**Continuous integration in Azure using Kudu**

WebApp is one of the core components in Azure Ecosystem. Azure allows you to deploy a web app built using different technologies like ASP.NET, Node, PHP, HTML + JS etc. These sites can be deployed using different methods like FTP, Web deployment using Visual Studio, using Team Foundation Server [TFS] or using Kudu! You might have heard about these deployment techniques earlier except Kudu, so let’s first understand what Kudu is all about?

**Kudu**

Azure supports continuous integration of Web Apps from source code control and repository tools like BitBucket, CodePlex, Dropbox, Git, GitHub, Mercurial, and TFS. Kudu is the execution engine behind Azure Web Apps deployment using Git. It is an [open source](https://github.com/projectkudu/kudu) project hosted on GitHub that can also run outside of Azure.

In this article we will cover how to deploy ASP.NET Website using Git, setup continuous deployment, push changes to Web App using Git and monitor continuous integration.

**Pre-requisites**

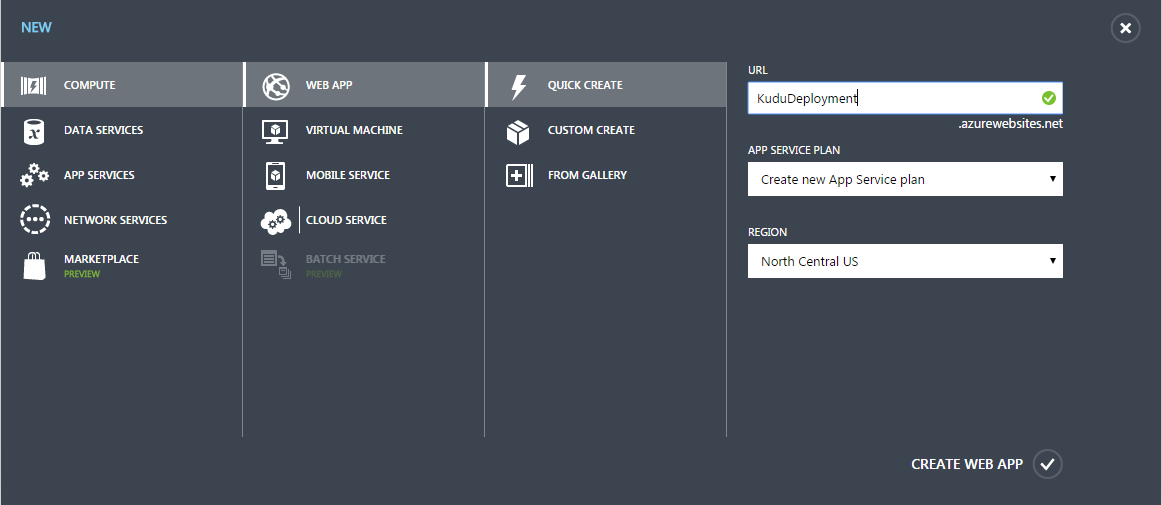
We have lot of things to cover in this article and I assume you’re familiar with few technical aspects and have development environment ready. In a nutshell you would need -

* Visual Studio 2012 / 2013 or above. [Visual Studio Community Edition](https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx) which is free will do as well.
* Active [GitHub](https://github.com/) account. If you want to create public repository, it’s free.
* Git client installed and basic understanding of git commands. Here are the [installation instructions](http://git-scm.com/book/en/v2/Getting-Started-Installing-Git).
* Active Azure subscription – Go for 30 days free [trial version](http://azure.microsoft.com/en-us/pricing/free-trial/) if you don’t have one.
* Basic understanding of ASP.NET Website.

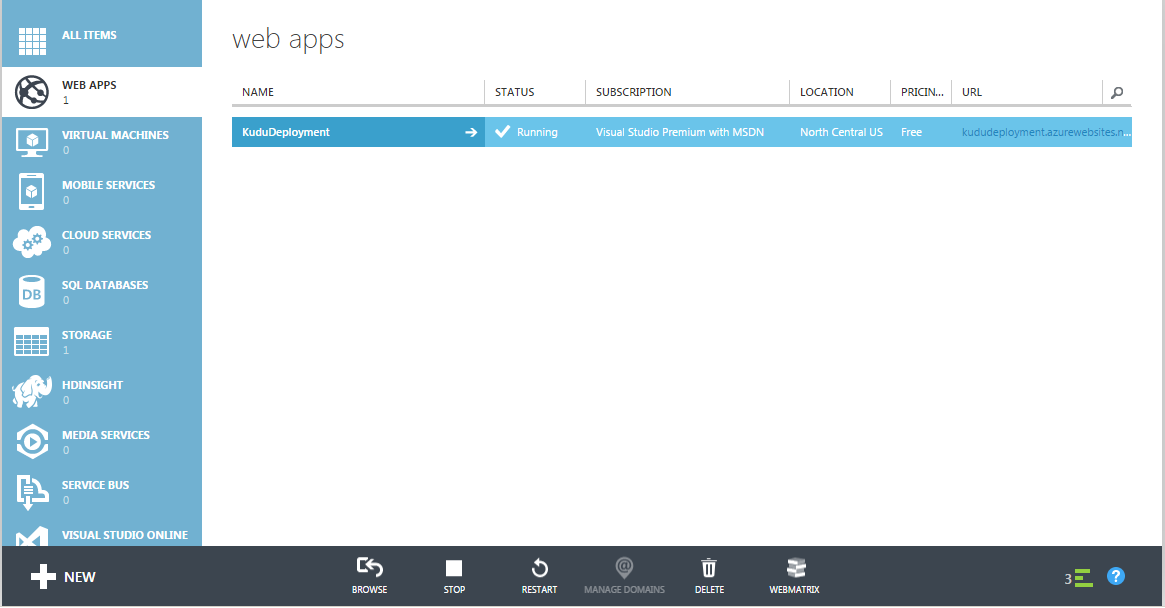
Assuming you have all the software installed and valid Azure subscription, you are all set to create Azure WebApp now.

**Creating Empty Azure WebApp**

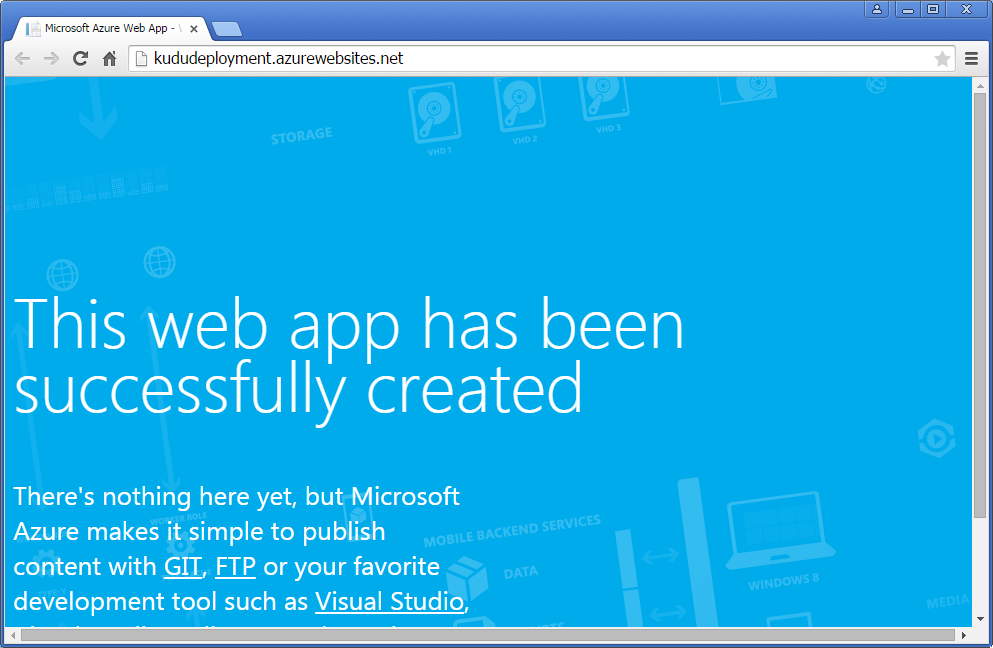
Azure dashboard provides a wizard like interface to create Azure WebApp without a single line of code! It allows you to create an empty WebApp or use different templates available in the template gallery. In this step, we will simply create an empty WebApp in Azure portal.



Once you provide all the details like WebApp URL, App Service Plan and Region, click on Create Web App button. Azure would take around 10-15 seconds to create an empty WebApp as shown in below screenshot.

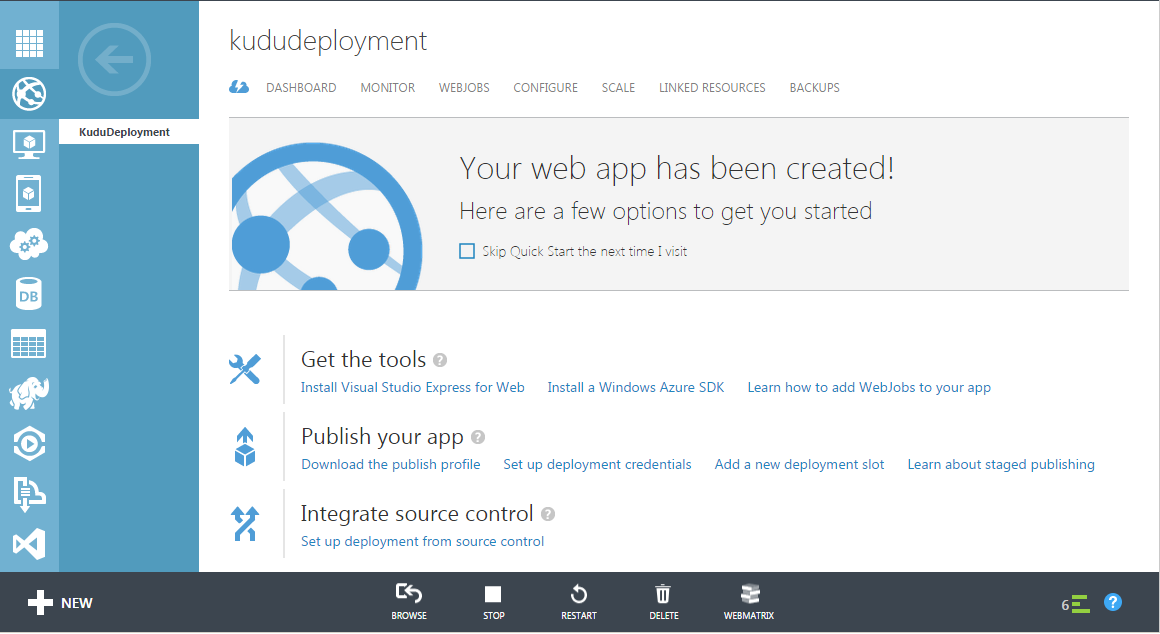


You can browse the website by navigating to <http://KuduDeployment.AzureWebsites.Net>



As the welcome page of the website says, it’s an empty website and we can use different methods like GIT, FTP or Visual Studio to publish the contents to it. In this article, I will stick to my favorite editor Visual Studio.

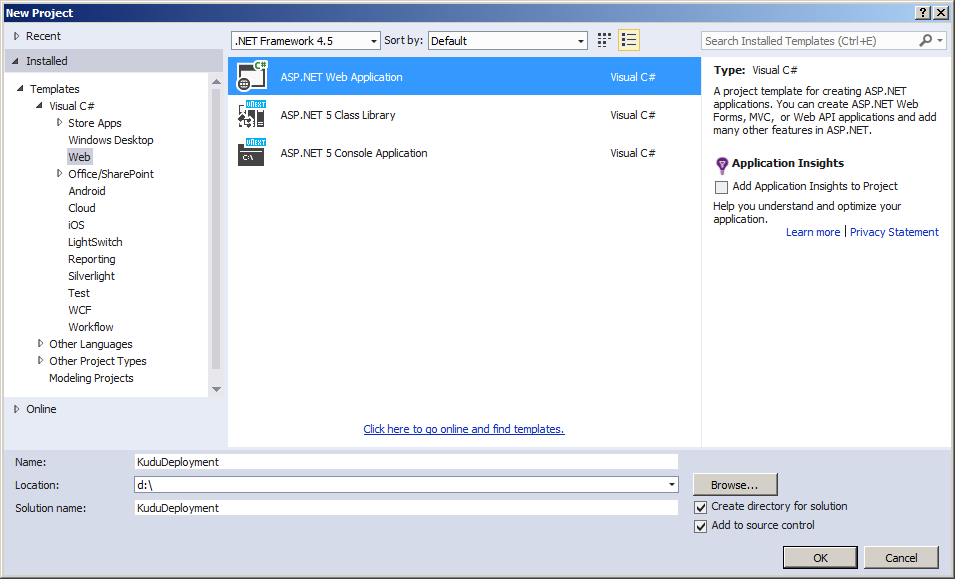
The Dashboard tab of the website contains many important features like ‘Integrate with source control’, ‘Download publish profile’, ‘Deployment slots’ etc. Click on ‘Download Publish Profile’ link, which will download *KuduDeployment.azurewebsites.net.PublishSettings* file. This file contains the publish profile information which we will need deploy ASP.NET MVC app at later stage.



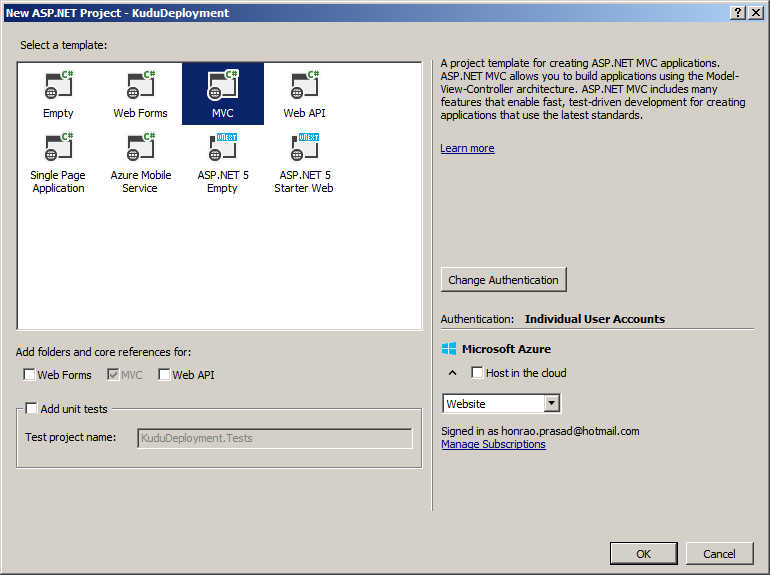
We will use ‘Integrate source control’ to link Azure WebSite to GitHub repository at later stage. We are all set now to create an ASP.NET App. You can use different types of Apps like Web Form, MVC, SPA and Mobile etc. I personally prefer MVC project, so we will stick to it in this article.

**Creating ASP.NET MVC App**

This is the easiest step in the entire article and I am not going to spend much time on it. Select ASP.NET Web Application option in New Project dialog box as shown below. Make sure that you check ‘Add to source control’ checkbox. This will help us to push the contents to GitHub repository at later stage.

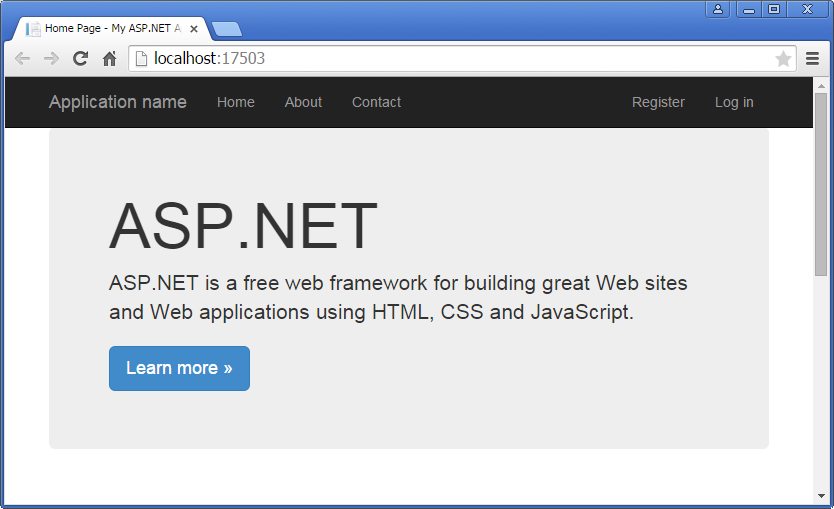


Select MVC template as shown below and hit OK.



Now, you can directly host the MVC App in the cloud by selecting ‘Host in the cloud’ checkbox, however this option somehow throws weird error messages, so I prefer to publish the app from Visual Studio.

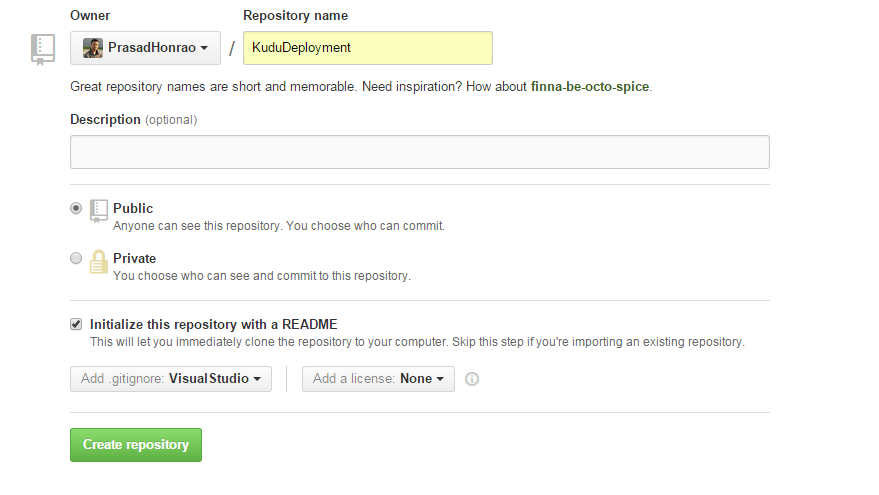
Run the web app just to ensure that it got created successfully.



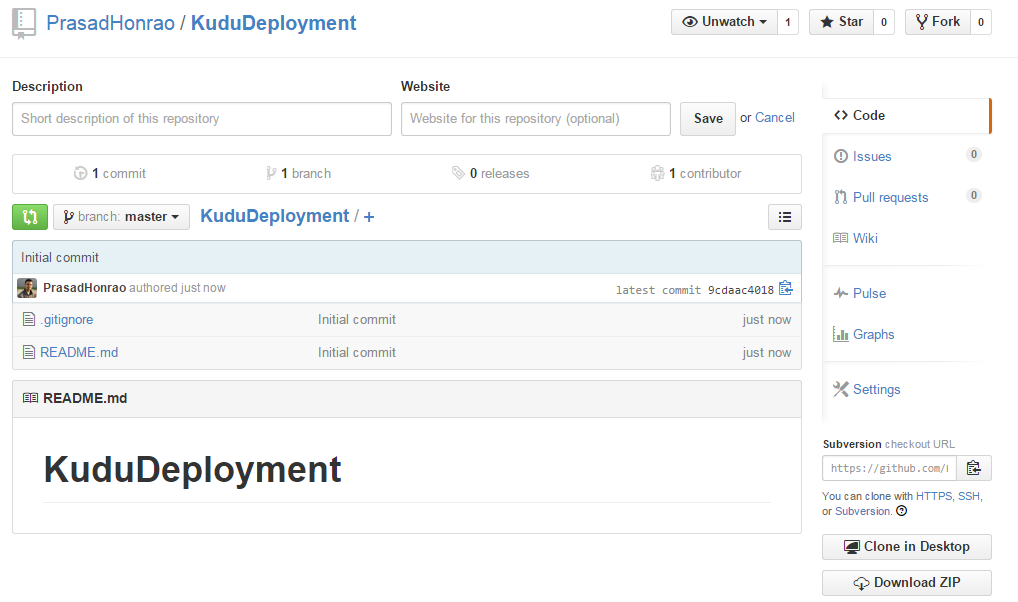
**Creating GitHub Repository**

GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. Unlike Git, which is strictly a command-line tool, GitHub provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

Login to GitHub portal and create a *KuduDeployment* repository as shown in below screenshot.



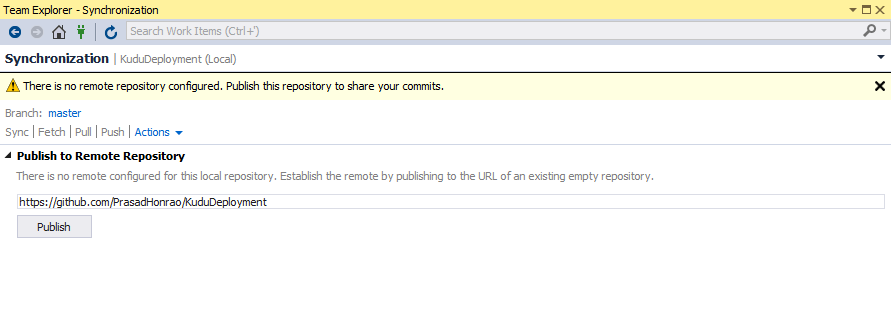
After you click on ‘Create repository’ button, you should get screenshot as shown below



You are now all set to push the contents from ASP.NET MVC app to GitHub repository.

**Push contents from Visual Studio to GitHub repository**

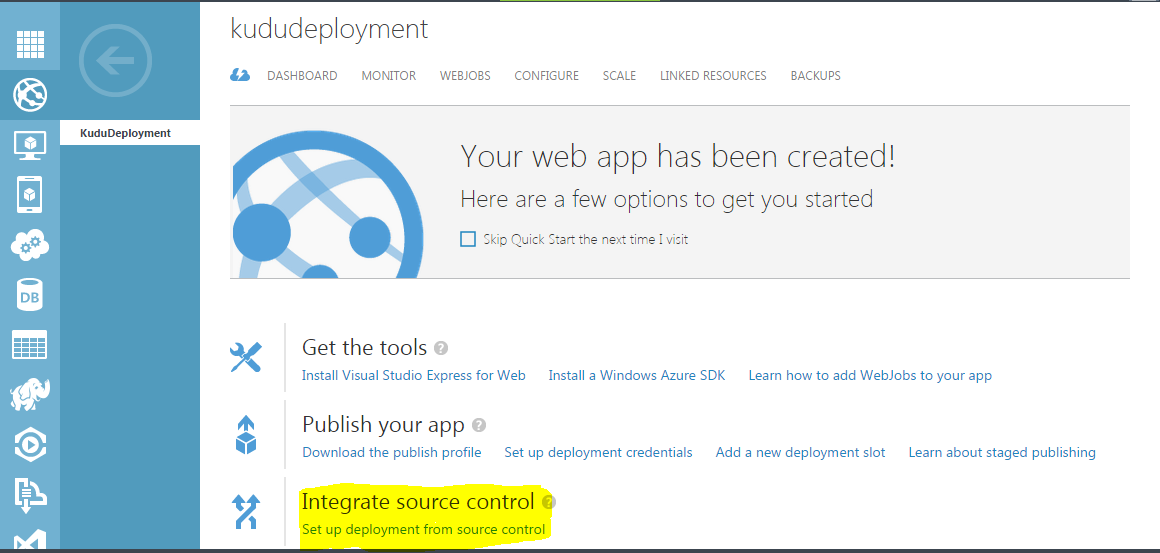
Assuming you have Visual Studio Git Client installed on your machine, navigate to Team Explorer and commit all the contents to local git repository. Next you need to provide GitHub repository URL to synchronize contents between local and remote repositories.



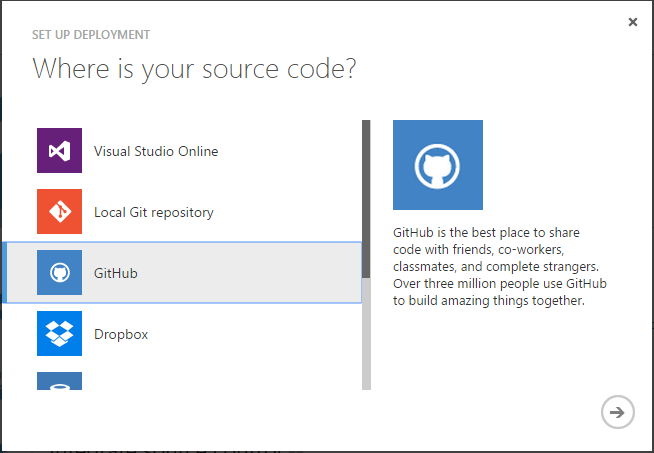
Now you have everything setup from source code perspective. Next you need to integrate GitHub repository [source control] with Azure Website.

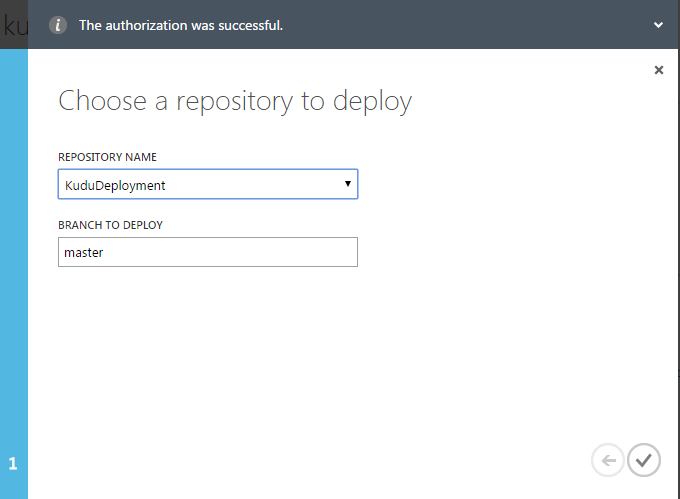
**Integrate source control in Azure portal**

Navigate to KuduDeployment dashboard in Azure portal and click on setup deployment from source control link.

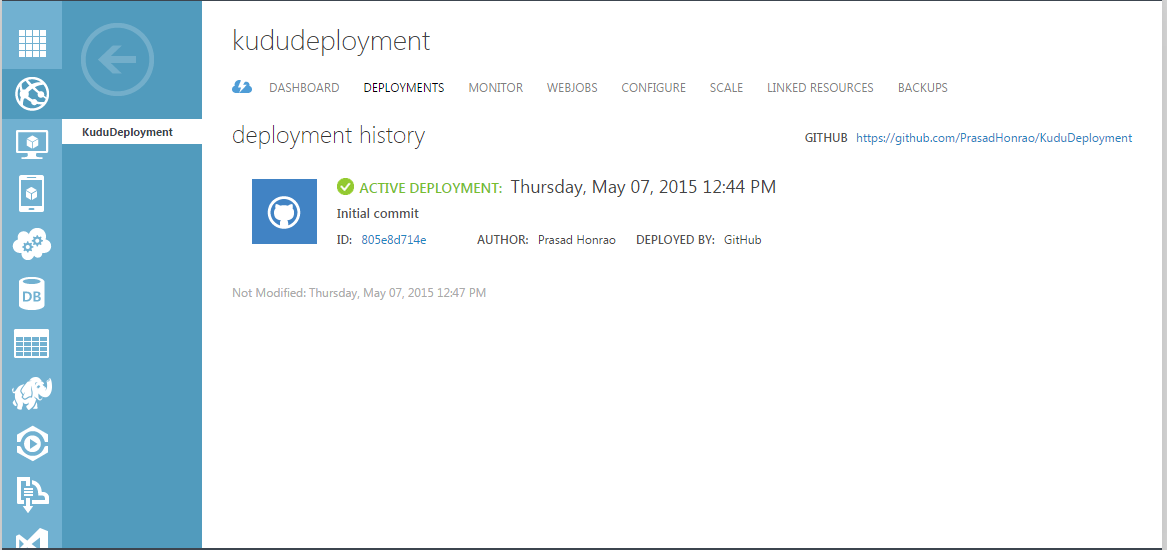


It will open up source code integration wizard where you need to provide the location of the GitHub repository. The wizard will also ask you for authorization to access the repository contents from your GitHub account.

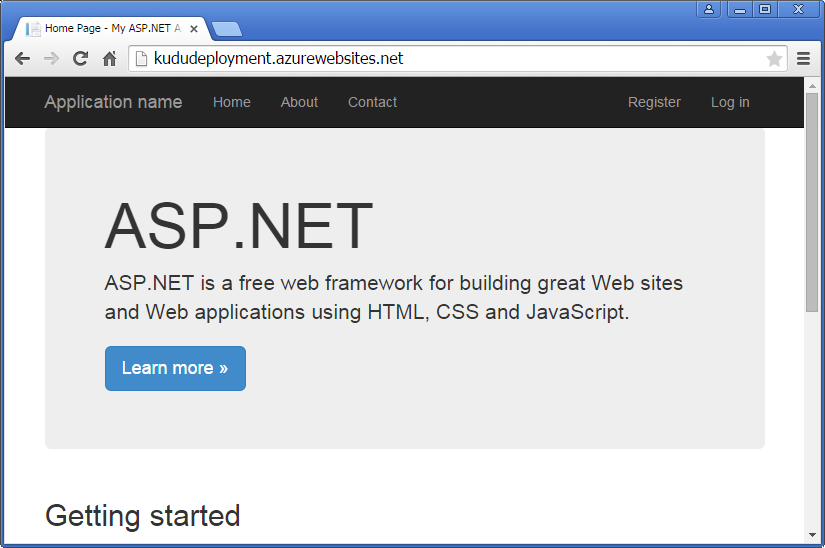




Complete all the required steps in the wizard. The website dashboard will display additional tab – DEPLOYMENTS as shown below. It will also deploy the source code from ASP.NET MVC app to Azure Website. As you can see that it shows GitHub repository URL, author name and comment – Initial commit which I had provided from Visual Studio during commit process.

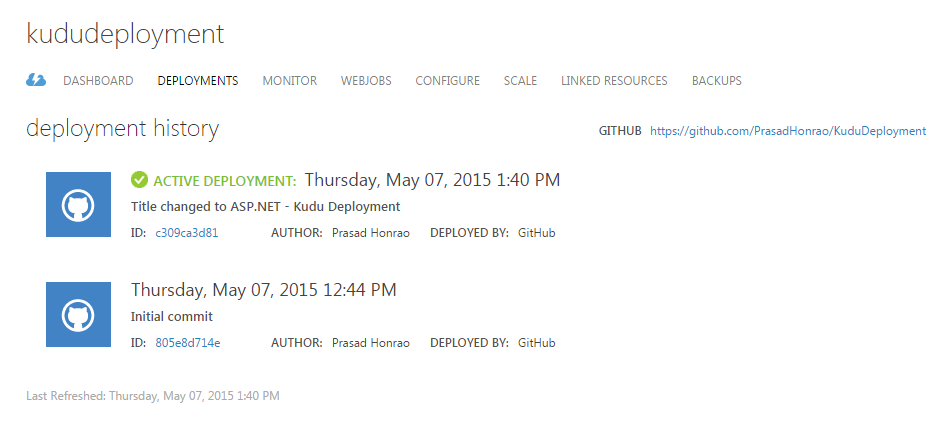


Navigate to URL <http://KuduDeployment.AzureWebSites.Net> to view the ASP.NET MVC app contents.

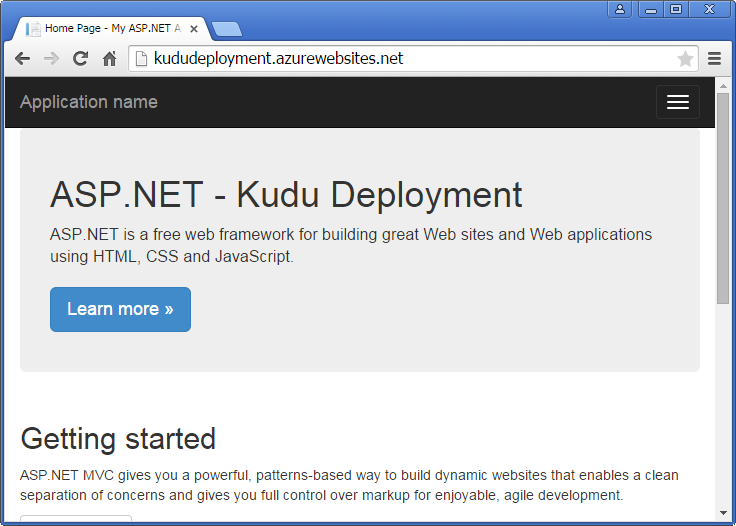


Now anytime you make any changes to ASP.NET MVC application and push it to GitHub repository, it gets deployed to Azure automatically [Kudu behind the scenes]. You can monitor it under the DEPLOYMENTS tab in Azure portal.

As a last step, let’s update the title in Index.cshtml page in MVC app, push it to Git and monitor the changes in Azure portal.



Navigate to URL <http://KuduDeployment.AzureWebSites.Net> to view updated title.



That’s all. We have successfully setup continuous integration between GitHub repository and Azure Website. I hope that was helpful.

Thanks for reading.