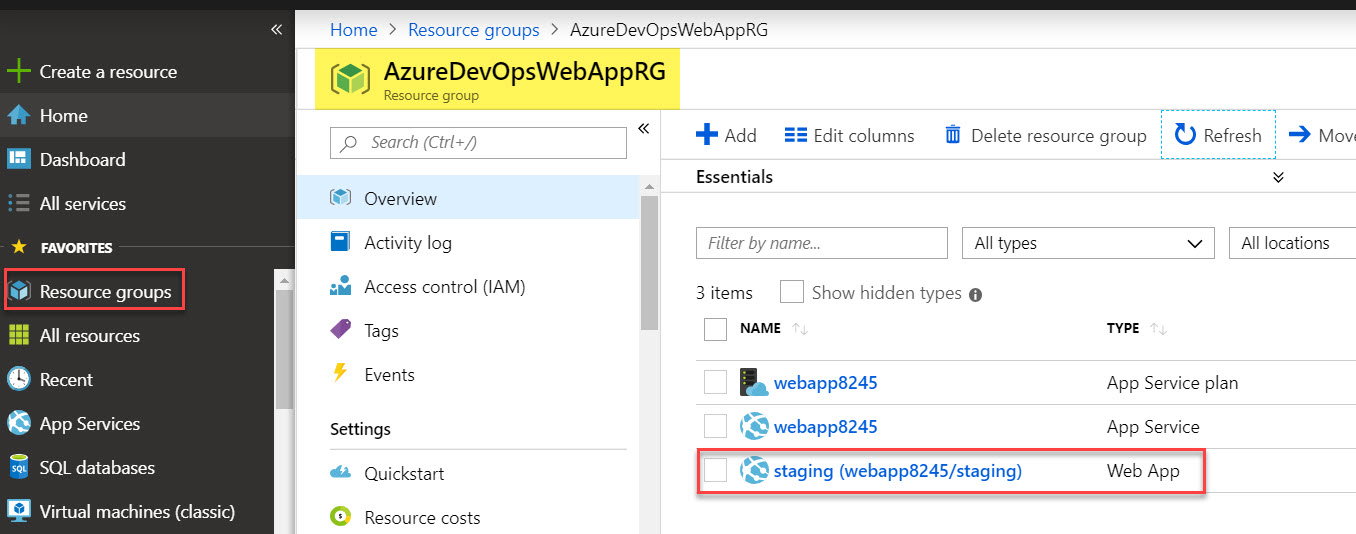
Step 47: Navigate to github site to check the contents.



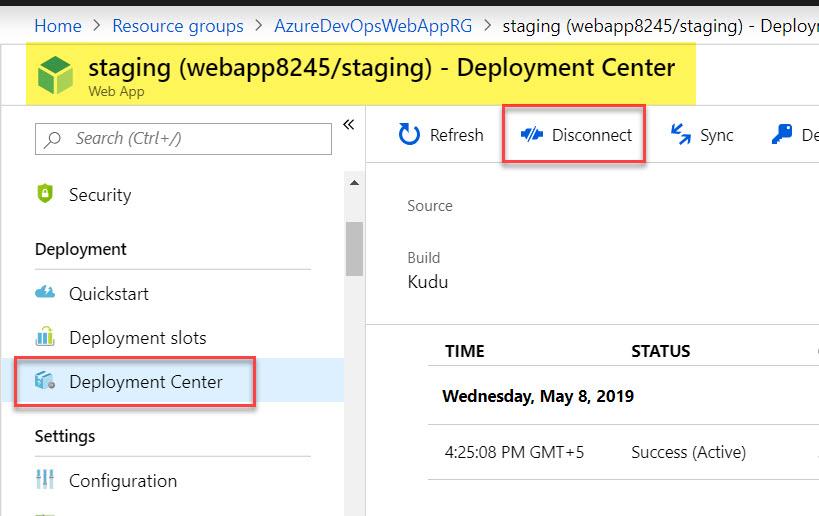
**Disconnect local Git Deployment**

Step 48: Navigate to Azure Portal and select Staging Web App

**Resource Groups -> AzureDevOpsWebAppRG -> Select Staging Web App**

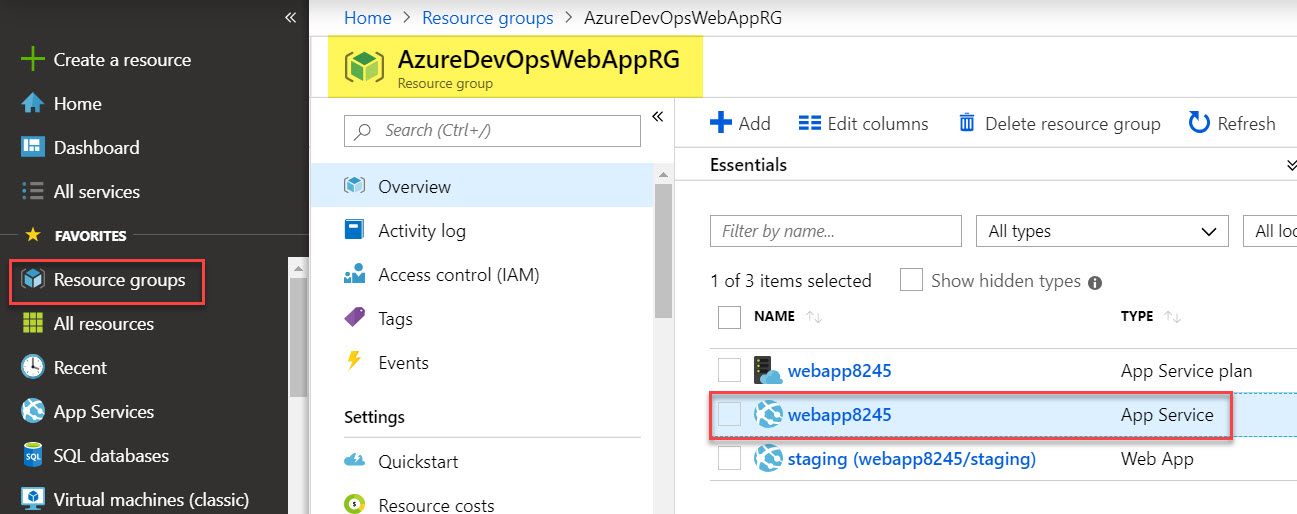


Step 49: Scroll down and select **Deployment Center** option and click on **Disconnect** button.

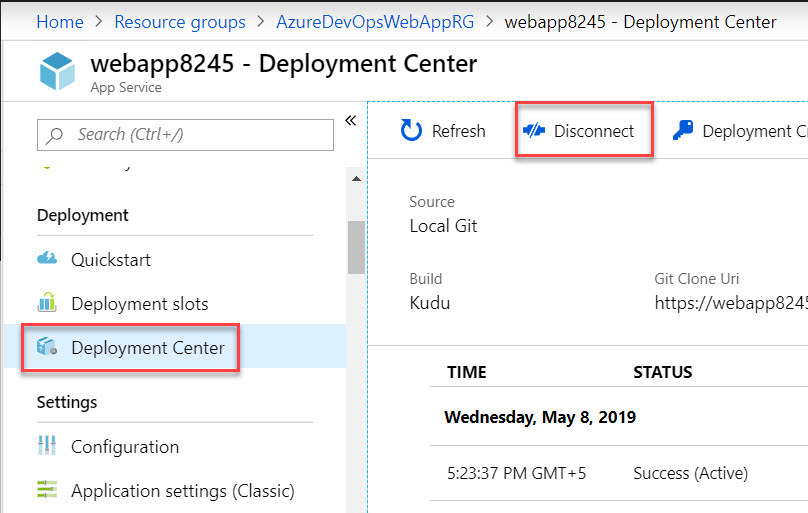


Step 50: Now Select **Web App**

**Resource Groups -> AzureDevOpsWebAppRG -> Select Web App**



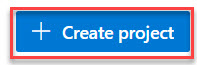
Step 51: Scroll down and select **Deployment Center** option and click on **Disconnect** button.



**Create Azure DevOps account**

Step 52: Navigate to <https://dev.azure.com/>

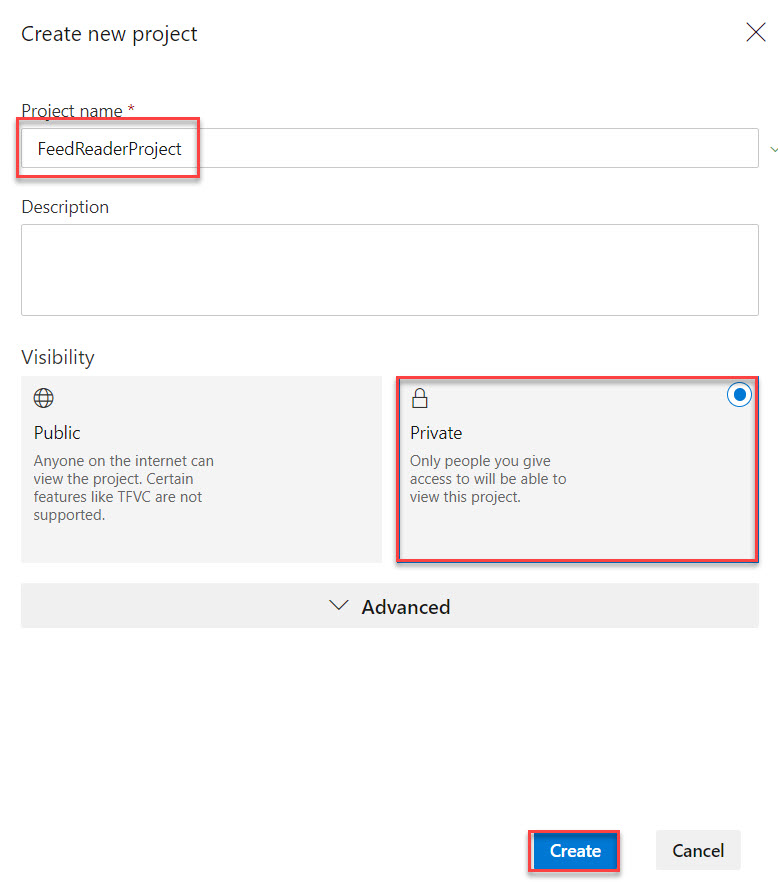
Step 53: **Create New Project**



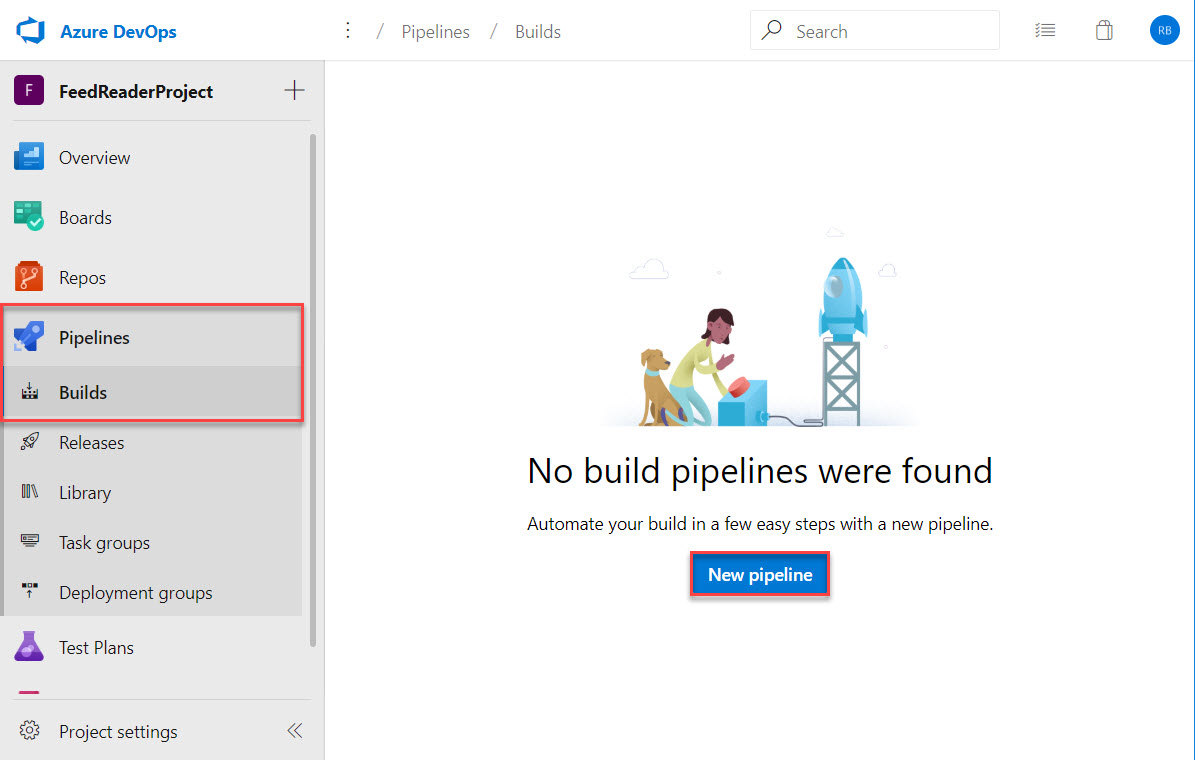
Step 54: Create New Project.

Project Name: **FeedReaderProject**

Visibility: **Private**

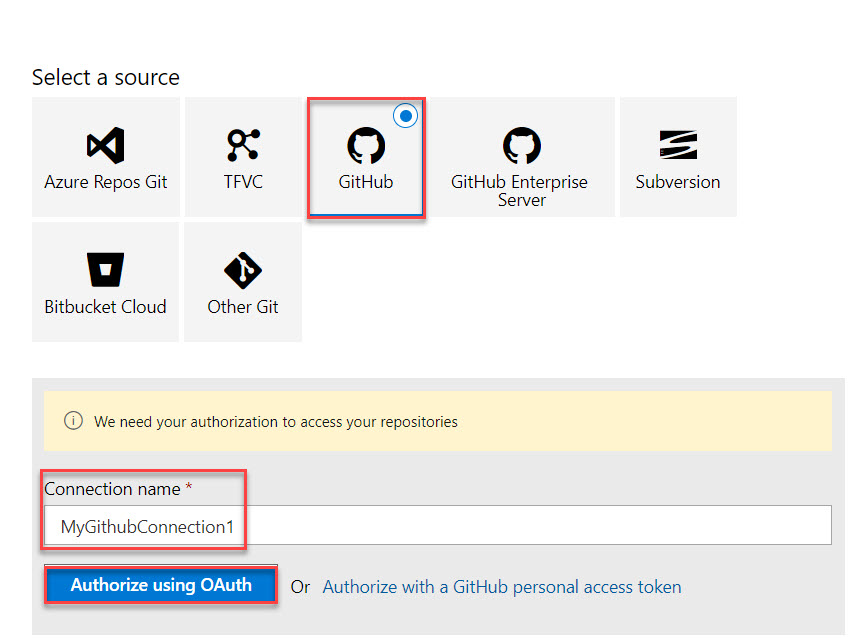


Step 55: Select **Pipelines -> Builds** and Click on **New Pipeline** button.

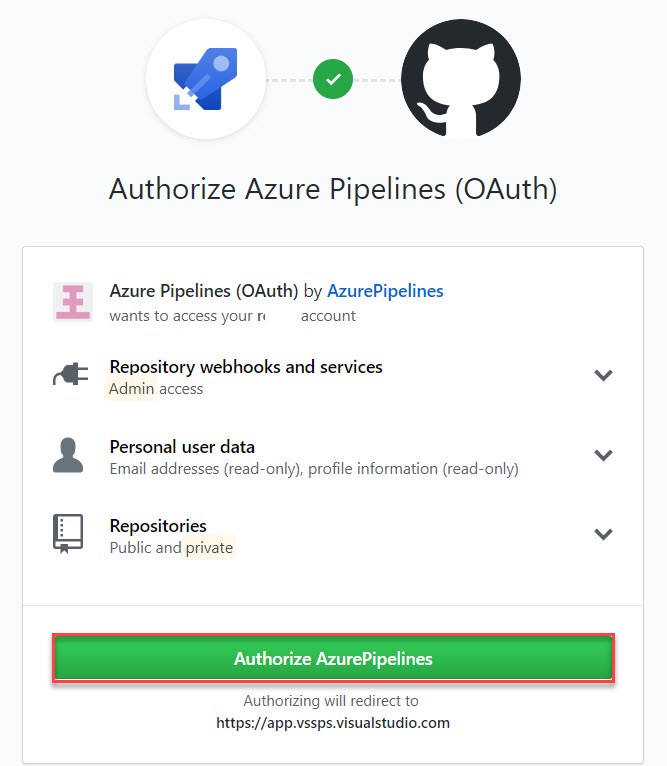


Step 56: Select Source: **GitHub** and Enter Connection Name: **MyGithubConnection**

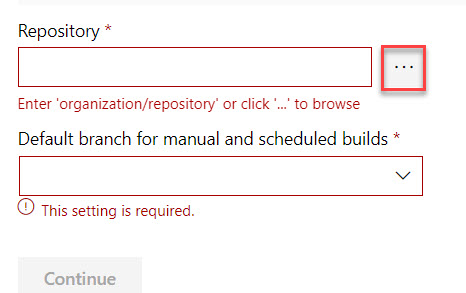
Click on **Authorize using OAuth** button



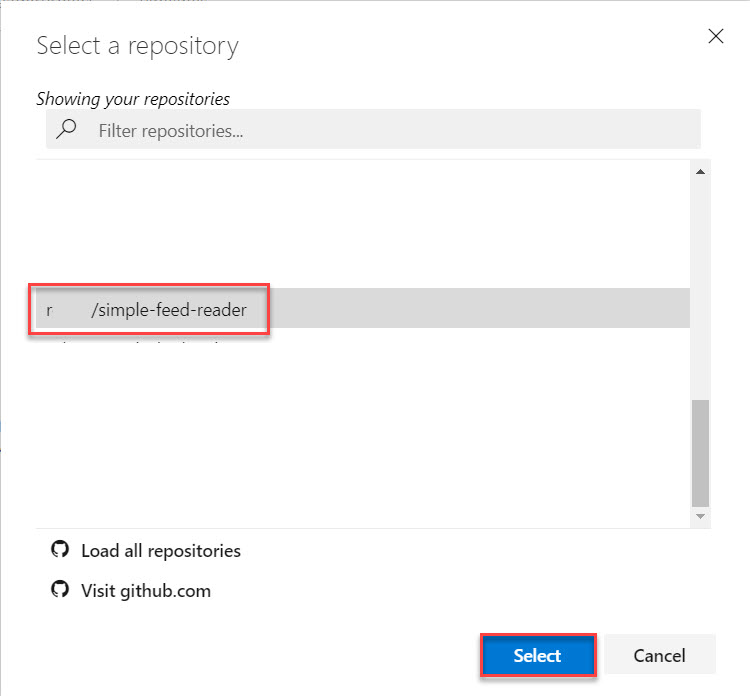
Step 57: Click on **Authorize AzurePipelines**



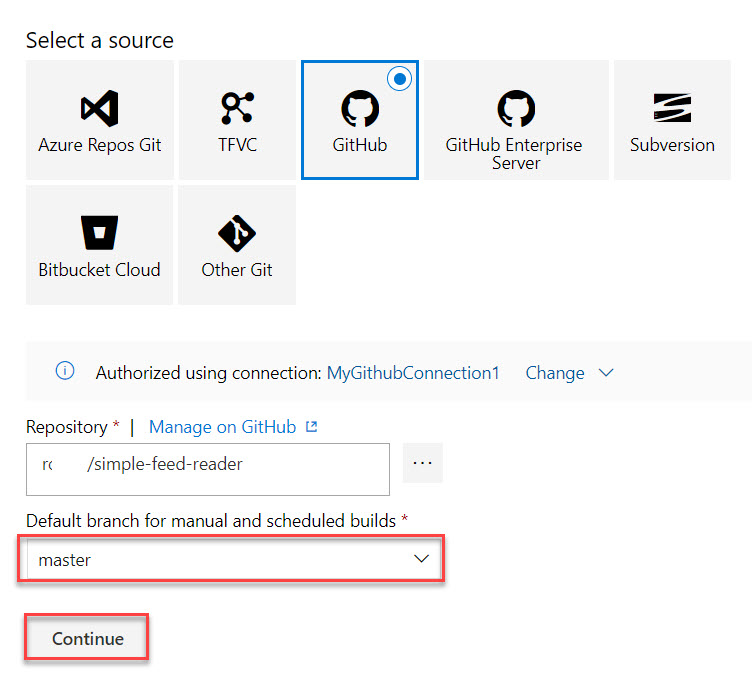
Step 58: Click on **…** option



Step 59: Select **your repository** which is recently created and click on **Select** option

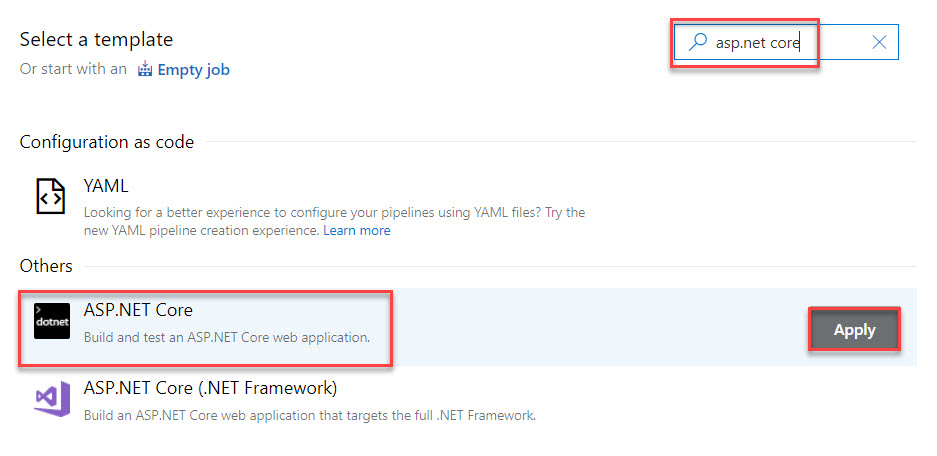


Step 60: Select Branch: **Master** and click on **Continue** button.

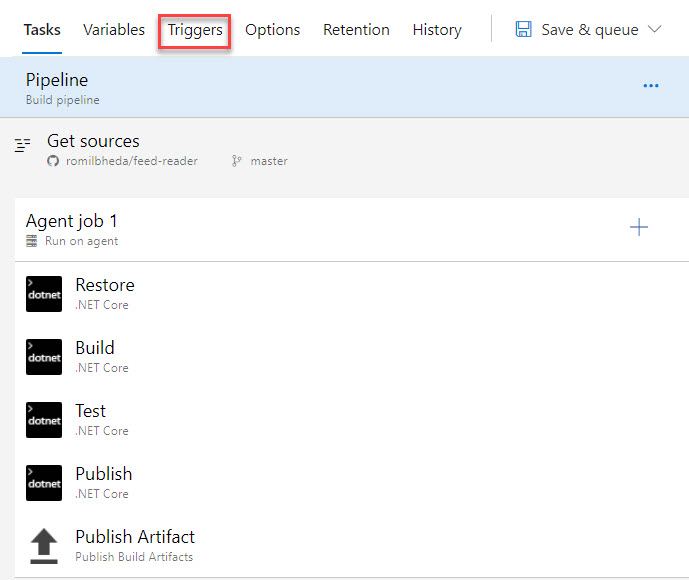


**Creating Build Definition**

Step 61: Search for **ASP.NET Core** and Select **ASP.NET Core (.NET Framework)**. Click on **Apply** button.

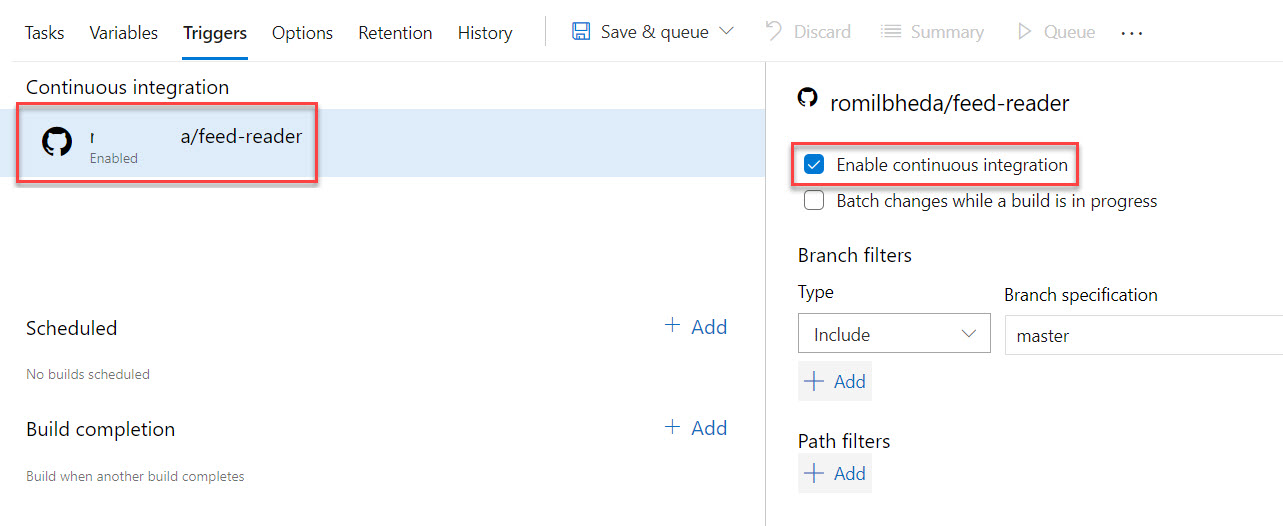


Step 62: Check the **Tasks** of Build pipeline. Click on **Triggers**

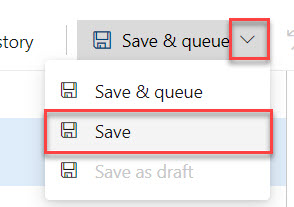


Step 63: Check mark on **Enable Continuous integration**

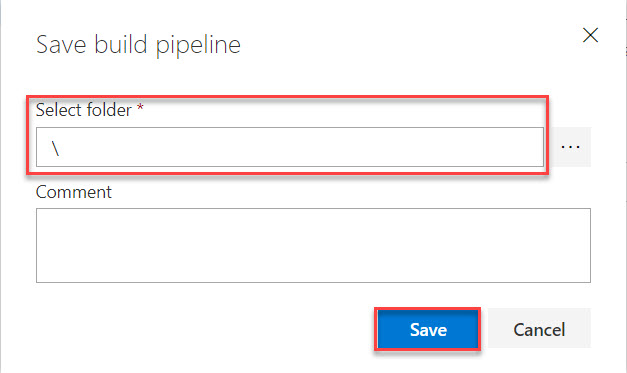
Type: Include and Branch specification: master



Step 64: Click on **Save** button.

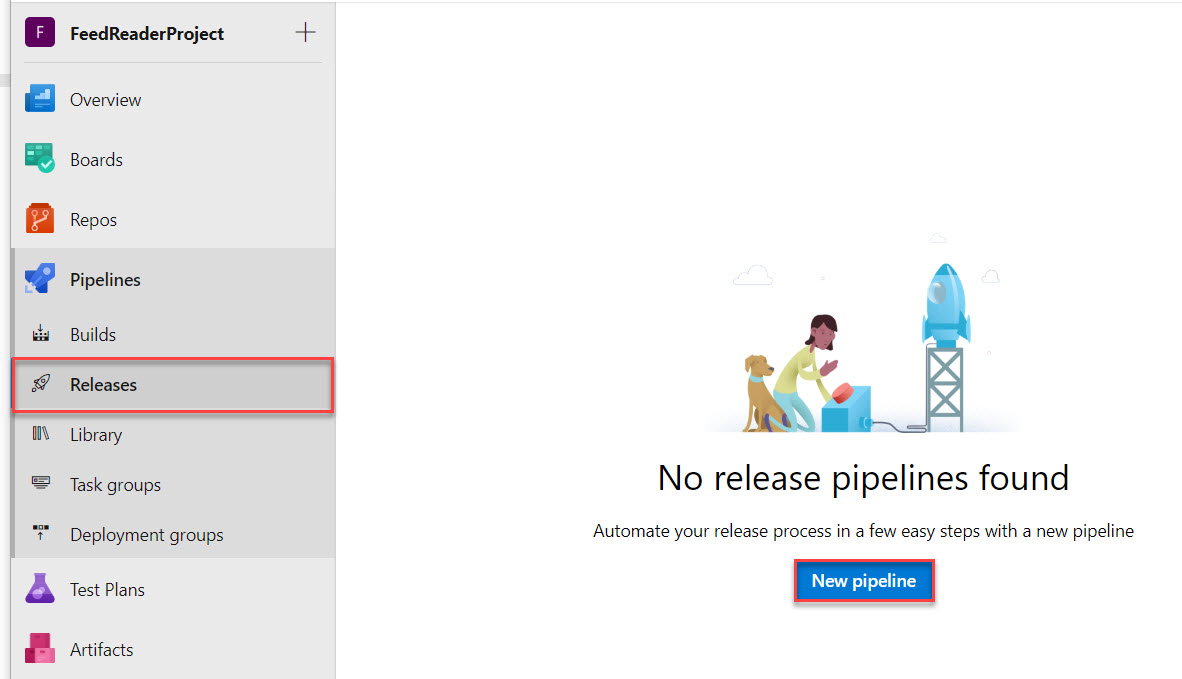


Step 65: Continue with default folder \ and click on **Save** button.



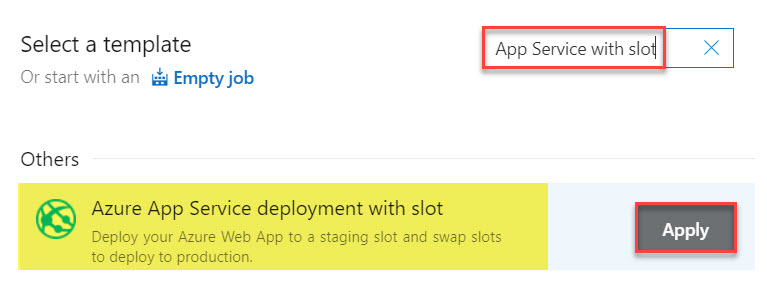
**Continuous Deployment – Release Pipeline**

Step 66: Select **Release** option click on **New Pipeline**

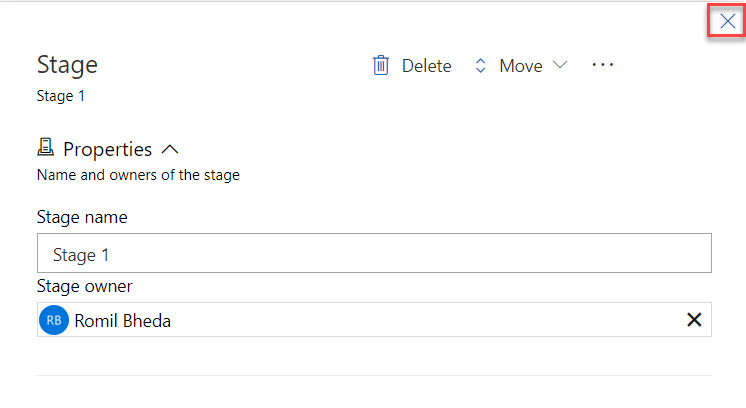


Step 67: Search for **App Service with Slot**

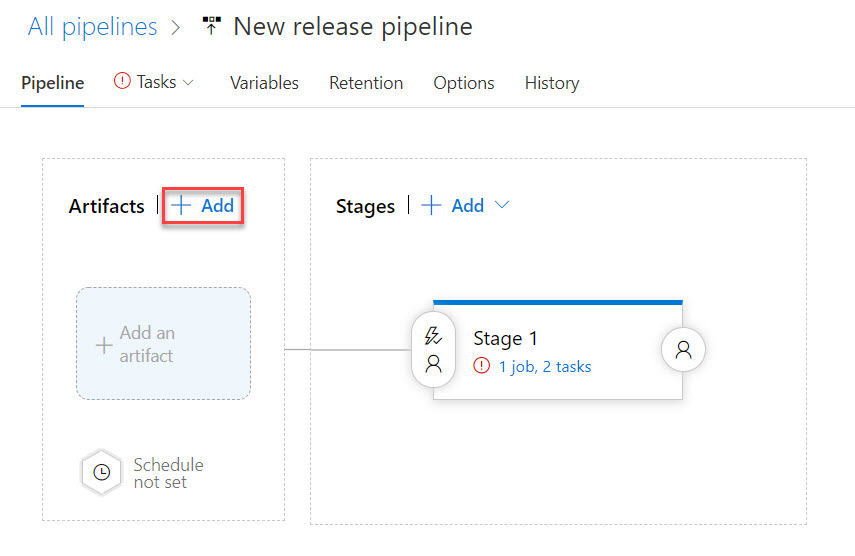
Select **Azure App Service deployment with slot** and click on **Apply** button.



Step 68: Click on **Close** button.



Step 69: Now click on **+ Add** in Artifacts



Step 70: Add an artifact

Source type: **Build**

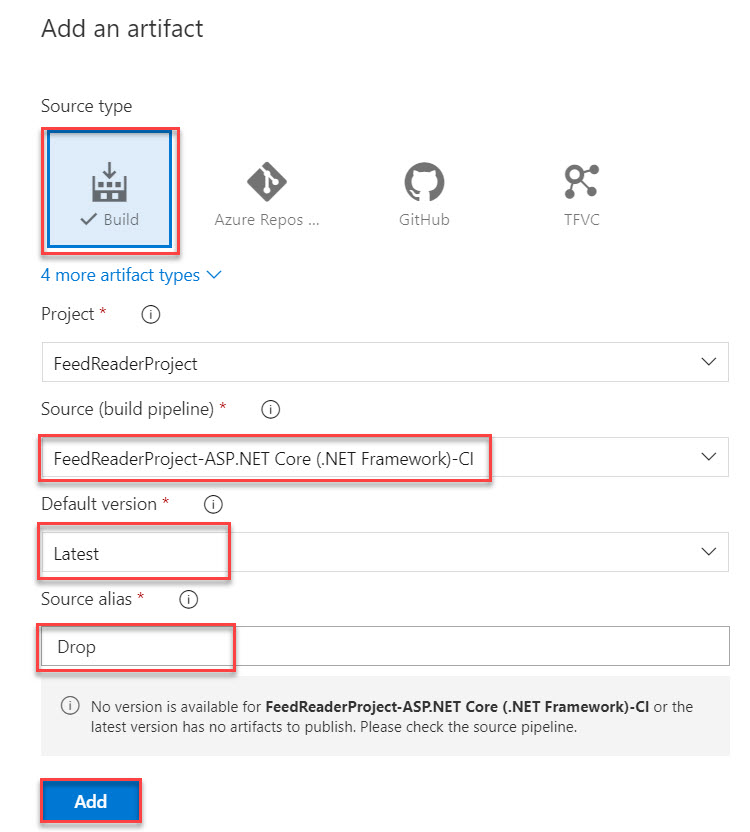
Project: **FeedReaderProject**

Source (build pipeline): **FeedReaderProject-ASP.NET Core (.NET Framework)-CI**

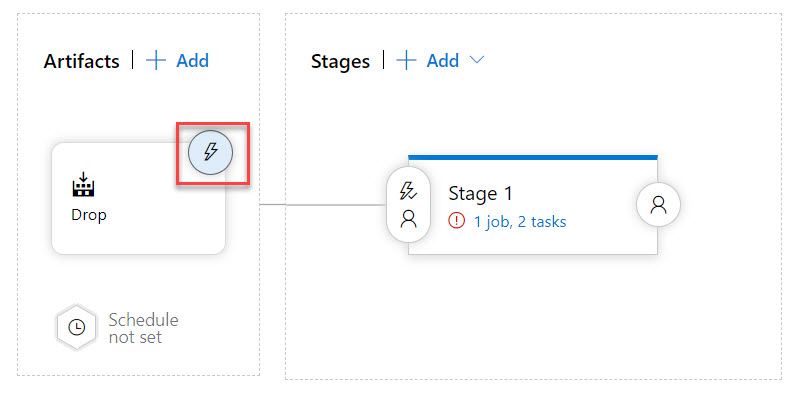
Default version: **Latest**

Source alias: **Drop**

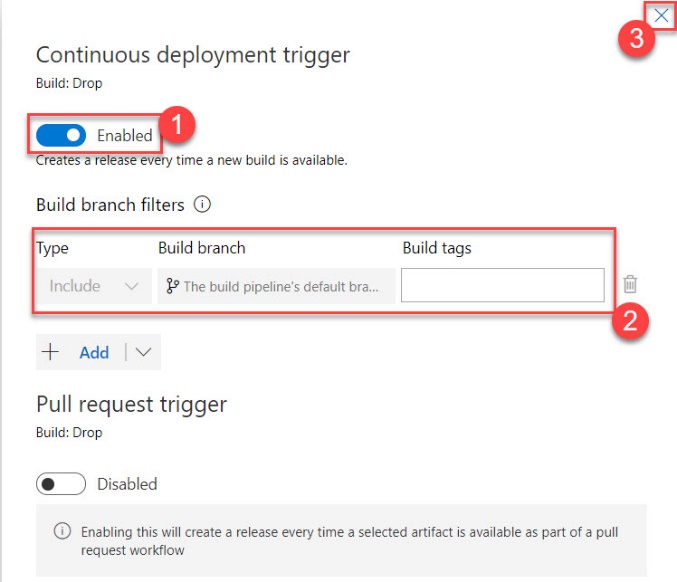
Click on **Add** button.



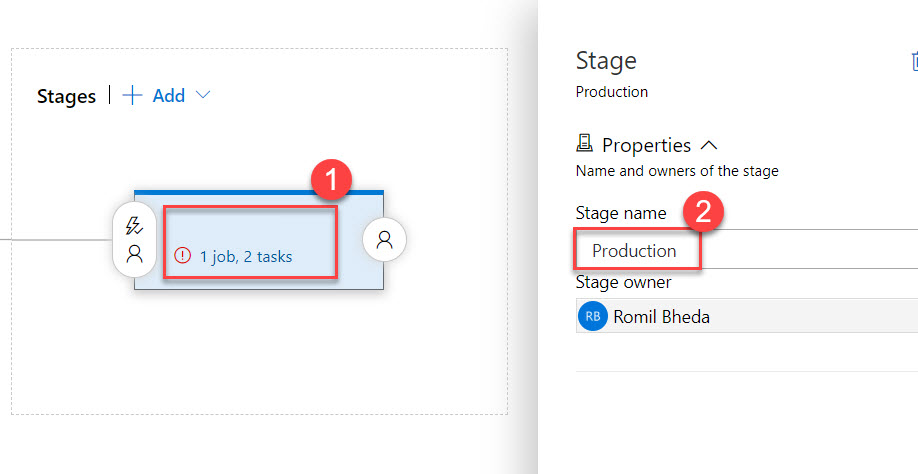
Step 71: Click on **Thunder** icon in **Artifacts**



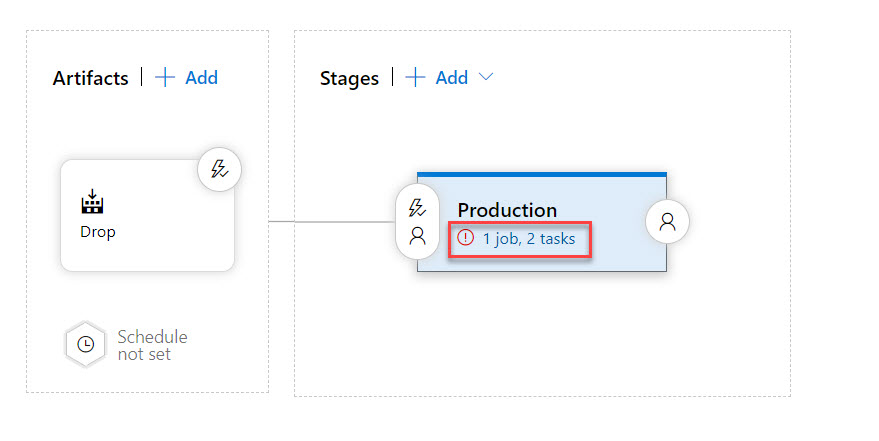
Step 32: **Enable Continuous Deployment** **trigger** and click on **Close** button.



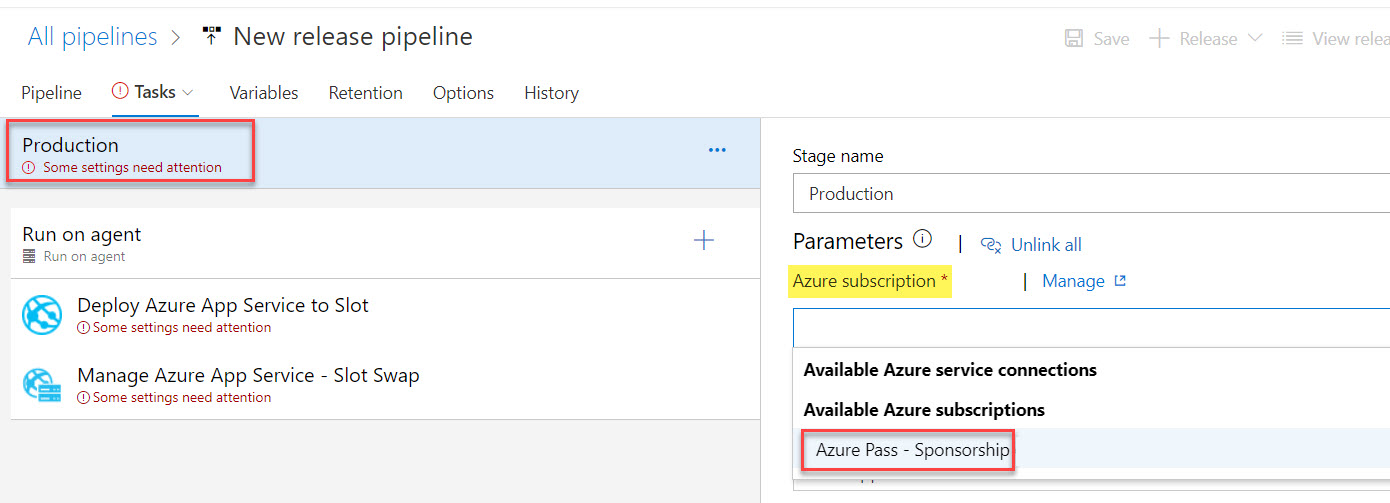
Step 33: First Select Stage and enter stage name: **Production**



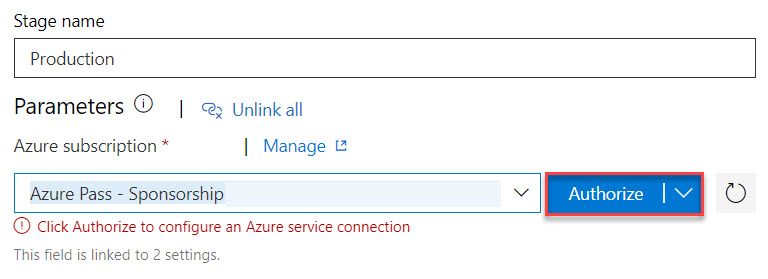
Step 34: Click on **1 job, 2 tasks**.



Step 35: Select **Production** and select **Azure Subscription** from list.



Step 36: Click on **Authorize** button.

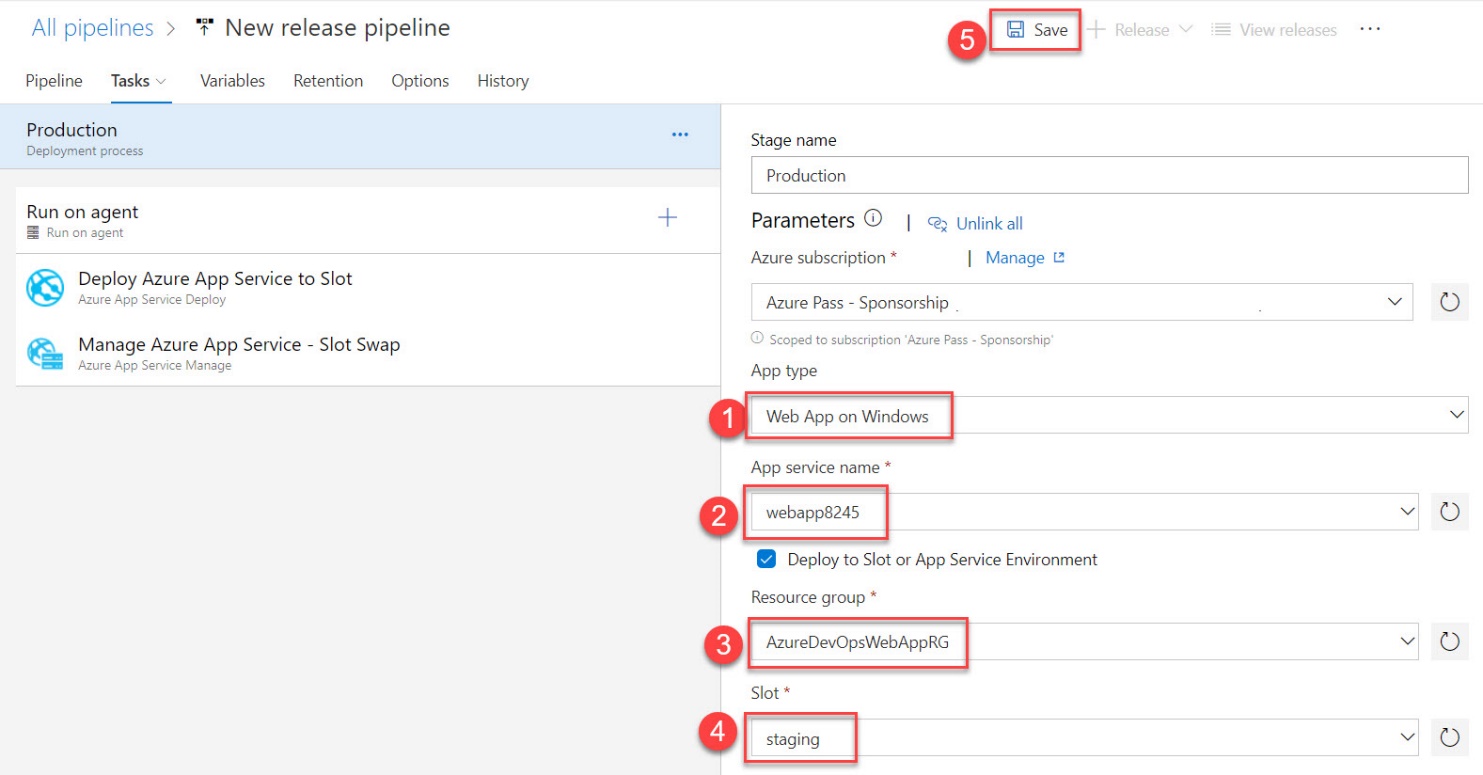


Step 37: App Type: **Web App on Windows**

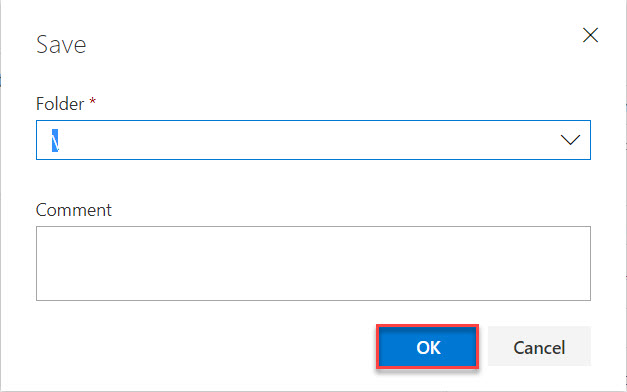
App Service name: Select from list

Resource Group: Select from list. Ex. **AzureDevOpsWebAppRG**

Slot: **Staging**

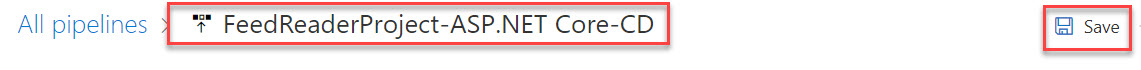


Step 38: continue with default option folder \ and click on **OK** button.



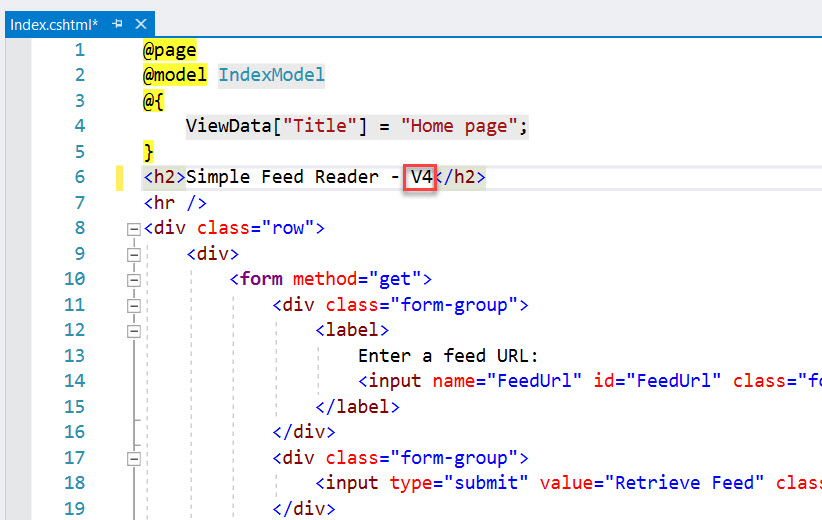
Step 39: Edit Pipeline name

**FeedReaderProject-ASP.NET Core-CD** and click on **Save** button.



Step 40: Commit changes to Github and automatically deploy to Azure

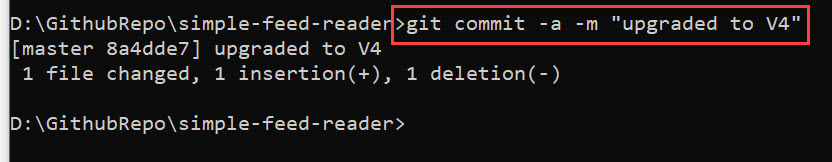
Navigate to Visual Studio and modify text **– V4**



Now **Build** the Solution

Step 41: Navigate to Command Prompt and type below command:

git commit -a -m "upgraded to V4"



Step 42: Push the change in the master branch to the origin remote of Github repo

git push origin master

Step 43: Now check Azure DevOps Build and Release Pipeline.

Step 44: Also check website.