AI Live 2018

Build a Microsoft Bot Framework bot with the Bot Builder SDK v4

Using Visual Studio 2017, and ASP.NET Core

Estimated time to complete: **30 Minutes**

This walkthrough will show you how you can deploy your Bot built in the previous lab with Visual Studio 2017 to Microsoft Azure.

Contents

What You Need

- The bot created in the previous lab working locally.

Deploy Your Bot to Azure

First you will deploy your bot to Azure using Azure App Service. Then you'll configure your bot with the Azure Bot Service using the *Bot Channels Registration* item. Once you update your Bot's settings, you can test it in production.

Publish from Visual Studio

First use Visual Studio to create your resources in Azure and publish your code.

1. In the *Solution Explorer* window, right click on your bot's **project** node and select **Publish**.
2. In the *Pick a publish target* dialog, ensure **App Service** is selected on the left and **Create New** is selected in the middle.
3. Click the **Publish** button.
4. In upper right of the dialog, ensure the dialog is showing the correct user ID for your Azure subscription.
5. The *Create App Service* dialog is very similar to the Azure Portal blade you used earlier, albeit not complete.
6. Fill out the dialog:
    a. App Name—this will be part of your bot's URL and is NOT your bot's name. You will set it later.
    b. Subscription—choose the previously used subscription
    c. Resource Group—You can use the group you created earlier
    d. Hosting Plan—You can use the plan you created earlier
    e. Application Insights—**None**
7. When ready, click **Create**. It can take a few minutes to complete the process.

   Once complete, your default web browser will open showing your bot's public URL. If it doesn't open, you'll find the URL in the middle of the *Publish* window to the left of the **Site URL** label.

8. Make a copy of this URL (it will be something like https://yourbotname.azurewebsites.net/) and save it somewhere.

   Note You'll need to use the HTTPS version of the URL when registering your bot. Azure provides SSL support with Azure App Service.

Create your Bot Channels Registration

With your bot deployed in Azure you need to register it with the Azure Bot Service.

1. Access the Azure Portal at https://portal.azure.com.
2. Log in using the same identity you used earlier from Visual Studio to publish your bot.
3. Click **Create a resource**.
4. In the **Search the Marketplace** field type **Bot Channels Registration** and press **Enter**.
5. In the returned list, select **Bot Channels Registration**:
6. Click **Create** in the blade that opens.
7. Provide a **Name** for your bot.
8. Choose the same **Subscription** where you deployed your bot's code.
9. Pick your *existing* **Resource group** which set the location to match your original choice.
10. You can choose the **F0 Pricing tier** for development and testing.
11. Enter your bot's URL. Make sure you start with HTTPS **and** that you add the **/api/messages** suffix. For example https://yourbotname.azurewebsites.net/api/messages

12. Set **Application Insights** to the correct location.
13. Click the **Microsoft App ID and password** item.
14. In the new blade click **Create New**.
15. In the new blade that opens to the right, click the **Create App ID in the App Registration Portal** link. A new browser tab opens.

    Note: You may be required to authenticate again.

16. In the new tab, make a copy of the **App ID** and save it somewhere.
17. Click the **Generate an app password to continue button**.
18. A browser dialog opens and provides you with your app's password.

    Copy and save this secret somewhere you can get to later.

19. Click **OK** once you've copied and saved the password.
20. Just close the browser tab and return to the Azure Portal tab.
21. Paste in your App ID and Password into the correct fields and click **OK**.
22. Last but not least click the **Create** button to set up your channel registration.

    This can take a few seconds to a few minutes.

23. Wait for the process to complete, watching the Notification pane. When done continue on to the next section.

Update Your Bot’s Application Settings

In order for your bot to authenticate with the Azure Bot Service, you need to add two settings to your bot’s configuration file and let Azure know how to load it.

1. Return to Visual Studio 2017 and open the **BotConfiguration.bot** file from the *Solution Explorer*.
2. Duplicate the existing block of JSON for the deployment. You file should now look *similar* to this with the **bolded section** being what was duplicated:

```
{
  "name": "botname",
  "services": [
    {
      "type": "endpoint",
      "name": "development",
      "endpoint": "http://localhost:3978/api/messages",
      "appId": "",
      "appPassword": "",
      "id": "1"
    },
    {
      "type": "endpoint",
      "name": "development",
      "endpoint": "http://localhost:3978/api/messages",
      "appId": "",
      "appPassword": "",
      "id": "1"
    }
  ],
  "padlock": "",
  "version": "2.0"
}
```

3. Modify the second block by changing the **name** to **production**.
4. Change the **endpoint** to use your App Service URL.
5. Update the **appID** with your App ID you saved earlier.
6. Update the **appPassword** with your App password you saved earlier.
7. Change the **id** to **2**.

Go to the next page.

8. Your completed file should *resemble* the following:

```
{
  "name": "botname",
  "services": [
    {
      "type": "endpoint",
      "name": "development",
      "endpoint": "http://localhost:3978/api/messages",
      "appId": "",
      "appPassword": "",
      "id": "1"
    },
    {
      "type": "endpoint",
      "name": "production",
      "endpoint": "https://yourbotname.azurewebsites.net/api/messages",
      "appId": "16ace94c-ab12-3abc-4abc-0123a0b9f454",
      "appPassword": "xrdFDSDEER[lwafGWF93%;}",
      "id": "2"
    }
  ],
  "padlock": "",
  "version": "2.0"
}
```

9. Save your changes.
10. Switch to the Azure Portal and open the Resource group where you deployed your code from Visual Studio and locate the App Service for your bot.
11. In the list of options on the right, locate **Application Settings** under the **Settings** header and click it.
12. Scroll until you find the **Application settings** block of values.
13. Click the **Add new setting** link under the existing table of settings.
14. Type **botFilePath** for the name and **./BotConfiguration.bot** for the value.
15. Click **Add new setting** again.
16. Click the **Save** button up at the top of the page.
17. Return to Visual Studio.
18. In the *Solution Explorer* window, right click on your bot's **project** node and select **Publish**.
19. Click the **Publish** button and wait for it to complete and show your bot's home page.
20. Minimize Visual Studio and continue on to the next section.

Test Your Bot in Production

At this point, you can test your bot from Azure using the built-in Web Chat client.

1. Go back to your Resource group in the portal
2. Open your bot.
3. Under **Bot management**, select **Test in Web Chat**.
4. Type a message like **Hello, Web Chat!** and press **Enter**.
5. If all goes well you should see your message echoed back.
6. Try a few more commands like before, including **goodbye**.