

Microsoft

ASP.NET Workshop

By Howard University Microsoft Student Partners

Jeff Beauplan

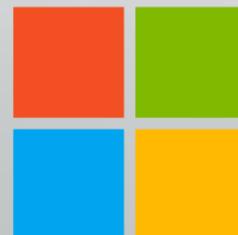
Michelle Brown

Github: https://github.com/JayDougie/Test_website

Website: <http://jeffbeauplan.eastus.cloudapp.azure.com/>

Contents

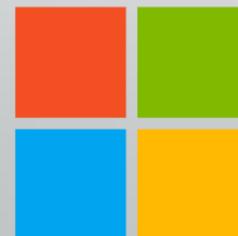
- Setting up the project
- Understanding the structure of the project folder
- Building/customizing the website
- Deployment process via Azure Cloud Services



Microsoft

Things you need

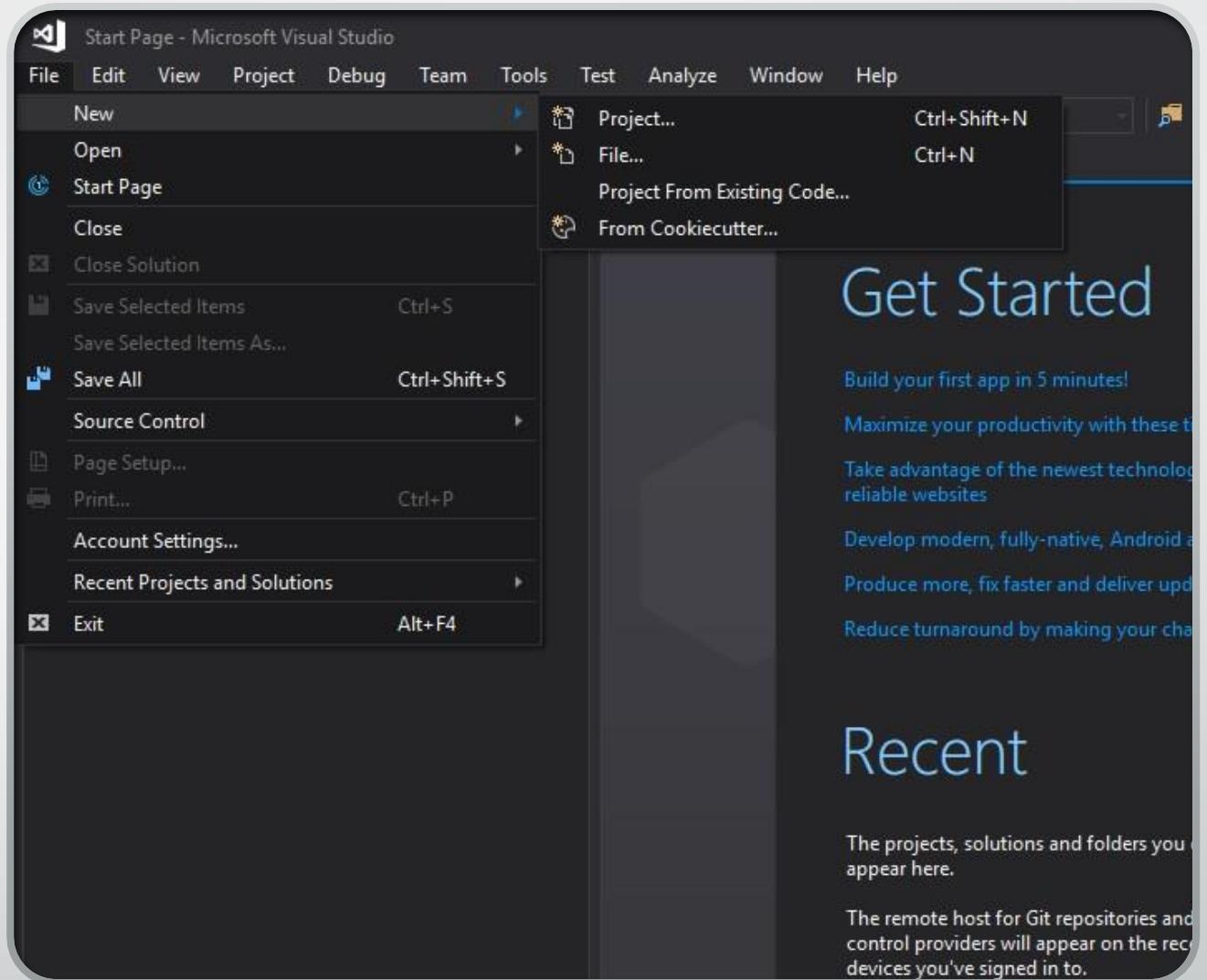
- Microsoft Visual Studio (With ASP.NET Package installed)
- Azure Account (free passes provided)
- Visual Studio Team Services Account (visualstudio.com)



Microsoft

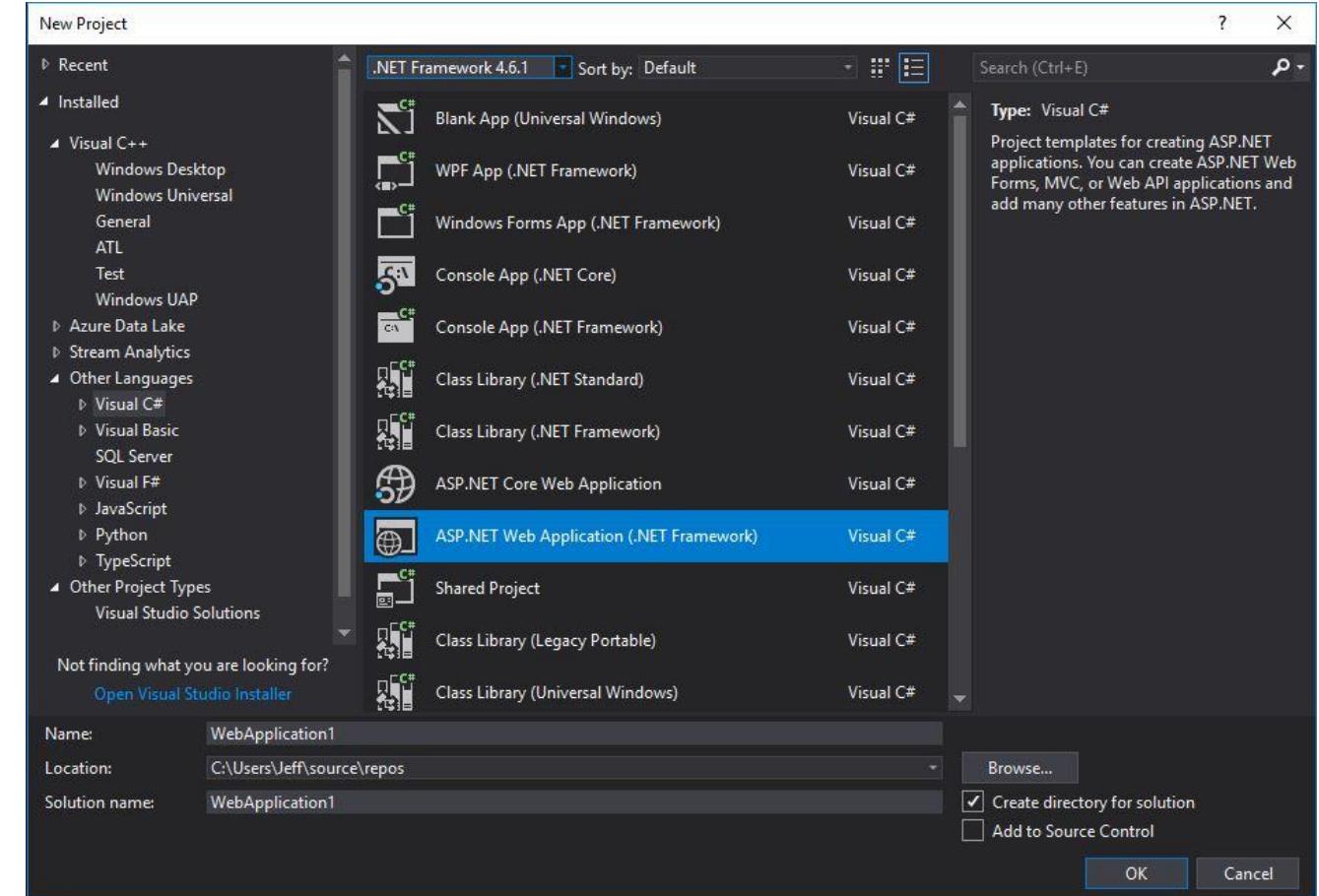
Setting up the Project

- Open Visual Studio
- Click **New**
- Click **Project**

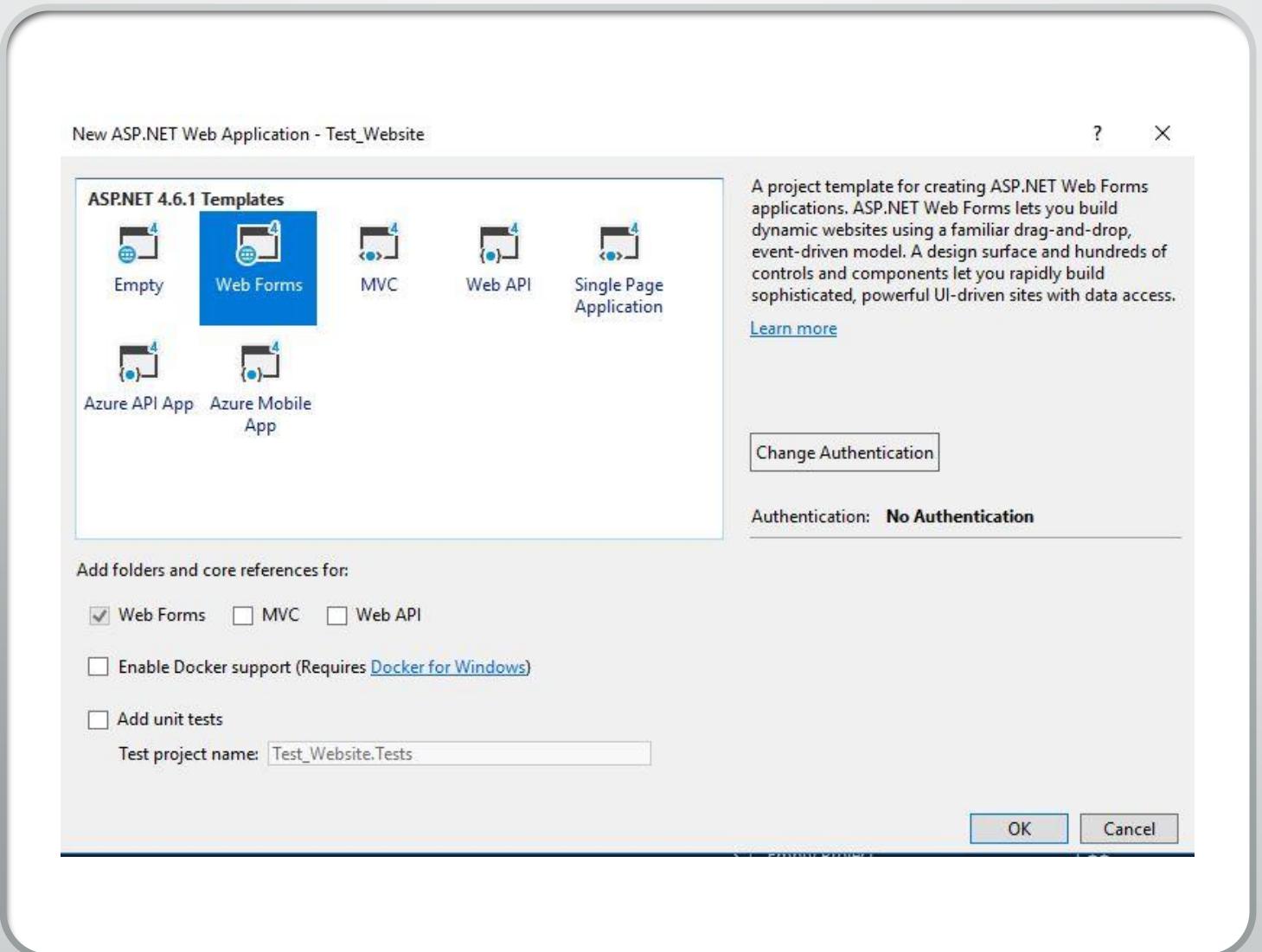


New Project Panel

- On the side panel select **Other Languages**
- Select **Visual C#**
- Select **ASP.NET Web Application**
- Click **OK**



- Select **Web Forms**
- Click **OK**



Test_Website - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug Any CPU Google Chrome

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

Solution 'Test_Website' (1 project)

Test_Website

- Connected Services
- Properties
- References
 - App_Data
 - App_Start
 - Content
 - fonts
 - Scripts
 - About.aspx
 - ApplicationInsights.config
 - Bundle.config
 - Contact.aspx
 - Default.aspx
 - favicon.ico
 - Global.asax
 - packages.config
- Site.Master
- Site.Mobile.Master
- ViewSwitcher.ascx
- Web.config

ASP.NET

Learn about the .NET platform, create your first application and extend it to the cloud.

</>

Build Your App

Get started with ASP.NET

Browse docs, samples, and tutorials

Connect to Azure

Sign up for free

Publish your website to Azure

Set up continuous delivery

Azure Publish Quickstarts

Add a service

Telemetry with Application Insights

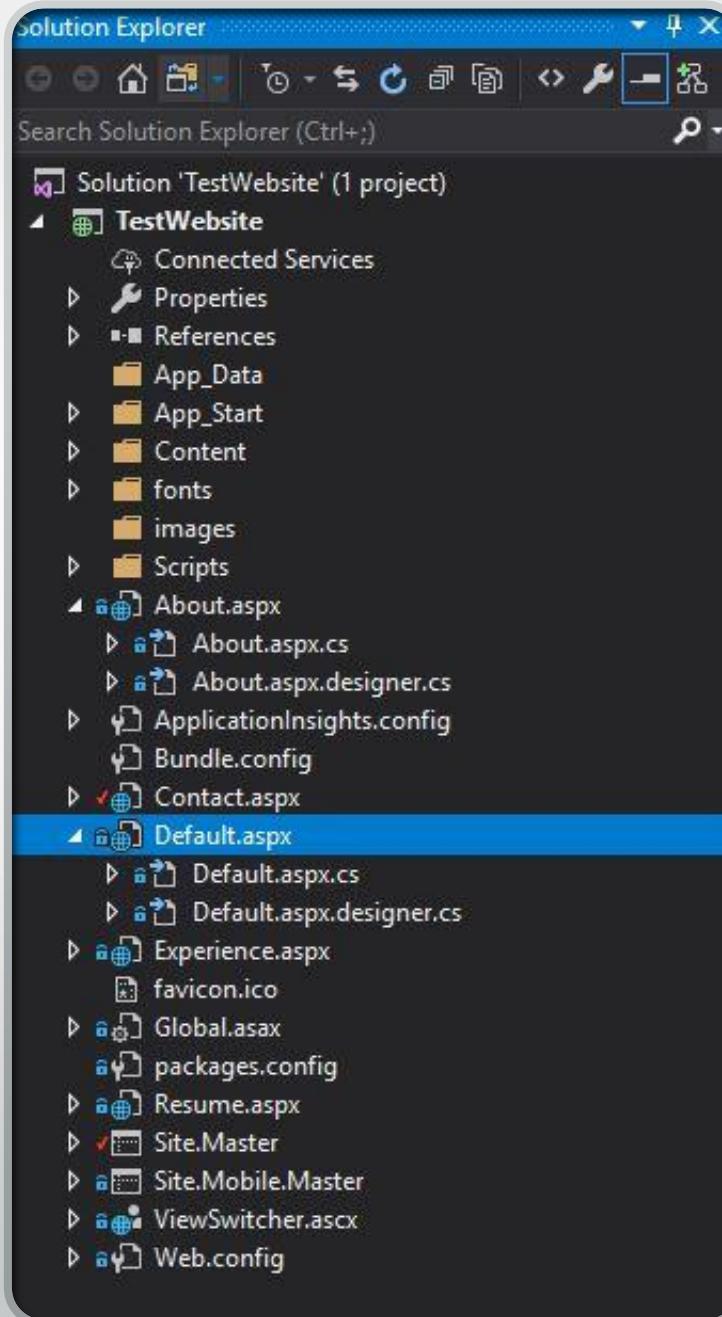
More connected services

Server Explorer Toolbox Properties

Solution Expl... Class View Property Man... Team Explorer

Understanding the Structure of the Project Folder

- Each Page has a markup file and a code file
- .aspx = Markup File
- .aspx.cs = Code Files
- .aspx.designer.cs = Joins Markup & Code
- The master page is the default template that every page uses (your navigation menu and footer should be in the master page). Site.Master and Mobile.Master
- Web.config holds specific settings about the behavior of your website.
- The scripts folder contains any libraries that you import





Building/Customizing the website

Jeff's Portfolio Built on ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more »](#)

Home Page (default.aspx)

[Get more libraries](#)[Web Hosting](#)

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

```

1  <%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="TestWebsite.Default" %>
2
3  <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
4
5      <div class="jumbotron">
6          <h1>Jeff's Portfolio Built on ASP.NET</h1>
7          <%--<asp:Image runat="server" ImageUrl("~/images/jeff_headshot.jpg" Width="300" Height="300"/><!--%&gt;
8          &lt;p class="lead"&gt;ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.&lt;/p&gt;
9          &lt;p&gt;&lt;a href="http://www.asp.net" class="btn btn-primary btn-lg"&gt;Learn more &amp;raquo;&lt;/a&gt;&lt;/p&gt;
10     &lt;/div&gt;
11
12     &lt;div class="row"&gt;
13         &lt;div class="col-md-4"&gt;
14             &lt;h2&gt;Getting started&lt;/h2&gt;
15             &lt;p&gt;
16                 ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model.
17                 A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with
18             &lt;/p&gt;
19             &lt;p&gt;
20                 &lt;a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301948"&gt;Learn more &amp;raquo;&lt;/a&gt;
21             &lt;/p&gt;
22         &lt;/div&gt;
23         &lt;div class="col-md-4"&gt;
24             &lt;h2&gt;Get more libraries&lt;/h2&gt;
25             &lt;p&gt;
26                 NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio.
27             &lt;/p&gt;
28             &lt;p&gt;
29                 &lt;a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301949"&gt;Learn more &amp;raquo;&lt;/a&gt;
30             &lt;/p&gt;
31         &lt;/div&gt;
32         &lt;div class="col-md-4"&gt;
33             &lt;h2&gt;Web Hosting&lt;/h2&gt;
34             &lt;p&gt;
35                 You can easily find a web hosting company that offers the right mix of features and price for your applications.
36             &lt;/p&gt;
37             &lt;p&gt;
38                 &lt;a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301950"&gt;Learn more &amp;raquo;&lt;/a&gt;
39             &lt;/p&gt;
40         &lt;/div&gt;
41     &lt;/div&gt;
42
43 &lt;/asp:Content&gt;
44
</pre>

```

Jeff Beauplan Home About Resume Experience Contact

Jeff's Portfolio Built on ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

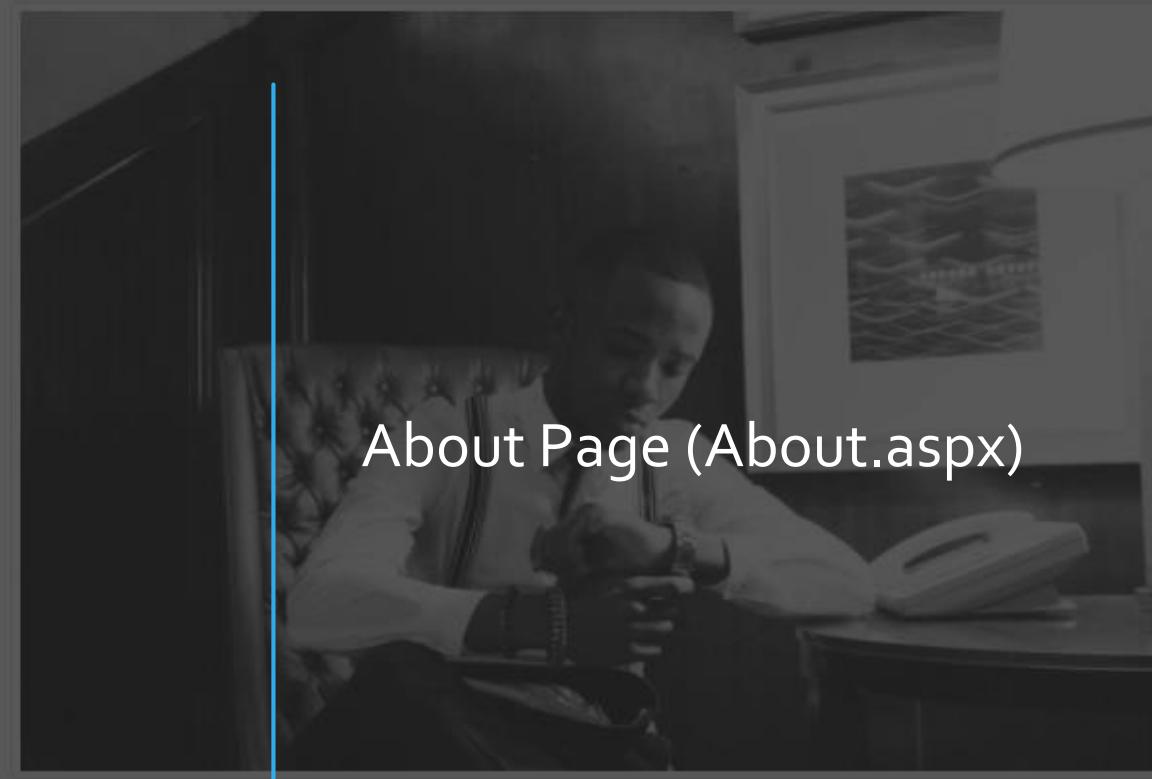
Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2018 - Jeff's ASP.NET Application

Personal Bio



About Page (About.aspx)

My name is Jeff Samuel Beauplan. I was born in Port-au-Prince, Haiti on November 29, 1996. I relocated to the United States at a very young age after an assassination attempt on my father due to his financial standings. I now attend Howard University as a Computer Science major. Throughout my life I have faced a series of obstacles and challenges, all of which have helped to shape me not only as a person but also as a scholar. In this portfolio you will see some of the projects I've worked on that demonstrate some of my strengths and weaknesses as an engineer and a scholar. By partaking in these projects I've learned a lot about myself, team work, and the many different ways one can go about solving a problem. This site will serve as a catalog for my experiences and endeavors as an engineer.

Personal Bio

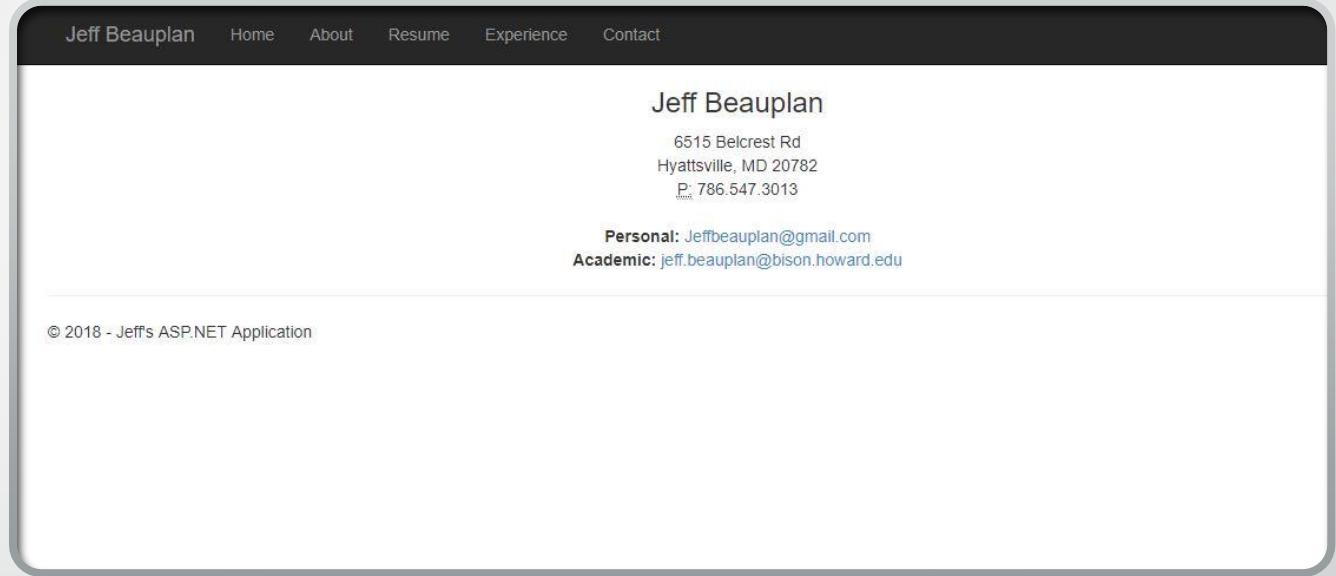


My name is Jeff Samuel Beauplan. I was born in Port-au-Prince, Haiti on November 29, 1996. I relocated to the United States at a very young age after an assassination attempt on my father due to his financial standings. I now attend Howard University as a Computer Science major. Throughout my life I have faced a series of obstacles and challenges, all of which have helped to shape me not only as a person but also as a scholar. In this portfolio you will see some of the projects I've worked on that demonstrate some of my strengths and weaknesses as an engineer and a scholar. By partaking in these projects I've learned a lot about myself, team work, and the many different ways one can go about solving a problem. This site will serve as a catalog for my experiences and endeavors as an engineer.

© 2018 - Jeff's ASP.NET Application

```
1 <%@ Page Title="About" Language="C#" MasterPageFile("~/Site.Master" AutoEventWireup="true" CodeBehind="About.aspx.cs" Inherits="TestWebsite.About" %>
2
3 <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
4   <asp:Panel runat="server" ID="Pnl_Content" HorizontalAlign="Center">
5     <h2> Personal Bio</h2>
6     <%--<h3>Your application description page.</h3>--%>
7     <div>
8       <asp:Image runat="server" ImageUrl="~/images/jeff_chair.jpg" Width="600" Height="400" class="img-thumbnail"/>
9     </div>
10
11     <p>My name is Jeff Samuel Beauplan. I was born in Port-au-Prince, Haiti on November 29, 1996. I relocated to the United States at a very young age after an assassination attempt on my father due to
12
13   </asp:Panel>
14 </asp:Content>
15
```

Contact Page (Contact.aspx)



Jeff Beauplan

6515 Belcrest Rd
Hyattsville, MD 20782
P: 786.547.3013

Personal: jeffbeauplan@gmail.com
Academic: jeff.beauplan@bison.howard.edu

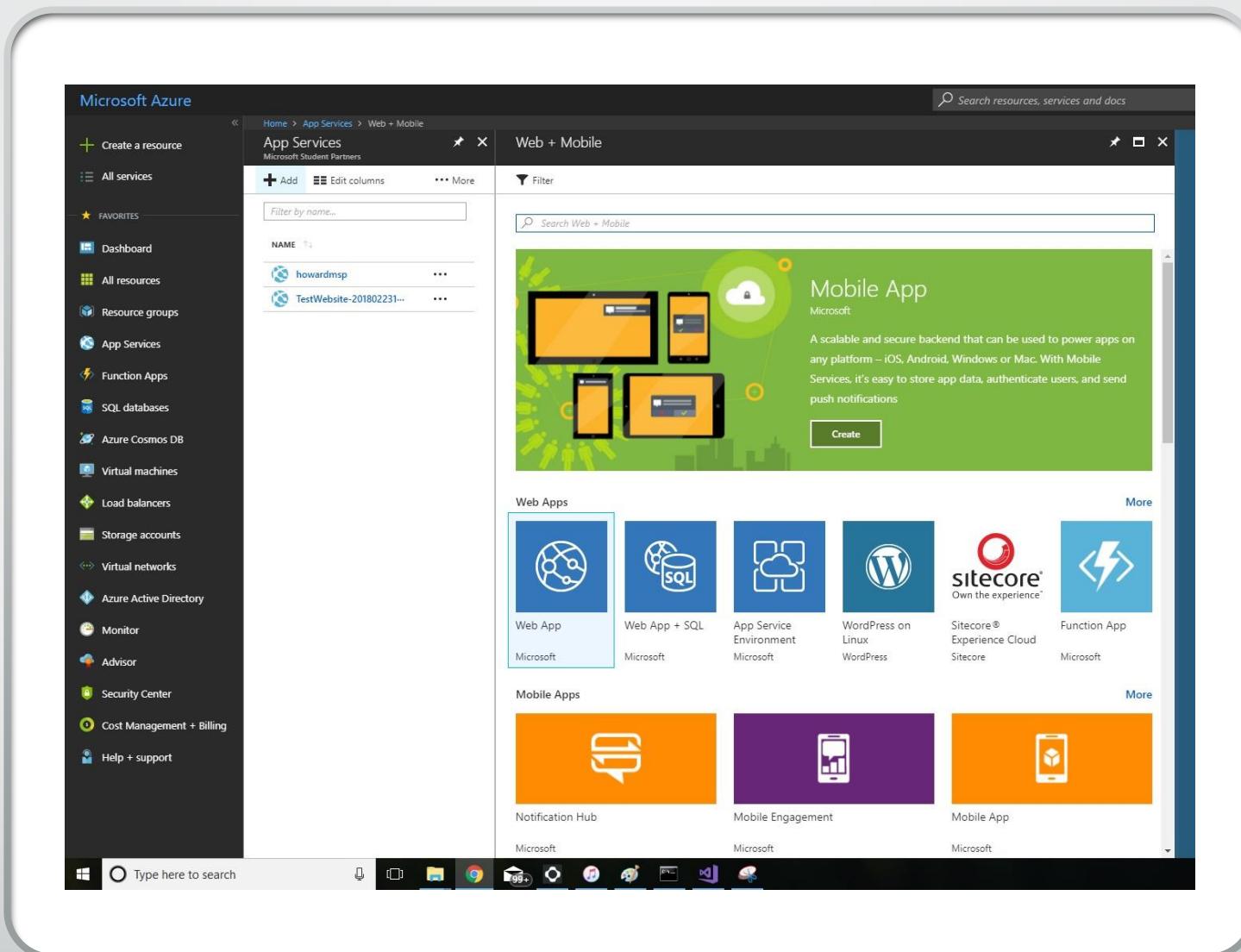
© 2018 - Jeff's ASP.NET Application

```
1 <%@ Page Title="Contact" Language="C#" MasterPageFile("~/Site.Master" AutoEventWireup="true" CodeBehind="Contact.aspx.cs" Inherits="TestWebsite.Contact" %>
2
3 <asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
4   <asp:Panel runat="server" ID="Pnl_Content" HorizontalAlign="Center">
5     <%--<h2><%: Title %>.</h2>--%>
6     <h3>Jeff Beauplan</h3>
7     <address>
8       6515 Belcrest Rd<br />
9       Hyattsville, MD 20782<br />
10      <abbr title="Phone">P:</abbr>
11      786.547.3013
12    </address>
13
14    <address>
15      <strong>Personal:</strong> <a href="mailto:jeffbeauplan@gmail.com">Jeffbeauplan@gmail.com</a><br />
16      <strong>Academic:</strong> <a href="mailto:jeff.beauplan@bison.howard.edu">jeff.beauplan@bison.howard.edu</a>
17    </address>
18  </asp:Panel>
19
20 </asp:Content>
21
```



Deployment Process via Azure Cloud Services

- Go to **portal.azure.com**
- Log in
- Select **App Services** from side panel



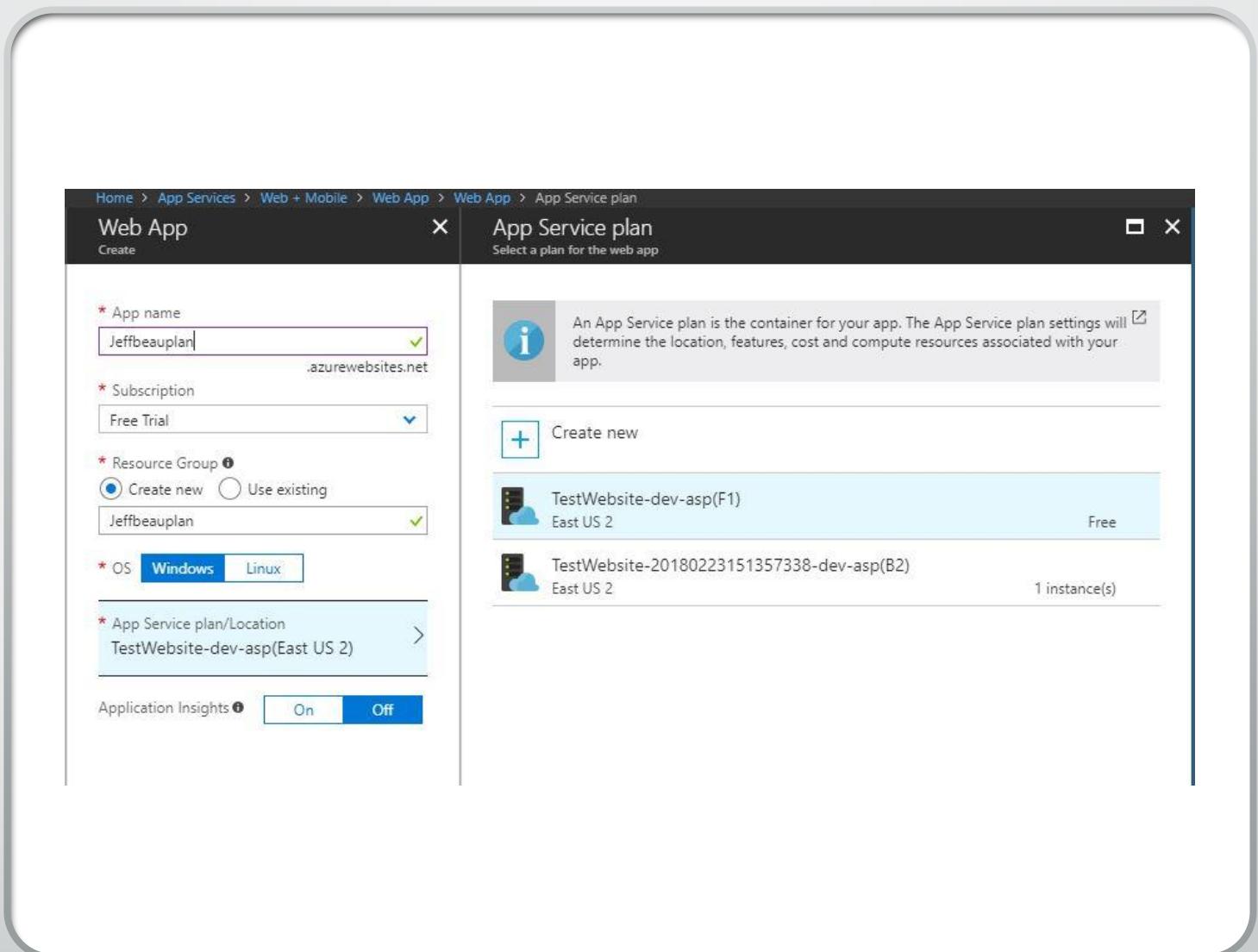
The screenshot shows the Microsoft Azure portal interface. The left sidebar lists various services under 'All services'. The main area is titled 'Web + Mobile' and displays several service options:

- Mobile App**: Described as a scalable and secure backend for mobile platforms. It includes a 'Create' button.
- Web Apps**: Includes 'Web App' (Microsoft), 'Web App + SQL' (Microsoft), 'App Service Environment' (Microsoft), 'WordPress on Linux' (Wordpress), 'Sitecore® Experience Cloud' (Sitecore), and 'Function App' (Microsoft).
- Mobile Apps**: Includes 'Notification Hub' (Orange icon), 'Mobile Engagement' (Purple icon), and 'Mobile App' (Orange icon).

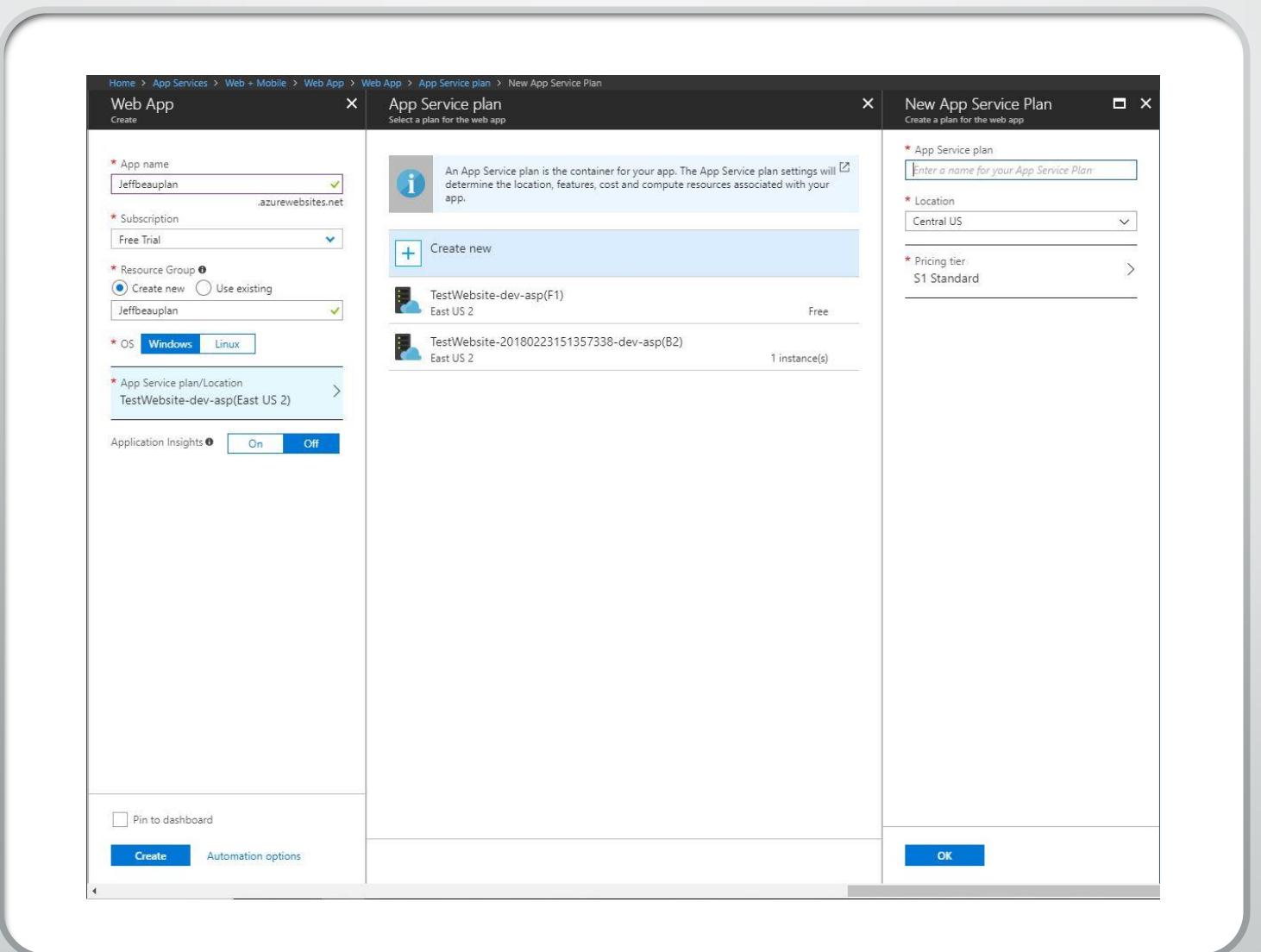
A separate window titled 'Web App' is open on the right, providing more details about the service, including its benefits and a 'Create' button.

Select **Web App** and then click **Create**

- Enter your app name (this will eventually be the link that you use to go to your webstie)
- Leave everything else on default. It automatically makes a new resource group for you with the app name



- Click **Create new** under App Service Plan
- Enter a name for the app service plan
- Click **Ok**
- Then Click **Create**



The screenshot shows the Azure App Services Overview page for the 'howardmsp' app service. The left sidebar lists various settings and monitoring options. The main area displays the app's status (Running), location (East US 2), and deployment metrics. The 'Data In' chart shows 0 B transferred at 7:45 PM. The 'Data Out' chart shows 0 B transferred at 7:45 PM. The 'Http 5xx' chart shows 0 errors from 7:45 PM to 8:30 PM.

Home > App Services > howardmsp

App Services Microsoft Student Partners

+ Add Edit columns More

Filter by name...

NAME ↗

howardmsp ...

TestWebsite-201802231... ...

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

DEPLOYMENT

Quickstart

Deployment credentials

Deployment slots

Deployment options

Continuous Delivery (Preview)

SETTINGS

Application settings

Authentication / Authorization

Managed service identity

Backups

Custom domains

SSL certificates

Networking

Scale up (App Service plan)

Scale out (App Service plan)

Search (Ctrl+ /)

Browse Stop Swap Restart Delete Get publish profile Reset publish profile

Resource group (change)
howardmsp

Status
Running

Location
East US 2

Subscription (change)
Free Trial

Subscription ID
81fca1b9-3108-45d6-9f98-1b263f6430c

URL
<https://howardmsp.azurewebsites.net>

App Service plan/pricing tier
TestWebsite-20180223151357338-dev-asp (Basic: 1 Medium)

Continuous delivery status
--

Edit continuous delivery
https://jeffbeauplan.visualstudio.com/d2329335-4fac-45e6-8d4d-633aacbd3b4/_apps/hub...

Http 5xx

100
80
60
40
20
0

7:45 PM 8 PM 8:15 PM 8:30 PM

HTTP SERVER ERRORS 0

Data In

100B
80B
60B
40B
20B
0B

7:45 PM 8 PM 8:15 PM 8:30 PM

DATA IN 0 B

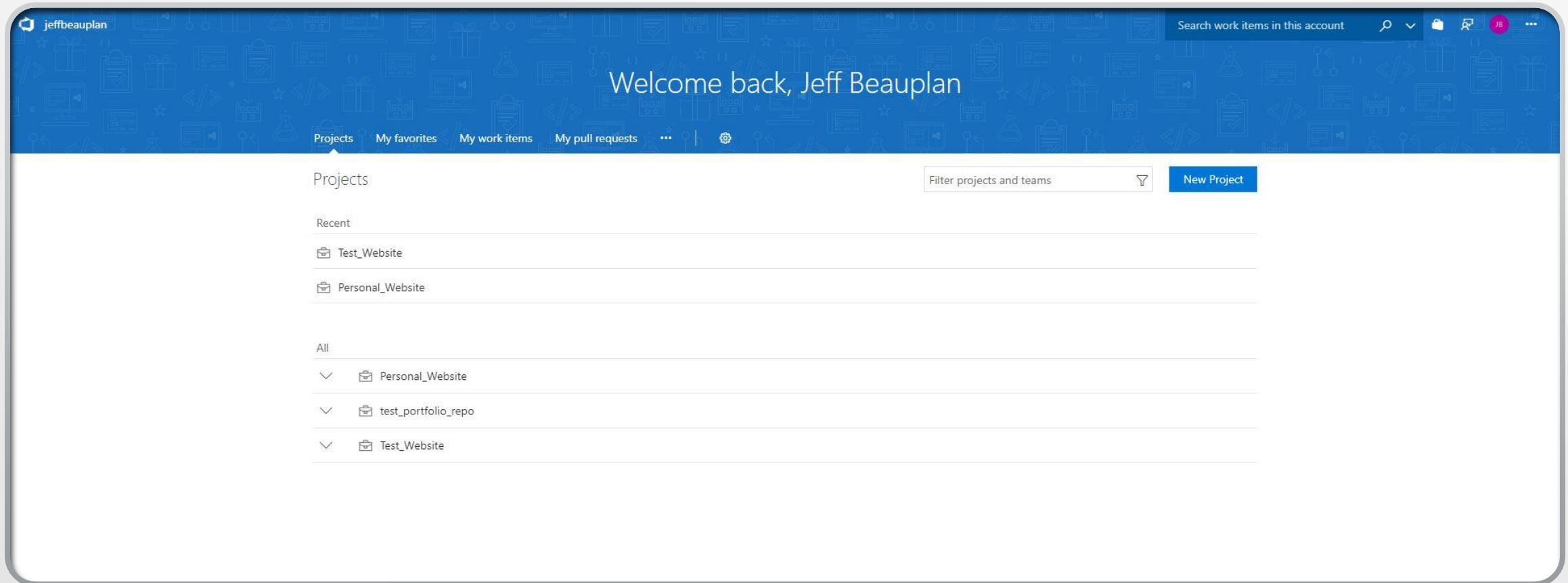
Data Out

100B
80B
60B
40B
20B
0B

7:45 PM 8 PM 8:15 PM 8:30 PM

DATA OUT 0 B

You should now be able to see your newly created app service under the **App Services** Menu



- Log into your Visual Studio Team Services Account
- Click **New Project**

The screenshot shows the Microsoft Visual Studio Team Services (VSTS) web interface. At the top, there's a navigation bar with links for Jeff_Portfolio, Dashboards, Code, Work, Build and Release, Test, Wiki, and a gear icon. To the right of the navigation is a search bar labeled "Search work items in this project" and a set of icons for filtering and saving. On the far right, there are user profile icons for "JB" and "...".

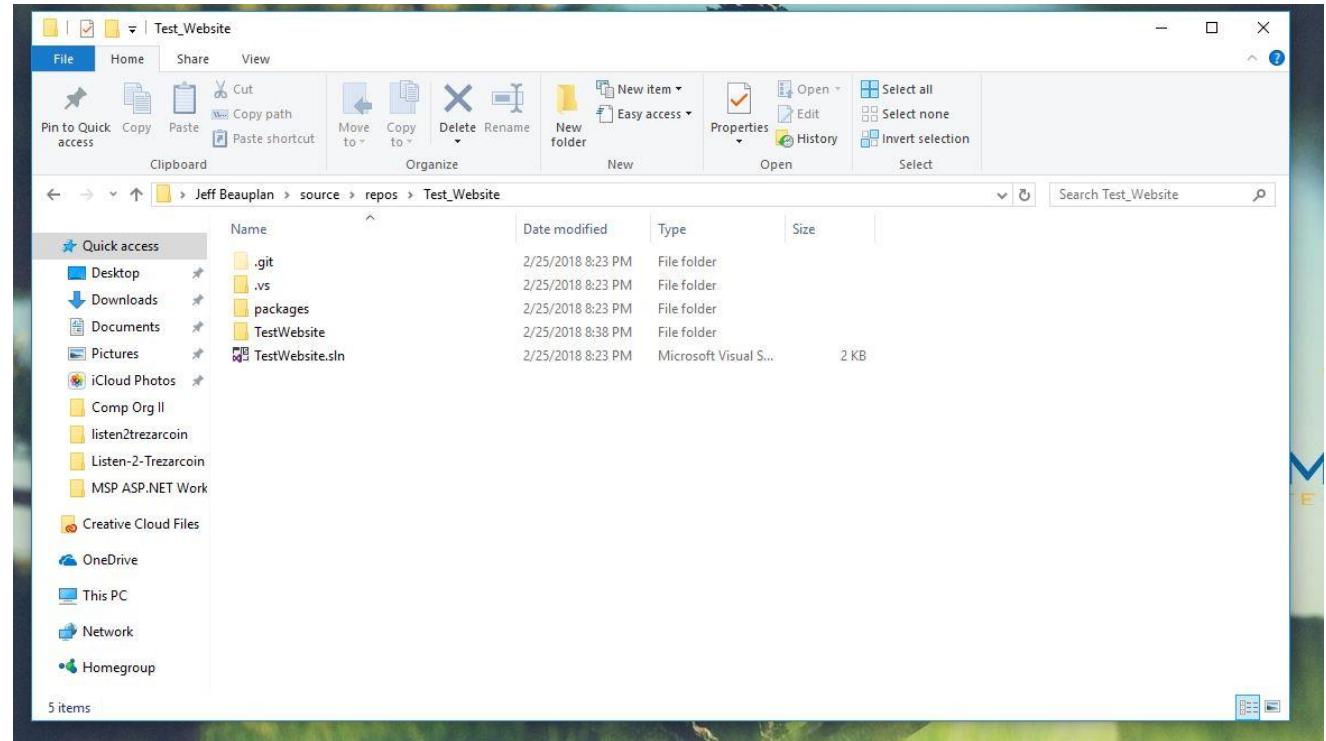
The main content area features a large orange circular icon with a white letter "J" and the text "Jeff_Portfolio ☆". Below it is a subtitle: "This project is my personal portfolio website". There's a "Add tags" button and a "Get started with your new project!" message.

A section titled "Clone to your computer" provides cloning options: "HTTPS" (selected), "SSH", and a URL "https://jeffbeauplan.visualstudio.com/_git/Jeff_Portfolio". An "OR" link leads to "Clone in Visual Studio". Below these are buttons for "Generate Git credentials" and a note about authentication.

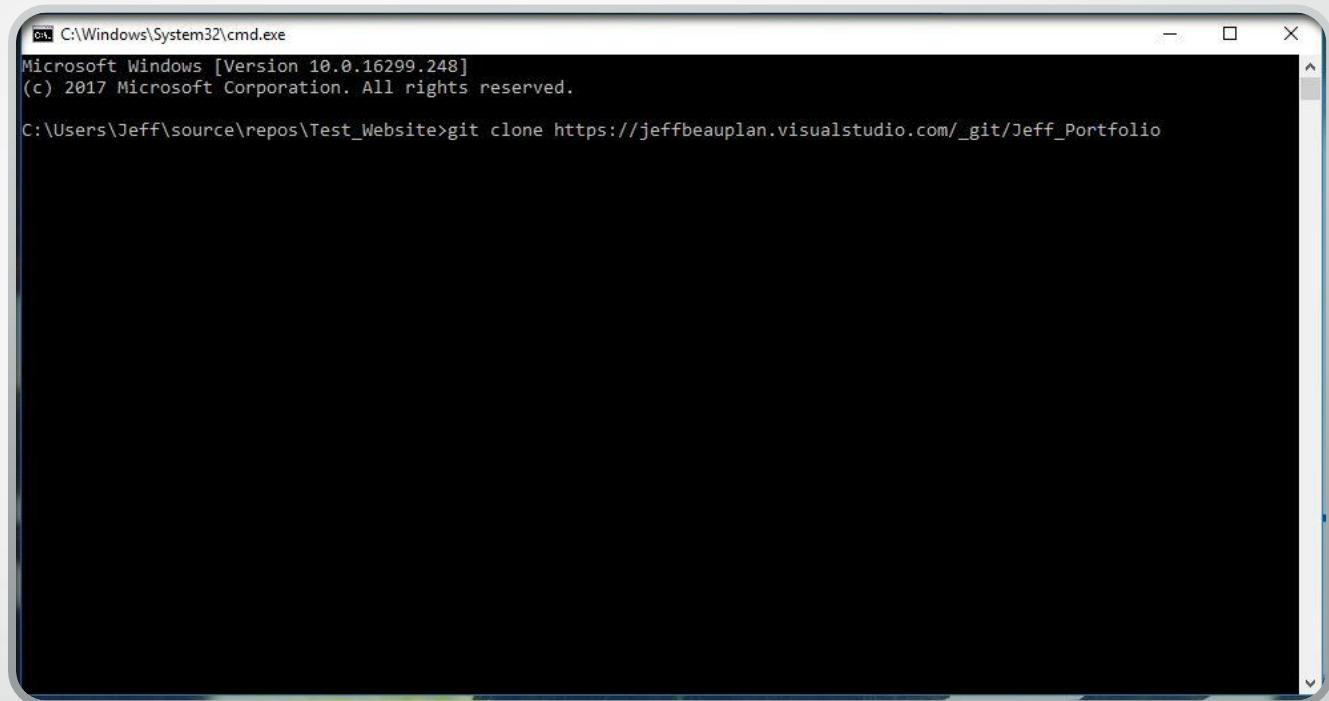
On the right side, there are several sections: "Members (1)" with a "JB" profile and a plus sign; "Activity" (7 Days) with a "Code" section showing "No code yet" and a "Build & Release" section showing "No builds yet"; "Work" with a "Work" section showing "No work items yet".

- Once your Project is created its now time to clone the repository to your computer
- Copy the HTTPS clone link

- Navigate to your Visual Studio Project folder for the website



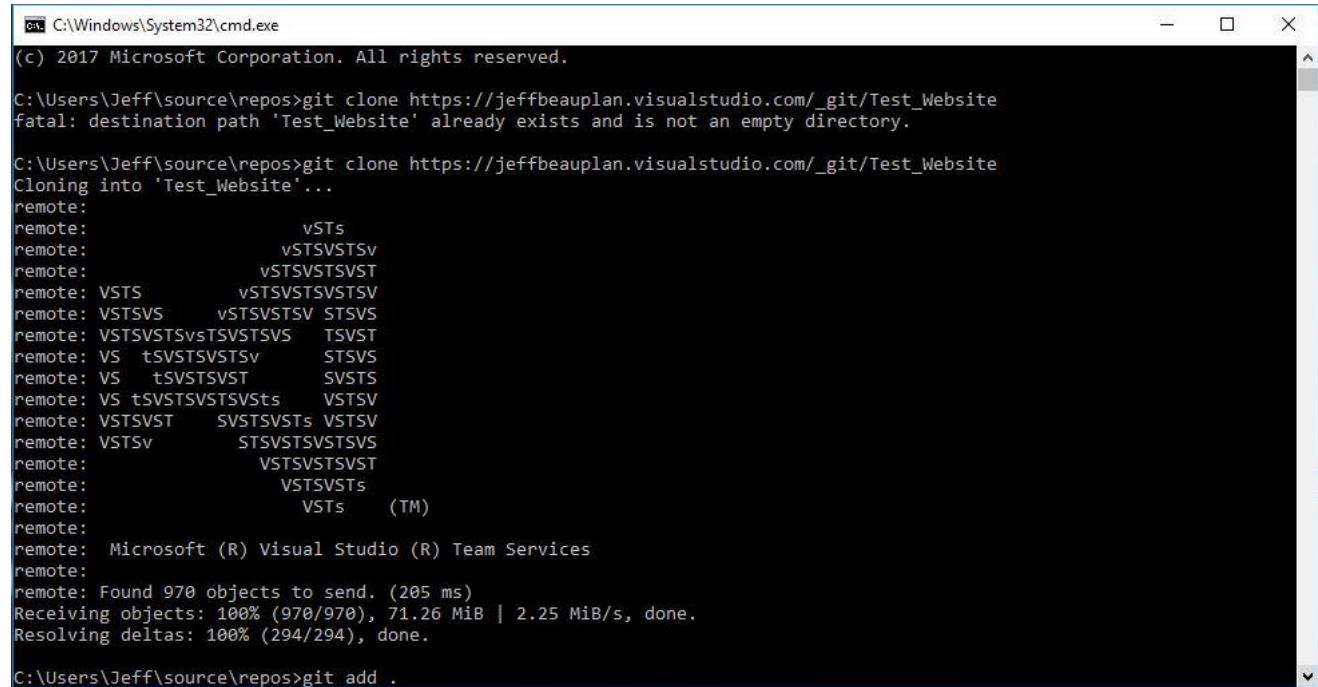
- Open up a git terminal inside of the project folder
- Type **git clone [paste https clone link here]**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.16299.248]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Jeff\source\repos\Test_Website>git clone https://jeffbeuplan.visualstudio.com/_git/Jeff_Portfolio
```

- After cloning the repo its now time to sync your project files with the repo
- Type **git add .**



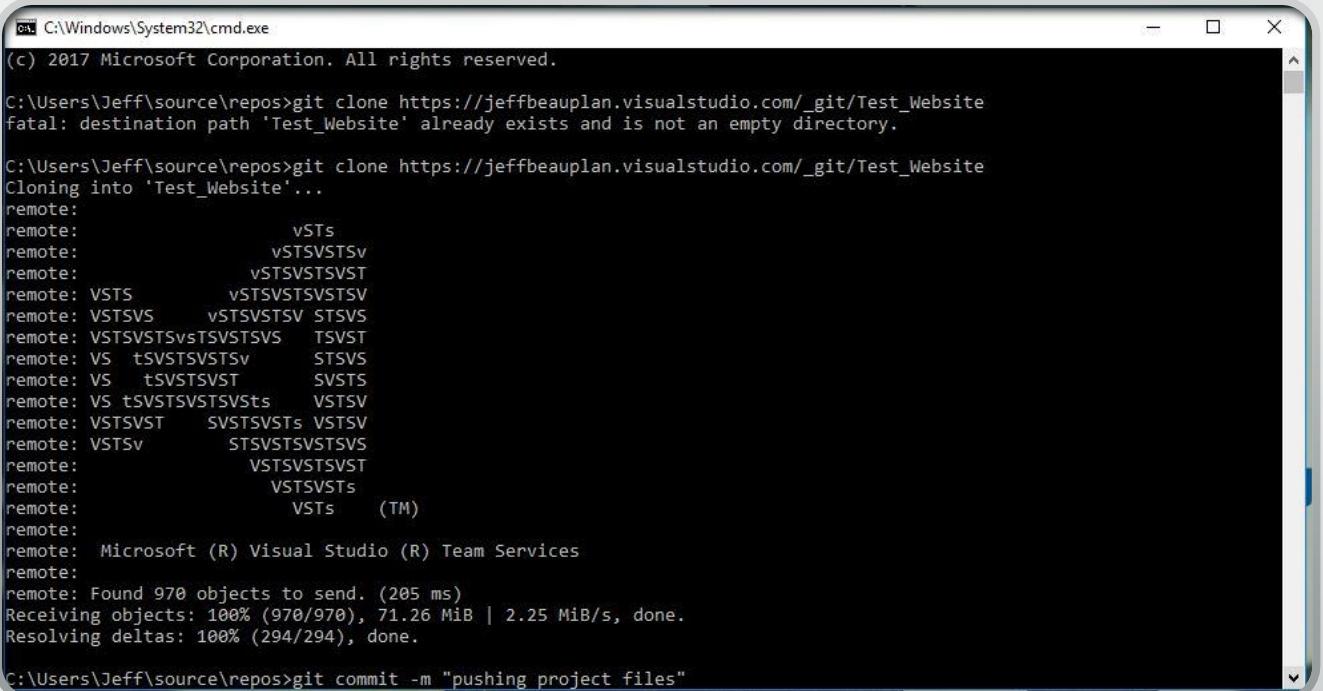
```
C:\Windows\System32\cmd.exe
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Jeff\source\repos>git clone https://jeffbeauplan.visualstudio.com/_git/Test_Website
fatal: destination path 'Test_Website' already exists and is not an empty directory.

C:\Users\Jeff\source\repos>git clone https://jeffbeauplan.visualstudio.com/_git/Test_Website
Cloning into 'Test_Website'...
remote:
remote:          vSTS
remote:          vSTSvSTSV
remote:          vSTSvSTSVST
remote: VSTS          vSTSvSTSVSTSV
remote: VSTSvS          vSTSvSTSV STSVS
remote: VSTSvSTSVsTSVSTSVS  TSVST
remote: VS   tSVSTSVSTv          STSVS
remote: VS   tSVSTSVST          SVSTS
remote: VS tSVSTSVSTSVsts      VSTSv
remote: VSTSvST          SVSTSvSTs VSTSv
remote: VSTSv          STSVSTSVSTSVS
remote:          VSTSvSTSVST
remote:          VSTSvSTS
remote:          VSTSs (TM)
remote:
remote: Microsoft (R) Visual Studio (R) Team Services
remote:
remote: Found 970 objects to send. (205 ms)
Receiving objects: 100% (970/970), 71.26 MiB | 2.25 MiB/s, done.
Resolving deltas: 100% (294/294), done.

C:\Users\Jeff\source\repos>git add .
```

- Type `git commit -m "pushing project files"`



The screenshot shows a Windows Command Prompt window titled "C:\Windows\System32\cmd.exe". The window displays the following command-line session:

```
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Jeff\source/repos>git clone https://jeffbeaulan.visualstudio.com/_git/Test_Website
fatal: destination path 'Test_Website' already exists and is not an empty directory.

C:\Users\Jeff\source/repos>git clone https://jeffbeaulan.visualstudio.com/_git/Test_Website
Cloning into 'Test_Website'...
remote:
remote:           VSTS
remote:           VSTSvSTSV
remote:           VSTSvSTSVST
remote: VSTS           VSTSvSTSVSTSV
remote: VSTSvS           VSTSvSTSV STSVS
remote: VSTSvTSvTSVSTSVS   TSVST
remote: VS tSVTSVSTSV           STSVS
remote: VS tSVTSVST           SVSTS
remote: VS tSVTSVSTSVsts   VTSV
remote: VSTSvST           SVTSvSTs VTSV
remote: VSTSv           STSVTSVSTSVS
remote:           VSTSvSTSVST
remote:           VSTSvSTS
remote:           VSTS (TM)
remote:
remote: Microsoft (R) Visual Studio (R) Team Services
remote:
remote: Found 970 objects to send. (205 ms)
Receiving objects: 100% (970/970), 71.26 MiB | 2.25 MiB/s, done.
Resolving deltas: 100% (294/294), done.

C:\Users\Jeff\source/repos>git commit -m "pushing project files"
```

- Type **git push**
- Your visual studio Team Services repo should now be in sync with your visual studio project folder

```
C:\Windows\System32\cmd.exe
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Jeff\source\repos>git clone https://jeffbeauplan.visualstudio.com/_git/Test_Website
fatal: destination path 'Test_Website' already exists and is not an empty directory.

C:\Users\Jeff\source\repos>git clone https://jeffbeauplan.visualstudio.com/_git/Test_Website
Cloning into 'Test_Website'...
remote:
remote:          vSTS
remote:          vSTSvSTSV
remote:          vSTSvSTSvST
remote: VSTS          vSTSvSTSvSTSV
remote: VSTSvS      vSTSvSTSvSTSVS
remote: VSTSvSTSvSvTSvSTSvS      TSVST
remote: VS   tSVSTSvSTsv      STSVS
remote: VS   tSVSTSvST      SVSTS
remote: VS   tSVSTSvSTSvsts      VSTSv
remote: VSTSvST      SVSTSvSTSv      VSTSv
remote: VSTSv      STSVSTSvSTSvS
remote:          VSTSvSTSvST
remote:          VSTSvSTS
remote:          VSTS      (TM)
remote:
remote: Microsoft (R) Visual Studio (R) Team Services
remote:
remote: Found 970 objects to send. (205 ms)
remote: Receiving objects: 100% (970/970), 71.26 MiB | 2.25 MiB/s, done.
remote: Resolving deltas: 100% (294/294), done.

C:\Users\Jeff\source\repos>git push
```

Test_Website

Search work items in this project

Test_Website Files Commits Pushes Branches Tags Pull Requests Fork Clone

master Test_Website / Type to find a file or folder... ✓ succeeded

Name	Last change	Commits
.vs	2/23/2018	b100912d changed header in resume and experience Jeff
packages	2/23/2018	4688d973 pushed project Jeff
TestWebsite	2/23/2018	b100912d changed header in resume and experience Jeff
TestWebsite.sln	2/23/2018	4688d973 pushed project Jeff

You should now be able to see your project files on the Visual Studio website

The screenshot shows the Microsoft Azure DevOps interface for a project named "Test_Website". The top navigation bar includes links for Test_Website, Dashboards, Code, Work, Build and Release, Test, Wiki, and a gear icon. A search bar at the top right says "Search work items in this project". The "Build and Release" menu is open, with "Builds" selected. On the left, there's a sidebar with "Test_Website" expanded, showing ".vs", "packages", "TestWebsite", and "TestWebsite.sln". Below this is a "Contents" tab and a "History" tab. The main area has tabs for "Builds", "Releases", "Library", "Task Groups", and "Deployment Groups*". The "Builds" tab is active, showing a list of commits:

			Commits
1	.vs	2/23/2018	b100912d changed header in resume and experience Jeff
2	packages	2/23/2018	4688d973 pushed project Jeff
3	TestWebsite	2/23/2018	b100912d changed header in resume and experience Jeff
4	TestWebsite.sln	2/23/2018	4688d973 pushed project Jeff

At the bottom of the interface, there is a URL: https://jeffbeauplan.visualstudio.com/Test_Website/_build.

- In the menu under **Build and Release** select **Builds**

The screenshot shows the 'Select your repository' step in the Azure DevOps pipeline setup. On the left, there's a large circular arrow icon with an arrow pointing right, followed by the text 'Select your repository'. Below that, a note says 'Tell us where your sources are. You can customize how to get these sources from the repository later.' On the right, the 'Sources' section is visible, featuring icons for 'This account' (selected), GitHub, GitHub Enterprise, Bitbucket Cloud, External Git, and Subversion. Below this, the 'Team project' dropdown is set to 'Test_Website', the 'Repository' dropdown is set to 'Test_Website', and the 'Branch' dropdown is set to 'master'. A 'Continue' button is at the bottom.

- Make sure the correct project and repository is selected
- Click **Continue**

Test_Website

Builds Releases Library Task Groups Deployment Groups*

Search work items in this project

Search

Select a template

Or start with an Empty process

Search

Featured

-  .NET Desktop
Build and run tests for .NET Desktop or Windows Classic Desktop solutions. This template requires that Visual Studio be installed on the build agent.
-  ASP.NET
Build ASP.NET web applications
-  ASP.NET Core
Build ASP.NET Core web applications targeting .NET Core
-  ASP.NET Core (.NET Framework)
Build ASP.NET Core web applications targeting the full .NET Framework
-  Azure Web App
Build, package, test and deploy your Azure Web App.
-  Universal Windows Platform
Build Universal Windows Platform applications using Visual Studio. This template requires that Visual Studio and the Universal templates are installed on the build agent.

Apply

Others

-  ASP.NET with Containers
Build and push an ASP.NET project with container support
-  Android
Build your Android projects, run tests, sign and align Android App Package files. This template requires the Android SDK to be installed on the build agent.
-  Ant
Build your Java projects and run tests with Apache Ant. This template requires Ant to be installed on the build agent.
-  Azure Cloud Services
Build, package, test and deploy your Azure Cloud Service.

- Select **ASP.NET**
- Click **Apply**

The screenshot shows the Microsoft Azure DevOps interface for managing a build pipeline. The top navigation bar includes links for Test_Website, Dashboards, Code, Work, Build and Release, Test, Wiki, and a gear icon. A search bar and various project management icons are also present.

The main area displays a build pipeline named "Test_Website-ASP.NET-CI (1)". The pipeline consists of the following stages:

- Get sources**: Set to pull from the "Test_Website" repository on the "master" branch.
- Phase 1** (Run on agent):
 - Use NuGet 4.3.0**: NuGet Tool Installer
 - NuGet restore**: NuGet
 - Build solution**: Visual Studio Build
 - Test Assemblies**: Visual Studio Test
 - Publish symbols path**: Index Sources & Publish Symbols
 - Publish Artifact**: Publish Build Artifacts

On the right side, there are configuration options for the pipeline:

- Name**: Test_Website-ASP.NET-CI (1)
- Agent queue**: Hosted VS2017
- Parameters**: Path to solution or packages.config: ***.sln
- Artifact Name**: drop

Click Triggers

The screenshot shows the 'Triggers' tab of a build definition in Microsoft Azure DevOps. The build definition is named 'Test_Website-ASP.NET-CI (1)'. In the 'Continuous integration' section, there is one trigger named 'Test_Website' which is currently enabled. Under 'Branch filters', the 'Type' is set to 'Include' and the 'Branch specification' is 'master'. There are also sections for 'Path filters' and 'Add' buttons for both branch and path filters.

- Select **Enable continuous integration**
- Click **Save & queue**

Test_Website

Builds Releases Library Task Groups Deployment Groups*

... > Test_Website-ASP.NET-CI (1)

Save & queue Discard Summary Queue ...

Tasks Variables Triggers Options Retention History

Process Build process

Get sources Test_Website master

Phase 1 Run on agent

- Use NuGet 4.3.0 NuGet Tool Installer
- NuGet restore NuGet
- Build solution Visual Studio Build
- Test Assemblies Visual Studio Test
- Publish symbols path Index Sources & Publish Symbols
- Publish Artifact Publish Build Artifacts

Name * Test_Website-ASP.NET-CI (1)

Agent queue * Manage Hosted VS2017

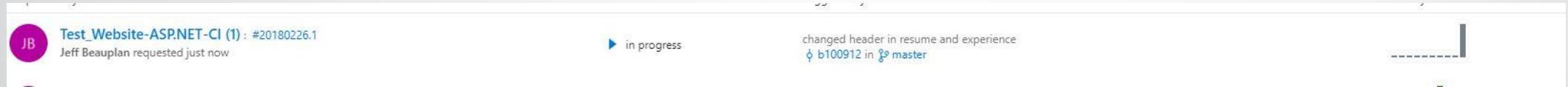
Save build definition

Select folder *

Comment

Save Cancel

Click Save



Your first build should now be in progress

The screenshot shows the Microsoft Azure DevOps interface for a project named 'Test_Website'. The top navigation bar includes options like Dashboards, Code, Work, Build and Release, Test, Wiki, and a search bar. The 'Build and Release' tab is active. Below it, the 'Releases' tab is selected. On the left, there's a sidebar with a '+ Create release definition' button and a list of existing releases: 'AzureServerVm', 'HowardMsp', and 'TestWebsite'. The main area displays a table of releases:

	Title	Environments	Build	Branch	Created	Created By	Description
1	Release-4	...	20180223.5 (Build)	master	2/23/2018	Jeff Beauplan	Triggered by Test_Website-ASP...
2	Release-3	✓	20180223.4 (Build)	master	2/23/2018	Jeff Beauplan	Triggered by Test_Website-ASP...
3	Release-2	✓	20180223.3 (Build)	master	2/23/2018	Jeff Beauplan	Triggered by Test_Website-ASP...
4	Release-1	✓	20180223.2 (Build)	master	2/23/2018	Jeff Beauplan	Triggered by Test_Website-ASP...

A context menu is open over the '+ Create release definition' button, with 'Create release definition' highlighted.

- Click on **Release** from the **Build and Release** option from the top menu
- Click the **+** icon and select **Create release definition**

The screenshot shows the Azure DevOps interface for creating a new release definition. On the left, the pipeline configuration is visible, featuring sections for Artifacts, Environments, and Tasks. The Environments section contains a single environment named "Environment 1" with a "Select template" button. On the right, a list of deployment templates is displayed under the heading "Select a Template". The "Featured" section includes options for Azure App Service Deployment, Deploy Node.js App to Azure App Service, Deploy PHP App to Azure App Service, IIS Website and SQL Database Deployment, and Azure Cloud Service Deployment. Below this, the "Others" section lists Azure App Service Deployment with Continuous Monitoring, Azure App Service Deployment with Performance Test, Azure App Service Deployment with Slot, Azure App Service Deployment with Test, and Azure Service Fabric Compose Deployment. A search bar at the top right allows users to find specific templates.

- Select **Azure App Service Deployment**

* We're going to deploy to the app service you created earlier in Azure

The screenshot shows the Azure DevOps interface for creating a new release definition. On the left, the main pipeline screen displays sections for 'Artifacts' and 'Environments'. The 'Artifacts' section has a button to 'Add artifact'. The 'Environments' section shows 'Environment 1' with a note '1 phase, 1 task'. On the right, a modal window titled 'Add artifact' is open, allowing the selection of a source type (Build, Git, GitHub, Team Foundation ...). The 'Build' option is selected. Below this, fields are filled out for 'Project' (Test_Website), 'Source (Build definition)' (Test_Website-ASP.NET-CI), and 'Default version' (Latest). A note at the bottom indicates that artifacts from the build will be available for deployment. A large blue 'Add' button is at the bottom right of the dialog.

Add artifact

Source type

Build

Git

Github

Team Foundation ...

3 more artifact types ▾

Project * ①

Test_Website

Source (Build definition) * ①

Test_Website-ASP.NET-CI

Default version * ①

Latest

Source alias ①

Test_Website-ASP.NET-CI

ⓘ The artifacts published by each version will be available for deployment in Release Management. The latest successful build of Test_Website-ASP.NET-CI published the following artifacts: drop.

Add

- Click **Add artifact**
- Select the build definition we created earlier
- Click **Add**

The screenshot shows the 'New Release Definition' page in Azure DevOps. On the left, there's a sidebar with 'Builds', 'Releases' (which is selected), 'Library', 'Task Groups', and 'Deployment Groups'. The main area has tabs for 'Pipeline' (selected) and 'Tasks'. Below these are sections for 'Environment 1' (Run on agent, Deploy Azure App Service), 'Parameters' (with a 'Manage' link highlighted by a red box), 'App type' (Web App), and 'App service name'.

Environment name: Environment 1

Parameters: [Manage](#)

Azure subscription: [Manage](#)

App type: Web App

App service name: [Manage](#)

- Click Task
- Click **Manage** (next to the Azure Subscription option)

https://jeffbeauplan.visualstudio.com/Test_Website/Test_Website%20Team/_admin/_services

The screenshot shows the Microsoft DevOps Services interface. The top navigation bar includes links for Test_Website / Test_We..., Dashboards, Code, Work, Build and Release, Test, Wiki, and a gear icon. A search bar on the right says "Search work items in this project" with a magnifying glass icon. Below the navigation is a secondary menu with links for Overview, Work, Security, Version Control, Policies, Agent Queues, Notifications, Service Hooks, **Services**, Test, Release, and Dashboards. The "Services" link is underlined, indicating it is the active section.

In the main content area, the title is "Endpoint: dev". On the left, a sidebar titled "Endpoints" lists various service types: Azure Classic, Azure Resource Manager (selected), Bitbucket Cloud, Chef, Docker Host, Docker Registry, External Git, Generic, and GitHub. The "Azure Resource Manager" option is highlighted with a blue background and white text. The main panel displays the selected endpoint details:

- INFORMATION**: Type: Azure Resource Manager, Created by Jeff Beauplan, Connecting to service using Service Principal.
- ACTIONS**: List of actions that can be performed on this service endpoint: Update Service Configuration, Manage Endpoint Roles, Manage Service Principal, Disconnect.

• Click + New Service Endpoint
• Select Azure Resource Manager

The screenshot shows the Azure DevOps Services interface for managing service endpoints. The top navigation bar includes links for Test_Website / Test_We..., Dashboards, Code, Work, Build and Release, Test, Wiki, and a gear icon for settings. A search bar at the top right allows searching for work items in the project. The main menu below the navigation bar includes Overview, Work, Security, Version Control, Policies, Agent Queues, Notifications, Service Hooks, Services (which is currently selected), Test, Release, and Dashboards. Under the Services menu, the 'Endpoints' tab is active, showing a list of existing endpoints: dev, Dev(Old), howardmsp, and VSTS. A search bar is available to find specific services. Below the list, there's a 'New Service Endpoint' button and a link to 'Search Services...'. The 'INFORMATION' section for the 'dev' endpoint shows it was created by Jeff Beauplan using an Azure Resource Manager service principal. The 'ACTIONS' section provides options like Update Service Configuration, Manage Endpoint Roles, Manage Service Principal, and Disconnect. A modal dialog titled 'Add Azure Resource Manager Service Endpoint' is displayed in the center. It contains fields for 'Connection name' (set to 'MyAzureConnection'), 'Subscription' (set to 'Free Trial (81fca1b9-3108-45d6-9f98-1b263f64330c)'), and 'Resource Group' (set to 'howardmsp'). Below these fields, a note states that subscriptions listed are from Azure Cloud, and a new Azure Service Principal will be created with 'Contributor' role. It also mentions that if the subscription is not listed, or the account is not backed by Azure Active Directory, a full version of the endpoint dialog can be used. At the bottom of the dialog are 'OK' and 'Close' buttons.

- Type in a connection name (this establishes a connection between visual studio team services and your Azure account)
- Select your subscription
- Select the Resource group (should be the same name as your app service)
- Click OK

The screenshot shows the Azure DevOps interface for creating a new release definition. The top navigation bar includes links for Test_Website, Dashboards, Code, Work, Build and Release, Test, Wiki, and a gear icon. The search bar at the top right contains the placeholder "Search work items in this project". Below the navigation is a secondary menu with options: Builds, Releases, Library, Task Groups, Deployment Groups*, and a dropdown arrow. The "Releases" option is currently selected.

The main content area displays the "All definitions > New Release Definition" page. At the top of this page are buttons for Save, Release, View releases, and three dots for more options. Below these are tabs for Pipeline, Tasks (which is selected), Variables, Retention, Options, and History.

The Pipeline configuration for "Environment 1" is shown on the left. It includes a "Deployment process" step, a "Run on agent" step (with a "Run on agent" icon and a plus sign), and a "Deploy Azure App Service" step (with an "Azure App Service Deploy" icon). An ellipsis button is located next to the deployment process step.

The right side of the screen shows the detailed configuration for "Environment 1". The "Environment name" field is set to "Environment 1". Under "Parameters", there is a link to "Unlink all". The "Azure subscription" dropdown is set to "howardmsp", with a note below stating "Scope: /subscriptions/81fca1b9-3108-45d6-9f98-1b263f64330c/resourcegroups/howardmsp". The "App type" dropdown is set to "Web App". The "App service name" dropdown is also set to "howardmsp".

**• Go back to the release page
• Click Pipeline**

The screenshot shows the 'New Release Definition' page in Azure DevOps. On the left, there are two cards: 'Artifacts' and 'Environments'. The 'Artifacts' card contains a card for 'Test_Website-ASP.NET-CI' with a gear icon highlighted by a red circle. The 'Environments' card shows 'Environment 1' with '1 phase, 1 task'. On the right, a callout box details the 'Continuous deployment trigger' for build 'Test_Website-ASP.NET-CI', stating it is 'Enabled' and creates a release every time a new build is available. Red arrows point from the gear icon on the artifact card to this callout box.

- Click on this icon and enable Continuous deployment trigger
- Then Click Save

The screenshot shows the Microsoft DevOps interface for a project named 'Test_Website'. The left sidebar has 'Builds' selected under 'Releases'. The main area displays a list of releases for a deployment group named 'HowardMsp'. The list includes four entries: 'Release-4', 'Release-3', 'Release-2', and 'Release-1', each associated with a specific build and branch.

A modal dialog titled 'Create new release' is open on the right. It contains sections for 'Pipeline' (with an 'Environment' step highlighted) and 'Artifacts' (with a source alias 'Test_Website-ASP.NET-CI' and version '20180223.5'). There is also a 'Release description' field and 'Create' and 'Cancel' buttons at the bottom.

- Click Create new release
- Click Create
- You should then see your release running
- Click on the Release

The screenshot shows the Microsoft DevOps interface for a project named 'Test_Website'. The top navigation bar includes links for Test_Website, Dashboards, Code, Work, Build and Release, Test, Wiki, and a gear icon. A search bar at the top right allows searching for work items in the project. The main content area is titled 'HowardMsp / Release-4'. It features tabs for Summary, Environments, Artifacts, Variables, General, Commits, Work items, Tests, Logs, and History. The 'Deploy' tab is currently selected. Below the tabs are buttons for Refresh, Deploy, Save, Abandon, and Send Email. A prominent 'Deploy to an environment' button is highlighted with a yellow box. The status bar indicates the release was triggered by 'Test_Website-ASPNET-CI' on 20180223.5 and was continuous deployment requested for Jeff Beauplan 2 days ago. The deployment log shows 'Test_Website-ASPNET-CI / 20180223.5 (Build)' on the master branch with a status of 'SUCCEEDED'. The 'Environments' section lists 'Environment 1' with actions, deployment status 'SUCCEEDED', triggered 2 days ago, completed 2 days ago, and no tests. The 'Issues' section states 'No issues reported in this release.' To the right, sections for 'Work items' (no associated work items found) and 'Tags' (with an 'Add...' button) are visible.

- When you click on your release it should open this page.
- At this point your environment isn't deployed yet so click deploy
- Once Deployment has Succeeded your website should be live! But where?

The screenshot shows the Azure portal interface for managing app services. On the left, there's a list of services with 'howardmsp' selected. The main area is the 'Overview' tab for 'howardmsp'. It displays basic information like the resource group ('howardmsp'), status ('Running'), location ('East US 2'), and subscription details ('Free Trial'). A red arrow points from the bottom of the list of steps to the 'URL' section, which contains the website address 'https://howardmsp.azurewebsites.net'.

Home > App Services > howardmsp

App Services

Microsoft Student Partners

+ Add Edit columns More

Filter by name...

NAME

howardmsp ...

TestWebsite-201802231... ...

Search (Ctrl+)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Browse Stop Swap Restart Delete Get publish profile Reset publish profile

Resource group (change)
howardmsp

Status
Running

Location
East US 2

Subscription (change)
Free Trial

Subscription ID
81fca1b9-3108-45d6-9f98-1b263f64330c

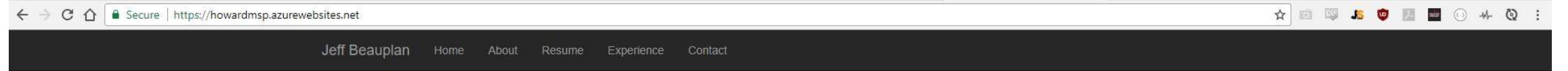
URL
<https://howardmsp.azurewebsites.net>

App Service plan/pricing tier
TestWebsite-20180223151357338-dev-asp (Basic: 1 Medium)

Continuous delivery status
--

Edit continuous delivery
https://jeffbeauplan.visualstudio.com/d2329335-4fac-45e6-8d4d-633aacbd3b4/_apps/hub...

- Under App Services in Azure select your app service and click overview
- The URL is the link to your website



Jeff's Portfolio Built on ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

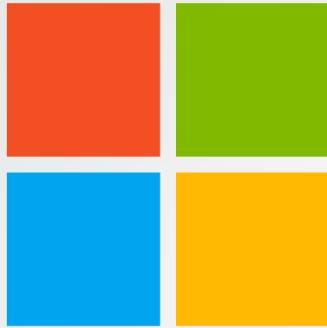
Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2018 - Jeff's ASP.NET Application

Done! Website is live! Congrats you've now built and deployed a website to the cloud



Microsoft

ASP.NET Workshop

By Howard University Microsoft Student Partners

Jeff Beauplan

Michelle Brown

Thank you for your time!



Join us on Facebook!

[Click here to join our Facebook group](#)