Microsoft

GLOBAL AI BOOTCAMP
Letterkenny

Letterkenny .MUG

Getting started with Microsoft Bot Framework and Azure Bot Services

@

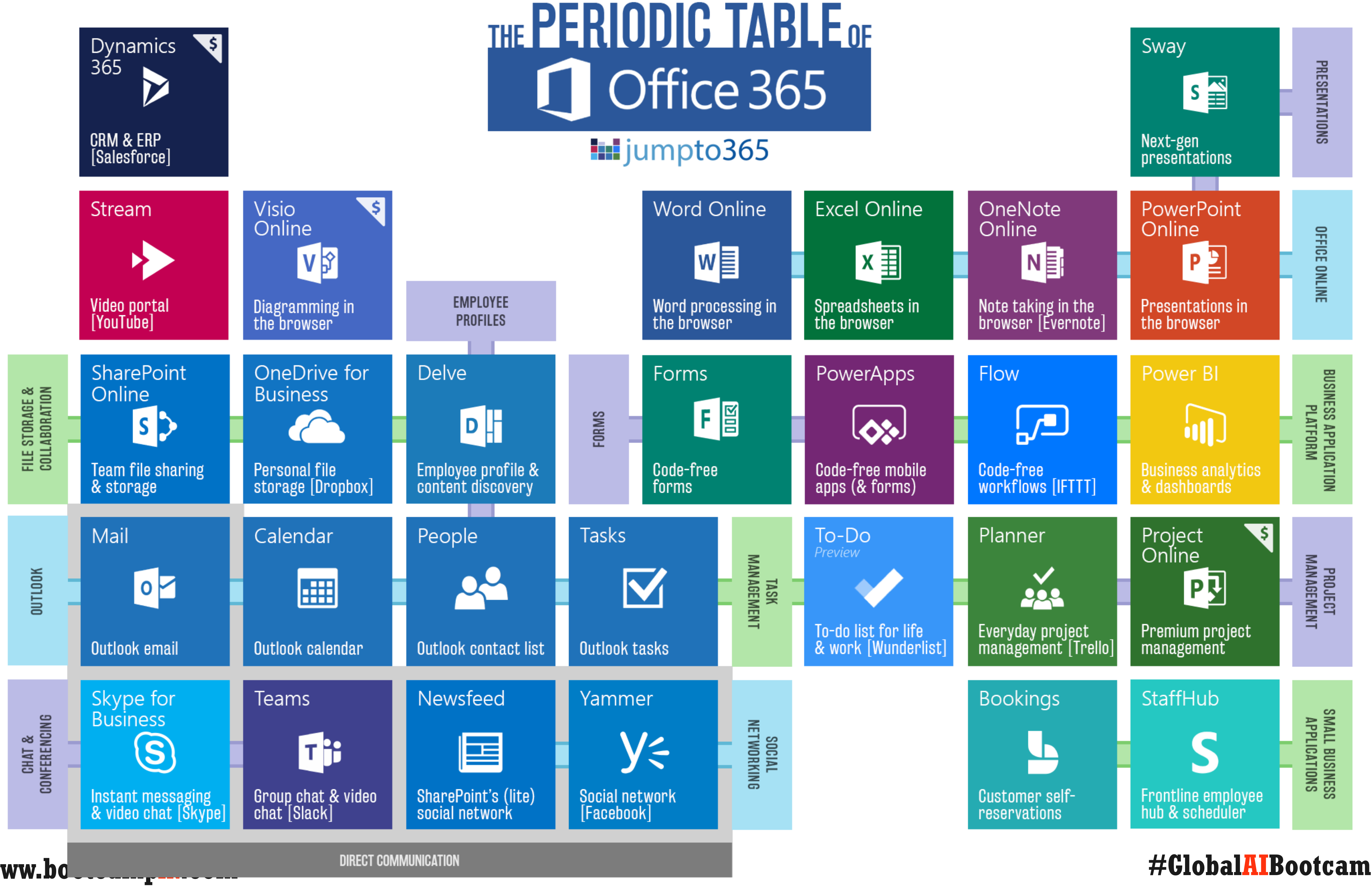
/in/

www.bootcamplk.com

#GlobalAIBootcamp

THE PERIODIC TABLE OF Office 365

jumpto365



Dynamics 365 $ — CRM & ERP [Salesforce]

Sway — Next-gen presentations | PRESENTATIONS

Stream — Video portal [YouTube]

Visio Online $ — Diagramming in the browser

EMPLOYEE PROFILES

Word Online — Word processing in the browser

Excel Online — Spreadsheets in the browser

OneNote Online — Note taking in the browser [Evernote]

PowerPoint Online — Presentations in the browser

OFFICE ONLINE

FILE STORAGE & COLLABORATION

SharePoint Online — Team file sharing & storage

OneDrive for Business — Personal file storage [Dropbox]

Delve — Employee profile & content discovery

FORMS

Forms — Code-free forms

PowerApps — Code-free mobile apps (& forms)

Flow — Code-free workflows [IFTTT]

Power BI — Business analytics & dashboards

BUSINESS APPLICATION PLATFORM

OUTLOOK

Mail — Outlook email

Calendar — Outlook calendar

People — Outlook contact list

Tasks — Outlook tasks

TASK MANAGEMENT

To-Do *Preview* — To-do list for life & work [Wunderlist]

Planner — Everyday project management [Trello]

Project Online $ — Premium project management

PROJECT MANAGEMENT

CHAT & CONFERENCING

Skype for Business — Instant messaging & video chat [Skype]

Teams — Group chat & video chat [Slack]

Newsfeed — SharePoint's (lite) social network

Yammer — Social network [Facebook]

SOCIAL NETWORKING

Bookings — Customer self-reservations

StaffHub — Frontline employee hub & scheduler

SMALL BUSINESS APPLICATIONS

DIRECT COMMUNICATION

www.bo

#GlobalAIBootcamp

Agenda

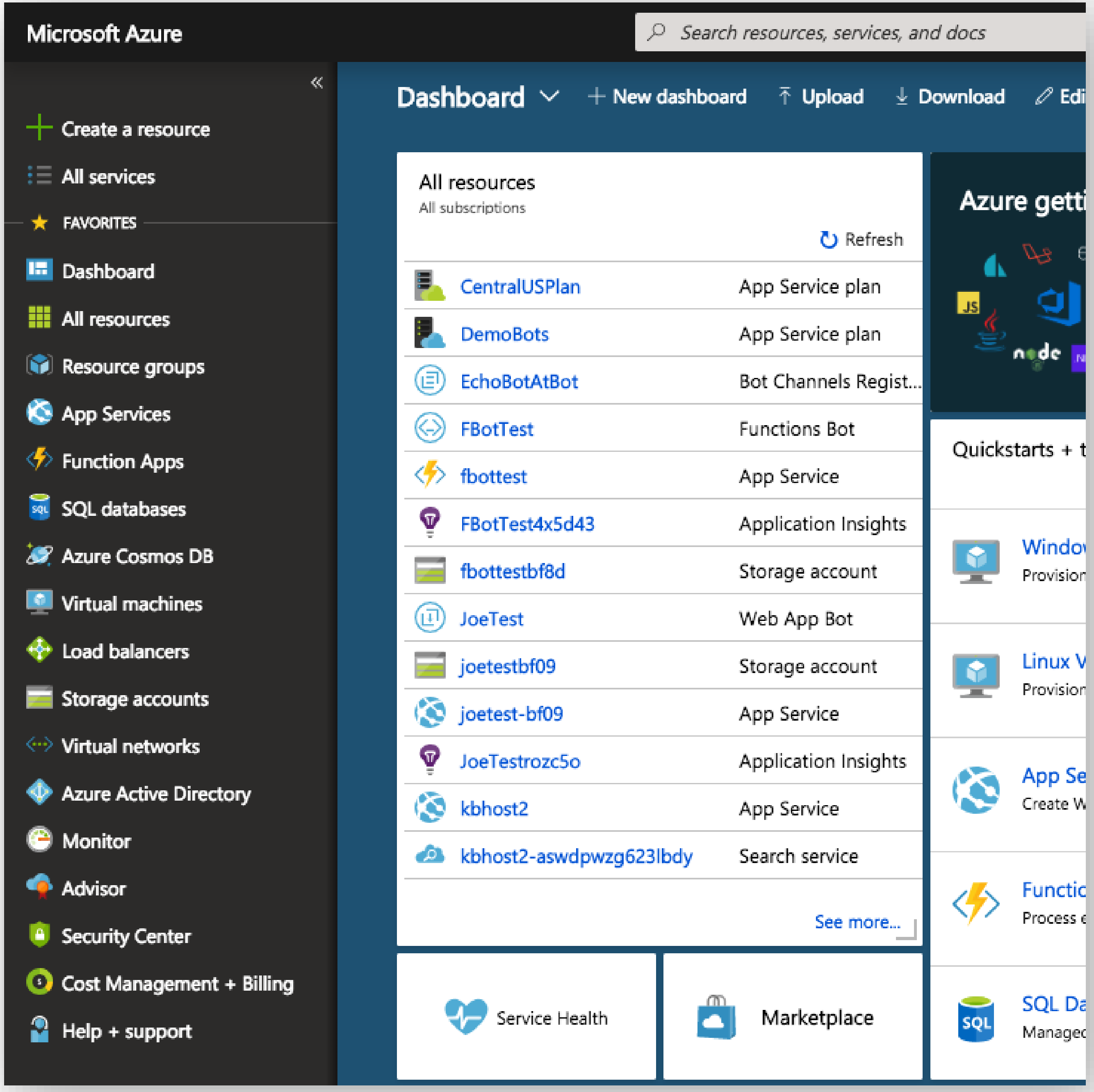
1 **Build a bot (SDK v4.0)**

2 Connect to an FAQ KB

3 Best practices & architecture

Creating a bot has become super easy

Browse to portal.azure.com



All resources
All subscriptions

Refresh

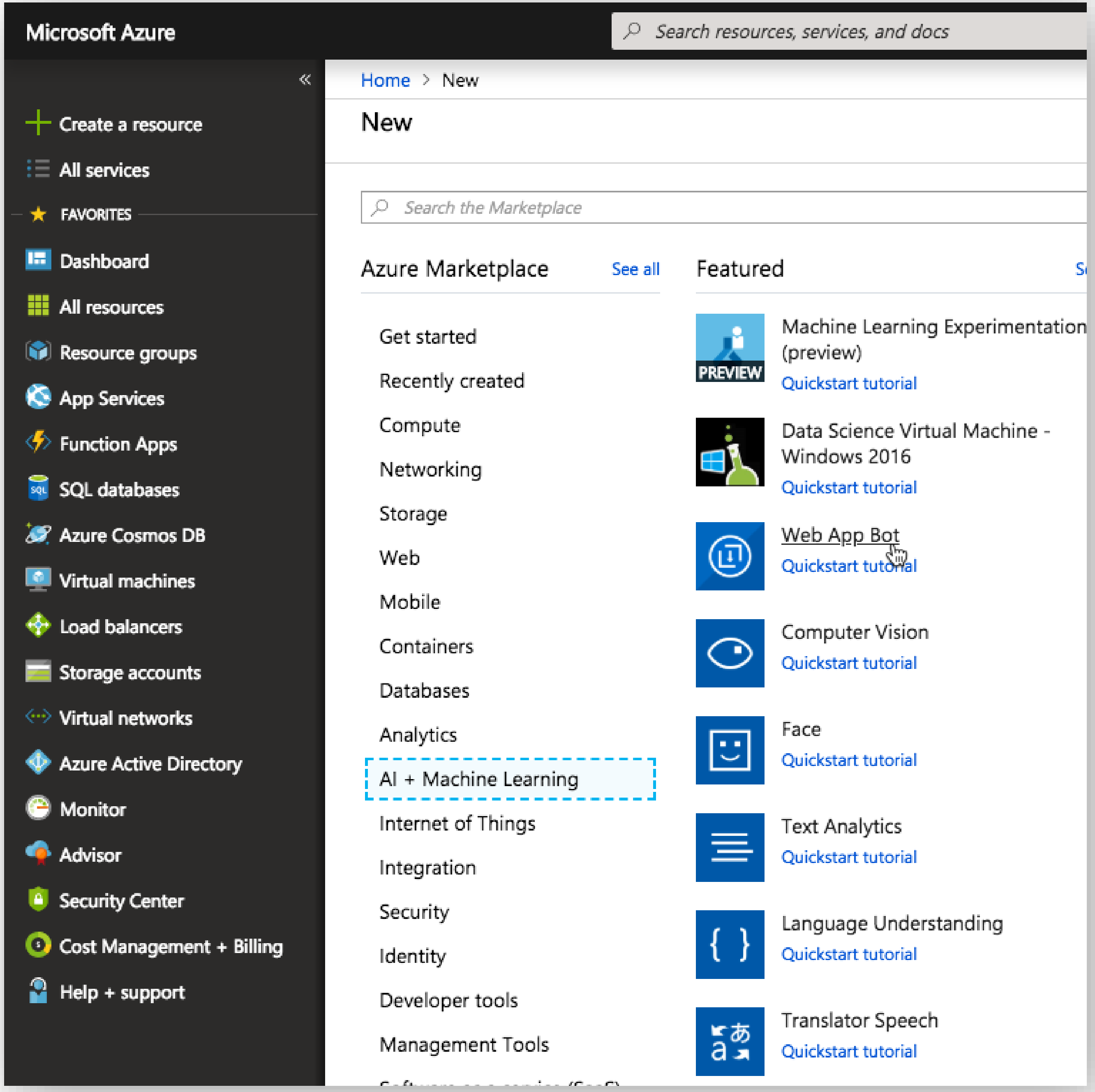
CentralUSPlan	App Service plan
DemoBots	App Service plan
EchoBotAtBot	Bot Channels Regist...
FBotTest	Functions Bot
fbottest	App Service
FBotTest4x5d43	Application Insights
fbottestbf8d	Storage account
JoeTest	Web App Bot
joetestbf09	Storage account
joetest-bf09	App Service
JoeTestrozc5o	Application Insights
kbhost2	App Service
kbhost2-aswdpwzg623lbdy	Search service

See more...

Deploy your bot from the Azure Portal

Create a resource >
AI + Machine Learning >
Web App Bot

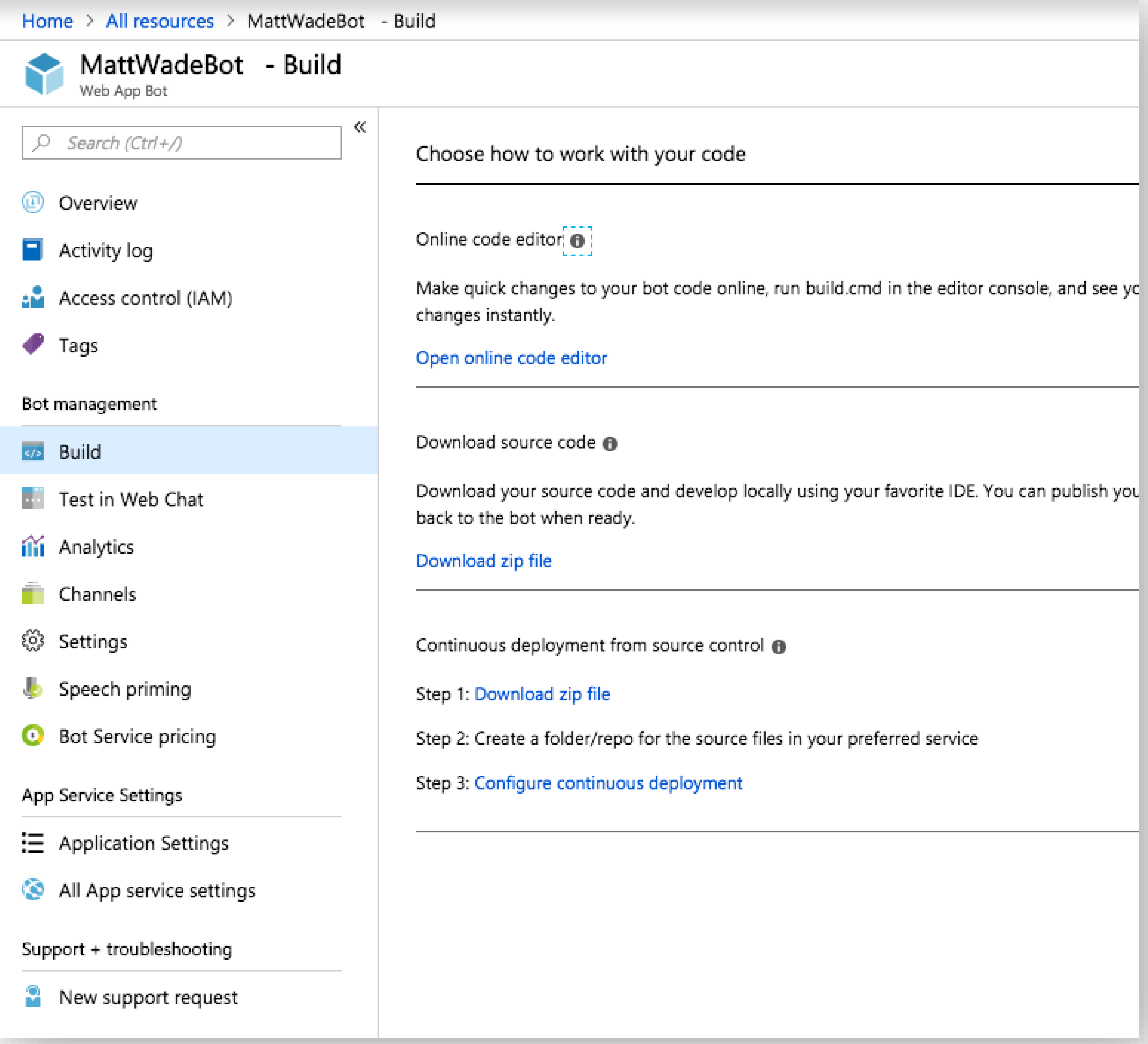
Requires an Azure subscription.
Register a free trial to get started.



Create your bot

Add your code in the web editor or download the ZIP to work in Visual Studio.

You'll do most work in MessageController.cs or RootDialogue.cs



Home > All resources > MattWadeBot - Build

MattWadeBot - Build
Web App Bot

Choose how to work with your code

Online code editor

Make quick changes to your bot code online, run build.cmd in the editor console, and see yo changes instantly.

Open online code editor

Download source code

Download your source code and develop locally using your favorite IDE. You can publish you back to the bot when ready.

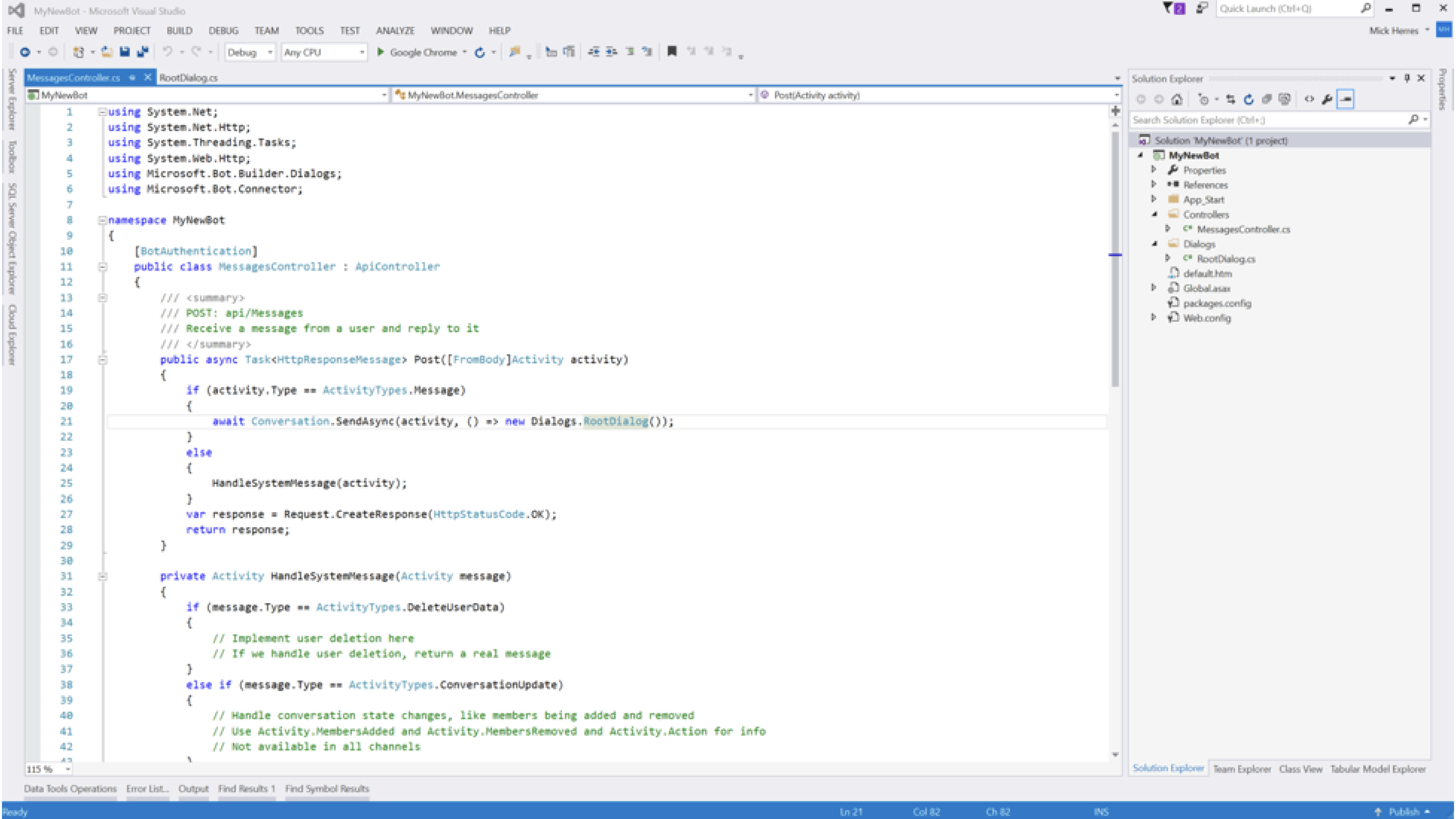
Download zip file

Continuous deployment from source control

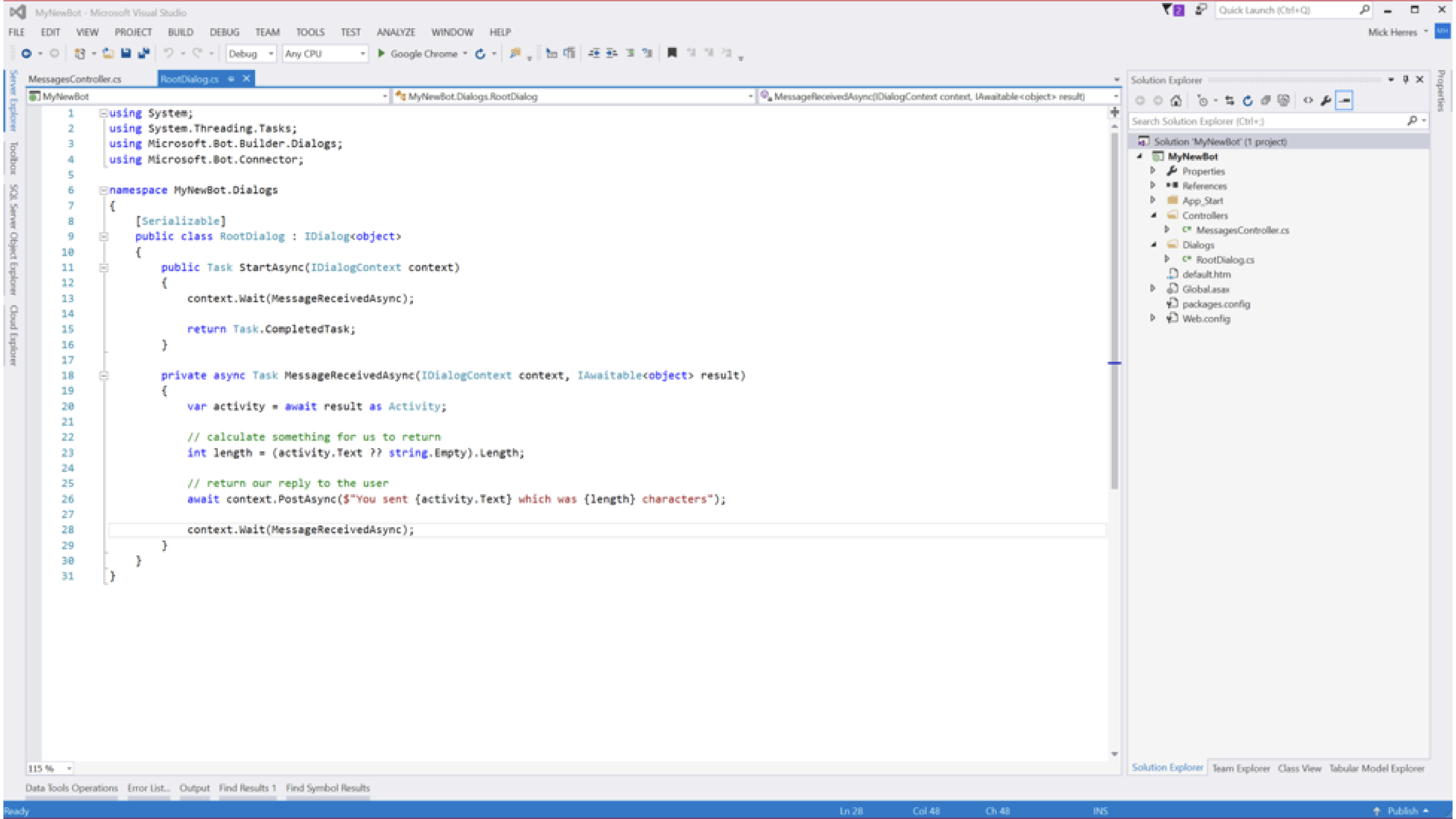
Step 1: Download zip file

Step 2: Create a folder/repo for the source files in your preferred service

Step 3: Configure continuous deployment



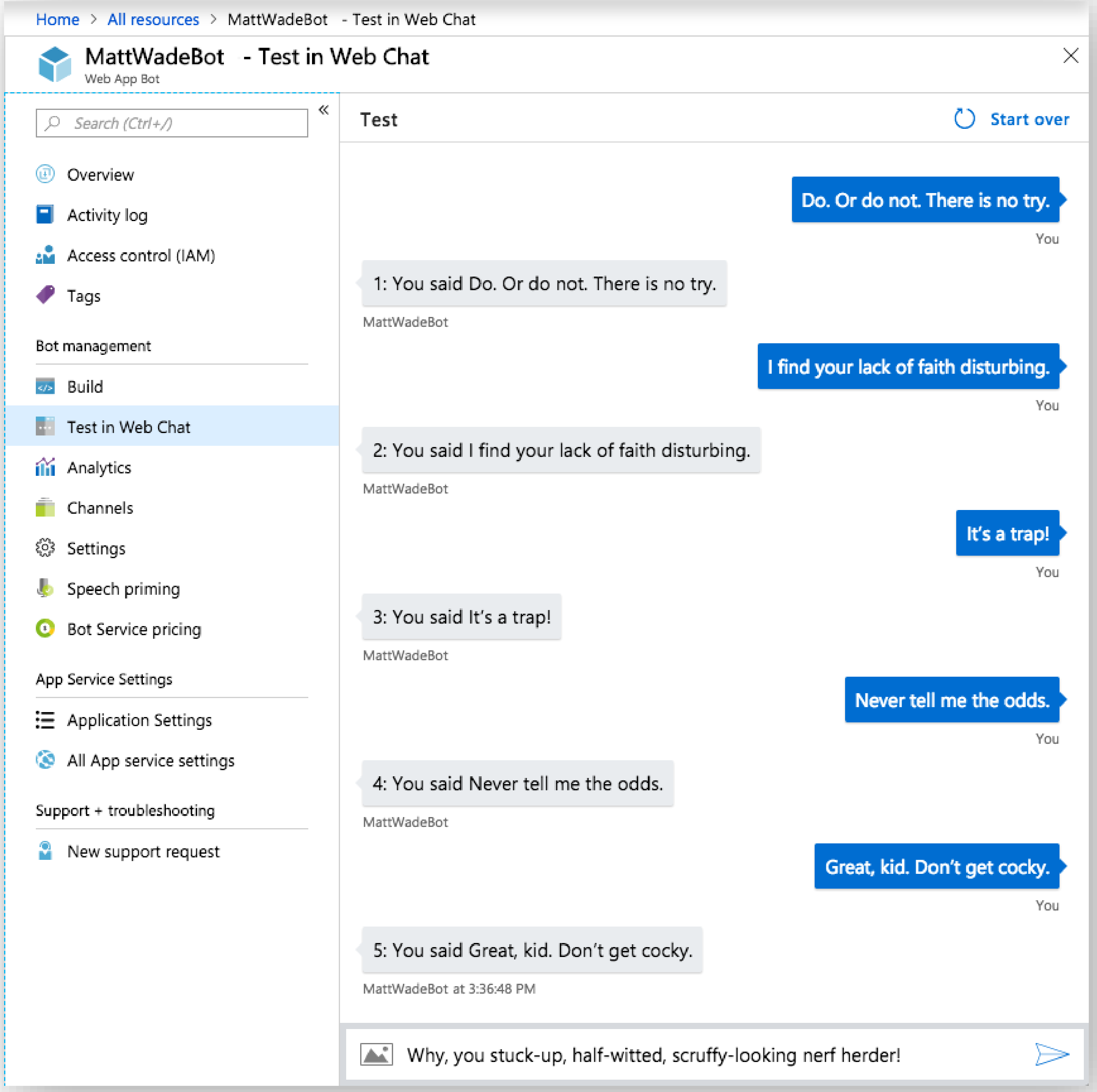
```
using System.Net;
using System.Net.Http;
using System.Threading.Tasks;
using System.Web.Http;
using Microsoft.Bot.Builder.Dialogs;
using Microsoft.Bot.Connector;

namespace MyNewBot
{
    [BotAuthentication]
    public class MessagesController : ApiController
    {
        /// <summary>
        /// POST: api/Messages
        /// Receive a message from a user and reply to it
        /// </summary>
        public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
        {
            if (activity.Type == ActivityTypes.Message)
            {
                await Conversation.SendAsync(activity, () => new Dialogs.RootDialog());
            }
            else
            {
                HandleSystemMessage(activity);
            }
            var response = Request.CreateResponse(HttpStatusCode.OK);
            return response;
        }

        private Activity HandleSystemMessage(Activity message)
        {
            if (message.Type == ActivityTypes.DeleteUserData)
            {
                // Implement user deletion here
                // If we handle user deletion, return a real message
            }
            else if (message.Type == ActivityTypes.ConversationUpdate)
            {
                // Handle conversation state changes, like members being added and removed
                // Use Activity.MembersAdded and Activity.MembersRemoved and Activity.Action for info
                // Not available in all channels
            }
```



```csharp
using System;
using System.Threading.Tasks;
using Microsoft.Bot.Builder.Dialogs;
using Microsoft.Bot.Connector;

namespace MyNewBot.Dialogs
{
    [Serializable]
    public class RootDialog : IDialog<object>
    {
        public Task StartAsync(IDialogContext context)
        {
            context.Wait(MessageReceivedAsync);

            return Task.CompletedTask;
        }

        private async Task MessageReceivedAsync(IDialogContext context, IAwaitable<object> result)
        {
            var activity = await result as Activity;

            // calculate something for us to return
            int length = (activity.Text ?? string.Empty).Length;

            // return our reply to the user
            await context.PostAsync($"You sent {activity.Text} which was {length} characters");

            context.Wait(MessageReceivedAsync);
        }
    }
}
```

Run tests in online or local emulator

Bot Dashboard > Test in Web Chat



Do. Or do not. There is no try.
You

1: You said Do. Or do not. There is no try.
MattWadeBot

I find your lack of faith disturbing.
You

2: You said I find your lack of faith disturbing.
MattWadeBot

It's a trap!
You

3: You said It's a trap!
MattWadeBot

Never tell me the odds.
You

4: You said Never tell me the odds.
MattWadeBot

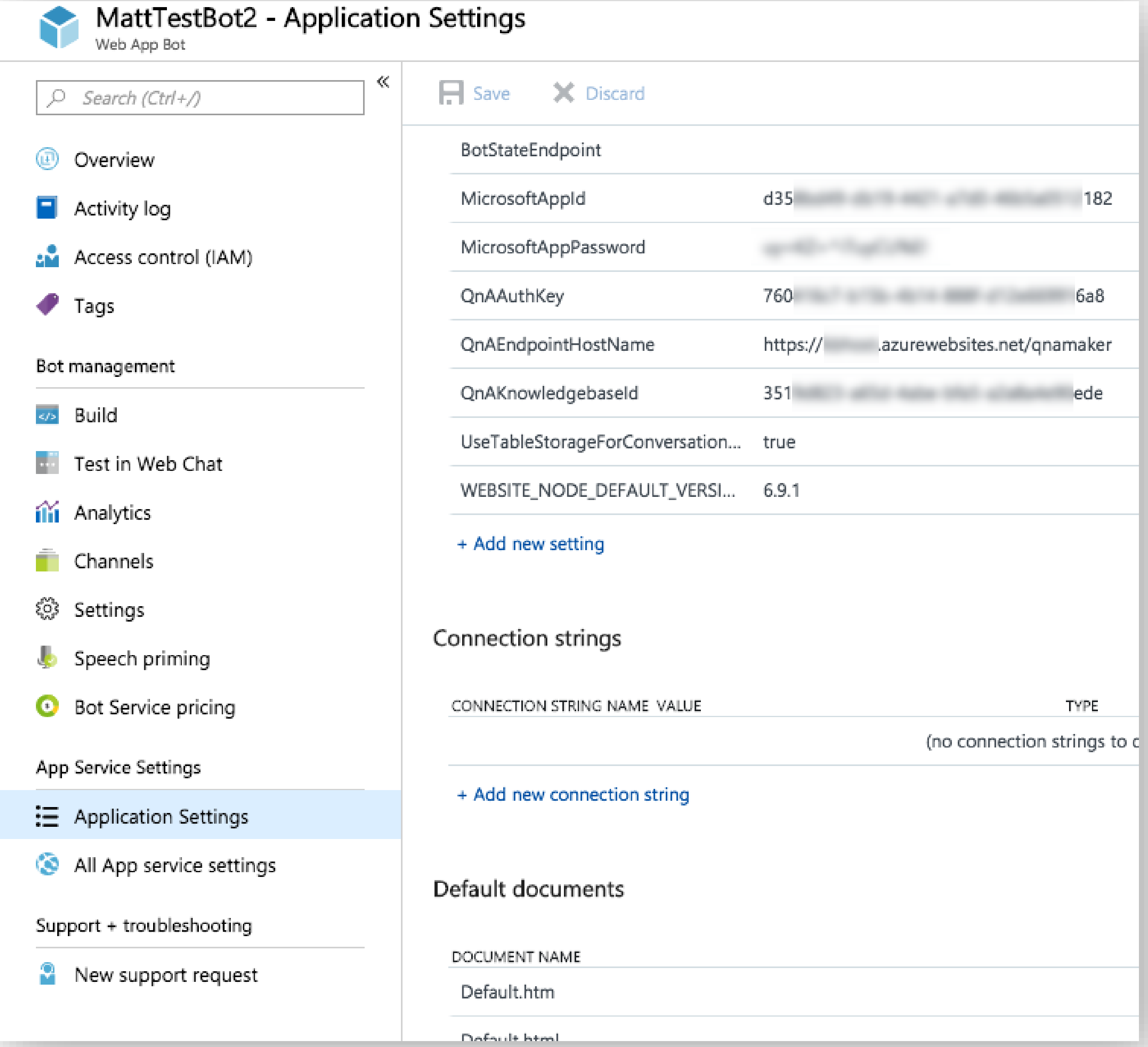
Great, kid. Don't get cocky.
You

5: You said Great, kid. Don't get cocky.
MattWadeBot at 3:36:48 PM

Why, you stuck-up, half-witted, scruffy-looking nerf herder!

Connect to QnA Maker

Instructions:
icsh.pt/CreateQnABot



BotStateEndpoint	
MicrosoftAppId	d35[redacted]182
MicrosoftAppPassword	[redacted]
QnAAuthKey	760[redacted]6a8
QnAEndpointHostName	https://[redacted].azurewebsites.net/qnamaker
QnAKnowledgebaseId	351[redacted]ede
UseTableStorageForConversation...	true
WEBSITE_NODE_DEFAULT_VERSI...	6.9.1

+ Add new setting

Connection strings

CONNECTION STRING NAME	VALUE	TYPE
	(no connection strings to d	

+ Add new connection string

Default documents

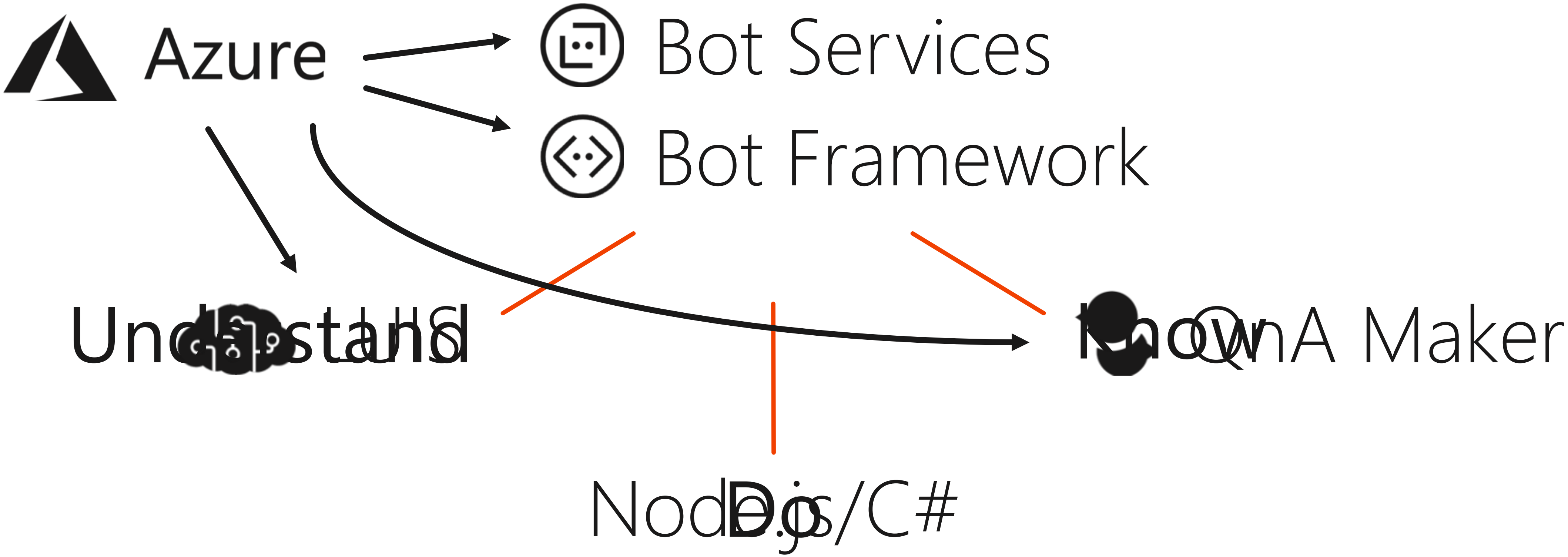
DOCUMENT NAME
Default.htm
Default.html

QnA Maker best practices:
tinyurl.com/AtBotQnABP

Let's build a bot

Best practices & architecture

A good bot stands on three legs



Azure

Bot Services

Bot Framework

Understand

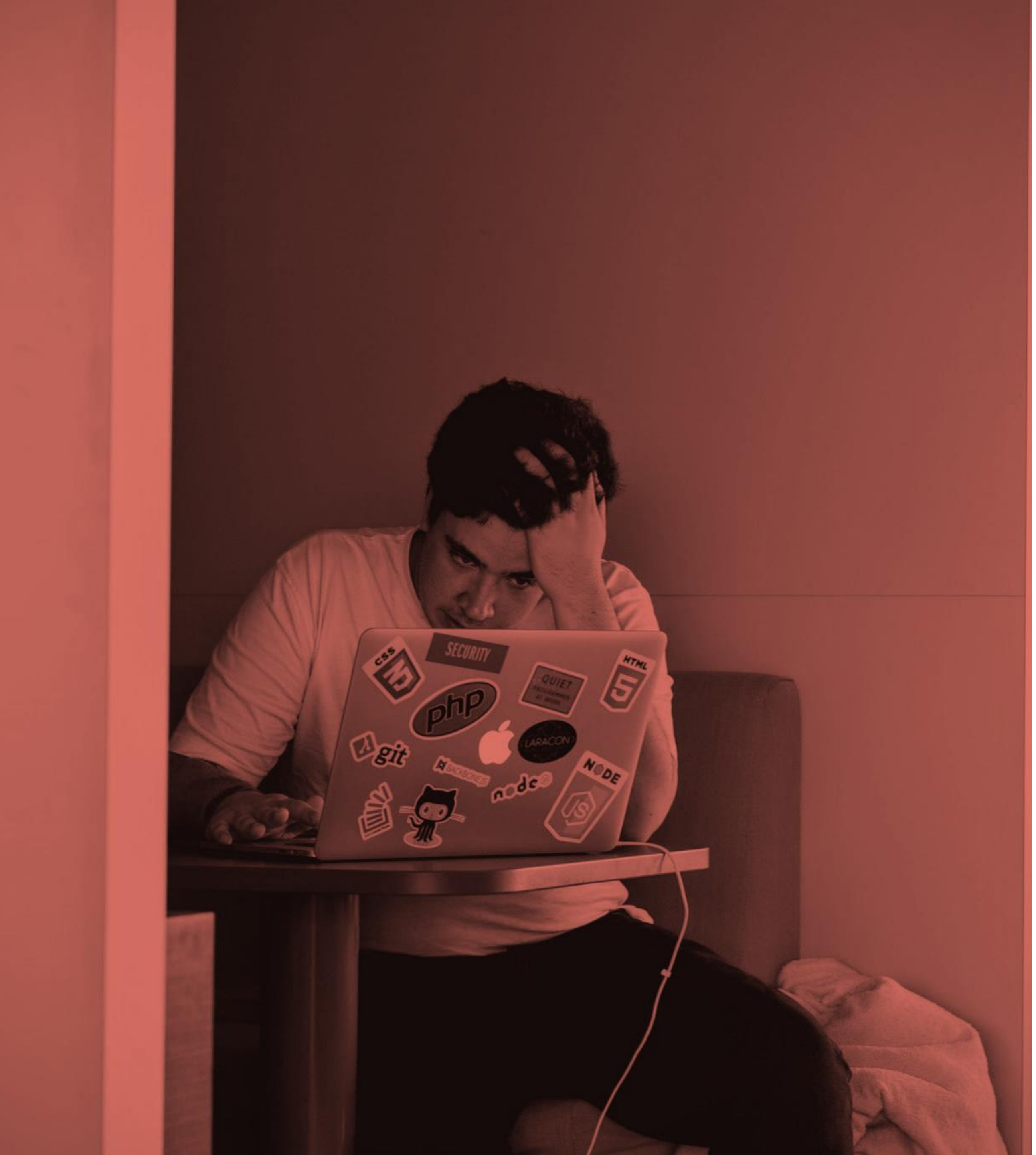
LUIS

Know

QnA Maker

Do

Node.js/C#



Don’t fall into the trap

A good FAQ bot is not good enough

#GlobalAIBootcamp

Surfacing your bot

Internal
(Authenticated)

Teams

SharePoint

Intranet

Custom
(Direct Line)

External
(Public)

Website

Social
Media

Custom
(Direct Line)

Demo

https://youtu.be/wAPcafOL_tg

Demo

https://youtu.be/EWXBYDQ4TVQ

Some resources

- Azure Bot Service Documentation: https://icsh.pt/BotServiceSDK4
- Create a QnA bot: https://icsh.pt/CreateQnABot
- Getting started with the Bot Framework (video): https://youtu.be/cumYtCVjl6Q
- Incorporating LUIS (Natural Language Processing): https://tinyurl.com/AtBotLUIS
- QnA Maker best practices: https://tinyurl.com/AtBotQnABP
- Getting started with code-free, Microsoft Flow bots: https://icsh.pt/NoCodeBots

Microsoft

#GlobalAIBootcamp