Azure Website Deployment with Visual Studio Ultimate 2013

This HOL will walk you through a day in the life of a web applications developer and implementing a continuous integration development model. The continuous integration environment is created by leveraging the continuous integration features that are part of Visual Studio Online.

During this HOL, you will use Visual Studio 2013 that is connected to your Visual Studio Online (VSO) instance to develop, build, and deploy a modern MVC Web Application. You will walk through the individual steps of creating a web application, adding the web application into a VSO source control repository, and enabling continuous integration deployment when checking in new code changes.

Prerequisites

- In order to complete this lab you will need **Visual Studio 2103 Premium or Ultimate**, **Azure SDK 2.4 RTM Release**, **Visual Studio Online Subscription**, and a **Windows Azure Subscription**.
- Visual Studio 2013 Update 3 - http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs#d-visual-studio-2013-update
- Windows Azure SDK 2.4 RTM Release - http://go.microsoft.com/fwlink/?LinkId=404332
- Visual Studio Online Subscription - http://www.visualstudio.com/
- Windows Azure Free Trial Subscription - http://azure.microsoft.com/en-us/pricing/free-trial

Exercises

1. This hands-on lab includes the following exercises:
2. Connecting Visual Studio Team Explorer to your Visual Studio Online Collection
3. Creating a Source Control Workspace to manage source code
4. Creating a Web Application
5. Integrating Visual Studio Online Source Control with Azure
6. Making Modifications to the Web Application
7. Review Visual Studio Online Build Definition
8. Building and Deploying the Web Application

Expected duration: 60 minutes

Exercise 1: Connecting Visual Studio Team Explorer to your Visual Studio Online Collection

In this exercise, you will configure Team Explorer to connect to your Visual Studio Online hosted TFS Team Project Collection.

1. Launch **Visual Studio 2013** from the taskbar and open **Team Explorer**.

2. Select the **Connect to Team Projects** button.

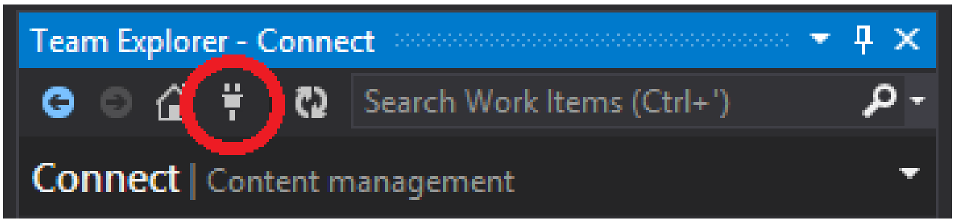


Figure 1: Connecting to a different team project

3. In Team Explorer, click on Select Team Projects.

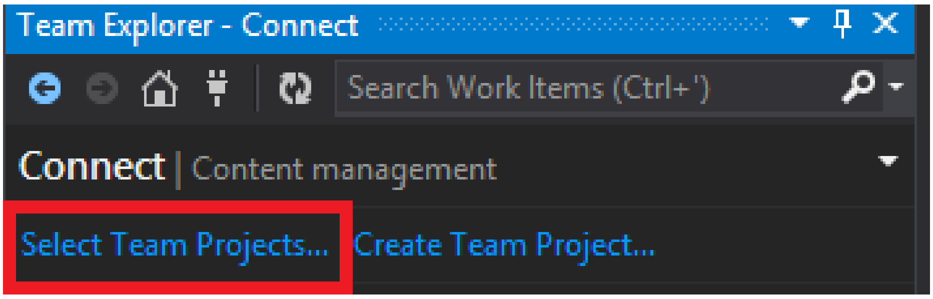
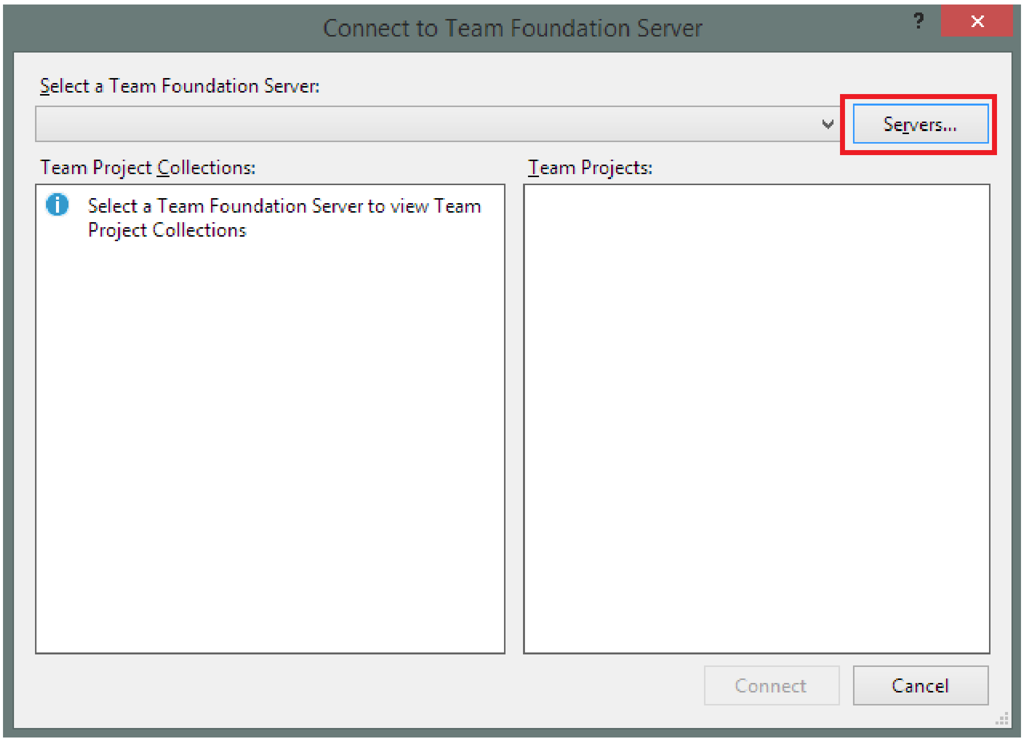


Figure 2: Selecting Team Projects

4. Click the Servers button on the Connect to Team Foundation Server window.


Connect to Team Foundation Server
Select a Team Foundation Server:
Servers...
Team Project Collections:
Team Projects:
Connect
Cancel

Select a Team Foundation Server to view Team Project Collections

Figure 3: Adding new Server

5. In the Add/Remove Team Foundation Server window, click the **Add** button to add a new TFS Server address.

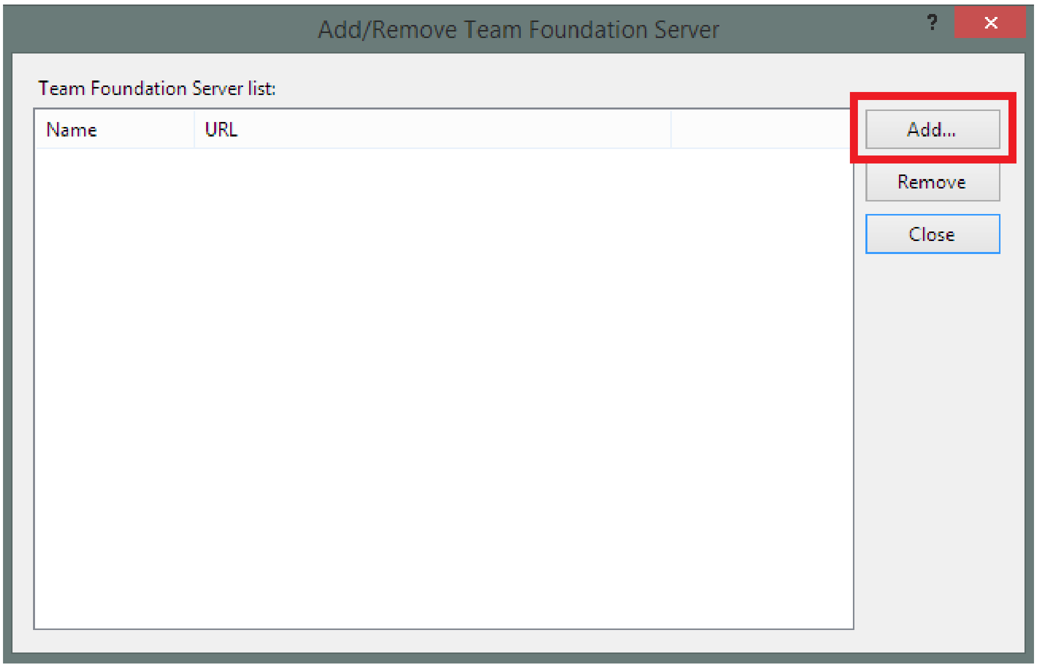
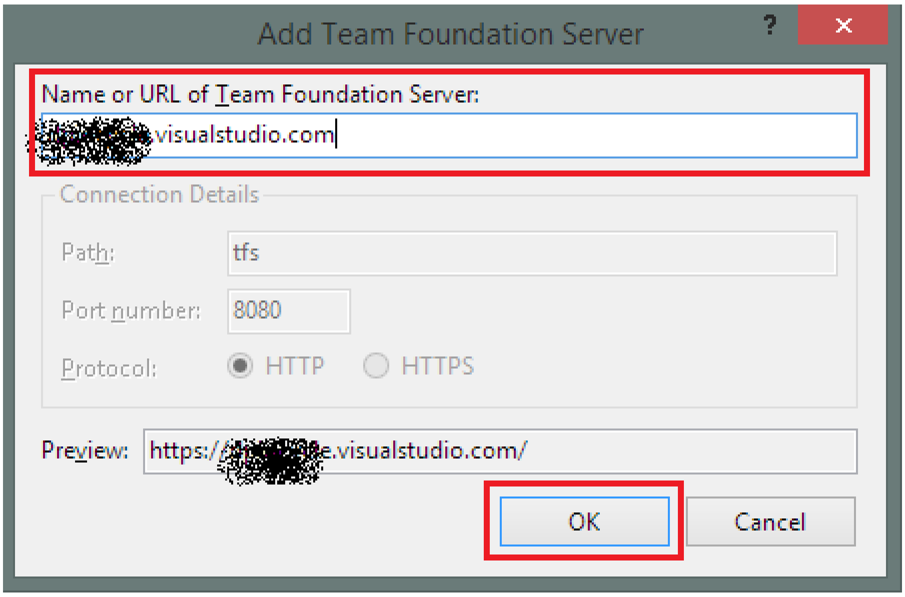

Add/Remove Team Foundation Server
Team Foundation Server list:
Name
URL
Add...
Remove
Close

Figure 4: Add new TFS Server

6. In the Add/Remove Team Foundation window, enter the name of your VSO subscription, such as **myvso.visualstudio.com** as the **URL of the Team Foundation Sever**. Select the **OK** button to continue.



Name or URL of Team Foundation Server: [redacted]visualstudio.com

Path: tfs

Port number: 8080

Protocol: ● HTTP ○ HTTPS

Preview: https://[redacted]e.visualstudio.com/

Figure 5: New TFS Server Connection Values

7. If prompted login to your VSO subscription with the Microsoft account used for the VSO subscription.
8. Make sure that **your VSO Subscription URL** is selected in the **Team Foundation Server List** in the Add/Remove Team Foundation Sever window and then click on **Close**.
9. In the Connect to Team Foundation Server window select the **DefaultCollection** in the **Team Project Collections** list and select **(Select All)** project in the **Team Projects** list.
10. Select the **Connect** button to continue.

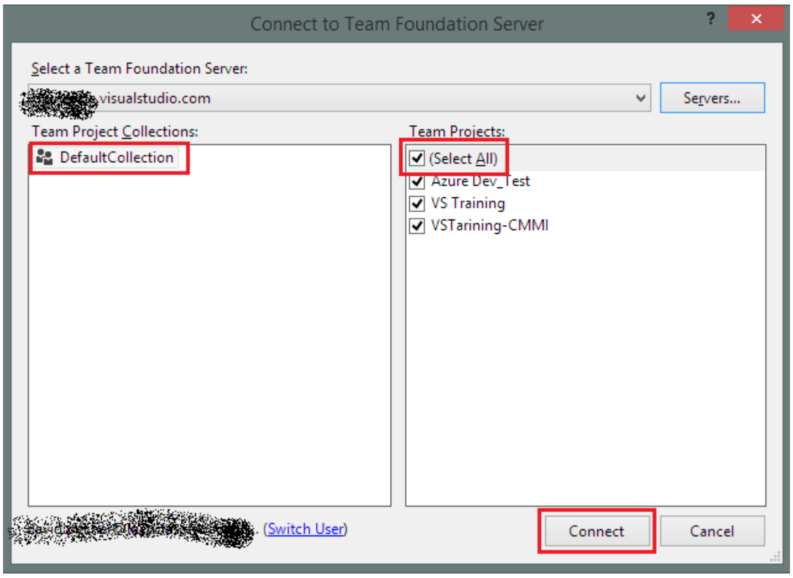


__Figure 6:__ Connecting to VSO Collection project

11. In the Team Explorer view, click on **Home** and select Projects and My Teams and then click on New Team Project.

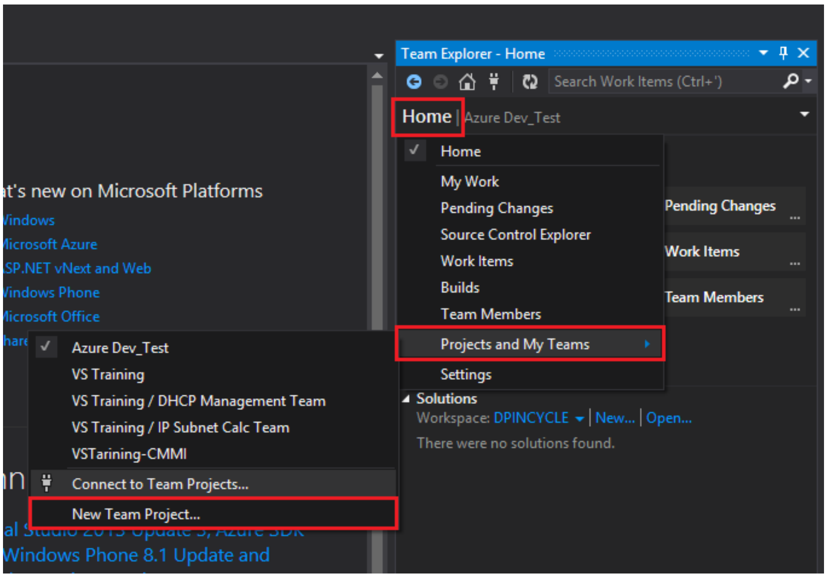
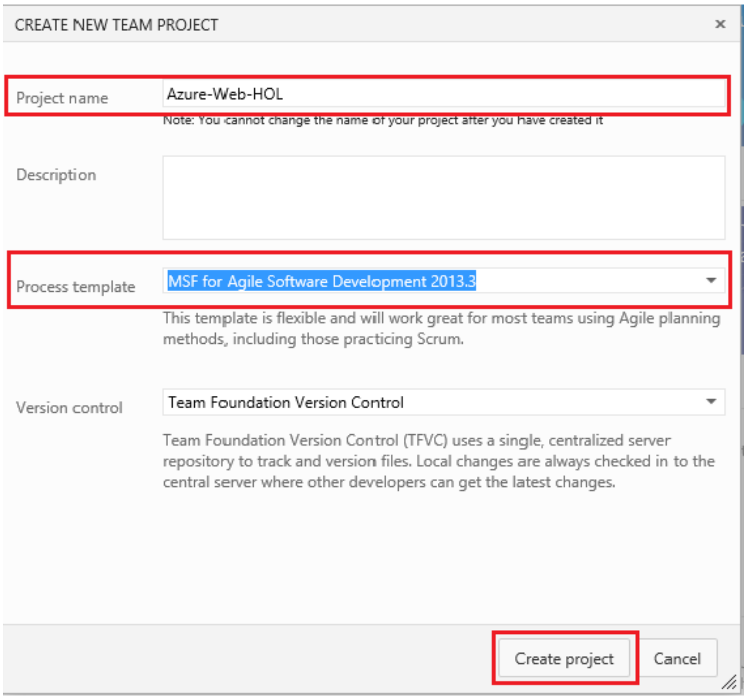


__Figure 7__: Create new team project

12. A new instance of Internet Explorer will open and prompt for login credentials for your VSO subscription, enter the credentials and proceed forward

13. The Create New Team Project web dialog will now be displayed, in the dialog **enter** the value of **Azure-Web-HOL** for the Project name and **select** the **MSF for Agile Software Development 2013.3** Process template and finally **click** on the **Create** project button.



CREATE NEW TEAM PROJECT

Project name: Azure-Web-HOL

Note: You cannot change the name of your project after you have created it

Description:

Process template: MSF for Agile Software Development 2013.3

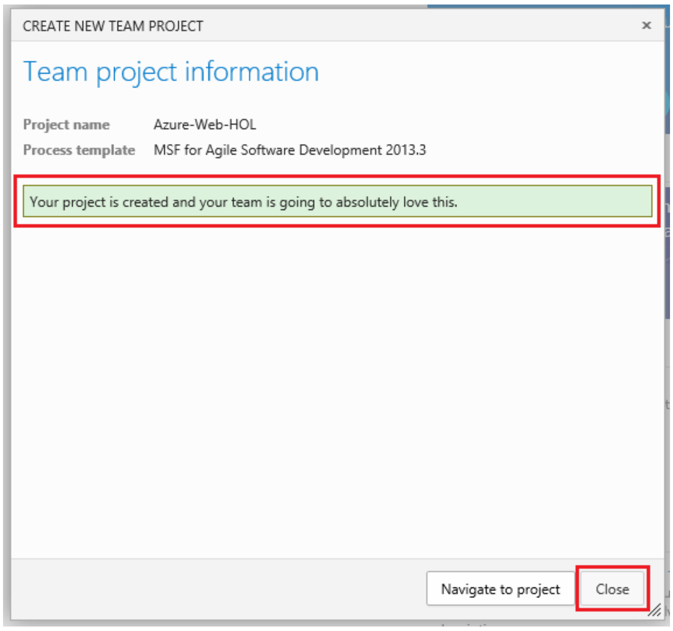
This template is flexible and will work great for most teams using Agile planning methods, including those practicing Scrum.

Version control: Team Foundation Version Control

Team Foundation Version Control (TFVC) uses a single, centralized server repository to track and version files. Local changes are always checked in to the central server where other developers can get the latest changes.

Figure 8*: New Team Project properties*

14. Once the team project is complete, click the **Close** button.



CREATE NEW TEAM PROJECT

Team project information

Project name: Azure-Web-HOL

Process template: MSF for Agile Software Development 2013.3

Your project is created and your team is going to absolutely love this.

Figure 9*: Team Project created*

15. Switch back to Visual Studio 2013 and in **Team Explorer** click on **Select Team Projects**.

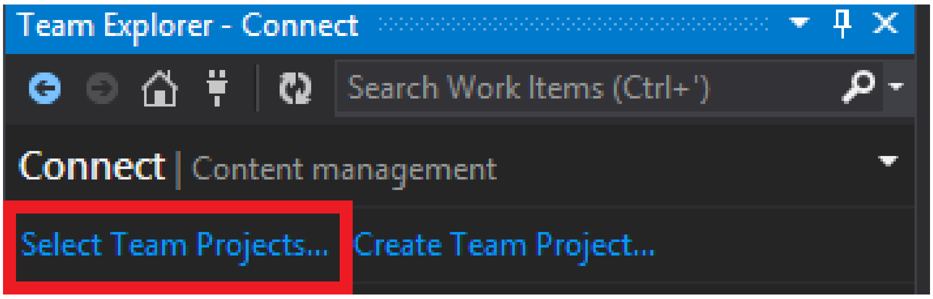
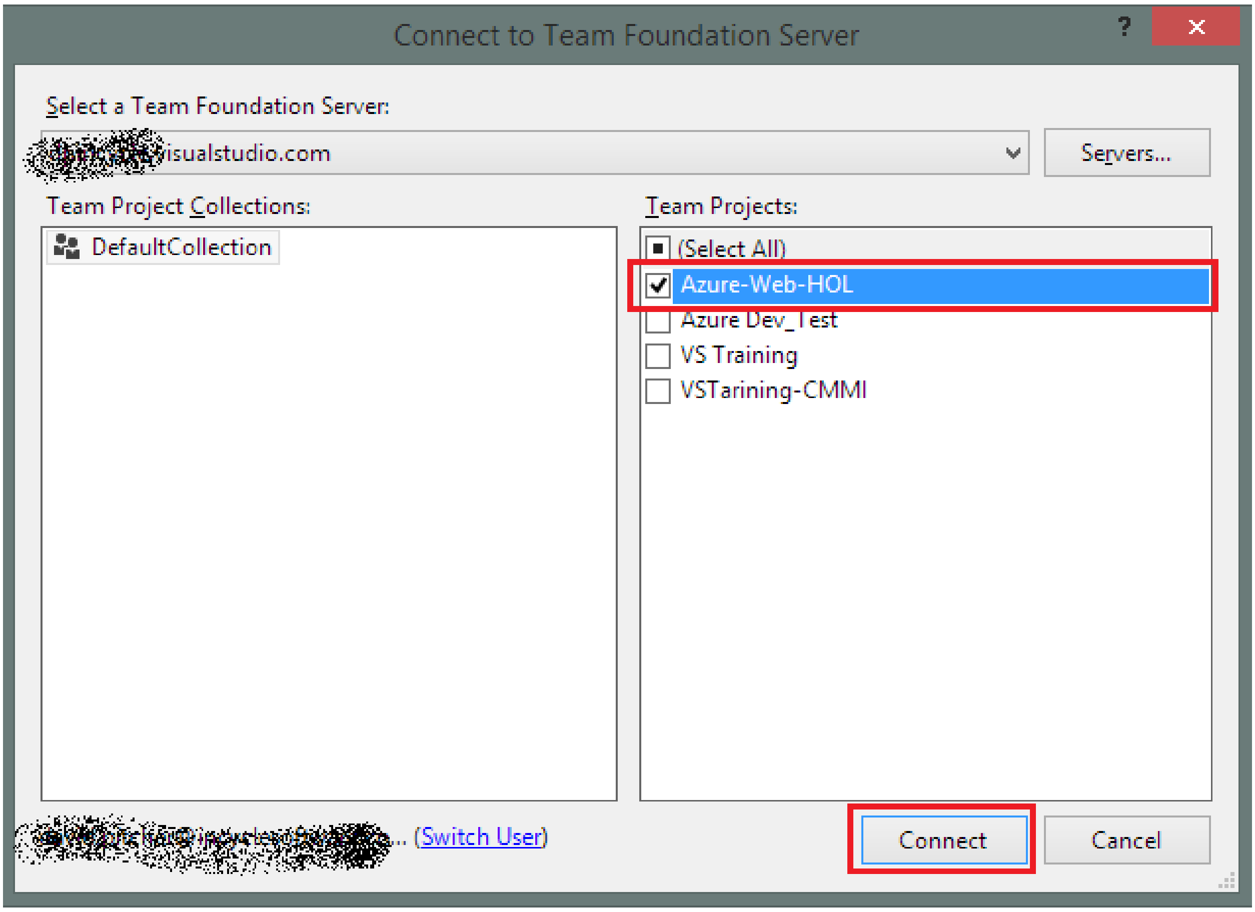


Figure 10: Selecting Team Projects

In the Connect to Team Foundation Server window, de-select all selected projects, then select the Azure-Web-HOL team project from the list of Team Projects, and finally click on the Connect button.



Connect to Team Foundation Server

Select a Team Foundation Server:

[redacted]isualstudio.com

Team Project Collections:

- DefaultCollection

Team Projects:

- ■ (Select All)
- ☑ Azure-Web-HOL
- □ Azure Dev_Test
- □ VS Training
- □ VSTarining-CMMI

[redacted]... (Switch User)

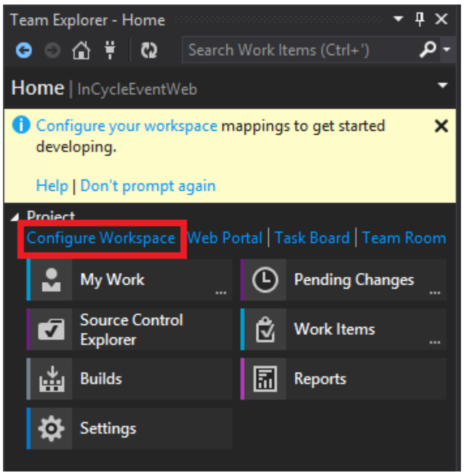
Figure 11: Selecting Azure-Web-HOL project

16. In the Team Explorer window, click on the Home icon.

Exercise 2: Creating a Source Control Workspace to manage source code

In this exercise, you will learn create a mapping between the TFS Source Control repository for the TFS Team Project and your local computer disks.

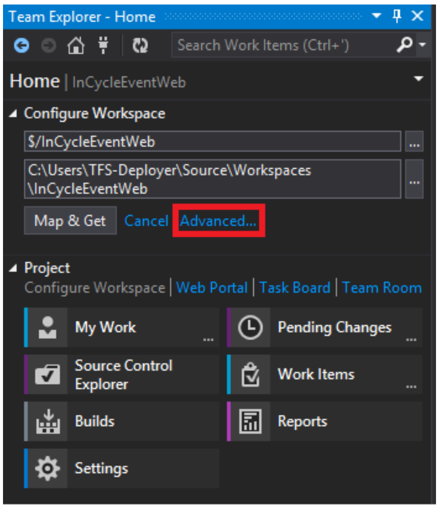
1. Within Team Explorer click on Configure Workspace located under the Project node.



Configure your workspace mappings to get started developing.

Figure 12: Configure new workspace

2. Click on the Advanced… link to bring up the advanced workspace configuration window.

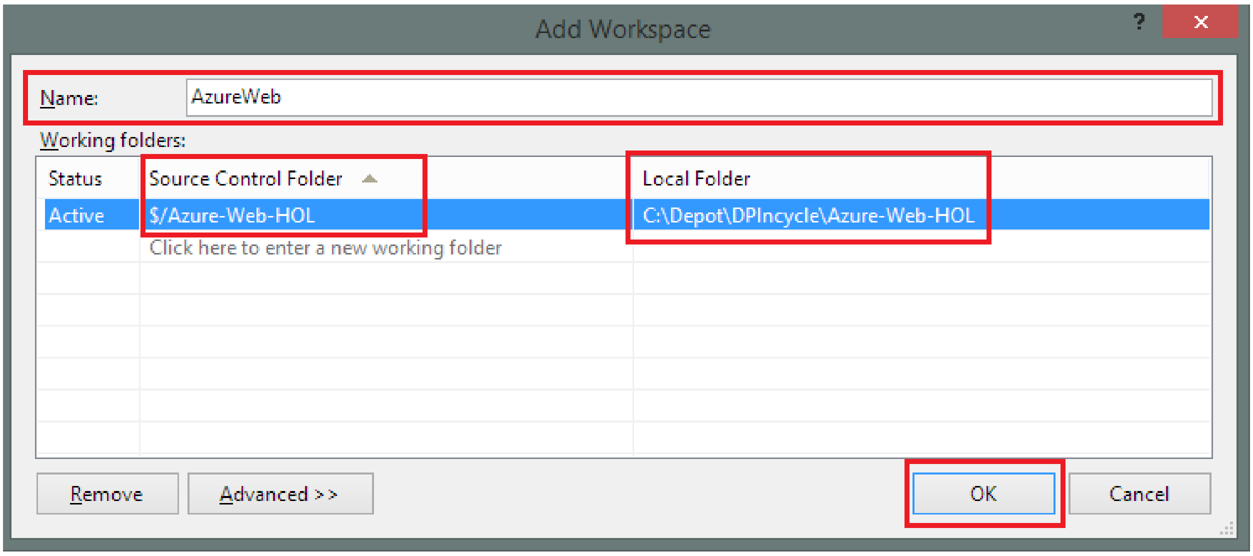


$/InCycleEventWeb

C:\Users\TFS-Deployer\Source\Workspaces\InCycleEventWeb

Figure 13: Advance Workspace creation

3. In the Add Workspace window, enter a workspace name value of AzureWeb. In the Working folders grid keep the existing value for **Source Control Folder**. Change the **Local Folder** to a preferred location on your computer. To continue click **OK.**



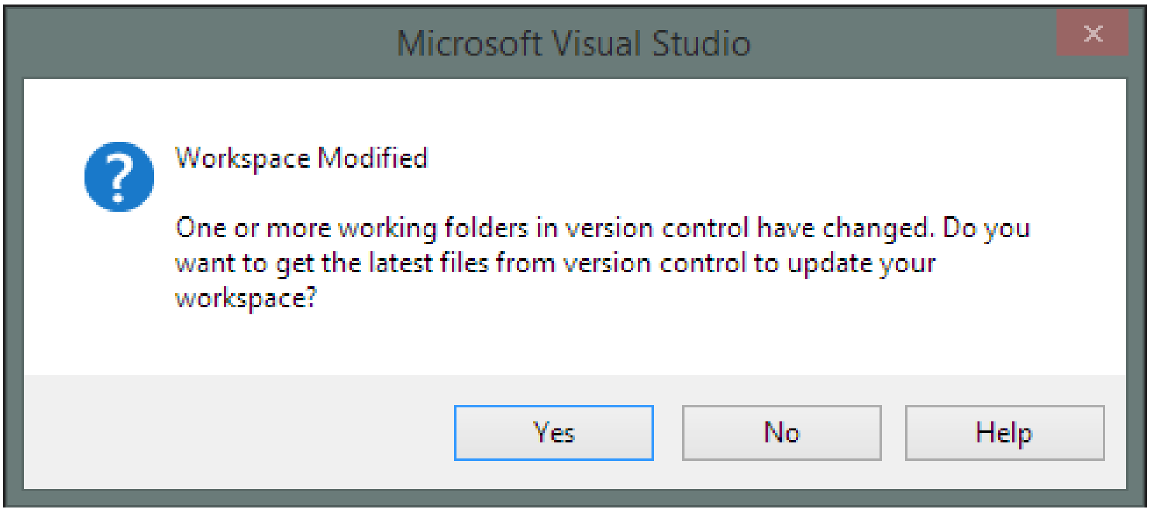
Name: AzureWeb

Working folders:

Status	Source Control Folder	Local Folder
Active	$/Azure-Web-HOL	C:\Depot\DPIncycle\Azure-Web-HOL
	Click here to enter a new working folder	

Figure 14: Workspace settings

4. If the Workspace Modified information dialog appears click **No** to continue.



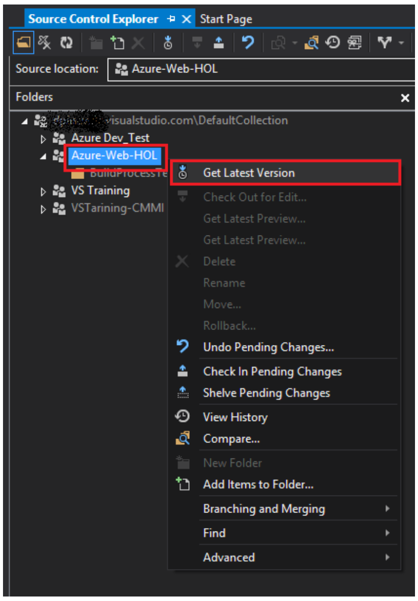
Microsoft Visual Studio

Workspace Modified

One or more working folders in version control have changed. Do you want to get the latest files from version control to update your workspace?

Figure 15: Modified Workspace Information dialog

5. In Visual Studio Open Source Control Explorer by clicking on **View** -> **Other Windows** -> **Source Control Explorer**.
6. In the Source Control Explorer window expand the **DefaultCollection** folder, **right-click** on **Azure-Web-HOL**, and click on **Get Latest Version**.



Source location: Azure-Web-HOL

Figure 16: Get Latest Version of source code

7. Next create a folder to contain your web application project. Right-click on **Azure-Web-HOL** and click on **New Folder** name the folder **Main**. **Right click** on the **Main** folder and **click** on **New Folder** again, this time name the folder **AzureWebSite**.
8. Finally **right-click** on your **AzureWebSite** folder and click on **Check In Pending Changes**. In Team Explorer commit the check in by clicking on the **Check In** button.

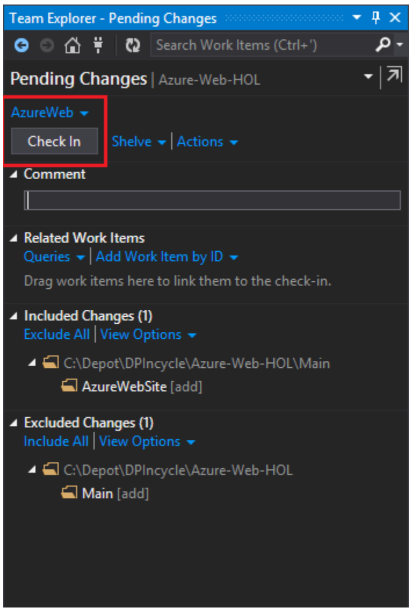

Team Explorer - Pending Changes
Search Work Items (Ctrl+')
Pending Changes | Azure-Web-HOL
AzureWeb
Check In
Shelve
Actions
Comment
Related Work Items
Queries
Add Work Item by ID
Drag work items here to link them to the check-in.
Included Changes (1)
Exclude All
View Options
C:\Depot\DPIncycle\Azure-Web-HOL\Main
AzureWebSite [add]
Excluded Changes (1)
Include All
View Options
C:\Depot\DPIncycle\Azure-Web-HOL
Main [add]

Figure 17: Add and Check In a new folder

Exercise 3: Creating a Web Application

In this exercise, you will create a new MVC Web Application project which will serve as the base project for the remaining exercises to demonstrate continuous deployment to an Azure Web Site whenever code changes are checked into source control.

1. In the **Visual Studio**, click on File -> New -> Project to bring up the new project dialog.

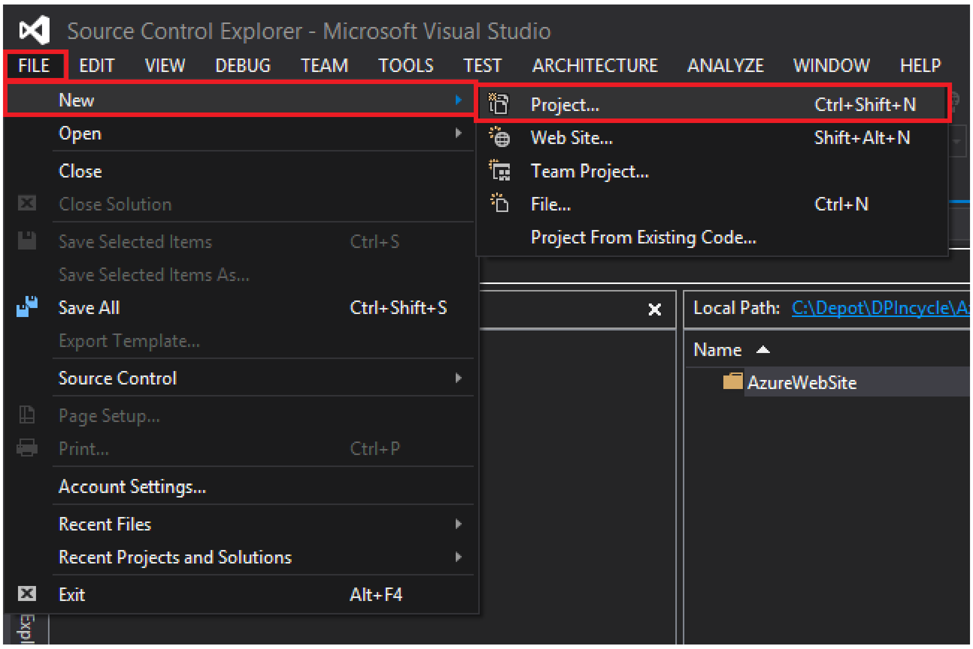
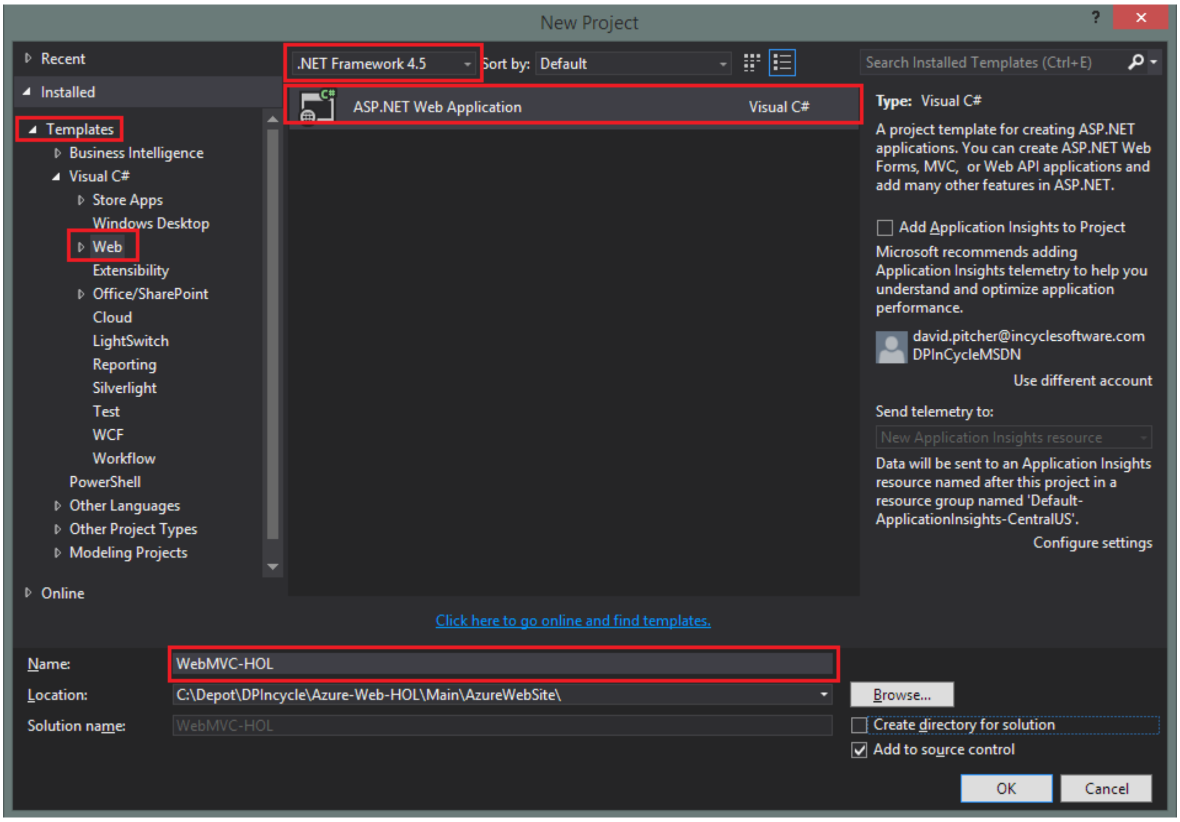


Figure 18: Create new project

2. Expand the **Templates** tree node and click the **Web** tree node. Next make sure **that .NET Framework 4.5** is selected as the default framework version. Next **click** on **ASP .NET Web Application**. For the project name, **enter** a value **of WebMVC-HOL**. **Click** on the **Browse** button to select the location of where your project files are stored.



.NET Framework 4.5

Sort by: Default

Type: Visual C#

A project template for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

□ Add Application Insights to Project

Microsoft recommends adding Application Insights telemetry to help you understand and optimize application performance.

david.pitcher@incyclesoftware.com
DPInCycleMSDN

Send telemetry to: New Application Insights resource

Data will be sent to an Application Insights resource named after this project in a resource group named 'Default-ApplicationInsights-CentralUS'.

Name: WebMVC-HOL

Location: C:\Depot\DPIncycle\Azure-Web-HOL\Main\AzureWebSite\

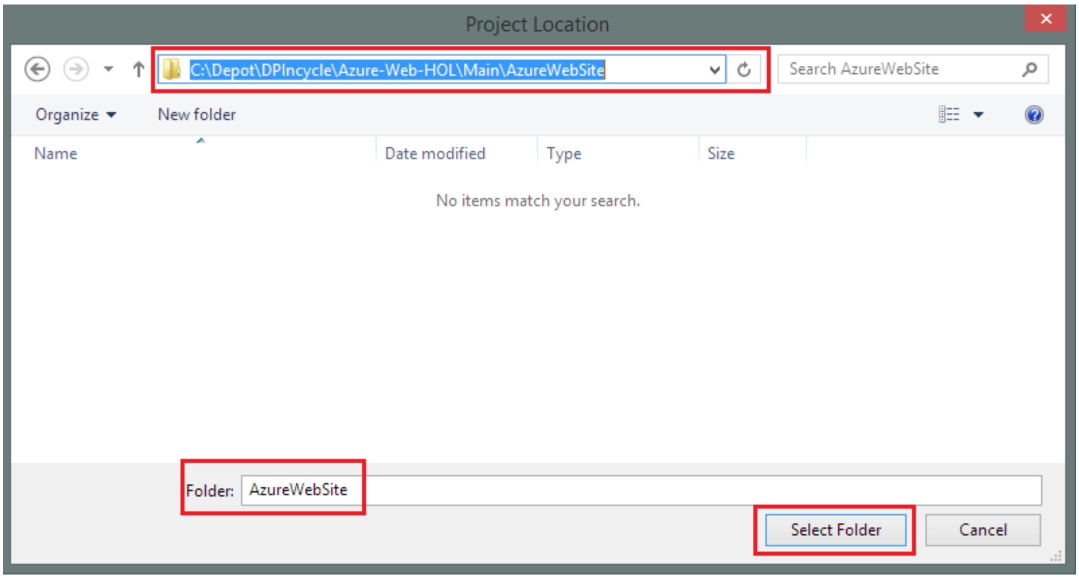
Solution name: WebMVC-HOL

□ Create directory for solution

☑ Add to source control

Figure 19: Create New Project Dialog

3. Navigate to the folder on your local machine which you created in step 7 of exercise 1 and click on the Select Folder button



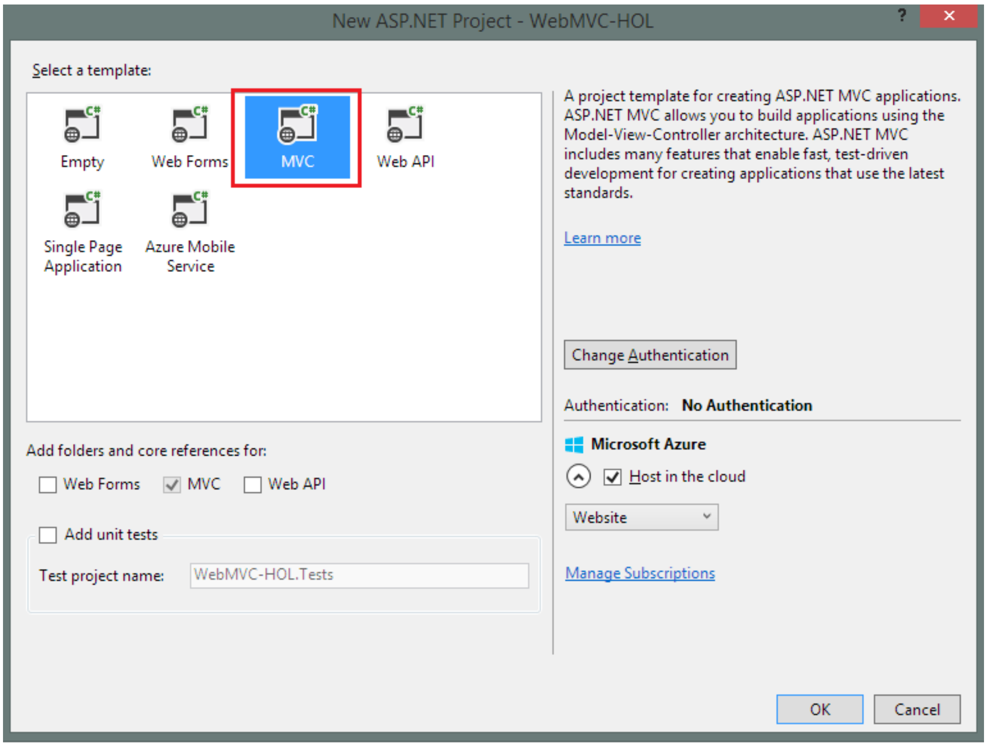
C:\Depot\DPIncycle\Azure-Web-HOL\Main\AzureWebSite

No items match your search.

Folder: AzureWebSite

Figure 20: Select folder for MVC Web project

4. In the New Project window, **un-select Create directory for solution** and **select Add to source control**. **Click** the **OK** button to launch the web project wizard.
5. In the New ASP.NET Project window, select the MVC template.


New ASP.NET Project - WebMVC-HOL

Select a template:

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Learn more

Change Authentication

Authentication: **No Authentication**

Microsoft Azure

☑ Host in the cloud

Website

Manage Subscriptions

Add folders and core references for:

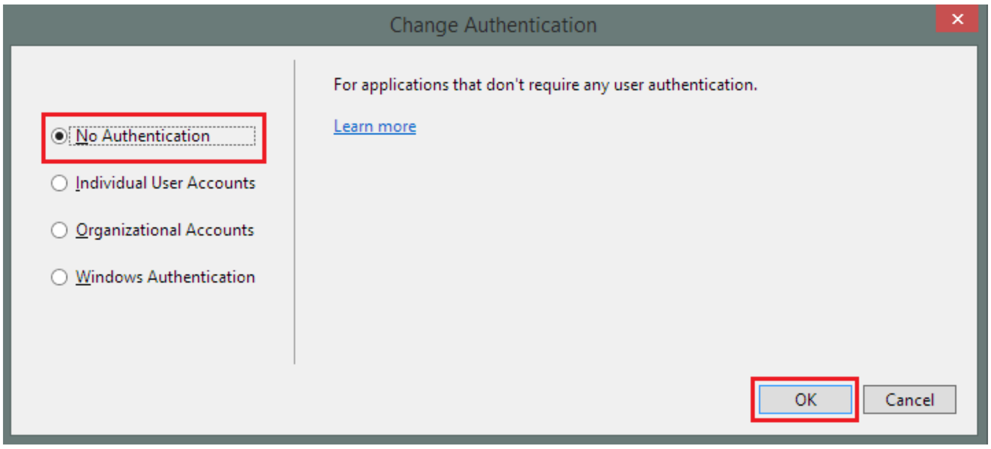
□ Web Forms ☑ MVC □ Web API

□ Add unit tests

Test project name: WebMVC-HOL.Tests

Figure 21: MVC Web Application Template

6. In the New ASP.NET Project window, **click** on the **Change Authentication** button. **Select No Authentication** and **click** the **OK** button.


Change Authentication

● No Authentication

○ Individual User Accounts

○ Organizational Accounts

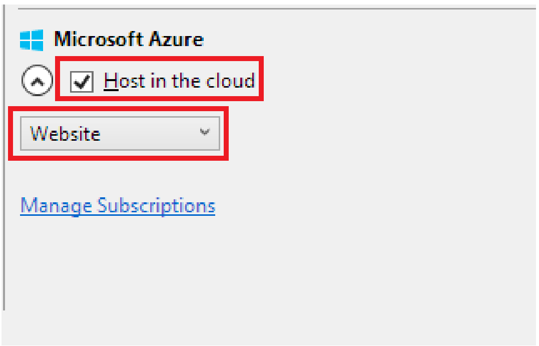
○ Windows Authentication

For applications that don't require any user authentication.

Learn more

Figure 22: Change to No Authentication

7. In the New ASP.NET Project window, click on Host in the Cloud, select Website from the drop-down menu.


Microsoft Azure

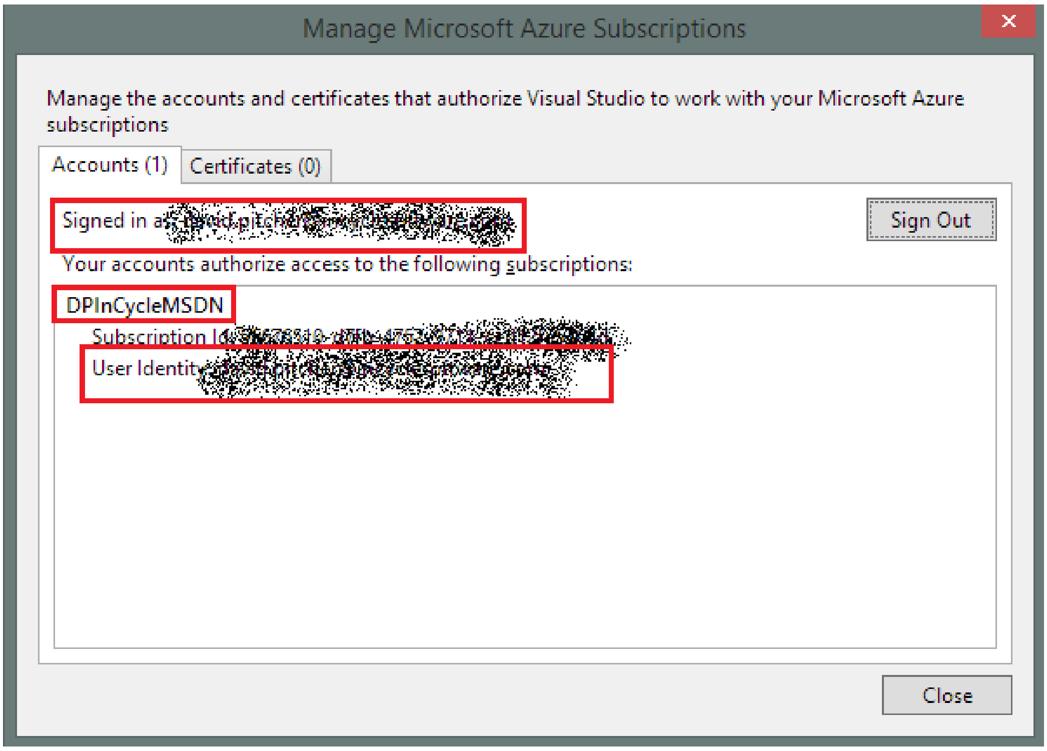
☑ Host in the cloud

Website

Manage Subscriptions

Figure 23: Adding Azure Website as host

8. In the New ASP.NET Project window, **click** on **Manage Subscription** link to configure a connection between Visual Studio and your Azure Subscription.
9. In the Manage Microsoft Azure Subscriptions window, **validate** that the **subscription name** and correct subscription **User Identity** is correct for your Azure subscription. If not click on the Sign In button and enter your Azure Subscription user name and password. **Click** the **Close** button to exit.



Manage Microsoft Azure Subscriptions

Manage the accounts and certificates that authorize Visual Studio to work with your Microsoft Azure subscriptions

Accounts (1) | Certificates (0)

Signed in as [redacted]

Your accounts authorize access to the following subscriptions:

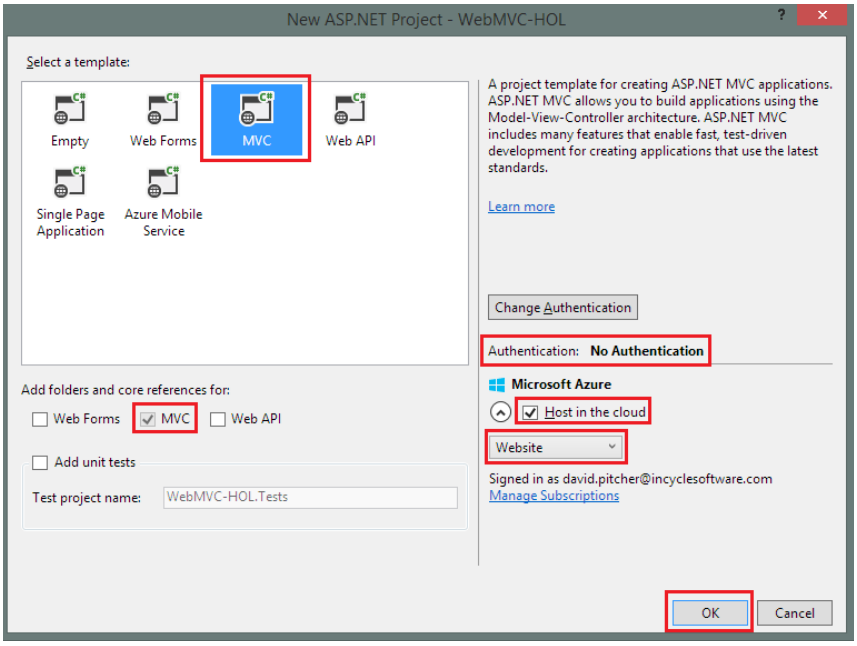
DPInCycleMSDN

Subscription Id: [redacted]

User Identity: [redacted]

Figure 24: Manage Azure Subscription

10. In the New ASP.NET Project window, reconfirm your settings and click the OK button.



New ASP.NET Project - WebMVC-HOL

Select a template:

Empty | Web Forms | MVC | Web API | Single Page Application | Azure Mobile Service

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Learn more

Authentication: **No Authentication**

Add folders and core references for:

□ Web Forms ☑ MVC □ Web API

□ Add unit tests

Test project name: WebMVC-HOL.Tests

Microsoft Azure

☑ Host in the cloud

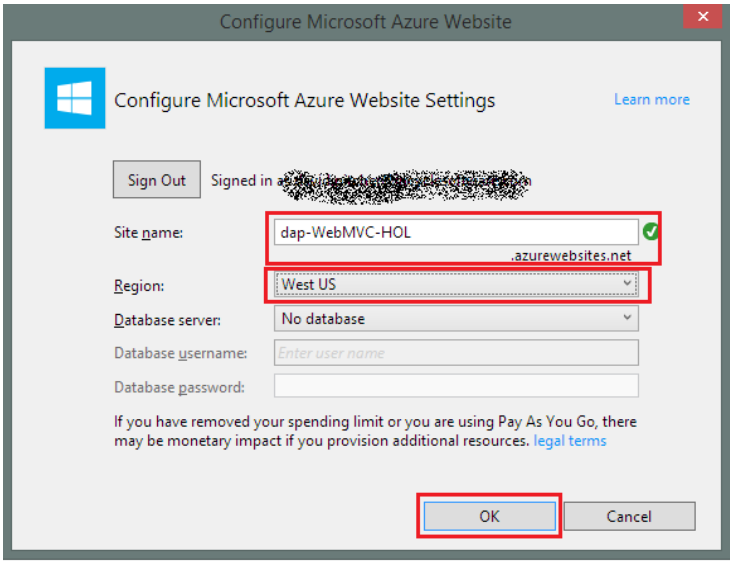
Website

Signed in as david.pitcher@incyclesoftware.com

Manage Subscriptions

Figure 25: Validate new web project settings

11. This next step will create a new Azure Web site that your MVC Web Application will be deployed to. In the Configure Microsoft Azure Website window, **change** the **site name** to be **prefixed** with your initials. The **first letter** of your **first name**, **middle name**, and **last name** followed by a **dash**. For example **dap-WebMVC-HOL**. All Azure Websites must be uniquely named. Next in the Region drop down menu, **select** the **West US** as the region. Finally **click** the **OK** button.



Configure Microsoft Azure Website

Configure Microsoft Azure Website Settings

Learn more

Sign Out Signed in a[redacted]

Site name: dap-WebMVC-HOL .azurewebsites.net

Region: West US

Database server: No database

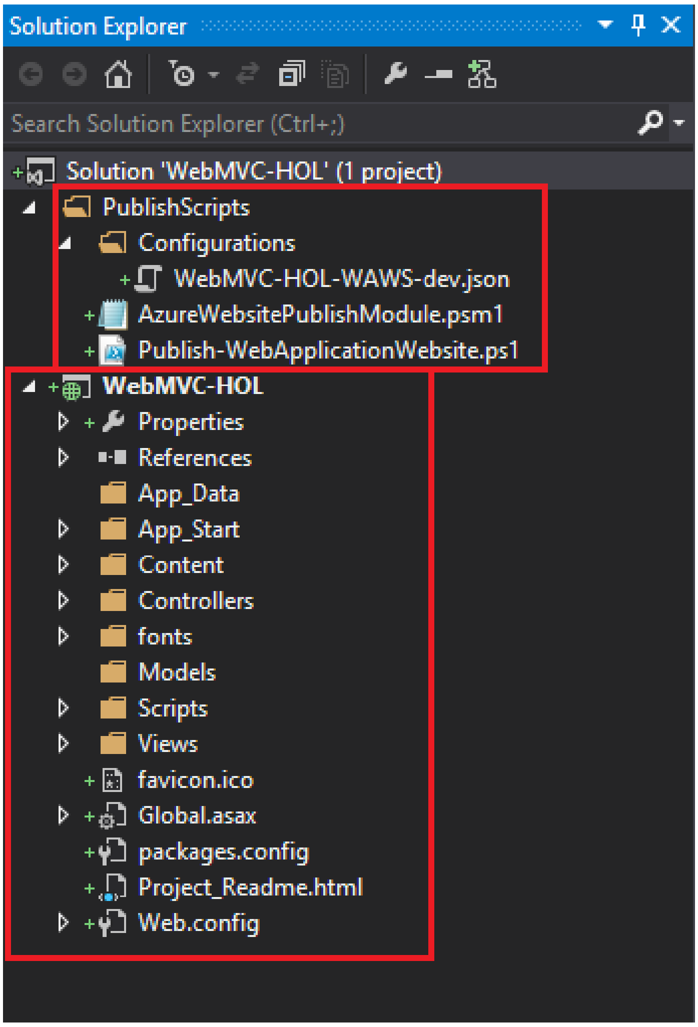
Database username: Enter user name

Database password:

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. legal terms

Figure 26: Add new Azure Website

12. Once your project is completed and the associated Azure Website is created review the MVC web project in the Solution Explorer Window. In particular, notice there is a single **Web project** named **WebMVC-HOL** and a **PublishScripts** folder as part of the created solution.



```
Solution 'WebMVC-HOL' (1 project)
  PublishScripts
    Configurations
      WebMVC-HOL-WAWS-dev.json
    AzureWebsitePublishModule.psm1
    Publish-WebApplicationWebsite.ps1
  WebMVC-HOL
    Properties
    References
    App_Data
    App_Start
    Content
    Controllers
    fonts
    Models
    Scripts
    Views
    favicon.ico
    Global.asax
    packages.config
    Project_Readme.html
    Web.config
```

Figure 27: Completed MVC Web Project

13. While in Visual Studio **enter Ctrl + S** or **click** on **File - > Save Solution** to save the current state of the project and solution files.

14. The last step for this exercise will be to check-in all the new project files into source control. You only need to check-in the files because Visual Studio as part of the project creation process already added them. Check-in the source code by **right clicking** on the Solution **WebMVC-HOL** in the Solution Explorer Window and then **click** on **Check In** menu item.

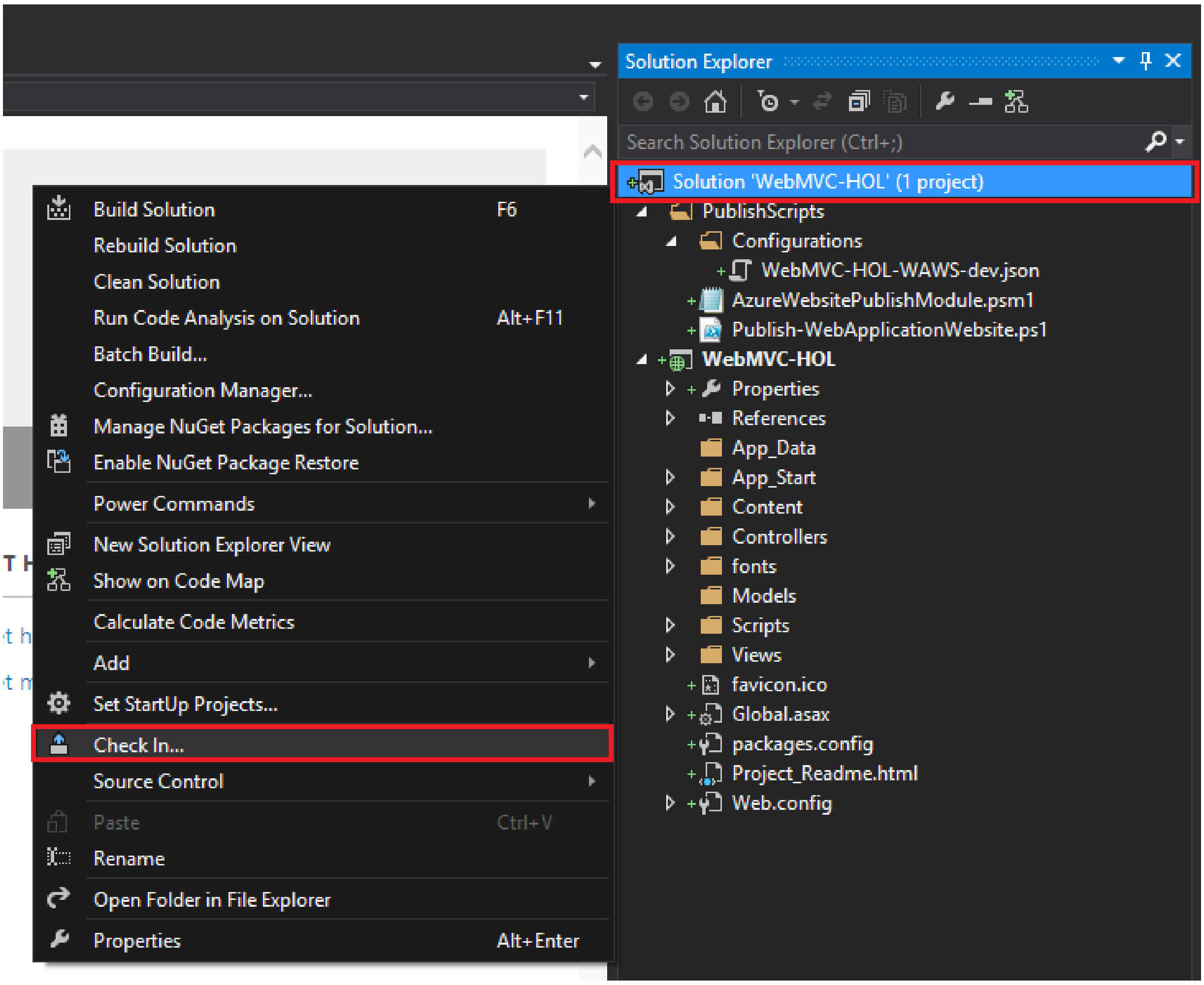


Figure 28: Check in web project

15. In the Team Explorer – Pending Changes window, click on the Check In button to complete the check-in process.

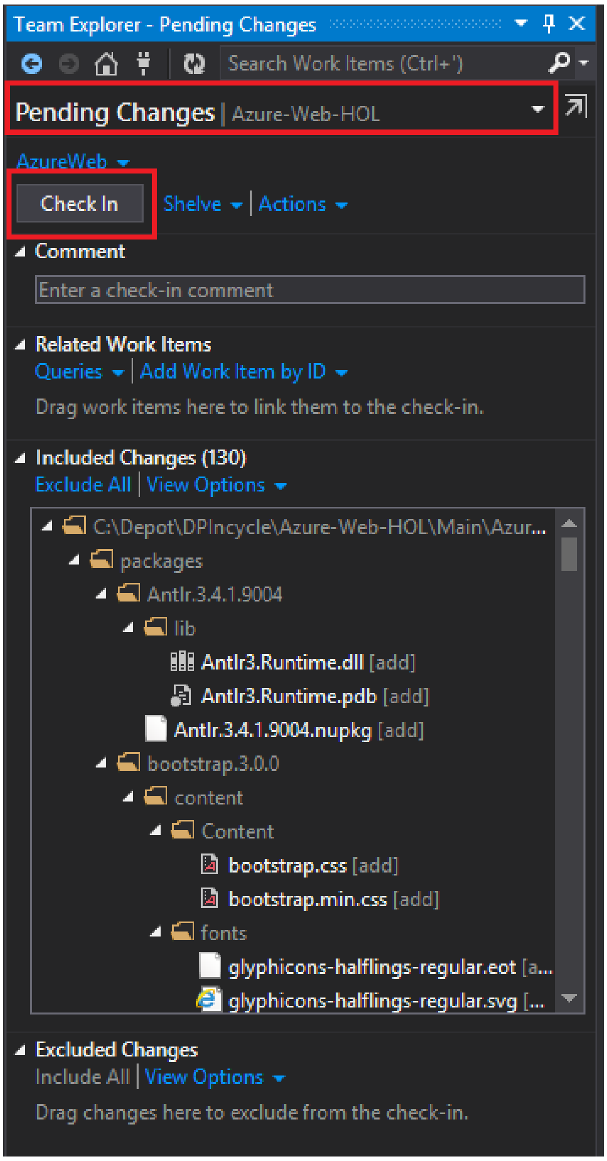


Figure 29: Check In commit changes

Exercise 4: Integrate Source Control with Azure

Now that we have created a new MVC Web application, added it into source control and have an Azure Website ready for deploying our project to, you will configure the Azure Website to integrate with Visual Studio Online Source control to enable continuous integration deployments.

1. In Visual Studio **click** on **View** and then **Server Explorer** to open the Server Explorer window. In the Server Explorer window, expand the Azure Websites tree node and right click on your Azure MVC website that was created in the previous exercise. Select the Open in Management Portal menu item.

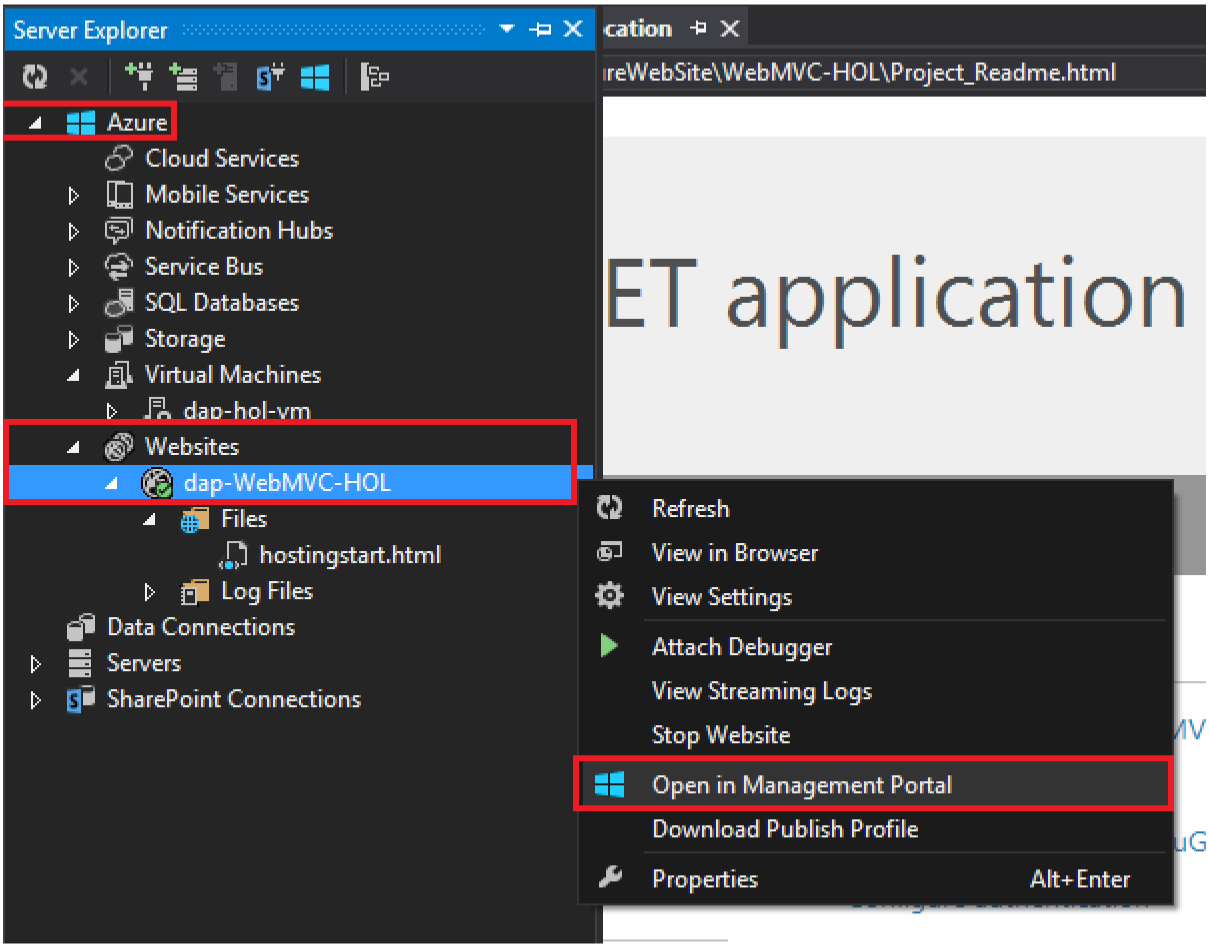
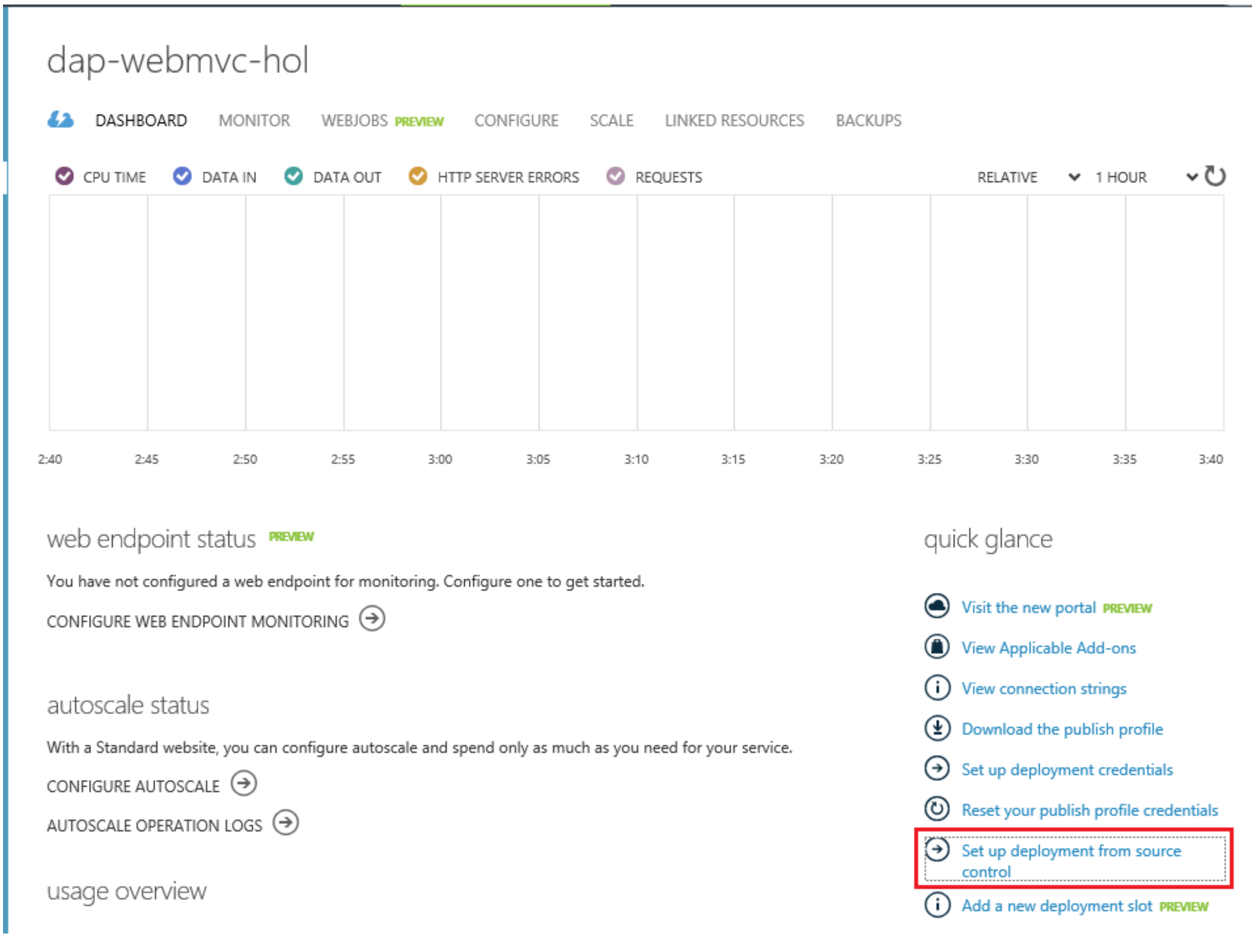


Figure 30: Open Azure Website in Management Portal

2. Your web browser should open with the **Azure Management Portal** loaded and the **Dashboard** page of your **Azure MVC website**. From this page you will begin to configure this website to integrate with VSO source control. In the **quick glance** column **click** on the **Set up deployment from source control** option.



dap-webmvc-hol

web endpoint status PREVIEW

You have not configured a web endpoint for monitoring. Configure one to get started.

CONFIGURE WEB ENDPOINT MONITORING

autoscale status

With a Standard website, you can configure autoscale and spend only as much as you need for your service.

CONFIGURE AUTOSCALE

AUTOSCALE OPERATION LOGS

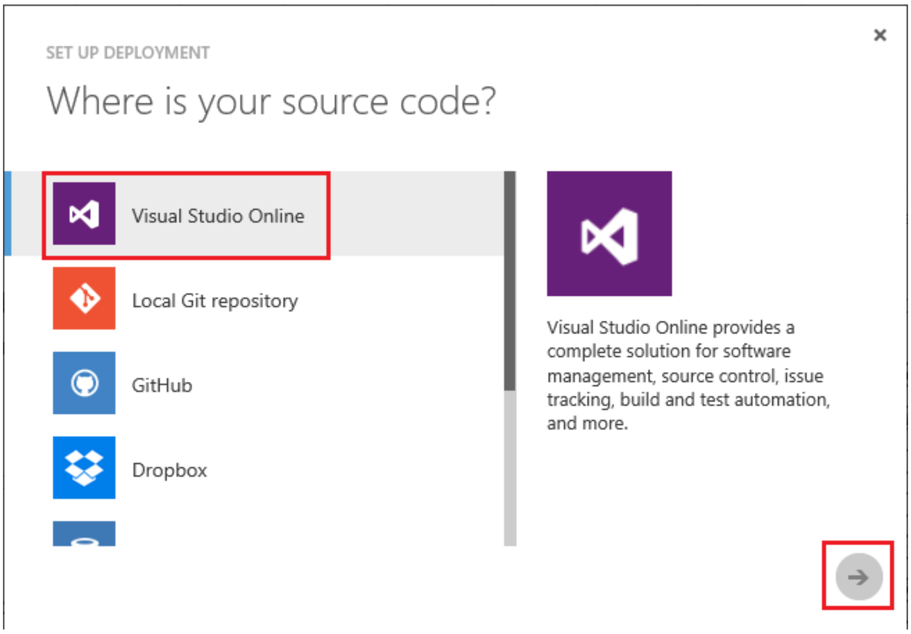
usage overview

quick glance

- Visit the new portal PREVIEW
- View Applicable Add-ons
- View connection strings
- Download the publish profile
- Set up deployment credentials
- Reset your publish profile credentials
- Set up deployment from source control
- Add a new deployment slot PREVIEW

Figure 31: Configure source control integration for website

3. In the Set Up Deployment window, **click** on **Visual Studio Online** option and then **click** on the **next** button.



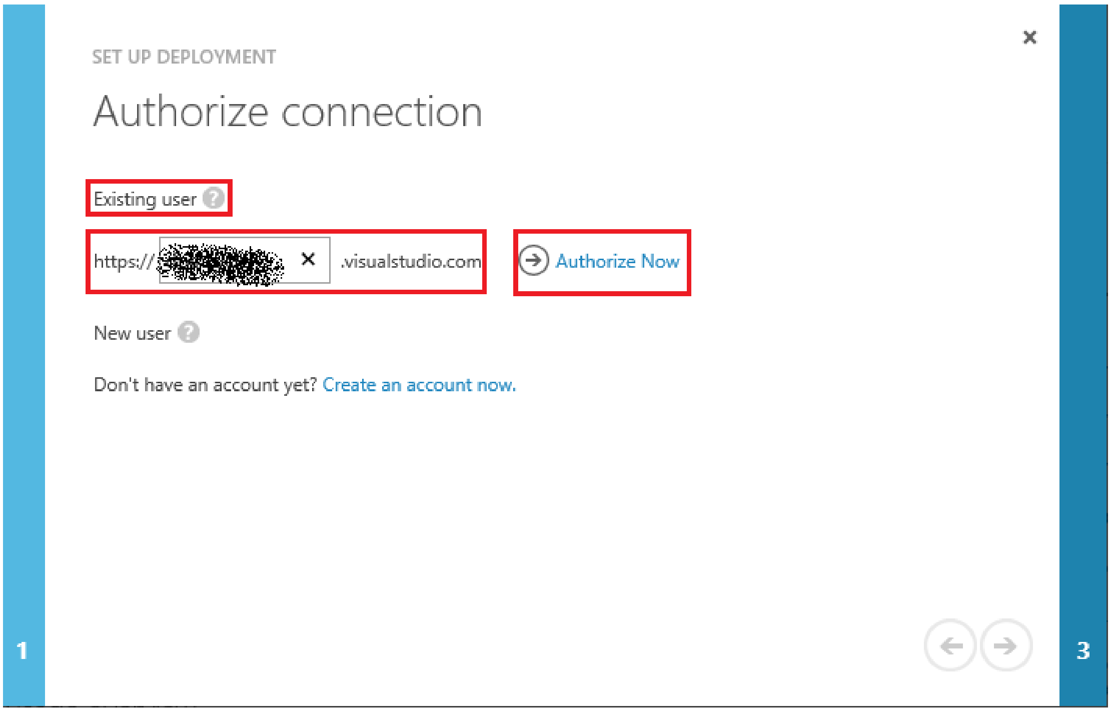
SET UP DEPLOYMENT

Where is your source code?

Visual Studio Online provides a complete solution for software management, source control, issue tracking, build and test automation, and more.

Figure 32: *Select Visual Studio for source code location*

4. In the Set Up Deployment window, **choose Existing user** and **enter** the name of your **Visual Studio Online subscription**. Click on the Authorize Now link to proceed to the next step.



SET UP DEPLOYMENT

Authorize connection

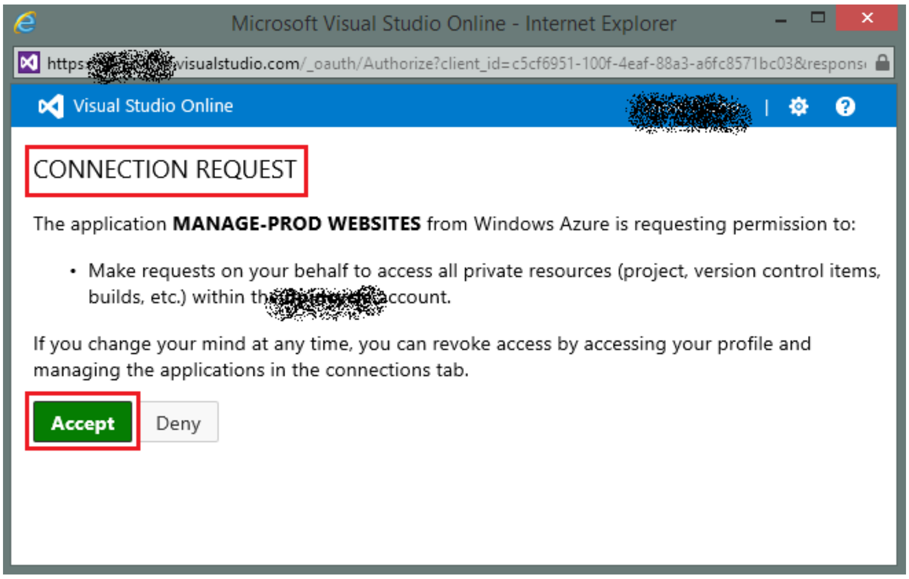
Existing user

https://[redacted] .visualstudio.com

Don't have an account yet? Create an account now.

Figure 33: Connect to Visual Studio Online

5. A new browser window is displayed asking for **approval** of an **Azure Connection Request**. **Click** on the **Accept** button to authorize.



Microsoft Visual Studio Online - Internet Explorer

https[redacted]visualstudio.com/_oauth/Authorize?client_id=c5cf6951-100f-4eaf-88a3-a6fc8571bc03&respons

[redacted]

CONNECTION REQUEST

The application **MANAGE-PROD WEBSITES** from Windows Azure is requesting permission to:

- Make requests on your behalf to access all private resources (project, version control items, builds, etc.) within th[redacted]account.

If you change your mind at any time, you can revoke access by accessing your profile and managing the applications in the connections tab.

Figure 34: Authorize Azure Connection to Visual Studio Online

6. In the Set Up Deployment window, **select** the **Azure-Web-HOL** source control repository. **Click OK** to complete the process.

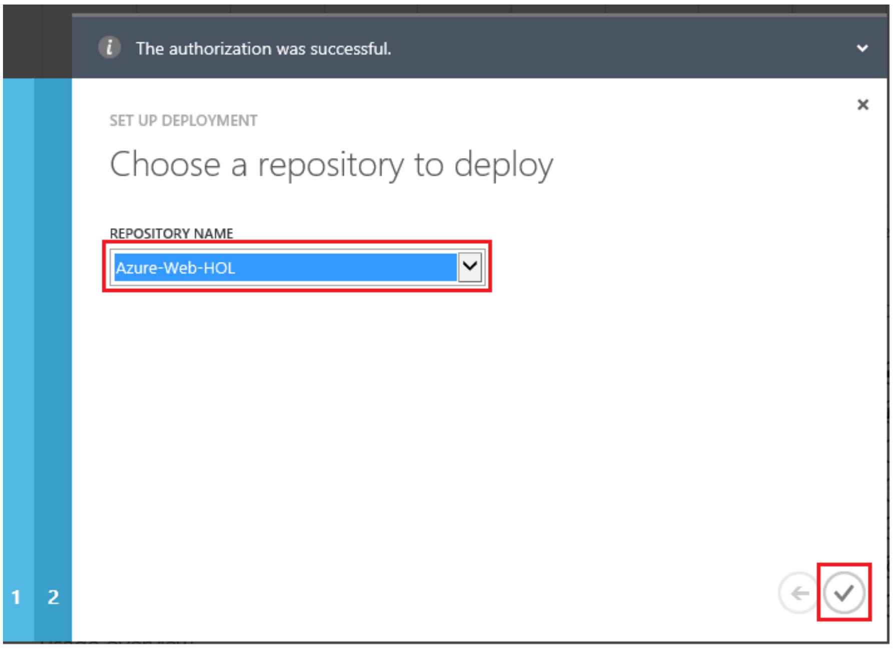
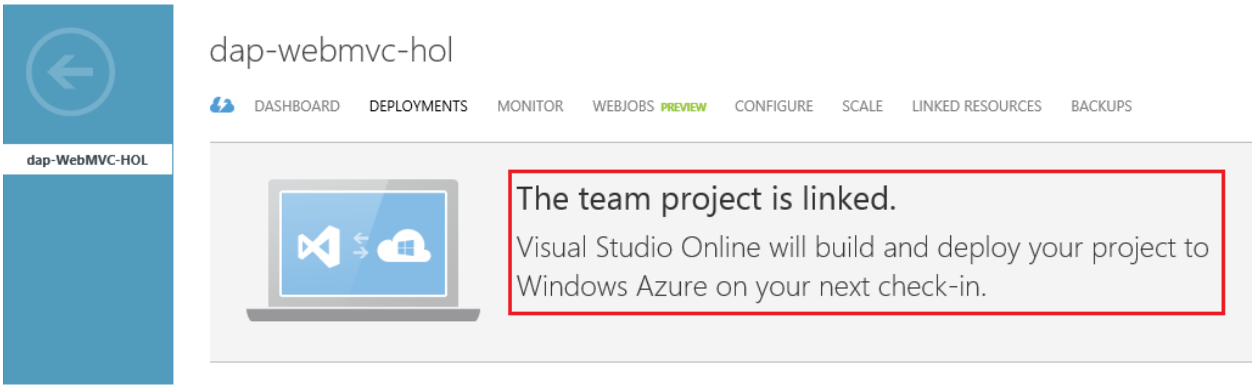

The authorization was successful.
SET UP DEPLOYMENT
Choose a repository to deploy
REPOSITORY NAME
Azure-Web-HOL
1
2

Figure 35: Select Azure-Web-HOL Source Repository

7. The Azure Management Portal will take a few moments to complete the request. Once completed the portal will display a message notifying you the project is now linked.



The team project is linked.

Visual Studio Online will build and deploy your project to Windows Azure on your next check-in.

Figure 36: Project linked to Azure

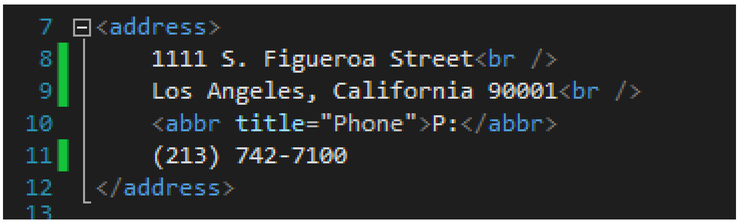
8. Close browser window.

Exercise 5: Making Modifications to the Web Application

In this exercise, you will modify the web application project to customize the contact page to display your contact information.

1. Switch back to Visual Studio and the **Server Explorer** view. Right click on the MVC web site and then click on the **View in Browser** Menu item. A web browser will open and display a generic message alerting you that the web site is created but nothing has been published yet.
2. Close the browser window
3. Switch to the Solution Explorer view and expand the **WebMVC-HOL** web project
4. Underneath the **WebMVC-HOL** project expand the **Views** folder, and then expand the **Home** folder.

5. Double-click on **Contact.cshtml** to open the file in Visual Studio using the default editor.
6. Modify the **address block** to include your contact information. Does not need to be your actual detail any fictitious address will do.



```
<address>
    1111 S. Figueroa Street<br />
    Los Angeles, California 90001<br />
    <abbr title="Phone">P:</abbr>
    (213) 742-7100
</address>
```

Figure 37: Changed contact address

7. Once you have completed the changes, click on File – Save to save the changes to local disk.

Exercise 6: Review Visual Studio Build Definition

In this exercise, you will review the Visual Studio Online Build definition that was automatically created for you by Visual Studio in exercise 4 when you integrated the project's source control with Azure.

1. Switch to the Team Explorer window and **click** on the **Home** icon then **select** the **Builds** link.

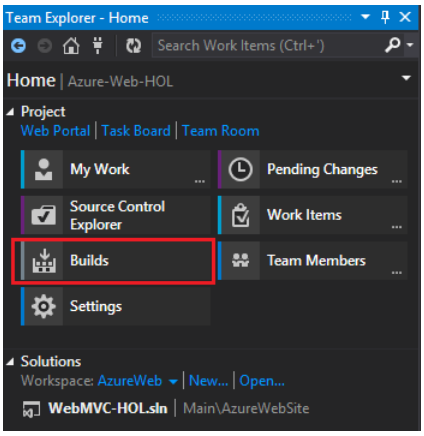


Figure 38: Selecting Team Project builds

2. Under the All Build Definitions Heading you should see a build definition that has the **name** of your Azure MVC **website suffixed** with **_CD**.

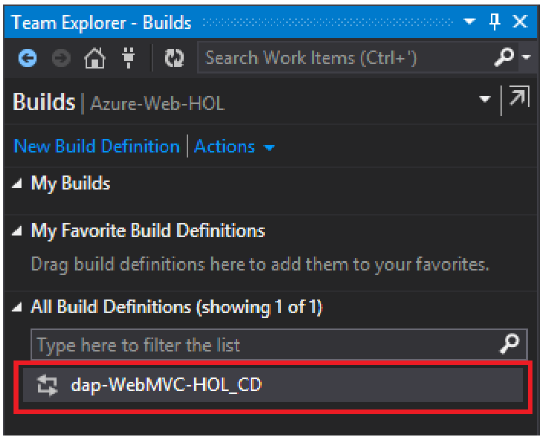

Team Explorer - Builds
Search Work Items (Ctrl+')
Builds | Azure-Web-HOL
New Build Definition | Actions
My Builds
My Favorite Build Definitions
Drag build definitions here to add them to your favorites.
All Build Definitions (showing 1 of 1)
Type here to filter the list
dap-WebMVC-HOL_CD

Figure 39: Identify Azure website build definition

3. **Right click** on the build definition name and then **click** on **Edit Build Definition** menu item.

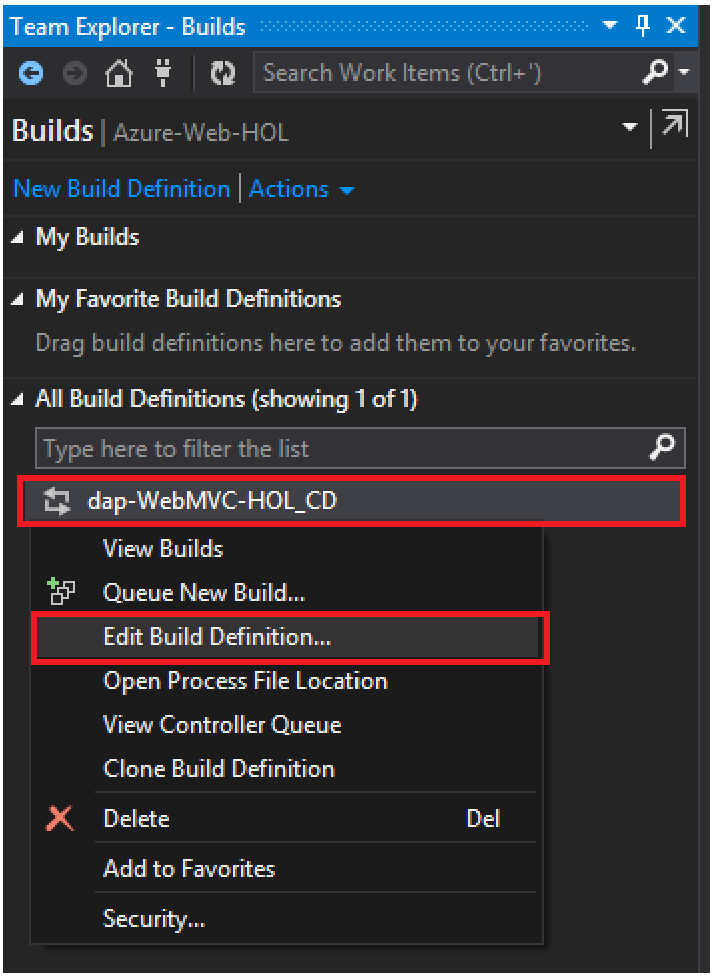
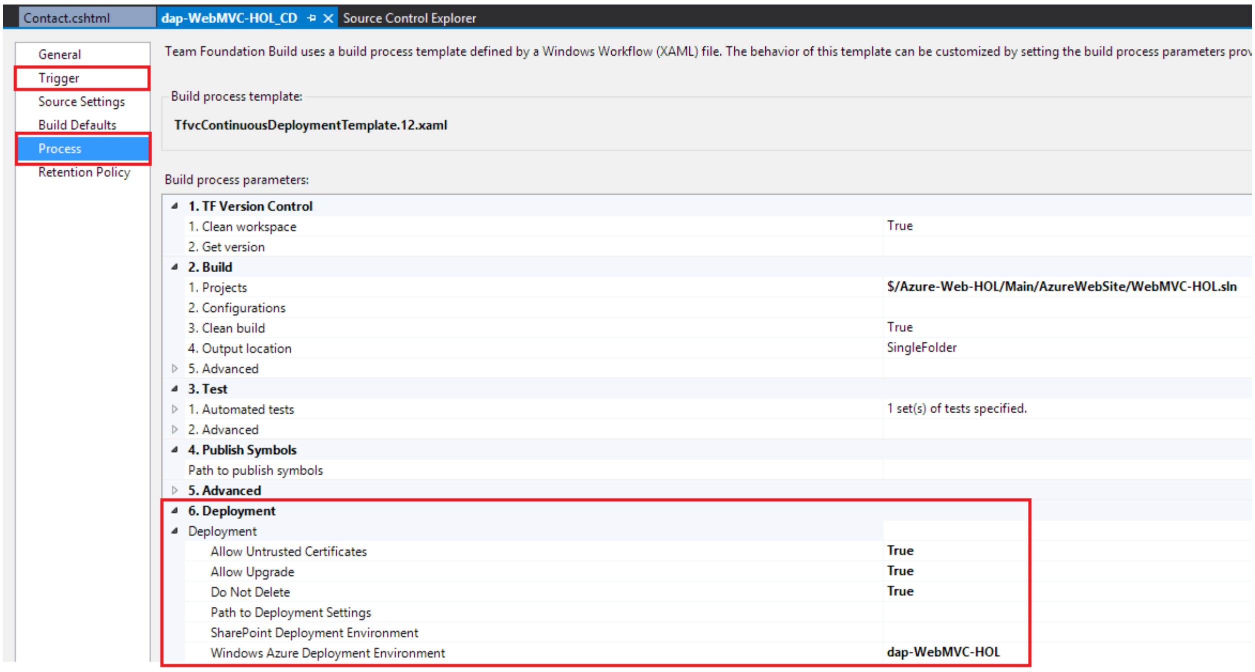

Team Explorer - Builds
Search Work Items (Ctrl+')
Builds | Azure-Web-HOL
New Build Definition | Actions
My Builds
My Favorite Build Definitions
Drag build definitions here to add them to your favorites.
All Build Definitions (showing 1 of 1)
Type here to filter the list
dap-WebMVC-HOL_CD
View Builds
Queue New Build...
Edit Build Definition...
Open Process File Location
View Controller Queue
Clone Build Definition
Delete
Del
Add to Favorites
Security...

Figure 40: Edit build definition

4. In the build definition window, take time to review and familiarize yourself with the build definition settings and configuration. Pay particular attention to the Trigger and Process tabs of the definition.



Team Foundation Build uses a build process template defined by a Windows Workflow (XAML) file. The behavior of this template can be customized by setting the build process parameters prov

Build process template:

TfvcContinuousDeploymentTemplate.12.xaml

Build process parameters:

1. TF Version Control	
1. Clean workspace	True
2. Get version	
2. Build	
1. Projects	**$/Azure-Web-HOL/Main/AzureWebSite/WebMVC-HOL.sln**
2. Configurations	
3. Clean build	True
4. Output location	SingleFolder
5. Advanced	
3. Test	
1. Automated tests	1 set(s) of tests specified.
2. Advanced	
4. Publish Symbols	
Path to publish symbols	
5. Advanced	
6. Deployment	
Deployment	
Allow Untrusted Certificates	**True**
Allow Upgrade	**True**
Do Not Delete	**True**
Path to Deployment Settings	
SharePoint Deployment Environment	
Windows Azure Deployment Environment	**dap-WebMVC-HOL**

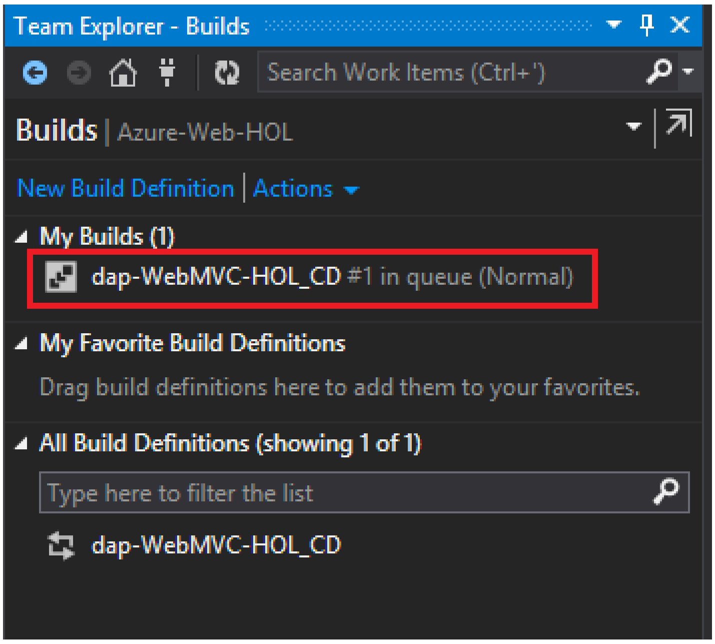
Figure 41: Review build definition

5. When finished reviewing the definition, go ahead and close the definition by clicking File -> Close.

Exercise 7: Building and Deploying the Web Application

In this exercise, you check-in the changes you made to the MVC web application. After completing the check-in you will monitor the automatically triggered build and wait for completion. After the build is completed you will navigate to the Azure website and validate that the changes you made were deployed.

1. Switch back to the Solution Explorer view, **right click** on **Contact.cshtml**, and then **click** on the **Check In** menu item.
2. Complete the check in by **clicking** on the **Check In** button in the Team Explorer Pending Changes window.
3. Click on the Home icon, and Click on the Builds button
4. Double click the first build displayed under My Builds to open up the build execution window to monitor the status of the build.



Team Explorer - Builds

Search Work Items (Ctrl+')

Builds | Azure-Web-HOL

New Build Definition | Actions

My Builds (1)

dap-WebMVC-HOL_CD #1 in queue (Normal)

My Favorite Build Definitions

Drag build definitions here to add them to your favorites.

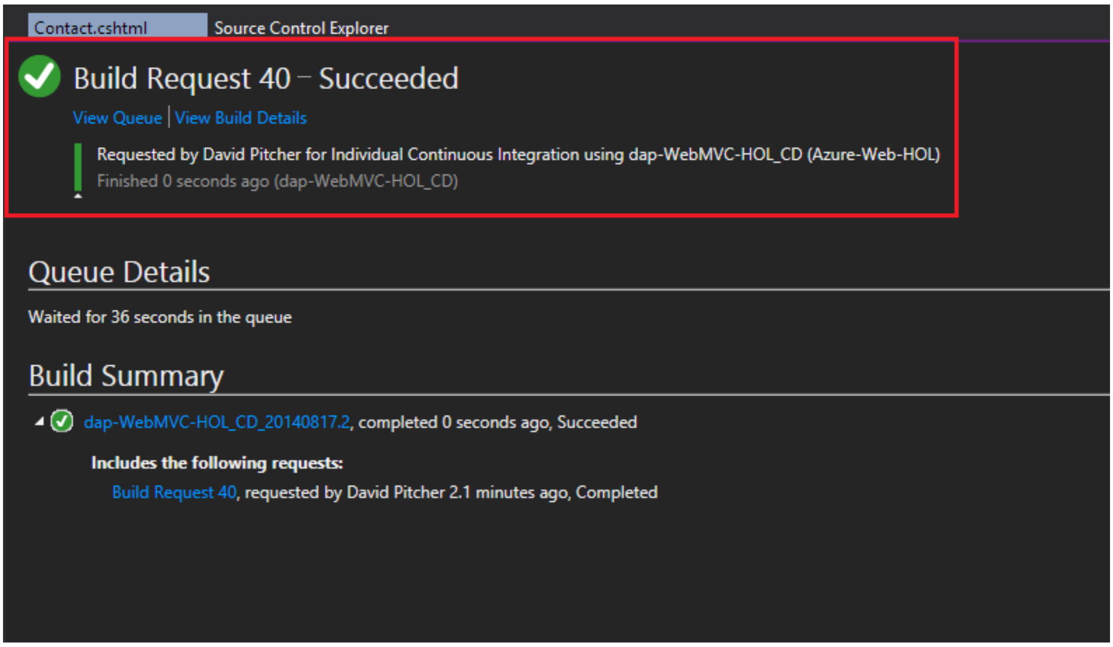
All Build Definitions (showing 1 of 1)

Type here to filter the list

dap-WebMVC-HOL_CD

Figure 42: New build queued from check-in

5. Wait for the build to complete.



Contact.cshtml Source Control Explorer

Build Request 40 – Succeeded

View Queue | View Build Details

Requested by David Pitcher for Individual Continuous Integration using dap-WebMVC-HOL_CD (Azure-Web-HOL)
Finished 0 seconds ago (dap-WebMVC-HOL_CD)

Queue Details

Waited for 36 seconds in the queue

Build Summary

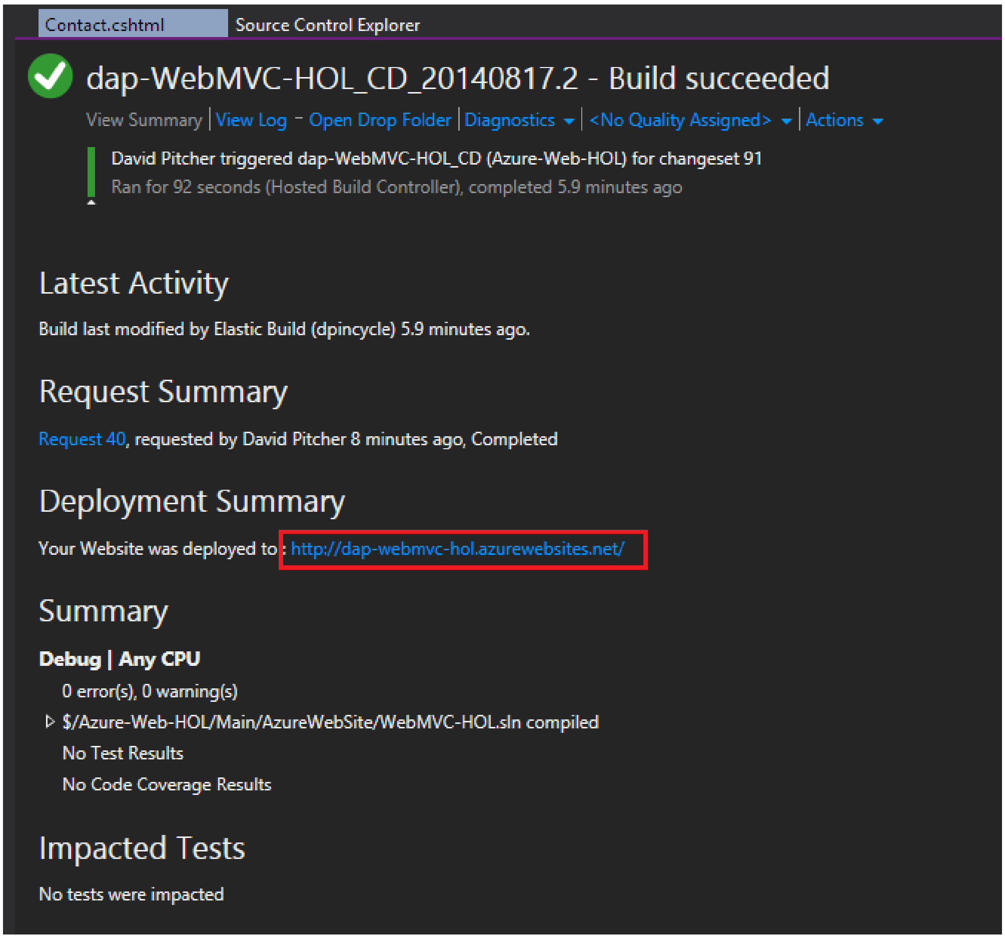
dap-WebMVC-HOL_CD_20140817.2, completed 0 seconds ago, Succeeded

Includes the following requests:

Build Request 40, requested by David Pitcher 2.1 minutes ago, Completed

Figure 43: Completed build

6. Click on the View Build Details link to review details about the build, such as changesets and viewing of the build log.
7. In the Deployment Summary section, click on the URL to your Azure MVC website.



Contact.cshtml Source Control Explorer

dap-WebMVC-HOL_CD_20140817.2 - Build succeeded

View Summary | View Log - Open Drop Folder | Diagnostics | <No Quality Assigned> | Actions

David Pitcher triggered dap-WebMVC-HOL_CD (Azure-Web-HOL) for changeset 91
Ran for 92 seconds (Hosted Build Controller), completed 5.9 minutes ago

Latest Activity

Build last modified by Elastic Build (dpincycle) 5.9 minutes ago.

Request Summary

Request 40, requested by David Pitcher 8 minutes ago, Completed

Deployment Summary

Your Website was deployed to : http://dap-webmvc-hol.azurewebsites.net/

Summary

Debug | Any CPU

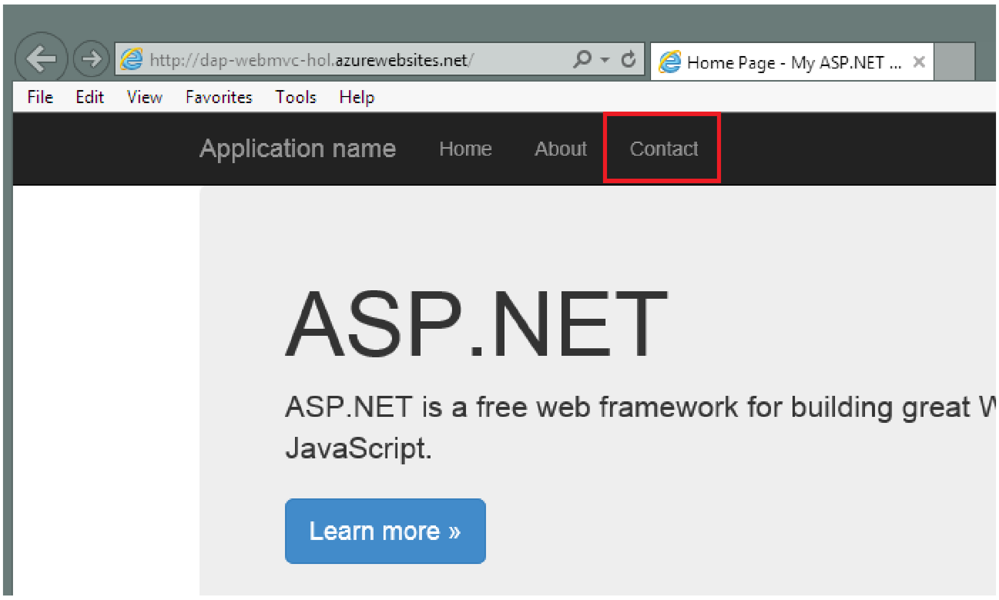
0 error(s), 0 warning(s)
$/Azure-Web-HOL/Main/AzureWebSite/WebMVC-HOL.sln compiled
No Test Results
No Code Coverage Results

Impacted Tests

No tests were impacted

Figure 44: Completed build details

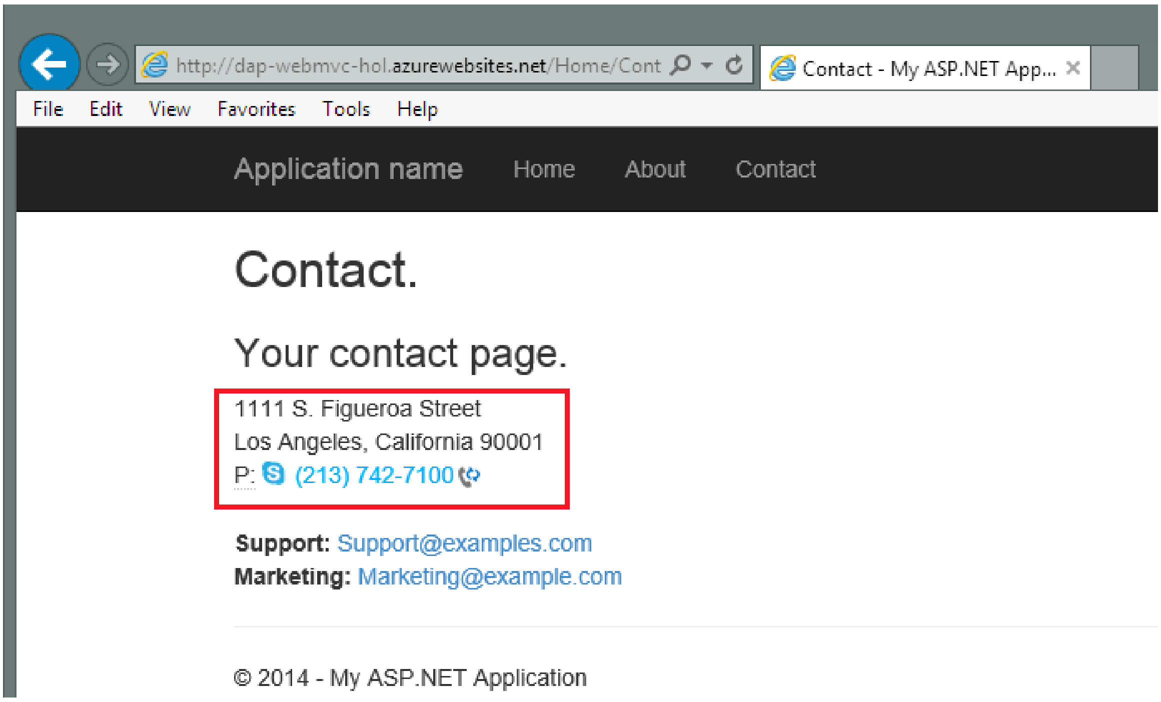
8. Once the website opens click on the Contact menu link and wait for the contacts page to load.



http://dap-webmvc-hol.azurewebsites.net/
Home Page - My ASP.NET ...
File Edit View Favorites Tools Help
Application name
Home
About
Contact
ASP.NET
ASP.NET is a free web framework for building great W
JavaScript.
Learn more »

***Figure 45**: Deployed MVC web site*

9. Validate the contact address information displayed is equivalent to the changes you made to the web application in exercise 5.



http://dap-webmvc-hol.azurewebsites.net/Home/Cont
Contact - My ASP.NET App...
File
Edit
View
Favorites
Tools
Help
Application name
Home
About
Contact

Contact.

Your contact page.

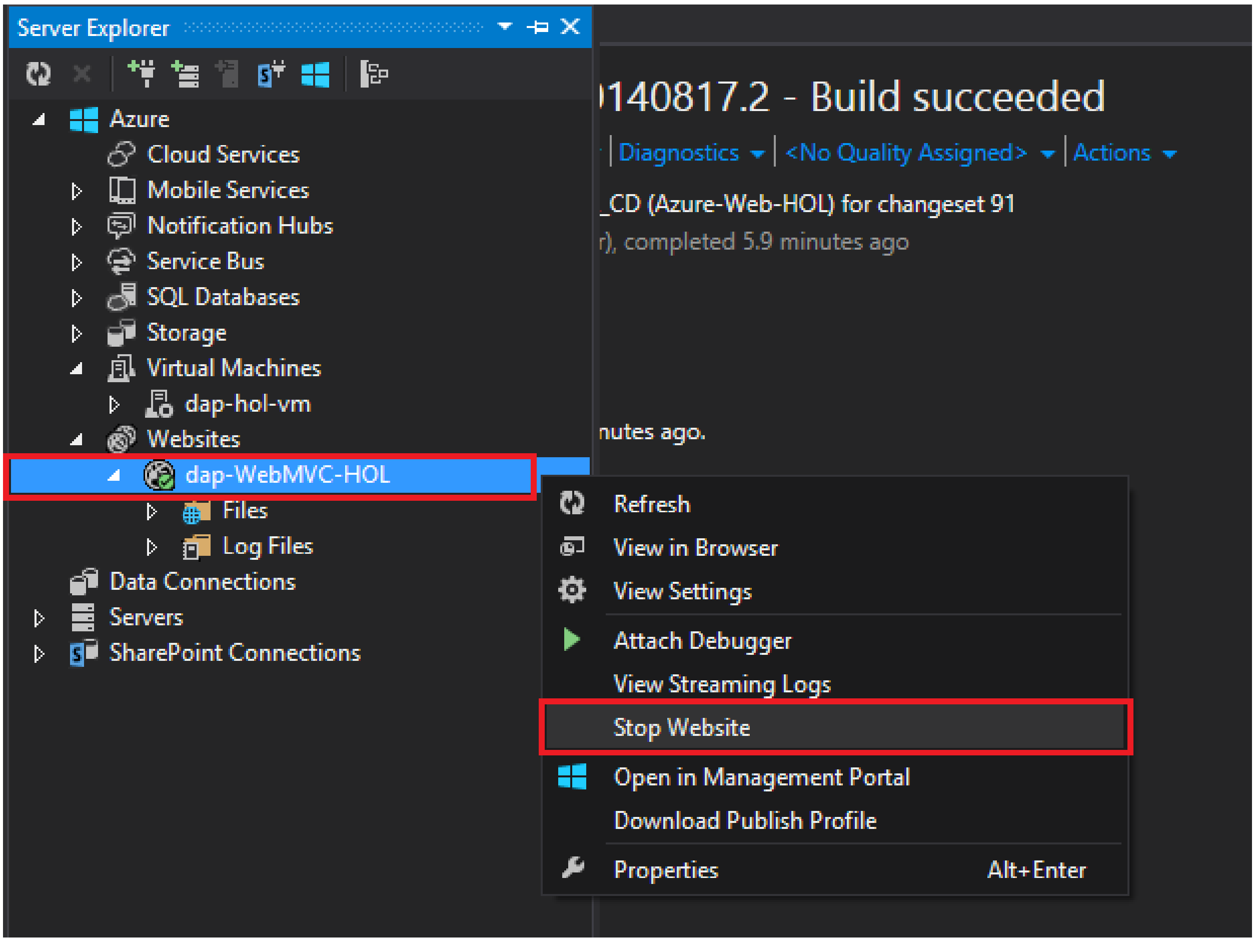
1111 S. Figueroa Street
Los Angeles, California 90001
P: (213) 742-7100

Support: Support@examples.com
Marketing: Marketing@example.com

© 2014 - My ASP.NET Application

Figure 46: Validate contact information changes

10. Close the Browser.
11. Switch back to Visual Studio and the Server Explorer view, right click on your Azure Website and click on the Stop Website button to stop and disable the website.



)140817.2 - Build succeeded

Diagnostics | <No Quality Assigned> | Actions

_CD (Azure-Web-HOL) for changeset 91

r), completed 5.9 minutes ago

nutes ago.

Figure 47: Stopping Azure website