Connect to on-premises SQL Server from a web app in Azure App Service using Hybrid Connections

2/9/2016 • 12 min to read • Contributors

In this article

Hybrid Connections can connect Azure App Service Web Apps to on-premises resources that use a static TCP port. Supported resources include Microsoft SQL Server, MySQL, HTTP Web APIs, App Service, and most custom Web Services.

In this tutorial, you will learn how to create an App Service web app in the Azure Portal, connect the web app to your local on-premises SQL Server database using the new Hybrid Connection feature, create a simple ASP.NET application that will use the hybrid connection, and deploy the application to the App Service web app. The completed web app on Azure stores user credentials in a membership database that is on-premises. The tutorial assumes no prior experience using Azure or ASP.NET.

> **Note**
>
> If you want to get started with Azure App Service before signing up for an Azure account, go to Try App Service, where you can immediately create a short-lived starter web app in App Service. No credit cards required; no commitments.
>
> The Web Apps portion of the Hybrid Connections feature is available only in the Azure Portal. To create a connection in BizTalk Services, see Hybrid Connections.

Prerequisites

To complete this tutorial, you'll need the following products. All are available in free versions, so you can start developing for Azure entirely for free.

- **Azure subscription** - For a free subscription, see Azure Free Trial.

- **Visual Studio 2013** - To download a free trial version of Visual Studio 2013, see Visual Studio Downloads. Install this before continuing.
- **Microsoft .NET Framework 3.5 Service Pack 1** - If your operating system is Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, or Windows Server 2008 R2, you can enable this in Control Panel > Programs and Features > Turn Windows features on or off. Otherwise, you can download it from the Microsoft Download Center.
- **SQL Server 2014 Express with Tools** - download Microsoft SQL Server Express for free at the Microsoft Web Platform Database page. Choose the **Express** (not LocalDB) version. The **Express with Tools** version includes SQL Server Management Studio, which you will use in this tutorial.
- **SQL Server Management Studio Express** - This is included with the SQL Server 2014 Express with Tools download mentioned above, but if you need to install it separately, you can download and install it from the SQL Server Express download page.

The tutorial assumes that you have an Azure subscription, that you have installed Visual Studio 2013, and that you have installed or enabled .NET Framework 3.5. The tutorial shows you how to install SQL Server 2014 Express in a configuration that works well with the Azure Hybrid Connections feature (a default instance with a static TCP port). Before starting the tutorial, download SQL Server 2014 Express with Tools from the location mentioned above if you do not have SQL Server installed.

Notes

To use an on-premises SQL Server or SQL Server Express database with a hybrid connection, TCP/IP needs to be enabled on a static port. Default instances on SQL Server use static port 1433, whereas named instances do not.

The computer on which you install the on-premises Hybrid Connection Manager agent:

- Must have outbound connectivity to Azure over:

Port	Why
80	**Required** for HTTP port for certificate validation and optionally for data connectivity.
443	**Optional** for data connectivity. If outbound connectivity to 443 is unavailable, TCP port 80 is used.
5671 and 9352	**Recommended** but Optional for data connectivity. Note this mode usually yields higher throughput. If outbound connectivity to these ports is unavailable, TCP port 443 is used.

- Must be able to reach the *hostname:portnumber* of your on-premises resource.

The steps in this article assume that you are using the browser from the computer that will host the on-premises hybrid connection agent.

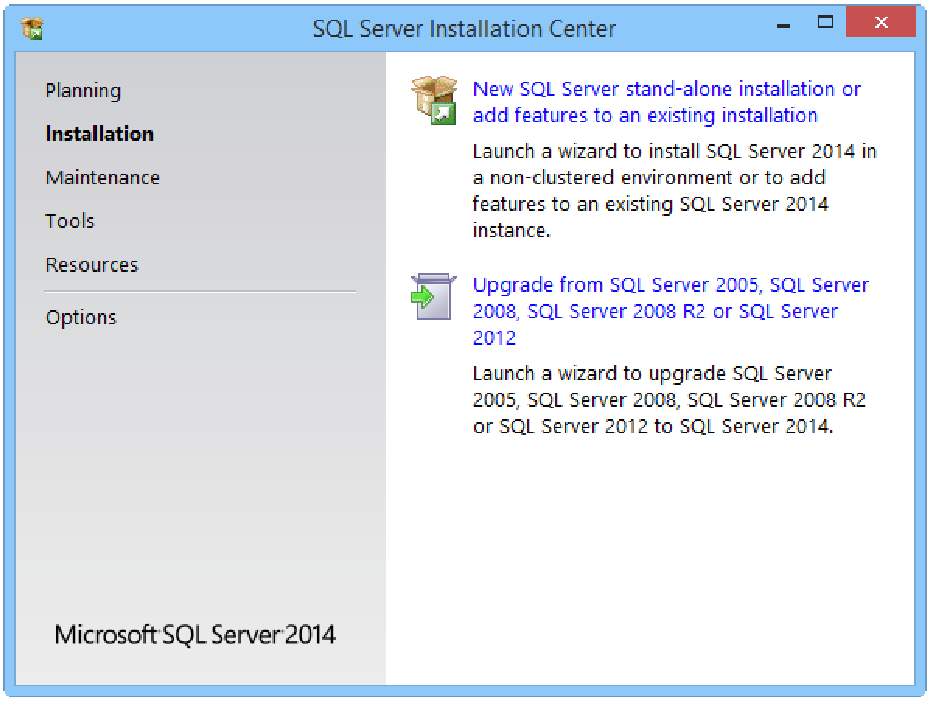
If you already have SQL Server installed in a configuration and in an environment that meets the conditions described above, you can skip ahead and start with Create a SQL Server database on-premises.

A. Install SQL Server Express, enable TCP/IP, and create a SQL Server database on-premises

This section shows you how to install SQL Server Express, enable TCP/IP, and create a database so that your web application will work with the Azure Portal.

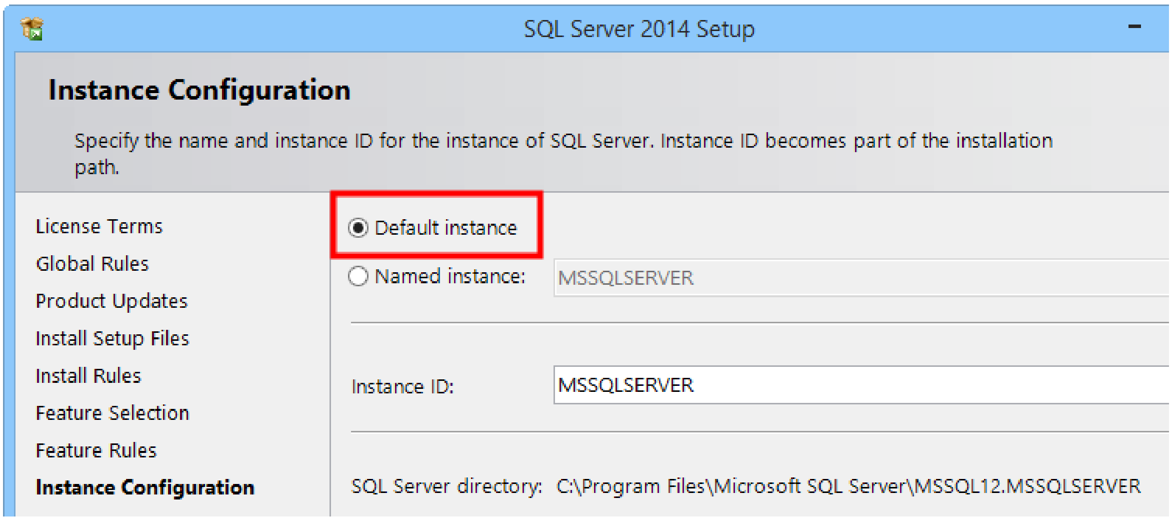
Install SQL Server Express

1. To install SQL Server Express, run the **SQLEXPRWT_x64_ENU.exe** or **SQLEXPR_x86_ENU.exe** file that you downloaded. The SQL Server Installation Center wizard appears.


SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Options
New SQL Server stand-alone installation or add features to an existing installation
Launch a wizard to install SQL Server 2014 in a non-clustered environment or to add features to an existing SQL Server 2014 instance.
Upgrade from SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 or SQL Server 2012
Launch a wizard to upgrade SQL Server 2005, SQL Server 2008, SQL Server 2008 R2 or SQL Server 2012 to SQL Server 2014.
Microsoft SQL Server 2014

2. Choose **New SQL Server stand-alone installation or add features to an existing installation**. Follow the instructions, accepting the default choices and settings, until you get to the **Instance Configuration** page.

3. On the **Instance Configuration** page, choose **Default instance**.


SQL Server 2014 Setup

Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

● Default instance

○ Named instance: MSSQLSERVER

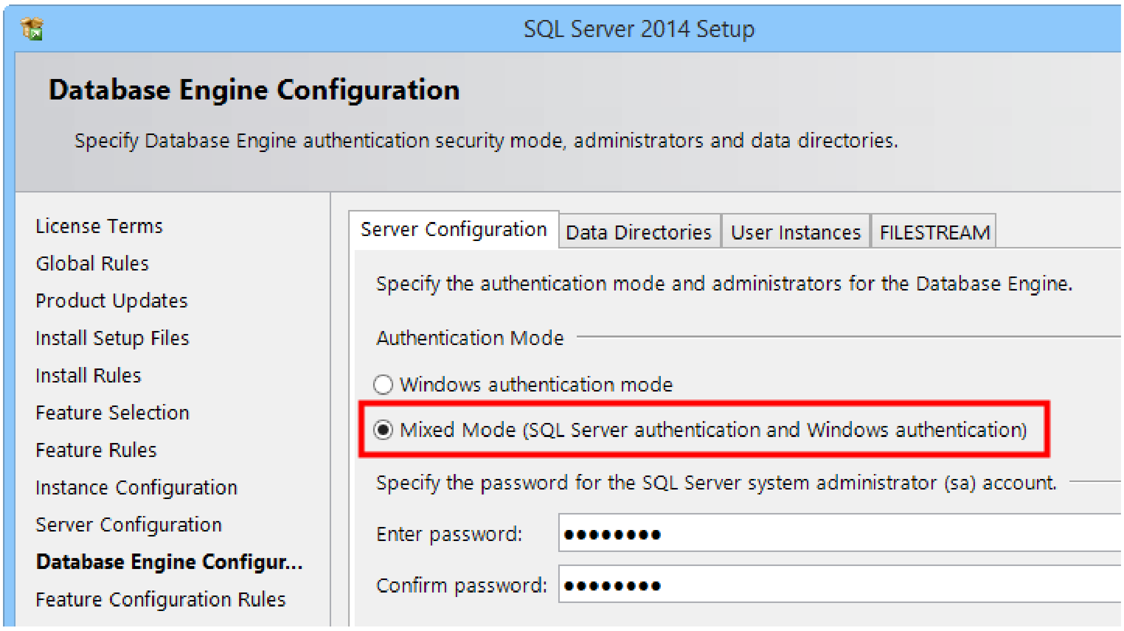
Instance ID: MSSQLSERVER

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER

By default, the default instance of SQL Server listens for requests from SQL Server clients on static port 1433, which is what the Hybrid Connections feature requires. Named instances use dynamic ports and UDP, which are not supported by Hybrid Connections.

4. Accept the defaults on the **Server Configuration** page.

5. On the **Database Engine Configuration** page, under **Authentication Mode**, choose **Mixed Mode (SQL Server authentication and Windows authentication)**, and provide a password.



Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

○ Windows authentication mode

● Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password: ••••••••

Confirm password: ••••••••

In this tutorial, you will be using SQL Server authentication. Be sure to remember the password that you provide, because you will need it later.

6. Step through the rest of the wizard to complete the installation.

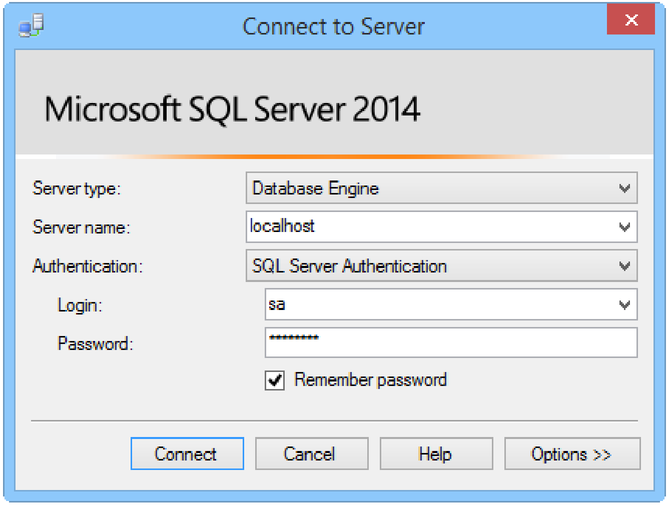
Enable TCP/IP

To enable TCP/IP, you will use SQL Server Configuration Manager, which was installed when you installed SQL Server Express. Follow the steps in Enable TCP/IP Network Protocol for SQL Server before continuing.

Create a SQL Server database on-premises

Your Visual Studio web application requires a membership database that can be accessed by Azure. This requires a SQL Server or SQL Server Express database (not the LocalDB database that the MVC template uses by default), so you'll create the membership database next.

1. In SQL Server Management Studio, connect to the SQL Server you just installed. (If the **Connect to Server** dialog does not appear automatically, navigate to **Object Explorer** in the left pane, click **Connect**, and then click **Database Engine**.)



Microsoft SQL Server 2014

Server type: Database Engine

Server name: localhost

Authentication: SQL Server Authentication

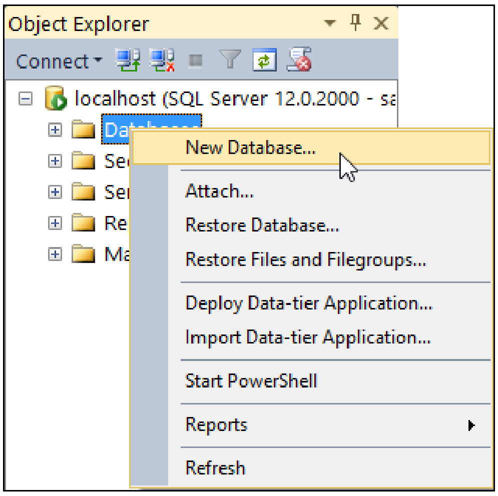
Login: sa

Password: ********

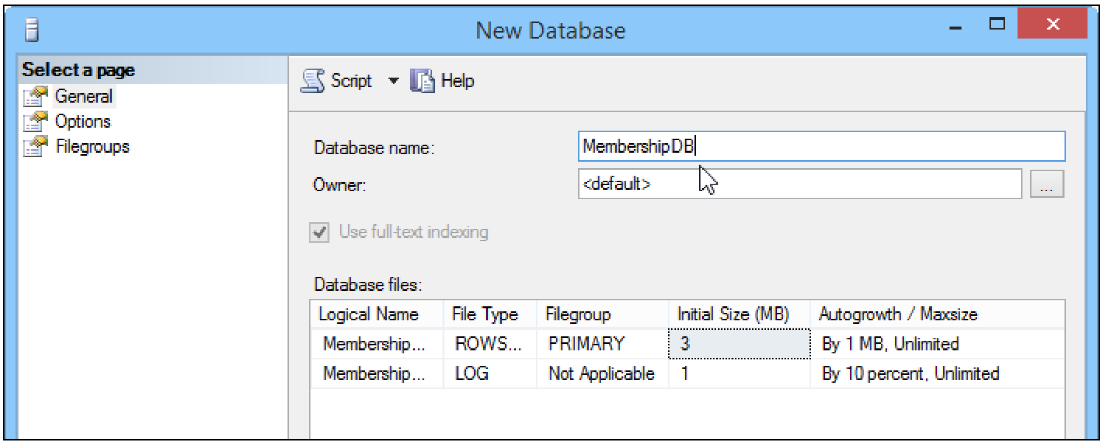
☑ Remember password

For **Server type**, choose **Database Engine**. For **Server name**, you can use **localhost** or the name of the computer that you are using. Choose **SQL Server authentication**, and then log in with the sa user name and the password that you created earlier.

2. To create a new database by using SQL Server Management Studio, right-click **Databases** in Object Explorer, and then click **New Database**.


Object Explorer
Connect
localhost (SQL Server 12.0.2000 - sa
Databases
Se
Se
Re
Ma
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh

3. In the **New Database** dialog, enter MembershipDB for the database name, and then click **OK**.



Database name: MembershipDB

Owner: <default>

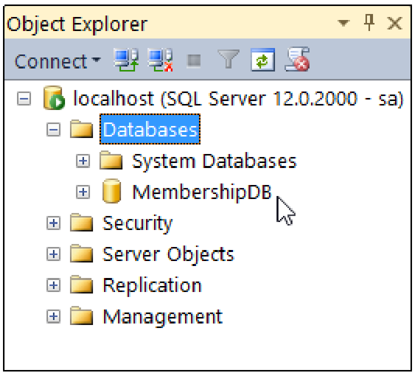
☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
Membership...	ROWS...	PRIMARY	3	By 1 MB, Unlimited
Membership...	LOG	Not Applicable	1	By 10 percent, Unlimited

Note that you do not make any changes to the database at this point. The membership information will be added automatically later by your web application when you run it.

4. In Object Explorer, if you expand **Databases**, you will see that the membership database has been created.

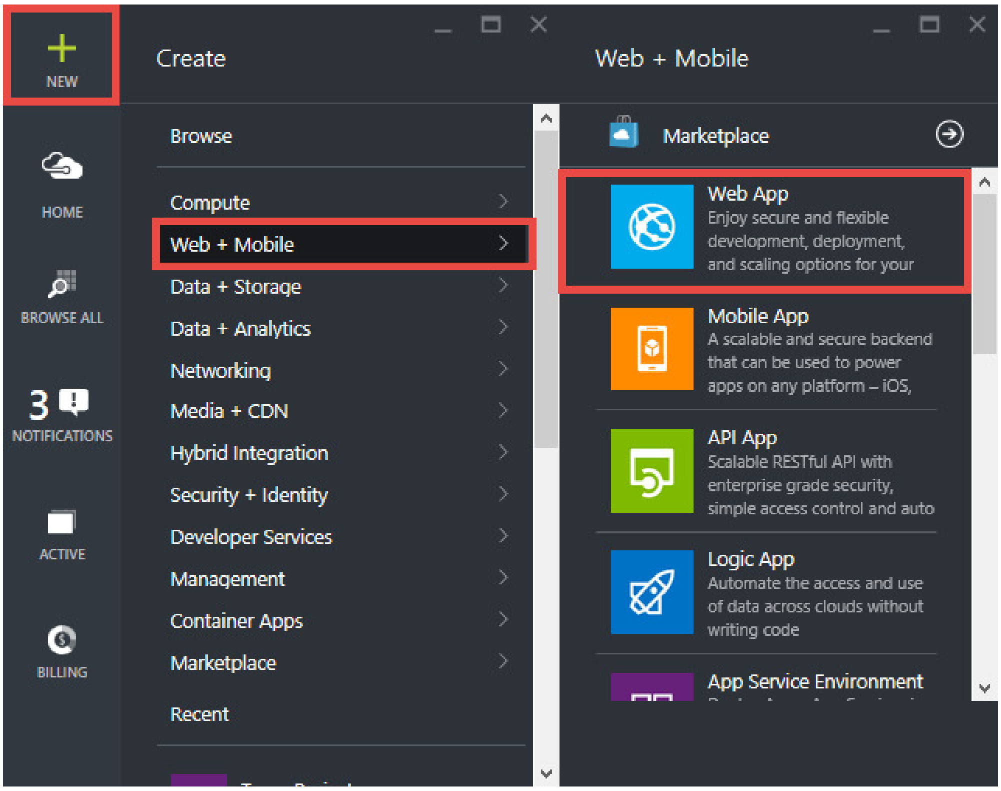


B. Create a web app in the Azure Portal

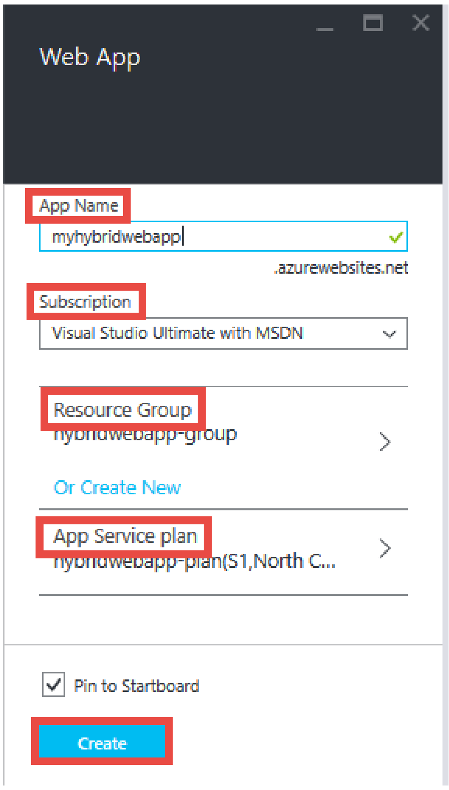
Note

If you have already created a web app in the Azure Portal that you want to use for this tutorial, you can skip ahead to Create a Hybrid Connection and a BizTalk Service and continue from there.

1. In the Azure Portal, click **New** > **Web + Mobile** > **Web app**.


NEW
HOME
BROWSE ALL
3 NOTIFICATIONS
ACTIVE
BILLING
Create
Browse
Compute
Web + Mobile
Data + Storage
Data + Analytics
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Container Apps
Marketplace
Recent
Web + Mobile
Marketplace
Web App
Enjoy secure and flexible development, deployment, and scaling options for your
Mobile App
A scalable and secure backend that can be used to power apps on any platform – iOS,
API App
Scalable RESTful API with enterprise grade security, simple access control and auto
Logic App
Automate the access and use of data across clouds without writing code
App Service Environment

2. Configure your web app, and then click **Create**.


Web App

App Name
myhybridwebapp
.azurewebsites.net

Subscription
Visual Studio Ultimate with MSDN

Resource Group
hybridwebapp-group

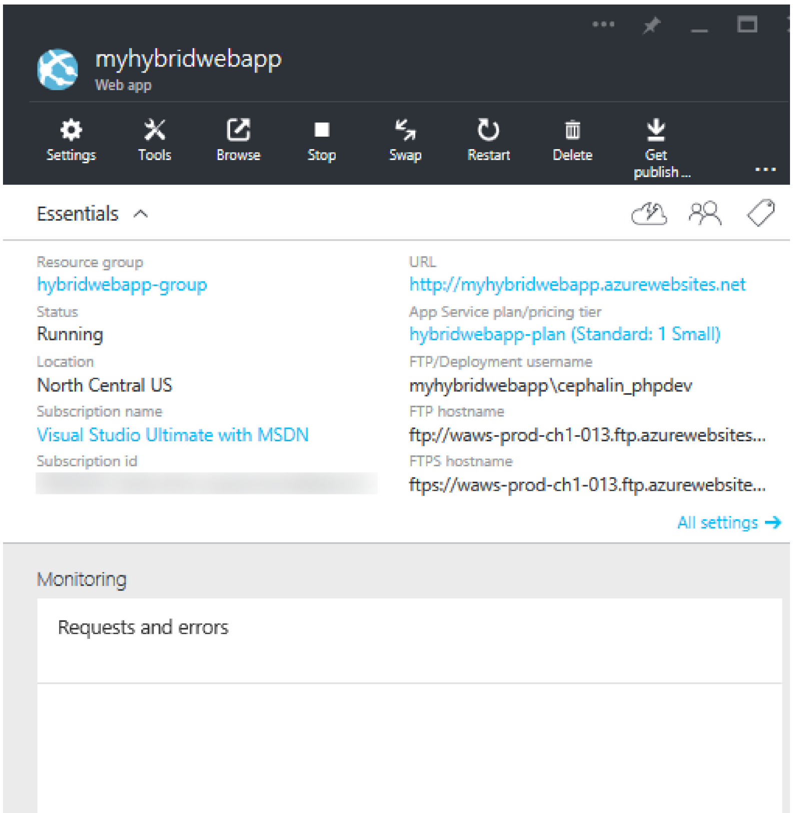
Or Create New

App Service plan
hybridwebapp-plan(S1,North C...

☑ Pin to Startboard

Create

3. After a few moments, the web app is created and its web app blade appears. The blade is a vertically scrollable dashboard that lets you manage your web app.



Resource group: hybridwebapp-group

Status: Running

Location: North Central US

Subscription name: Visual Studio Ultimate with MSDN

Subscription id: [redacted]

URL: http://myhybridwebapp.azurewebsites.net

App Service plan/pricing tier: hybridwebapp-plan (Standard: 1 Small)

FTP/Deployment username: myhybridwebapp\cephalin_phpdev

FTP hostname: ftp://waws-prod-ch1-013.ftp.azurewebsites...

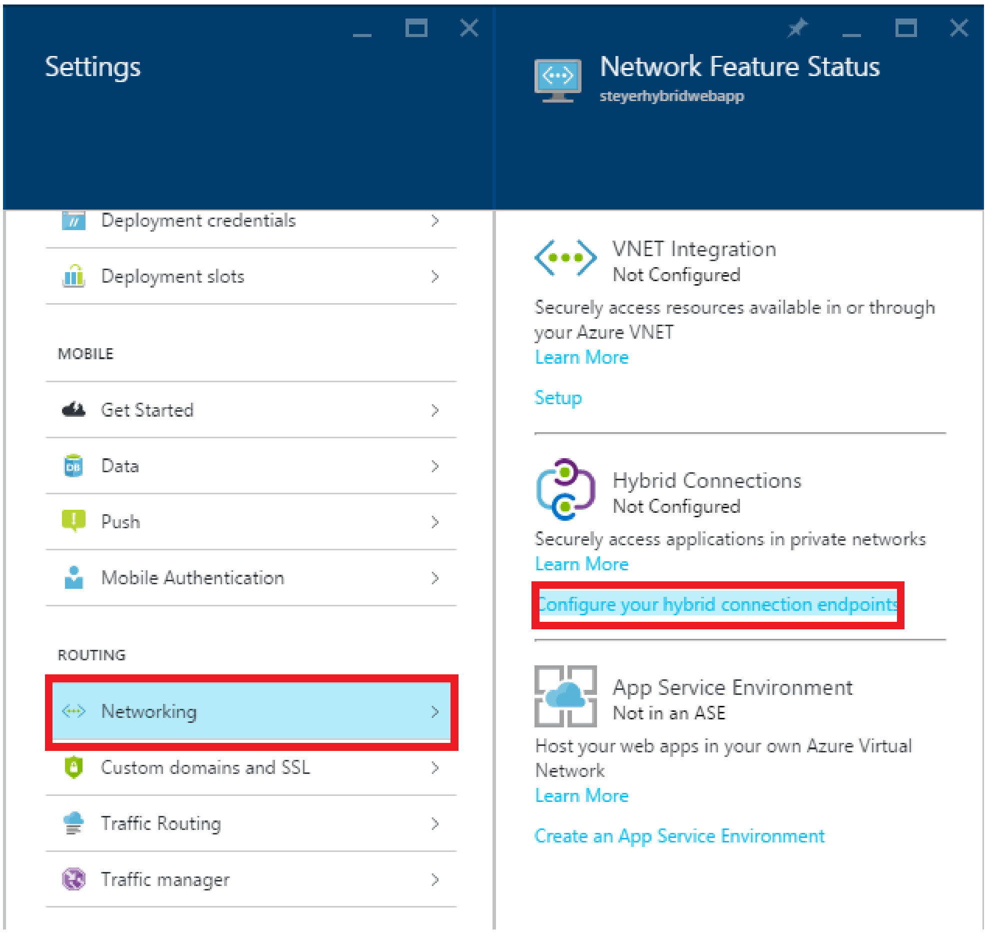
FTPS hostname: ftps://waws-prod-ch1-013.ftp.azurewebsite...

To verify the web app is live, you can click the **Browse** icon to display the default page.

Next, you will create a hybrid connection and a BizTalk service for the web app.

C. Create a Hybrid Connection and a BizTalk Service

1. Back in the Portal, go to settings and click **Networking** > **Configure your hybrid connection endpoints**.



VNET Integration: Not Configured

Securely access resources available in or through your Azure VNET

Hybrid Connections: Not Configured

Securely access applications in private networks

App Service Environment: Not in an ASE

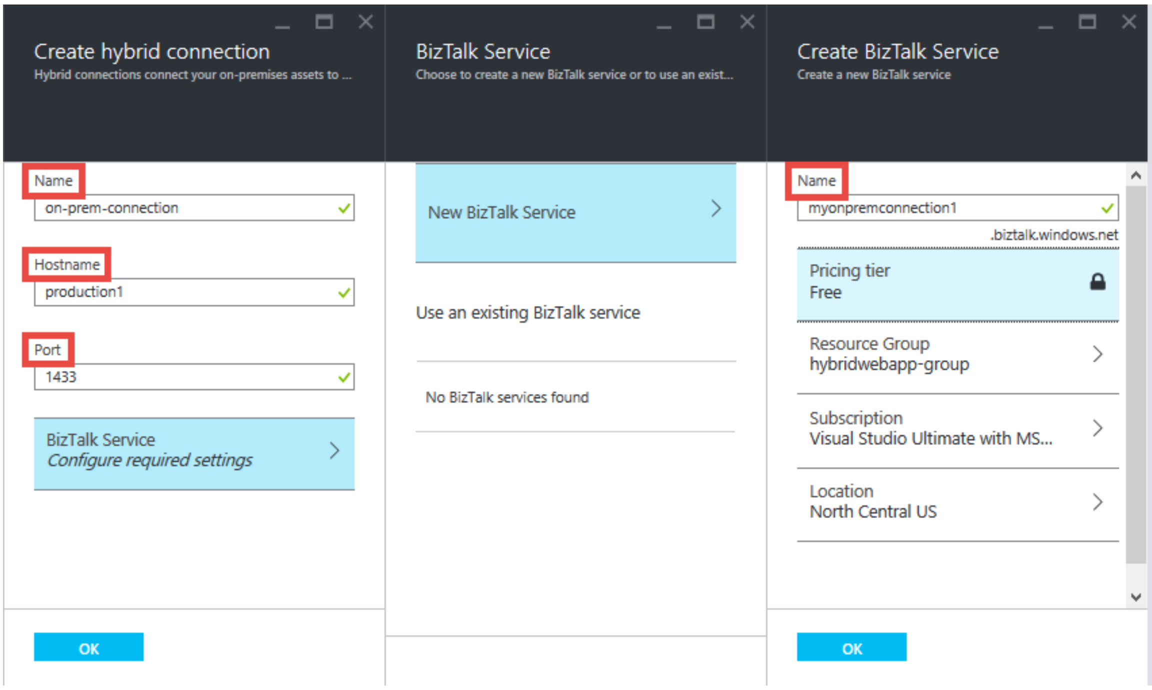
Host your web apps in your own Azure Virtual Network

2. On the Hybrid connections blade, click **Add** > **New hybrid connection**.

3. On the **Create hybrid connection** blade:

- For **Name**, provide a name for the connection.
- For **Hostname**, enter the computer name of your SQL Server host computer.
- For **Port**, enter 1433 (the default port for SQL Server).

- Click **BizTalk Service** > **New BizTalk Service** and enter a name for the BizTalk service.



Create hybrid connection
Hybrid connections connect your on-premises assets to ...

Name: on-prem-connection
Hostname: production1
Port: 1433
BizTalk Service: *Configure required settings*

BizTalk Service
Choose to create a new BizTalk service or to use an exist...

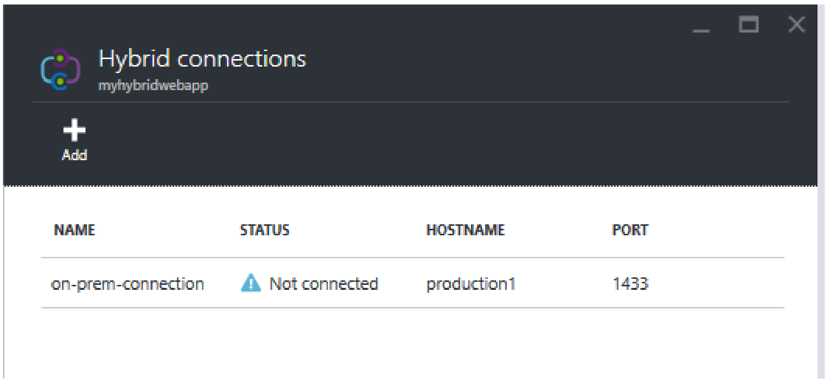
New BizTalk Service
Use an existing BizTalk service
No BizTalk services found

Create BizTalk Service
Create a new BizTalk service

Name: myonpremconnection1 .biztalk.windows.net
Pricing tier: Free
Resource Group: hybridwebapp-group
Subscription: Visual Studio Ultimate with MS...
Location: North Central US

4. Click **OK** twice.

When the process completes, the **Notifications** area will flash a green **SUCCESS** and the **Hybrid connection** blade will show the new hybrid connection with the status as **Not connected**.



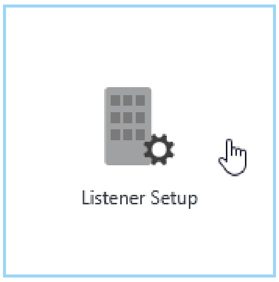
Hybrid connections
myhybridwebapp

NAME	STATUS	HOSTNAME	PORT
on-prem-connection	⚠ Not connected	production1	1433

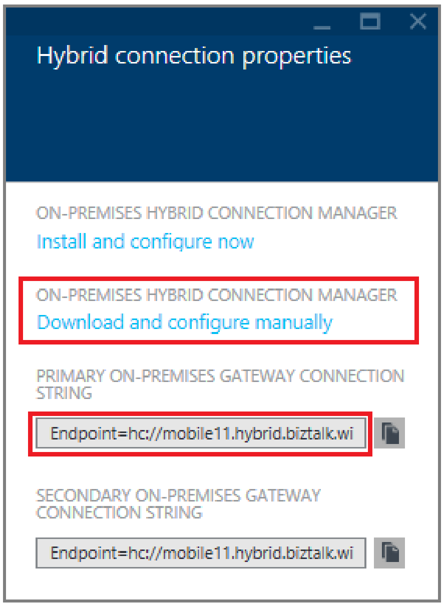
At this point, you have completed an important part of the cloud hybrid connection infrastructure. Next, you will create a corresponding on-premises piece.

D. Install the on-premises Hybrid Connection Manager to complete the connection

1. In the **Hybrid connections** blade, click the hybrid connection you just created, then click **Listener Setup**.



2. The **Hybrid connection properties** blade opens. Under **On-premises Hybrid Connection Manager**, choose **download and configure manually**, save the downloaded HybridConnectionManager.msi package, and copy the gateway connection string.



Hybrid connection properties

ON-PREMISES HYBRID CONNECTION MANAGER
Install and configure now

ON-PREMISES HYBRID CONNECTION MANAGER
Download and configure manually

PRIMARY ON-PREMISES GATEWAY CONNECTION STRING
Endpoint=hc://mobile11.hybrid.biztalk.wi

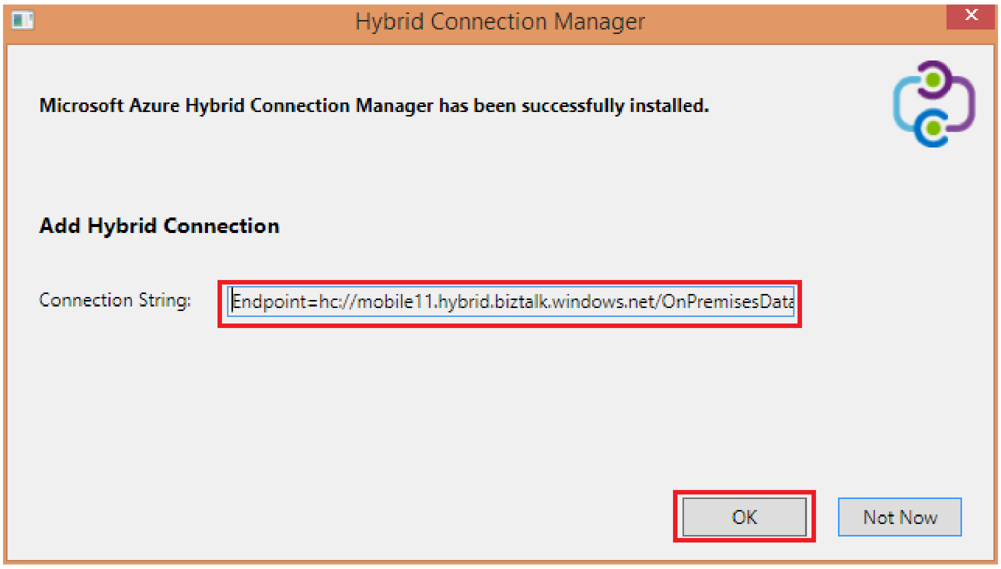
SECONDARY ON-PREMISES GATEWAY CONNECTION STRING
Endpoint=hc://mobile11.hybrid.biztalk.wi

3. From an administrator command prompt, type the following command to start the installer:

```
start HybridConnectionManager.msi
```

4. After the installer runs, click **Not now**, then browse to the %ProgramFiles%\Microsoft\HybridConnectionManager folder, run HCMConfigWizard.exe and click **Yes** in the **User Account Control** dialog.

5. Paste the hybrid connection string that you copied earlier and click **OK**.



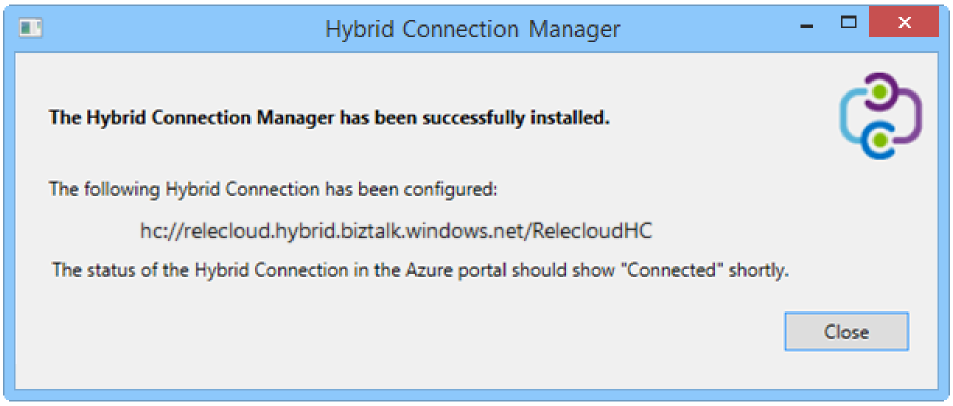
Hybrid Connection Manager

Microsoft Azure Hybrid Connection Manager has been successfully installed.

Add Hybrid Connection

Connection String: Endpoint=hc://mobile11.hybrid.biztalk.windows.net/OnPremisesData

6. When the install completes, click **Close**.



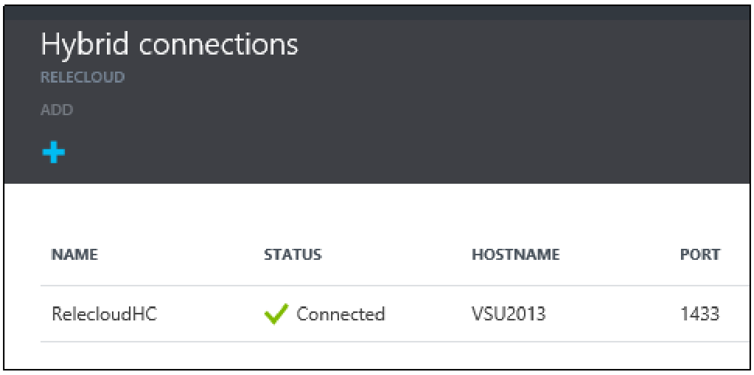
The Hybrid Connection Manager has been successfully installed.

The following Hybrid Connection has been configured:

hc://relecloud.hybrid.biztalk.windows.net/RelecloudHC

The status of the Hybrid Connection in the Azure portal should show "Connected" shortly.

On the **Hybrid connections** blade, the **Status** column now shows **Connected**.



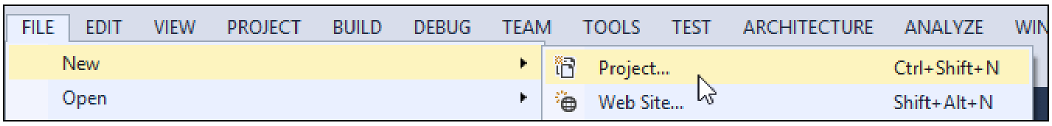
NAME	STATUS	HOSTNAME	PORT
RelecloudHC	✓ Connected	VSU2013	1433

Now that the hybrid connection infrastructure is complete, you will create a web application that uses it.

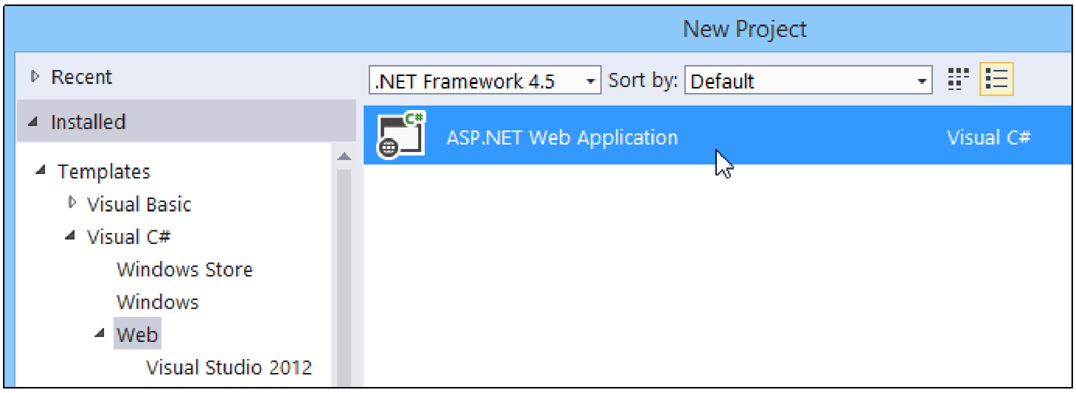
E. Create a basic ASP.NET web project, edit the database connection string, and run the project locally

Create a basic ASP.NET project

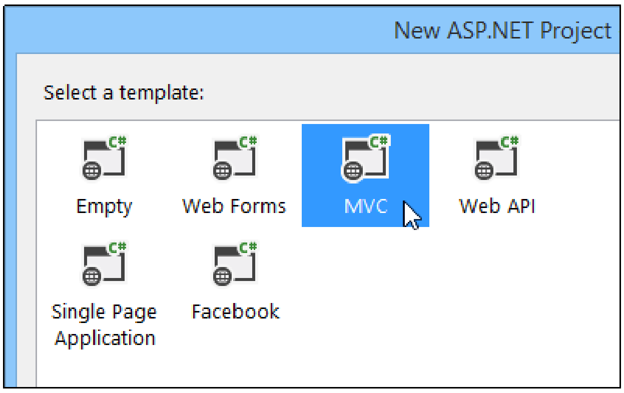
1. In Visual Studio, on the **File** menu, create a new Project:



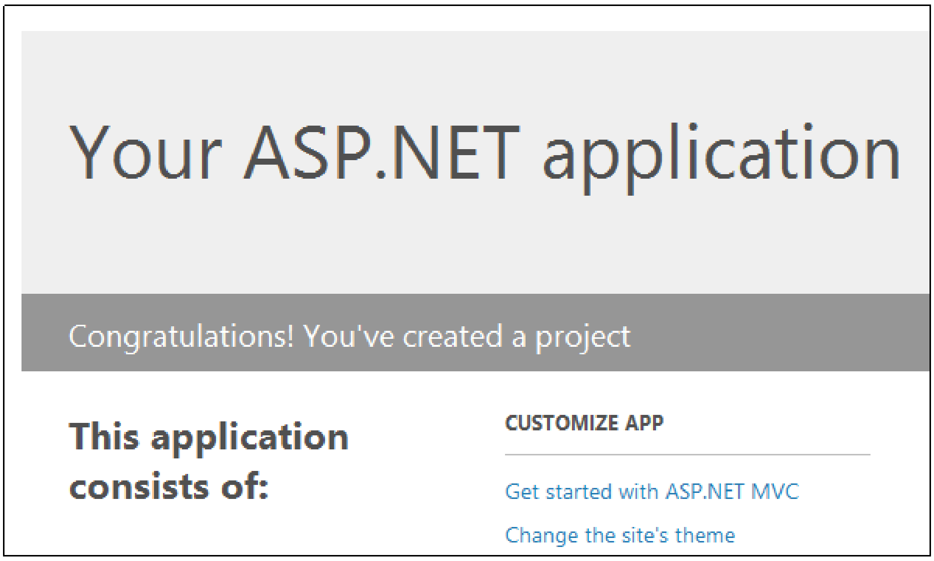
2. In the **Templates** section of the **New Project** dialog, select **Web** and choose **ASP.NET Web Application**, and then click **OK**.



3. In the **New ASP.NET Project** dialog, choose **MVC**, and then click **OK**.



4. When the project has been created, the application readme page appears. Do not run the web project yet.



Your ASP.NET application

Congratulations! You've created a project

This application consists of:

CUSTOMIZE APP

Get started with ASP.NET MVC

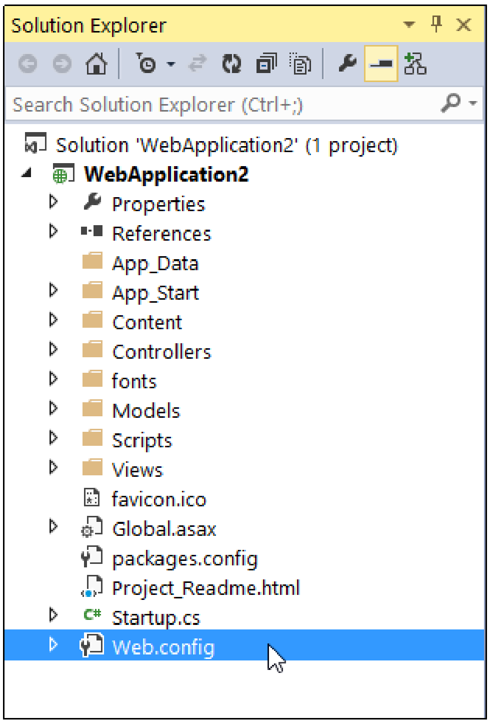
Change the site's theme

Edit the database connection string for the application

In this step, you edit the connection string that tells your application where to find your local SQL Server Express database. The connection string is in the application's Web.config file, which contains configuration information for the application.

> **Note**
>
> To ensure that your application uses the database that you created in SQL Server Express, and not the one in Visual Studio's default LocalDB, it is important that you complete this step before running your project.

1. In Solution Explorer, double-click the Web.config file.



2. Edit the **connectionStrings** section to point to the SQL Server database on your local machine, following the syntax in the following example:

```
<connectionStrings>
  <add name="DefaultConnection" connectionString="Server=ServerName,1433;
    Database=MembershipDB;User ID=sa;Password=sapassword"
    providerName="System.Data.SqlClient" />
</connectionStrings>
```

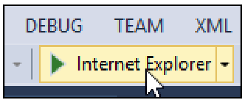
When composing the connection string, keep in mind the following:

- If you are connecting to a named instance instead of a default instance (for example, YourServer\SQLEXPRESS), you must configure your SQL Server to use static ports. For information on configuring static ports, see How to configure SQL Server to listen on a specific port. By default, named instances use UDP and dynamic ports, which are not supported by Hybrid Connections.
- It is recommended that you specify the port (1433 by default, as shown in the example) on the connection string so that you can be sure that your local SQL Server has TCP enabled and is using the correct port.
- Remember to use SQL Server Authentication to connect, specifying the user ID and password in your connection string.

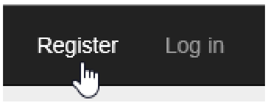
3. Click **Save** in Visual Studio to save the Web.config file.

Run the project locally and register a new user

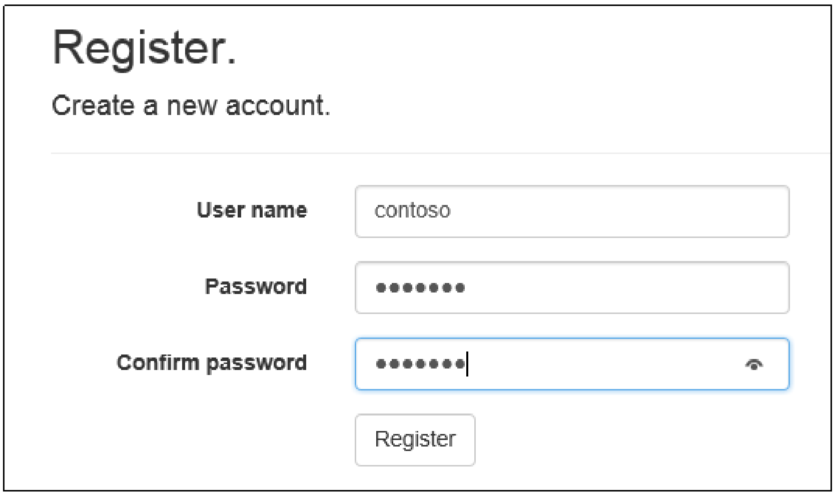
1. Now, run your new web project locally by clicking the browse button under Debug. This example uses Internet Explorer.



2. On the upper right of the default web page, choose **Register** to register a new account:



3. Enter a user name and password:



Register.

Create a new account.

User name: contoso

Password: •••••••

Confirm password: •••••••

Register

This automatically creates a database on your local SQL Server that holds the membership information for your application. One of the tables (**dbo.AspNetUsers**) holds web app user credentials like the ones that you just entered. You will see this table later in the tutorial.

4. Close the browser window of the default web page. This stops the application in Visual Studio.

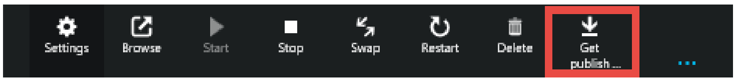
You are now ready for the next step, which is to publish the application to Azure and test it.

F. Publish the web application to Azure and test it

Now, you'll publish your application to your App Service web app and then test it to see how the hybrid connection you configured earlier is being used to connect your web app to the database on your local machine.

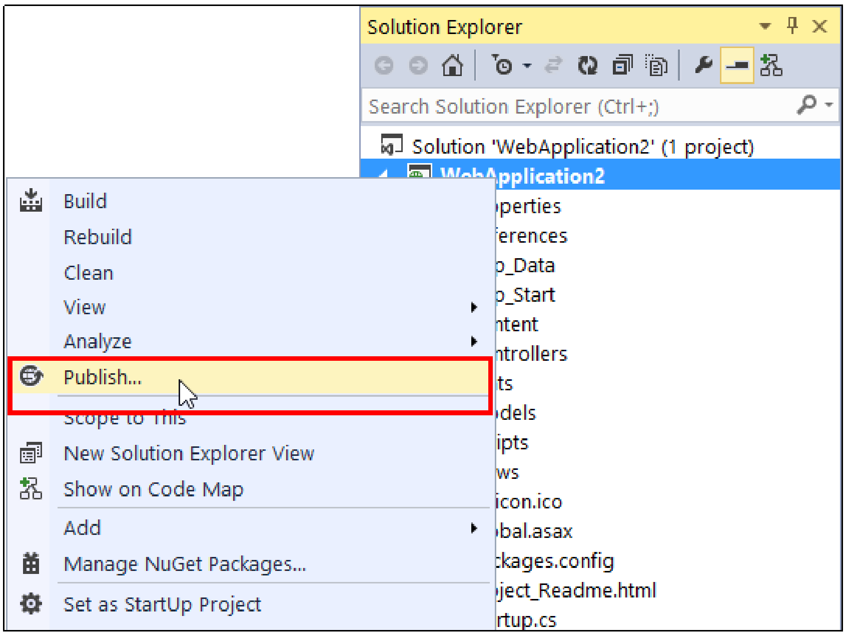
Publish the web application

1. You can download your publishing profile for the App Service web app in the Azure Portal. On the blade for your web app, click **Get publish profile**, and then save the file to your computer.

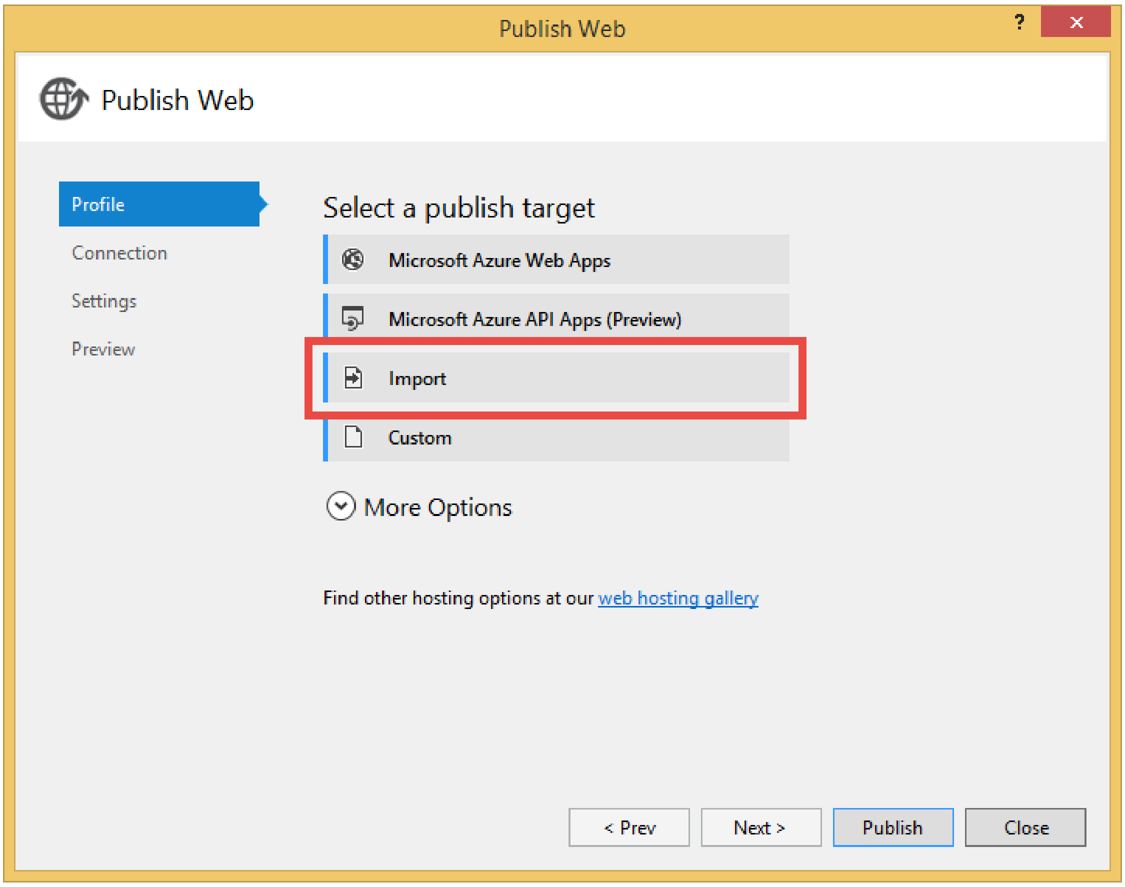


Next, you will import this file into your Visual Studio web application.

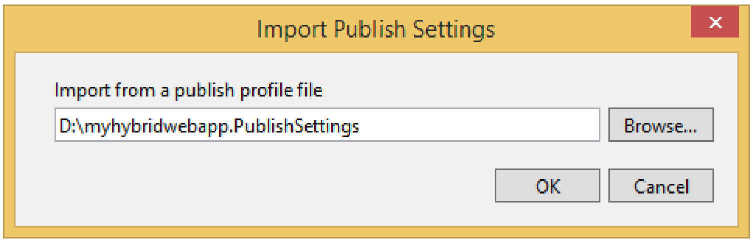
2. In Visual Studio, right-click the project name in Solution Explorer and select **Publish**.



3. In the **Publish Web** dialog, on the **Profile** tab, choose **Import**.

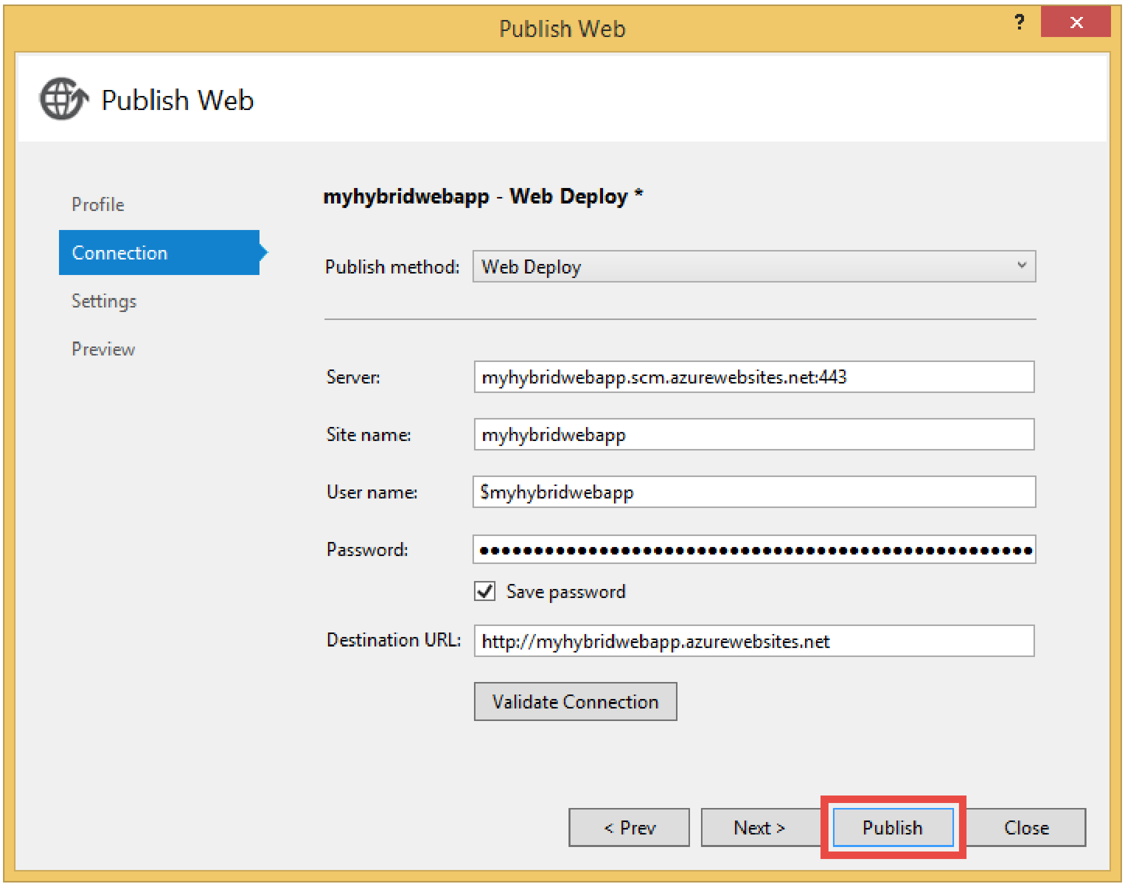


4. Browse to your downloaded publishing profile, select it, and then click **OK**.



D:\myhybridwebapp.PublishSettings

5. Your publishing information is imported and displays on the **Connection** tab of the dialog.



myhybridwebapp - Web Deploy *

Publish method: Web Deploy

Server: myhybridwebapp.scm.azurewebsites.net:443

Site name: myhybridwebapp

User name: $myhybridwebapp

Password: ••

☑ Save password

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

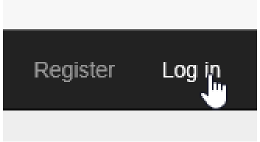
Click **Publish**.

When publishing completes, your browser will launch and show your now familiar ASP.NET application -- except that now it is live in the Azure cloud!

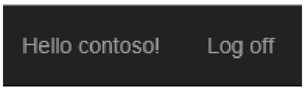
Next, you will use your live web application to see its Hybrid Connection in action.

Test the completed web application on Azure

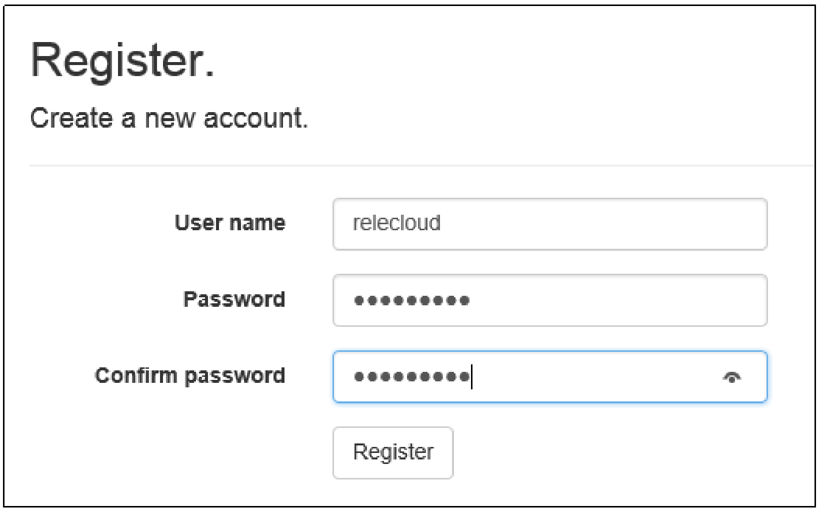
1. On the top right of your web page on Azure, choose **Log in**.



2. Your App Service web app is now connected to your web application's membership database on your local machine. To verify this, log in with the same credentials that you entered in the local database earlier.



3. To further test your new hybrid connection, log off of your Azure web application and register as another user. Provide a new user name and password, and then click **Register**.



Register.

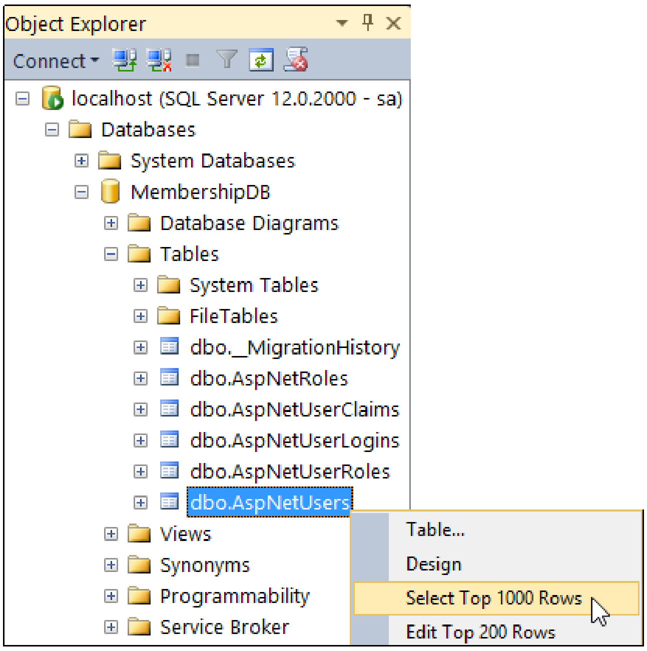
Create a new account.

User name: relecloud

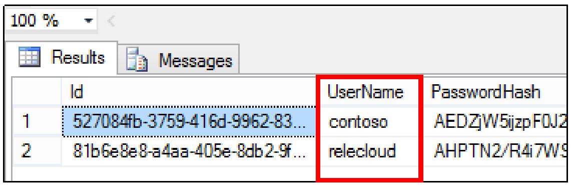
Password: •••••••••

Confirm password: •••••••••

4. To verify that the new user's credentials have been stored in your local database through your hybrid connection, open SQL Management Studio on your local computer. In Object Explorer, expand the **MembershipDB** database, and then expand **Tables**. Right-click the **dbo.AspNetUsers** membership table and choose **Select Top 1000 Rows** to view the results.



5. Your local membership table now shows both accounts - the one that you created locally, and the one that you created in the Azure cloud. The one that you created in the cloud has been saved to your on-premises database through Azure's Hybrid Connection feature.



	Id	UserName	PasswordHash
1	527084fb-3759-416d-9962-83...	contoso	AEDZjW5ijzpF0J2
2	81b6e8e8-a4aa-405e-8db2-9f...	relecloud	AHPTN2/R4i7WS

You have now created and deployed an ASP.NET web application that uses a hybrid connection between a web app in the Azure cloud and an on-premises SQL Server database. Congratulations!

See Also

Hybrid Connections overview

Josh Twist introduces hybrid connections (Channel 9 video)

Hybrid Connections overview

BizTalk Services: Dashboard, Monitor, Scale, Configure, and Hybrid Connection tabs

Building a Real-World Hybrid Cloud with Seamless Application Portability (Channel 9 video)

Access on-premises resources using hybrid connections in Azure App Service

ASP.NET Identity Overview

What's changed

- For a guide to the change from Websites to App Service see: Azure App Service and Its Impact on Existing Azure Services

0 Comments

Sign in **10 people listening**

+ Follow Post comment as...

English

Blog • Privacy & Cookies • Terms of Use • Feedback • Trademarks