Web Apps using VS and Azure

Lab 03
by
Andrea Hatch, Nishava Inc.

Deep Azure @McKesson

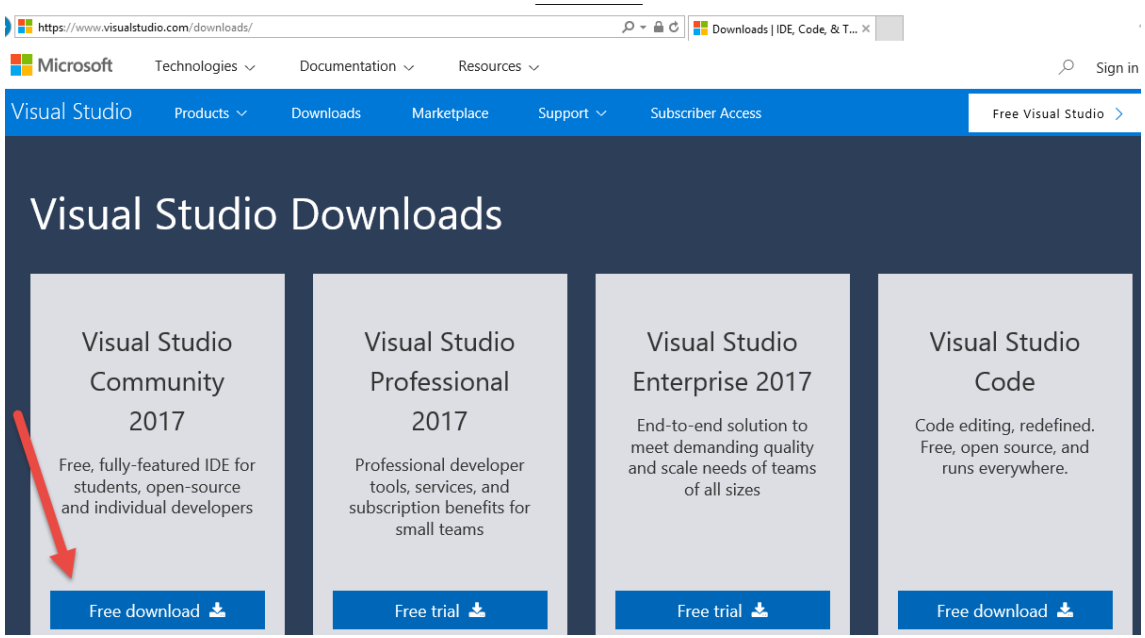
Overview

- Download Visual Studio and connect with Azure
- Add Ons, Extensions and Updates
- Create a web app in VS to deploy to Azure
- Create a web app in Azure and Update the web app in VS

1. Download Visual Studio and connect with Azure

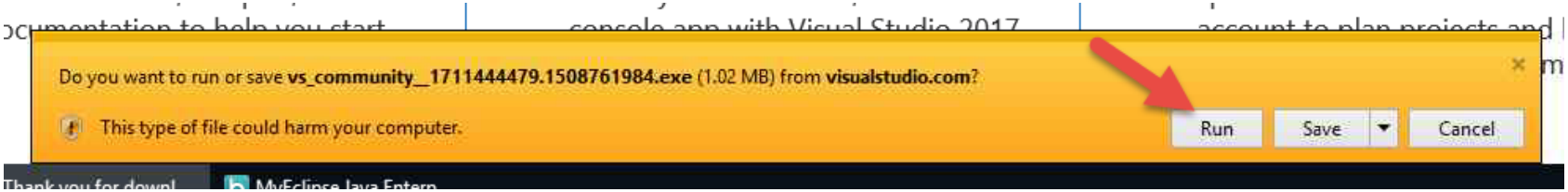
Download Visual Studio

- https://www.visualstudio.com/downloads/
- Select the Free download under Visual Studio Community 2017
- Note: If you already have an older version of VS that will work too



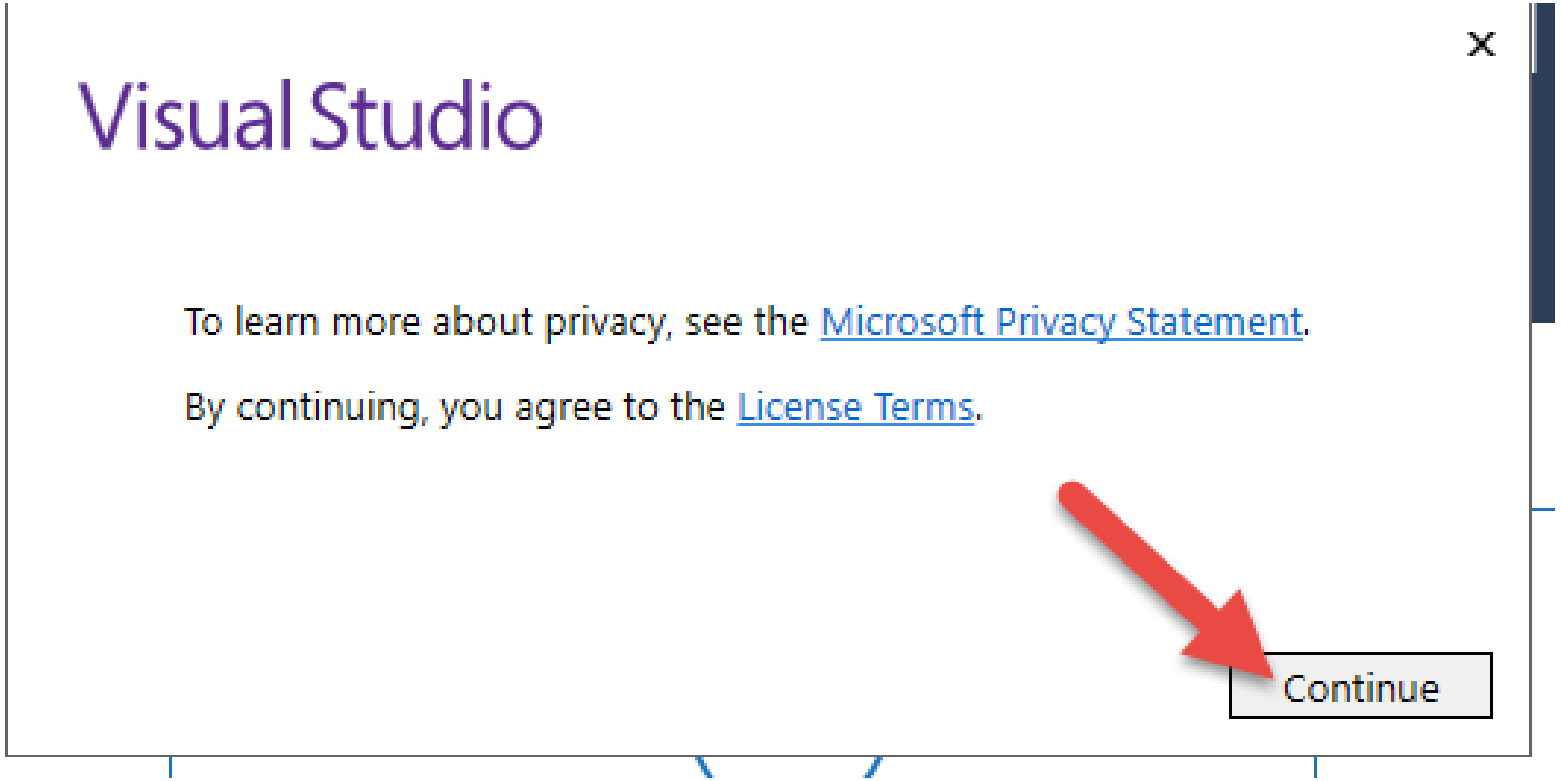
Download Visual Studio

- Select ‘Run’ on the pop-up to install Studio



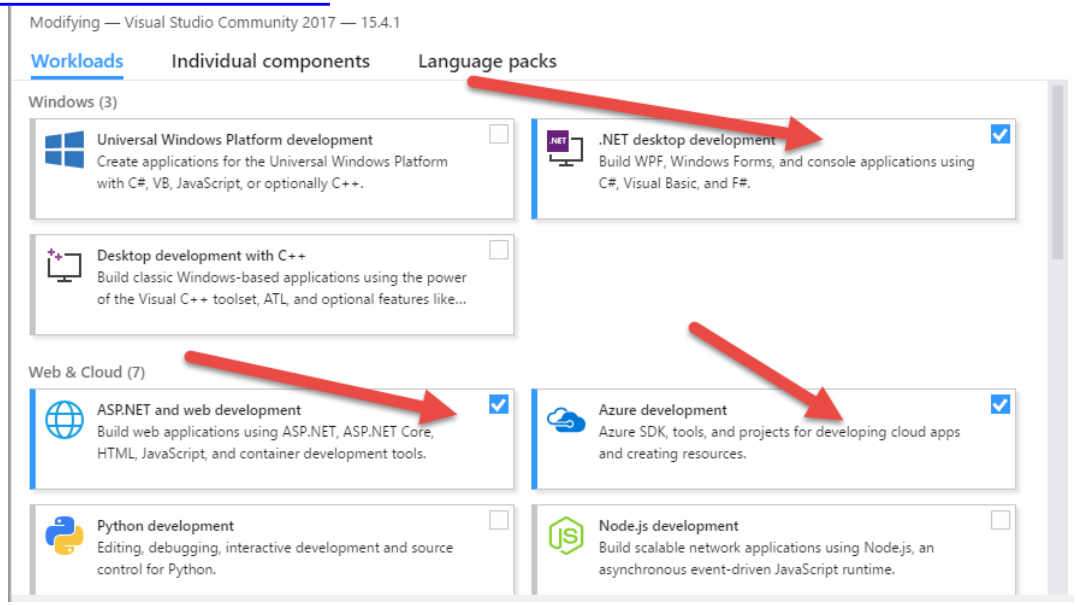
Download Visual Studio

- Agree to the licensing terms by selecting continue



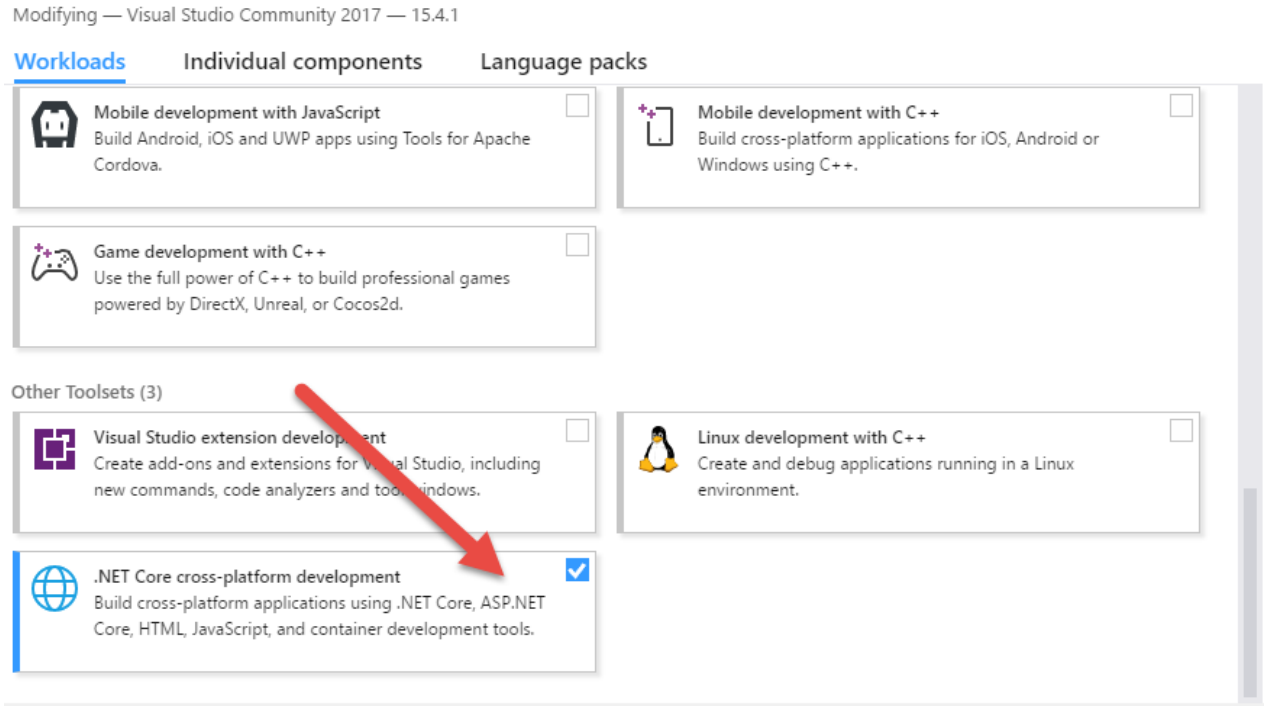
Download Visual Studio

- On 2017 - you will then be asked if you would like to add any workloads to your VB install, select the.NET desktop development, ASP.NET and web development and Azure development then Scroll down
- If you have 2015 or 2013 already you will need to install the respective SDK's for Azure
  - 2015: http://go.microsoft.com/fwlink/?linkid=518003
  - 2013: http://go.microsoft.com/fwlink/?LinkID=324322



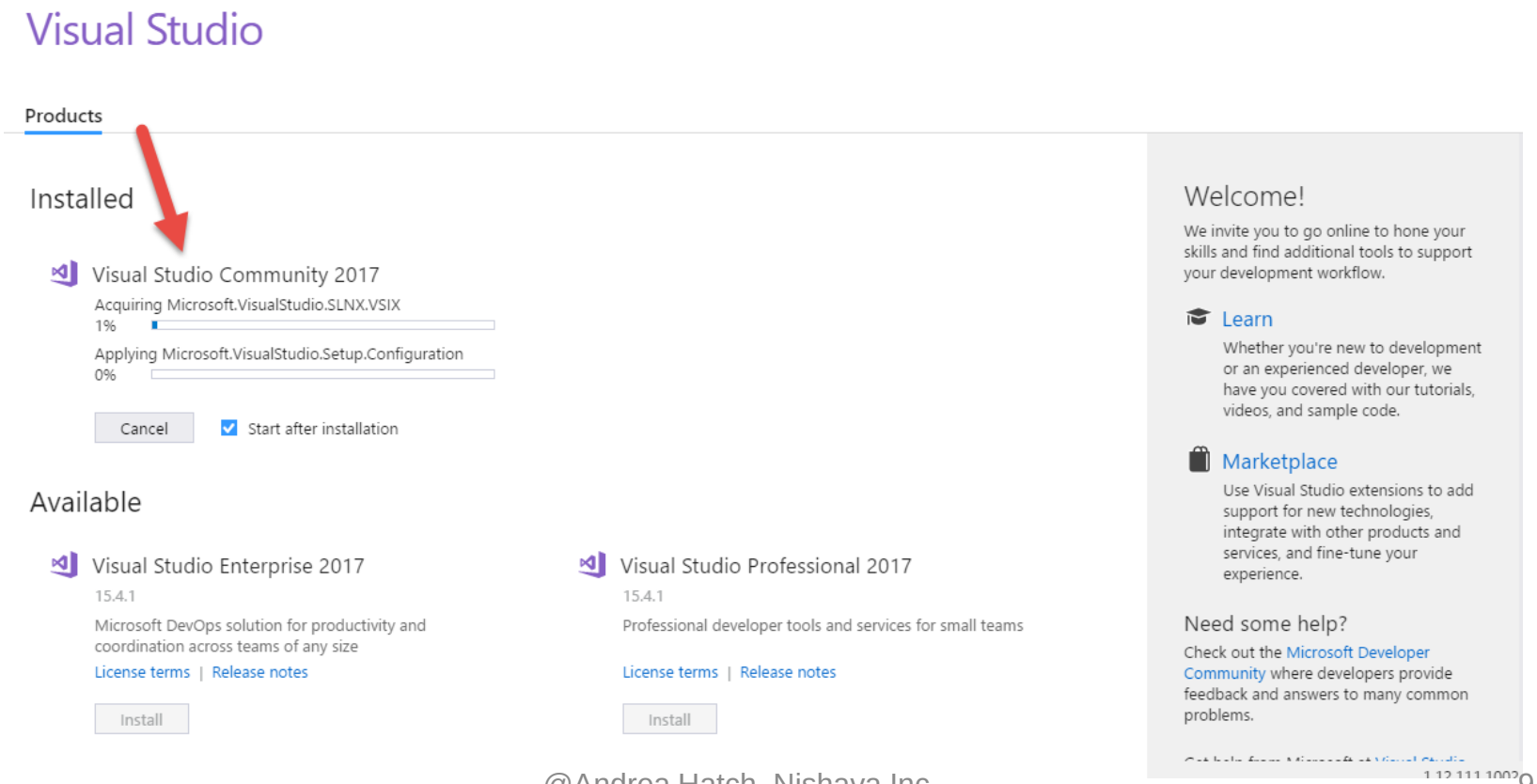
Download Visual Studio

- Under Other Toolsets make sure to also select .NET Core cross-platform development
- Note: if you miss selecting anything you can always ad them later by Tools – Get Tools and Features



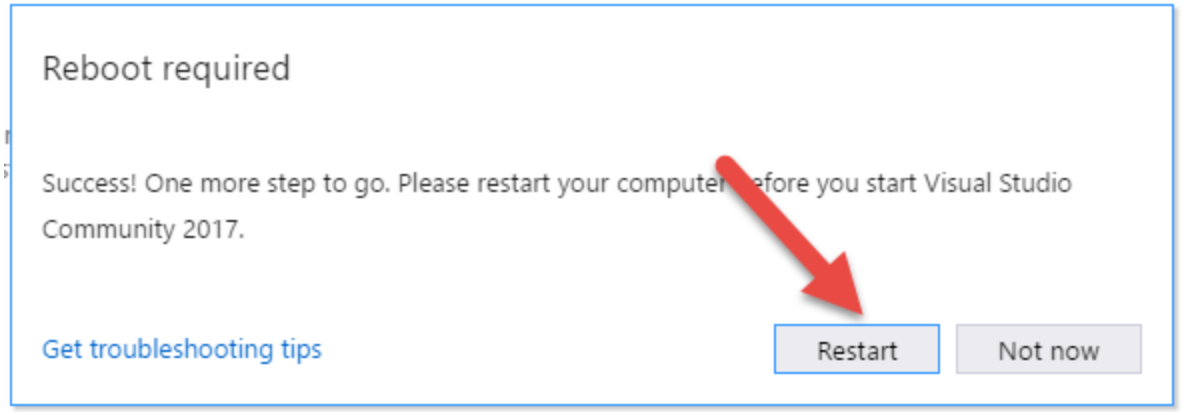
Download Visual Studio

- Your download may take a while to complete and should look similar to this.



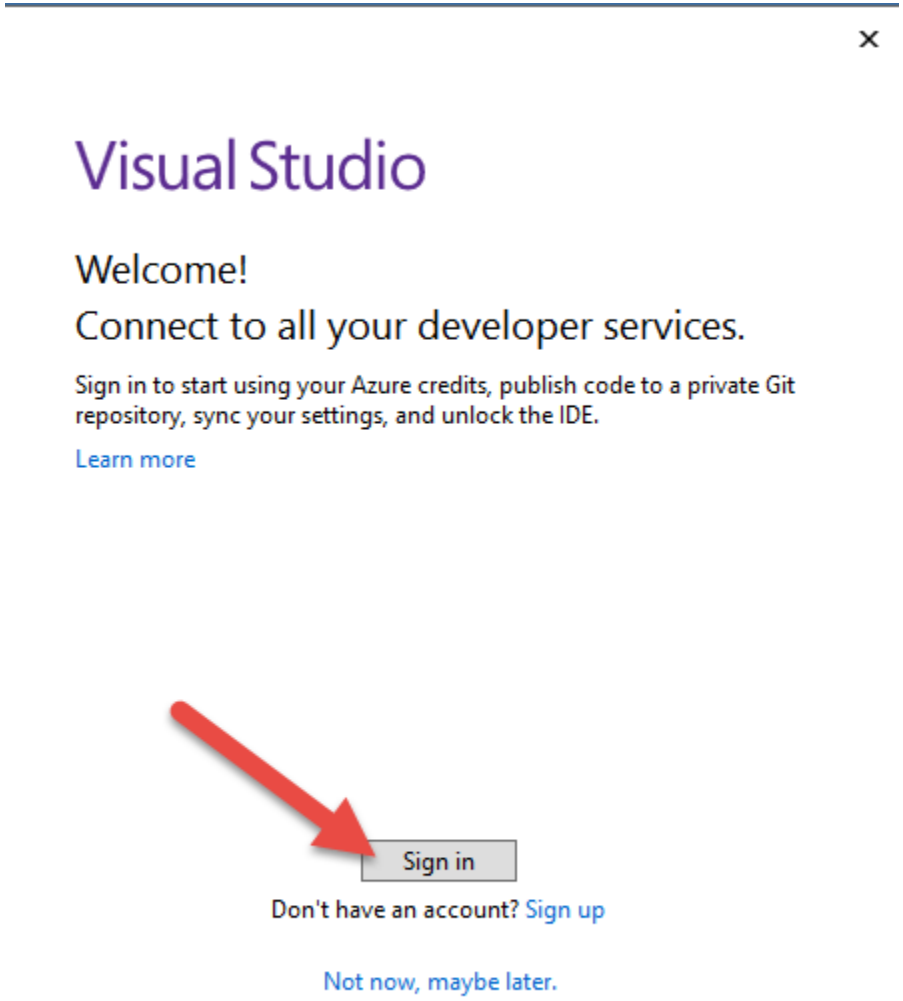
Download Visual Studio

- You may be asked to restart before you can use Visual Studio once the installation is complete.



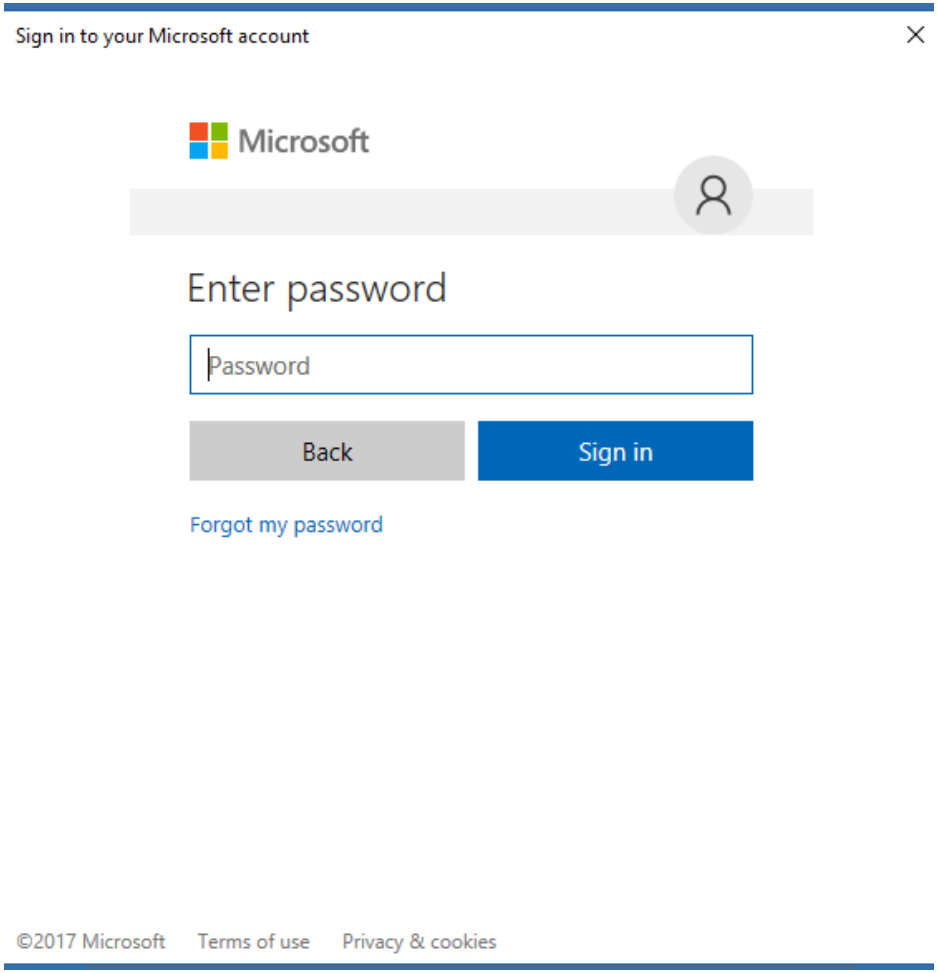
Connect Azure and VS

- On your first log in of VS you can connect your Azure account



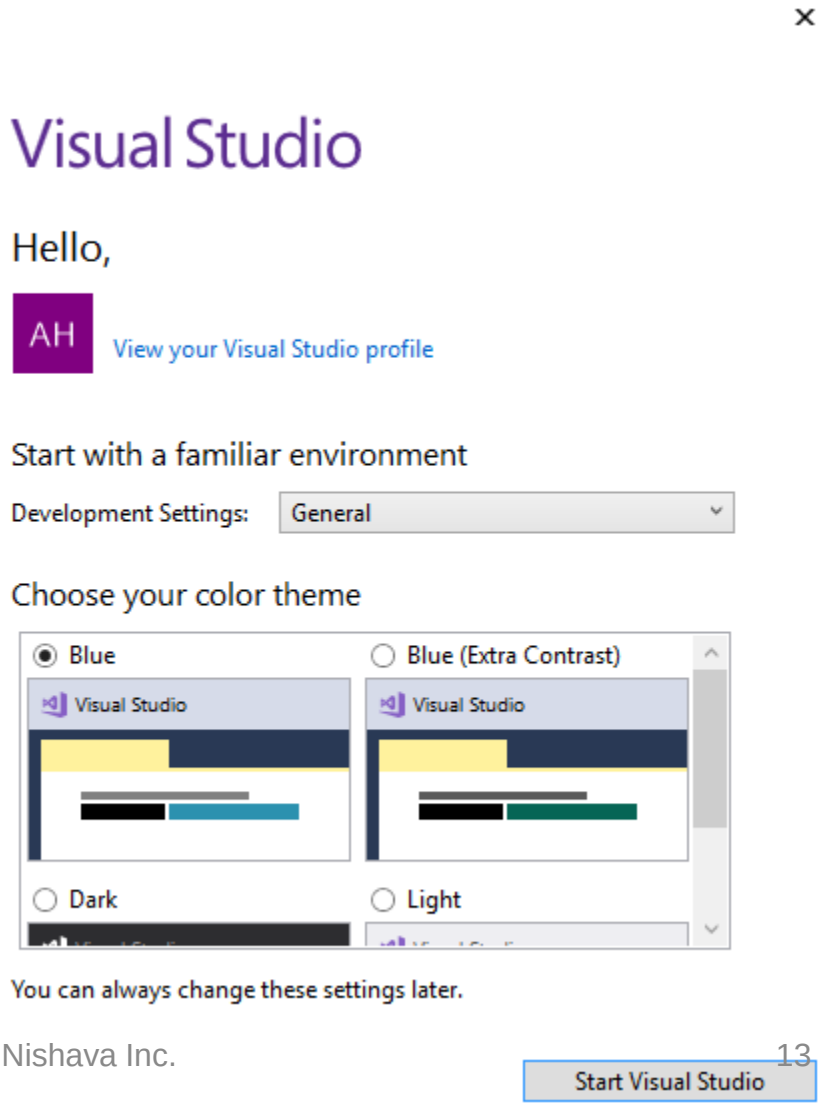
Connect Azure and VS

- You should recognize this screen from signing in to your Azure account in previous homeworks



Connect Azure and VS

- You can confirm you are connected and then pick a theme for your VS and then select ‘Start Visual Studio’



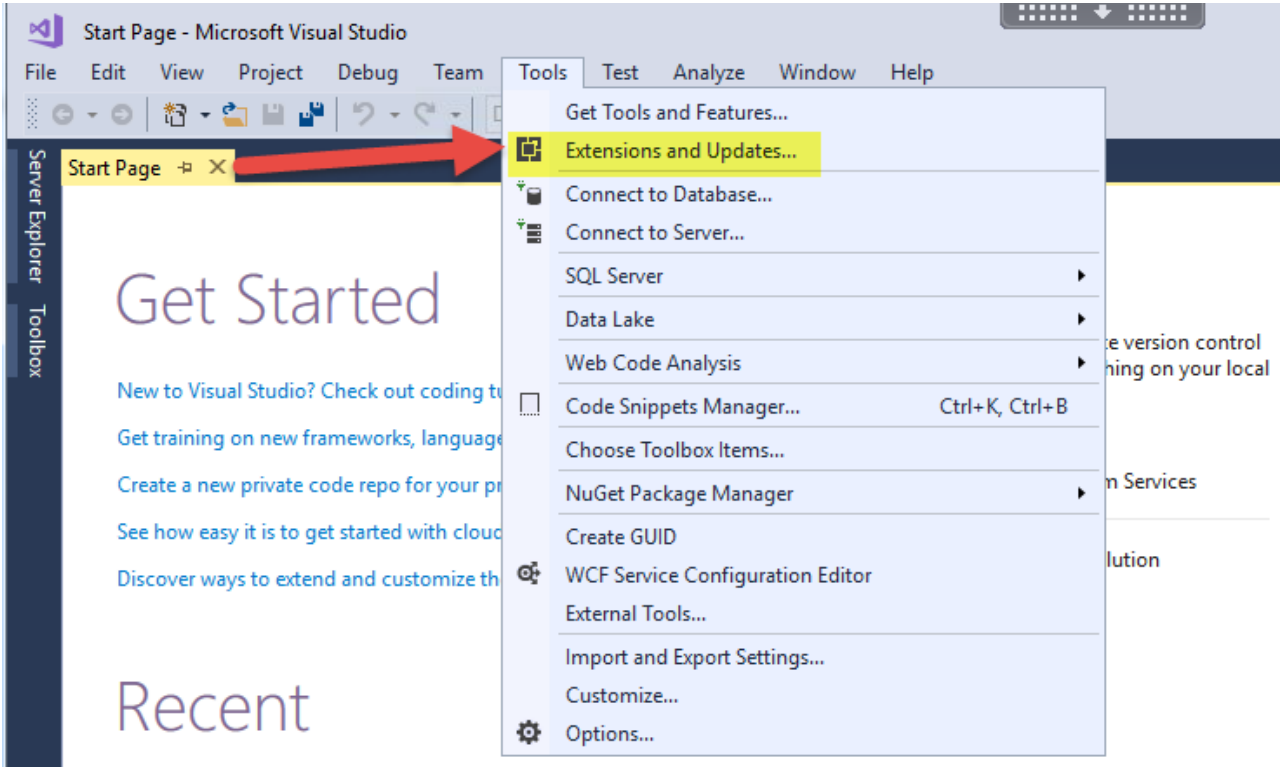
2. Add On’s, Extensions and Updates

Add On's to Download

- You will need to download the following in order to work with Git and .NET
  - GIT: https://git-scm.com/downloads
  - .NET Core 2.0: https://download.microsoft.com/download/F/6/E/F6ECBBCC-B02F-424E-8E03-D47E9FA631B7/DotNetCore.1.0.1-VS2015Tools.Preview2.0.3.exe

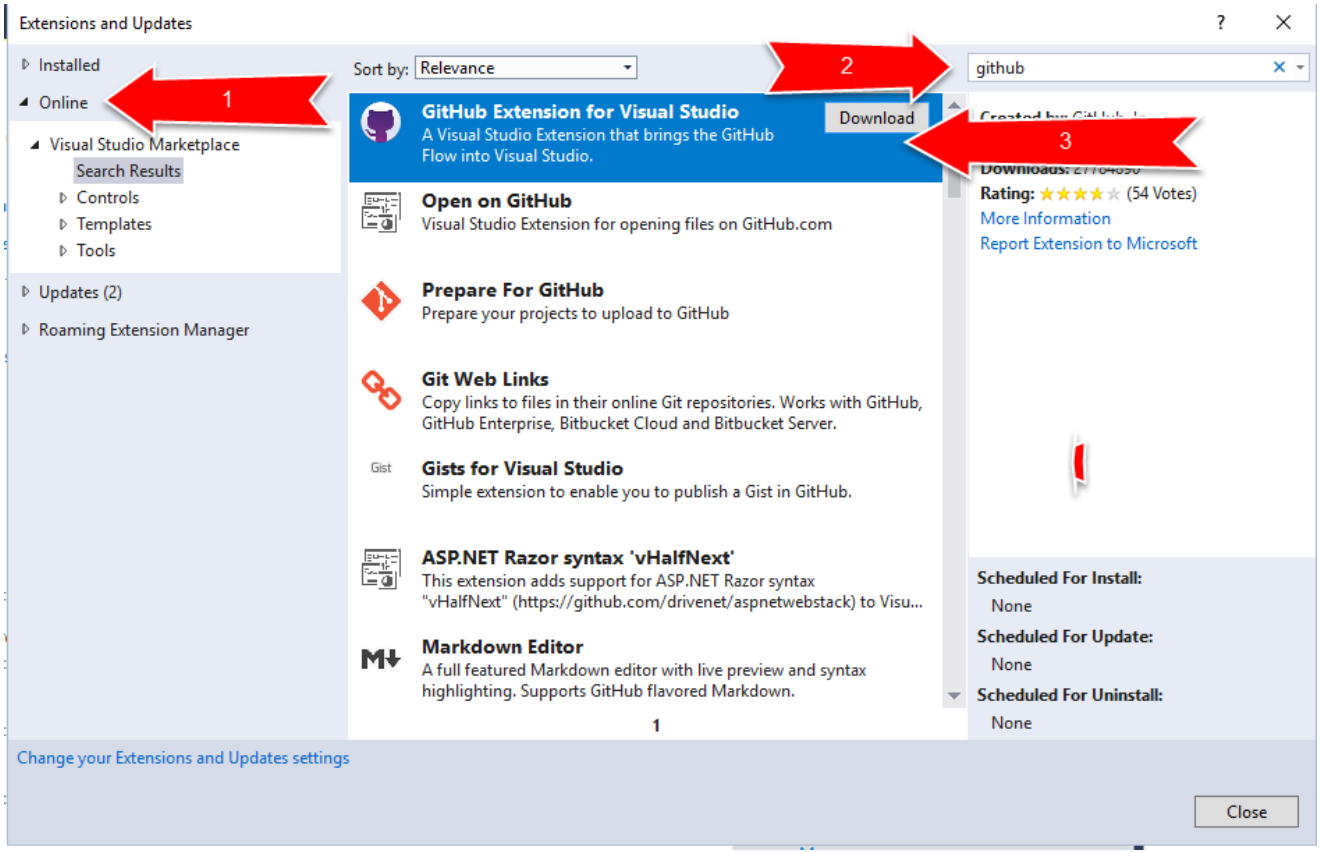
Add the GitHub Extension to VS

- In VS, under Tools select ‘Extensions and Updates’



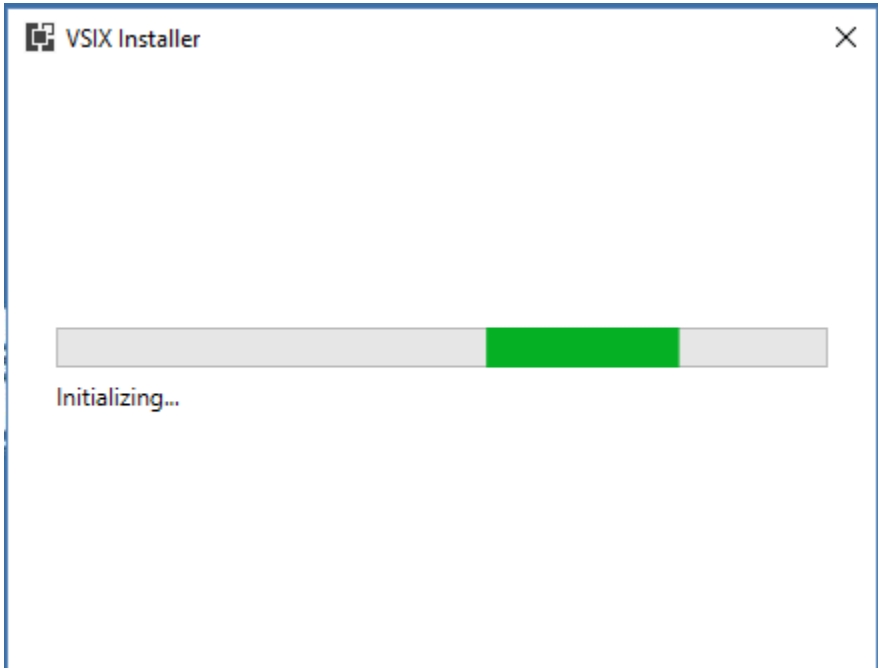
Add the GitHub Extension to VS

- Under Online search for github and then download it



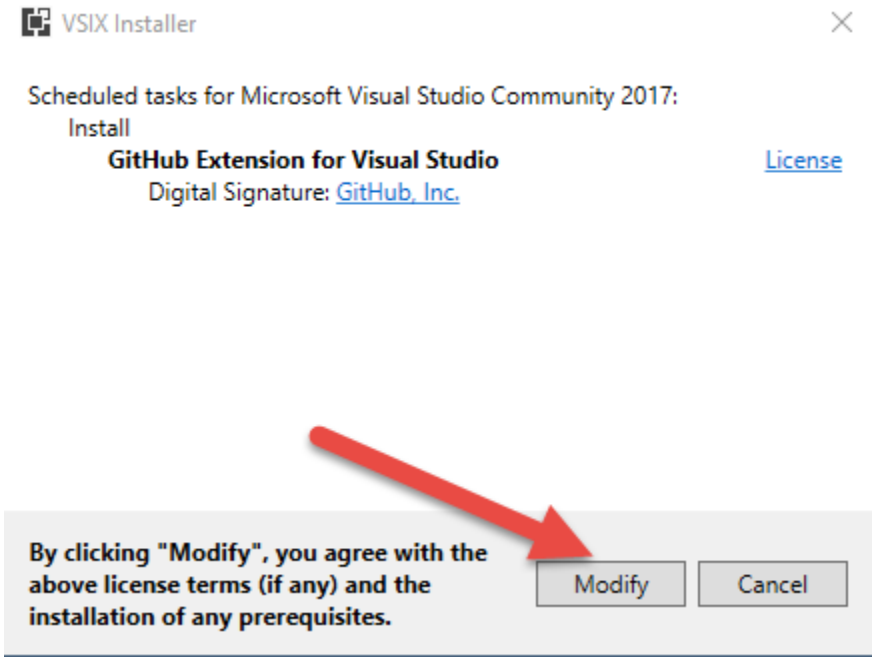
Add the GitHub Extension to VS

- You must close your VS and the installation should begin



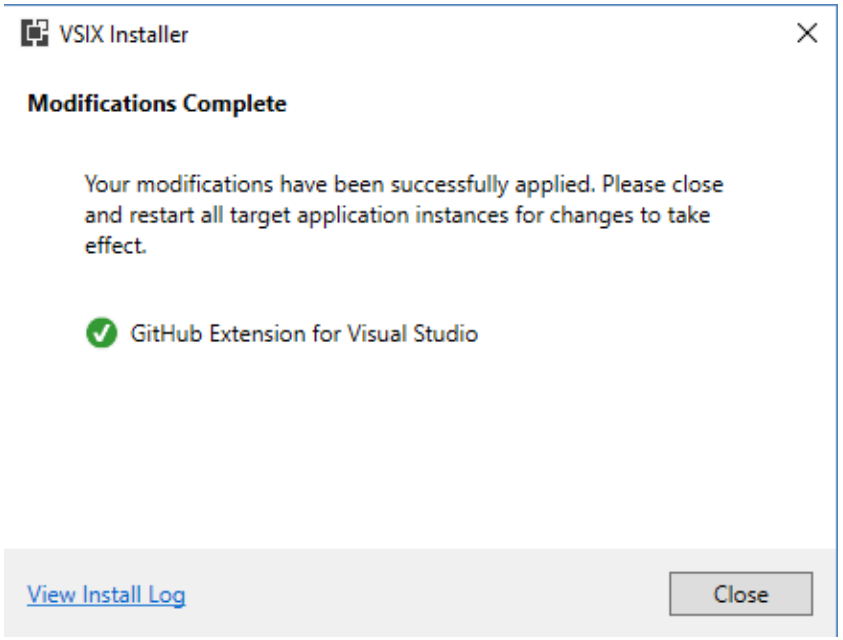
Add the GitHub Extension to VS

- Select ‘Modify’ so that the installation can update your VS



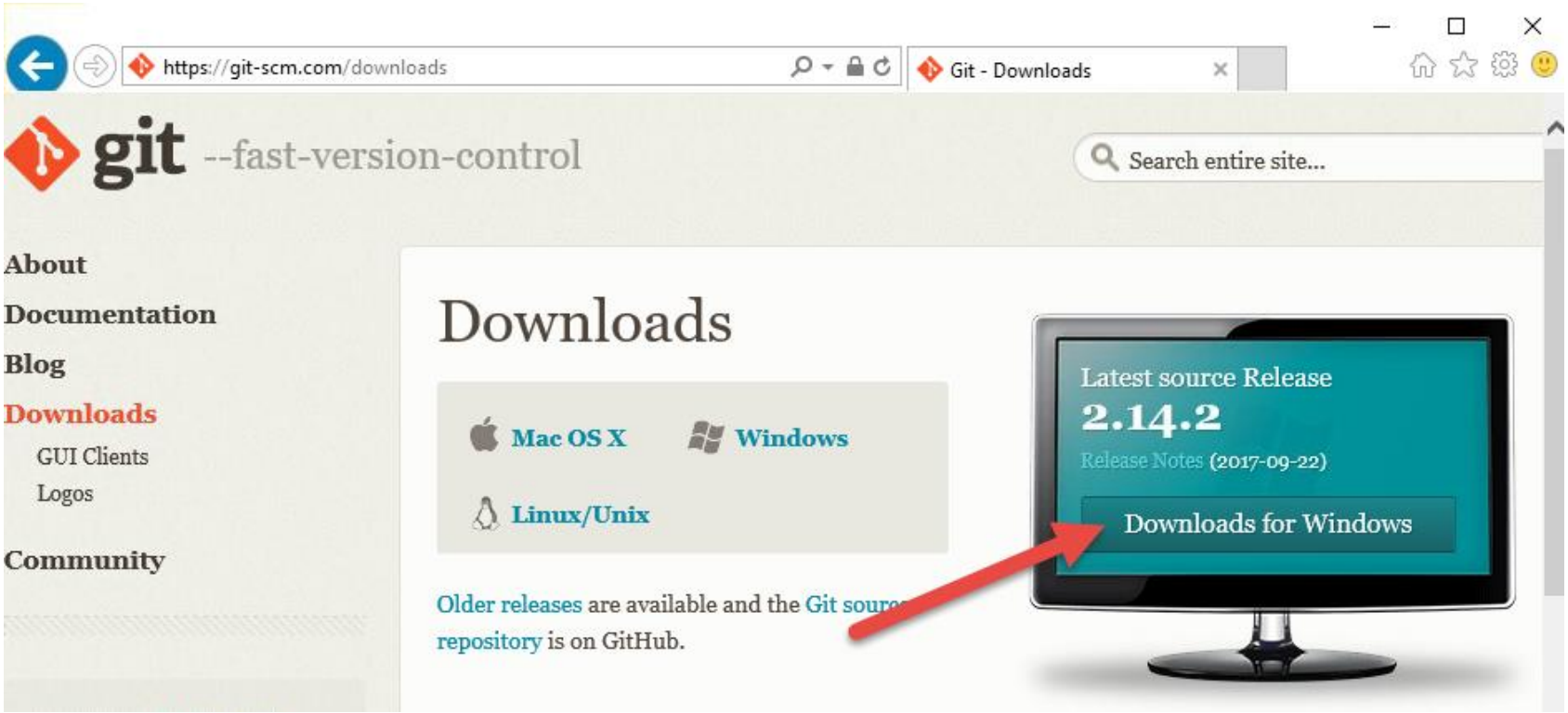
Add the GitHub Extension to VS

- Your installation will be complete once you receive a message like the following. Close this and open VS again.



Download Git

- https://git-scm.com/downloads
- Select ‘Downloads for Windows’ and scroll trough keeping the default settings



Download Git

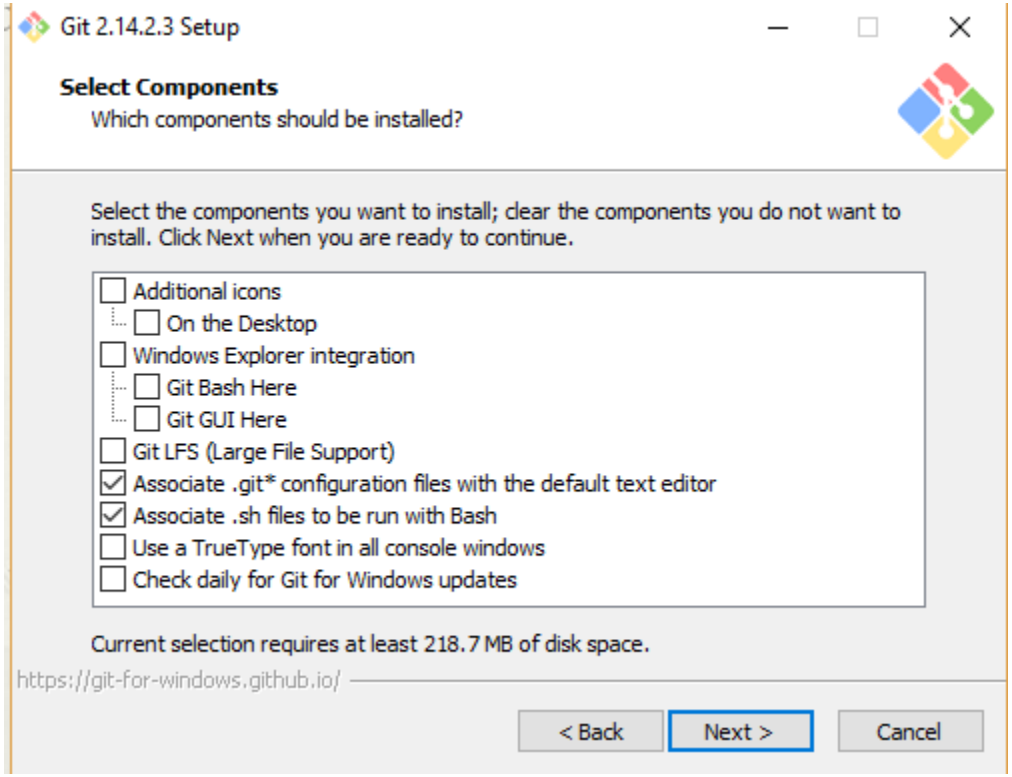
- Select ‘Run’ to start your download or select from the other downloads if you do not have a 64 bit machine

Download Git

- Keep defaults and select ‘Next’

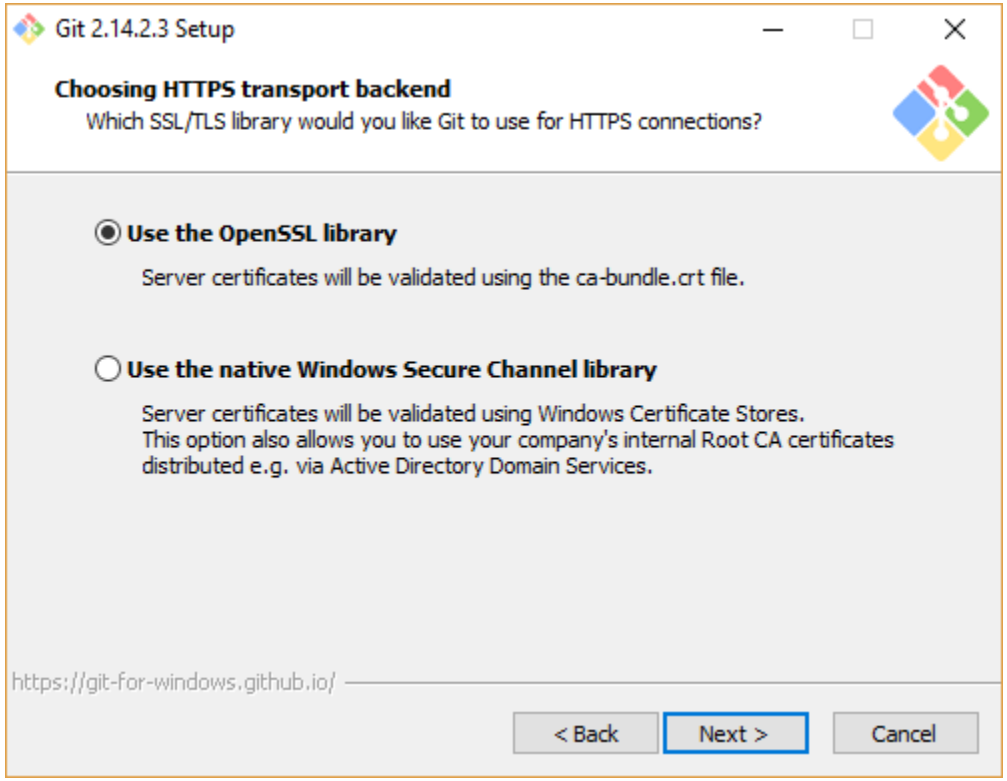
Download Git

- Keep defaults and select ‘Next’



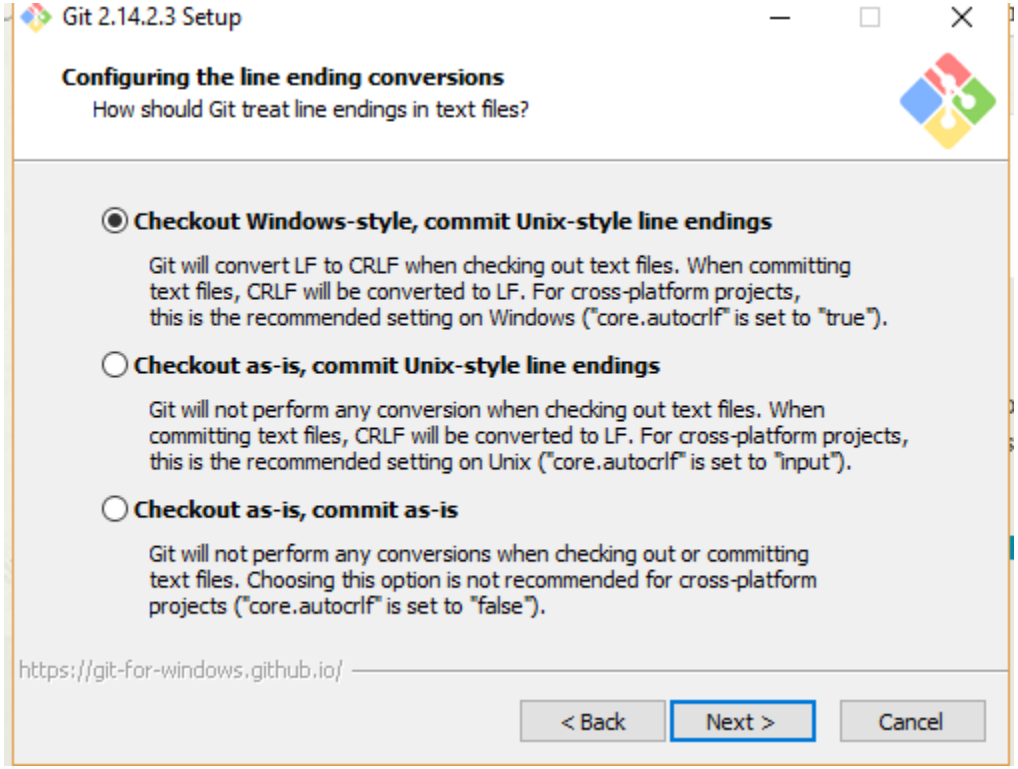
Download Git

- Keep defaults and select ‘Next’



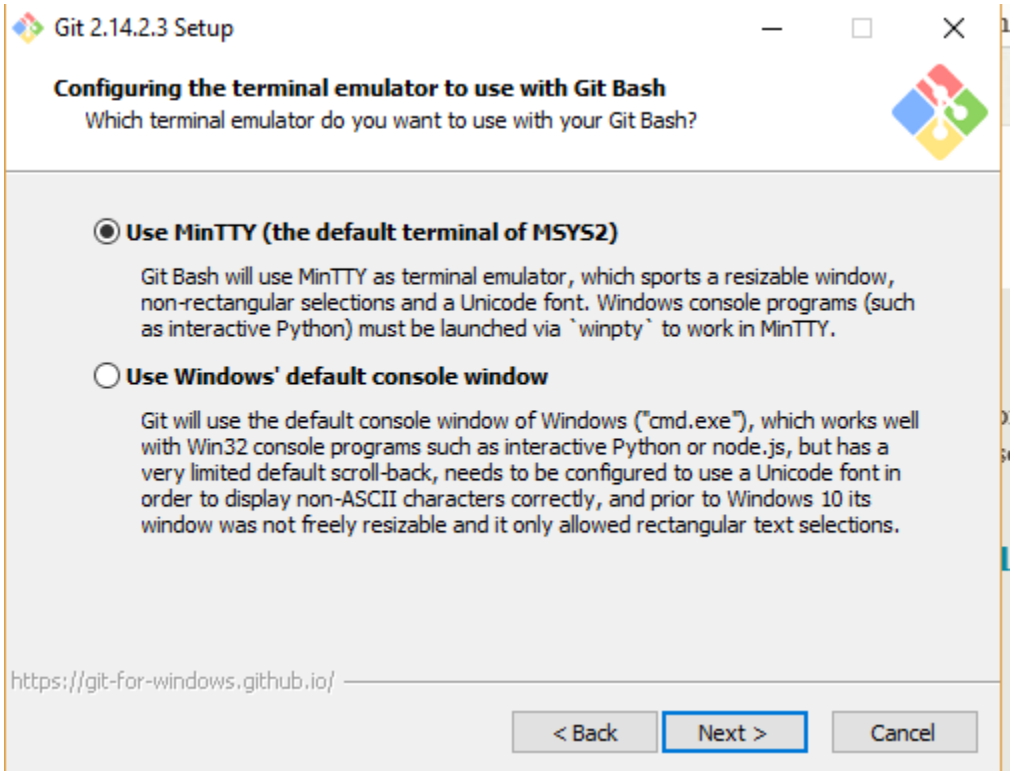
Download Git

- Keep defaults and select ‘Next’



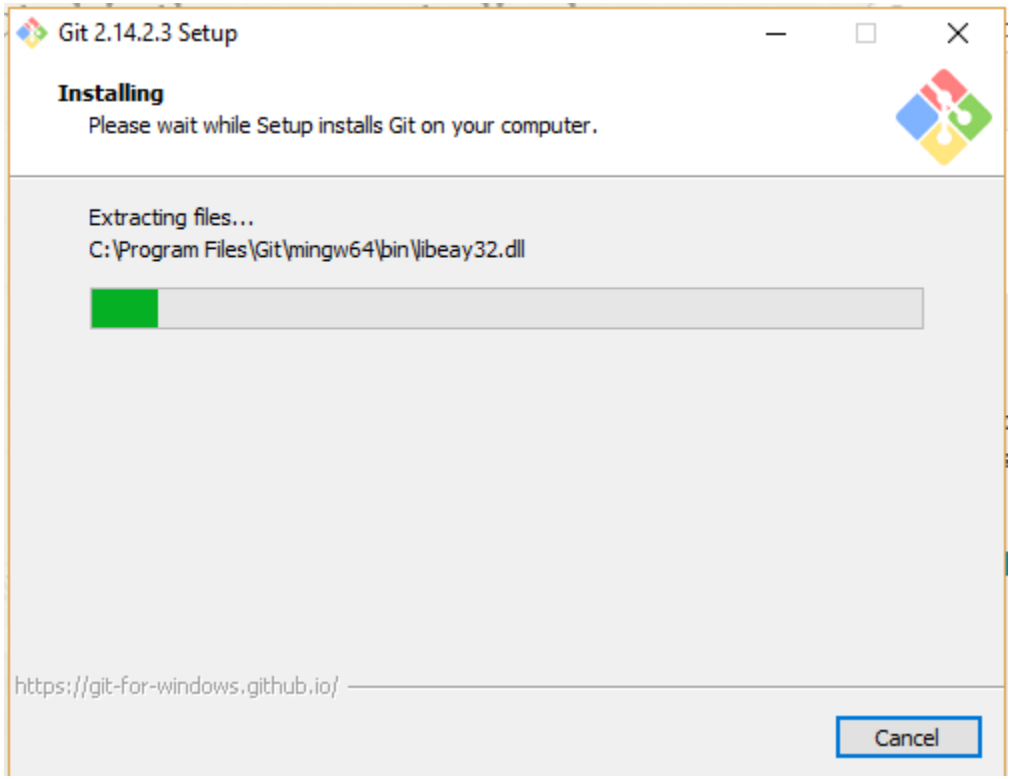
Download Git

- Keep defaults and select ‘Next’



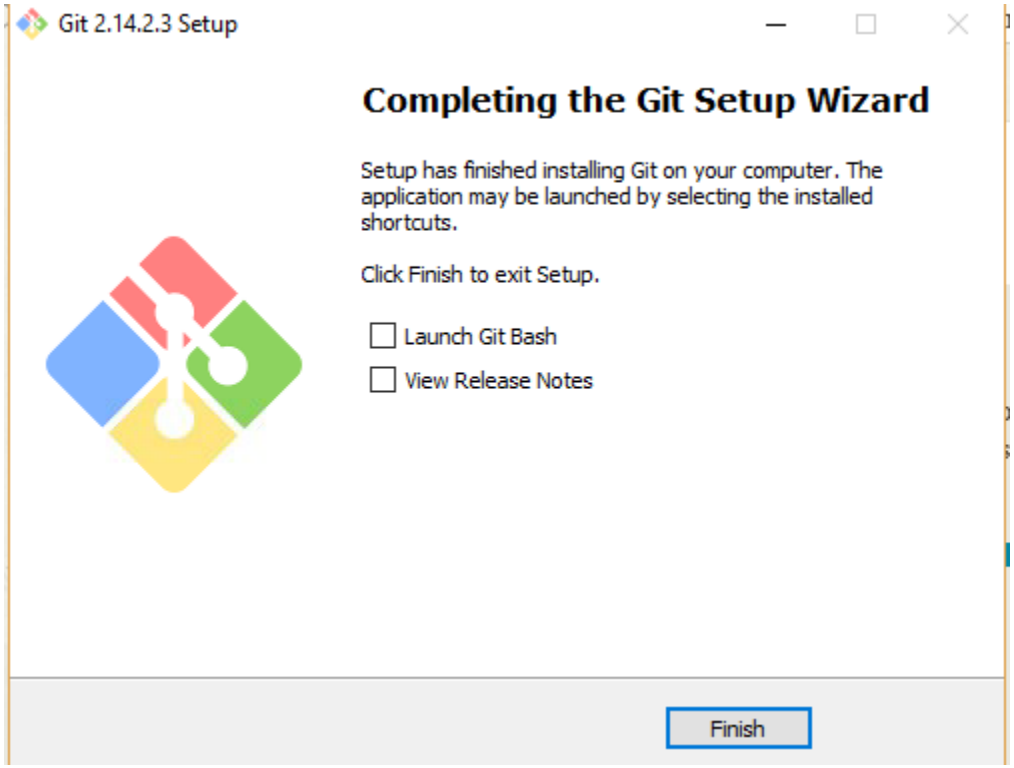
Download Git

- Your installation will begin…



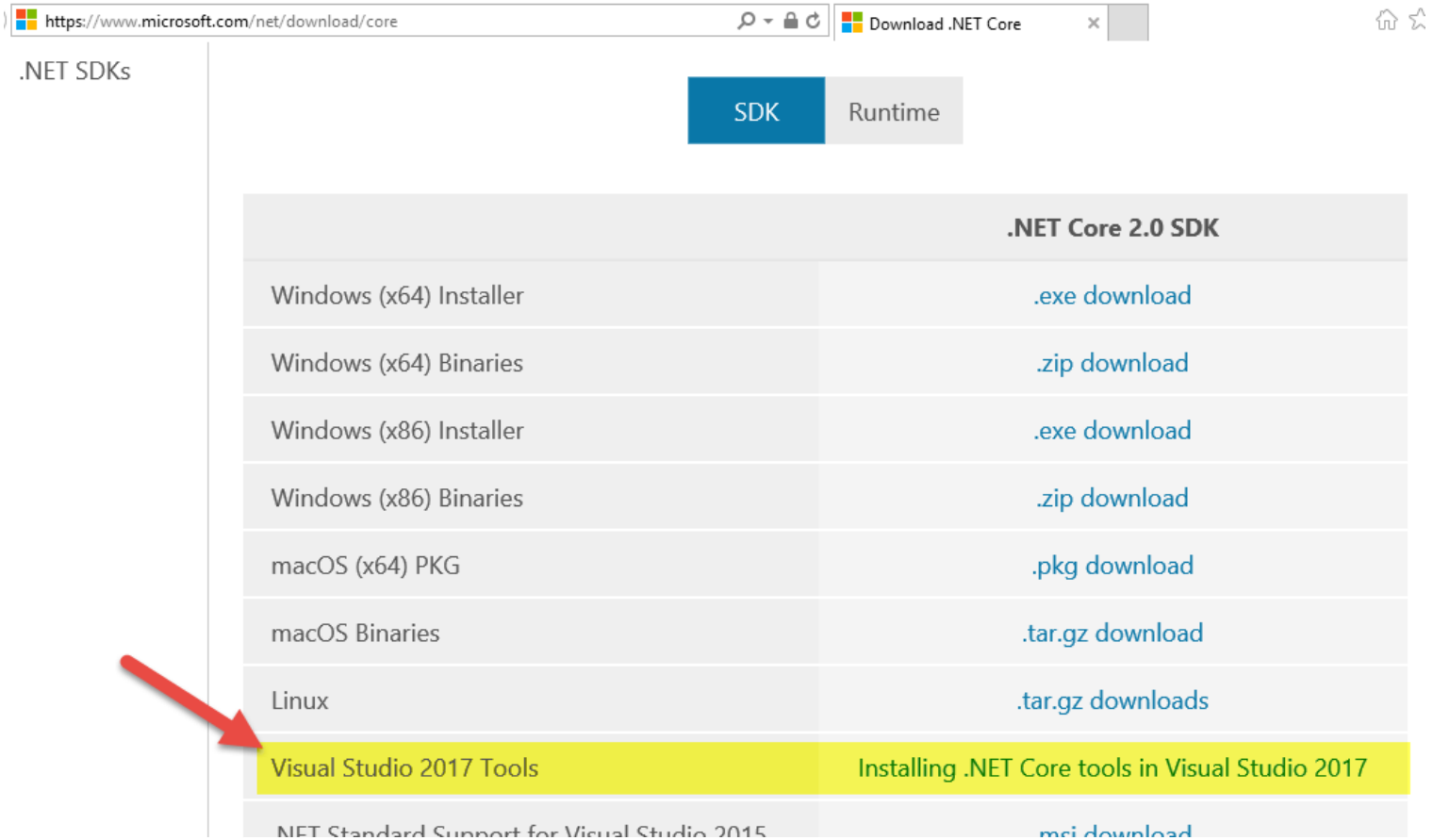
Download Git

- Once complete select ‘Finish’



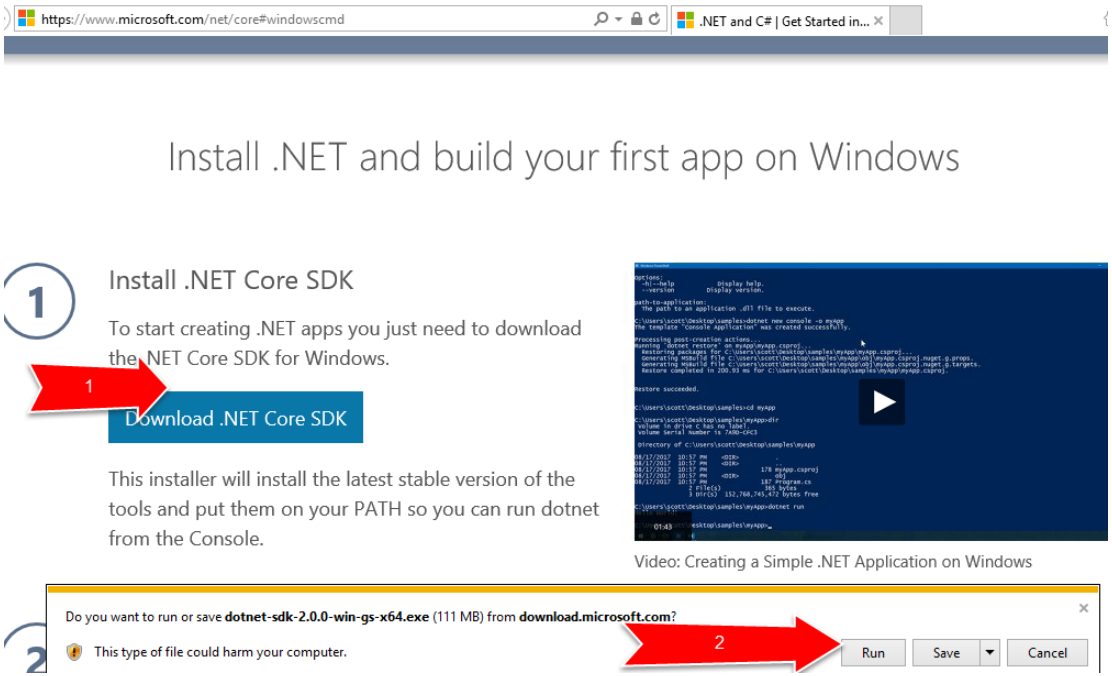
Download .NET Core SDK

- https://www.microsoft.com/net/download/core



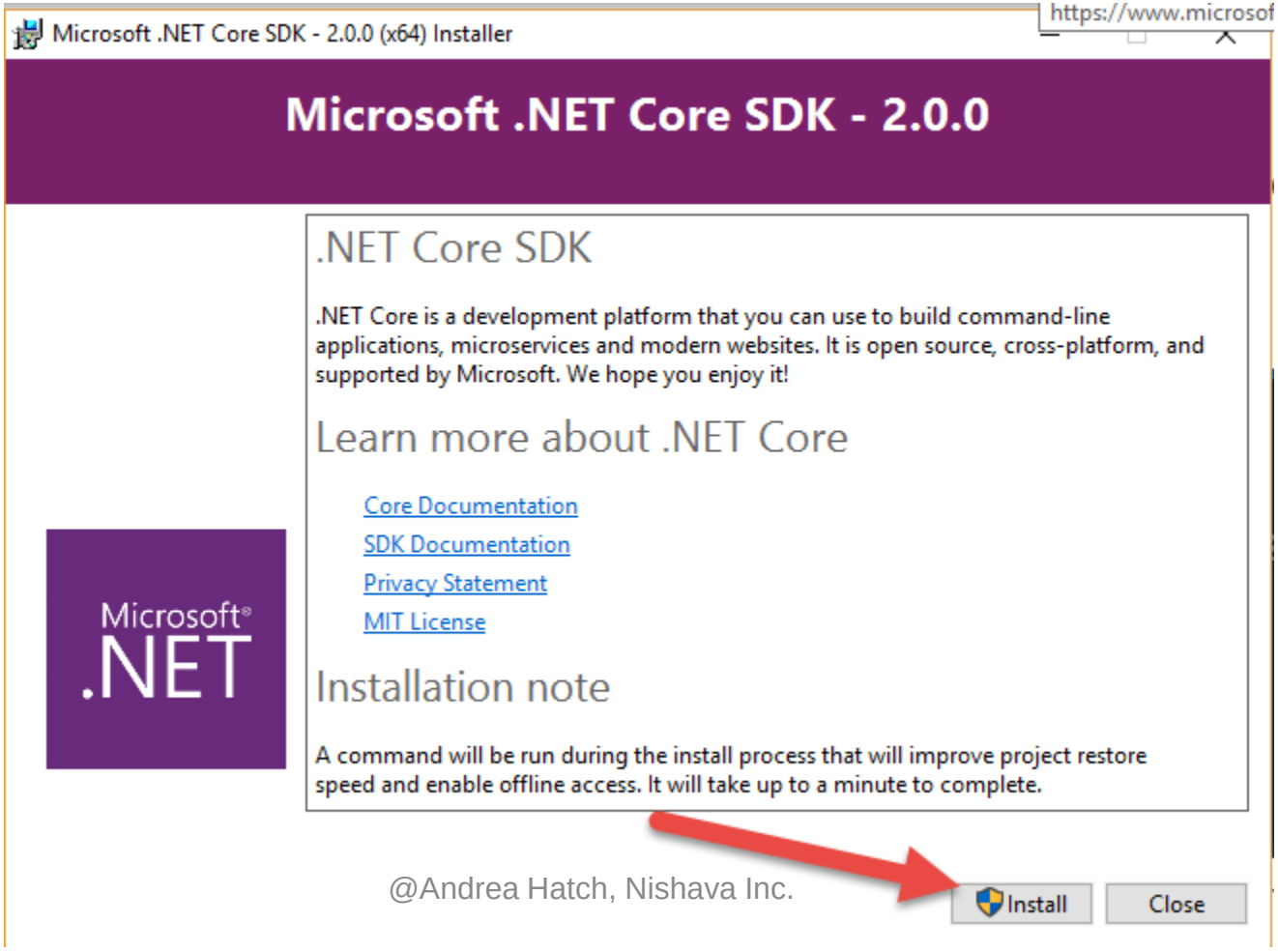
Download .NET Core SDK

- Select ‘Download .NET Core SDK’ and then select ‘Run’



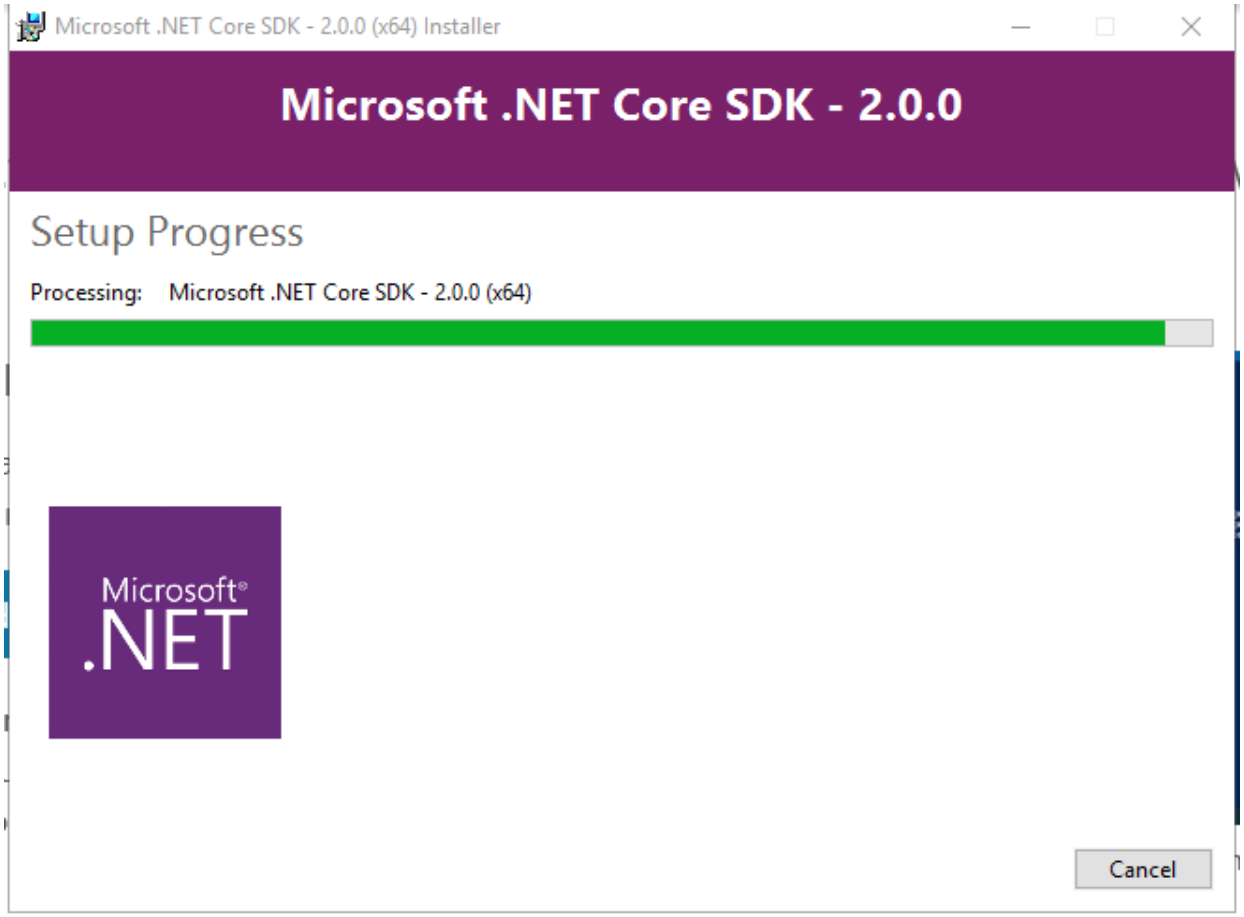
Download .NET Core SDK

- Select ‘Install’



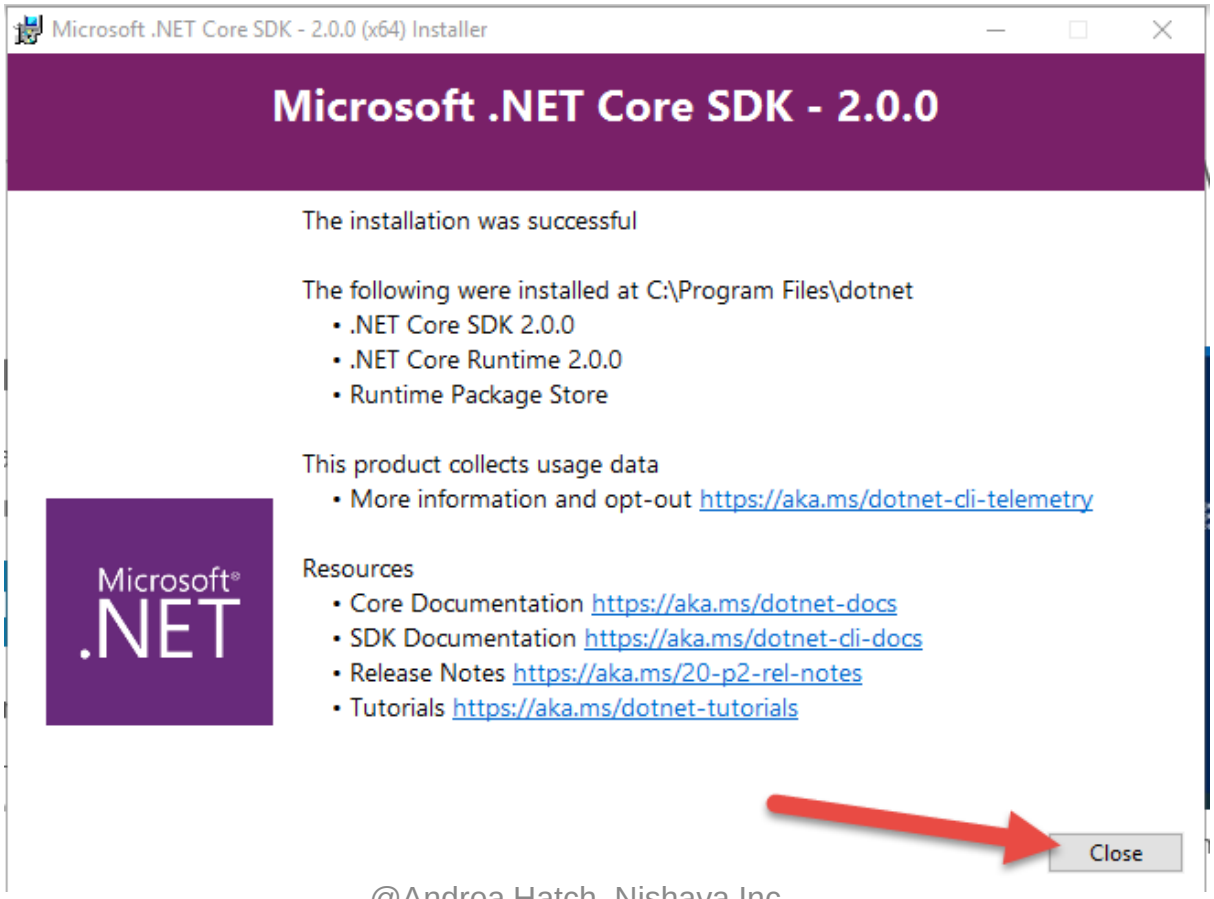
Download .NET Core SDK

- .NET Core should install



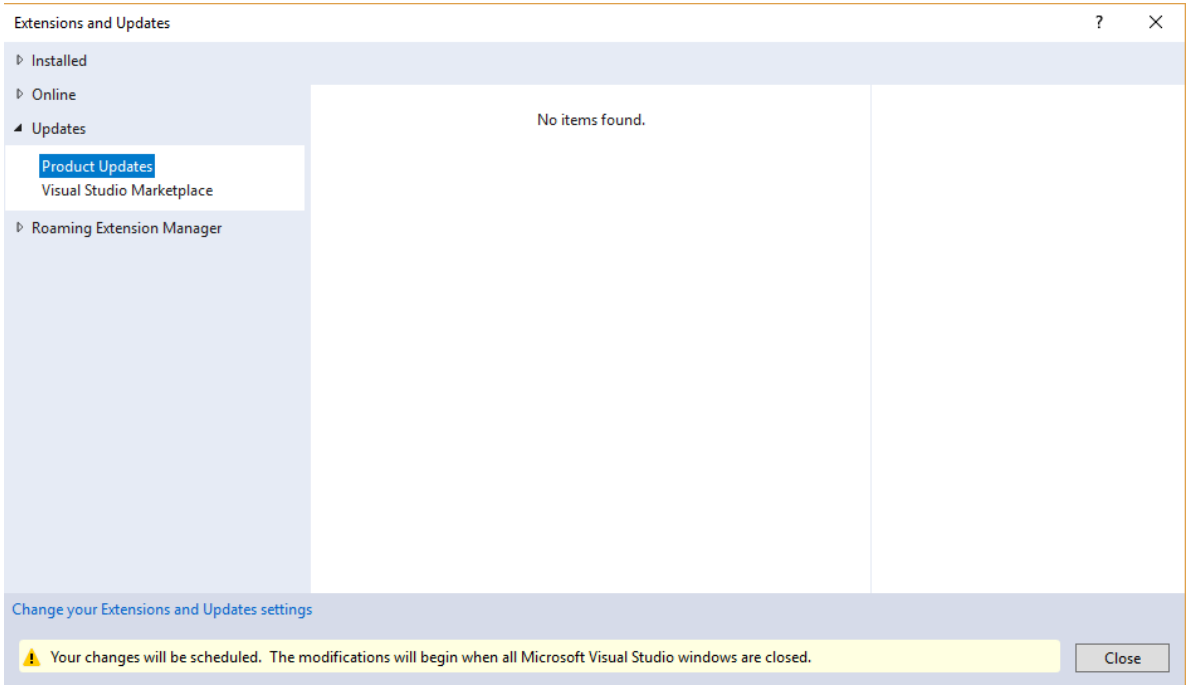
Download .NET Core SDK

- Once it is installed select ‘Close’



Check Updates

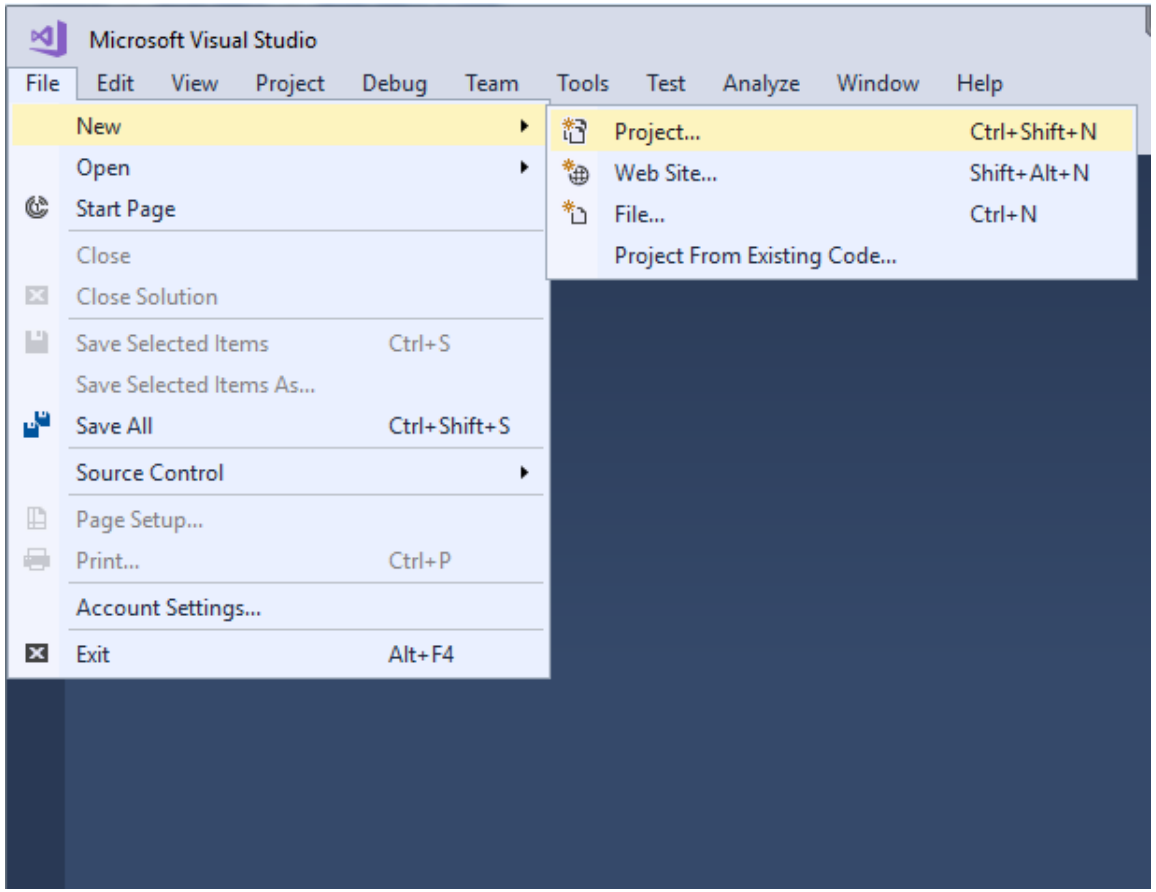
- In VS go to Tools -> Extensions and Updates
- Under Updates make sure that there are no product updates or VS Marketplace updates
- If there are, you must update these



3. Create a web app in VS to deploy to Azure

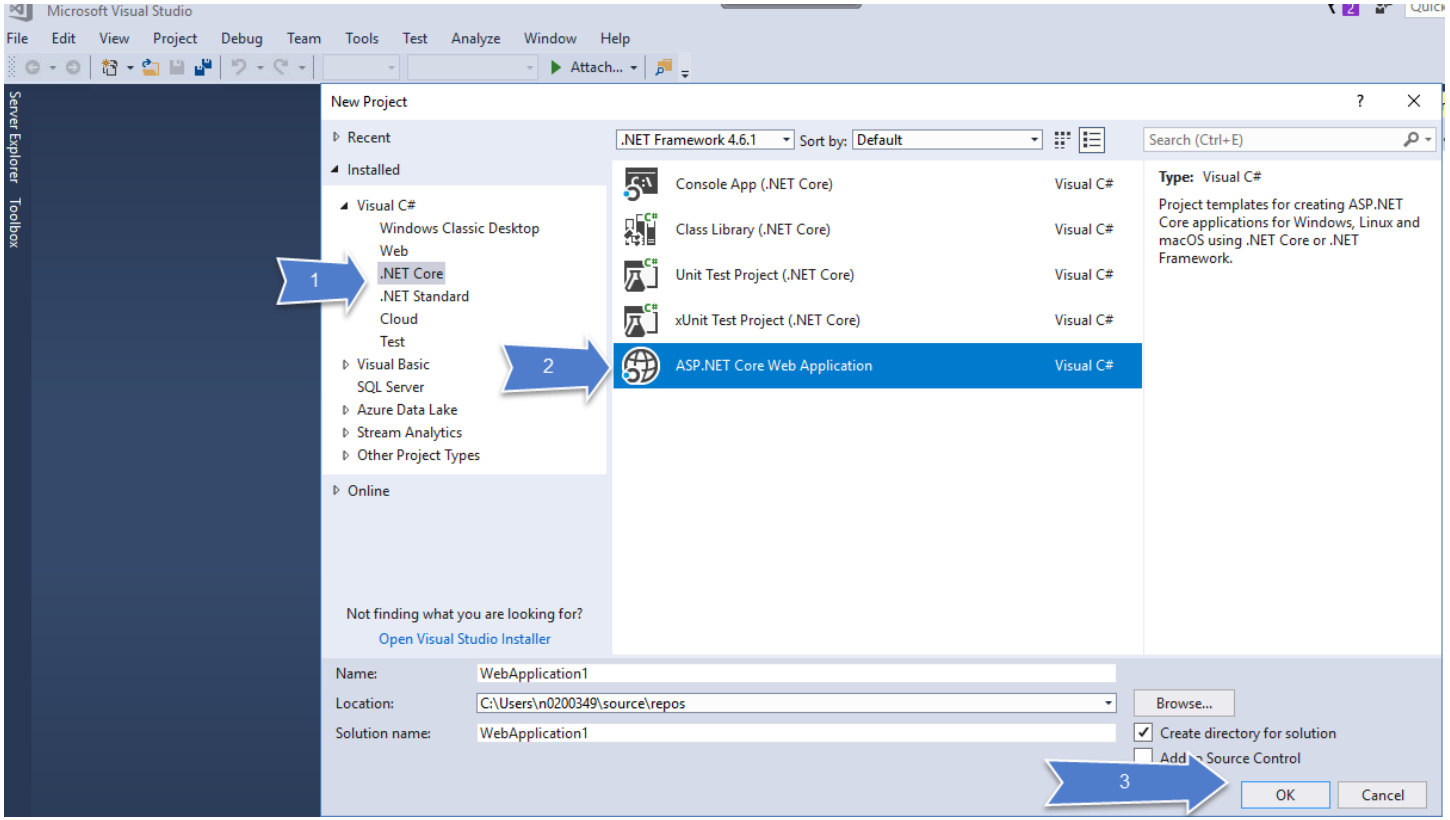
Creating a web app to deploy to Azure

- In VS, select File -> New -> Project



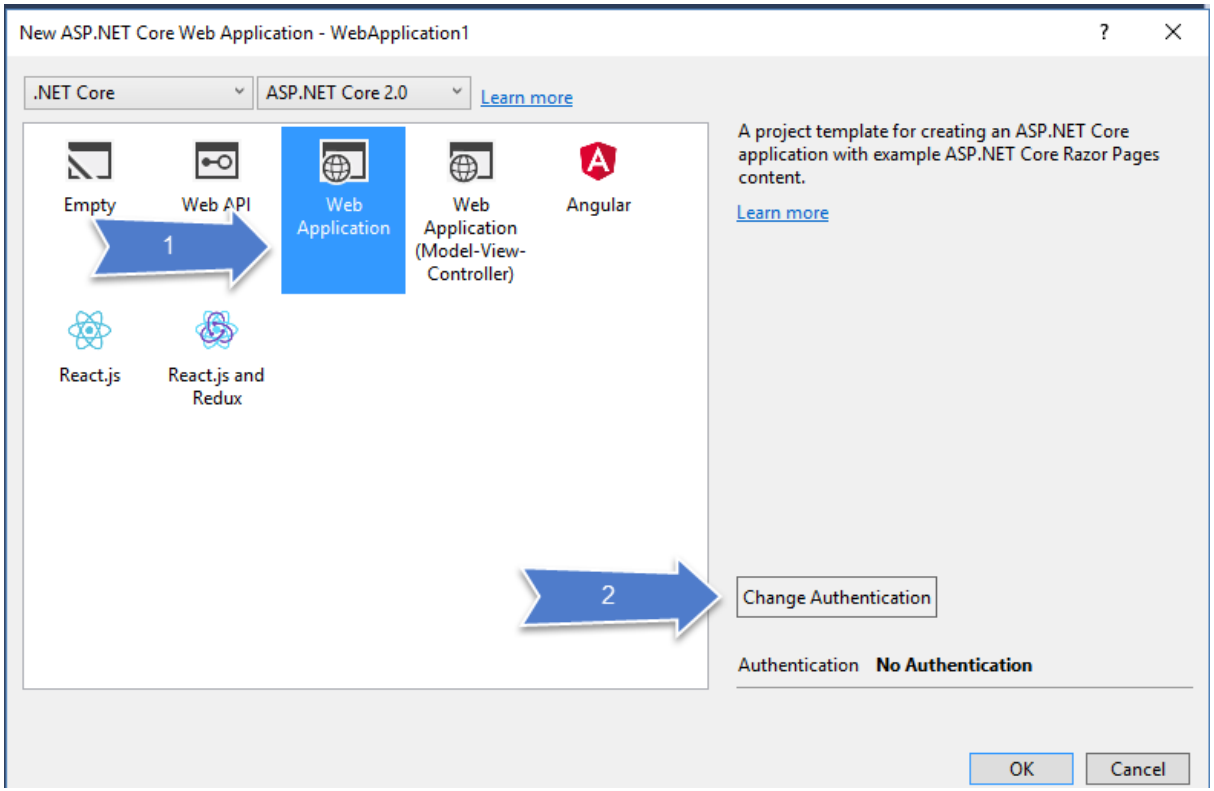
Creating a web app to deploy to Azure

- Select '.NET Core' under Visual C# and then 'ASP.NET Core Web Application' and then select 'OK'



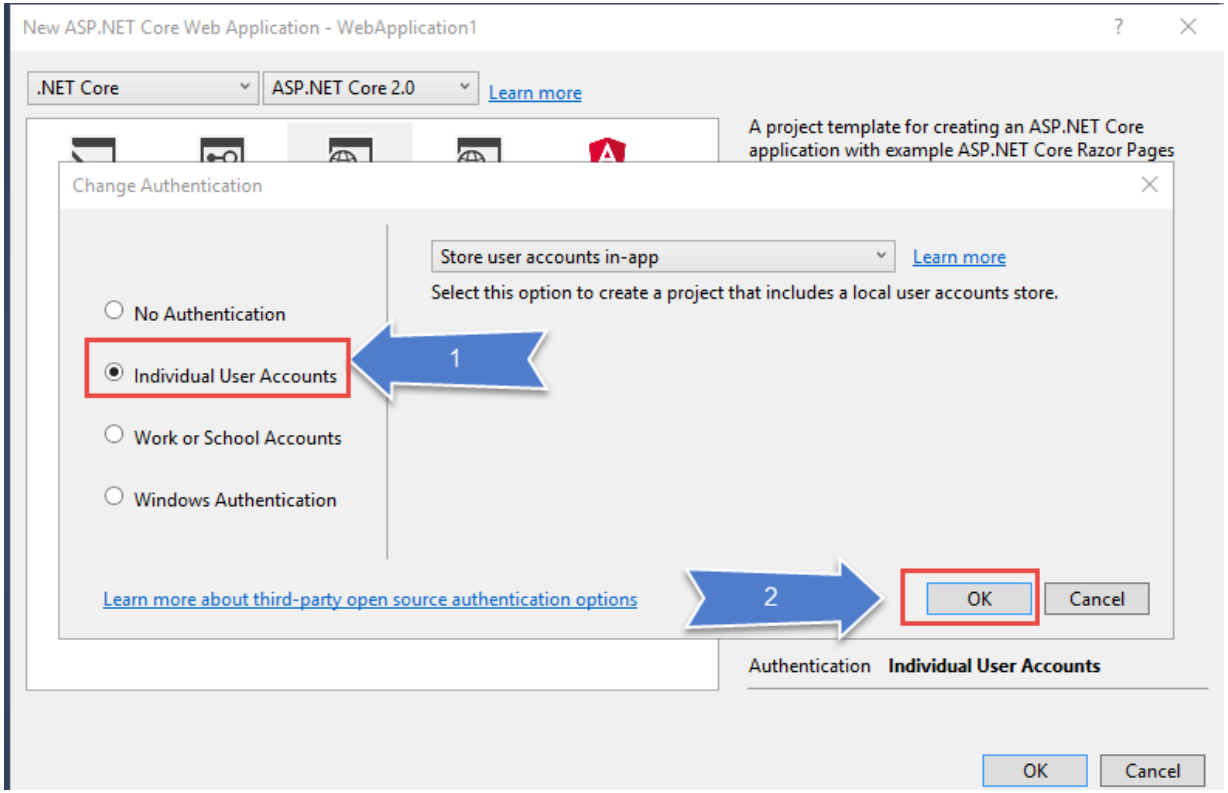
Creating a web app to deploy to Azure

- Select ‘Web Application’ and then ‘Change authentication’ on the right



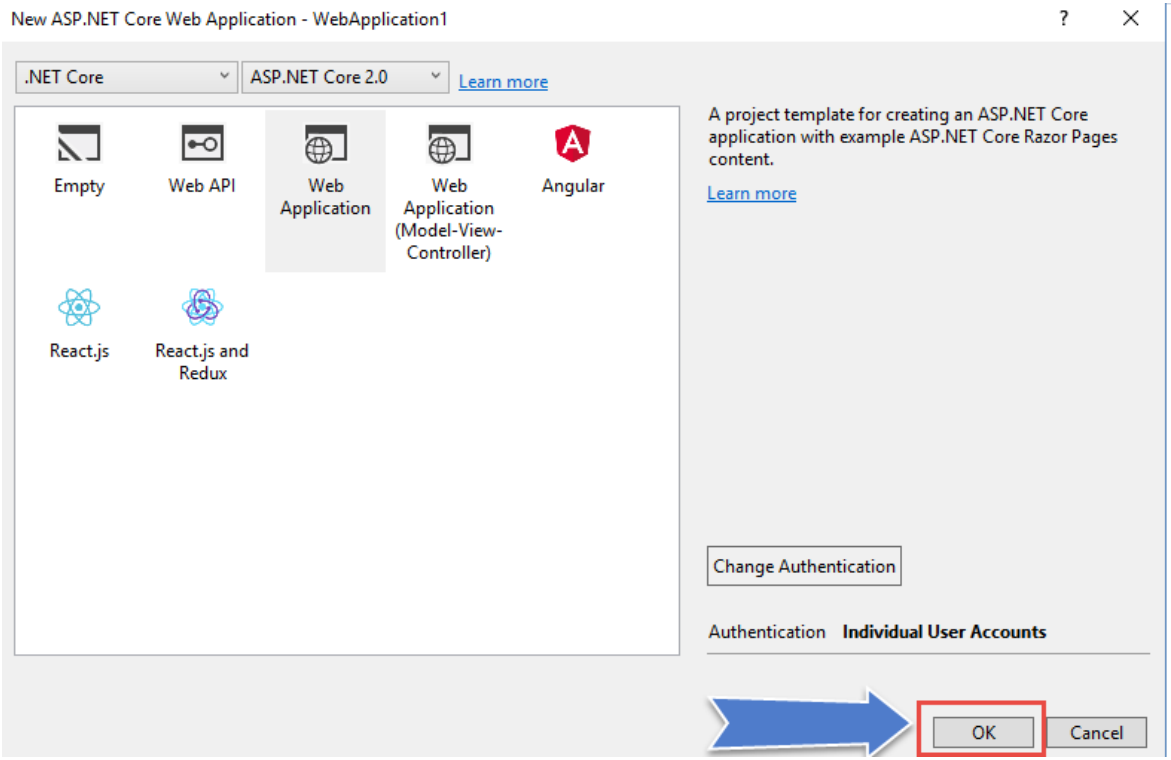
Creating a web app to deploy to Azure

- Change the authentication to Individual User Accounts and then select ‘OK’



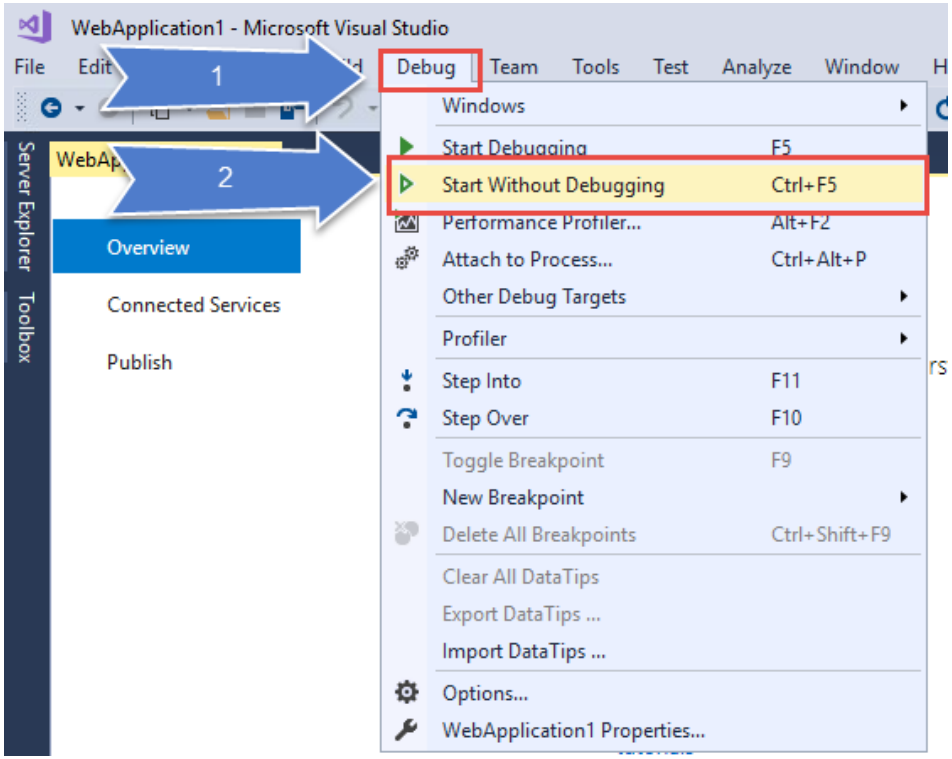
Creating a web app to deploy to Azure

- Select ‘OK’ on the Web Application screen and then your project will be created



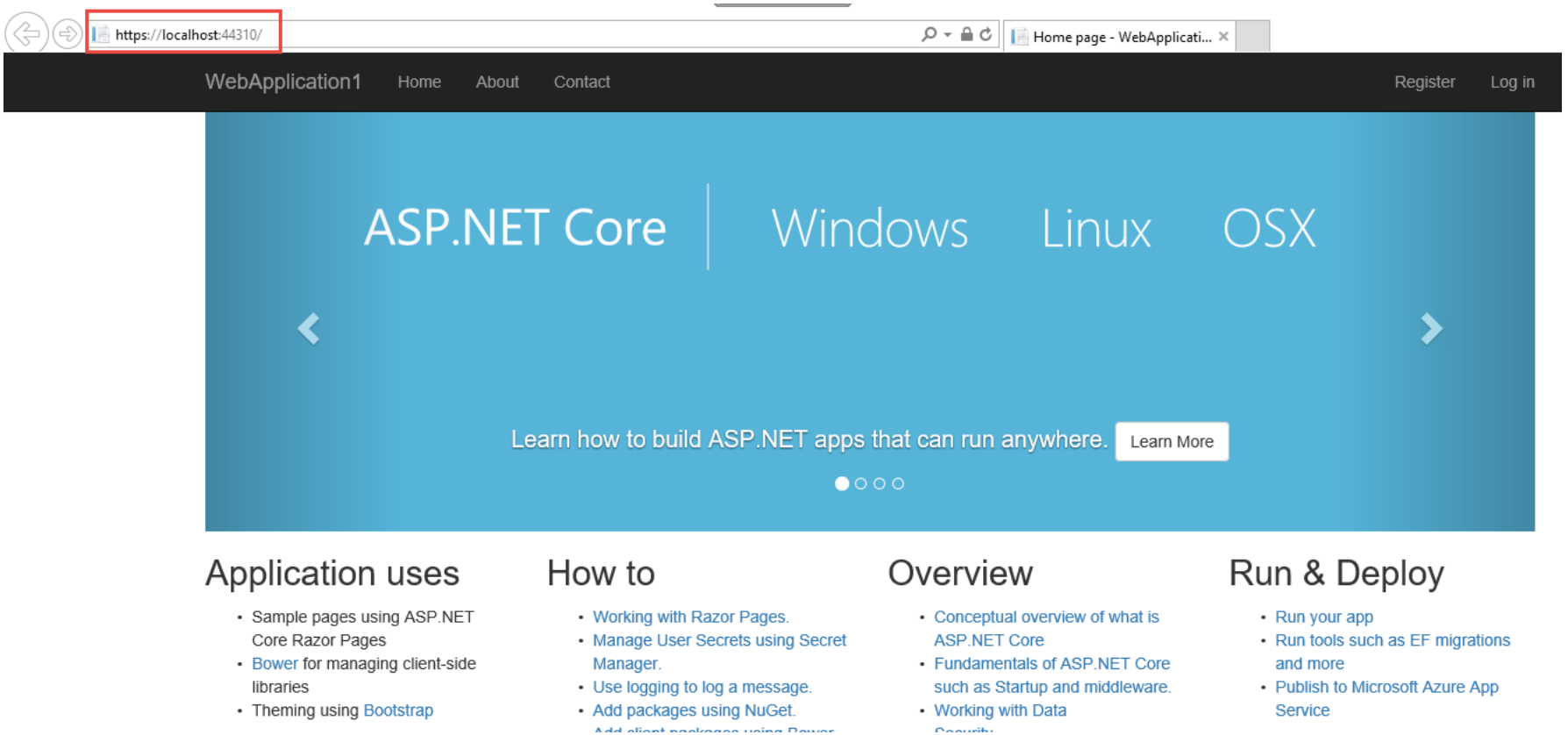
Running the app locally

- Select Debug -> Start without Debugging



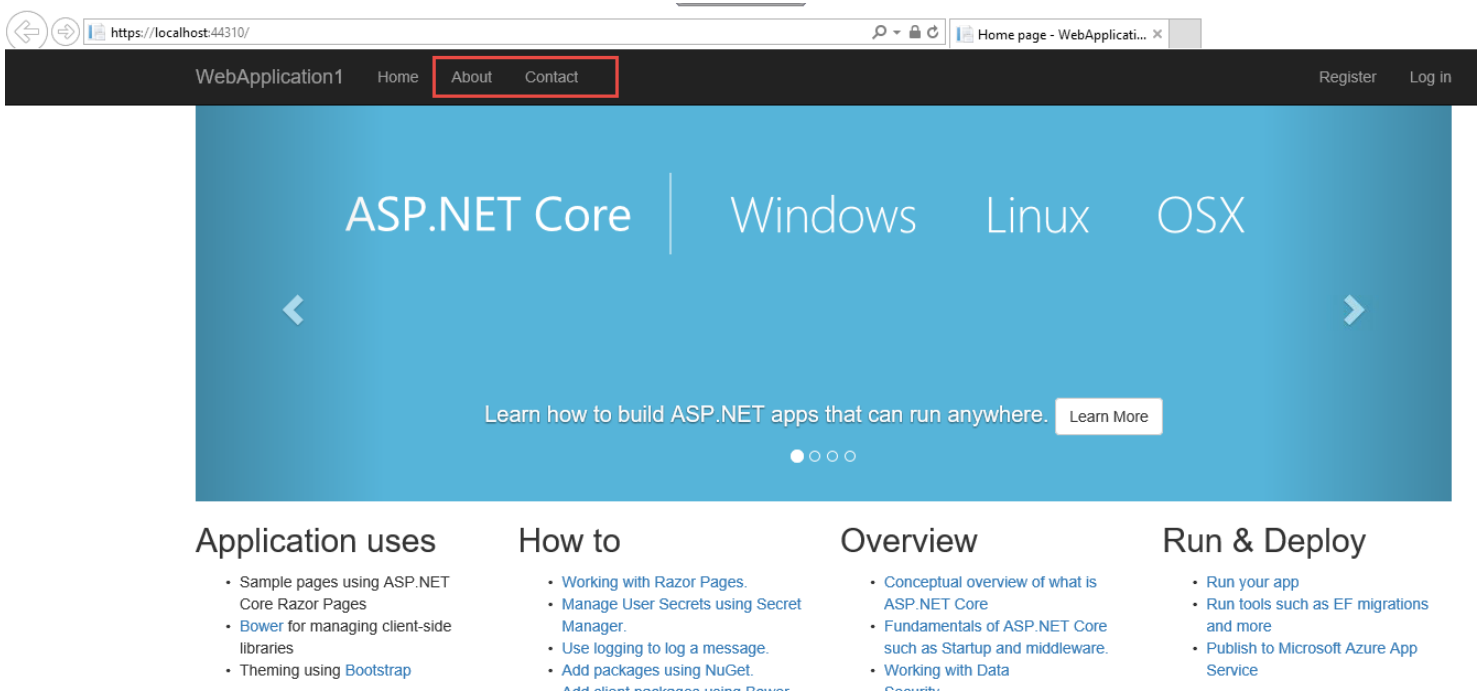
Running the app locally

- You should then see your app running on your local host



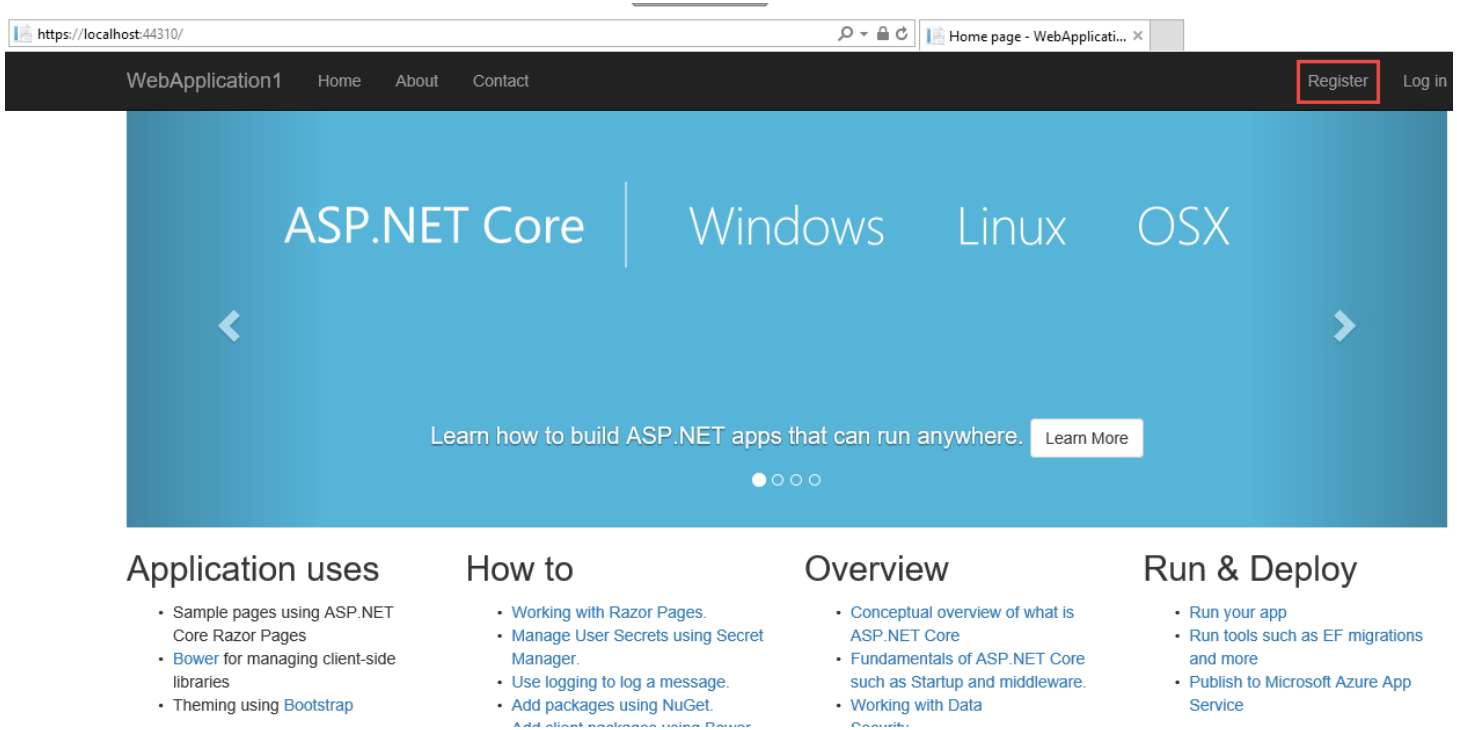
Running the app locally

- To test that the app is working select ‘About’ or ‘Contact’ on the top of your app screen



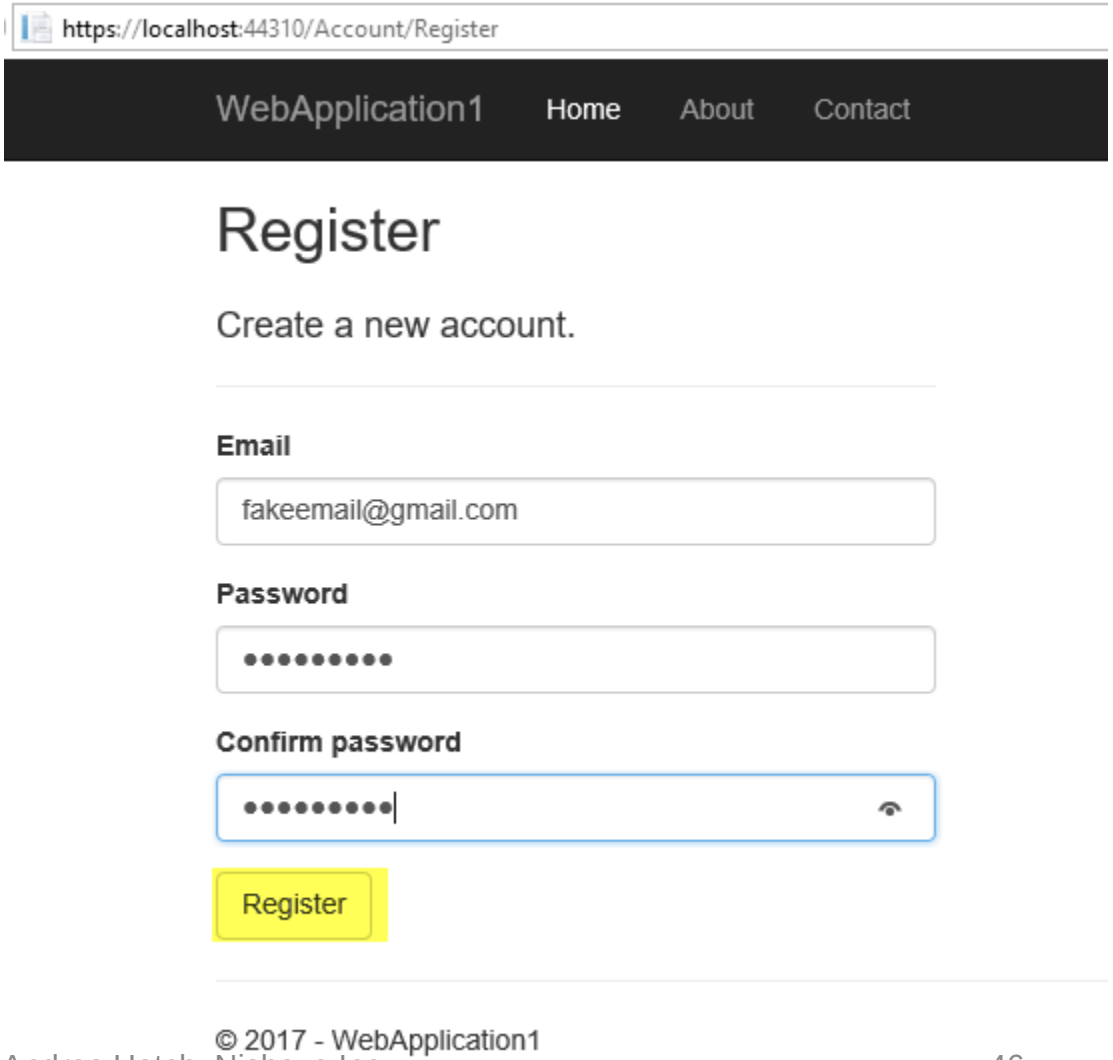
Running the app locally

- You can register your app by selecting Register on the top of the application



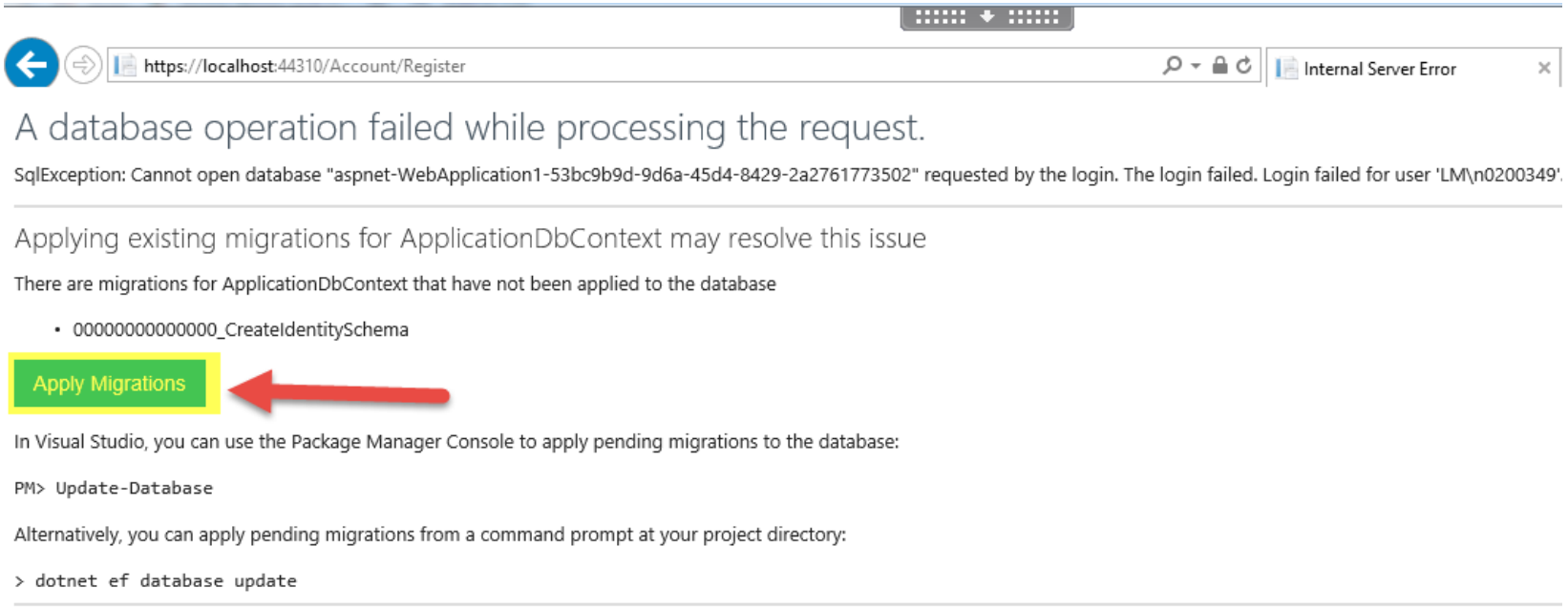
Running the app locally

- Add any fake email and password then select Register



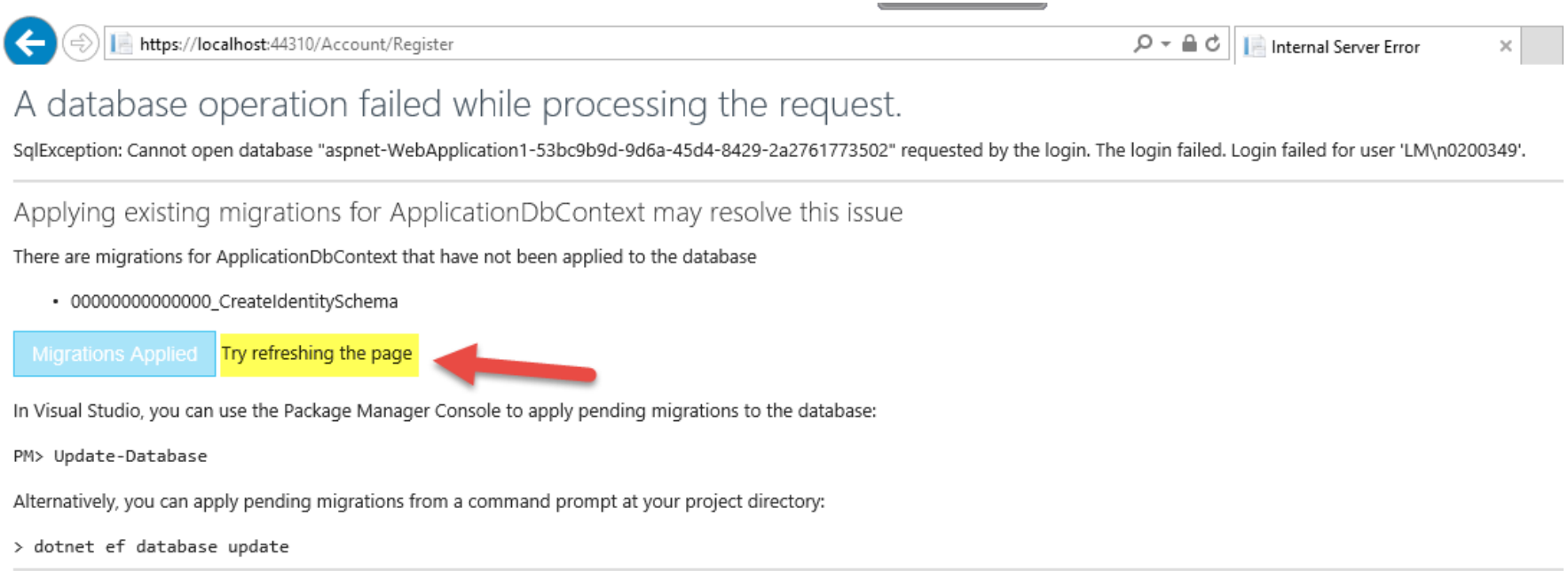
Running the app locally

- Your page will display an error until you select Apply Migrations



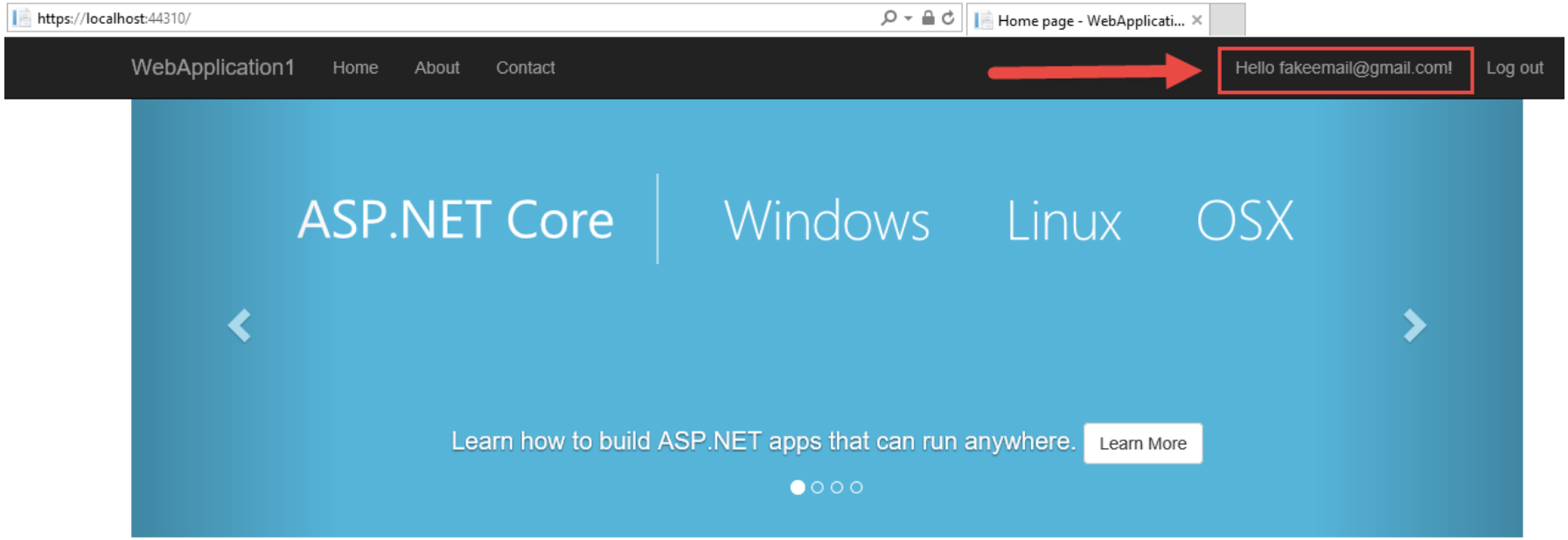
Running the app locally

- Once the migrations are applied you will be asked to refresh your screen



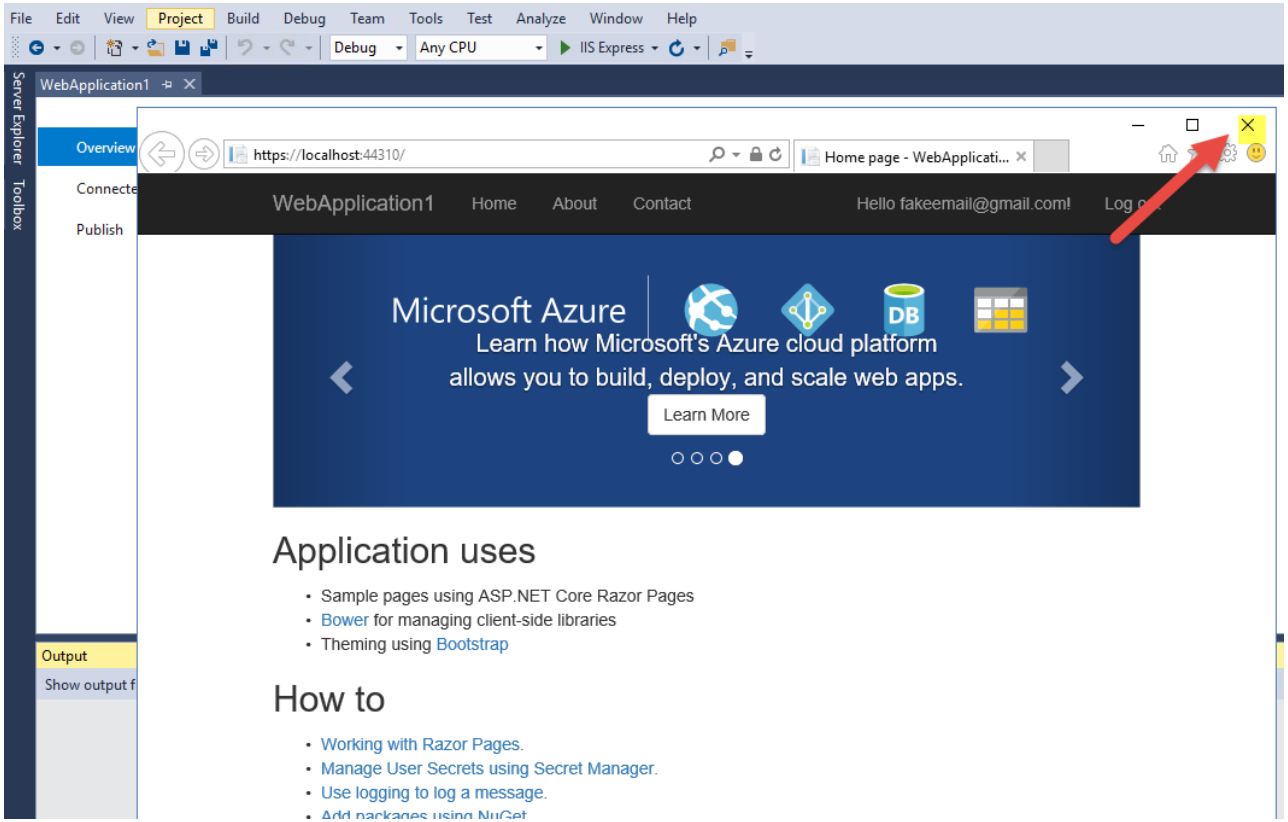
Running the app locally

- Your screen will now display the homepage again but this time with your email account



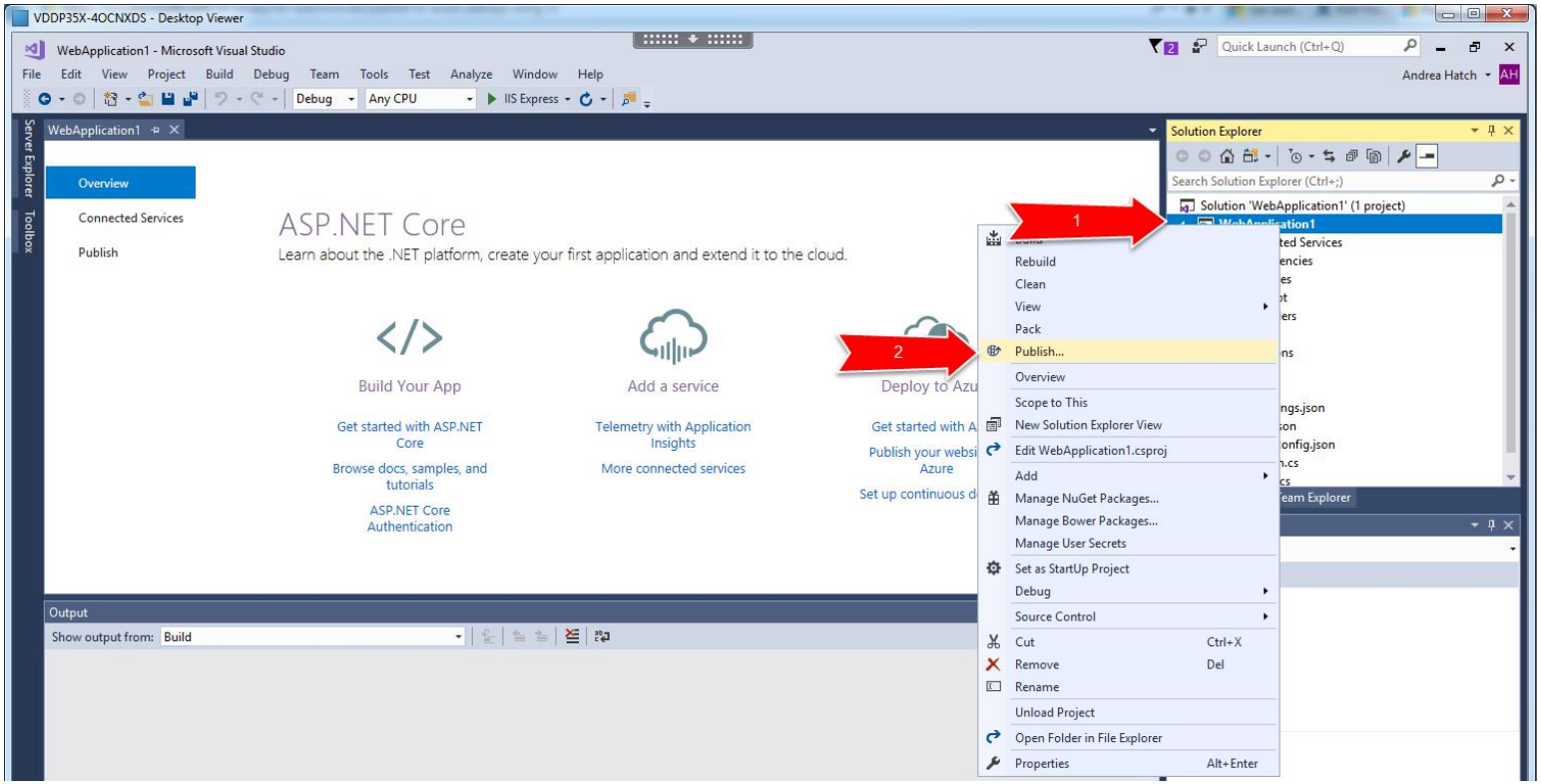
Running the app locally

- Click out of viewing your application by selecting the x button on the browser



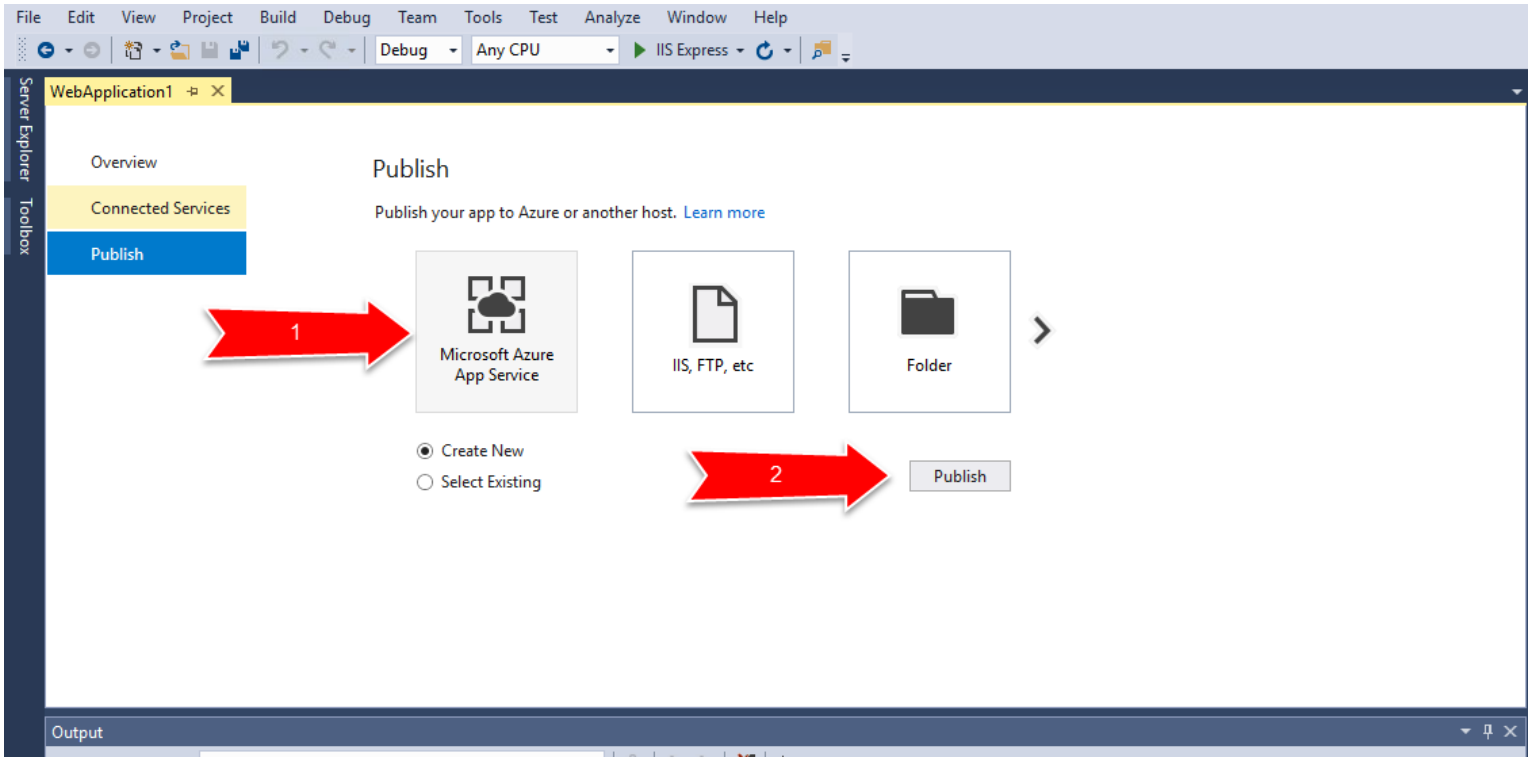
Deploy to Azure

- Next lets deploy your app to Azure by right clicking on your application name under the Solution Explorer and then selecting ‘Publish’



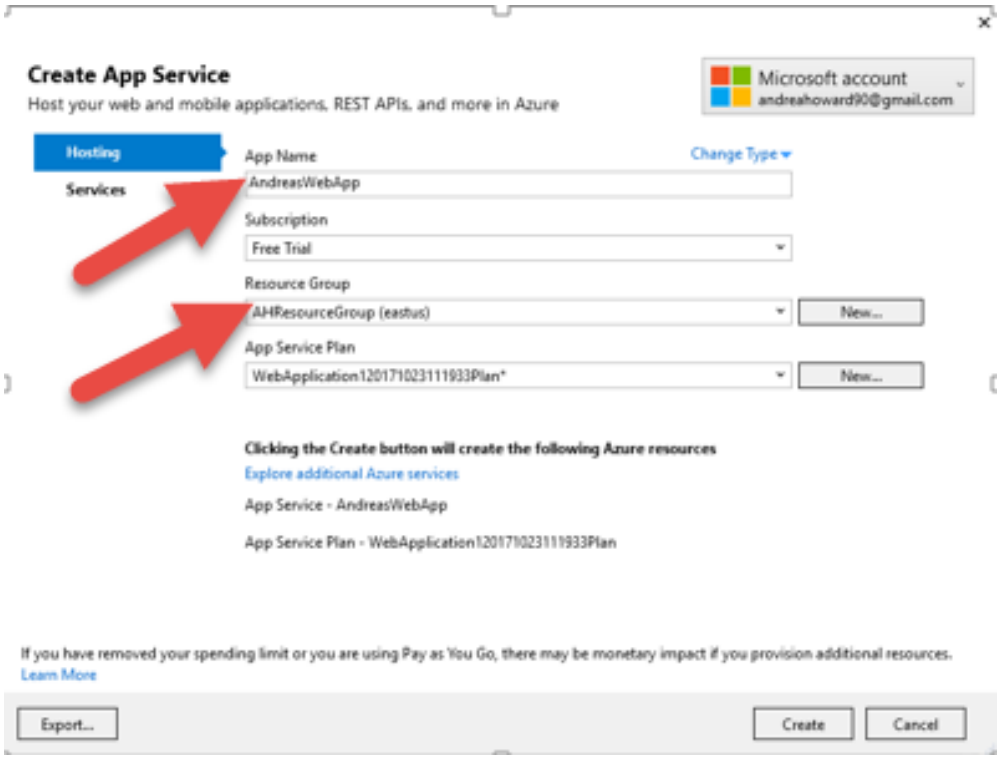
Deploy to Azure

- Select Microsoft Azure on the Publish screen and then select ‘Publish’



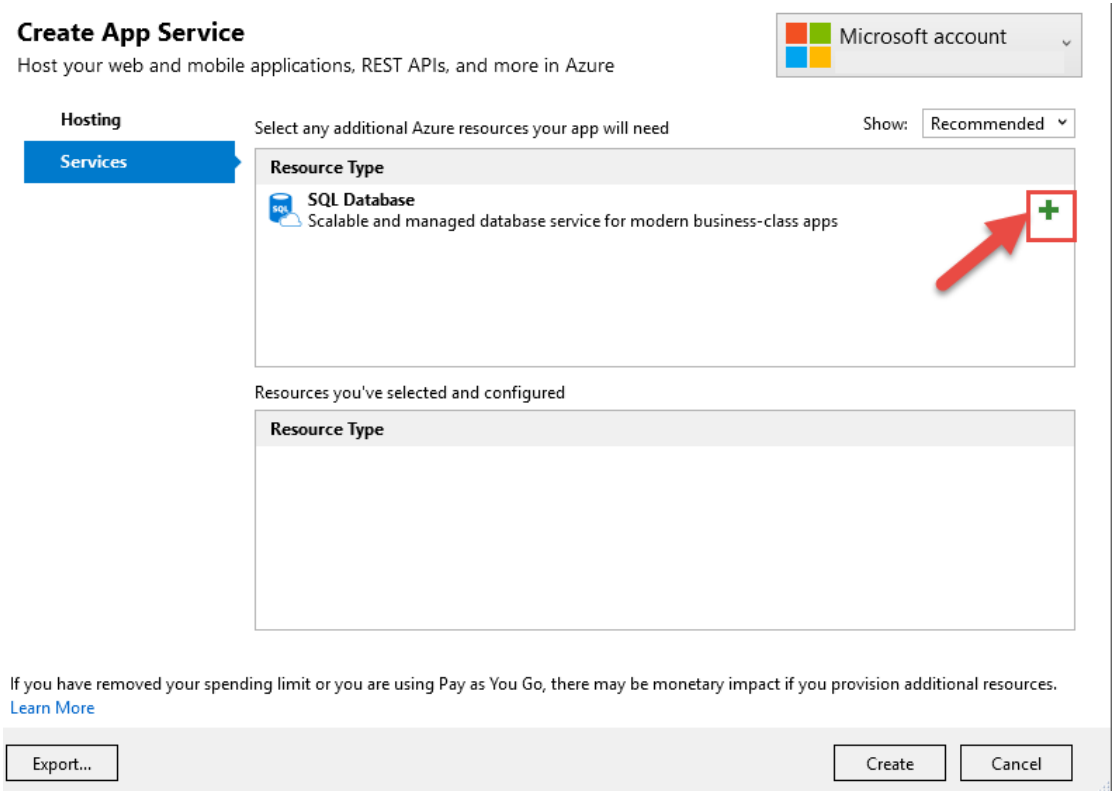
Deploy to Azure

- Add a unique name for your App, select the free trial subscription, create a new resource group and DO NOT select ‘Create’ just yet



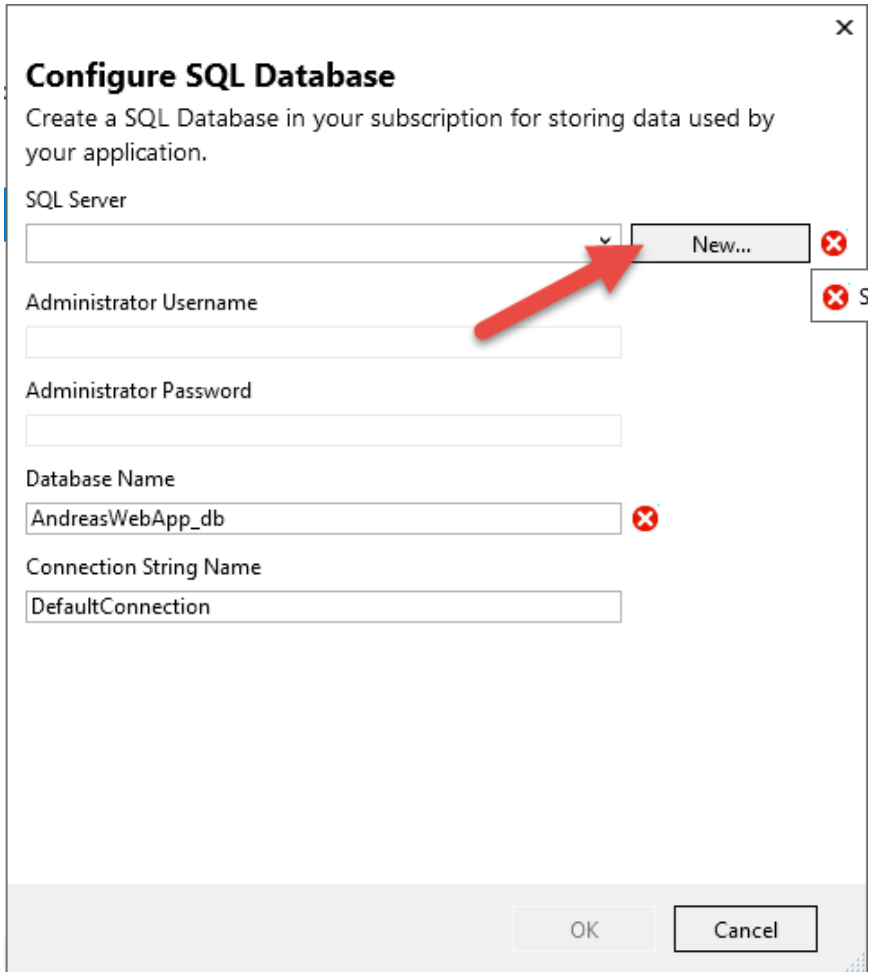
Deploy to Azure

- Select the ‘Services’ tab and then select the green plus sign to add a Database



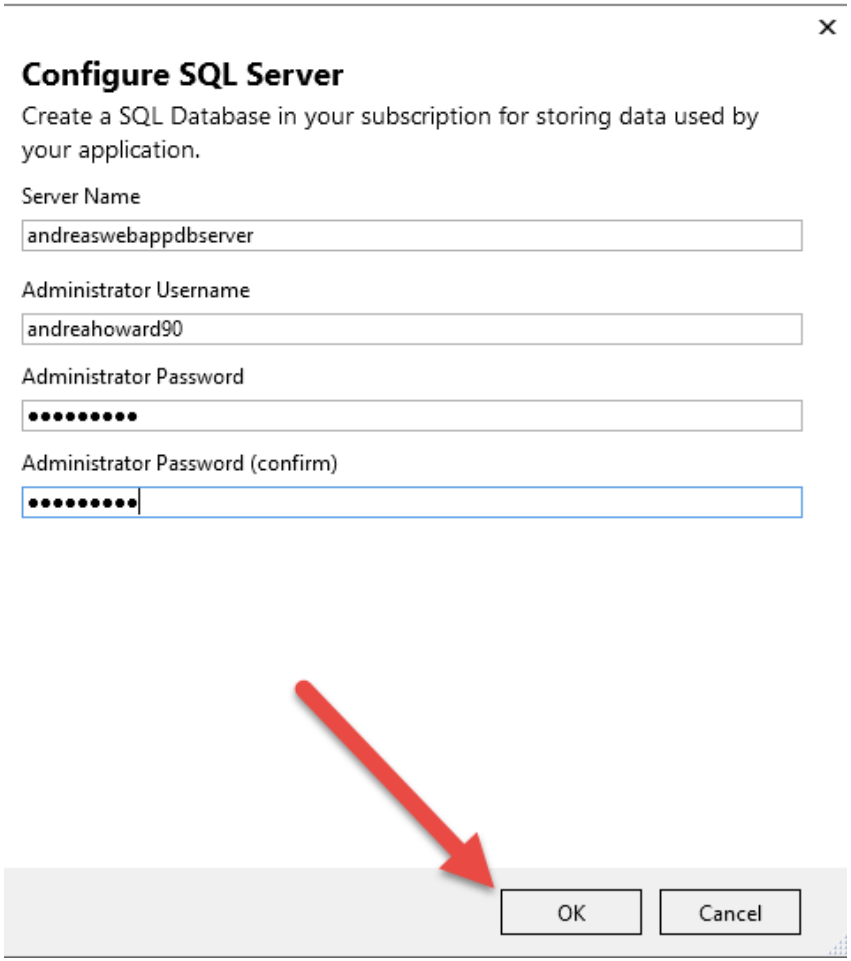
Deploy to Azure

- Select ‘New’ to create a new SQL Database



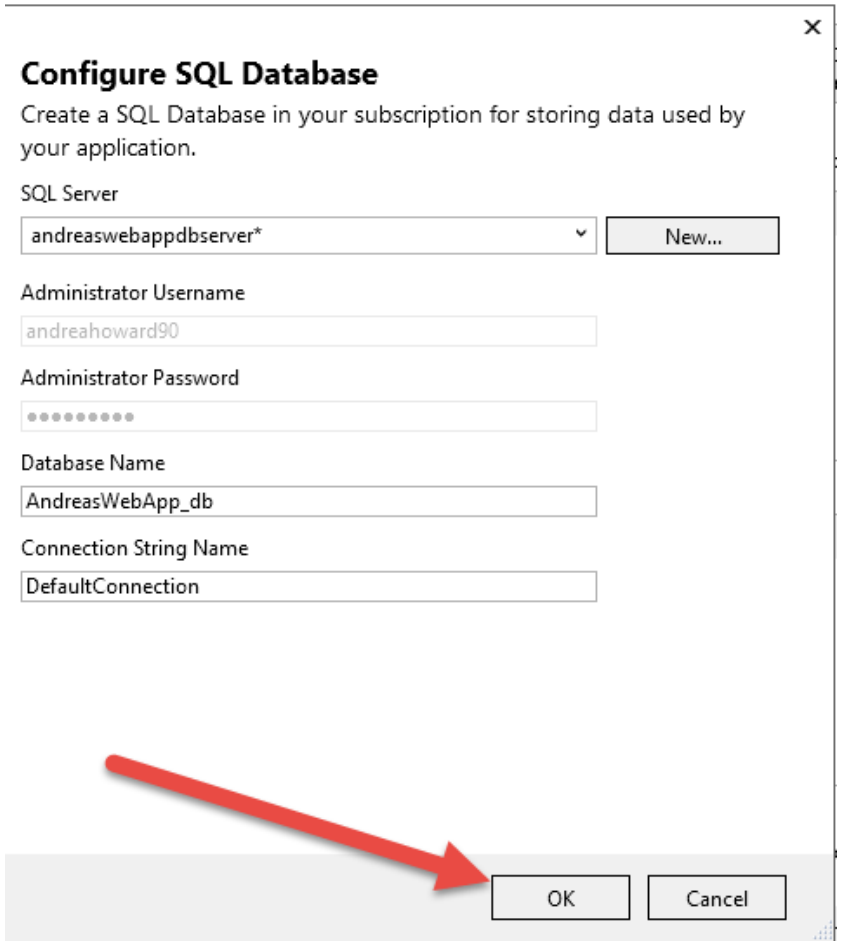
Deploy to Azure

- Create a Username and Password for your newly created DB and then select ‘OK’



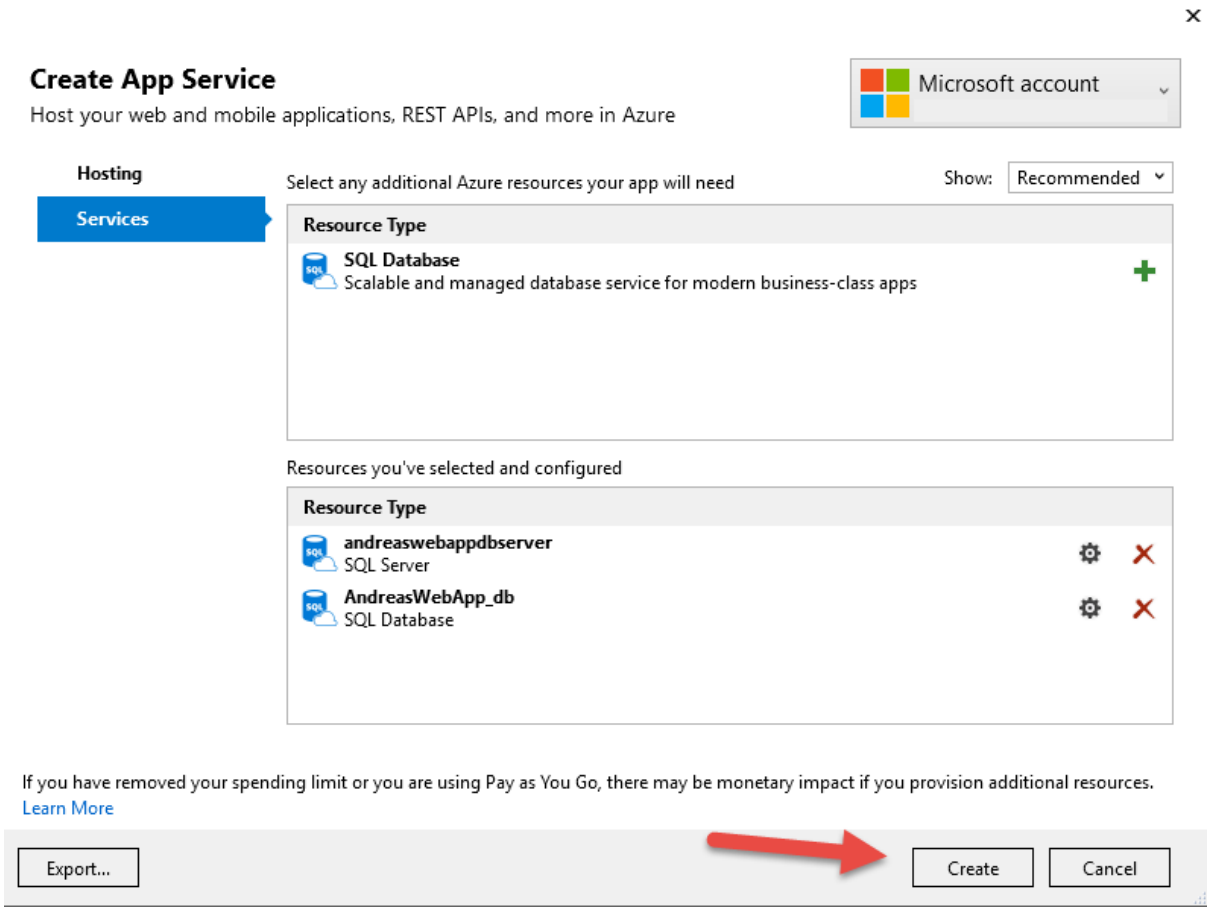
Deploy to Azure

- Your DB information should now appear and then select ‘OK’



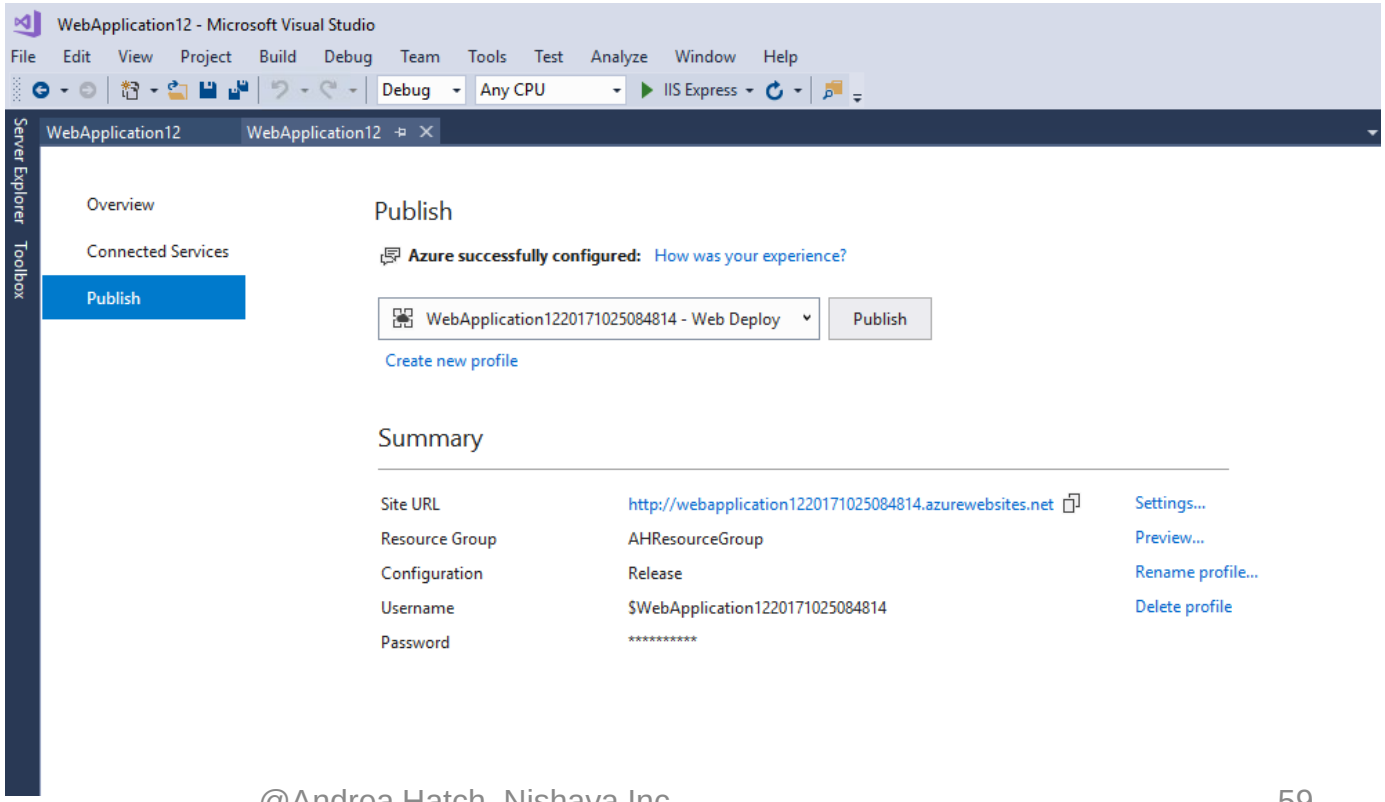
Deploy to Azure

- You can view your newly created Resources before selecting ‘Create’



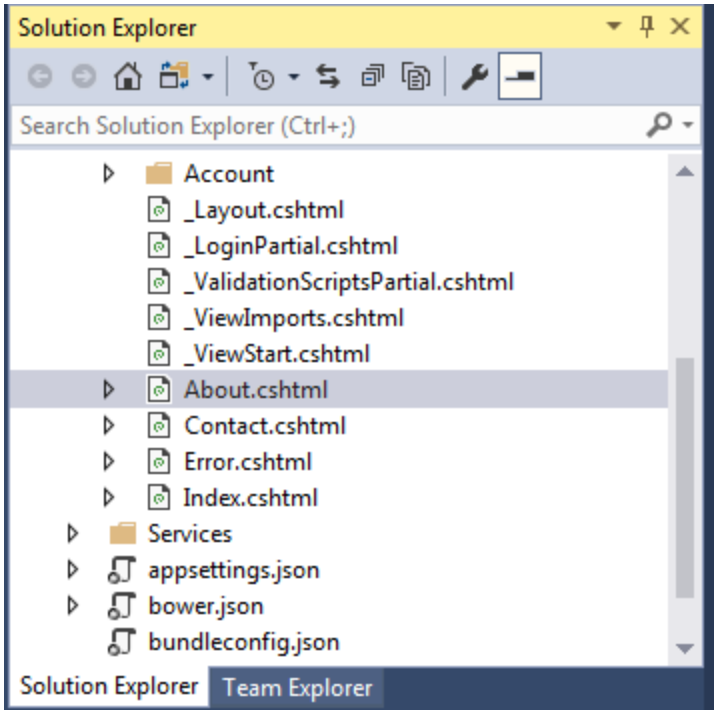
Deploy to Azure

- You will then see the Publish screen and your app should be published now. Your app will also appear in your browser.



Updating the App

- In VS in the Solution Explorer open Pages-> About

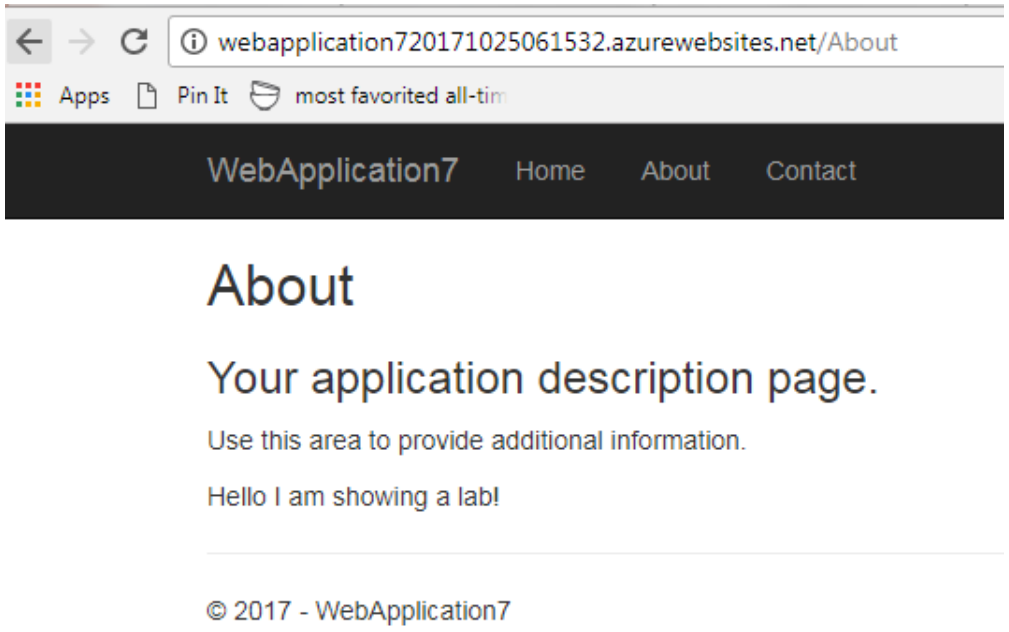


Updating the App

- Update your code to add a little information about the app.
  - Ex. I wrote “Hello I am showing a lab!”

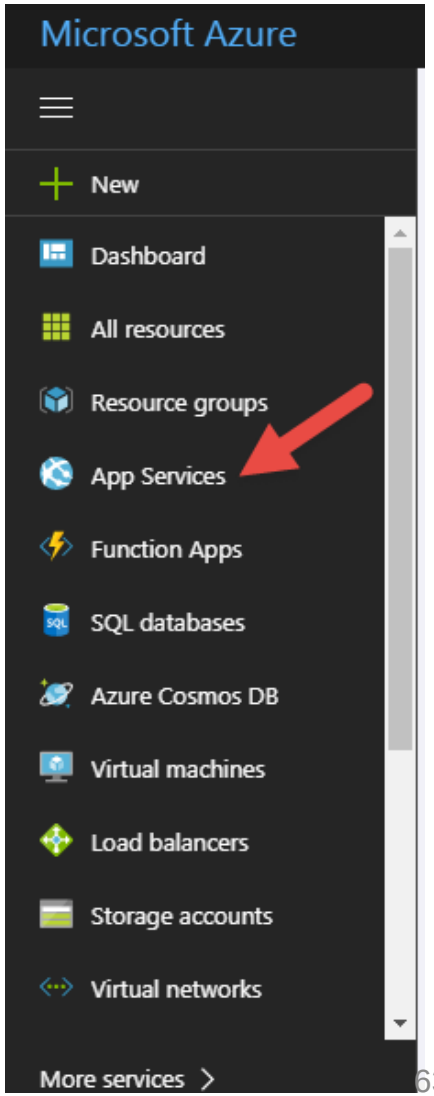
Updating the App

- Redeploy to Azure by right-clicking on the project in your Solution Explorer and selecting ‘Publish’
- Note: Make sure that you close the browser from the last time you published or your publishing will fail



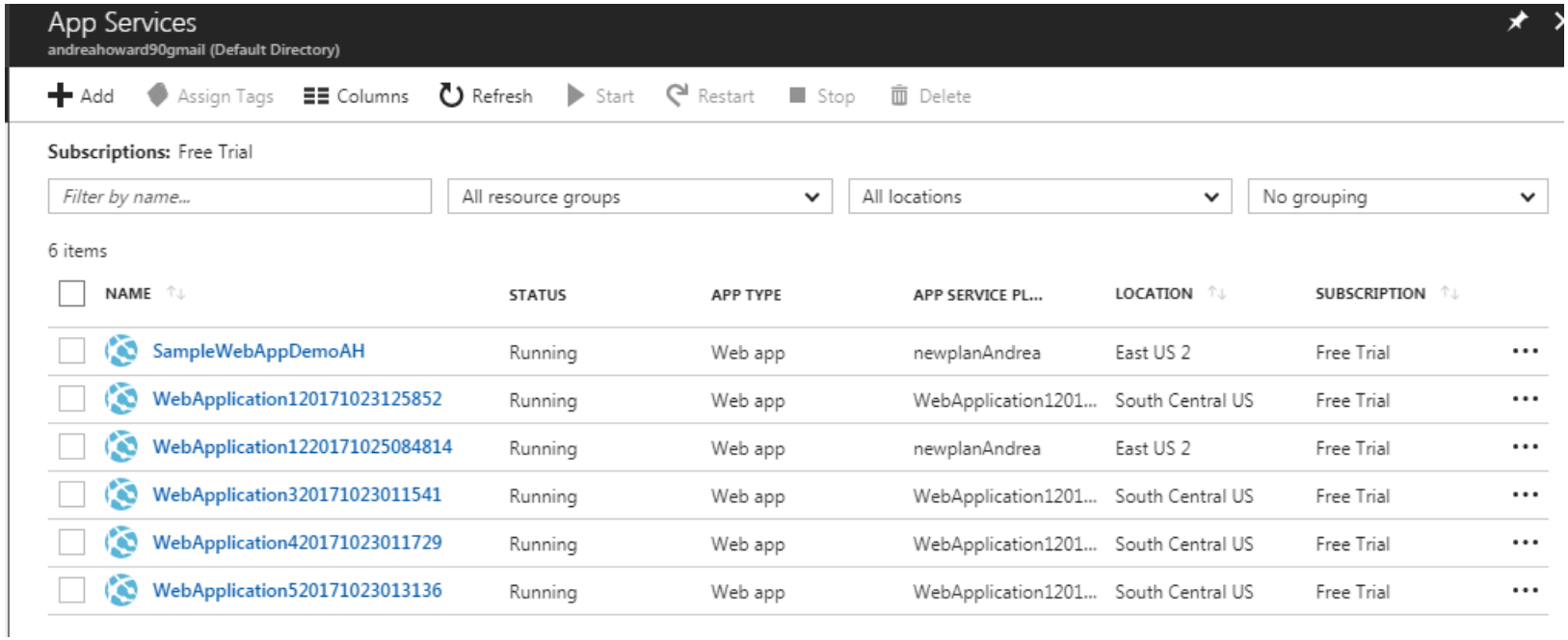
View Web App in Azure

- In Azure Select App Services



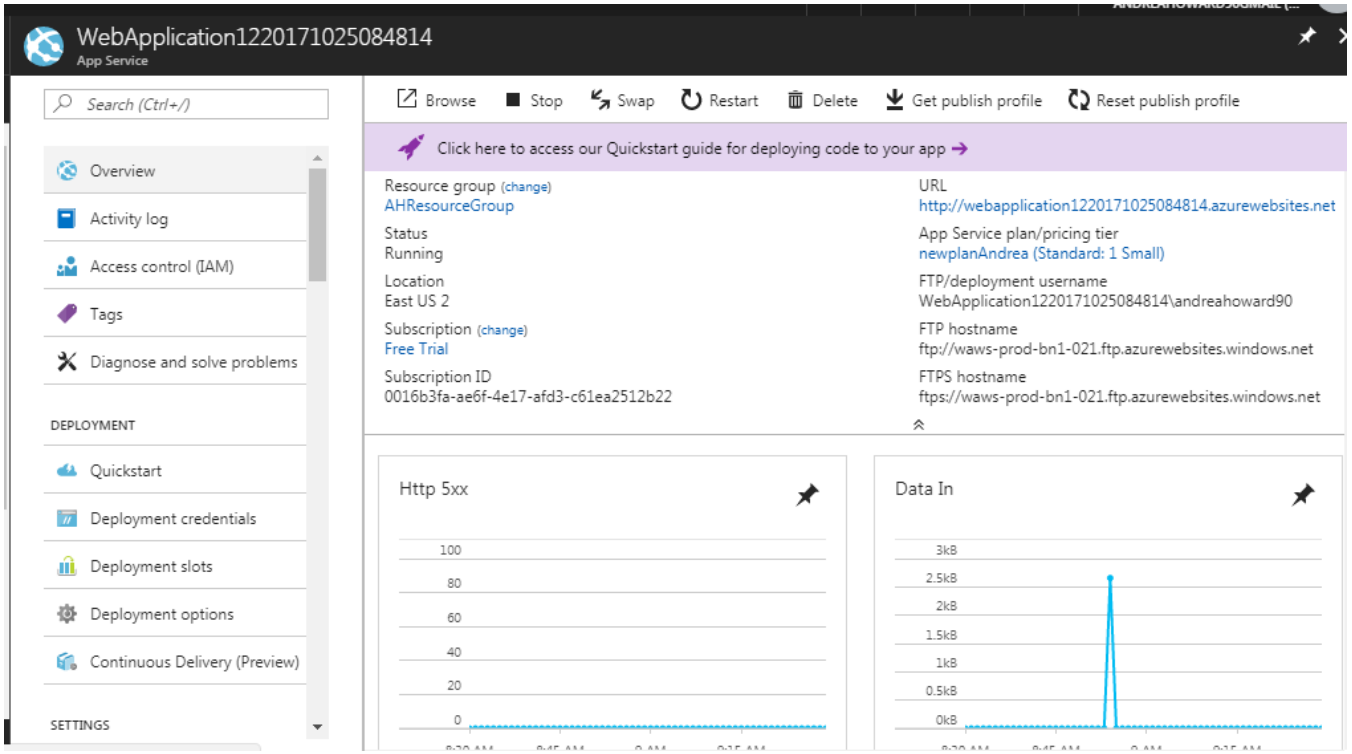
View Web App in Azure

- You can see a list of your App Services



Create an asp.net web app

- Select your web app and view any information about the app



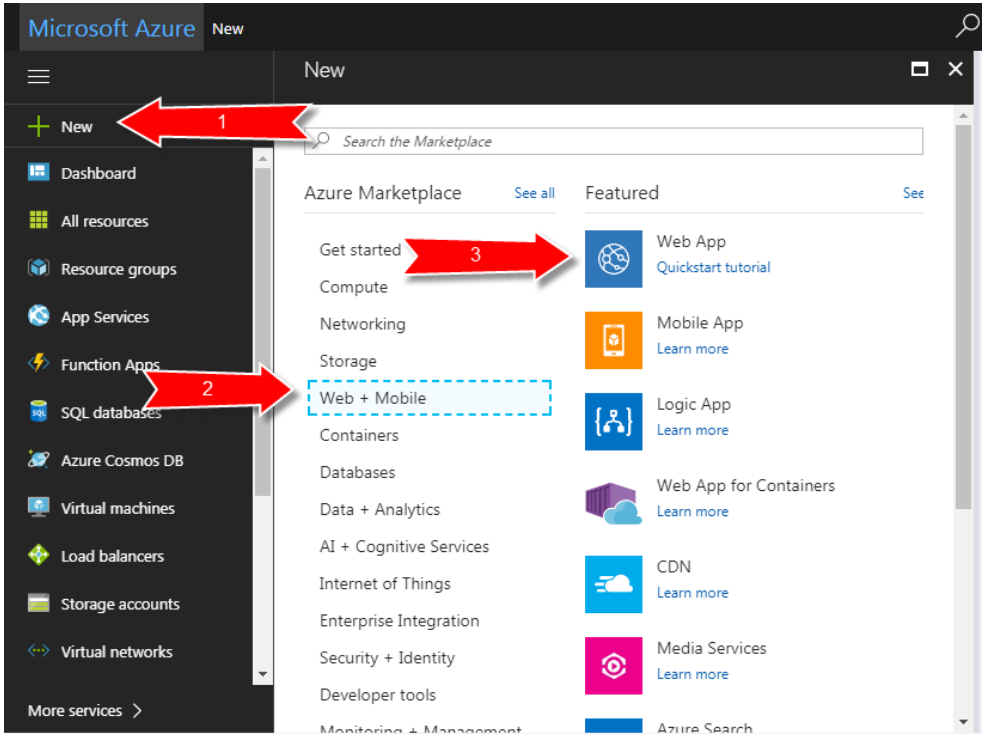
Delete Resources in Azure

- Finally close/delete your web apps and your resources so to ensure you don’t get charged

4. Create a web app in Azure and Update the web app in VS

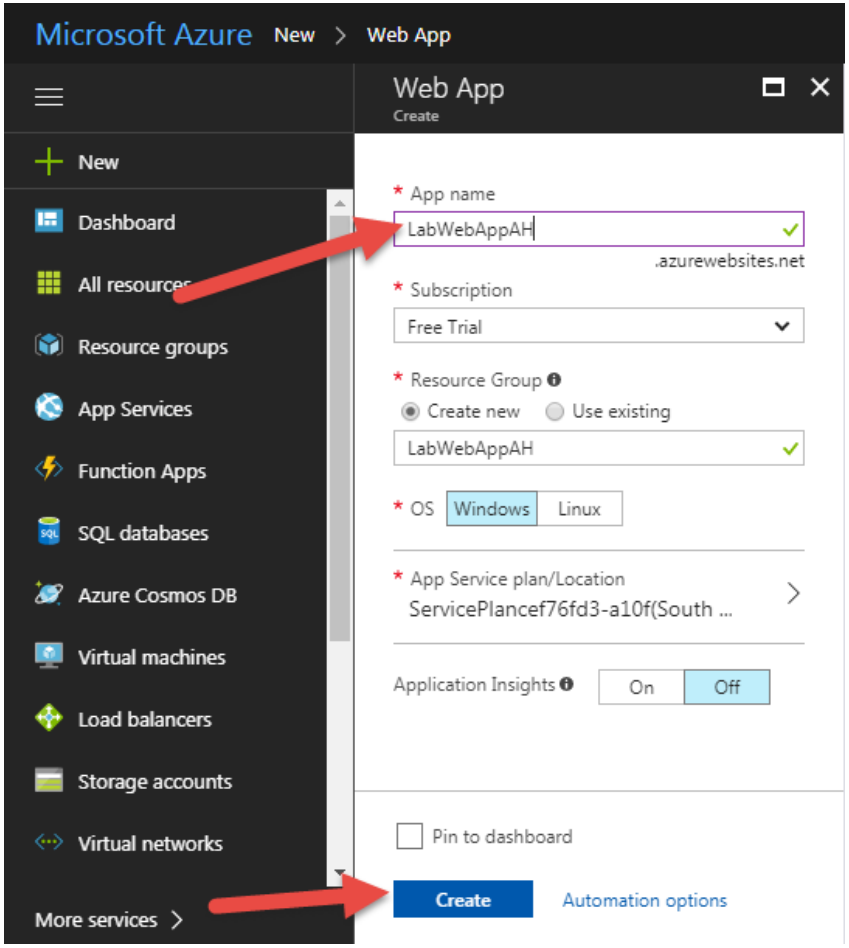
Create a web app in Azure

- Login to your Azure account and then Select ‘New’ then ‘Web + Mobile’ then ‘Web App’



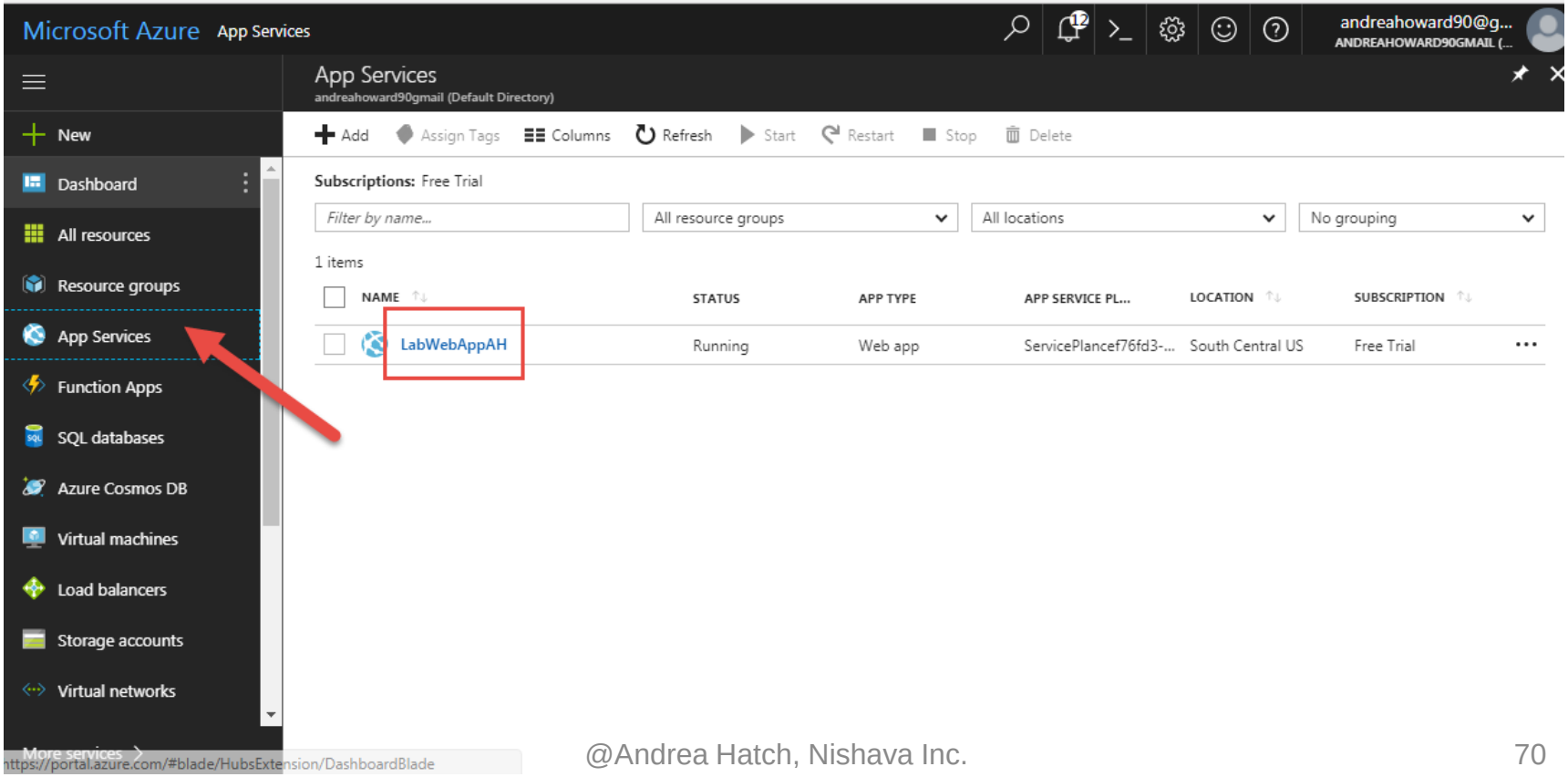
Create a web app in Azure

- Enter a name for your web app and then select ‘Create’



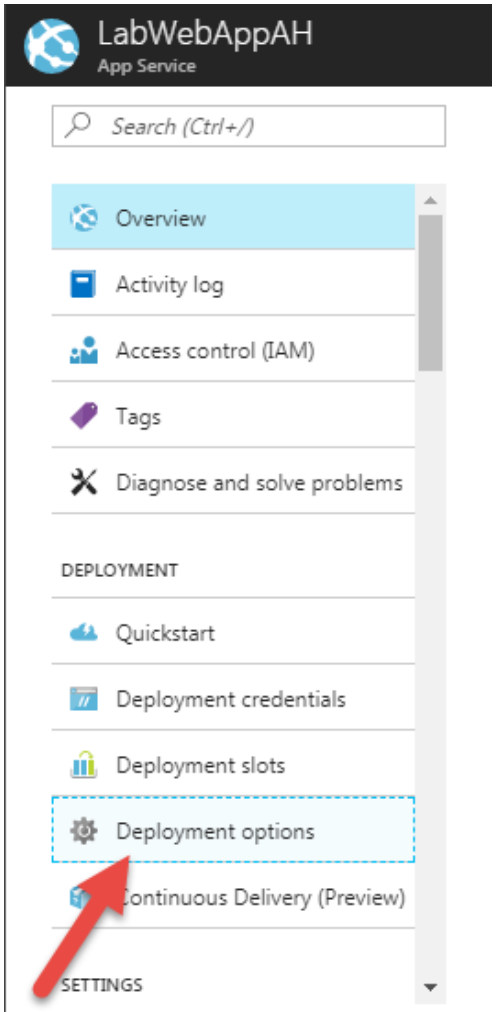
Create a web app in Azure

- You can see your Web App was created by clicking App Services



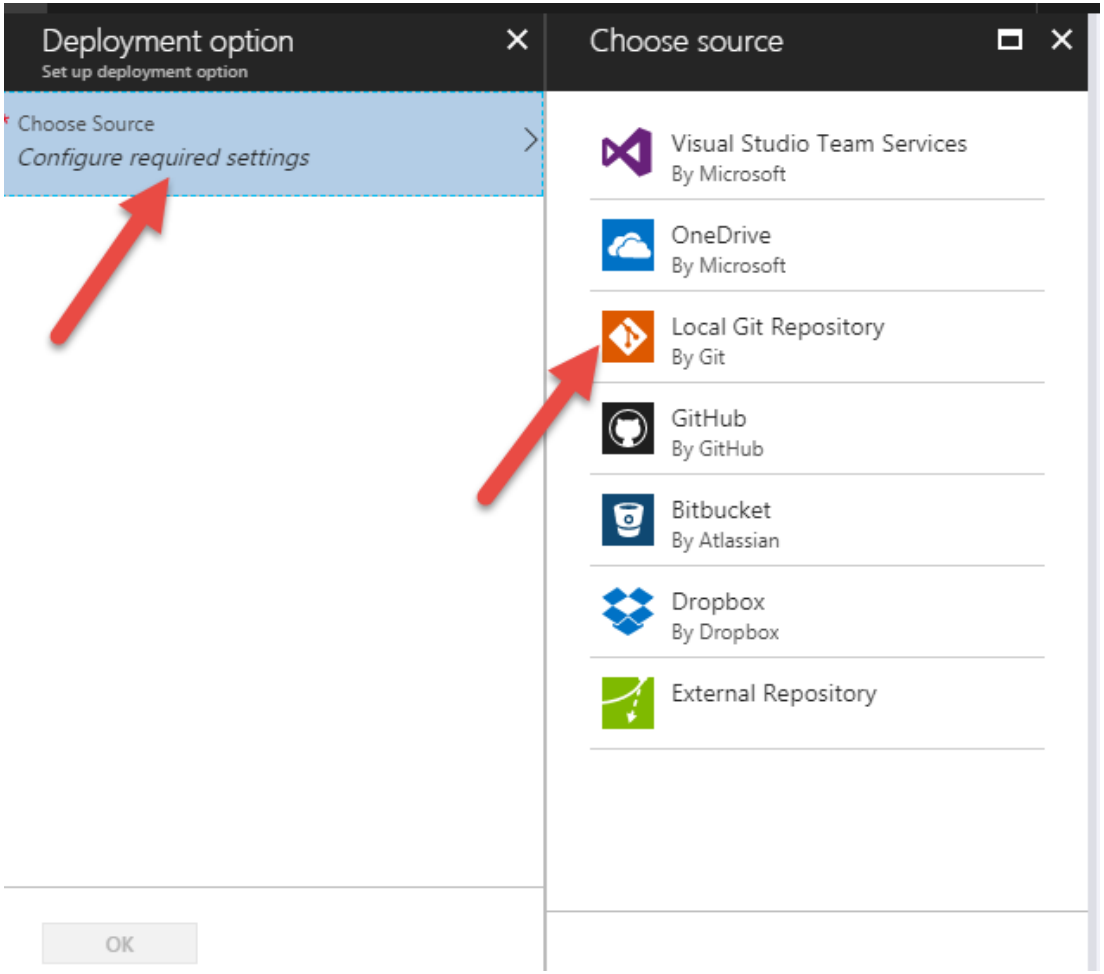
Create a web app in Azure

- Under the Deployment section of your web app select ‘Deployment Options’



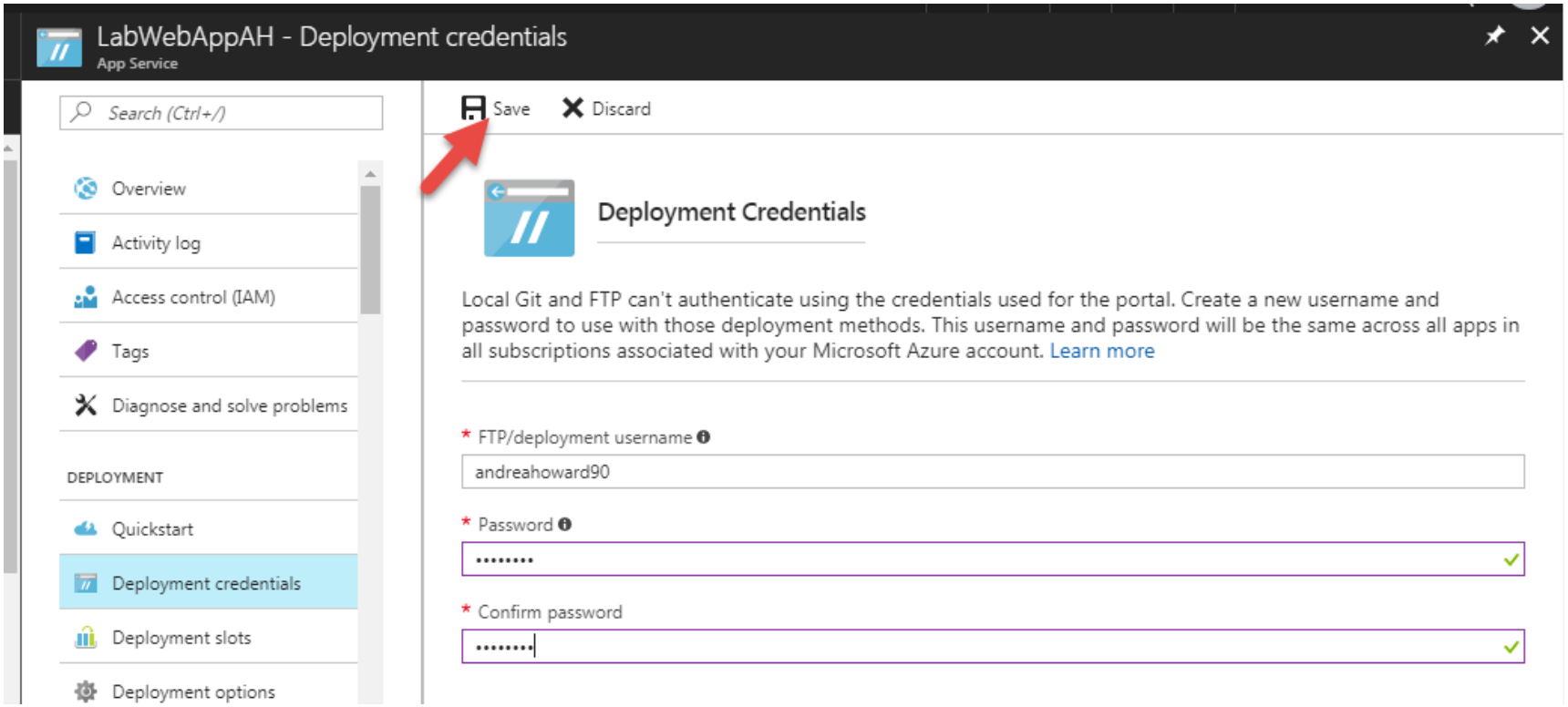
Create a web app in Azure

- Select ‘Choose Source’ and then select ‘Local Git Repository’ and then select ‘OK’



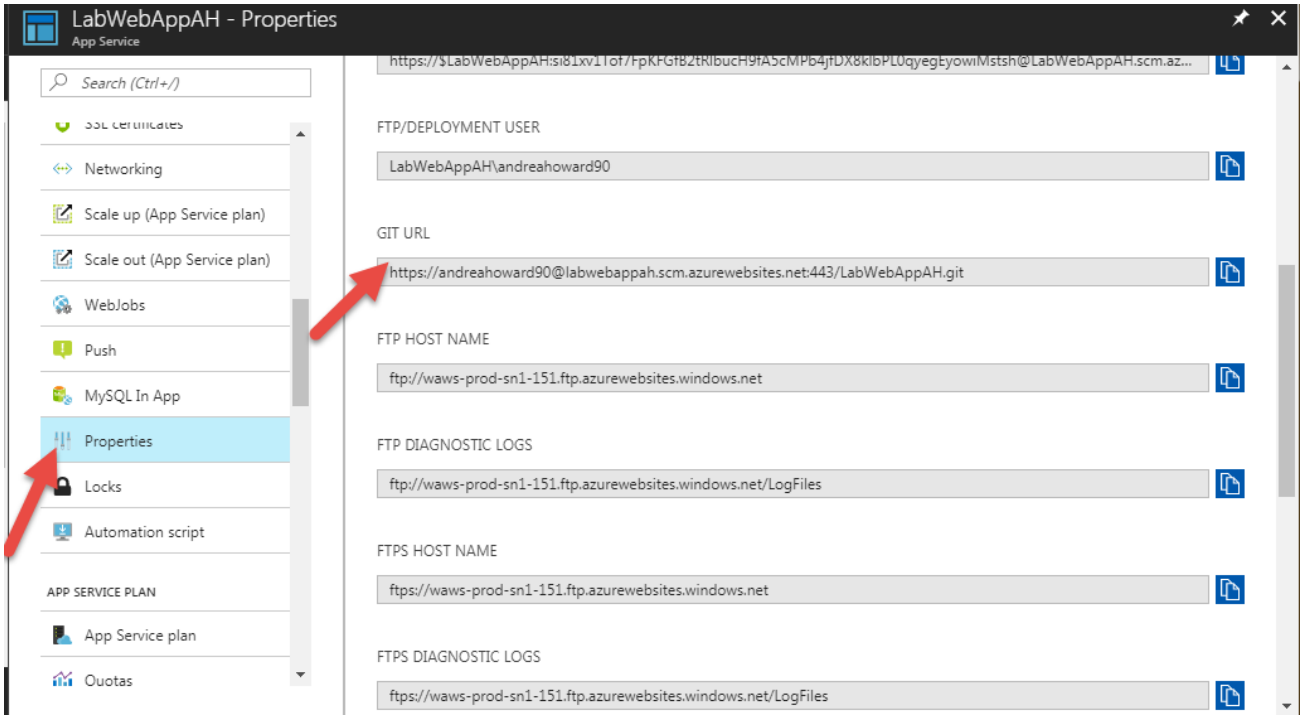
Create a web app in Azure

- Set up deployment credentials under the Deployment section and then select ‘Save’



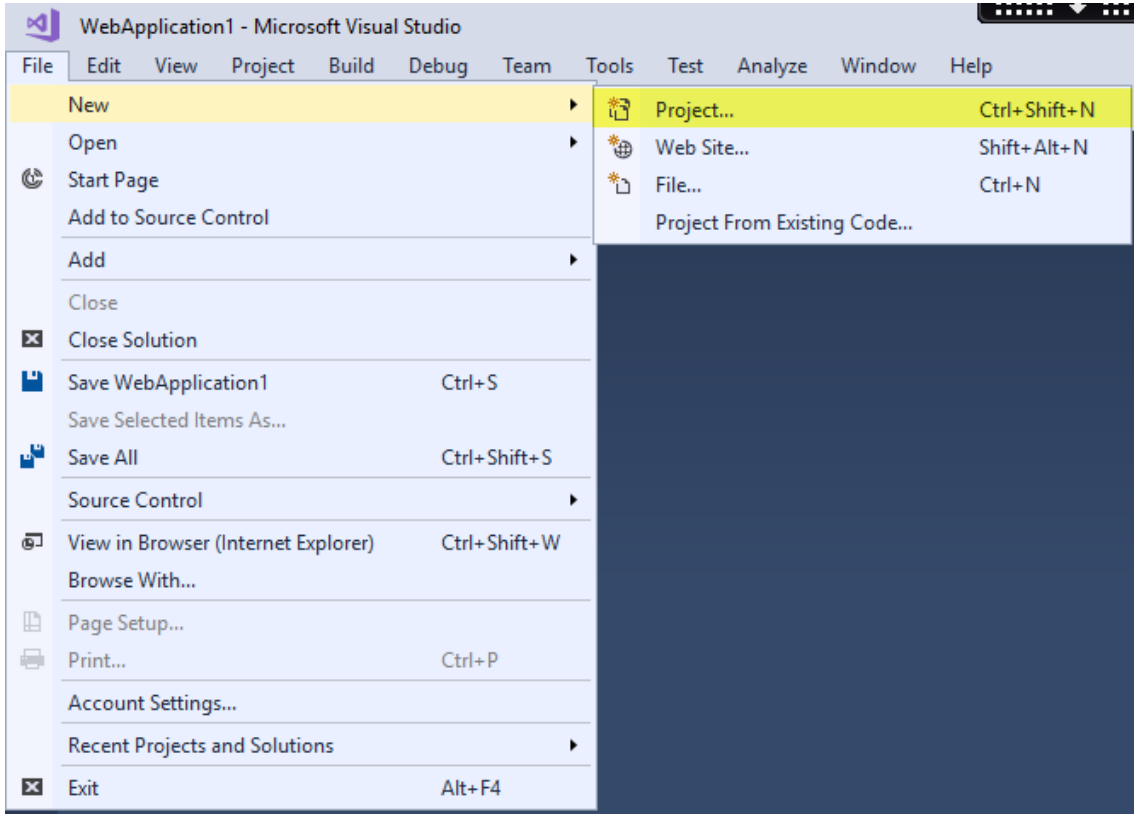
Create a web app in Azure

- Under Settings Properties you will be able to find information about your web app. Copy the GIT URL as we will need this to connect to your app in VS.



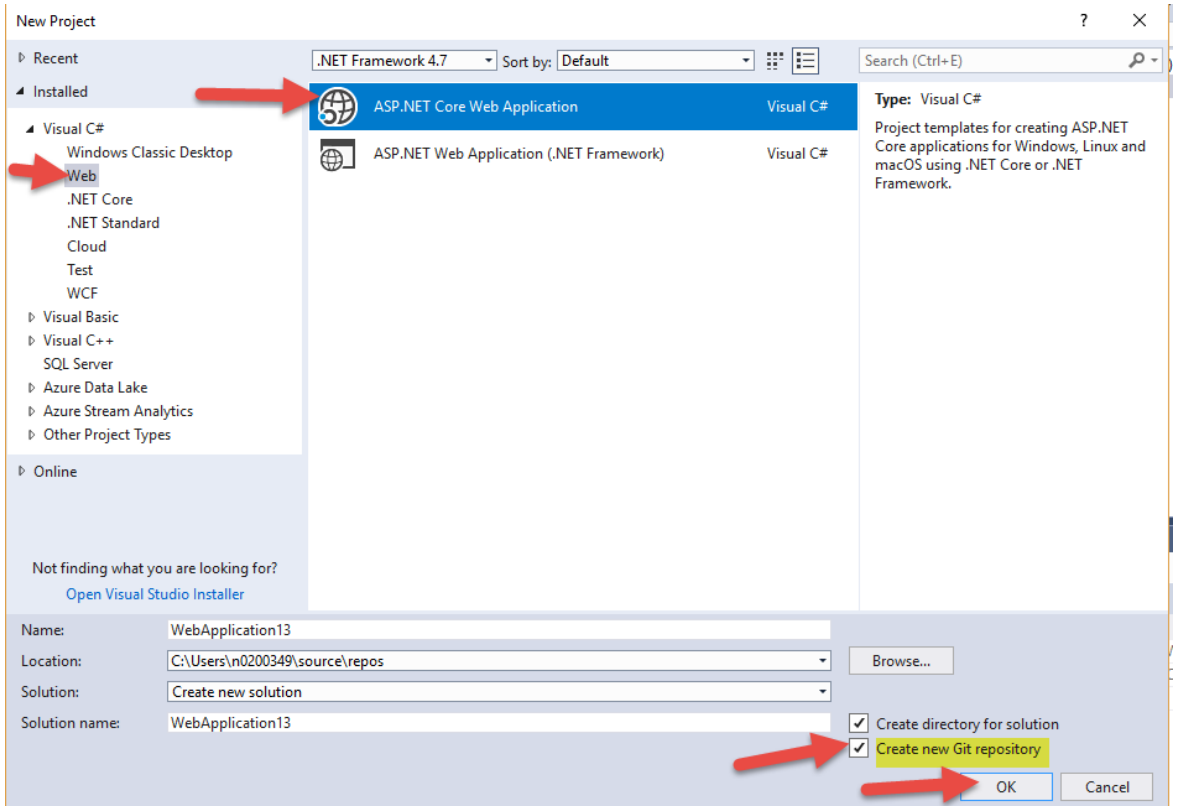
Create a project in VS

- In VS, select File -> New -> Project



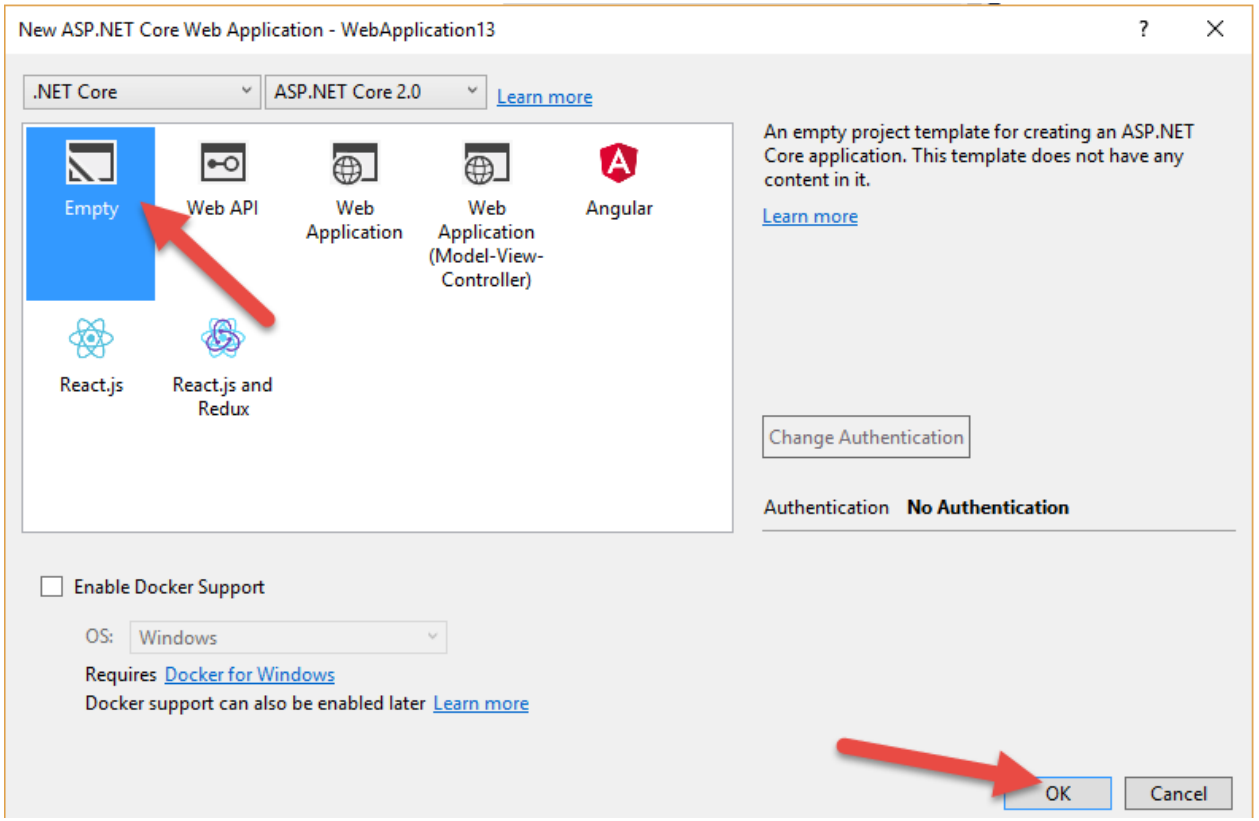
Create a project in VS

- Under Web, select the ASP.NET Core Web App, give the app a Name and Make sure Create new Git Repository is selected then select ‘OK’



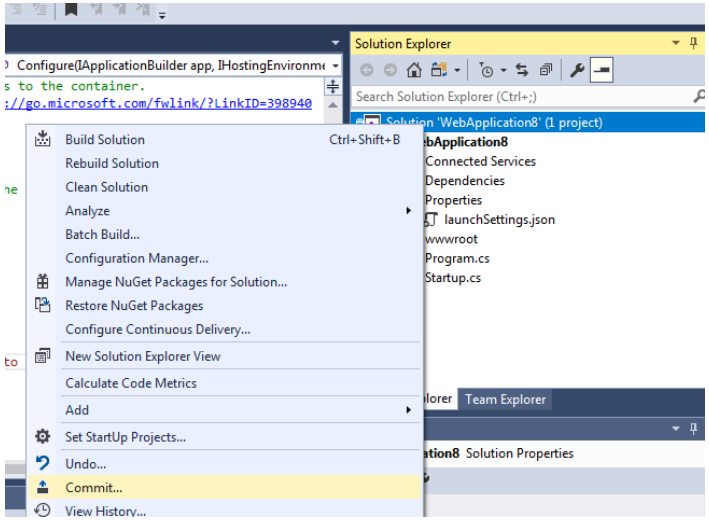
Create a project in VS

- Select ‘Empty’ then select ‘OK’

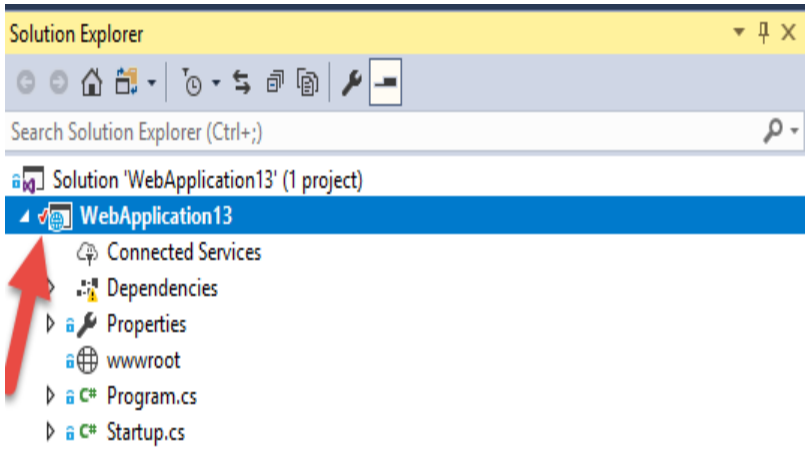


Update your web app in VS

- In VS find the Solution Explorer and right click on the Solution project.

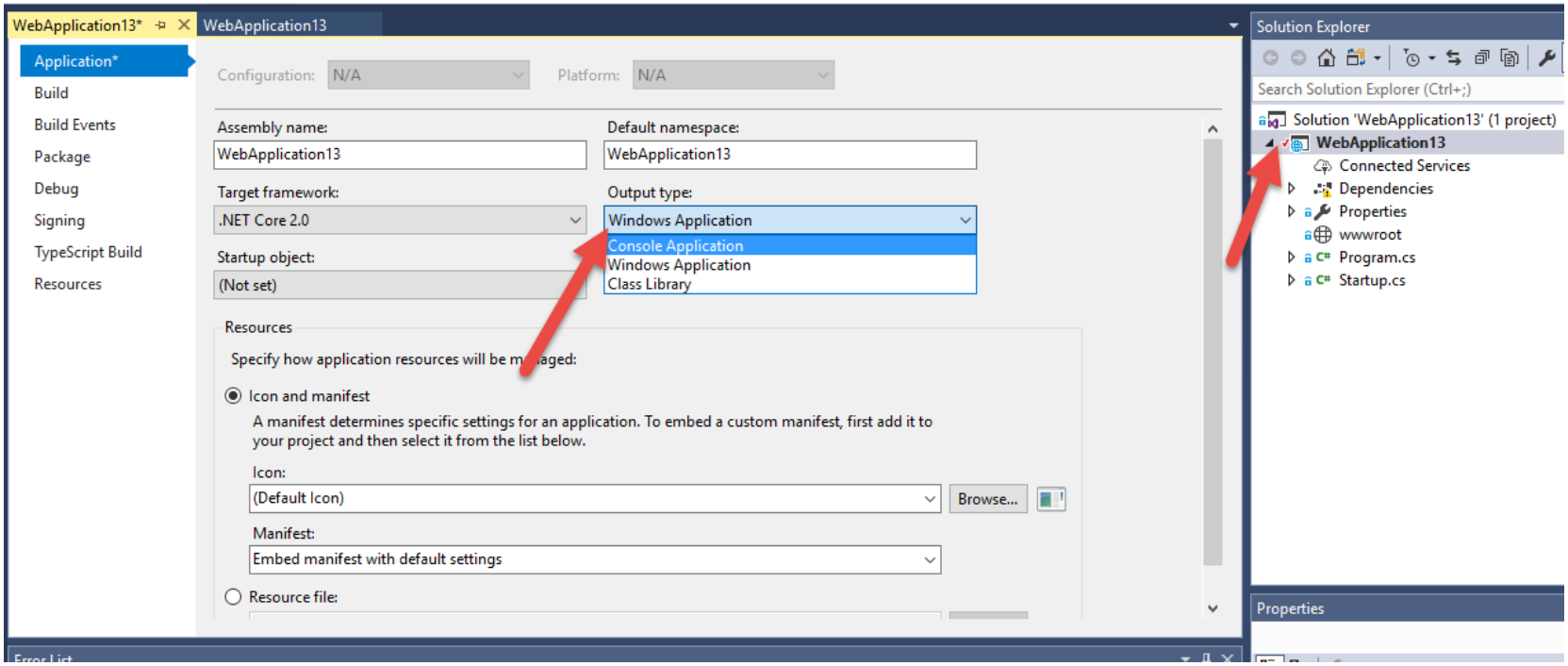


- Check to see if you have the option to Commit. You will know if you can commit your Web App if there is a red check mark next the Web App name.



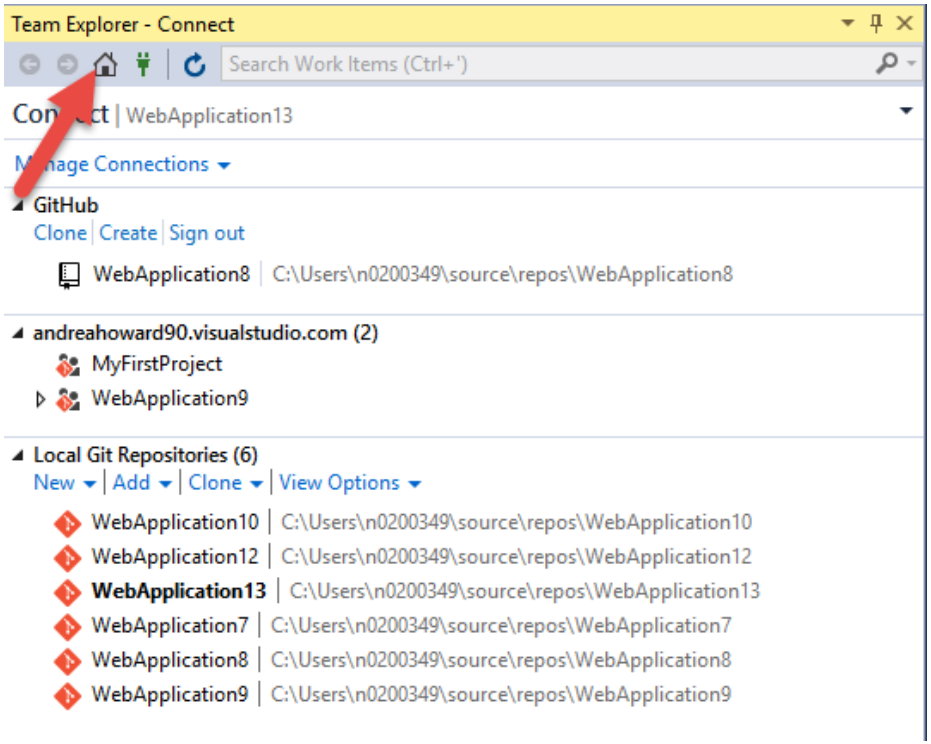
Update your web app in VS

- If you do not have a red check mark you need to make any change to your web app so that it can be committed. For example select the web application and then go under Project -> Properties and change the Output type and this will allow you to commit your changes.



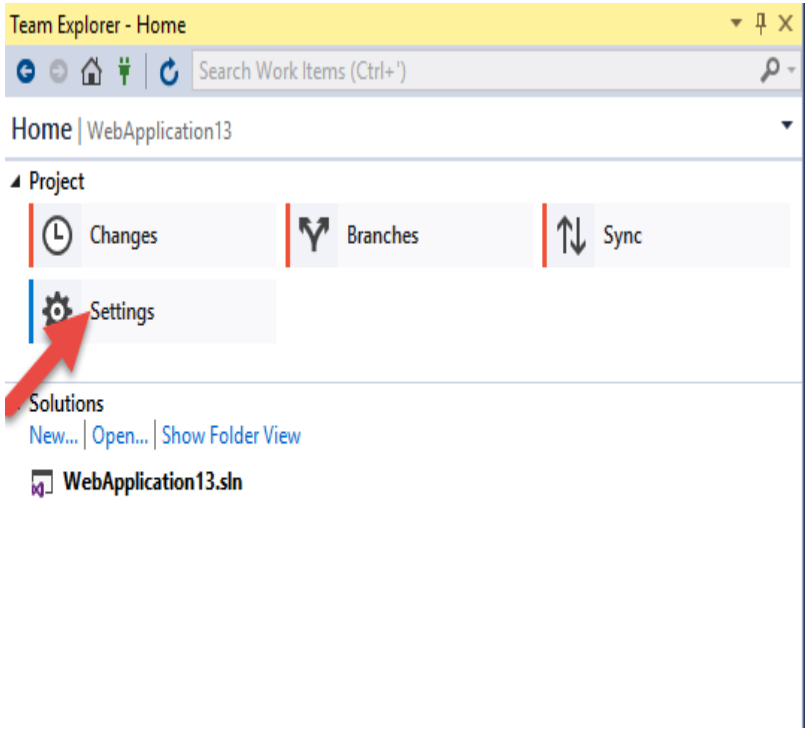
Update your web app in VS

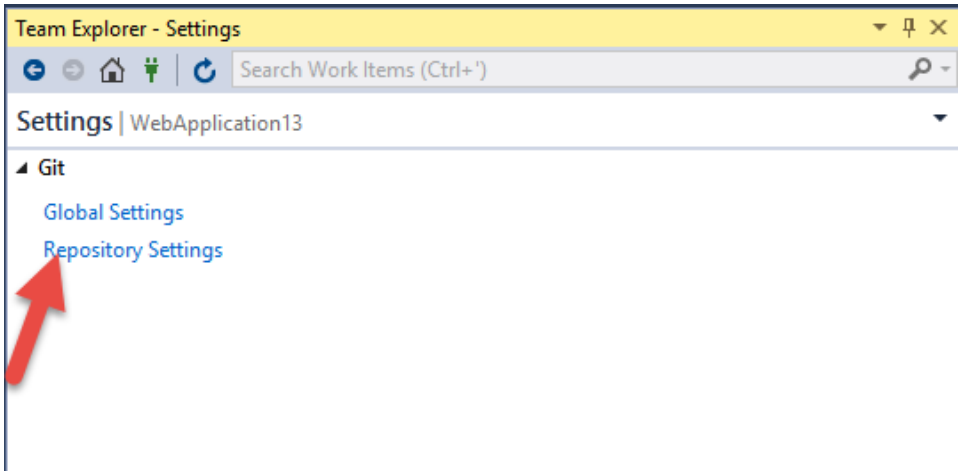
- Once your changes are committed you will be brought to the Team Explorer and you will need to select the home icon



Update your web app in VS

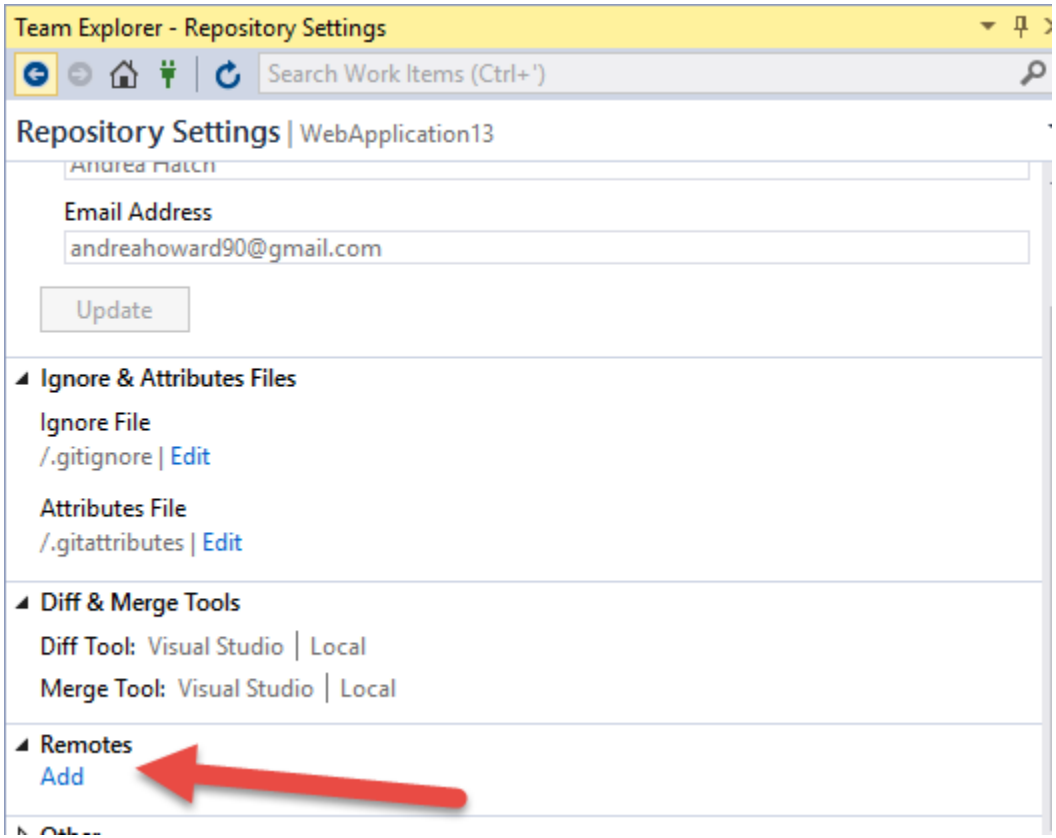
- Select Settings -> Repository Settings





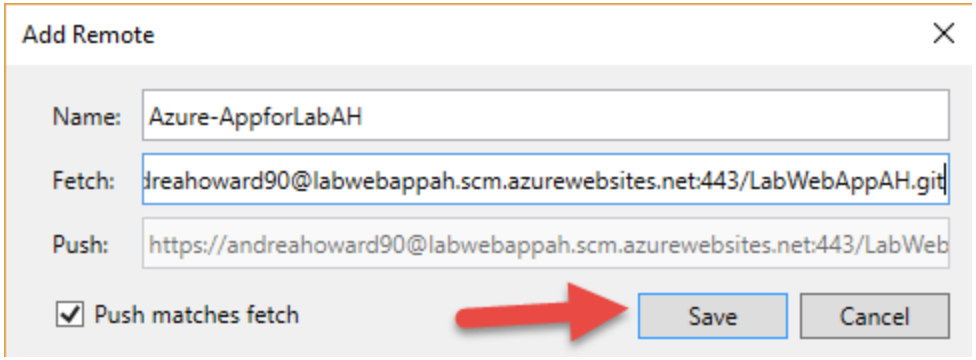
Update your web app in VS

- Under Remotes select ‘Add’



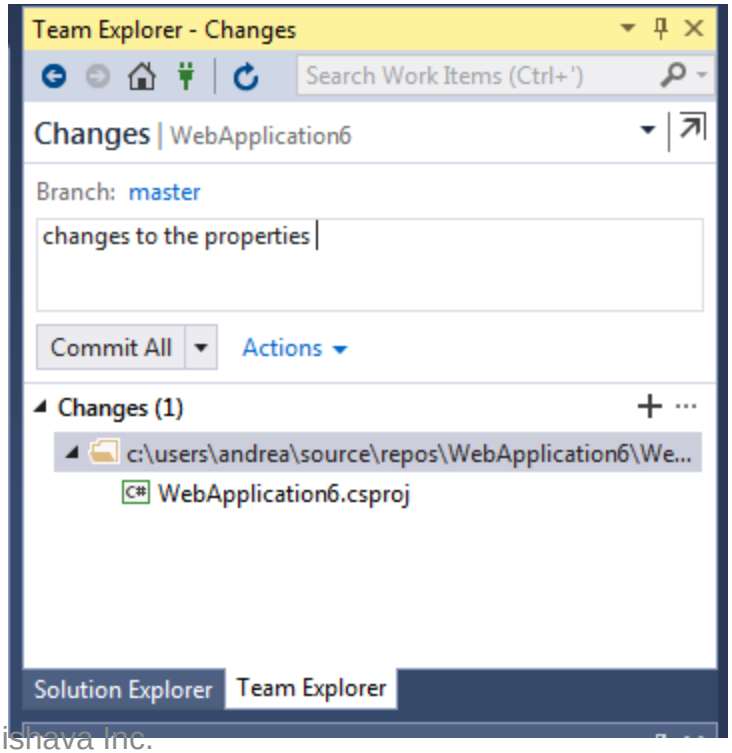
Update your web app in VS

- Give your Remote a name and then paste in your git URL from Azure to the Fetch section and then select ‘Save’



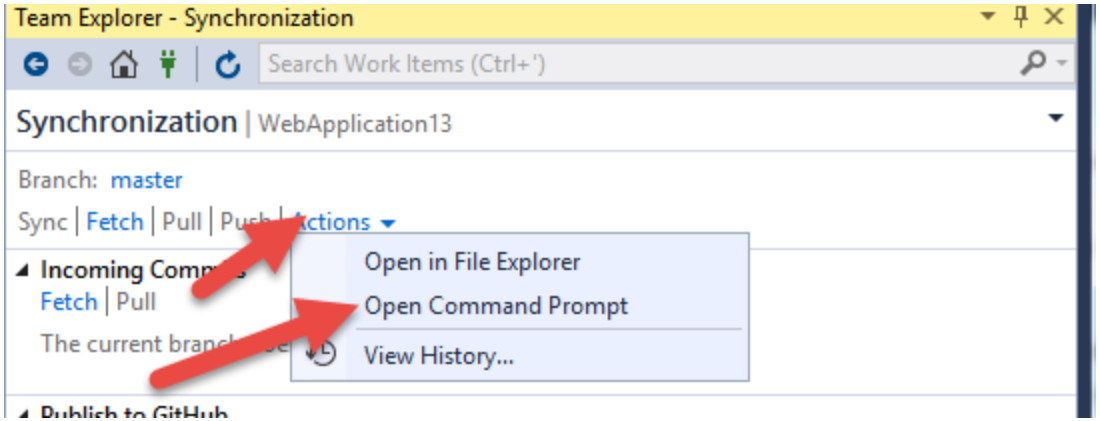
Update your web app in VS

- Select the home icon again and this time select ‘Changes’ so that we can commit the changes. Enter in a commit message and select ‘Commit All’



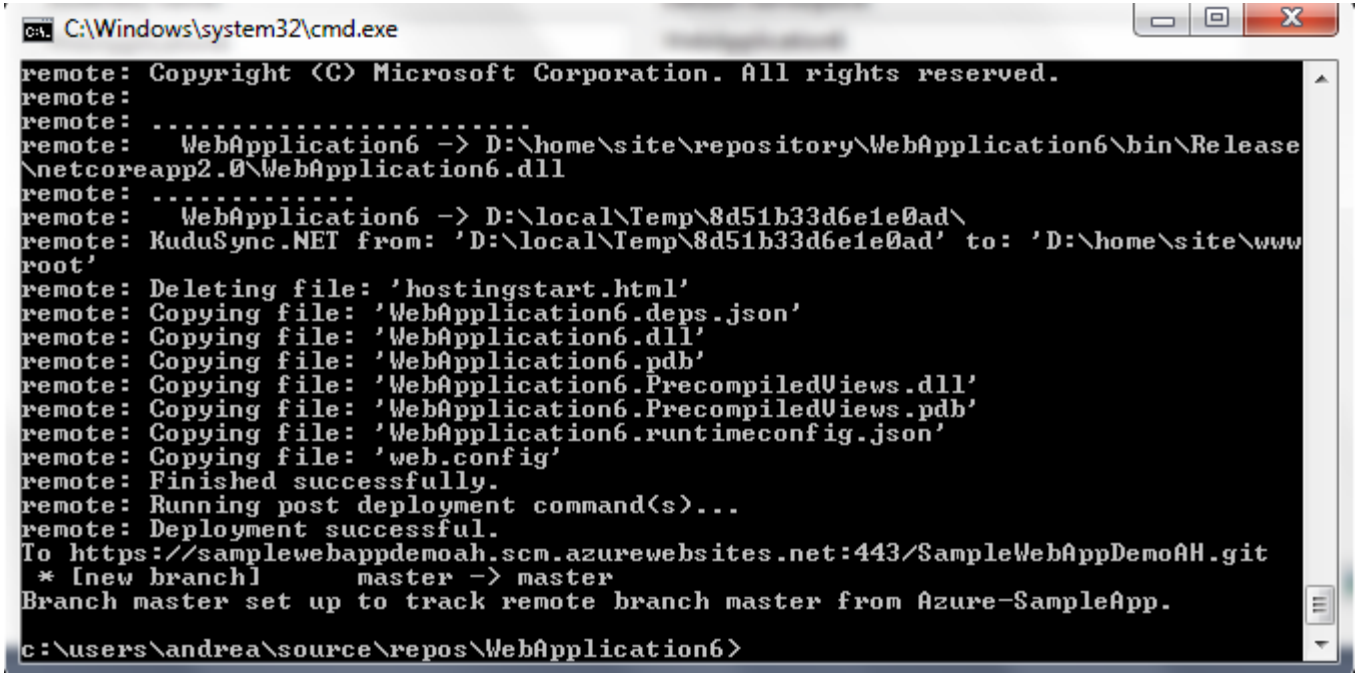
Update your web app in VS

- Open a command prompt to sync your changes with your Web App in Azure by going into Team Explorer and selecting the home icon -> Sync -> Actions -> Open Command Prompt



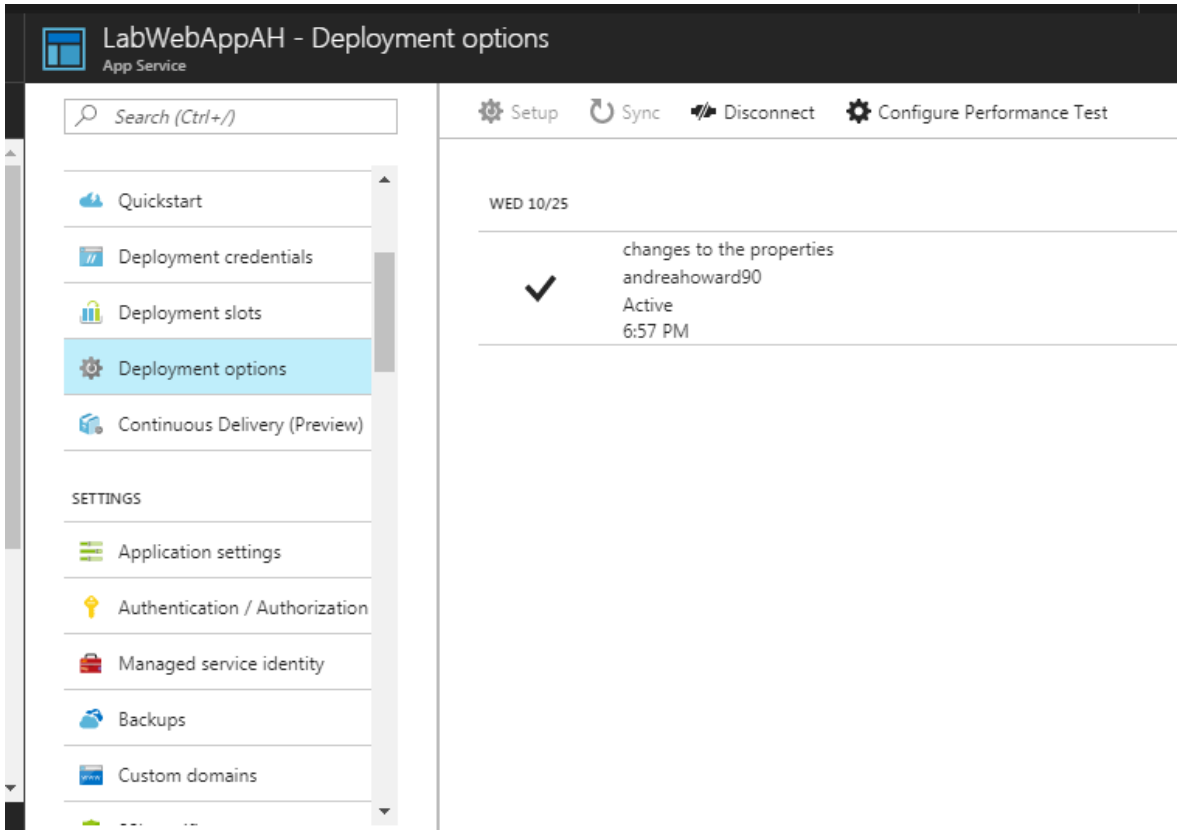
Update your web app in VS

- Enter in: git push –u (name of Remote) master. Your app will start syncing and once done will say ‘Deployment Successful’
  - Ex: git push –u Azure-AppforLabAH master



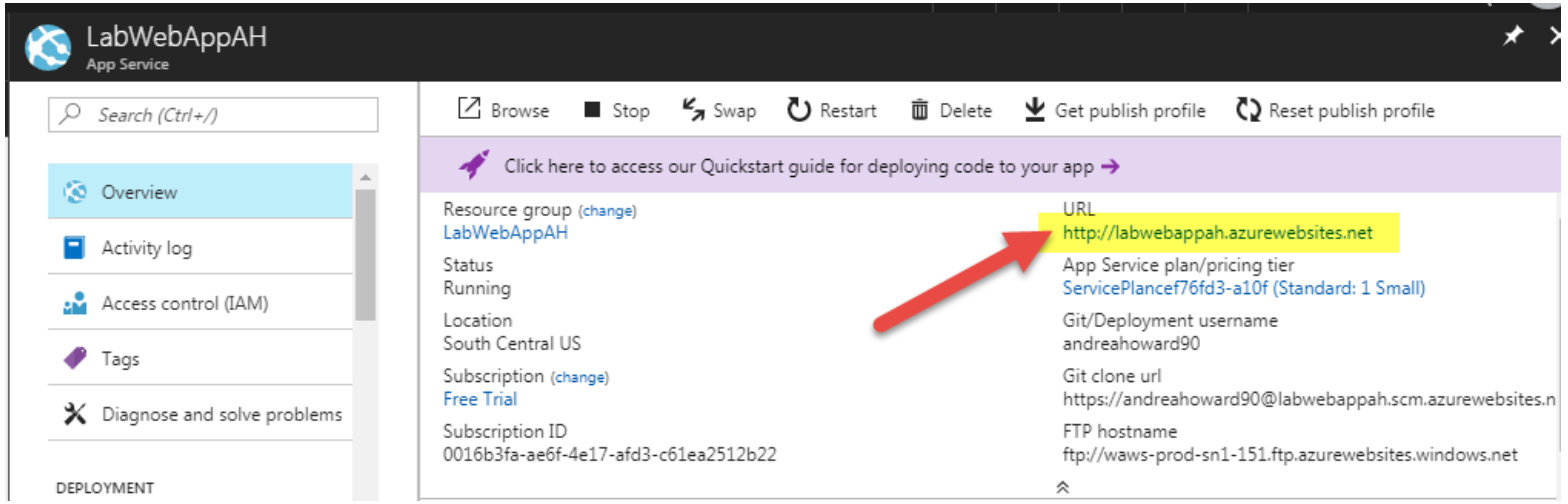
Update your web app in VS

- Verify your deployment in Azure by selecting your Web App and then Deployment options



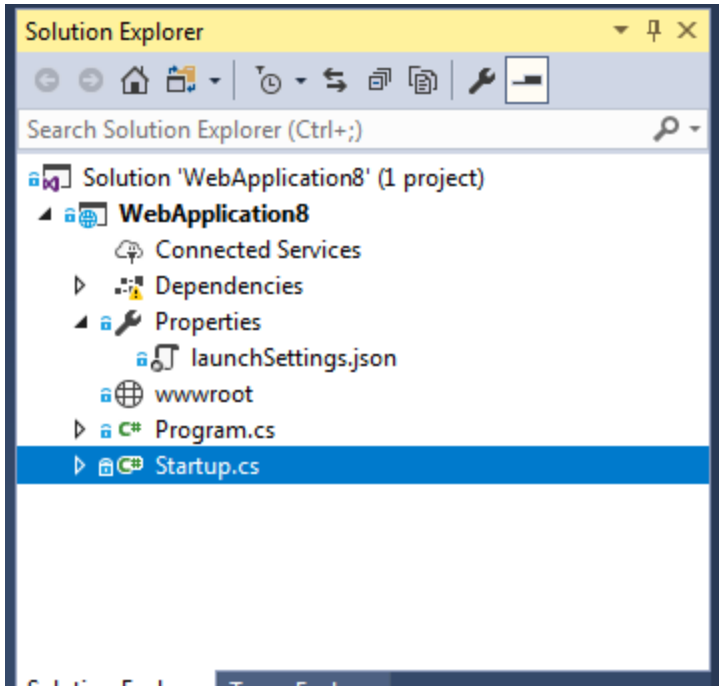
Update your web app in VS

- To run the app you can select Overview and see the URL to your app.



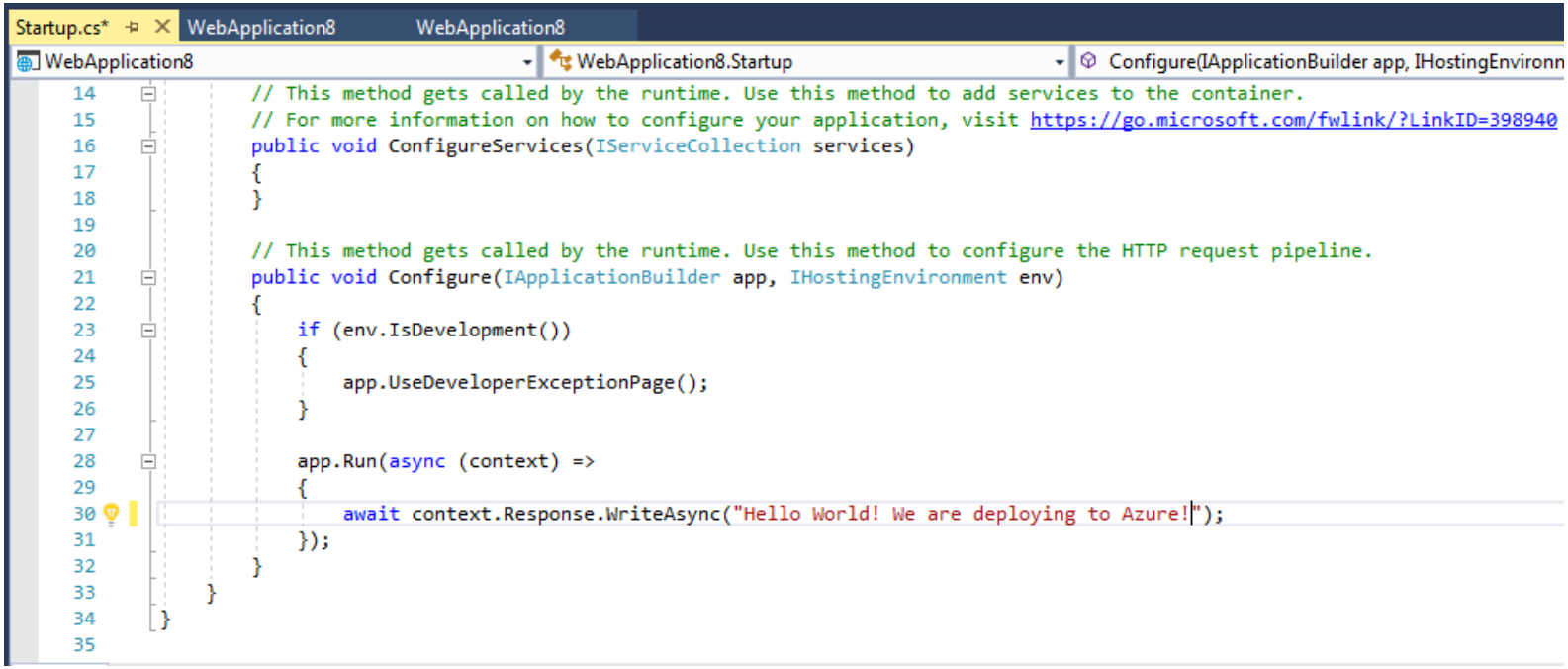
Update your web app in VS

- You can update your app again in VS by going to the Solution Explorer and opening up the Startup.cs.file



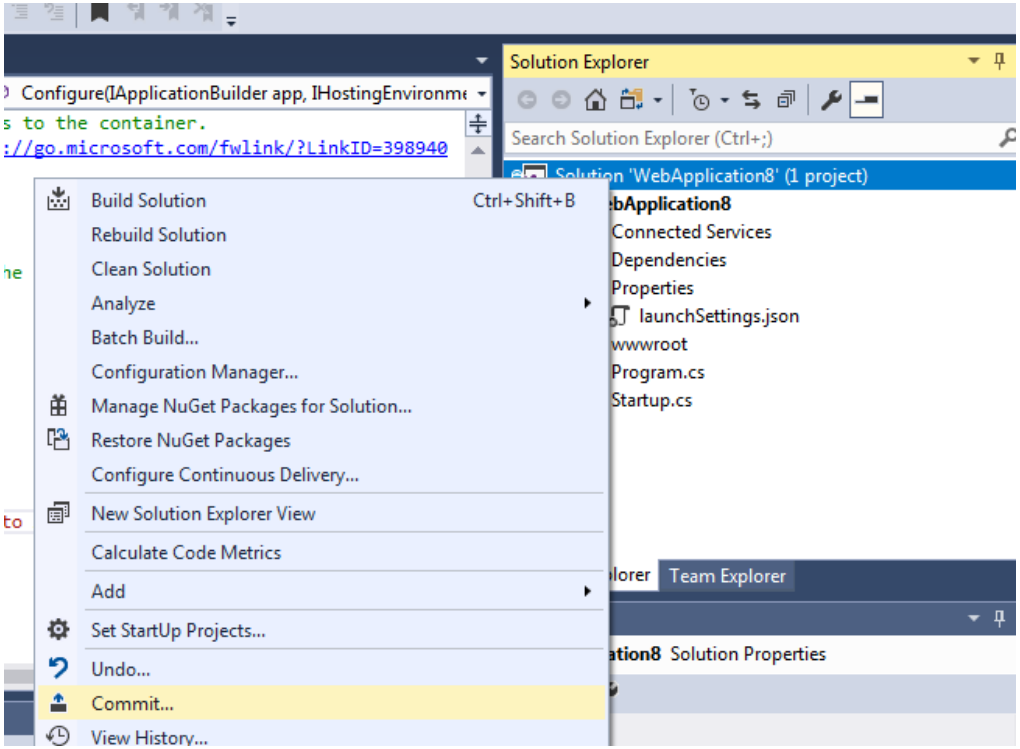
Update your web app in VS

- In the Configure method modify Response.WriteAsync and type a message and then select Save:
  - Ex: await context.Response.WriteAsync("Hello World! We are deploying to Azure.");



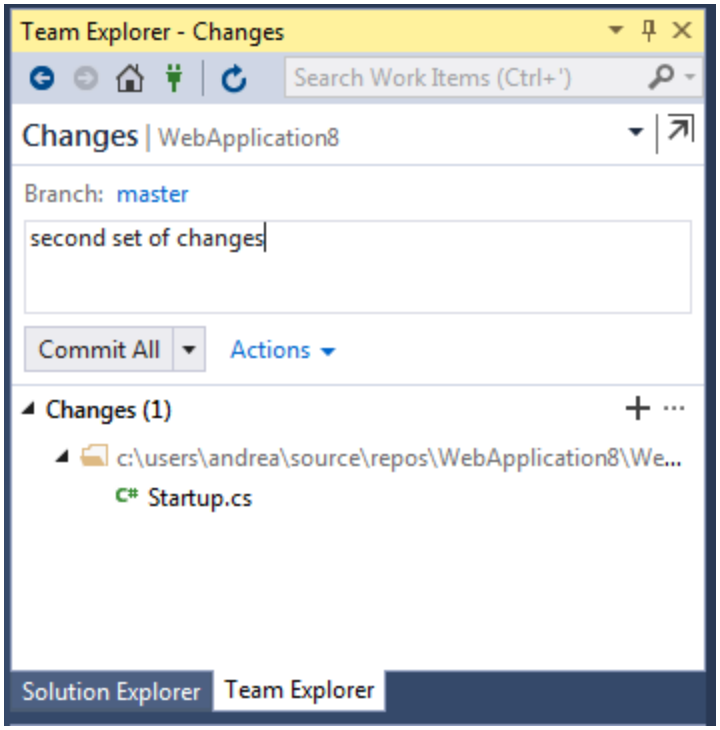
Update your web app in VS

- Make sure to save your changes and then right click again on the Solution Web App name in VS and select ‘Commit’. Enter your commit message and then press the Commit button.



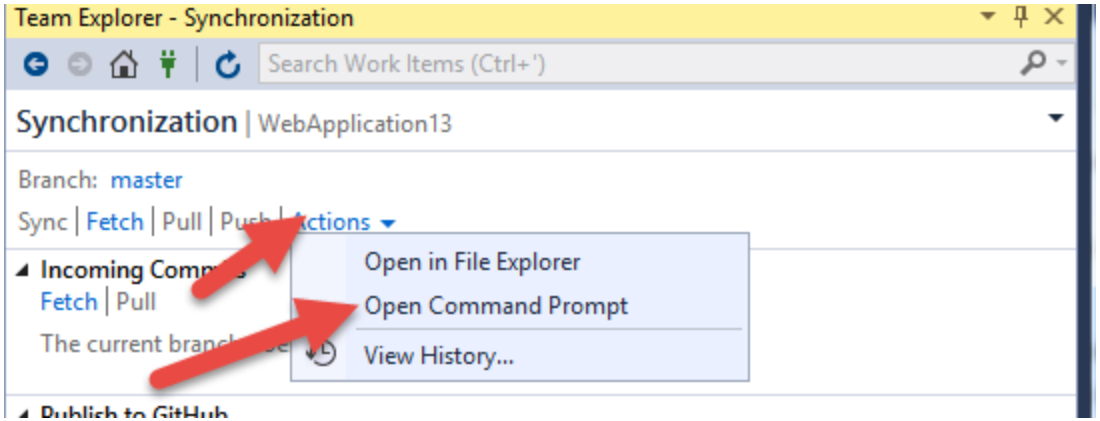
Update your web app in VS

- In the Team explorer click on the home icon -> changes then type a comment and Commit



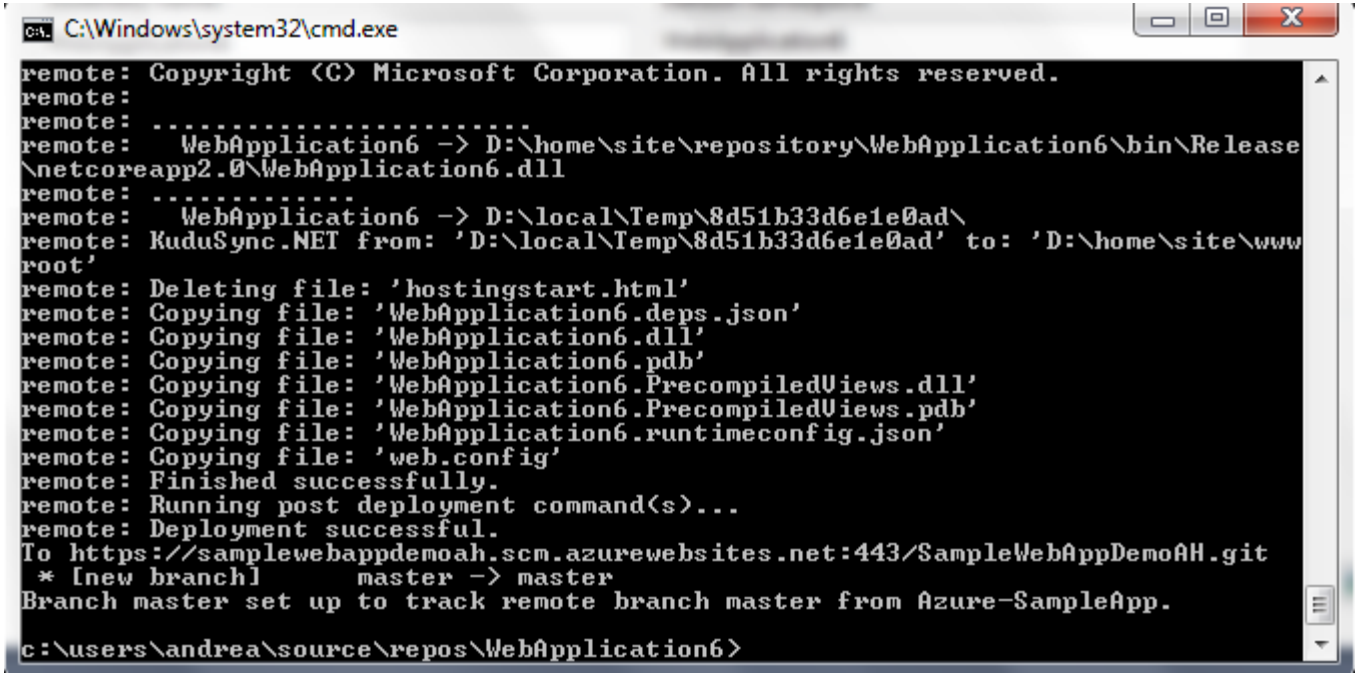
Update your web app in VS

- Open a command prompt to sync your changes with your Web App in Azure by going into Team Explorer and selecting the home icon -> Sync -> Actions -> Open Command Prompt



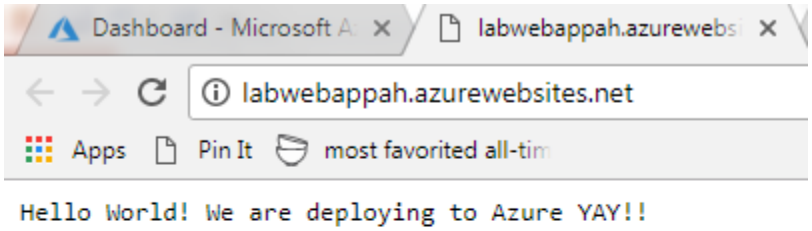
Update your web app in VS

- Enter in: git push –u (name of Remote) master. Your app will start syncing and once done will say ‘Deployment Successful’
  - Ex: git push –u Azure-AppforLabAH master



Update your web app in VS

- Finally, open a browser and enter the URL to your web app to view your update.



Update your web app in VS

- You can also view your webpage on your local host by selecting Debug -> Start without Debugging in VS