
layout: page title: Bot

permalink: bot.html

Create a Bot with the Azure Bot Service

The Azure Bot Service accelerates the process of developing a bot by provisioning a web host with one of five bot templates you can modify in an integrated environment that is purpose-built for bot development.

Create a New Bot Service

To get started, we will create an instance of the bot service.

1. Navigate to the [Azure Portal](#).
2. Select **New** in the menu blade.
3. In the **New** blade, navigate to the **Data + Analytics** category, and select **Bot Service**.
4. In the Bot Service blade, provide the following requested information:
 - ◊ Set **App name** to your bot's name. The name is used as the subdomain when your bot is deployed to the cloud (for example, *mybasicbot.azurewebsites.net*).
 - ◊ Select the subscription to use.
 - ◊ Create a new [resource group](#) named **CapitalOneWITBot**.
 - ◊ Select the [hosting plan](#) and [location](#).

The screenshot shows the 'Bot Service (preview)' creation page in the Azure portal. The page has a dark header with the Microsoft Azure logo and navigation icons. Below the header, there's a sidebar with various service icons. The main content area contains the following fields:

- App name:** A text input field with the placeholder 'Enter a name for your App'. The domain '.azurewebsites.net' is shown to the right.
- Subscription:** A dropdown menu with 'Bot Framework Support' selected.
- Resource Group:** Radio buttons for 'Create new' (selected) and 'Use existing'. Below is an empty text input field.
- Hosting Plan:** A dropdown menu with 'App Service Plan' selected.
- App Service plan/Location:** A dropdown menu with 'DemoSitePlan(South Central US)' selected.
- Pin to dashboard:** An unchecked checkbox.
- Create:** A blue button.
- Automation options:** A link.

5. Click **Create** to create the bot service and deploy it to the cloud.

6. Confirm that the bot service has been deployed.

- Click **Notifications** (the bell icon that is located along the top edge of the Azure portal). The notification will change from **Deployment started** to **Deployment succeeded**.
- After the notification changes to **Deployment succeeded**, click **Go to resource** on that notification.

The screenshot shows the 'Deployments succeeded' notification in the Azure portal. The notification is highlighted with a red box. It contains the following information:

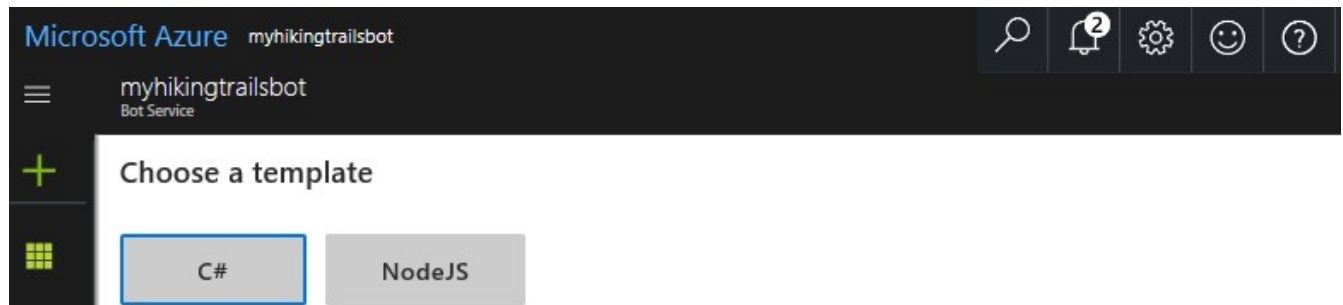
- DESCRIPTION:** Deployment to resource group 'ademobot' was successful.
- STATUS:** Informational
- TIME:** clientNotification-653db49-a34e-43c9-a564-
- CORRELATION IDS:** 3ebb552fb35e
- RESOURCE:**

The notification also includes a green checkmark icon and a 'Go to resource' link.

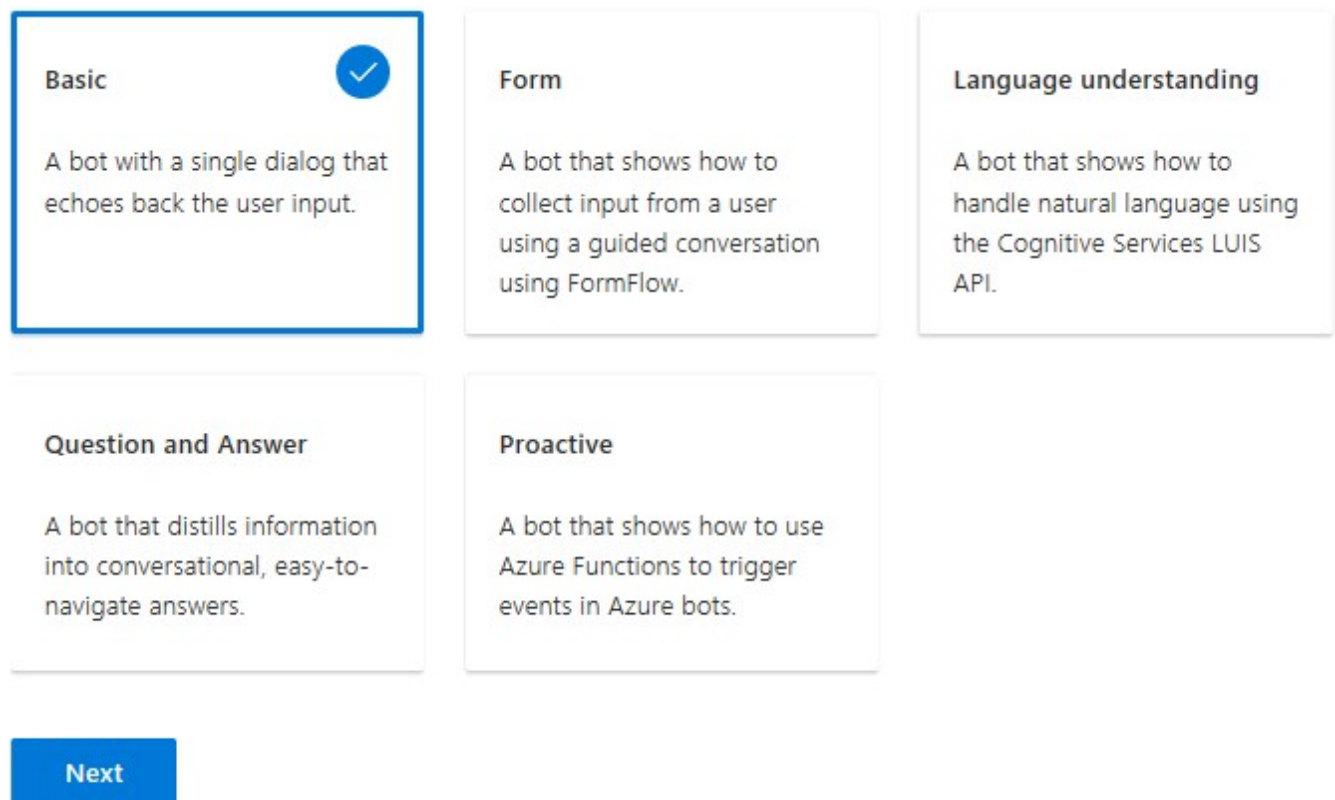
Create a New Bot

Now that our bot service is ready, we can create a bot using the service.

1. On the **Bot Service** blade, choose the programming language that you want to use to develop your bot. For this tutorial, choose the **C#** language.



2. Select the template to use as the starting point for developing your bot. For this tutorial, choose the **Basic** template.



3. Click **Next** to create the bot based on the programming language and template that you've chosen.

Create Bot App ID and password

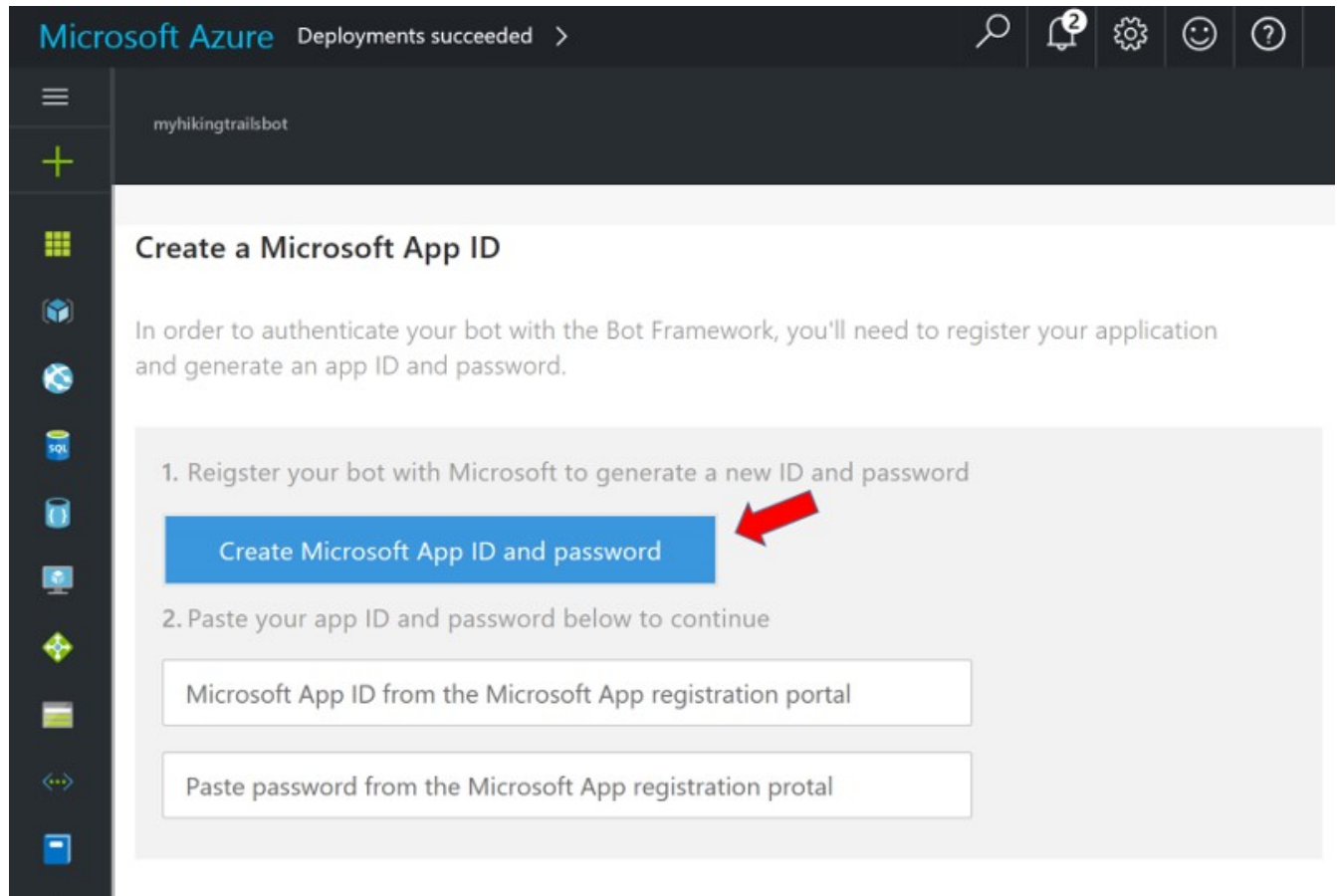
We will need to create authentication credentials for our new bot.

1. Create an app ID and password for your bot, so that it will be able to authenticate with the Bot

Framework.

2. Click **Create Microsoft App ID and password**.

You may be prompted to authenticate at this point.



3. On the page that opens in a new browser tab, record the **App name** and **App ID** values.

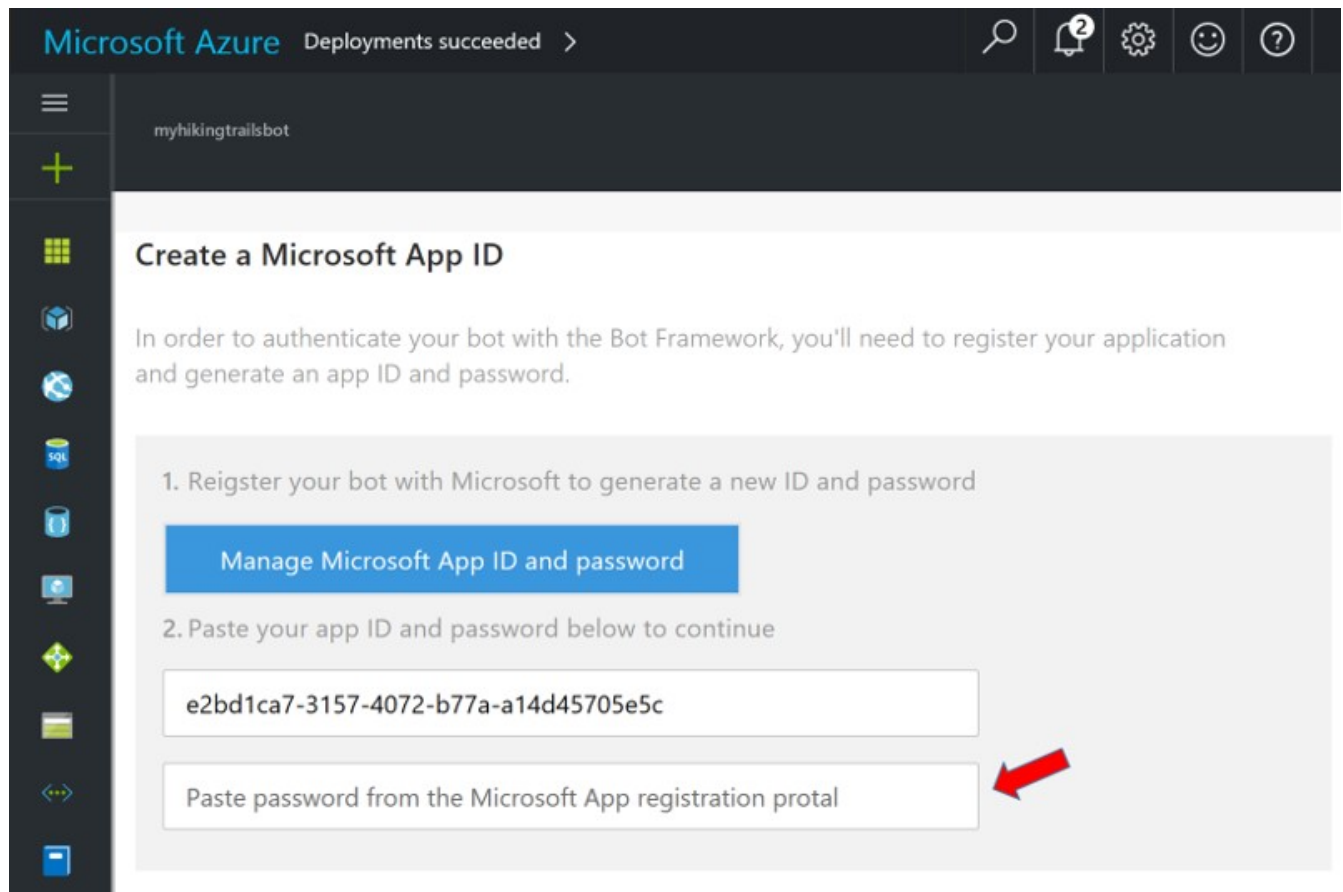
4. Click **Generate an app password to continue**.

5. Copy and securely store the password that is shown, and then click **Ok**.

6. Click **Finish and go back to Bot Framework**.

7. Back in the Azure Portal, assure the **app ID** field is auto-populated for you, and paste the password that you copied (in step 3 above) into the password field.

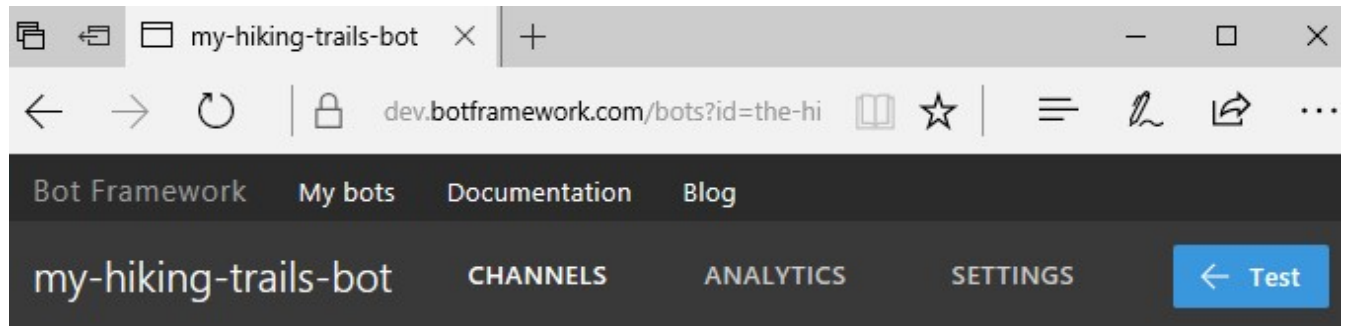
If the **app ID** field is not auto-populated, you can retrieve it by signing in to the [Microsoft Application Registration Portal](#) and copying the application ID from your application's registration settings.



8. Agree to terms, and then click **Create bot**.

When you click **Create bot**, there may be a slight delay before a splash screen renders to indicate that the bot service is generating your bot. *Do not* click **Create bot** again. Please wait for the splash screen to appear.

9. When the bot service finishes generating your bot, the Azure editor will contain the bot's source files. At this point, the bot has been created, registered with the Bot Framework, deployed to the cloud, and is fully functional.
10. If you sign in to the [Bot Framework Portal](#), you'll see that your bot is now listed under **My bots**. Congratulations! You've successfully created a bot by using the Azure Bot Service!



Connect to channels

Name	Health	Published	
 Skype	Running	--	Edit 
 Web Chat	Running	--	Edit 

[Get bot embed codes](#)

Add a channel



Test Your New Bot

Now that your bot is running in the cloud, try it out by typing a few messages into the built-in chat control that's located to the right of the code editor in Azure. You should see that the bot responds to each message you send by echoing back your message prefixed with the text *You said*.

The screenshot shows the Microsoft Azure Bot Service interface for a bot named 'my-biking-bot'. The top navigation bar includes 'BUILD', 'CHANNELS', 'ANALYTICS', and 'SETTINGS', with a 'Test' button on the right. The main content area is titled 'Connect to channels' and displays a table of connected channels:

Name	Health	Published	
Skype	Running	--	Edit
Web Chat	Running	--	Edit

Below the table is a link 'Get bot embed codes' and a section 'Add a channel' with several channel icons. On the right, a 'Chat' window shows a conversation:

- User: I want to go biking!
- Bot: Welcome You!
- User: 1: You said I want to go biking!

The chat window also shows the bot's name 'my-biking-bot' and a timestamp 'my-biking-bot at 2:15:08 PM'.

Up Next: Create your first LUIS app