**Azure Website Demo**

Step 1: Create ASP.Net website

Step 2: Create Git local repository

Step 3: Create Website on Azure portal

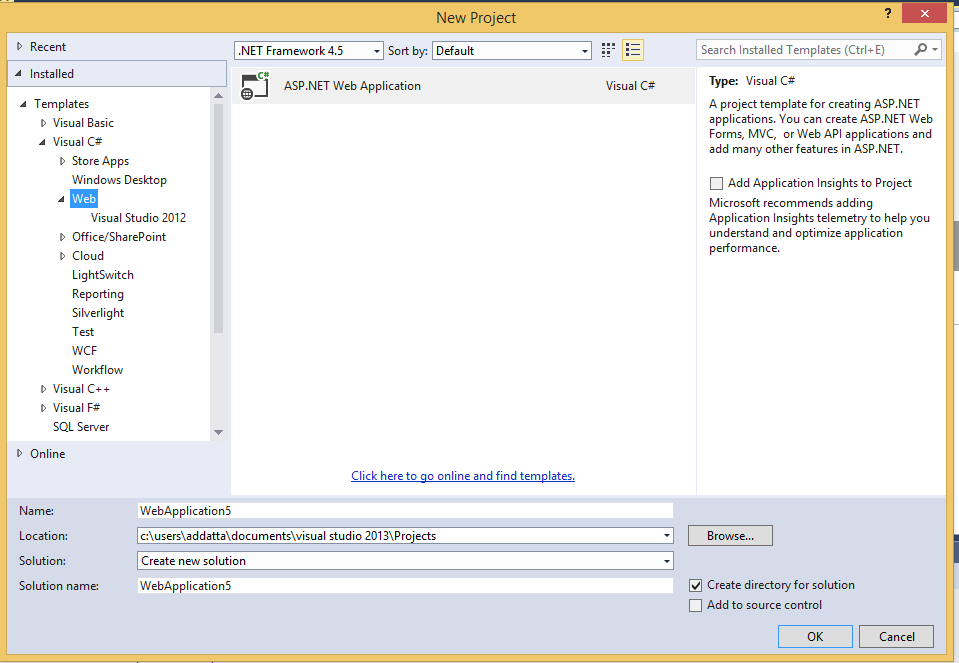
Step 4: Deploy using Git, publish website on Azure

Step 5: Continuous Integration: Modify code and redeploy

Step 6: Staging and Deployment Slots

Step 1: Create a ASP.NET website

1. Open Visual Studio, click on New-> Visual C#-> Web-> ASP.net web application



Step 2: Create Git local repository

Create the git repository for the website above. Open Command Prompt or git bash and execute the following

* 1. Initialize the directory:

Git init .

* 1. Add all /new changed files:

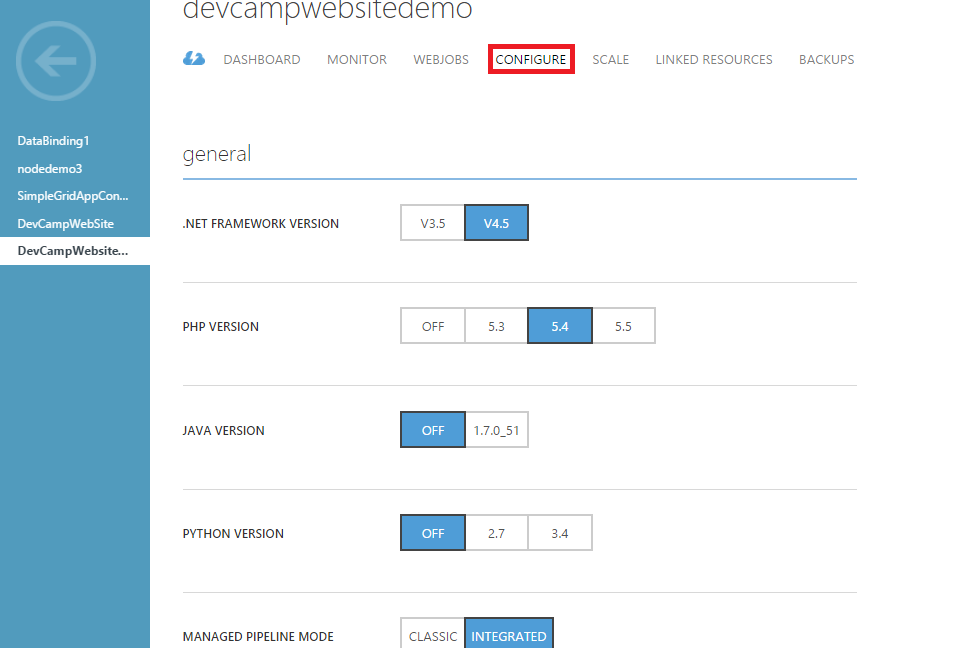
Git add .

* 1. Commit all files:

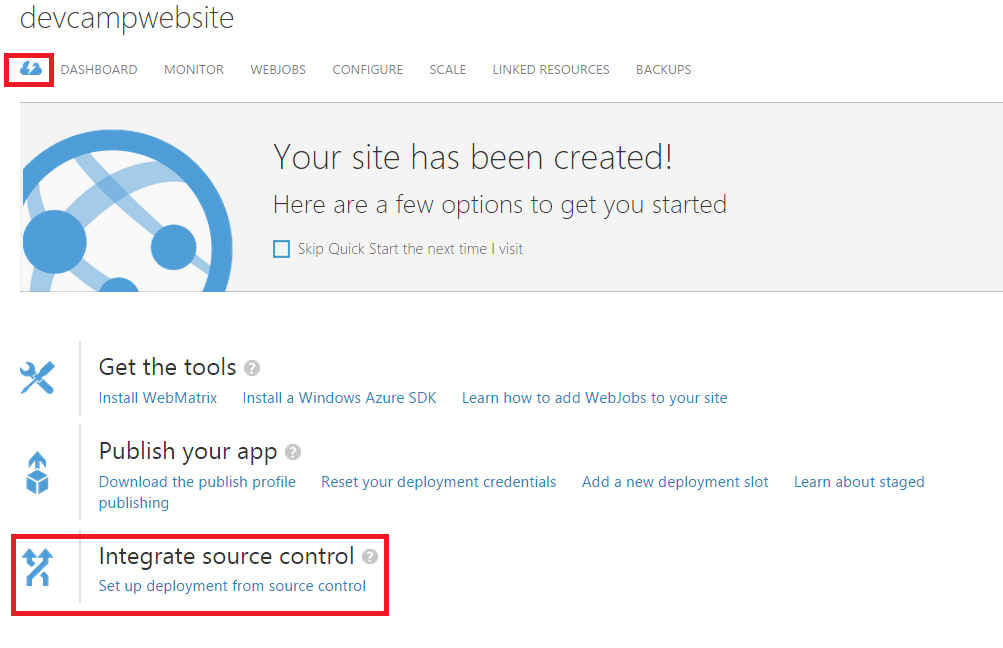
Git commit –am “Initial Commit”

Step 3: Create Website on Azure portal

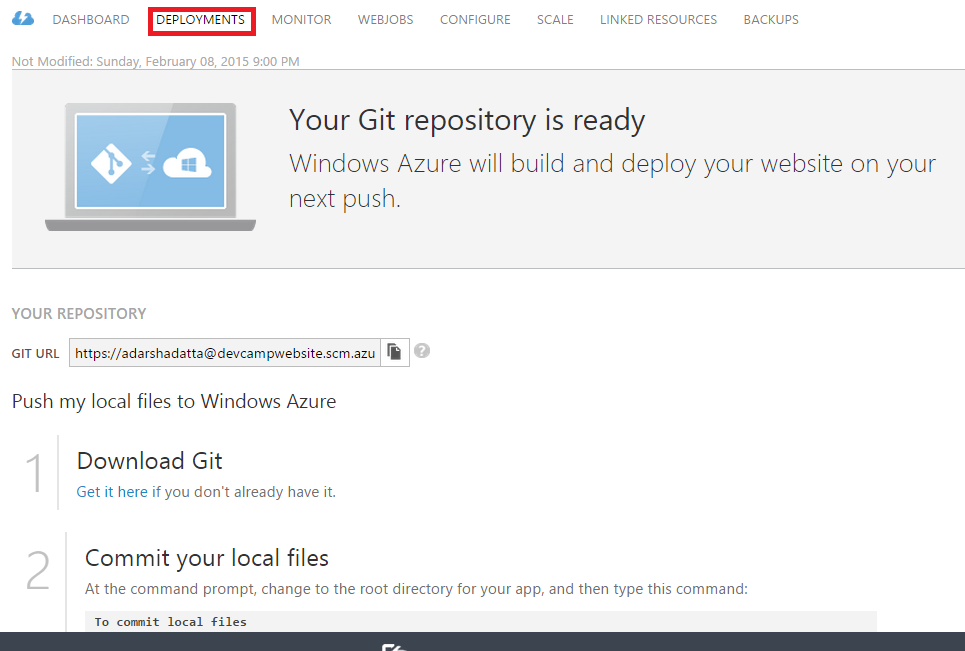
1. Login to manage.windowsazure.com. Click on New-> Websites ->Custom Create. Enter Site Url, Webhosting plans, Regions, create a New DB if not already created one. Click on Create. Once created you can enter the website, click on Configure and check the support for various programming platforms that are available:



1. Click on the Quick start screen, go to option 3 “Setup deployment from source control”. Then click on “Local Git Repository”.

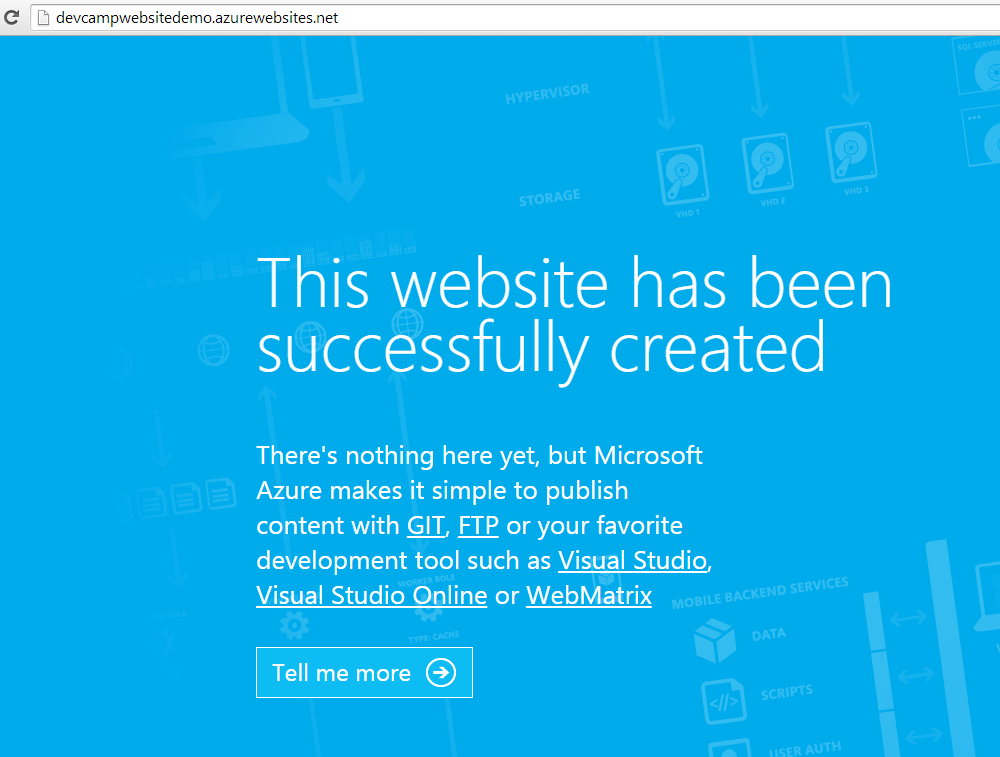


1. You will be redirected to the following page “Deployment”:

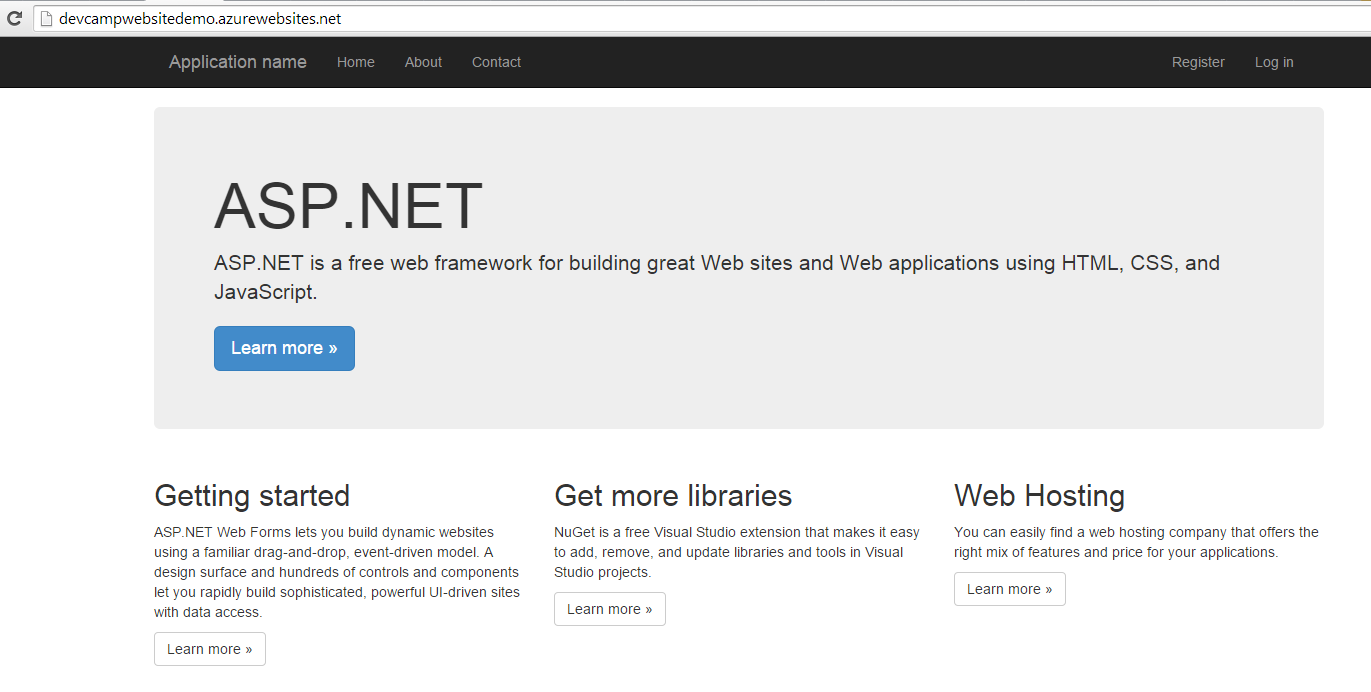


Step 4: Deploy using Git, publish website on Azure

1. Now if we click on the created website, this is what we see:

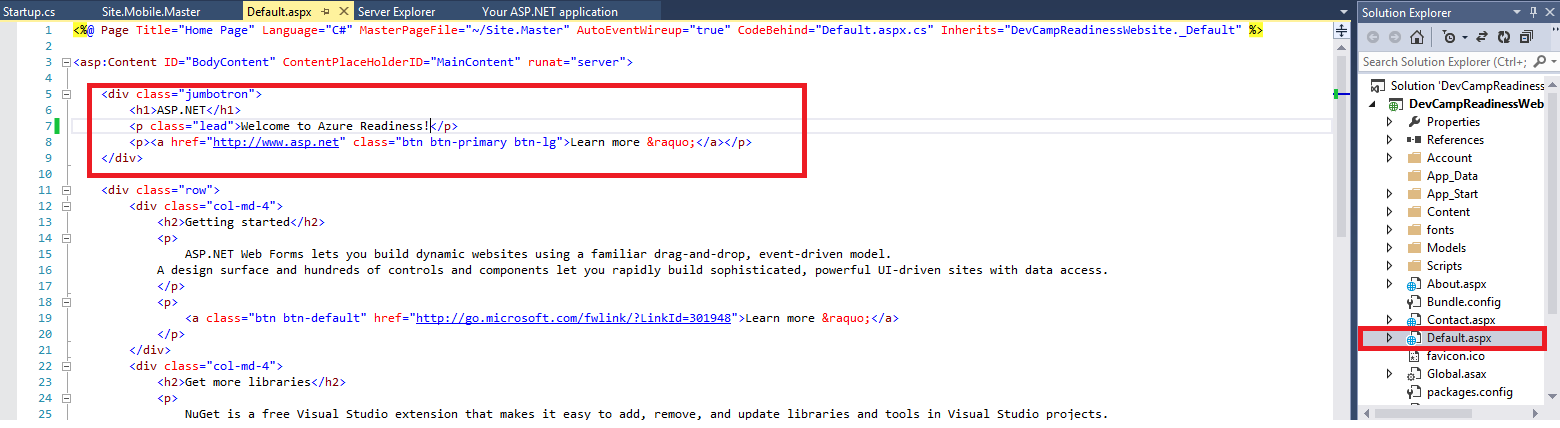


1. The steps to publish using Git is mentioned in Step 3 of the deployment page on the management portal as shown above. Copy/paste these commands on your command prompt and execute them.
2. Now, the website is published and if we refresh the above page, we will be able to see the published website:



Step 5: Modify code and redeploy

1. In the project solution in Visual Studio, go to Default.aspx, modify “ASP.Net is a free web framework….. “ to “Welcome to Azure Readiness”:

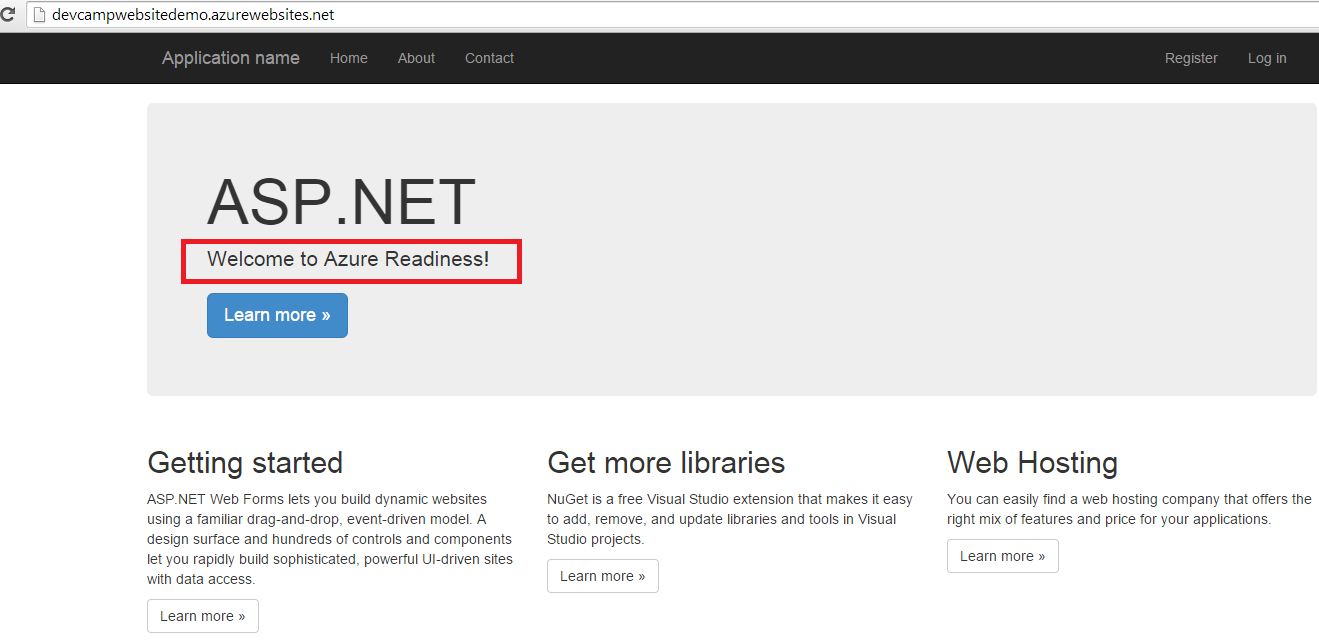


1. Now, let us redeploy this solution as before. Go to command prompt and execute the following:

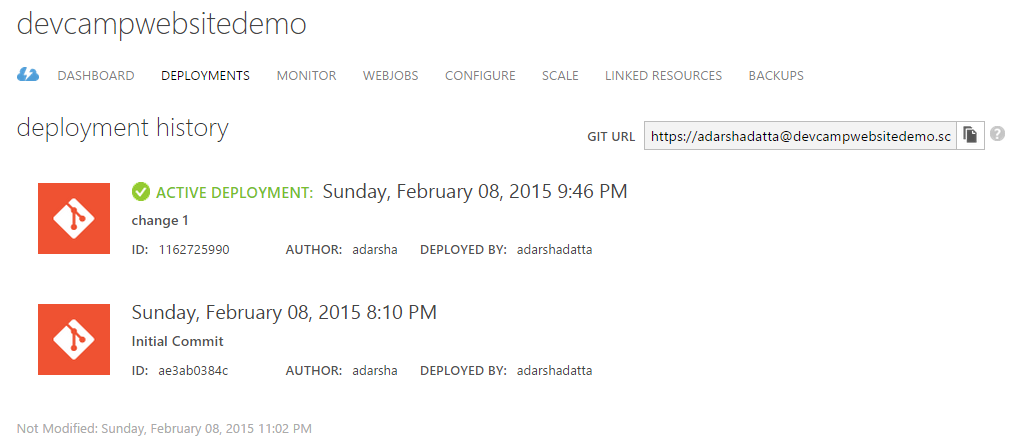
Git commit –am “version 2”

Git push azure master

Once deployed, go to the browser and refresh the previous deployment. Notice the change:

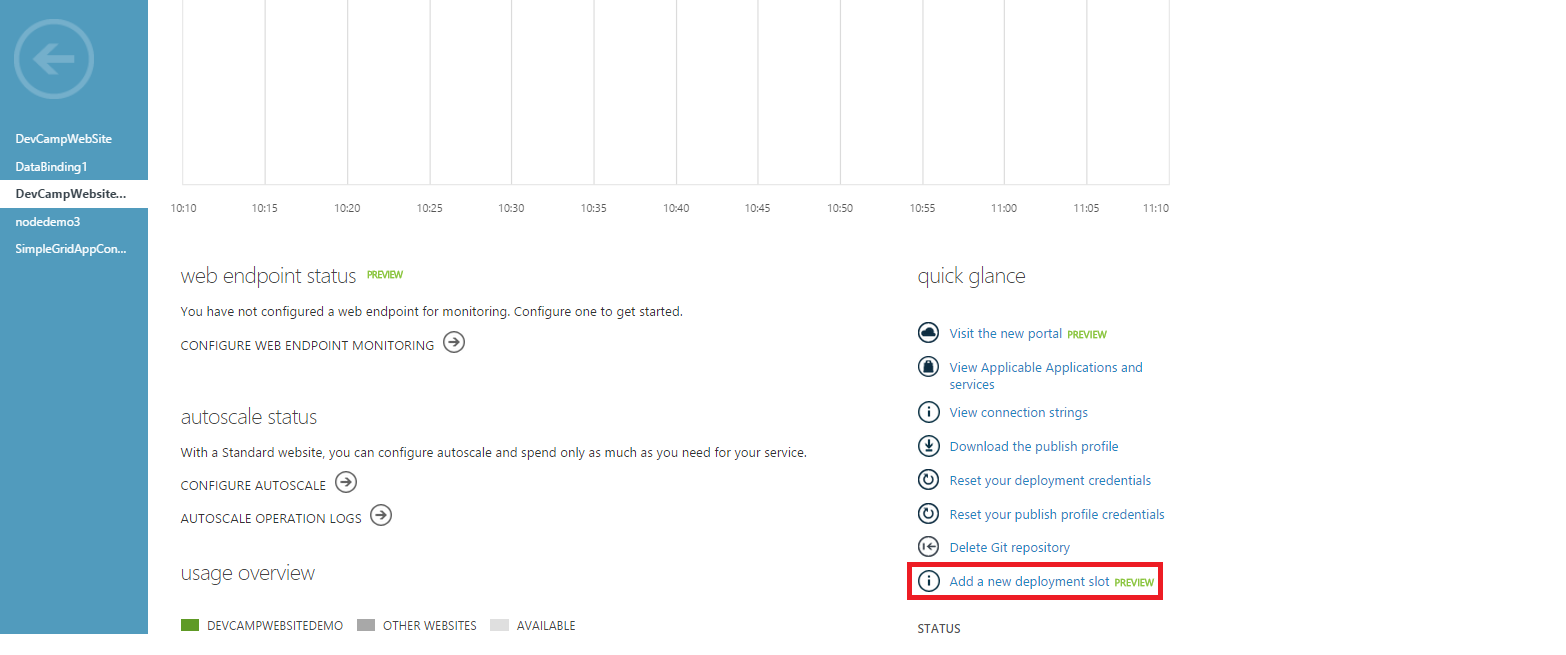


You can have a look at the deployment history too:

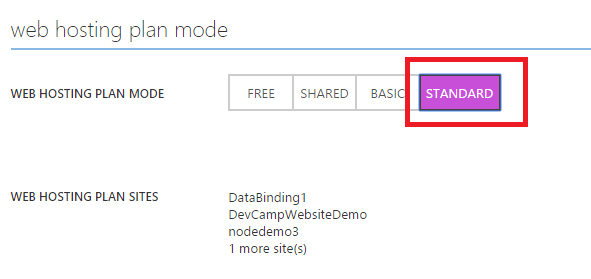


Step 6: Staging and Deployment Slots

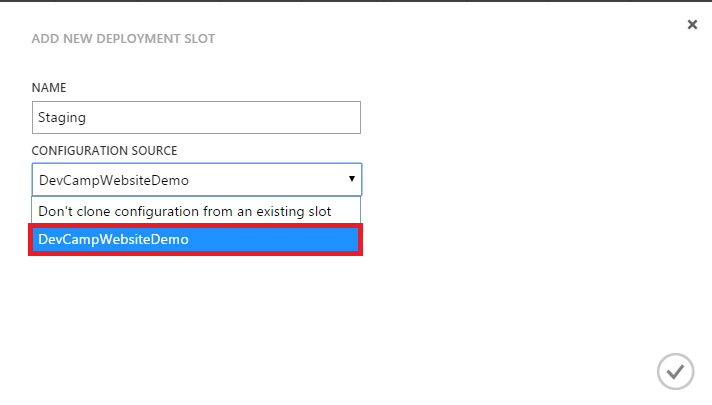
1. This option is only available with Standard mode, hence if your website is in a web hosting plan that is in ‘Free/Basic’ mode, you will need to upgrade it.
2. Click on Dashboard, and then “Add a new deployment slot”



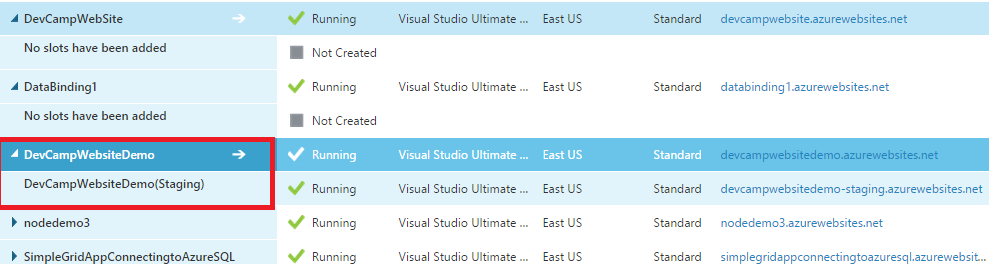
1. If you are not in Standard mode, it will redirect you to upgrade. Click on Standard, and then save. Then you could click on Add again:



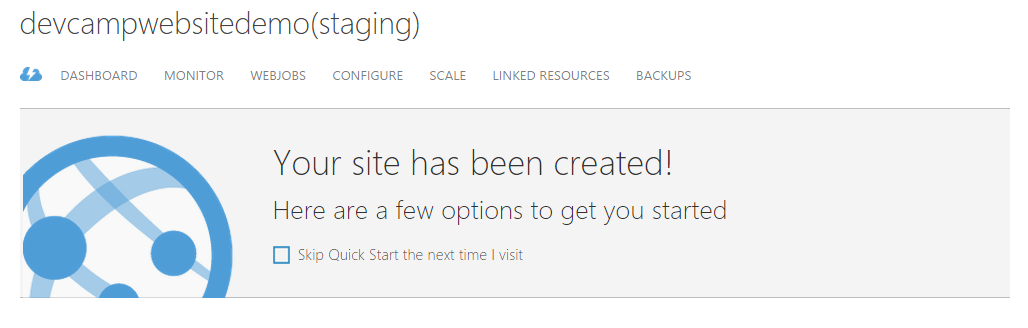
1. Enter a deployment slot name and select the existing website that you want to clone:



1. Now if you go back to the list of websites, you can see the additional deployment slot has been added if you expand:

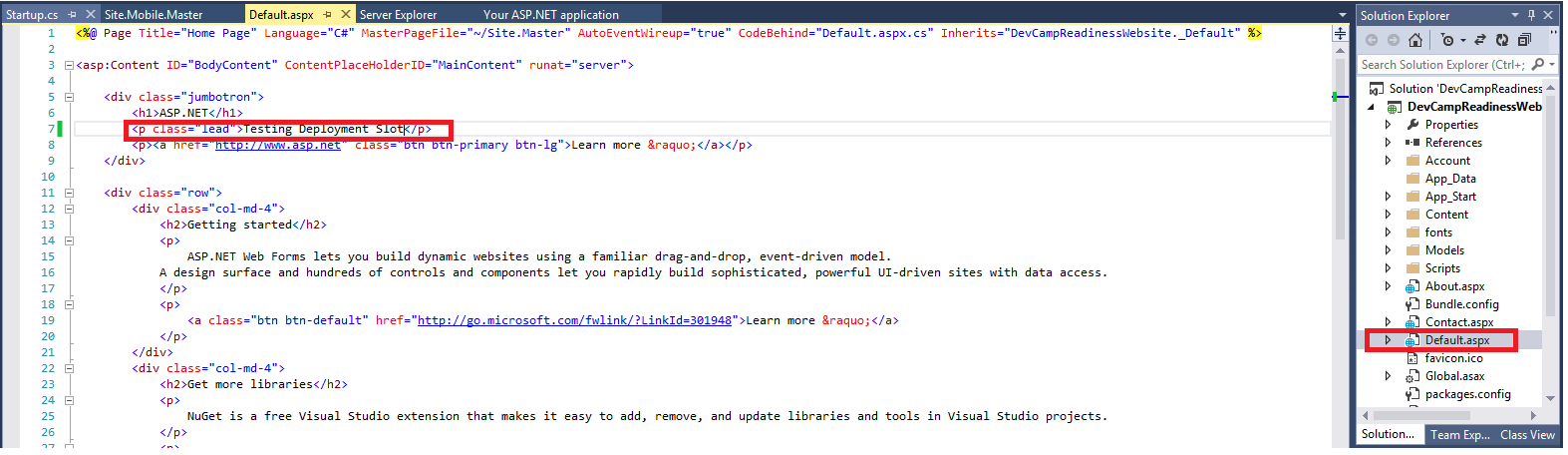


1. Now if we click on the Deployment Slot website, we will be directed to a similar looking dashboard as before. Difference being, the deployment slot name will appear in brackets to remind us:



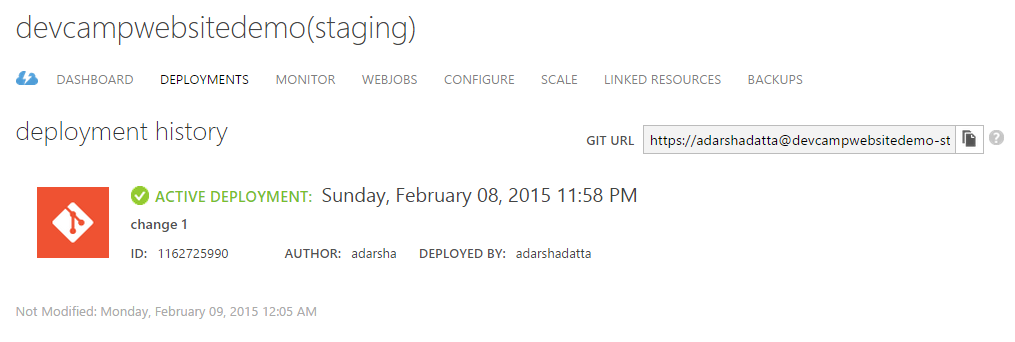
1. Now before making changes to the production slot, let us make changes and deploy to the Staging slot:

Make changes to default.aspx as before in Visual Studio, but this time change to “Testing Deployment Slot”

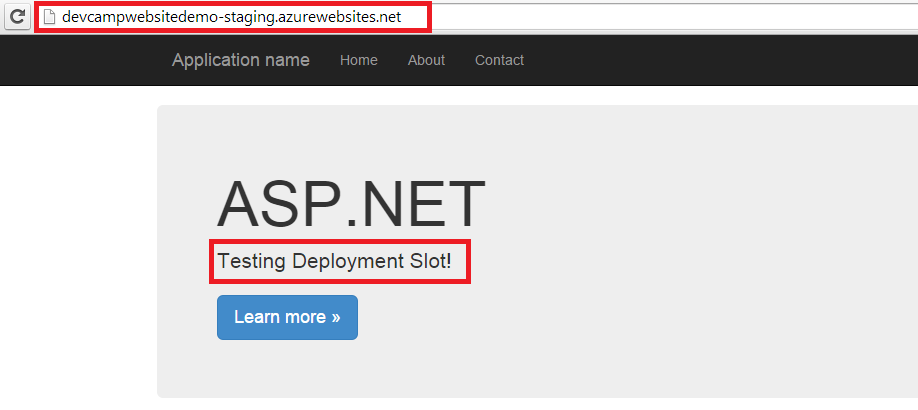


1. Now go back to the Deployment tab of the Staging website in the Azure portal. Go to Step 3 and copy paste the commands on command prompt. (Note: you will need to change the name of the deployment slot created by git from “azure” to “azure\_staging” (or as you may like).

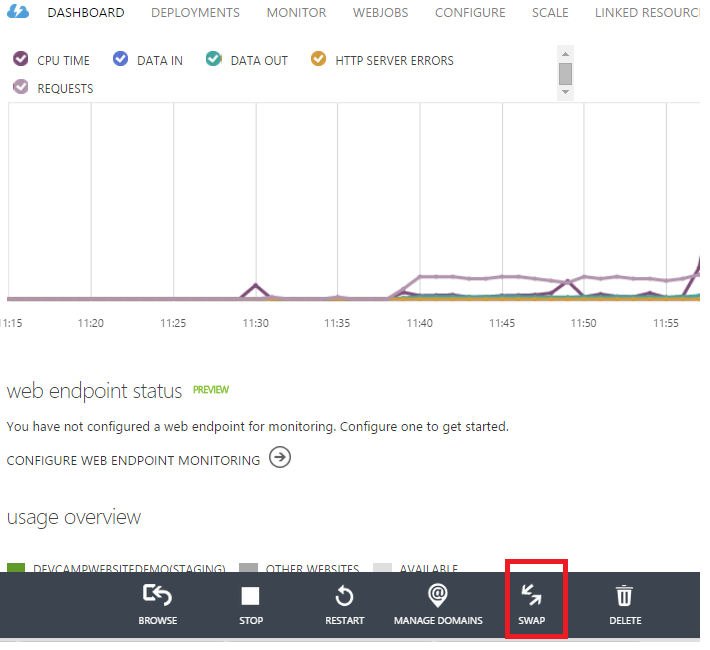
Newly deployed website:



The website in the staging slot looks like:



1. The changed content is in the Staging slot now, let us click on the Swap button and swap the Staging with the Production site:



1. Once deployed and refreshed you can see the change in the production site:

