

Source: C# Corner (www.c-sharpcorner.com)

PRINT

Article



Add Application Insight In Existing ASP.NET Web Application

By **Abdul Rasheed Feroz Khan** on **Jun 02 2016**

Introduction

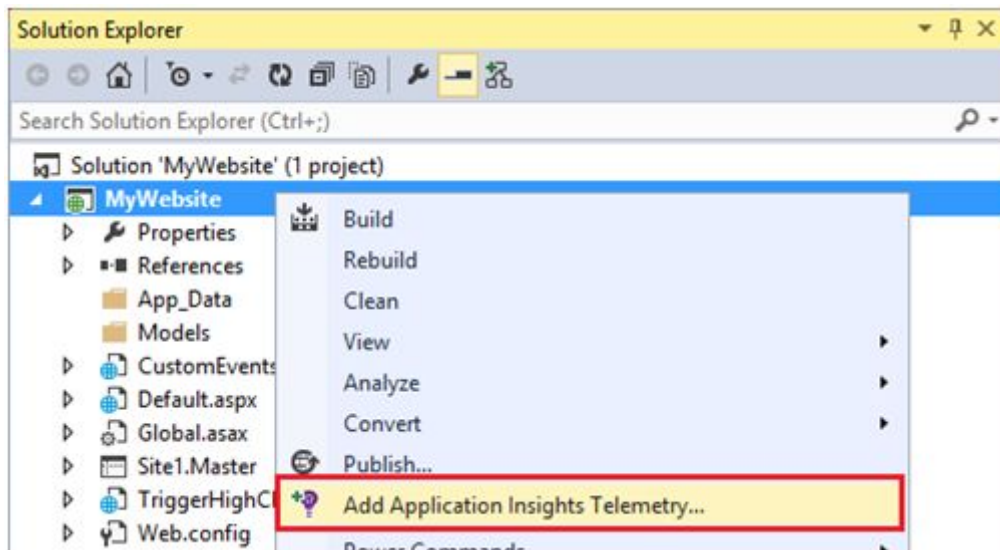
Do you have an existing Web Application and want to use Application Insights? Just get on to Visual Studio -- Visual Studio 2013 now has built-in capability to add Application Insights to your application.

Follow the steps given below now to add Application Insight in the existing ASP .NET Application:

Step 1: Open the ASP.NET Web Application in Visual studio 2013.

Step 2: This is an existing ASP.NET Web Application that you already have and you want to add “**Application Insights**” to capture telemetry data. Hence, go to the “**Solution Explorer**” and right click on project “**MyWebsite**” and you will see “Add Application Insights Telemetry”.

You need “**Visual Studio 2013 Update 3**” or more to get this option in the IDE.

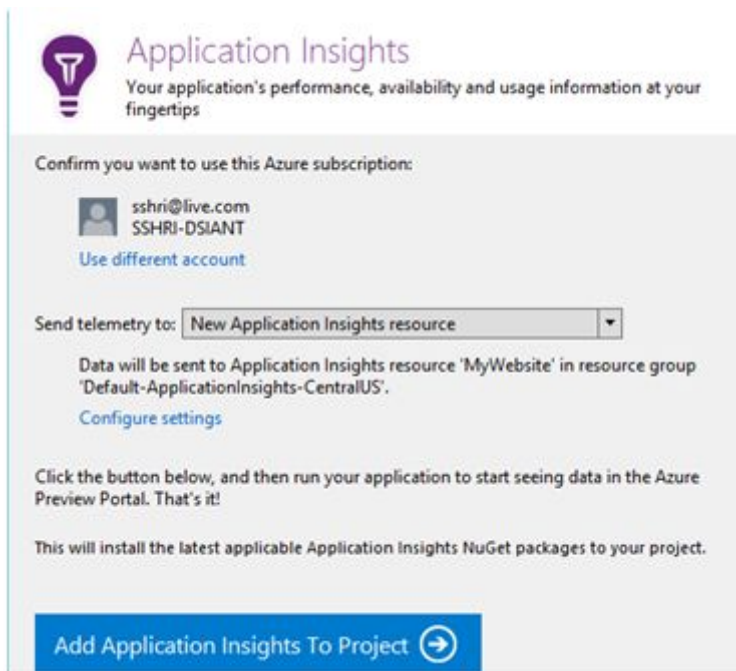


Step 3: Click “**Add Application Insights Telemetry**” and it will show option to “Sign-in” to Azure account to proceed further.

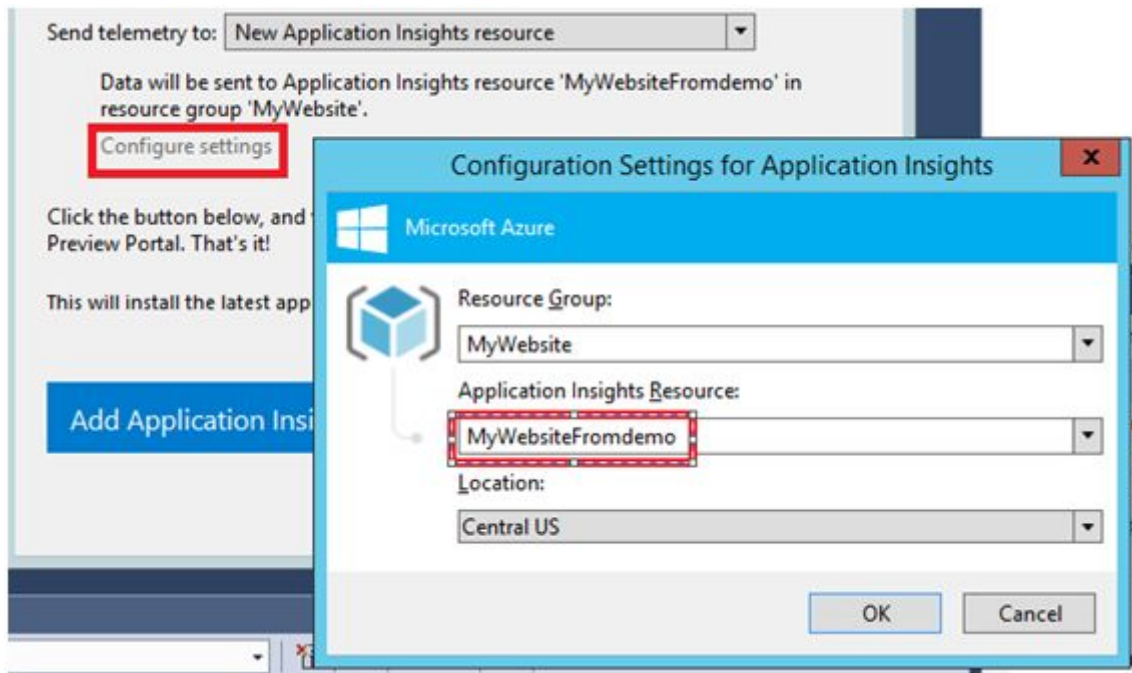
Click “**Sign-In with your Azure Account to get started**”. You will get the Sign-In page:



After successfully signing in, you will get the options to either use an existing “**Application Insight**” resource or create a new one now, as shown below:

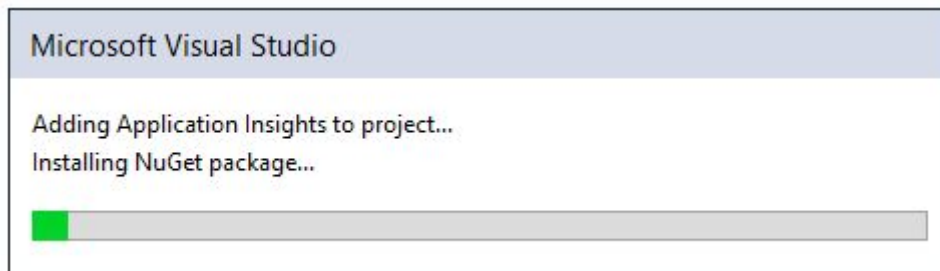


You can also click “**Configure settings**” to enter specific values:



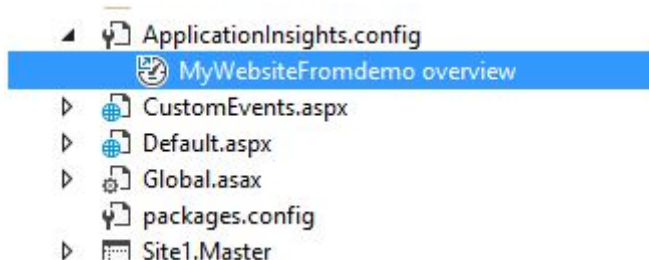
After entering all the values, click the blue button “**Add Application Insights To Project**”.

You will observe that Visual Studio will be adding few “**NuGet Packages**” to the project to enable “**Application Insights**”.



Step 4: After Visual Studio has finished adding “Application Insights” to your project, you will see couple of changes.

First, you will see a configuration file added to your project:



Open the Web.config file and check the modules added to the configuration:

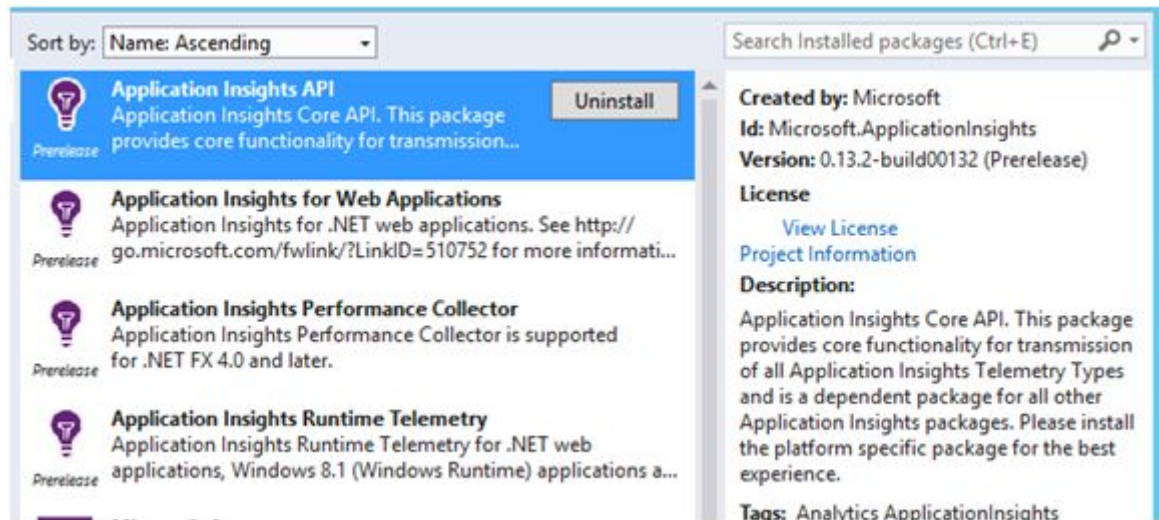
```

1  <httpModules>
2    <add name="ApplicationInsightsWebTracking"
3      type="Microsoft.ApplicationInsights.Extensibility.Web.RequestTracking.WebRequest
4    </httpModules>
5  </system.web>
6  <system.webServer>
7    <validation validateIntegratedModeConfiguration="false"/>
8    <modules>
9      <remove name="ApplicationInsightsWebTracking"/>
10     <add name="ApplicationInsightsWebTracking"
11       type="Microsoft.ApplicationInsights.Extensibility.Web.RequestTracking.WebRequest
12       precondition="managedHandler"/>
13   </modules>
14 </system.webServer>

```

Step 5: Now, right click the project in Solution Explorer and click **"Manager Nuget Packages"**

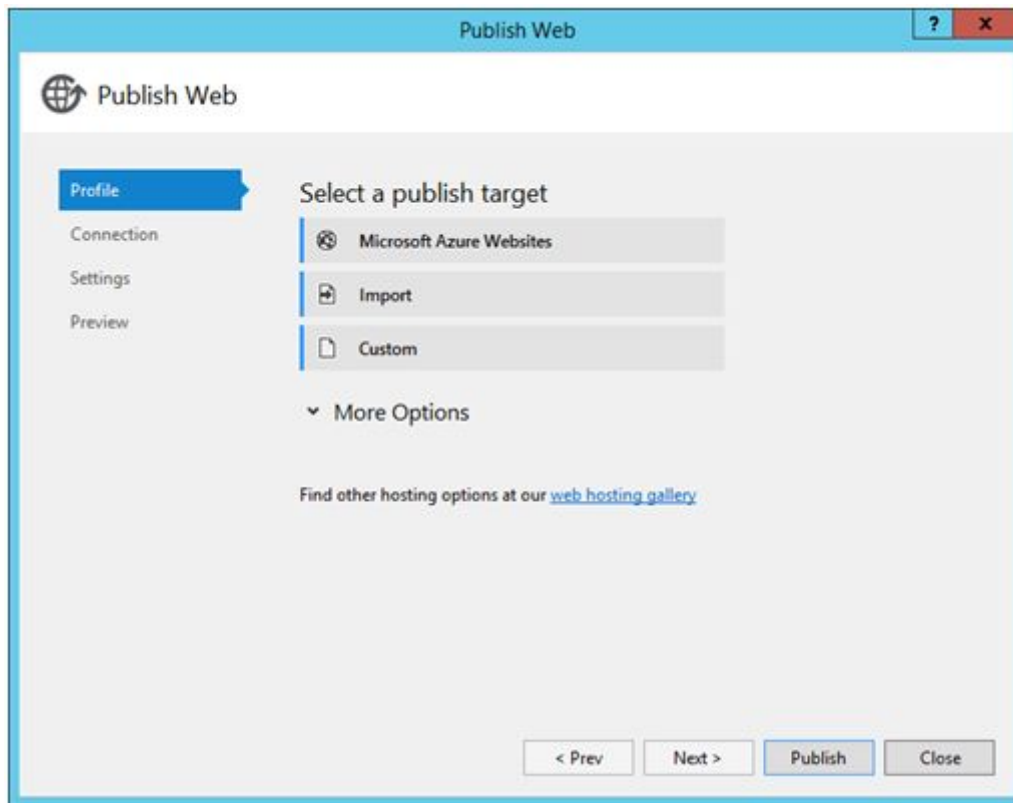
In the **"Installed Packages"** section, you will see the following **"Nuget Packages"** installed to enable your application to use "Application Insights":



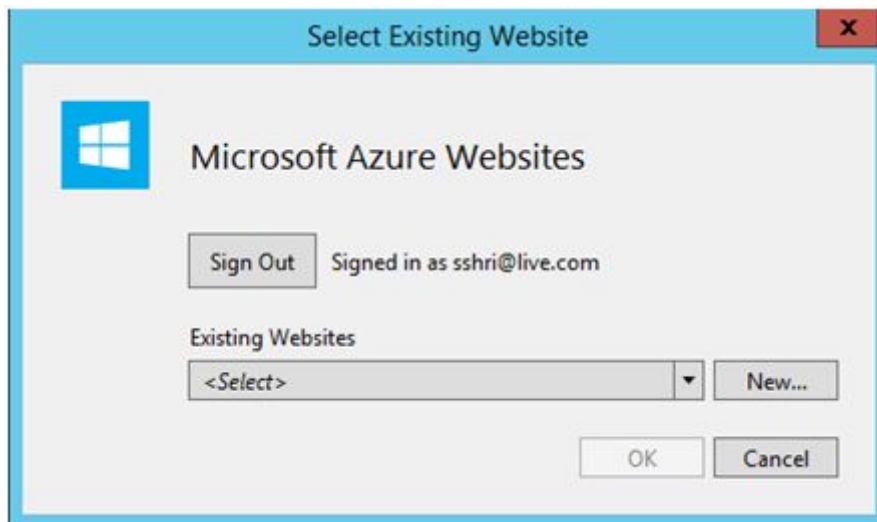
Run the Website locally by pressing **"Ctrl+F5"** to ensure it runs fine, locally.

Now, we will host this Website on MicroSoftAzue Web Sites and test it live on the Cloud.

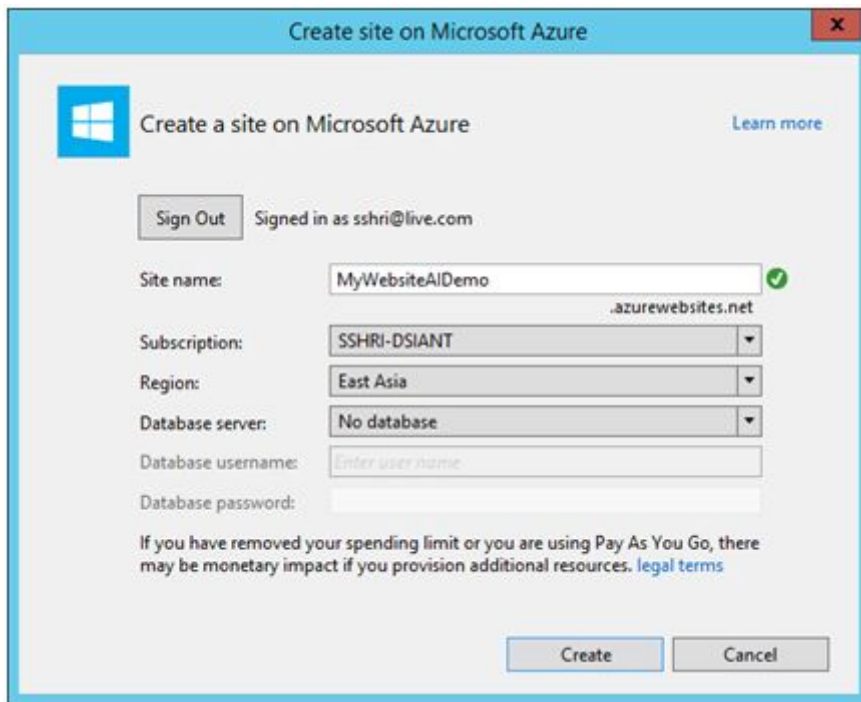
Right click on the **"Website"** project in Solution Explorer again and click **"Publish"**. Select **"Microsoft Azure Websites"**.



This brings up the list of Websites, which are already there and the option to create a new one also comes. Click **"New"**.

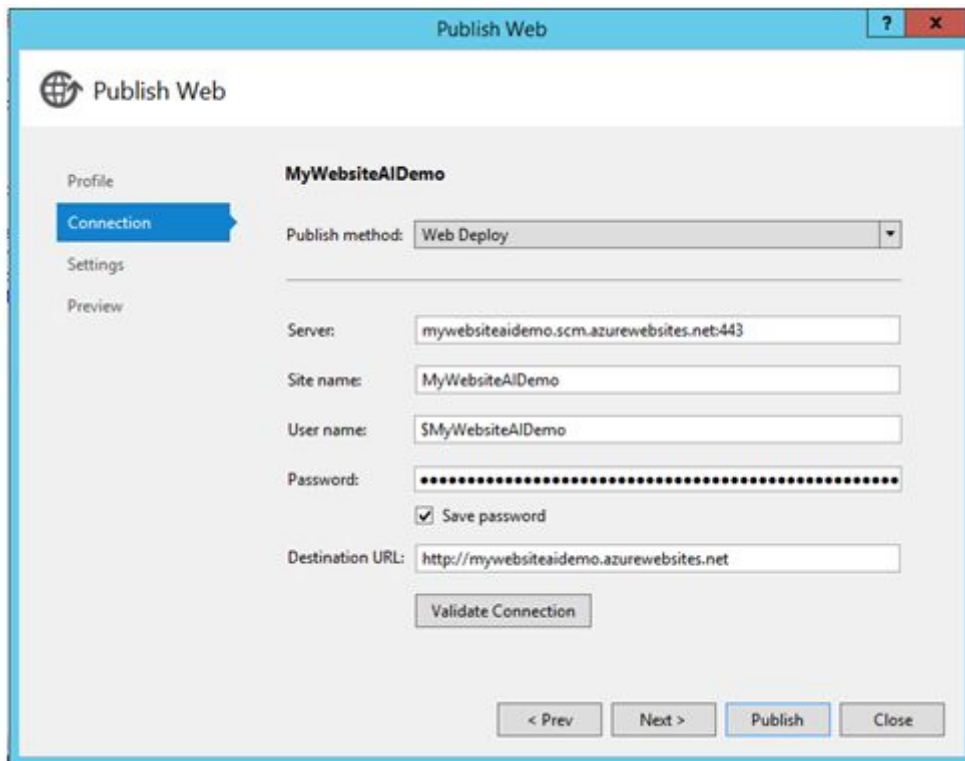


Step 6: Enter the values in the **"Create a Site On Microsoft Azure"** Window and click **Create**:



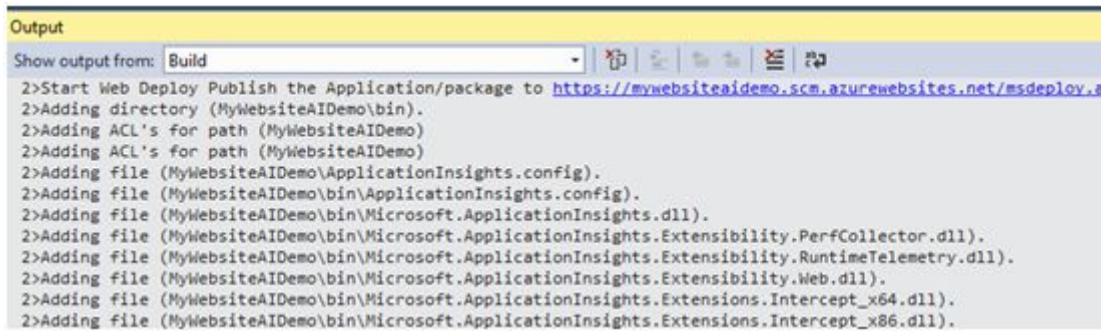
The "Create site on Microsoft Azure" dialog box is shown. It features a Windows logo and a "Learn more" link. A "Sign Out" button and "Signed in as sshri@live.com" text are at the top. The form includes fields for "Site name" (MyWebsiteAIDemo), "Subscription" (SSHRI-DSIANT), "Region" (East Asia), "Database server" (No database), "Database username" (Enter user name), and "Database password". A disclaimer about spending limits is present. "Create" and "Cancel" buttons are at the bottom.

After creating the Website, Visual Studio will automatically download the publishing project and will present you with an option to go ahead with the **"Publish"**:



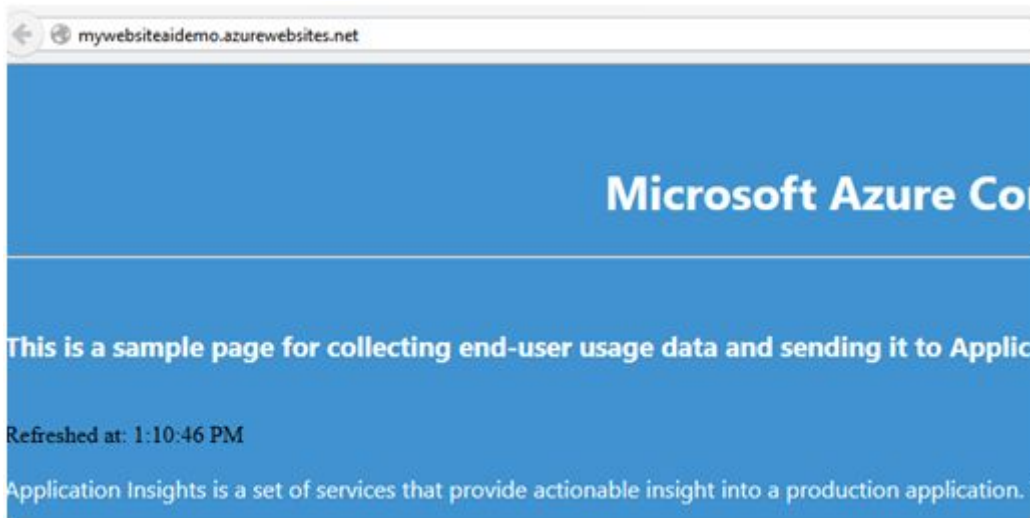
The "Publish Web" dialog box is shown. It has a sidebar with "Profile", "Connection" (selected), "Settings", and "Preview". The main area is titled "MyWebsiteAIDemo" and contains fields for "Publish method" (Web Deploy), "Server" (mywebsiteaidemo.scm.azurewebsites.net:443), "Site name" (MyWebsiteAIDemo), "User name" (\$MyWebsiteAIDemo), "Password" (masked), "Save password" (checked), and "Destination URL" (http://mywebsiteaidemo.azurewebsites.net). A "Validate Connection" button is below the URL. At the bottom are "< Prev", "Next >", "Publish", and "Close" buttons.

Click **"Next"** and then **"Publish"** button to start publishing the Website. You will observe the verbose logging at the bottom of Visual Studio in **"Output"** Window:



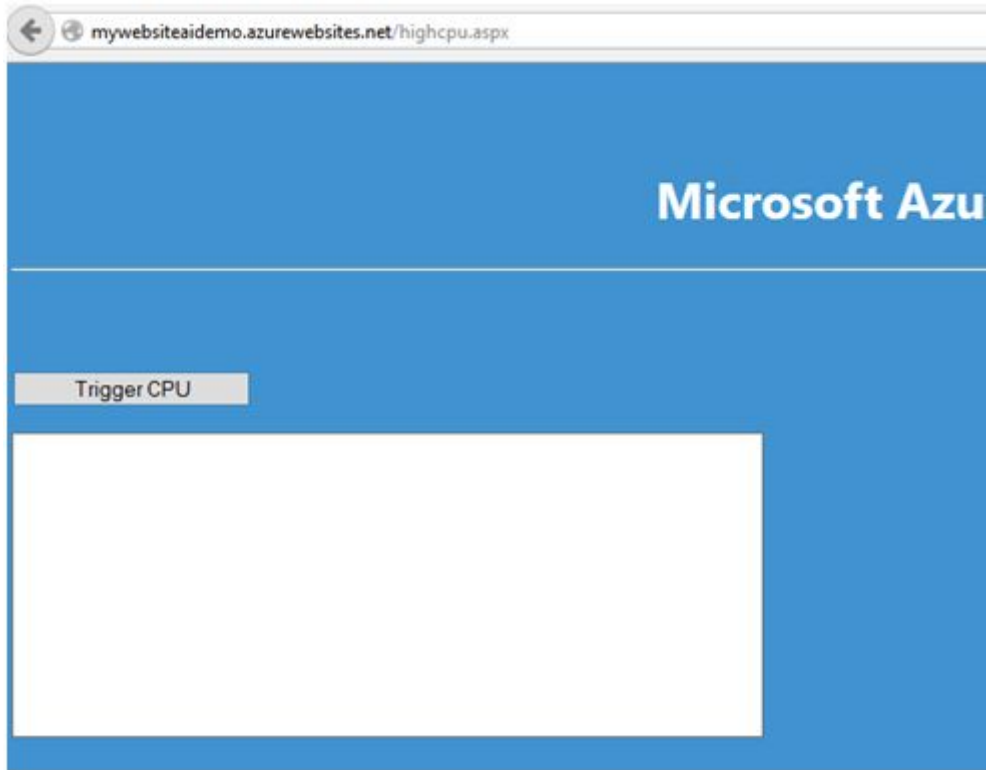
```
Output
Show output from: Build
2>Start Web Deploy Publish the Application/package to https://mywebsiteaidemo.scm.azurewebsites.net/msdeploy.s
2>Adding directory (MyWebsiteAIDemo\bin).
2>Adding ACL's for path (MyWebsiteAIDemo)
2>Adding ACL's for path (MyWebsiteAIDemo)
2>Adding file (MyWebsiteAIDemo\ApplicationInsights.config).
2>Adding file (MyWebsiteAIDemo\bin\ApplicationInsights.config).
2>Adding file (MyWebsiteAIDemo\bin\Microsoft.ApplicationInsights.dll).
2>Adding file (MyWebsiteAIDemo\bin\Microsoft.ApplicationInsights.Extensibility.PerfCollector.dll).
2>Adding file (MyWebsiteAIDemo\bin\Microsoft.ApplicationInsights.Extensibility.RuntimeTelemetry.dll).
2>Adding file (MyWebsiteAIDemo\bin\Microsoft.ApplicationInsights.Extensibility.Web.dll).
2>Adding file (MyWebsiteAIDemo\bin\Microsoft.ApplicationInsights.Extensions.Intercept_x64.dll).
2>Adding file (MyWebsiteAIDemo\bin\Microsoft.ApplicationInsights.Extensions.Intercept_x86.dll).
```

Once the Publish is finished, the Visual Studio will Pop-up a new browse instance with the Website URL.



Now browse the “**Default.aspx**” on your Website hosted on Microsoft Azure. It will keep refreshing itself, after every 2 seconds.

Open another tab and browse “**TriggerCPU.aspx**” page on your Website hosted on Microsoft Azure. Click on “**Trigger CPU**” to log some Server telemetry data on “**Application Insights**”.



Thank you for using C# Corner