

Getting started with Azure App Service Web Apps and ASP.NET

This lab shows how to create an ASP.NET web application and deploy it to an Azure App Service Web App by using Visual Studio 2015 Preview. It assumes that you have no prior experience using Azure or ASP.NET. On completing the lab, you will have a simple web application up and running in the cloud.

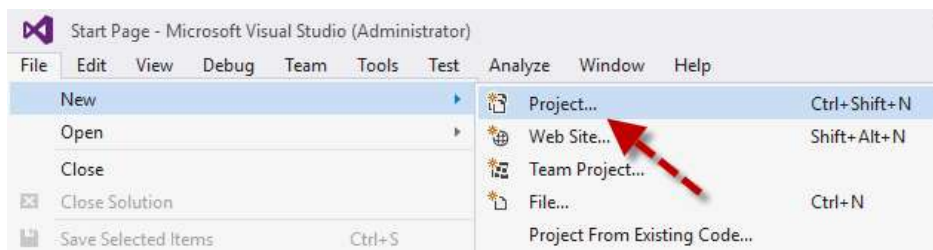
This lab includes the following sections:

1. [Create an ASP.NET web application in Visual Studio](#)
2. [Create a SQL Database online](#)
3. [Deploy the application to Azure](#)
4. [Make a change and redeploy](#)
5. [Monitor and manage the site in the portal](#)
6. [Appendix - Cleanup](#)

Create an ASP.NET web application

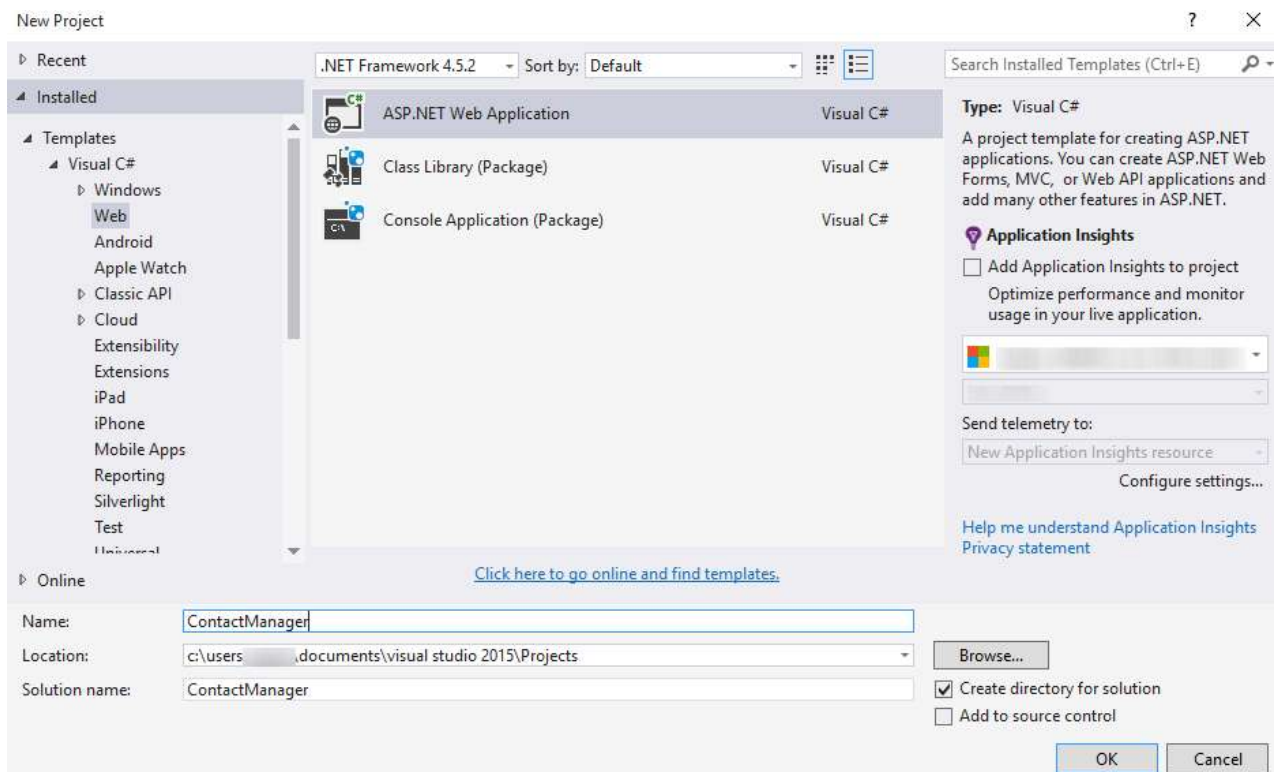
In this task you will create the web application that is going to be used throughout this lab.

1. Open Visual Studio. From the **File** menu, hover over the **New** option and click **Project**.



New Project in File menu

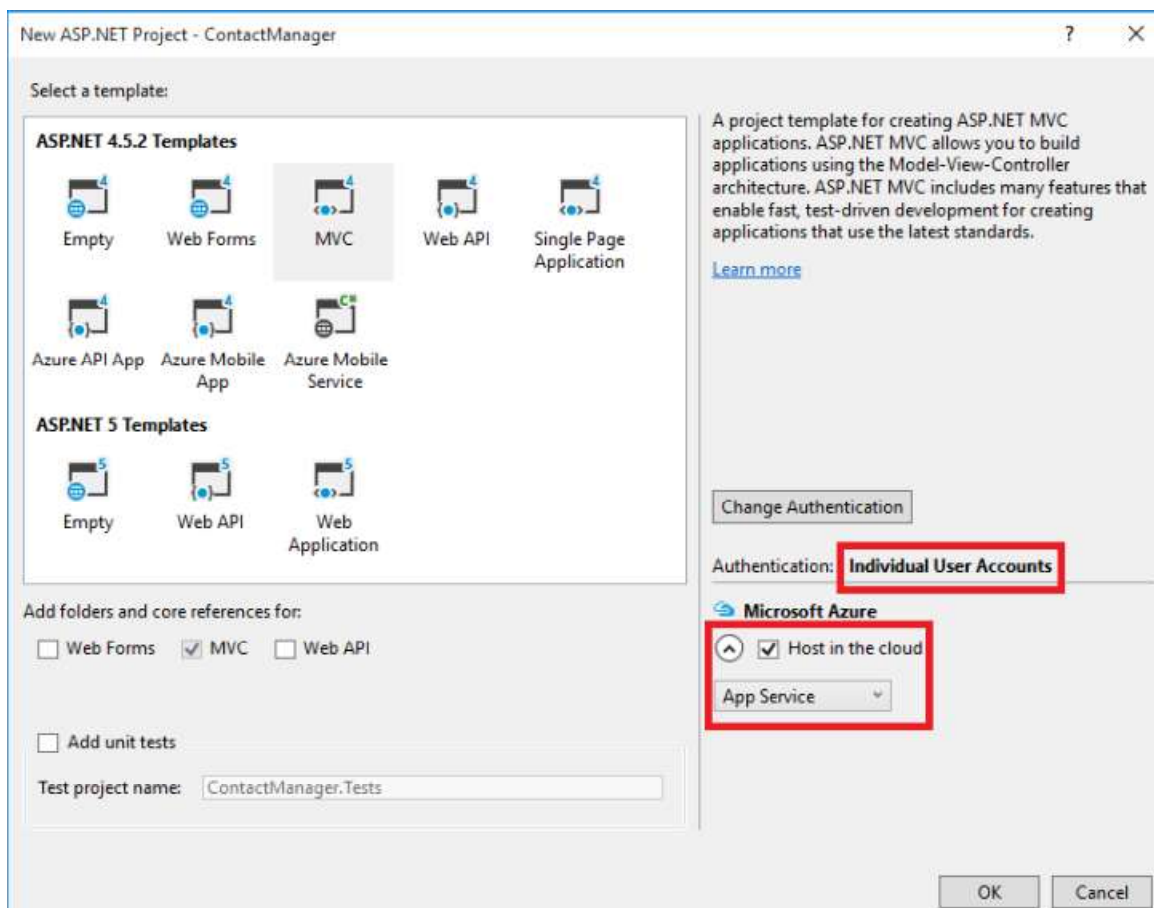
2. In the **New Project** dialog box, expand **C#** and select **Web** under **Installed Templates**, and then select **ASP.NET Web Application**.
3. Name the application **ContactManager** and click **OK**.



New Project dialog box


Note: Make sure you enter "ContactManager". Code blocks that you will be copying later assume that the project name is ContactManager.


- In the **New ASP.NET Project** dialog box, select the **MVC** template. Verify **Authentication** is set to **Individual User Accounts**, **Host in the cloud** is checked and **Web App** is selected. Then, click **OK**.






New ASP.NET Project dialog box

5. Once the application is selected you will get a pop-up to enter details for **Creating an App Service**. Once you enter the details like: WebApp Name, Subscription, Resource Group(if you have created earlier, if not create a Resource Group in Azure Portal and the list will be populated. Select the resource group)




**Create App Service**
Host your web and mobile applications, REST APIs, and more in Azure

 Microsoft account 

Hosting 

Services

Web App Name 


Subscription

Resource Group

App Service Plan

ContactManager20160112024541

New...



Clicking the Create button will create the following Azure resources

App Service - ContactManager20160112024541

If you have removed your spending limit or you are using Pay as You Go, there may be monetary impact if you provision additional resources.
[Learn More](#)


Export...


Create

Cancel

Details for Creating App Service

6. Click on **New** button under **App Service Plan**. In a Service plan we can select **App Service Plan Name**, **Location**, **Size** (can be Free, Shared(paid), and different configurations)



**Configure App Service Plan**
An App Service plan is the container for your app. The App...

App Service Plan

Location

Size

ContactManagerPlan

West US


Free


OK

Cancel

Configure Azure Web App - App Service Plan

7. Once the Service Plan is selected, select **Create** to create a **Web App Service**.





Create App Service

Host your web and mobile applications, REST APIs, and more in Azure

Hosting
Services

Web App Name
ContactManager20160112025522
[Change Type](#)

Subscription
▼

Resource Group
create-aspNet
▼

App Service Plan
ContactManagerPlan*
[New...](#)

Clicking the Create button will create the following Azure resources

App Service - ContactManager20160112025522

App Service Plan - ContactManagerPlan

If you have removed your spending limit or you are using Pay as You Go, there may be monetary impact if you provision additional resources.
[Learn More](#)

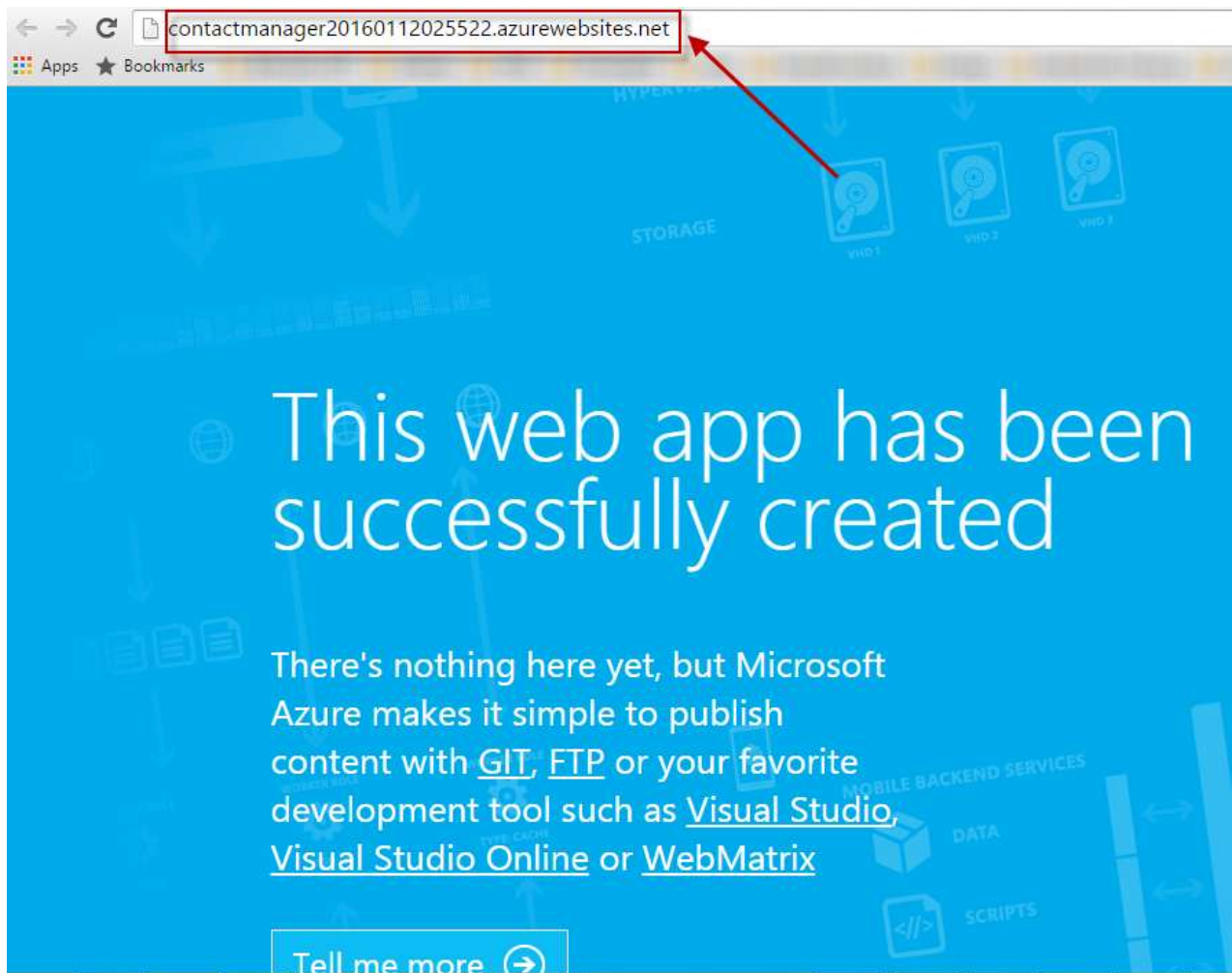
Export...

Create

Cancel

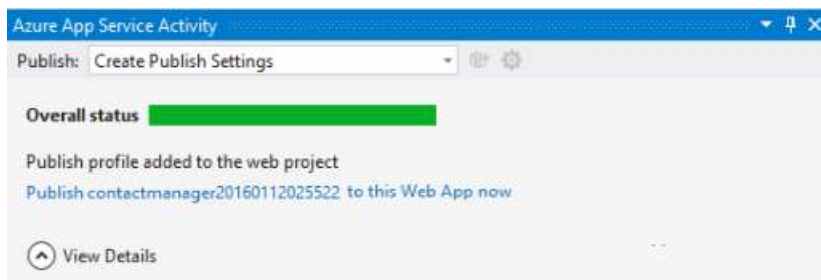
Configure Azure Web App - App Service Plan

8. As the application was created in Azure App Services, you can verify to your azure portal and check under the App Services Section you will have Contact Manager application hosted or navigate to the URL it will be **LIVE**



Created App Services - Web App

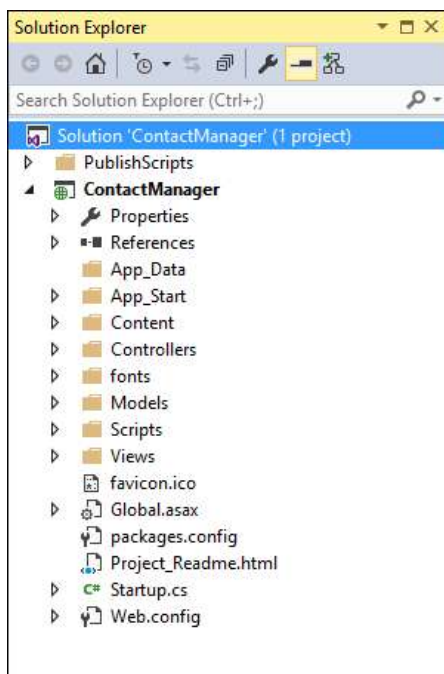
9. Once the App Service is created we are ready to publish the ASP.NET Application code, click on **Publish contactmanager to this web app now**



Configure Azure Web App - App Service Plan

In a few seconds, Visual Studio creates the web project in the folder you specified, and it creates the Web App in the Azure region you specified.

The **Solution Explorer** window shows the files and folders in the new project.



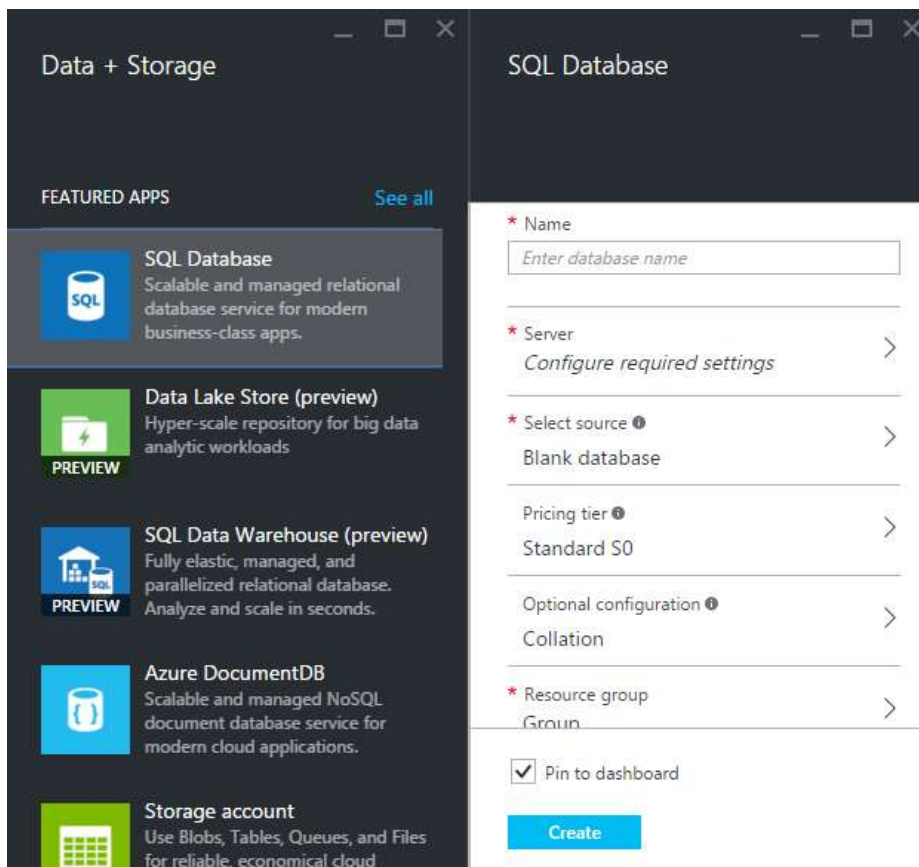
Solution Explorer

Create a SQL Database online

1. In your browser, go to <http://portal.azure.com>, and sign in with your Azure credentials.

On the left, click on **NEW** and **Data + Storage** and followed by **SQL Database** in the Data Storage section which is a database as a service.

2. Fill the details in **SQL Database** section.



Database details

3. Create a new **Server**(if you dont have an existing server) along with the Server Credentials, and select the **Source** as **Blank**, **Pricing Tier**, **Optional Configuration**, and select the existing **Resource Group** where we previously created App Service and Click **Create**

The screenshot displays the 'New server' configuration window in the Azure portal. The left pane, titled 'SQL Database', contains the following settings: Name (contactmanagerdemo), Server (Configure required settings), Select source (Blank database), Pricing tier (Standard S0), Optional configuration (Collation), and Resource group (create-aspnet). The middle pane, titled 'Server', shows options to 'Create a new server' or 'Use an existing server'. The right pane, titled 'New server', includes fields for Server name, Server admin login, Password, and Confirm password. The Location is set to 'West Europe'. At the bottom, there is a 'Create V12 server (Latest update)' button with 'Yes' and 'No' options.

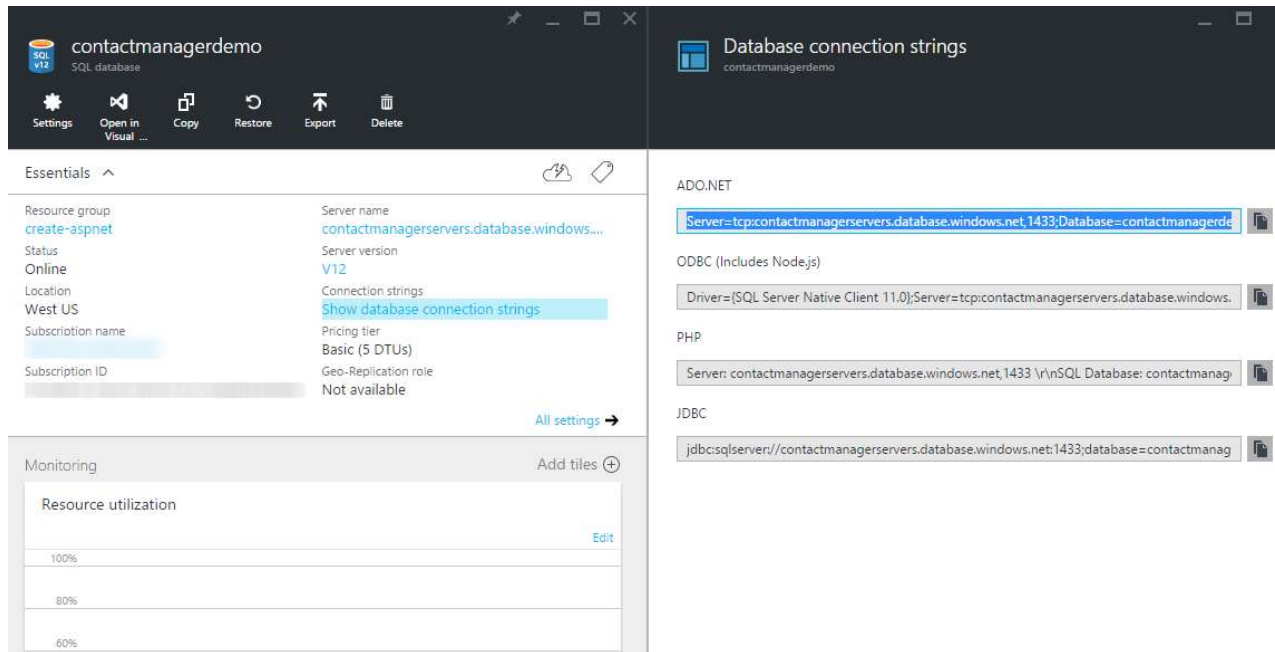
Create Database details

4. Once the SQL Database is created. Click on the SQL Databases at left pane. and select the database which we created just now.

The screenshot displays the 'contactmanagerdemo' SQL database details in the Azure portal. The top bar shows the database name and a toolbar with Settings, Open in Visual Studio, Copy, Restore, Export, and Delete. The 'Essentials' section displays key details: Resource group 'create-aspnet', Status 'Online', Location 'West US', Subscription name, Subscription ID, Server name 'contactmanagerservers.database.windows...', Server version 'V12', Connection strings, Pricing tier 'Basic (5 DTUs)', and Geo-Replication role 'Not available'. The 'Monitoring' section shows a 'Resource utilization' chart with a bar graph indicating 100%, 80%, and 60% utilization.

Created Database details

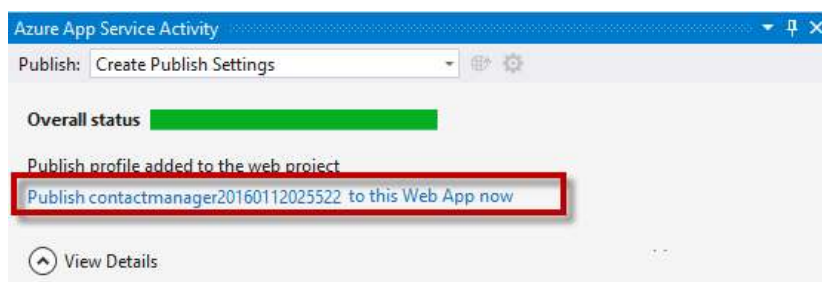
5. Copy the **Connection String** from the Azure portal which will be used when we deploy the App Server as we have to point the Connection String to this database.



Created Database details

Deploy the application to Azure

1. In the **Azure App Service Activity** window, click **Publish ContactManager to this Web App now**.



Azure App Service Activity Window

In a few seconds the **Publish Web** wizard appears.

The settings that Visual Studio needs to deploy your project to Azure have been saved in a *publish profile*. The wizard enables you to review and change those settings.

2. In the **Connection** tab of the **Publish Web** wizard, click **Validate Connection** to make sure that Visual Studio can connect to Azure in order to deploy the web project.

Publish Web

Profile

Connection

Settings

Preview

ContactManager20160112025522

Publish method: Web Deploy

Server: contactmanager20160112025522.scm.azurewebsites.net:443

Site name: ContactManager20160112025522

User name: \$ContactManager20160112025522

Password:

☒ Save password

Destination URL: http://contactmanager20160112025522.azurewebsites.net

Validate Connection


< Prev Next > Publish Close

Validating the connection

When the connection has been validated, a green check mark is shown next to the **Validate Connection** button.

- Click **Next**.

Destination URL: http://contactmanager20160112025522.azurewebsites.net

Validate Connection 

Successfully validated connection

- The **Settings** tab is shown. If you expand **File Publish Options** you will see several settings that enable you to handle scenarios that don't apply to this lab:

- **Remove additional files at destination.**

Deletes any files at the server that aren't in your project. You might need this if you were deploying a project to a site that you had deployed a different project to earlier.

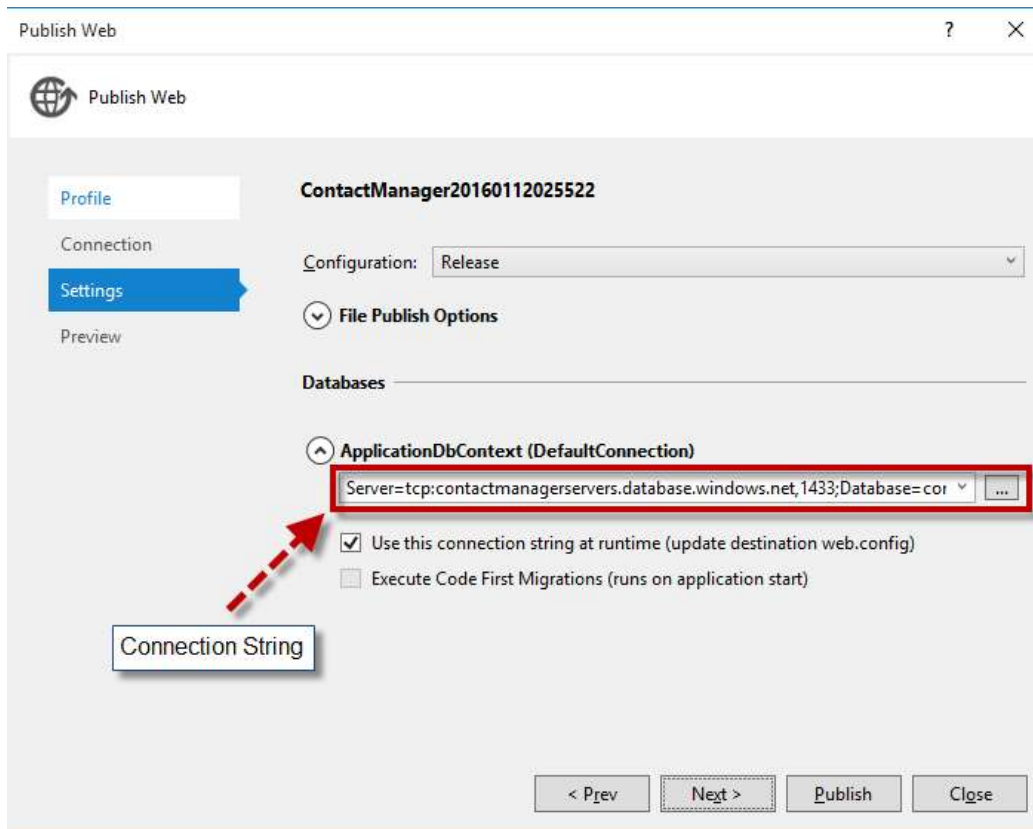
- **Precompile during publishing.**

Can reduce first-request warm up times for large sites.

- **Exclude files from the App_Data folder.**

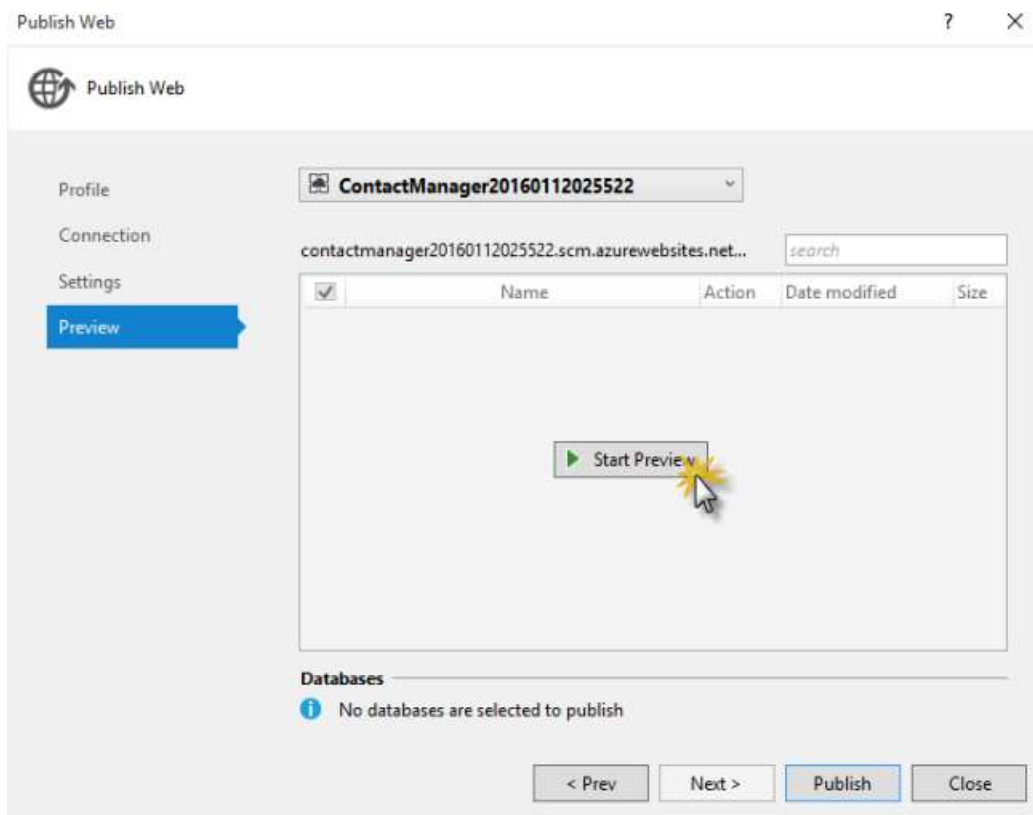
For testing you sometimes have a SQL Server database file in App_Data which you don't want to deploy to production.

In this case, keep the default values for **Configuration** and **File Publish Options** and click **Next**.



Settings tab

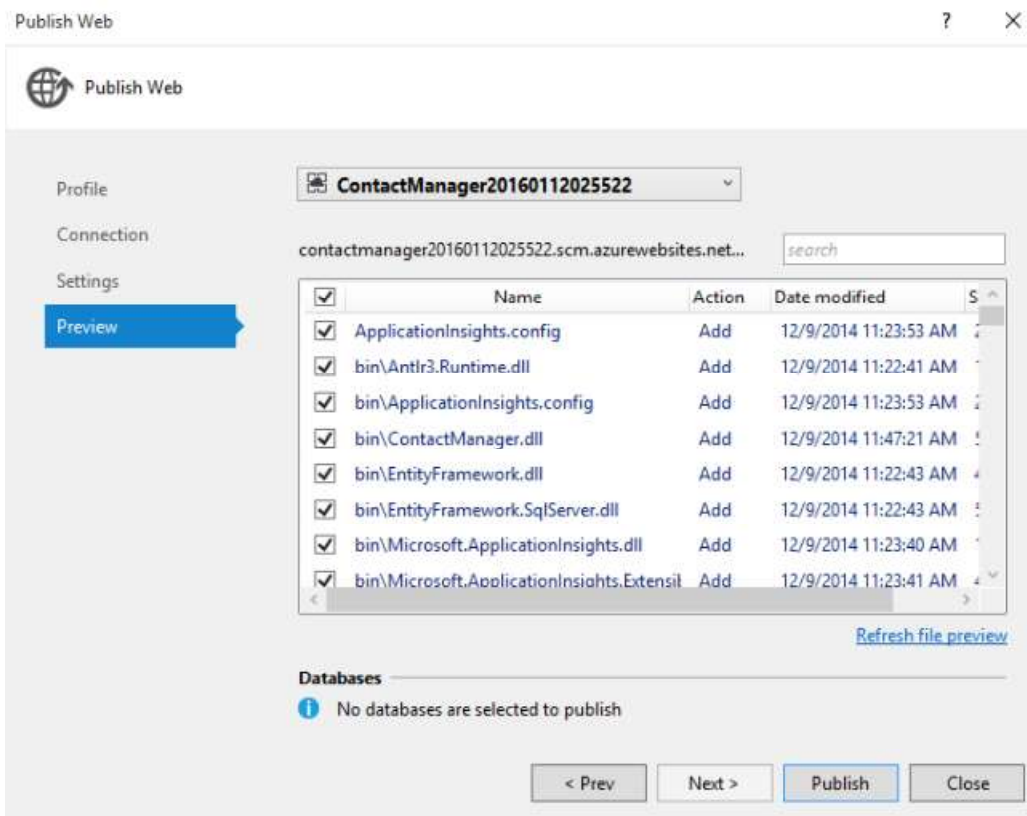
- In the **Preview** tab, click **Start Preview**.



Start Preview button - Preview tab

The tab displays a list of the files that will be copied to the server. Displaying the preview isn't required to publish the application but it's a useful function to be aware of.

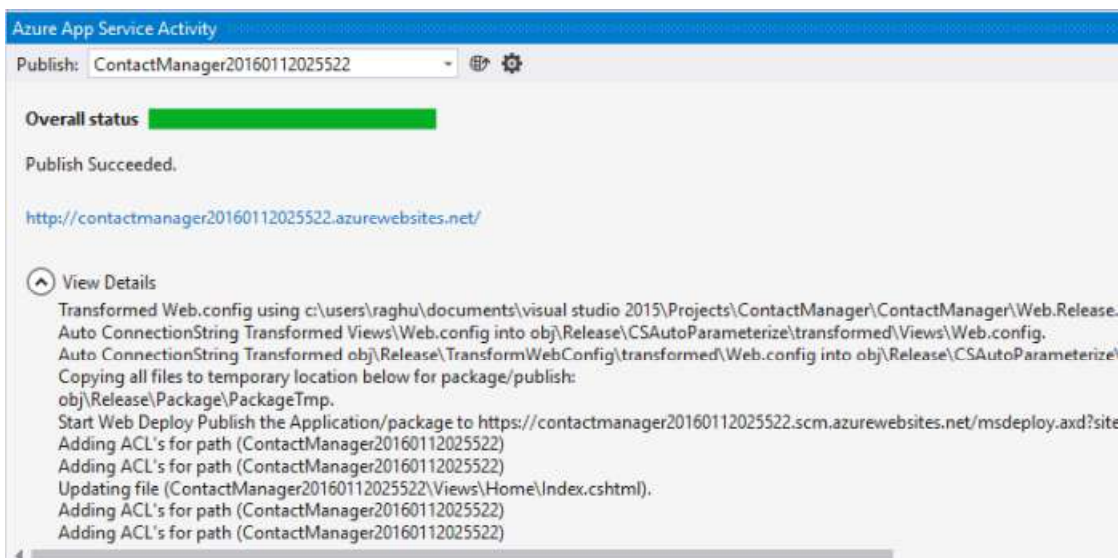
- Click **Publish**.



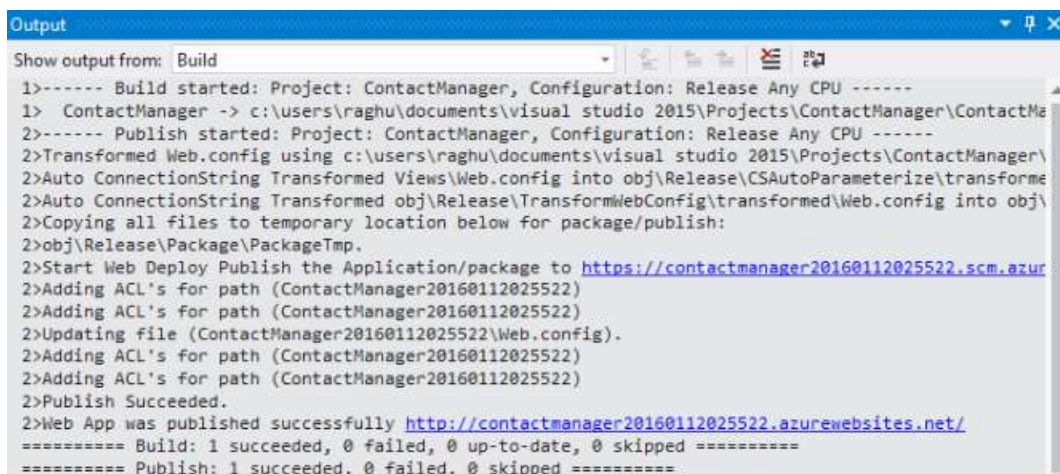
File Preview

Visual Studio begins the process of copying the files to the Azure server.

The **Output** and **Azure App Service Activity** windows show what deployment actions were taken and report successful completion of the deployment.



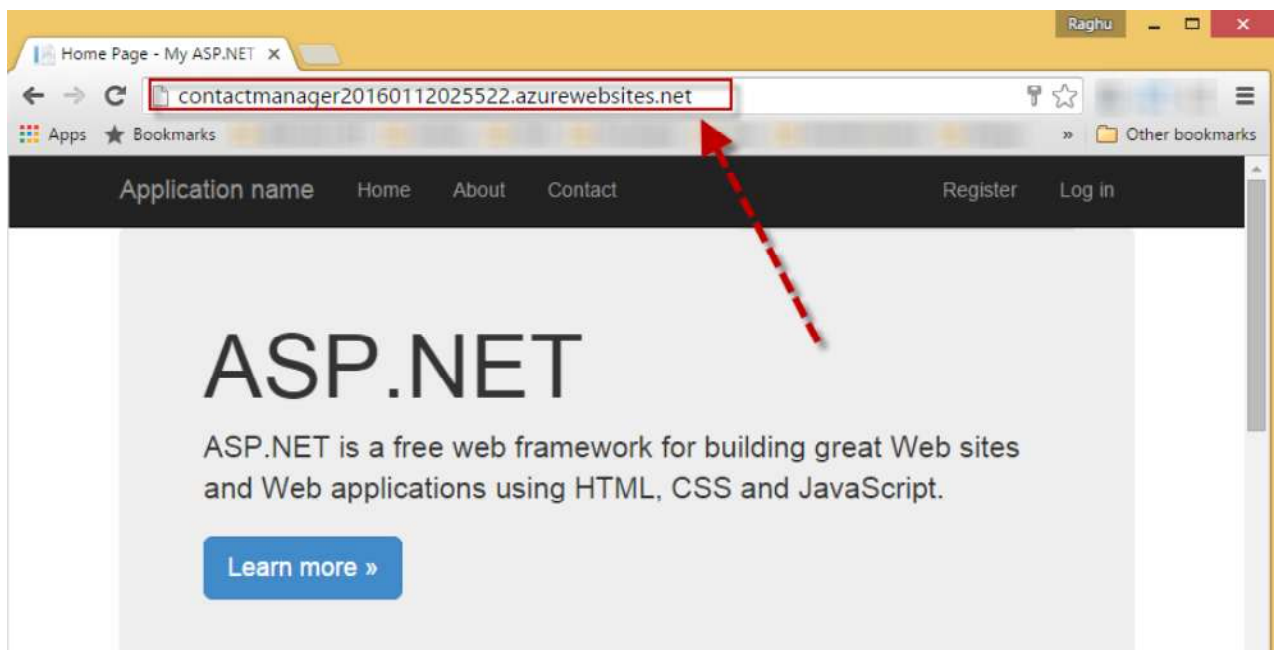
Azure App Service Activity reporting successful deployment



```
Output
Show output from: Build
1>----- Build started: Project: ContactManager, Configuration: Release Any CPU -----
1> ContactManager -> c:\users\raghu\documents\visual studio 2015\Projects\ContactManager\ContactMa
2>----- Publish started: Project: ContactManager, Configuration: Release Any CPU -----
2>Transformed Web.config using c:\users\raghu\documents\visual studio 2015\Projects\ContactManager\
2>Auto ConnectionString Transformed Views\Web.config into obj\Release\CSAutoParameterize\transforme
2>Auto ConnectionString Transformed obj\Release\TransformWebConfig\transformed\Web.config into obj\
2>Copying all files to temporary location below for package/publish:
2>obj\Release\Package\PackageTmp.
2>Start Web Deploy Publish the Application/package to https://contactmanager20160112025522.scm.azur
2>Adding ACL's for path (ContactManager20160112025522)
2>Adding ACL's for path (ContactManager20160112025522)
2>Updating file (ContactManager20160112025522\Web.config).
2>Adding ACL's for path (ContactManager20160112025522)
2>Adding ACL's for path (ContactManager20160112025522)
2>Publish Succeeded.
2>Web App was published successfully http://contactmanager20160112025522.azurewebsites.net/
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
===== Publish: 1 succeeded, 0 failed, 0 skipped =====
```

Output Window reporting successful deployment

Upon successful deployment, the default browser automatically opens to the URL of the deployed Web App, and the application that you created is now running in the cloud. The URL in the browser address bar shows that the site is being loaded from the Internet.



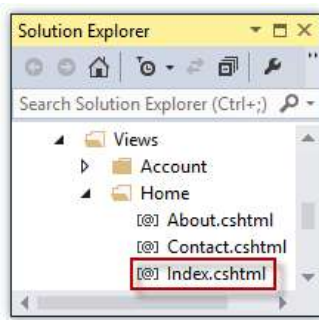
Web App running in Azure

7. Close the browser.

Make a change and redeploy

In this task, you will change the **h1** heading of the home page, run the project and redeploy on your App Service on Azure.

1. Open the *Views/Home/Index.cshtml* file in **Solution Explorer**, change the **h1** heading from "ASP.NET" to "ASP.NET and Azure", and save the file.



Index.cshtml

```

@{
    ViewBag.Title = "Home Page";
}

<div class="jumbotron">
    <h1>Welcome ASP.NET from Azure Web App Service</h1>
    <p class="lead">ASP.NET is a free web framework for building great Web sites and Web ap
    <p><a href="http://asp.net" class="btn btn-primary btn-lg">Learn more &raquo;</a></p>
</div>

<div class="row">

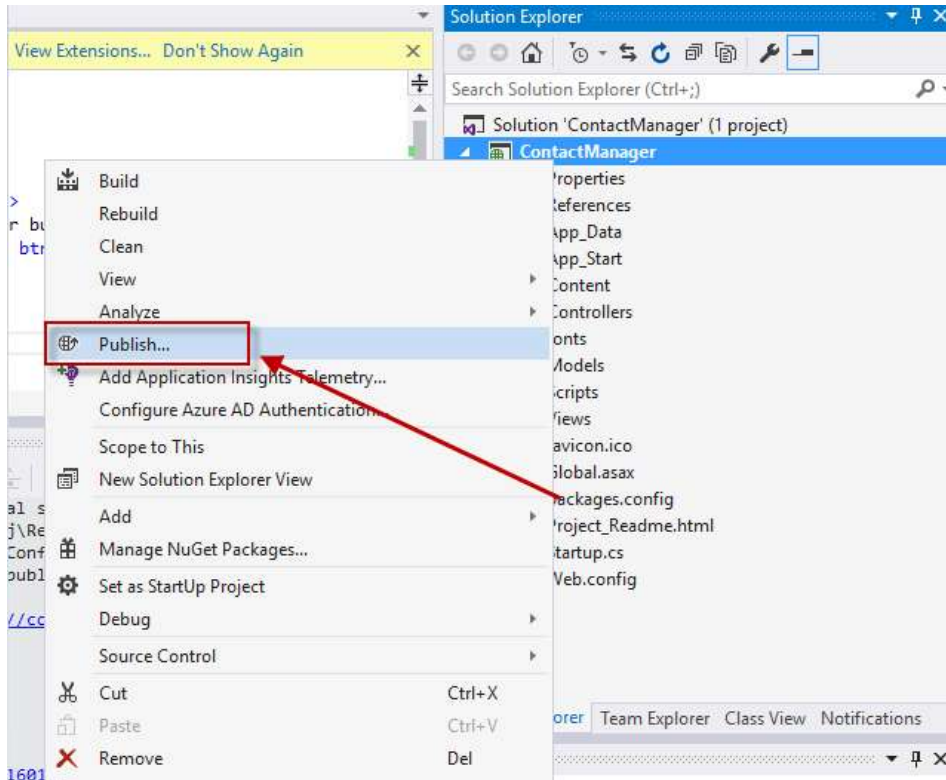
```

Changing the page's heading

2. Press **CTRL+F5** to see the updated heading by running the site.

The **http://localhost** URL shows that it's running on your local computer. By default it's running in IIS Express, which is a lightweight version of IIS designed for use during web application development.

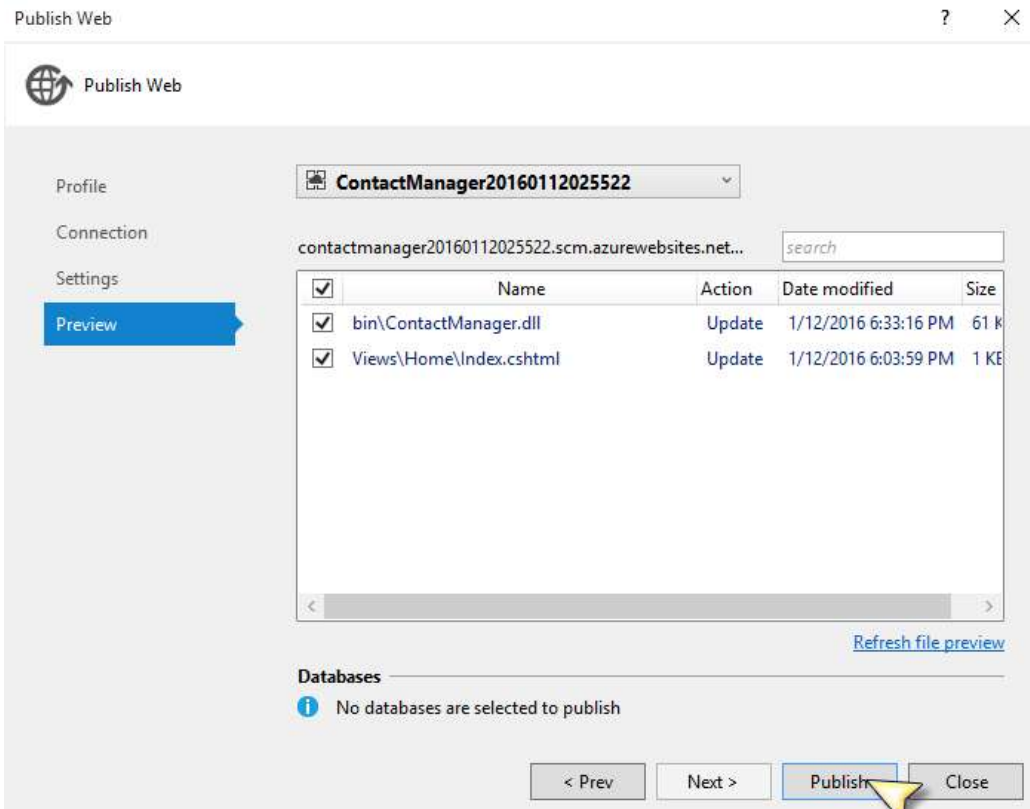
3. Close the browser.
4. In **Solution Explorer**, right-click the project, and choose **Publish**.



Preparing a new deployment

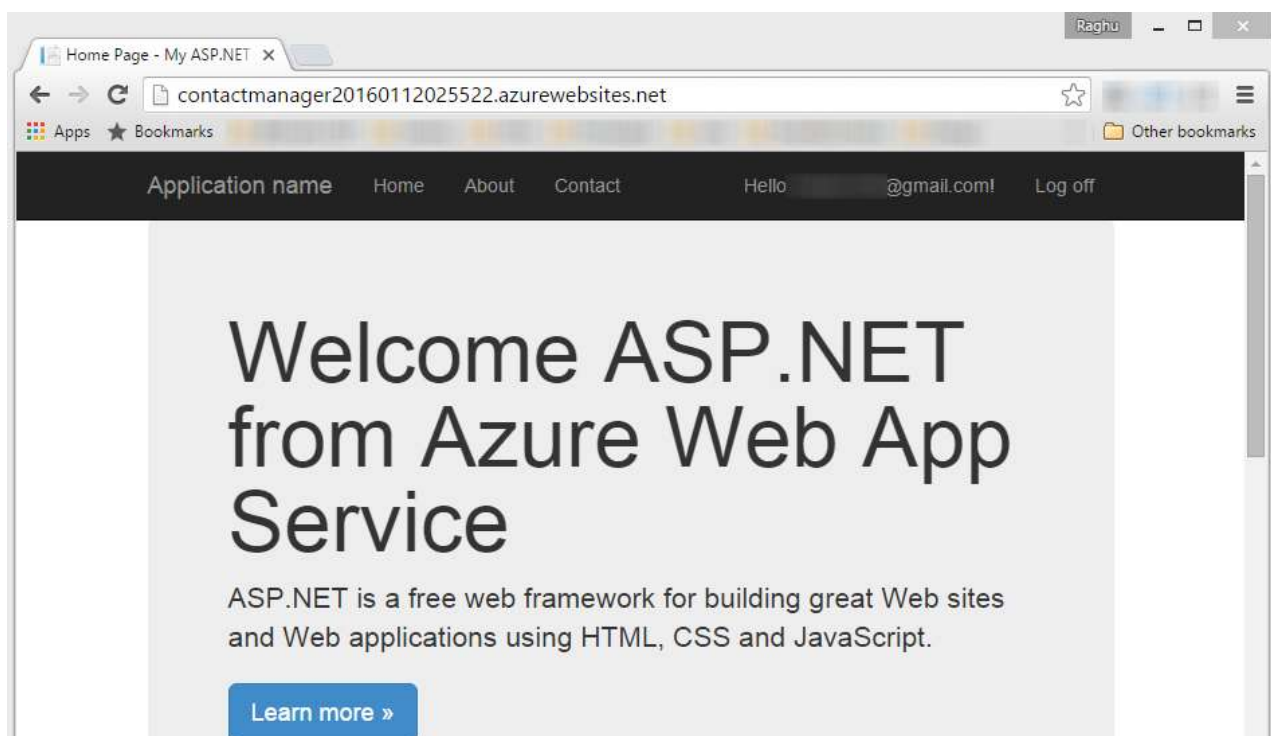
The Preview tab of the **Publish Web** wizard appears. If you needed to change any publish settings you could choose a different tab, but now all you want to do is redeploy with the same settings.

5. In the **Publish Web** wizard, click **Publish**.



Publish Web Wizard

Visual Studio deploys the project to Azure and opens the site in the default browser.



Changes deployed

Monitor and manage the site in the Azure portal

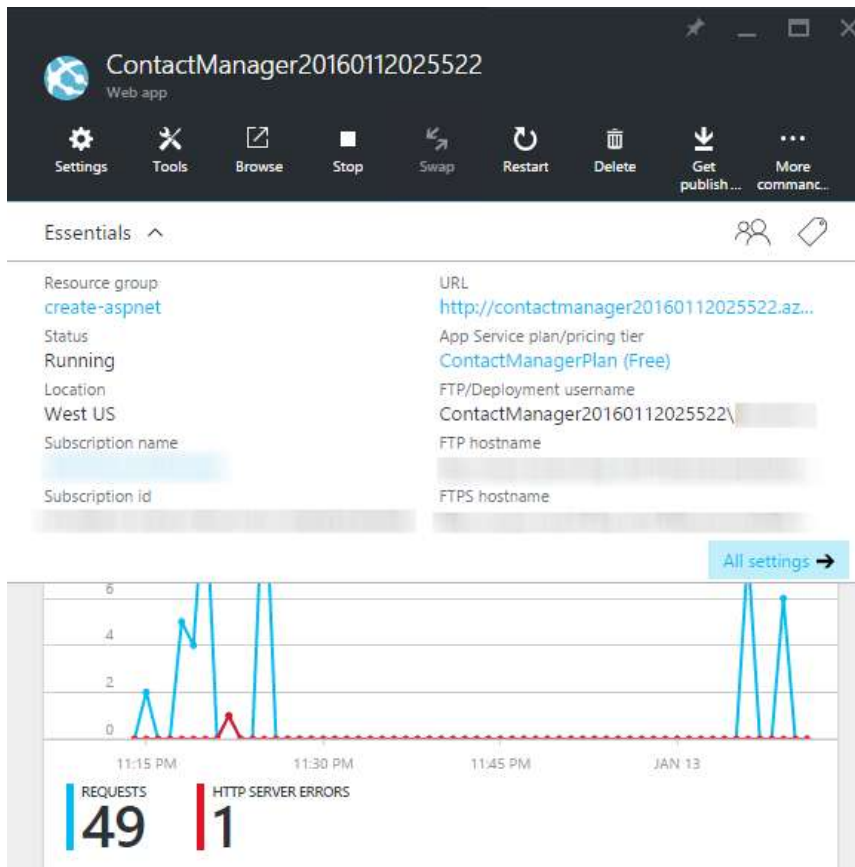
The [Azure Portal](#) is a web interface that enables you to manage and monitor your Azure services, such as the App Service Web App you just created. In this task you will look at some of what you can do in the portal.

1. In your browser, go to <http://portal.azure.com>, and sign in with your Azure credentials.

On the left, click on **App Services** and the Web App Services are listed.

2. Click the name of your Web App on the blade.
3. This blade gives you an overview of your Web App.

You can access monitoring information, change the pricing tier, configure continuous deployment and various other tasks.

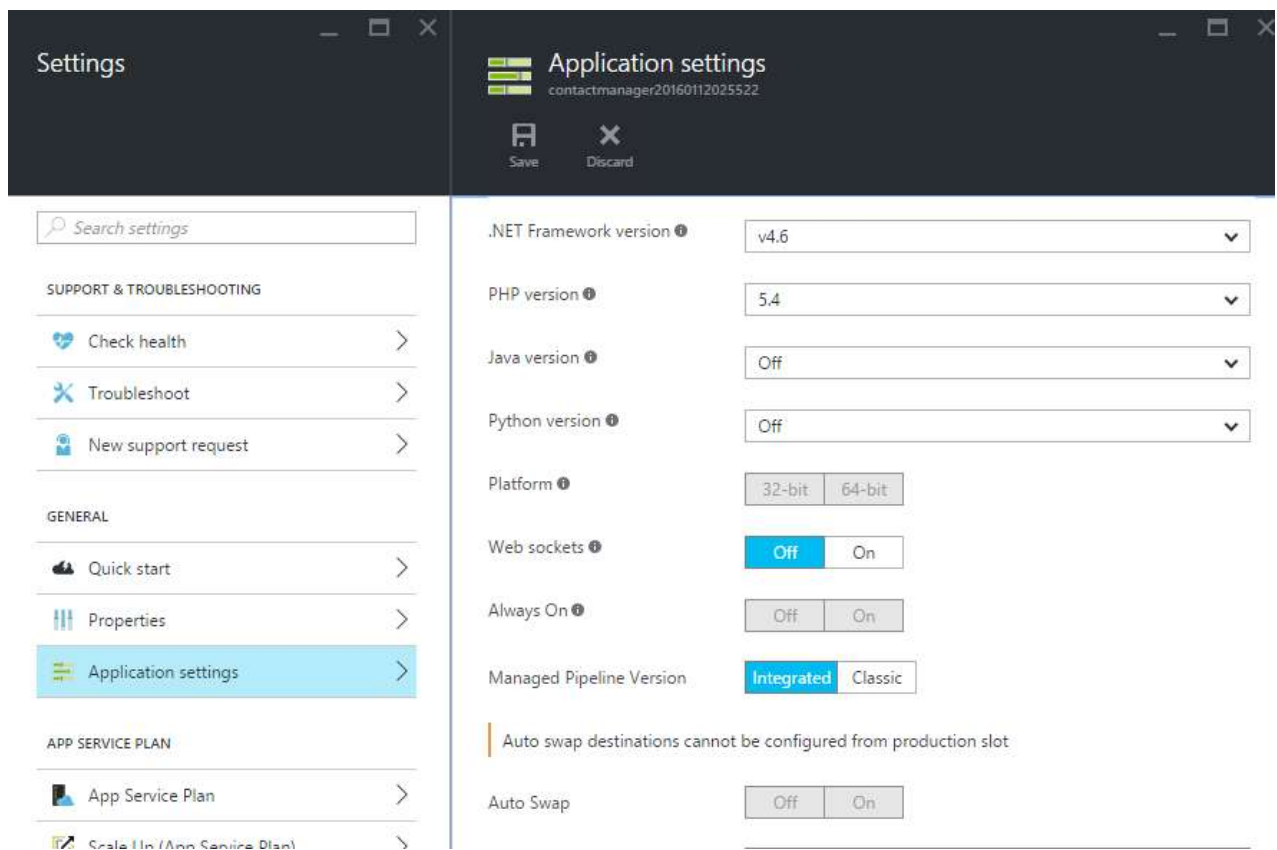


Web App's overview blade

At this point your site hasn't had much traffic and may not show anything in the graph. If you browse to your application, refresh the page a few times, and then come back again, it will show up some traffic.

4. Click at **Settings** at the top of the blade. Then click **Application Settings**.

The [Application Settings](#) blade enables you to control the .NET version used for the Web App, enable features such as [WebSockets](#) and [diagnostic logging](#), set [connection string values](#), and more.

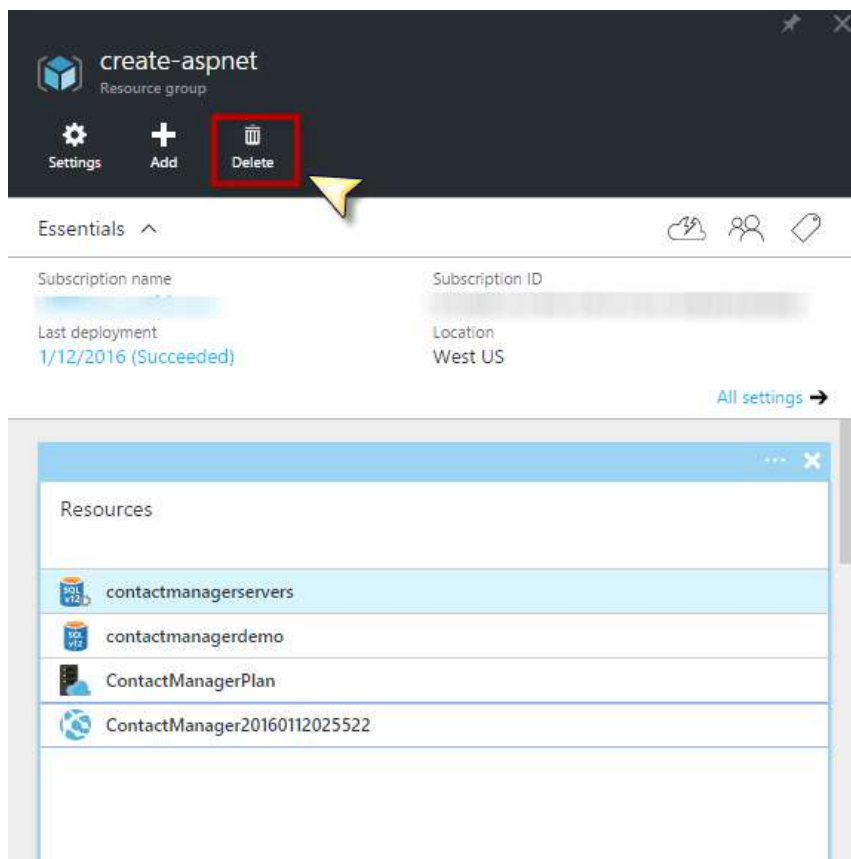


WebApp's settings blade

Appendix - Cleanup

In this task you will learn how to delete the Web App published in the previous section.

1. In your browser, go to <http://portal.azure.com>, and sign in with your Azure credentials.
2. Click on **Resource Groups** at left pane which will display all the resource groups created, Select the Resource group created earlier in this Lab.
3. Click **Delete** in the top bar.



Clicking Delete Resource Group

4. In the **Delete app** confirmation dialog, click **Yes**.

This will delete the Resource Group to which all resources are intact(i.e, App Service, App Service Plan, SQL Database, Server).

Summary

In this lab you have seen how to create a simple web application and deploy it to an Azure App Service Web App. You also made a quick tour around the Azure Portal.