

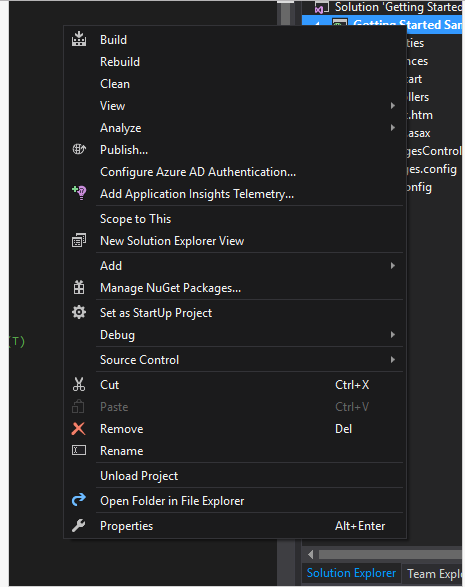
Developing and Deploying Intelligent Chat Bots

1. **Publishing your Bot Application to Microsoft Azure**

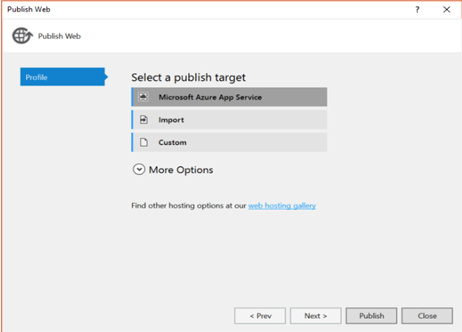
To publish your Bot Application you will need a Microsoft Azure subscription. You can get a free trial from here: azure.microsoft.com/en-us/

Right click on the project and choose "Publish", and then your appropriate Azure subscription information. By default, the bot should be published as an Microsoft Azure App Service. When publishing, keep track of the URL you chose because we'll need it to update the Bot Framework registration endpoint.

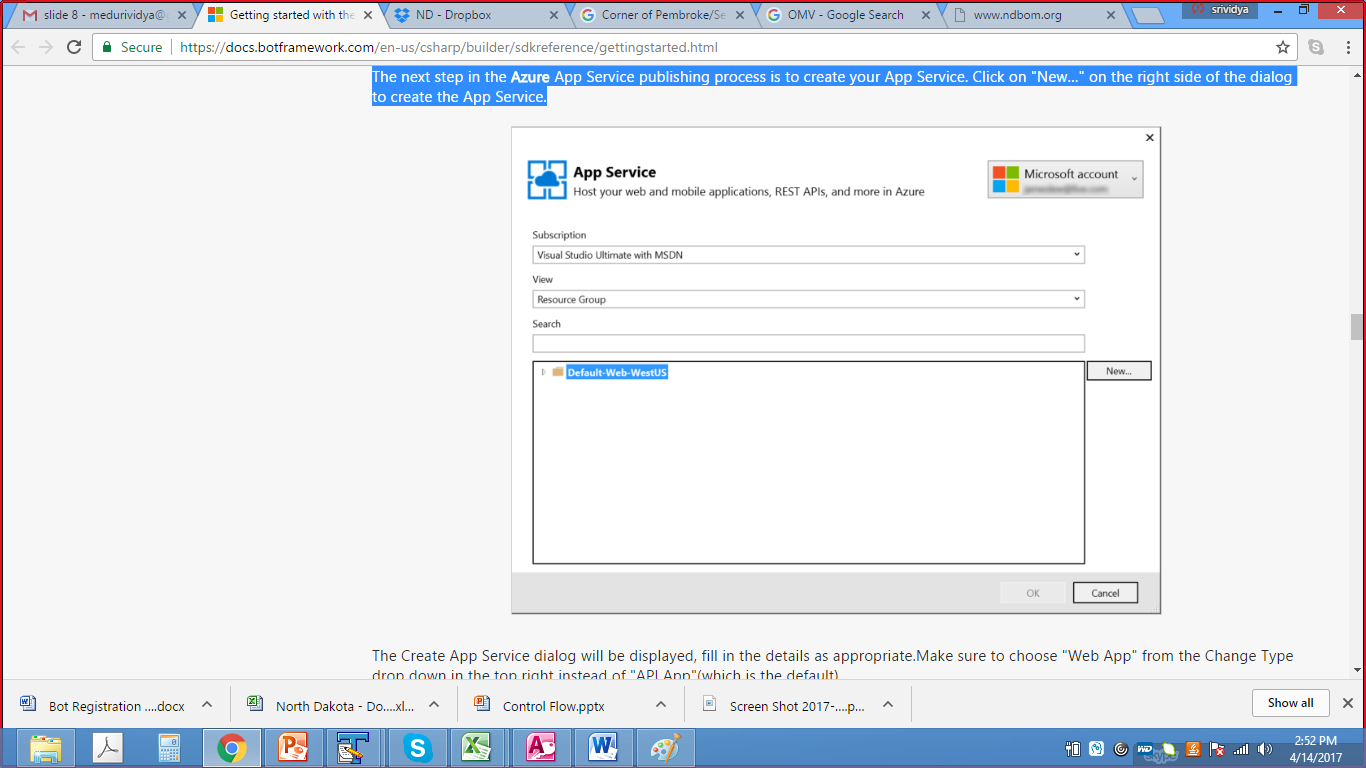
In Visual Studio, right click on the project in Solution Explorer and select "Publish"

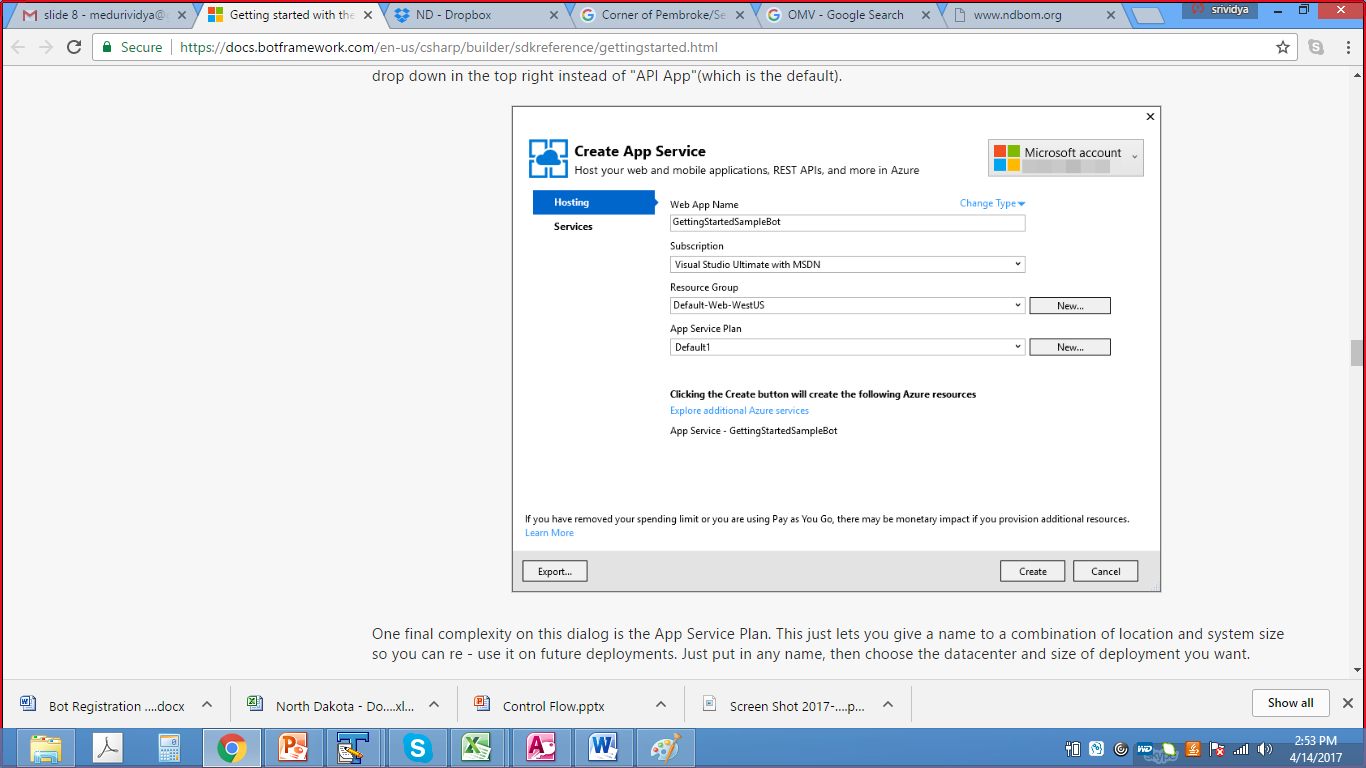


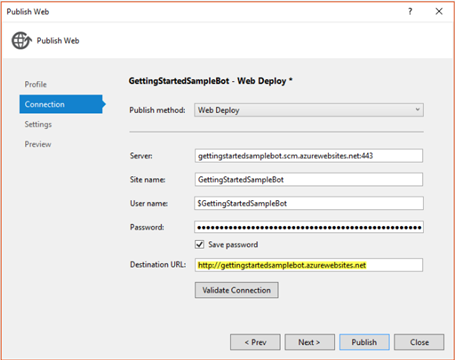
The Publish wizard will start. Select "Microsoft Azure App Service" as your project type.



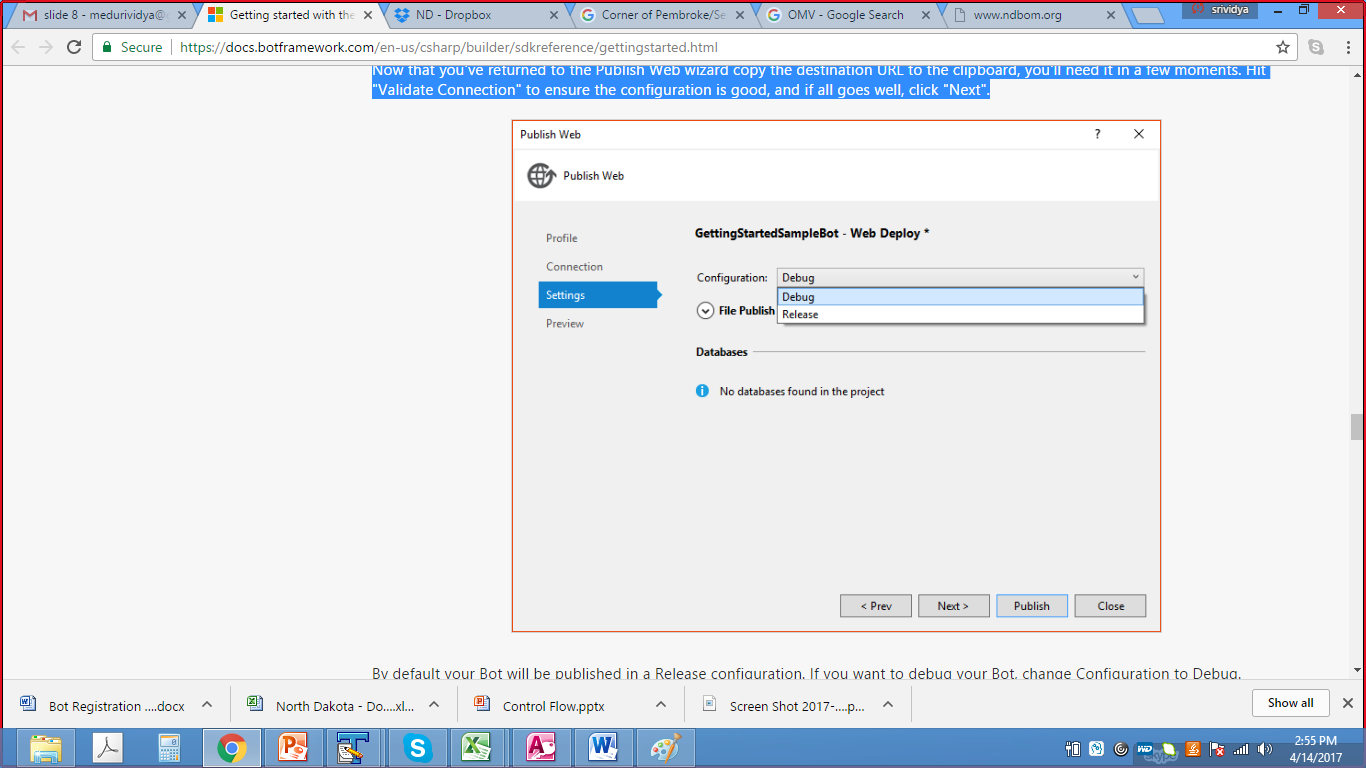
The next step in the Azure App Service publishing process is to create your App Service. Click on "New…" on the right side of the dialog to create the App Service.

The Create App Service dialog will be displayed, fill in the details as appropriate. Make sure to choose "Web App" from the Change Type drop down in the top right.

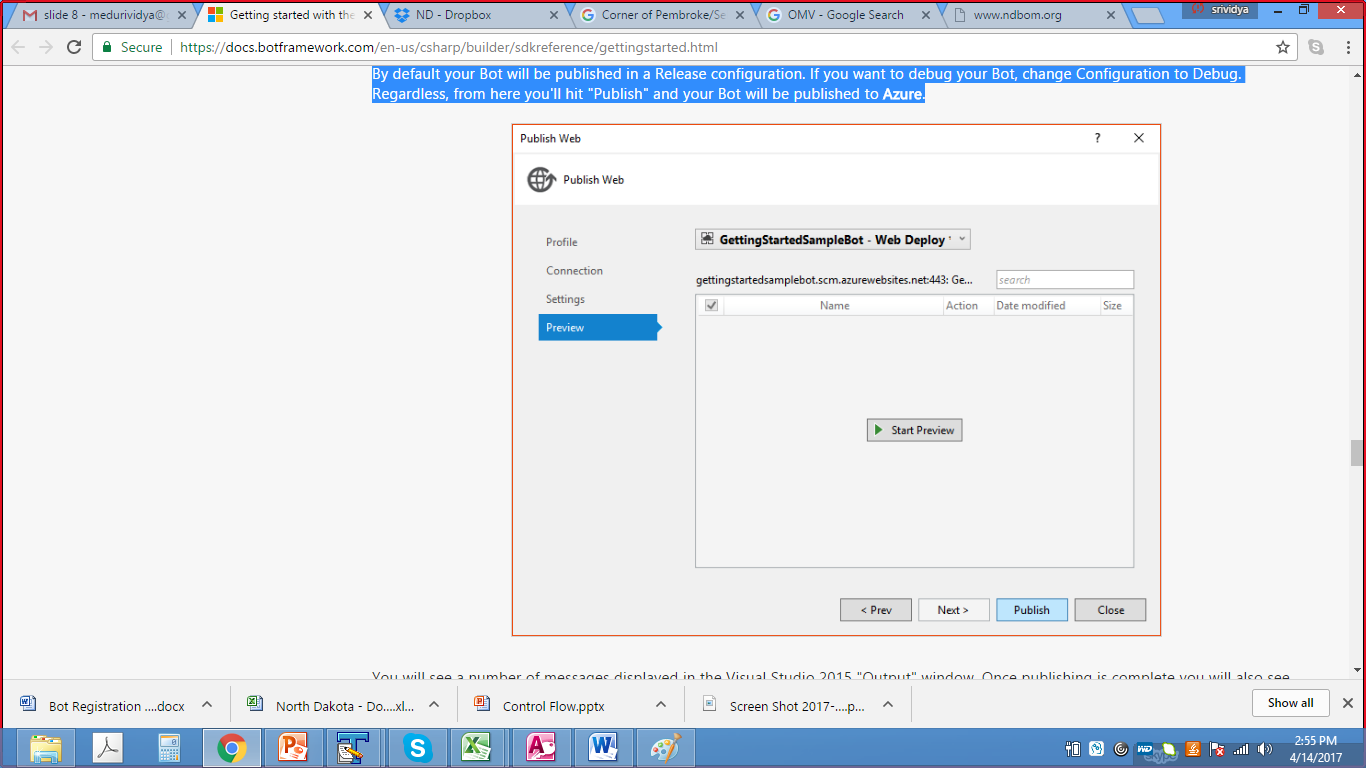




After returning to the Publish Web wizard, copy the destination URL to the clipboard. You will need it for the messaging endpoint during Bot registration (Section 2). Hit "Validate Connection" to ensure the configuration is good, and if all goes well, click "Next".



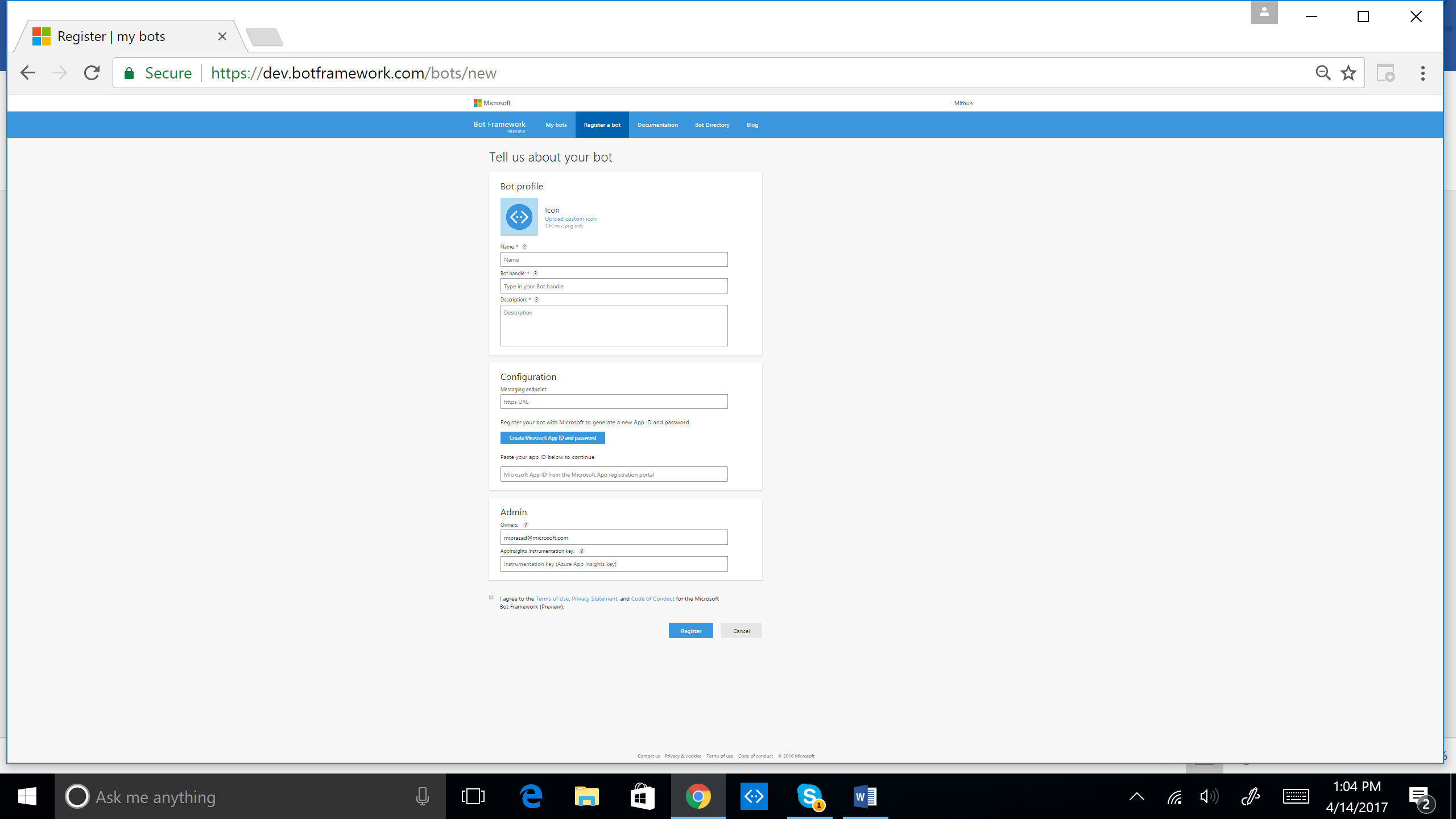
By default your Bot will be published in a Release configuration. If you want to debug your Bot, change Configuration to Debug. Regardless, from here you'll hit "Publish" and your Bot will be published to Azure.



You will see a number of messages displayed in the Visual Studio 2015 "Output" window. Once publishing is complete you will also see the web page for your Bot Application displayed in your browser.

**2. Bot Registration**

Sign in at <https://dev.botframework.com/bots> and click on the “Register a bot” tab. You will see the below page.



Refer to Section 3

Add "/api/messages" to your URL from Azure Web App. Ensure https is used and not http

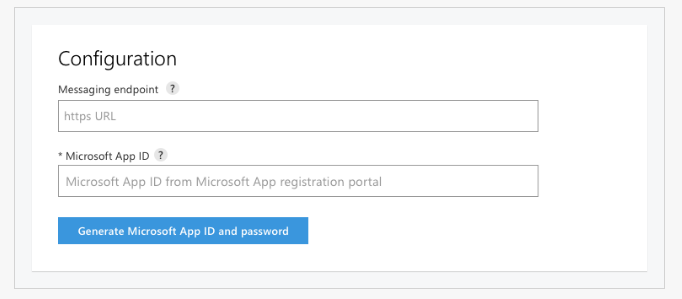
First 46 characters displayed on your bot card in Bot Directory

Used in the URL for your bot (alphanumeric and underscore only)

Unique Name Displayed in Bot Directory (35 character limit)

1. **Configuration**

Once your registration is created, Microsoft Bot Framework will take you through generating your MicrosoftAppId and MicrosoftAppPassword. These are used to generate your MicrosoftAppPassword, be sure to record it somewhere as you won’t be able to see it again.



Change the following keys in the web.config file to match the ones generated when you saved your registration, and you’re ready to build.

<? xml version = "1.0" encoding = "utf-8" ?>

< !--

For more information on how to configure your ASP.NET application, please visit

<http://go.microsoft.com/fwlink/?LinkId=301879>

-->

< configuration >

< appSettings >

< !--update these with your appid and one of your appsecret keys-->

< add key = "MicrosoftAppId" value = "[GUID]" />

< add key = "MicrosoftAppPassword" value = "[PASSWORD]" />

</ appSettings >

1. **Testing the connection to your bot**

In the developer dashboard of your Bot is a test chat window that you can use to interact with your Bot without further configuration, and verify that the Bot Framework can communicate with your Bot's web service.

Note that the first request after your Bot starts up can take 10 - 15 s as Azure starts up the web service for the first time. Subsequent requests will be quick. This simple viewer will let you see the JSON object returned by your Bot.

