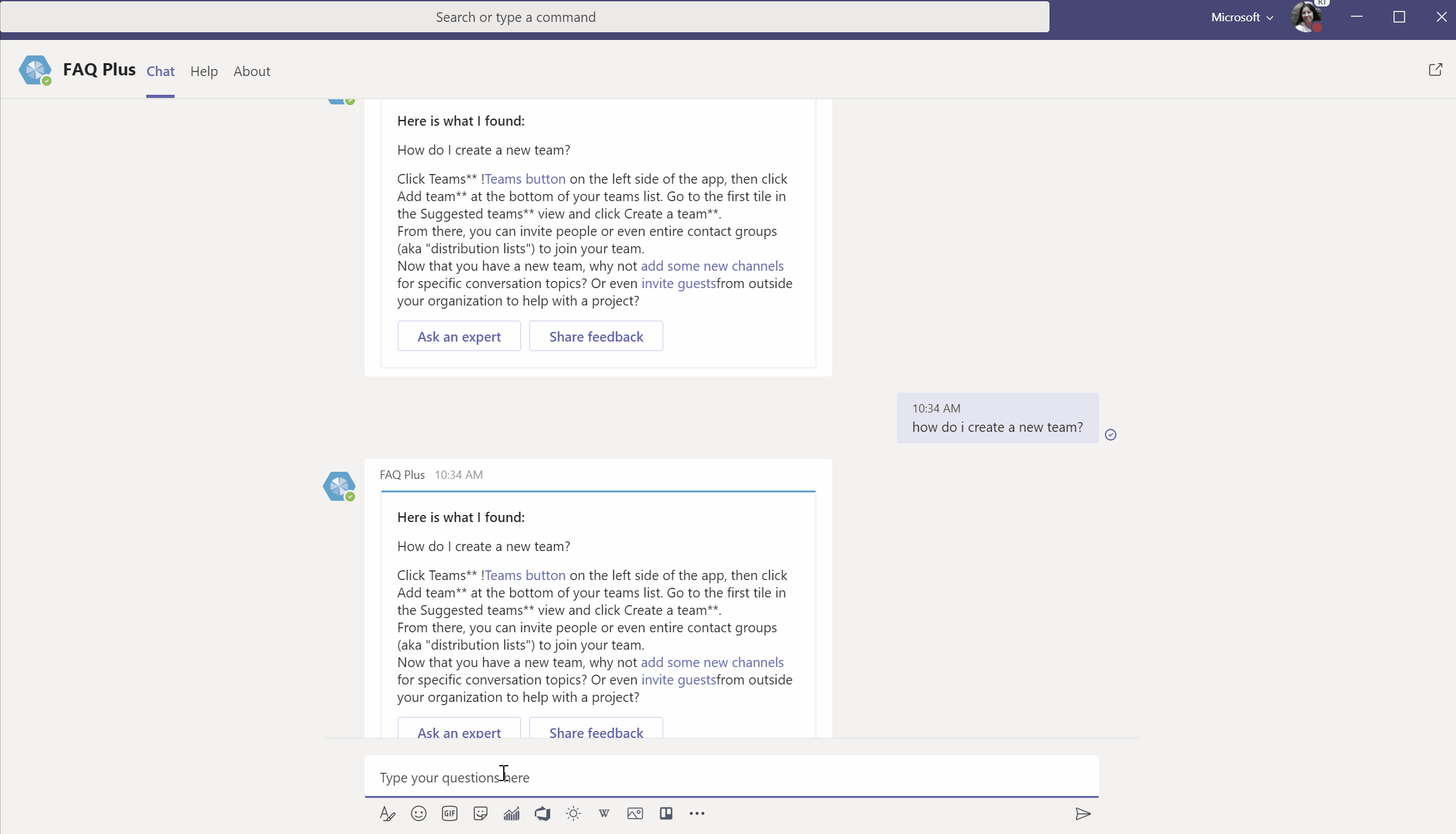
**FAQ Plus [Version 2] App Template**

Chatbots are an easy way to provide answers to frequently asked questions by users. However, most chatbots fail to engage with users in a meaningful way because there is no human in the loop when the chatbot fails.

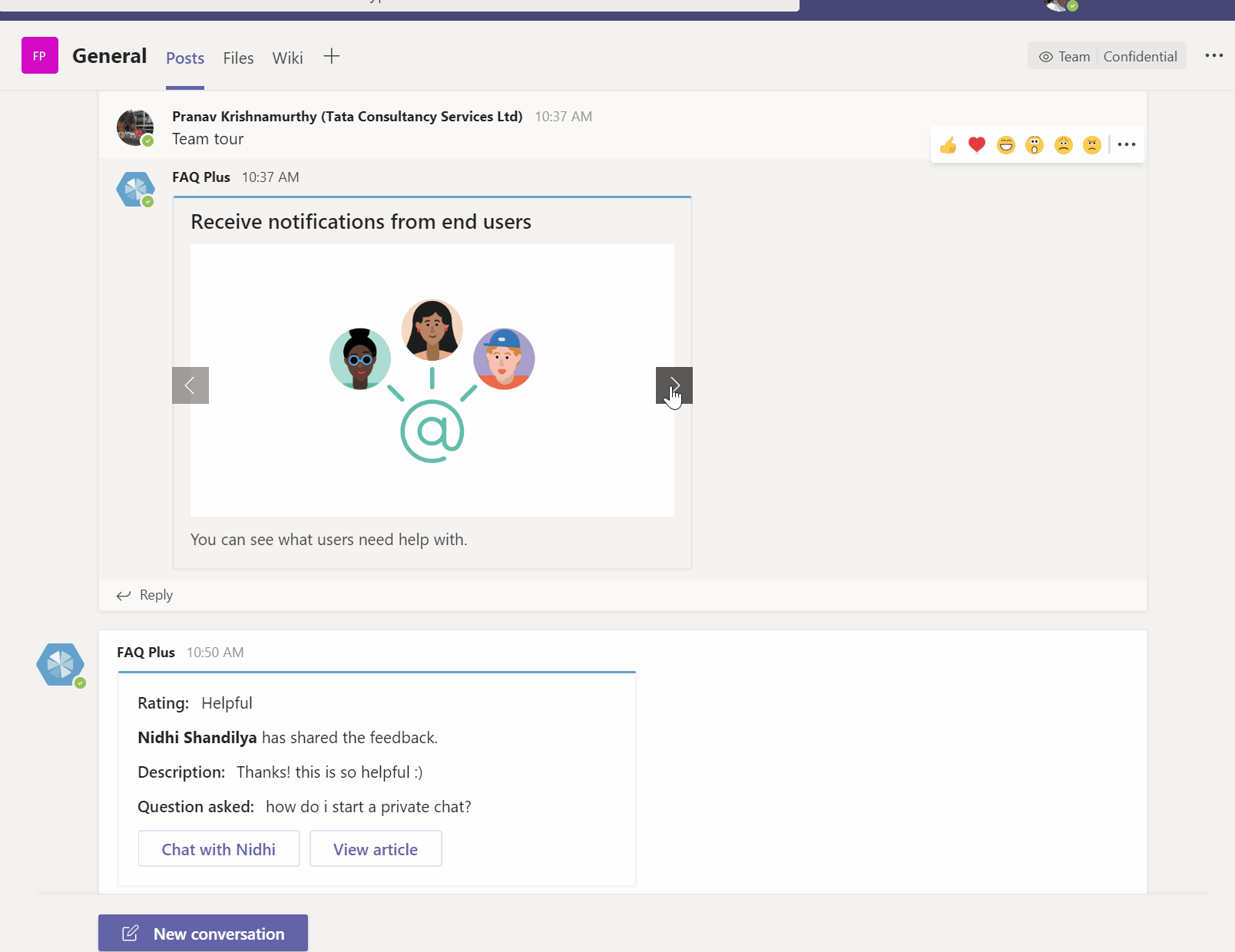
FAQ Plus bot is a friendly Q&A bot that brings a human in the loop when it is unable to help. One can ask the bot a question and the bot responds with an answer if it's in the knowledge base. If not, the bot allows the user to submit a query which then gets posted in a pre-configured team of experts who are help to provide support by acting upon the notifications from within their Team itself.

**FAQ Plus has following features:**

An end-user interacting with FAQ Plus:

[](https://github.com/OfficeDev/microsoft-teams-apps-faqplusv2/wiki/Images/FAQPlusEndUser.gif)

Experts team using FAQ Plus:

[](https://github.com/OfficeDev/microsoft-teams-apps-faqplusv2/wiki/Images/FAQPlusExperts.gif)

**FAQ Plus [Version 2] adds the following new capabilities**

Members of the experts team will now be able to

* add a new QnA pair to the knowledge base directly using messaging extension
* edit and delete QnA pairs added through the bot
* track history of experts updating QnA pair
* configure the answer to show up as an adaptive card with additional details like title, subtitle, image, etc. instead of usual text based answer

# Deployment Guide

## Prerequisites

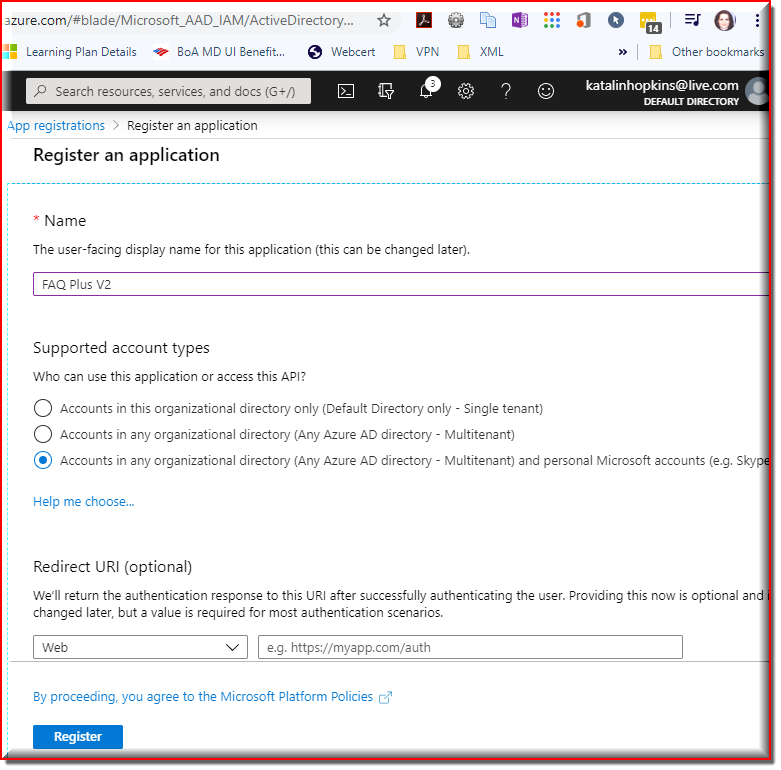
To begin, you will need:

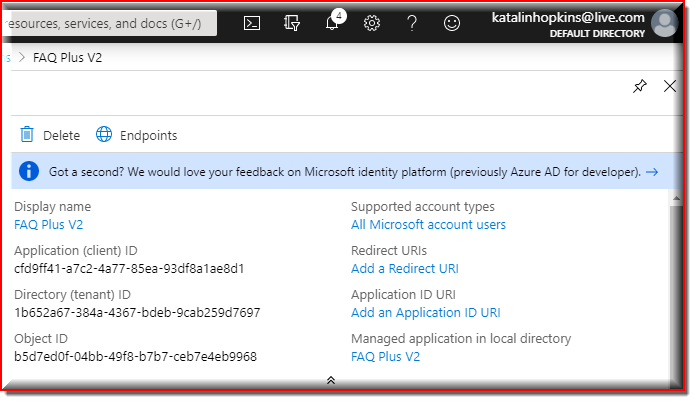
* An Azure subscription where you can create the following kind of resources:
  + App service
  + App service plan
  + Bot channels registration
  + Azure storage account
  + Azure search
  + Azure function
  + QnAMaker cognitive service
  + Application Insights
* A team in Microsoft Teams with your group of experts. (You can add and remove team members later!)
* A copy of the FAQ Plus [Version 2] app GitHub repo (<https://github.com/OfficeDev/microsoft-teams-apps-faqplusv2>)
* A reasonable set of Question and Answer pairs to set up the knowledge base for the bot.

## Step 1: Register Azure AD applications

Register two Azure AD applications in your tenant's directory: one for the bot, and another for the configuration app. Registering with the account/tenant: katalinhopkins@live.com

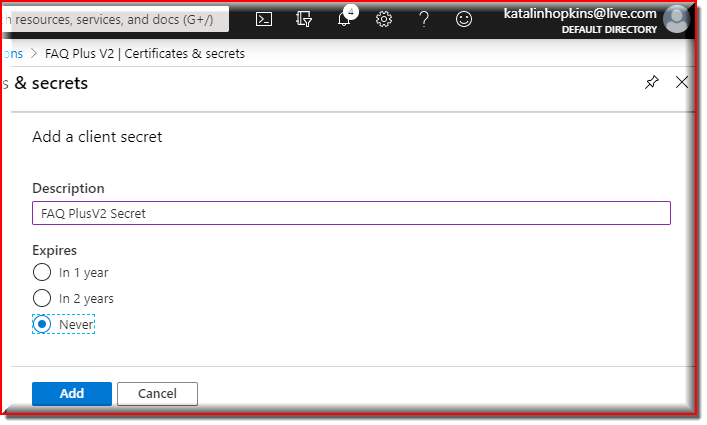
1. Log in to the Azure Portal for your subscription, and go to the "App registrations" blade [here](https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps).
2. Click on "New registration", and create an Azure AD application.
   1. **Name**: The name of your Teams app - if you are following the template for a default deployment, we recommend "FAQ PlusV2".
   2. **Supported account types**: Select "Accounts in any organizational directory"
   3. Leave the "Redirect URL" field blank.
3. Click on the "Register" button.





|  |  |
| --- | --- |
| Display name | FAQ Plus V2 |
| Application (client) ID | cfd9ff41-a7c2-4a77-85ea-93df8a1ae8d1 |
| Directory (tenant) ID | 1b652a67-384a-4367-bdeb-9cab259d7697 |
| Object ID | b5d7ed0f-04bb-49f8-b7b7-ceb7e4eb9968 |
| Supported account types | All Microsoft account users |
| Redirect URIs | Add a Redirect URI |
| Application ID URI | Add an Application ID URI |
| Managed application in local directory | FAQ Plus V2 |

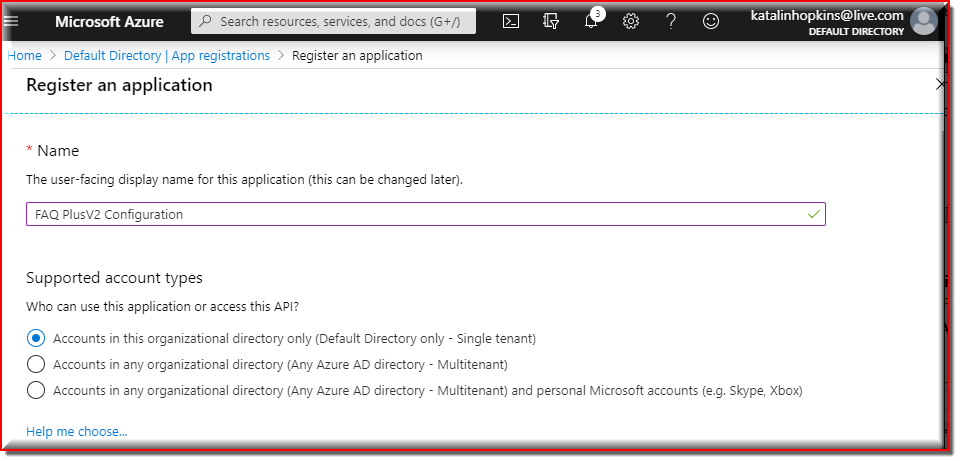
1. When the app is registered, you'll be taken to the app's "Overview" page. Copy the **Application (client) ID**; we will need it later. Verify that the "Supported account types" is set to **Multiple organizations**.
2. On the side rail in the Manage section, navigate to the "Certificates & secrets" section. In the Client secrets section, click on "+ New client secret". Add a description for the secret and select an expiry time. Click "Add".



1. Once the client secret is created, copy its **Value**; we will need it later.

|  |  |  |
| --- | --- | --- |
| FAQ PlusV2 Secret | 12/31/2299 | 2EgXO5E7ZIV:p3?:uRgOsbDc-L@b1Hs3 |

1. Go back to “App registrations”, then repeat steps 2-3 to create another Azure AD application for the configuration app.
   1. **Name**: The name of your configuration app. We advise appending “Configuration” to the name of this app; for example, “FAQ PlusV2 Configuration”.
   2. **Supported account types**: Select "Account in this organizational directory only"
   3. Leave the "Redirect URL" field blank for now.



|  |  |
| --- | --- |
| Display name | FAQ PlusV2 Configuration |
| Application (client) ID | 30d914bc-5f99-44ba-a11e-5bde1e162df6 |
| Directory (tenant) ID | 1b652a67-384a-4367-bdeb-9cab259d7697 |
| Object ID | a0eca793-d7d7-4657-bb90-37c981f7987b |
| Supported account types | My organization only |
| Redirect URIs | Add a Redirect URI |
| Application ID URI | Add an Application ID URI |
| Managed application in local directory | FAQ PlusV2 Configuration |

At this point you have 4 unique values:

* Application (client) ID for the bot
* Client secret for the bot
* Application (client) ID for the configuration app
* Directory (tenant) ID, which is the same for both apps

We recommend that you copy these values into a text file, using an application like Notepad. We will need these values later.

## Step 2: Deploy to your Azure subscription

1. Click on the "Deploy to Azure" button below.

[Deploy to Azure](https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FOfficeDev%2Fmicrosoft-teams-apps-faqplusv2%2Fmaster%2FDeployment%2Fazuredeploy.json)

1. When prompted, log in to your Azure subscription.
2. Azure will create a "Custom deployment" based on the ARM template and ask you to fill in the template parameters.
3. Select a subscription and resource group.
   * We recommend creating a new resource group. FAQPlusV2ResourceGroup
   * The resource group location MUST be in a data center that supports: Application Insights; Azure Search; and QnA Maker. For an up-to-date list, click [here] (<https://azure.microsoft.com/en-us/global-infrastructure/services/?products=logic-apps,cognitive-services,search,monitor>), and select a region where the following services are available:

Application Insights

QnA Maker

Azure Search

1. Enter a "Base Resource Name", which the template uses to generate names for the other resources. ISVChecklistFaqBot
   * The app service names [Base Resource Name], [Base Resource Name]-config, and [Base Resource Name]-qnamaker must be available. For example, if you select contosofaqplus as the base name, the names contosofaqplus, contosofaqplus-config, and contosofaqplus-qnamaker must be available (not taken); otherwise, the deployment will fail with a Conflict error.
   * Remember the base resource name that you selected. We will need it later.
2. Fill in the various IDs in the template:
   * **Bot Client ID**: The application (cli ent) ID of the Microsoft Teams Bot app
   * **Bot Client Secret**: The client secret of the Microsoft Teams Bot app
   * **Config App Client Id**: The application (client) ID of the configuration app
   * **Tenant Id**: The tenant ID above

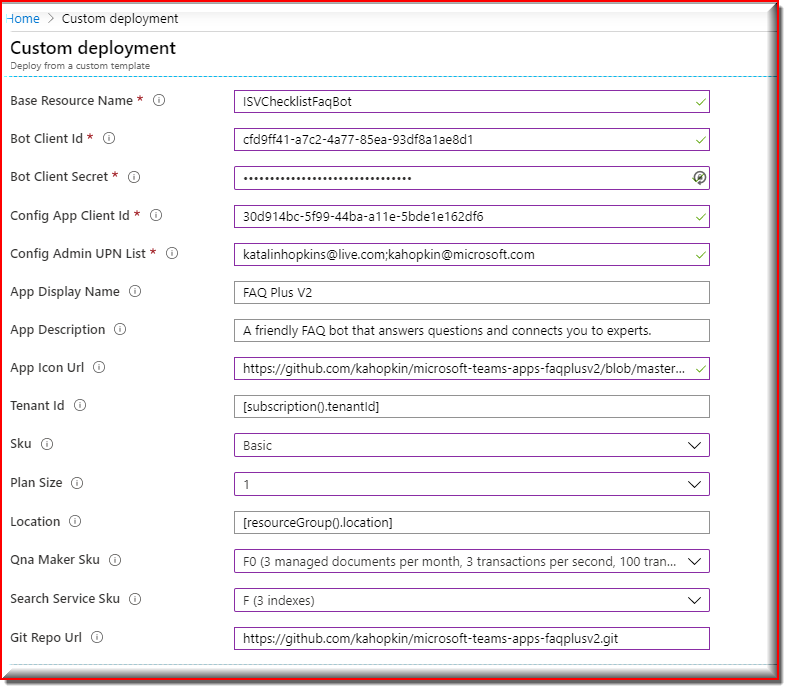
Make sure that the values are copied as-is, with no extra spaces. The template checks that GUIDs are exactly 36 characters.

1. Fill in the "Config Admin UPN List", which is a semicolon-delimited list of users who will be allowed to access the configuration app. \* For example, to allow Megan Bowen ([meganb@contoso.com](mailto:meganb@contoso.com)) and Adele Vance ([adelev@contoso.com](mailto:adelev@contoso.com)) to access the configuration app, set this parameter to meganb@contoso.com;adelv@contoso.com. \* You can change this list later by going to the configuration app service's "Configuration" blade.
2. If you wish to change the app name, description, and icon from the defaults, modify the corresponding template parameters.
3. Agree to the Azure terms and conditions by clicking on the check box "I agree to the terms and conditions stated above" located at the bottom of the page.
4. Click on "Purchase" to start the deployment.
5. Wait for the deployment to finish. You can check the progress of the deployment from the "Notifications" pane of the Azure Portal. It can take more than 10 minutes for the deployment to finish.
6. Once the deployment has finished, you would be directed to a page that has the following fields:

botId - This is the Microsoft Application ID for the FAQ Plus [Version 2] bot.

appDomain - This is the base domain for the FAQ Plus [Version 2] Bot.

configurationAppUrl - This is the URL for the configuration web application.



Customized template

**11 resources**

BASICS

SETTINGS

FAQPlusVer2ResourceGroup

11 resources

BASICS

|  |  |
| --- | --- |
| Subscription | Visual Studio Enterprise |
| Resource group | (New) FAQPlusV2ResourceGroup |
| Location | (US) West US |

SETTINGS

|  |  |
| --- | --- |
| Base Resource Name | ISVChecklistFaqBot |
| Bot Client Id | cfd9ff41-a7c2-4a77-85ea-93df8a1ae8d1 |
| Bot Client Secret | 2EgXO5E7ZIV:p3?:uRgOsbDc-L@b1Hs3 |
| Config App Client Id | 30d914bc-5f99-44ba-a11e-5bde1e162df6 |
| Config Admin UPN List | katalinhopkins@live.com;kahopkin@microsoft.com;admin@katalinhopkinslive.onmicrosoft.com |
| App Display Name | FAQ Plus V2 |
| App Description | A friendly FAQ bot that answers questions and connects you to experts. |
| App Icon Url | <https://github.com/kahopkin/microsoft-teams-apps-faqplusv2/blob/master/Manifest/color.png> |
| Tenant Id | [subscription().tenantId] |
| Sku | Basic |
| Plan Size | 1 |
| Location | [resourceGroup().location] |
| Qna Maker Sku | F0 (3 managed documents per month, 3 transactions per second, 100 transactions per minute, 50K transactions per month) |
| Search Service Sku | F (3 indexes) |
| Git Repo Url | <https://github.com/kahopkin/microsoft-teams-apps-faqplusv2.git> |
| Git Branch | master |

Output:

|  |  |
| --- | --- |
| botId | 1dbe1a35-6dd6-4e83-a640-406baeacd75f |
| appDomain | FAQPlusVer2.azurewebsites.net |
| configurationAppUrl | https://FAQPlusVer2-config.azurewebsites.net |

<https://github.com/OfficeDev/microsoft-teams-apps-faqplusv2.wiki.git>

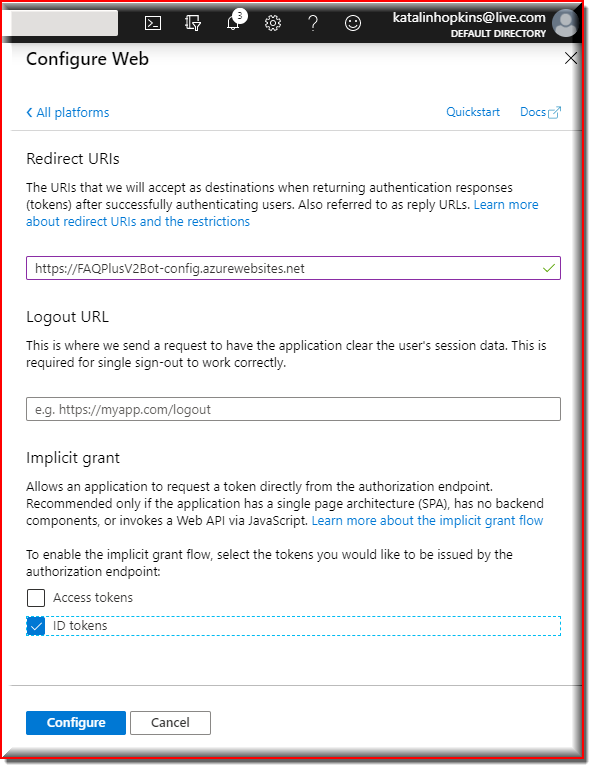
https://github.com/OfficeDev/microsoft-teams-apps-faqplusv2.wiki.git

## Step 3: Set up authentication for the configuration app

1. Note the location of the configuration app that you deployed, which is https://[BaseResourceName]-config.azurewebsites.net. For example, if you chose "contosofaqplus" as the base name, the configuration app will be at https://contosofaqplus-config.azurewebsites.net
2. Go back to the "App Registrations" page [here](https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredAppsPreview).
3. Click on the configuration app in the application list. Under "Manage", click on "Authentication" to bring up authentication settings.
4. Add a new entry to "Redirect URLs":
   * **Type**: Web
   * **Redirect URL**: Location (URL) of your configuration app.

<https://FAQPlusVer2-config.azurewebsites.net>

* + Under "Implicit grant", check "**ID tokens**".
  + Click "Save" to commit your changes.



## Step 4: Create the QnA Maker knowledge base

Create a knowledge base, following the instructions in the [QnA Maker documentation](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/tutorials/create-publish-query-in-portal#create-a-knowledge-base).

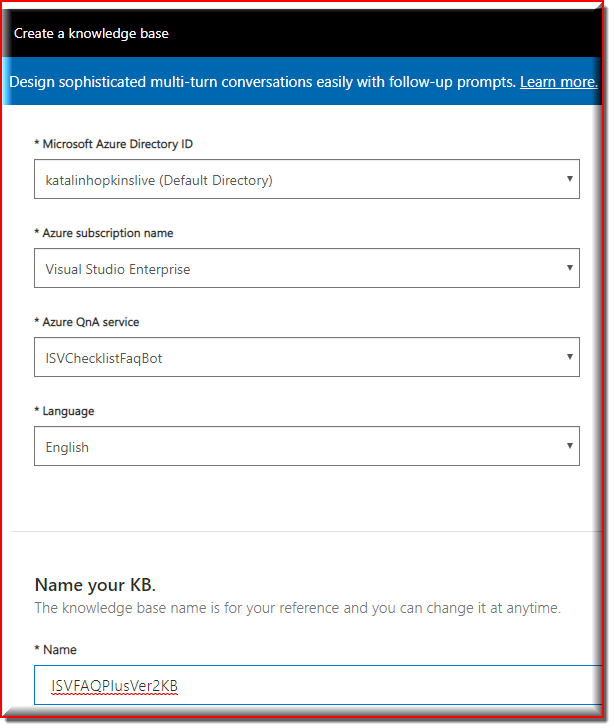
Skip the step, "Create a QnA service in Microsoft Azure", because the ARM template that you deployed in Step 2 "Deploy to your Azure subscription" already created the QnA service. Proceed directly to the next step, "Connect your QnA service to your KB".

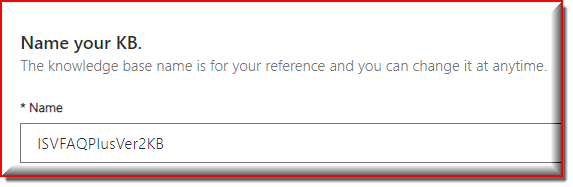
Use the following values when connecting to the QnA service:

* **Microsoft Azure Directory ID**: The tenant associated with the Azure subscription selected in Step 2.1.
* **Azure subscription name**: The Azure subscription to which the ARM template was deployed.
* **Azure QnA service**: The QnA service created during the deployment. This is the same as the "Base resource name"; for example, if you chose "contosofaqplus" as the base name, the QnA Maker service will be named contosofaqplus.

**Note:** Kindly add at least one QnA pair, before publishing the knowledge base otherwise it will throw error.

ISVFAQPlusVer2KB





<https://gearup.microsoft.com/checklists/isv>

Phases:

Discover

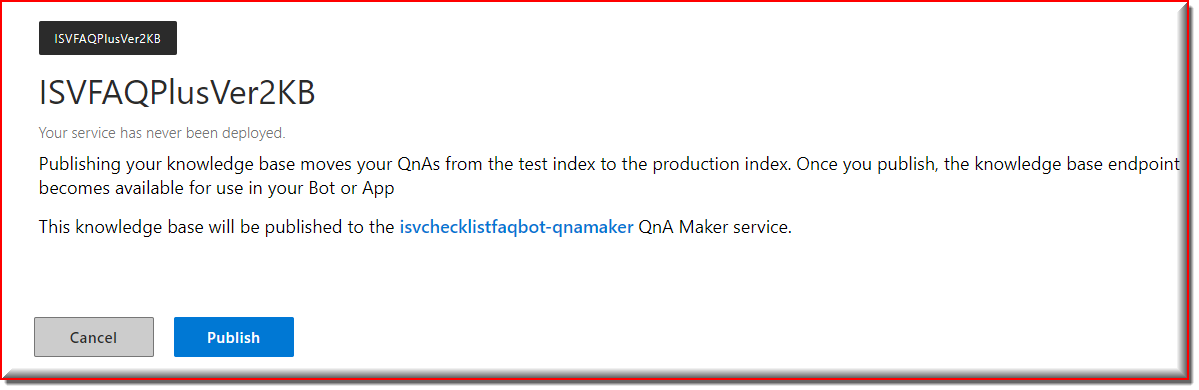
Envision

Validate

Commit

Onboard

Realize Value



After [publishing the knowledge base](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/tutorials/create-publish-query-in-portal#publish-to-get-knowledge-base-endpoints), note the knowledge base ID (see screenshot).

POST /knowledgebases/08a39ff8-a59a-44e8-bbca-aa16f34f030d/generateAnswer

Host: https://isvchecklistfaqbot-qnamaker.azurewebsites.net/qnamaker

Authorization: EndpointKey 28cada9c-4547-444f-a7c0-0adf1186be67

Content-Type: application/json

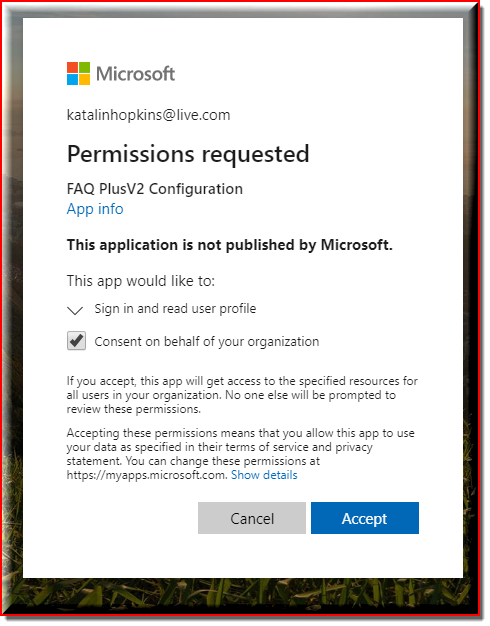
{"question":"<Your question>"}

## Step 5: Finish configuring the FAQ Plus [Version 2] app

1. Go to the configuration app, which is at https://[BaseResourceName]-config.azurewebsites.net. For example, if you chose “contosofaqplus” as the base name, the configuration app will be at <https://contosofaqplus-config.azurewebsites.net>.

<https://FAQPlusV2Bot-config.azurewebsites.net>

Sign is as Current User



No access…hmm…\

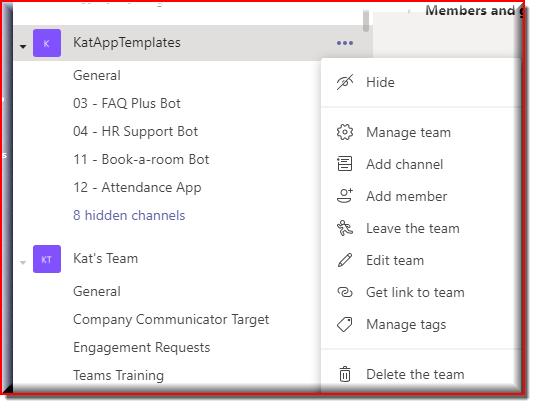
1. You will be prompted to login with your credentials. Make sure that you log in with an account that is in the list of users allowed to access the configuration app.

admin@katalinhopkinslive.onmicrosoft.com

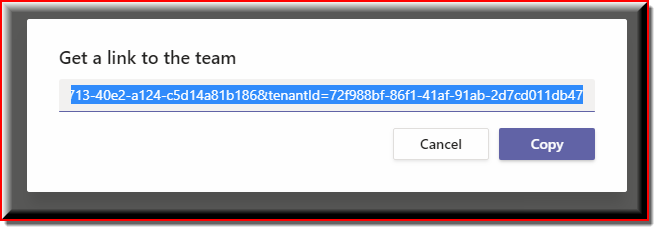
1. Get the link to the team with your **experts** from the Teams client. To do so, open Microsoft Teams, and navigate to the team. Click on the "..." next to the team name, then select "Get link to team".

Team: KatAppTemplates:

<https://teams.microsoft.com/l/team/19%3a491fe55b827f40938e38c9146f891e27%40thread.tacv2/conversations?groupId=9c98da3f-1713-40e2-a124-c5d14a81b186&tenantId=72f988bf-86f1-41af-91ab-2d7cd011db47>



Click on "Copy" to copy the link to the clipboard.



4. Paste the copied link into the "Team Id" field, then press "OK".

1. Enter the QnA Maker knowledge base ID into the "Knowledge base ID" field, then press "OK".

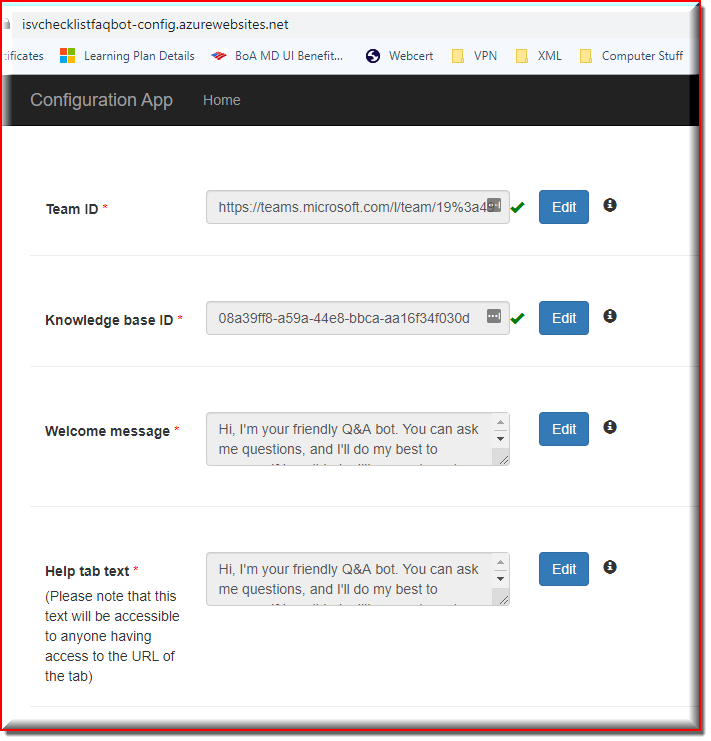
08a39ff8-a59a-44e8-bbca-aa16f34f030d

1. Customize the "Welcome message" that's sent to end users when they install the app. This message supports basic markdown, such as bold, italics, bullet lists, numbered lists, and hyperlinks. See [here](https://docs.microsoft.com/en-us/adaptive-cards/authoring-cards/text-features#markdown) for complete details on what Markdown features are supported.

|  |  |
| --- | --- |
| Team ID \* | <https://teams.microsoft.com/l/team/19%3a491fe55b827f40938e38c9146f891e27%40thread.tacv2/conversations?groupId=9c98da3f-1713-40e2-a124-c5d14a81b186&tenantId=72f988bf-86f1-41af-91ab-2d7cd011db47> |
| Knowledge base ID \* | d5a4788c-03c1-48c2-bf36-1b52ebe7ef3e |

Notes

Remember to click on "OK" after changing a setting. To edit the setting later, click on "Edit" to make the text box editable.



## Step 6: Create the Teams app packages

Create two Teams app packages: one for end-users to install personally, and one to be installed to the experts team.

1. Open the Manifest\manifest\_enduser.json file in a text editor.
2. Change the placeholder fields in the manifest to values appropriate for your organization.

"developer": {

"name": "Kat Hopkins",

"websiteUrl": "<https://kahopkin.sharepoint.com/>",

"privacyUrl": "<https://kahopkin.sharepoint.com/SitePages/PrivacyStatement.aspx>",

"termsOfUseUrl": "<https://kahopkin.sharepoint.com/SitePages/ServicesAgreement.aspx>"

},

FAQPlusV2Base.azurewebsites.net

1. Change the <<botId>> placeholder to your Azure AD application's ID from above. This is the same GUID that you entered in the template under "Bot Client ID".
2. In the "validDomains" section, replace the <<appDomain>> with your Bot App Service's domain. This will be [BaseResourceName].azurewebsites.net. For example if you chose "contosofaqplus" as the base name, change the placeholder to contosofaqplus.azurewebsites.net.
3. Save and Rename manifest\_enduser.json file to a file named manifest.json.
4. Create a ZIP package with the manifest.json,color.png, and outline.png. The two image files are the icons for your app in Teams.

* Name this package faqplusv2-enduser.zip, so you know that this is the app for end-users.
* Make sure that the 3 files are the *top level* of the ZIP package, with no nested folders.

"developer": {

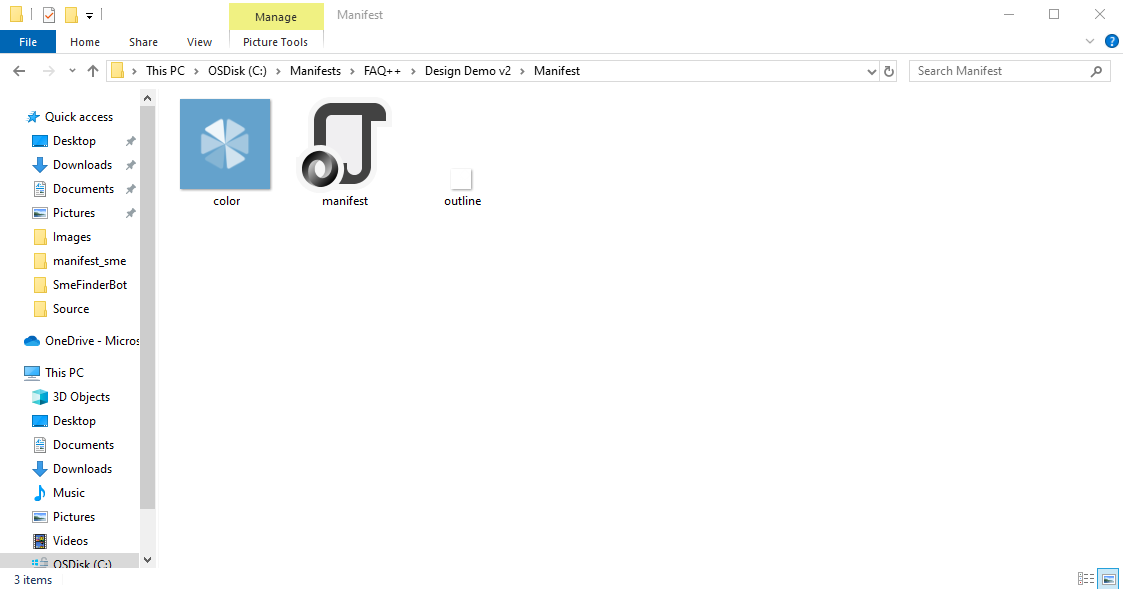
"name": "<company name>",

"websiteUrl": "<website url>",

"privacyUrl": "<privacy url>",

"termsOfUseUrl": "<terms of use url>"

},



1. Delete the manifest.json file.

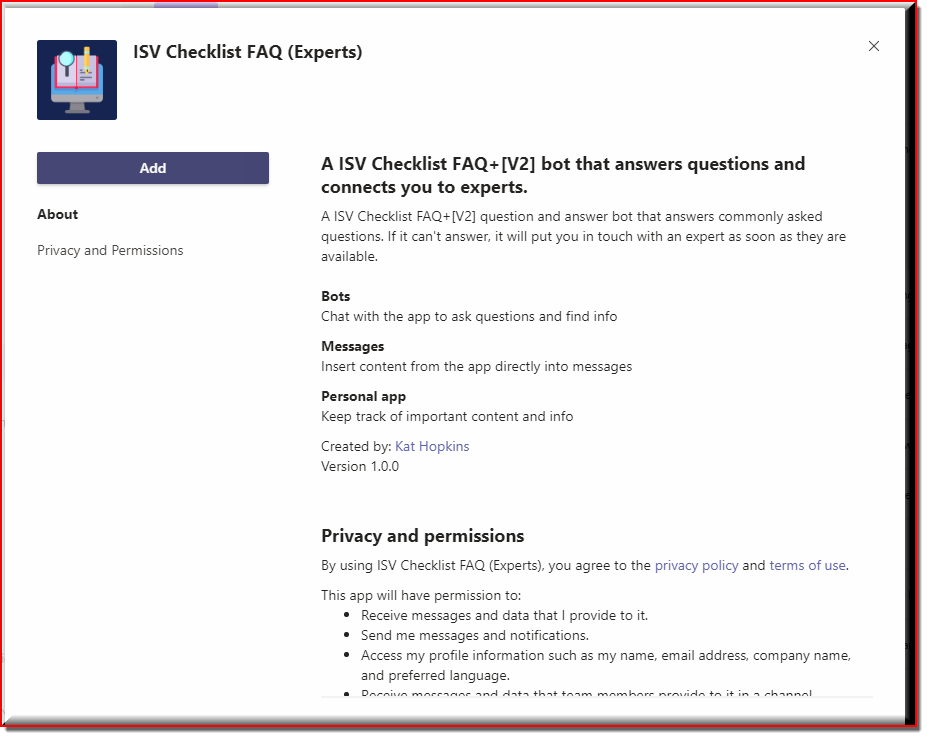
Repeat the steps above but with the file Manifest\manifest\_sme.json. Name the resulting package faqplusv2-experts.zip, so you know that this is the app for experts.

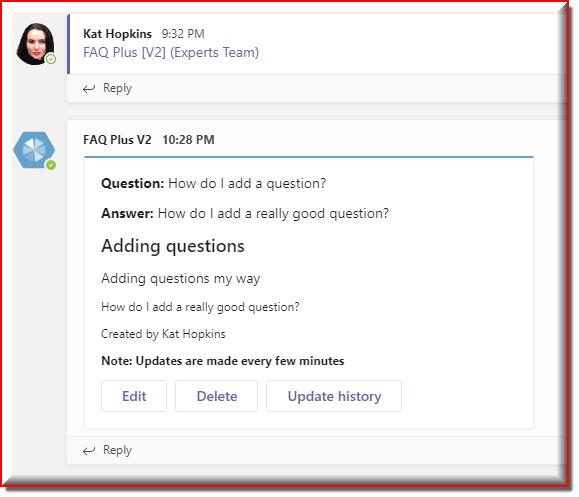
## Step 7: Run the apps in Microsoft Teams

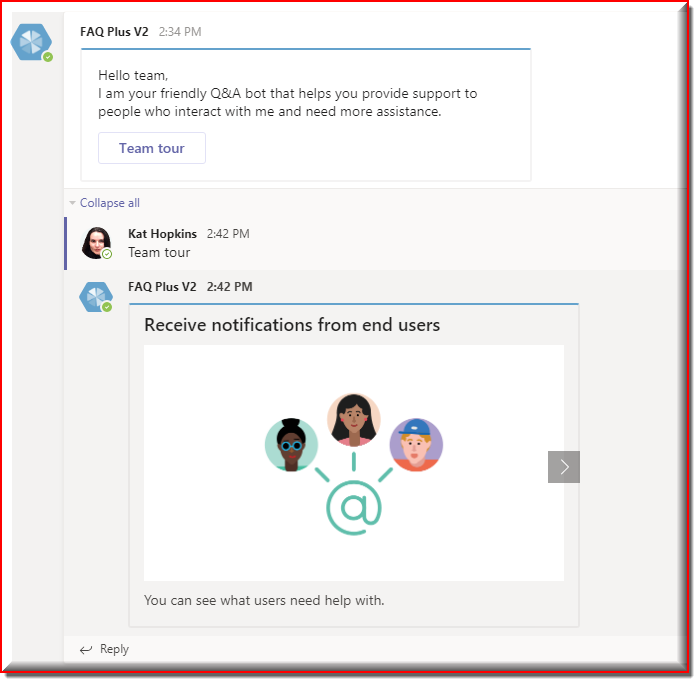
1. If your tenant has side-loading apps enabled, you can install your app by following the instructions [here](https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/apps/apps-upload#load-your-package-into-teams)
2. You can also upload it to your tenant's app catalog, so that it can be available for everyone in your tenant to install. See [here](https://docs.microsoft.com/en-us/microsoftteams/tenant-apps-catalog-teams)
3. Install the experts app (the faqplus-experts.zip package) to your team of subject-matter experts. This **MUST** be the same team that you selected in Step 5.3 above.

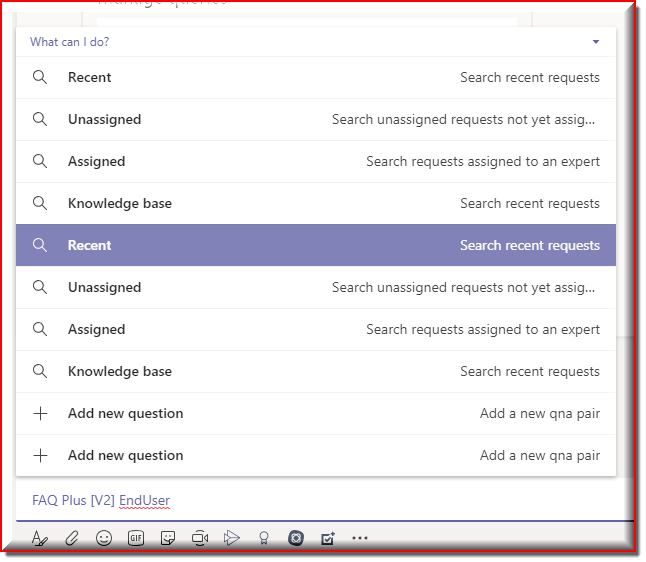
* We recommend using [app permission policies](https://docs.microsoft.com/en-us/microsoftteams/teams-app-permission-policies) to restrict access to this app to the members of the experts team.

1. Install the end-user app (the faqplus-enduser.zip package) to your users.

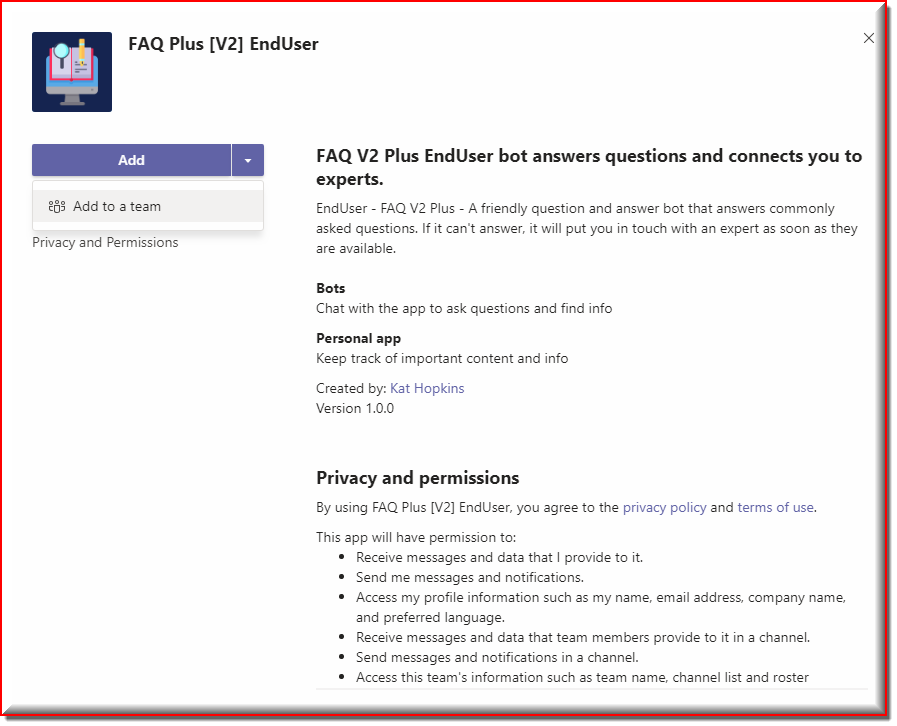


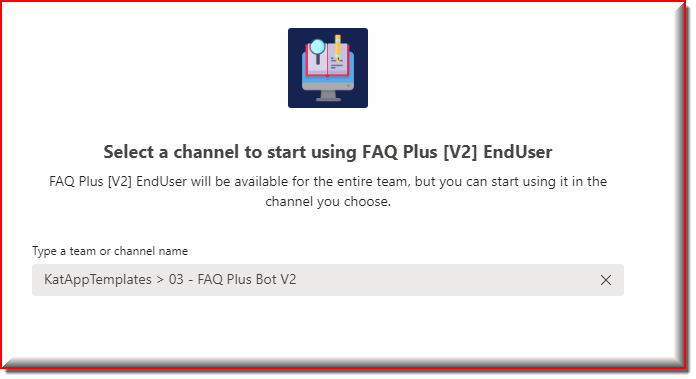




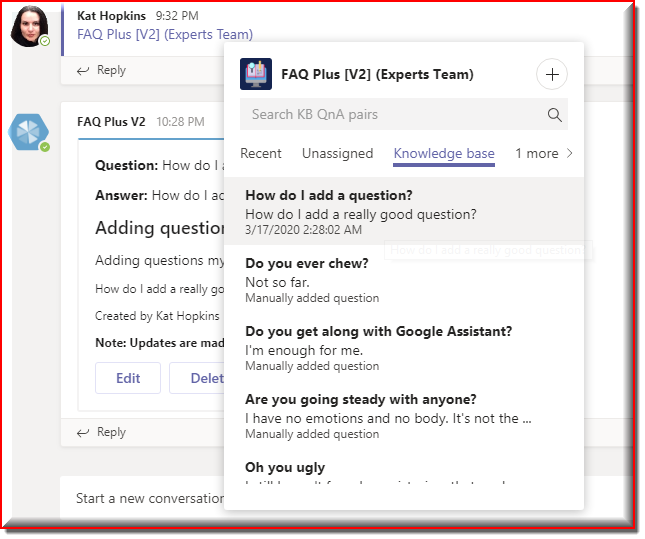


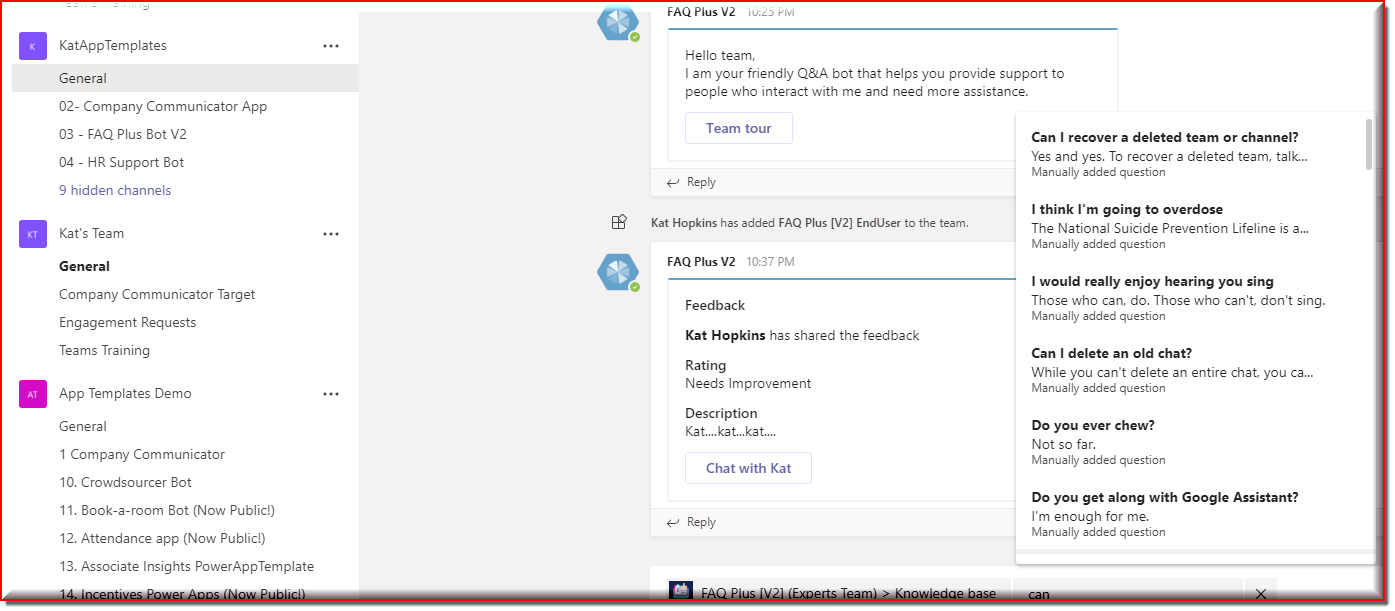
EndUsers:

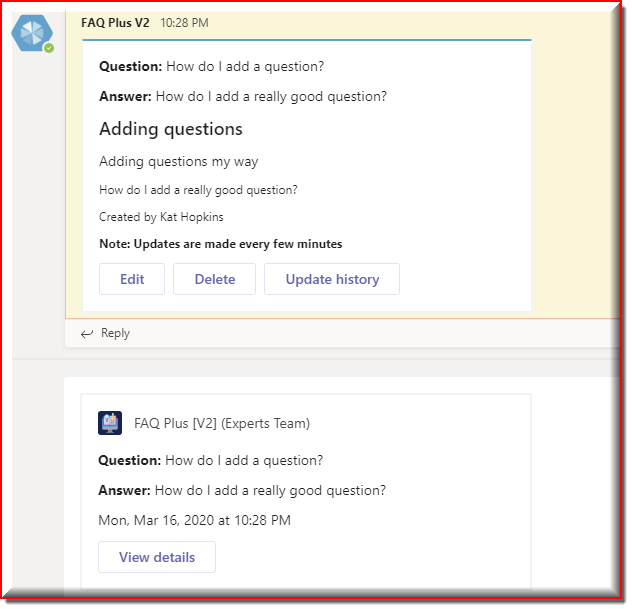


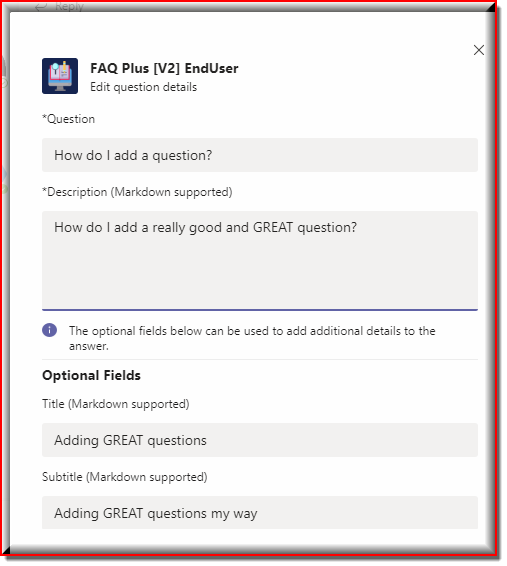


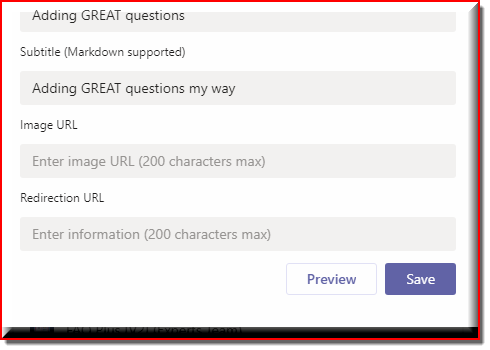
Click Set Up Bot











**Troubleshooting**

Please see our [Troubleshooting](https://github.com/OfficeDev/microsoft-teams-apps-faqplusv2/wiki/Troubleshooting.md) page.

# General template issues

### \*\*Generic possible issues\*\*

There are certain issues that can arise that are common to many of the app templates. Please check [here](https://github.com/OfficeDev/microsoft-teams-stickers-app/wiki/Troubleshooting) for reference to these.

### \*\*Problems deploying to Azure\*\*

### \*\*1. Error when attempting to reuse a Microsoft Azure AD application ID for the bot registration\*\*

#### Description

Bot is not valid.

```

Errors: MsaAppId is already in use.

```

Creating the resource of type Microsoft.BotService/botServices failed with status "BadRequest"

This happens when the Microsoft Azure application ID entered during the setup of the deployment has already been used and registered for a bot.

#### Fix

Either register a new Microsoft Azure AD application or delete the bot registration that is currently using the attempted Microsoft Azure application ID.

### \*\*2. Bot is unable to create more KBs and store additional questions\*\*

#### Description

Bot will reply to the user post with the error message if it finds that it cannot store any additional QnA pair to the Knowledge base

```

Errors: I cannot save this qna pair due to storage space limitations. Please contact your system administrator to provision additional storage space.

```

#### Fix

In case of such a scenario, system administrator or the app installer will need to update the pricing tier accordingly for QnA service in Azure Portal.

### \*\*3. Error while deploying the ARM Template\*\*

#### Description

This happens when the resources are already created or due to some conflicts.

```

Errors: The resource operation completed with terminal provisioning state 'Failed'

```

#### Fix

In case of such a scenario, user needs to navigate to deployment center section of failed/conflict resources through the azure portal and check the error logs to get the actual errors and fix it accordingly.

Redeploy it after fixing the issue/conflict.

### \*\*Problems related to Messaging Extension\*\*

### \*\*1. Experts team members unable to see data in "Knowledge base" tab\*\*

#### Description

This happens when the admin user does not update the knowledge base id in the configuration app or if there are no QnA pairs in the knowledge base.

#### Fix

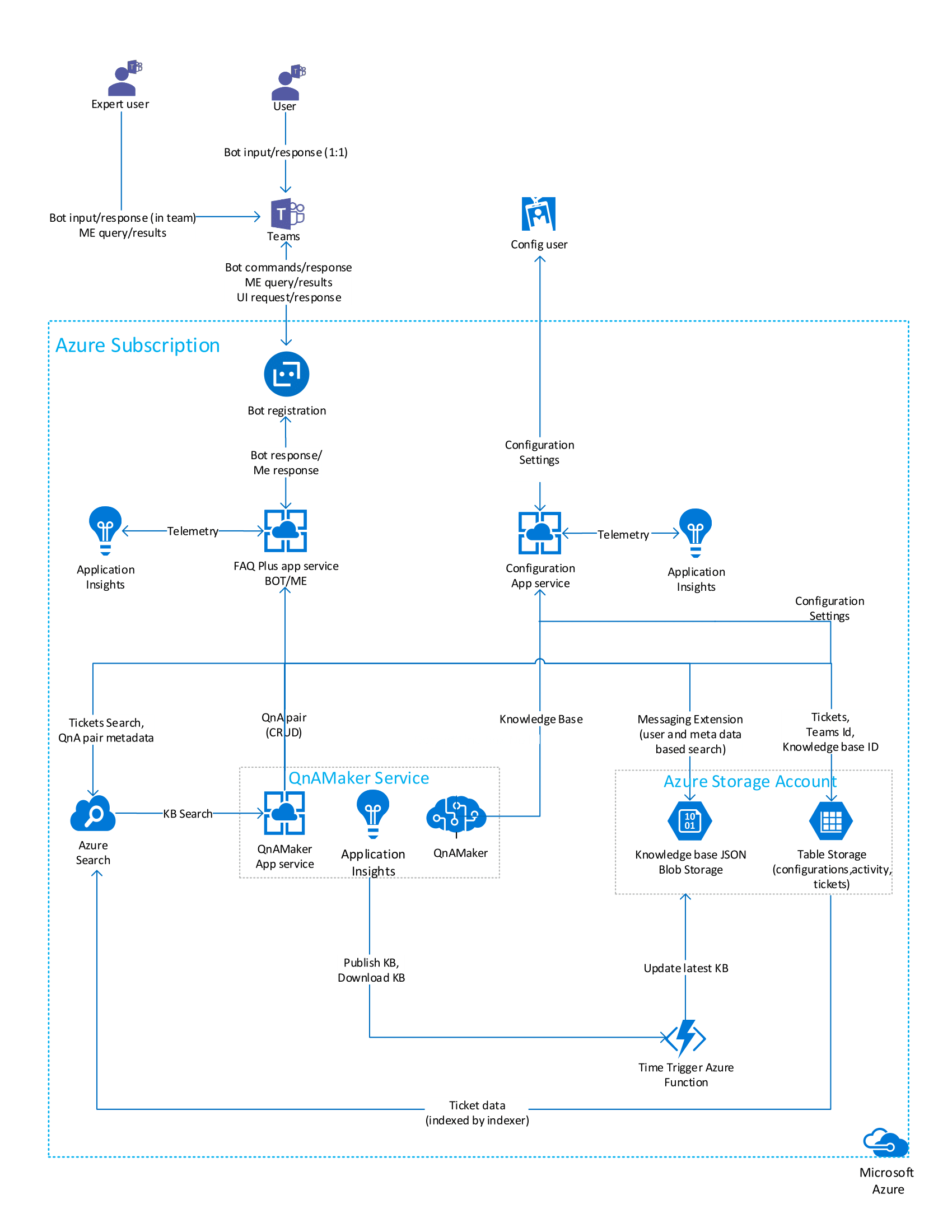
In case of such a scenario, admin user needs to make sure that the knowledge base id is updated in the configuration app and QnA pairs are existing in the knowledge base.

If no QnA pairs exist in knowledge base then add new ones either directly from the QnA maker portal or add it from the messaging extension in the experts team.

\*\*Didn't find your problem here?\*\*

Please, report the issue [here](https://github.com/OfficeDev/microsoft-teams-apps-faqplusv2/issues/new)

# Solution Overview



The **FAQ Plus** application has the following main components:

* **QnA Maker**: Resources that comprise the QnAMaker service, which implements the "FAQ" part of the application. The installer creates a [knowledge base](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/concepts/knowledge-base) using the [tools](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/concepts/development-lifecycle-knowledge-base) provided by QnAMaker.
* **FAQ Plus Bot**: The bot serves both end-users and experts team
  + The knowledge base (KB) in QnA Maker is presented to end-user in a 1:1 conversational bot interface. Through the bot, end-user can ask the question to bot, escalate to a designated experts team, send feedback about the app, or give feedback on specific answers.
  + The experts team receives notifications from the bot when end-users ask questions to expert or create feedback items. The bot tracks questions in a simple "ticketing system", with a basic life cycle of Unassigned -> Assigned to expert -> Closed. The bot notifies both the end-user and the experts team as the request changes states.
  + Experts in team can add new QnA pair to the knowledge base.
  + Application provides messaging extension that expert team members to search for tickets and QnA pairs from knowledge base.
* **Blob Storage** : The knowledge base with QnA and associated metadata is stored in blob storage by Azure function. The same is shown by messaging extension for respective section and search categories using Azure search service.
* **Azure Function**: Changes in QnA Maker are published every fifteen minutes to knowledge base by time triggered azure function.
* **Configuration Application**: An Azure App Service lets app admins configure the application to provide team and knowledge base details. These values are necessary to map the expert team and the associated knowledge base. Currently app supports only one knowledge base per tenant(deployment).

## QnA Maker

FAQ Plus v2 uses QnA Maker to respond to user questions; in fact, you can have a blank knowledge base to start using FAQ Plus. The precision and recall of the bot responses to end-user questions are directly tied to the quality of the knowledge base, so it's important to follow QnA Maker's recommended [best practices](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/concepts/best-practices). Please keep in mind that a good knowledge base requires curation and feedback: see [Development lifecycle of a knowledge base](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/concepts/development-lifecycle-knowledge-base).

For more details about QnA Maker, please refer to the [QnAMaker documentation](https://docs.microsoft.com/en-us/azure/cognitive-services/qnamaker/overview/overview).

## Bot and Messaging Extension

The bot is built using the [Bot Framework SDK v4 for .NET](https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-4.0) and [ASP.NET Core 2.1](https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.1). The bot has a conversational interface in personal (1:1) scope for end-users and in team scope for the experts team. It also implements a messaging extension with [query commands](https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/messaging-extensions/search-extensions), which the experts team can use to search for and share requests or knowledge base questions.

## Azure Function

Azure function [ASP.NET Core 2.1](https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.1) is used in building the bot. It publishes the QnA pair in the QnAMaker and stores the QnA in the blob storage. It creates a search index to search questions in the knowledge base tab on messaging extension.

## Blob Storage

Blob Storage stores the QnA pair in JSON format. QnA can be searched from the search tab in the messaging extension.

## Configuration App

The configuration app is a standard [ASP.NET MVC 5](https://docs.microsoft.com/en-us/aspnet/mvc/mvc5) web app. The configuration app will be used infrequently, so the included ARM template puts it in the same App Service Plan as the bot and QnAMaker.

From this simple web interface, app administrators can:

* designate the experts team
* set the knowledge base to query
* set the welcome message that's sent to all end-users
* set the content of the Help tab