Desarrollado por el equipo de Software.

Solución para adopción de herramientas de Microsoft

.

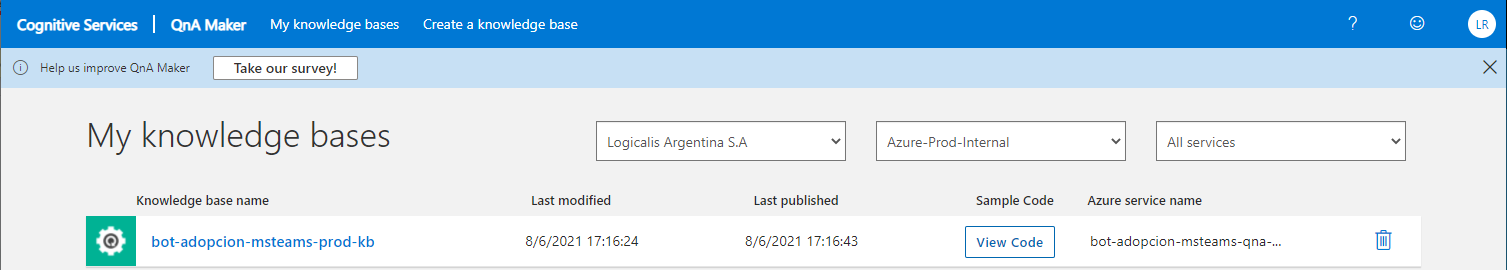
Creado: 10/6/2021

Actualizado: 10/6/2021

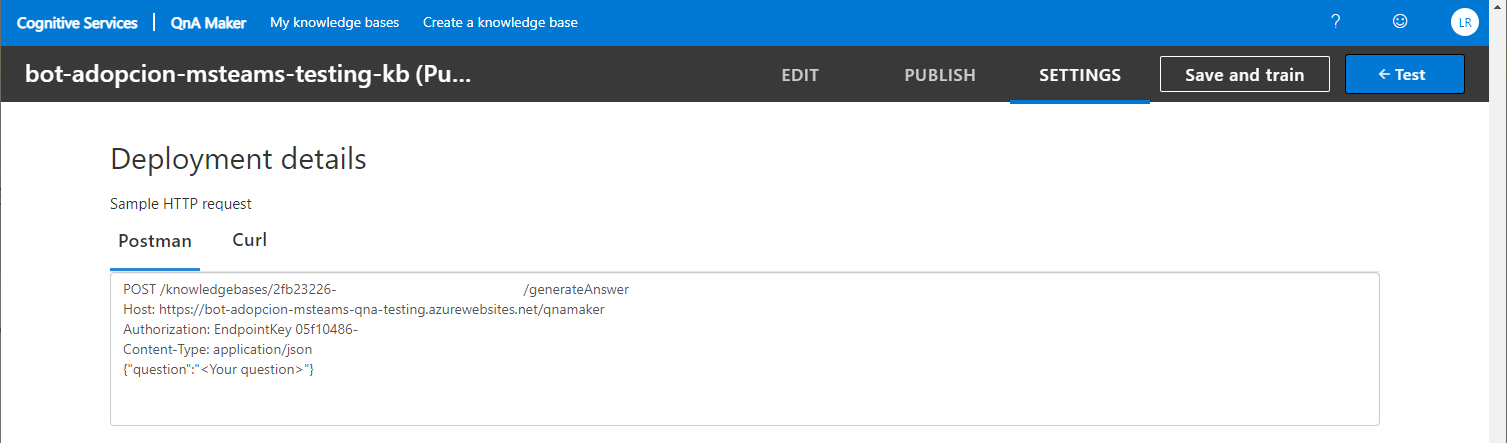
# Configuración en Azure

## Obtener datos de la Base de Conocimientos

* Dirigirse a <https://www.qnamaker.ai/Home/MyServices> e ingresar con las credenciales de Azure. Se mostrará la siguiente pantalla:



* Seleccionar la base de conocimientos llamada bot-adopcion-msteams-prod-kb.
* Dirigirse a la sección *Settings* del menú principal y en la parte inferior se encuentran las credenciales:



* Copiarlas y guardarlas de la siguiente forma que se necesitan más adelante en el proceso:

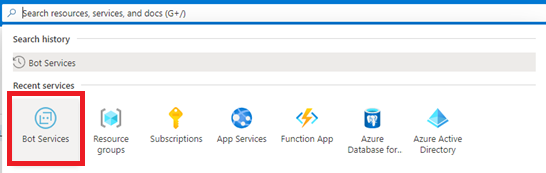
"QnAKnowledgebaseId": "2fb232a6-abbc-45cc-b4ec-7a01e3acf1e2",

"QnAEndpointKey": "05f10486-bd11-4fd6-bfa6-fedd7e86baa0",

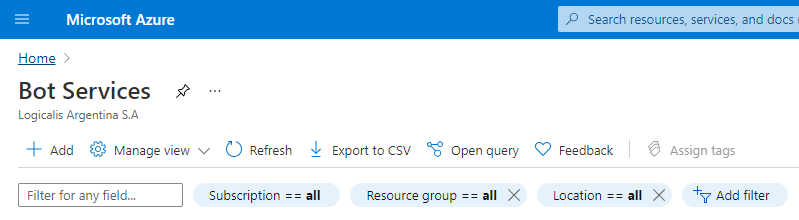
"QnAEndpointHostName": "https://bot-adopcion-msteams-qna-testing.azurewebsites.net/qnamaker"

## Generar Bot en Azure

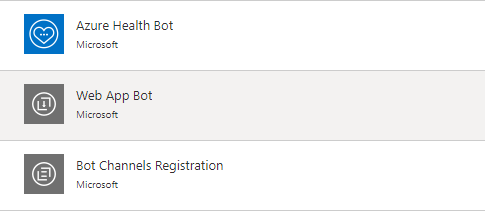
* Dirigirse al portal de Azure <https://portal.azure.com/> e ingresar con las credenciales.
* En la barra de búsqueda ingresar *Bot Services* y seleccionar el servicio.



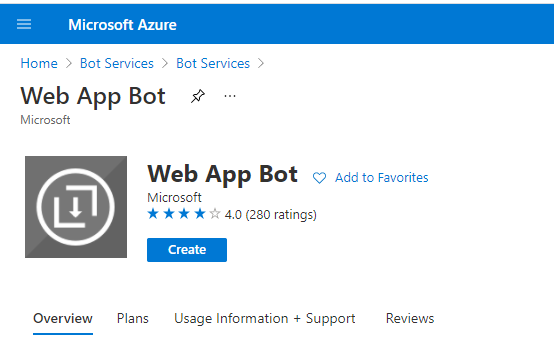
* En el administrador del servicio seleccionar *Add*



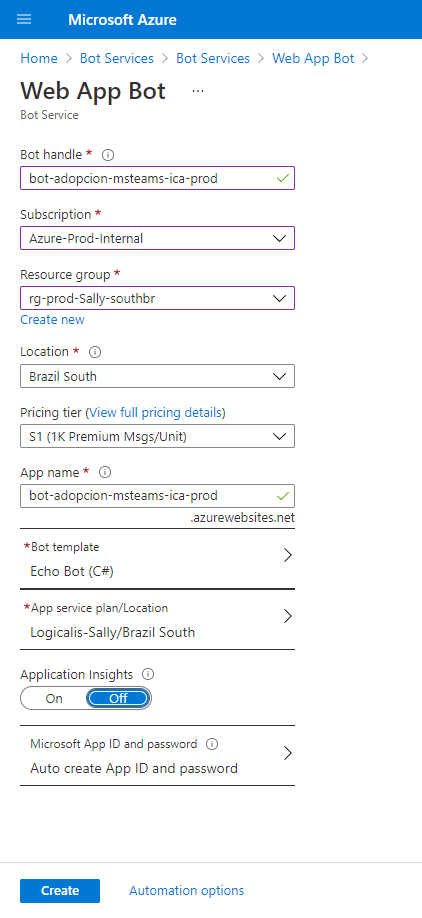
* Buscar el servicio llamado Web App Bot



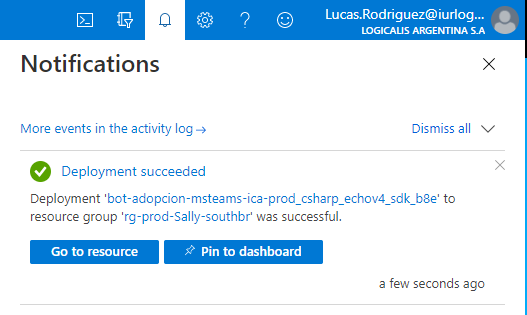
* Luego seleccionar *Create*



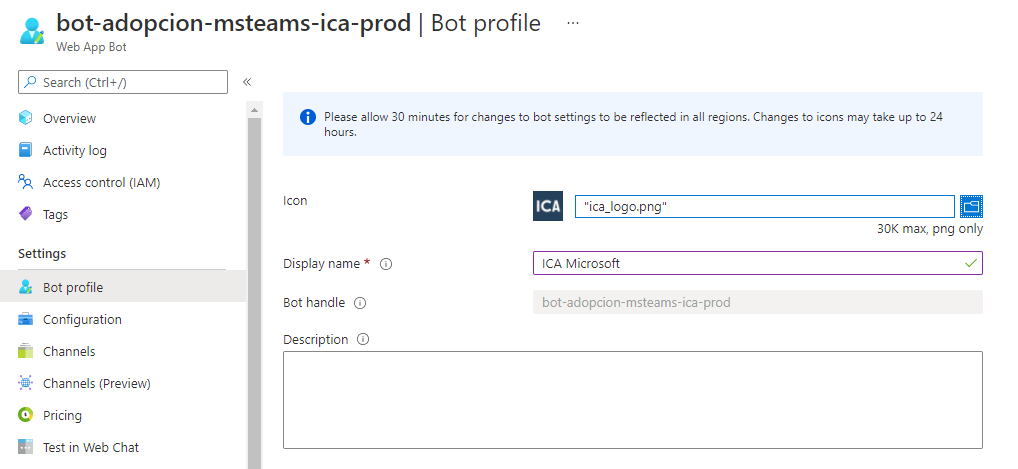
* Completar el formulario con los datos del proyecto de la siguiente forma y seleccionar Crear



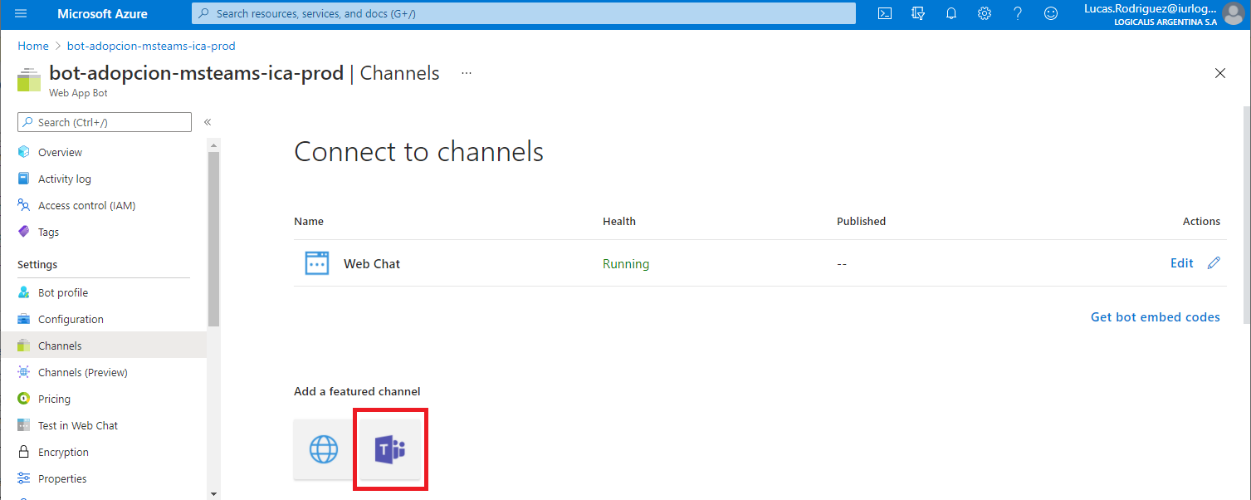
* Una vez creados los servicios seleccionar “Go to resourse”



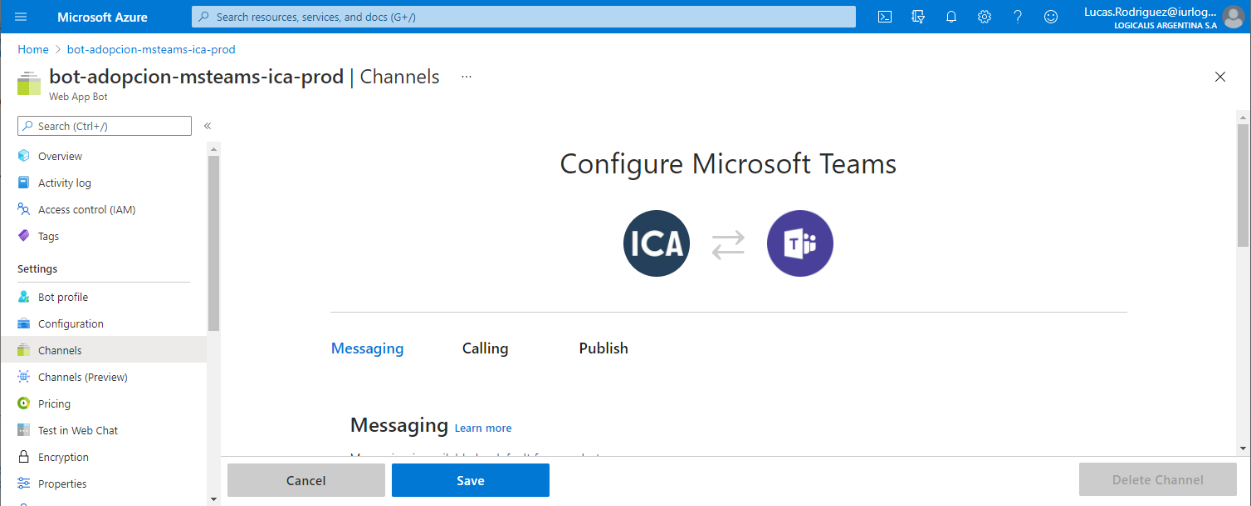
* Dirigirse a *Bot Profile* donde deberá cambiar el nombre y el icono del bot, los cuales serán mostrados en Ms Teams, y seleccionar *Apply*:



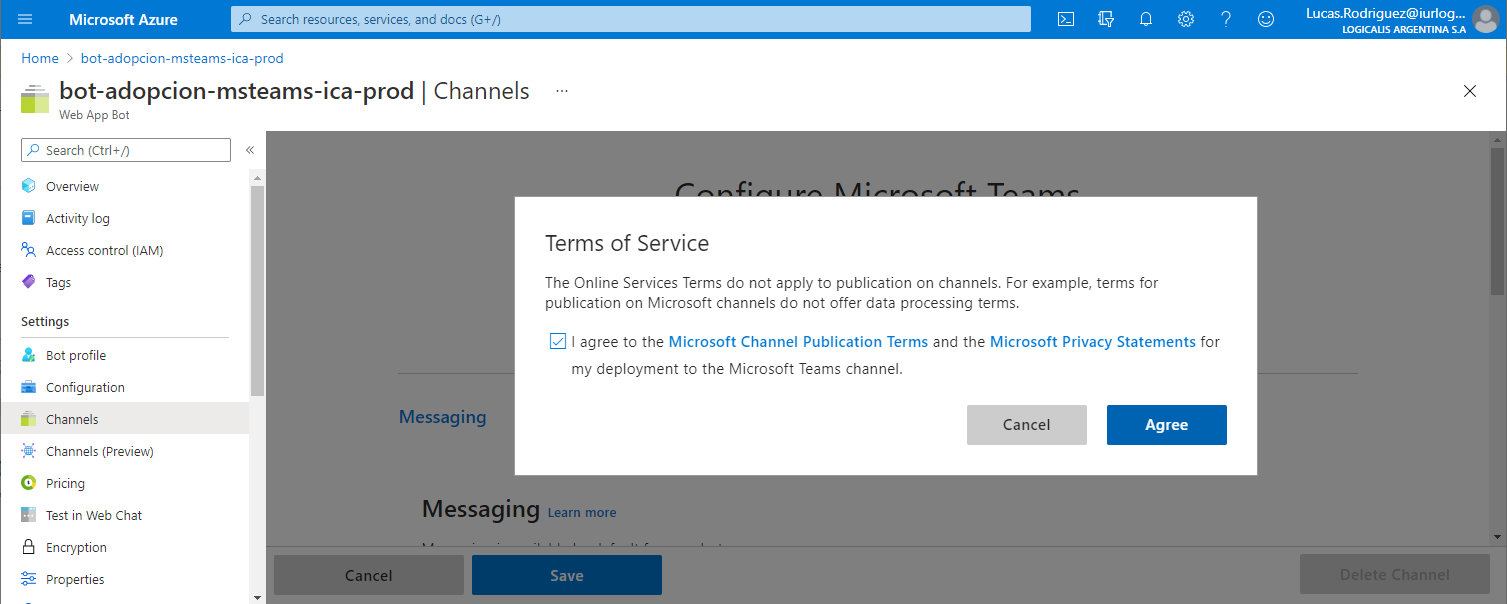
* Dirigirse a *Channels* y seleccionar *Ms Teams*:



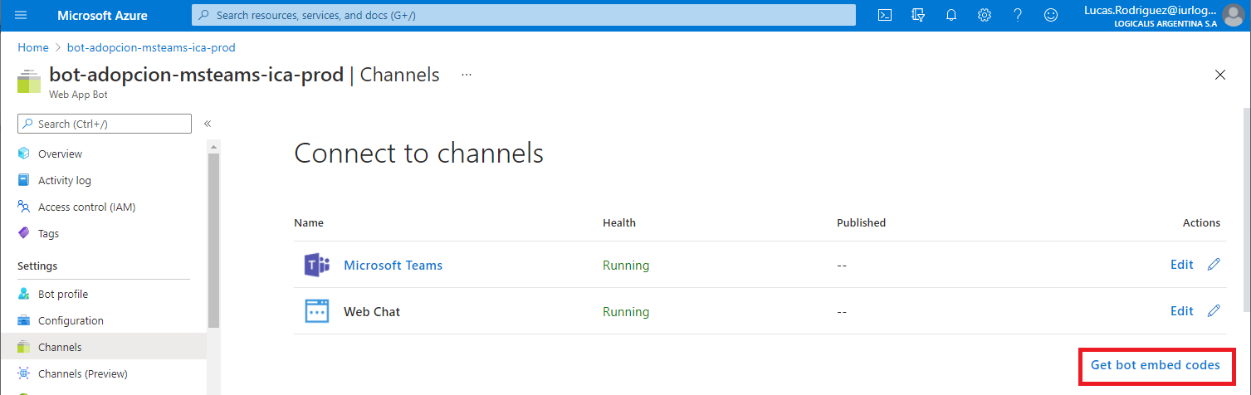
* Se abrirá la siguiente pantalla, presionar *Save*:



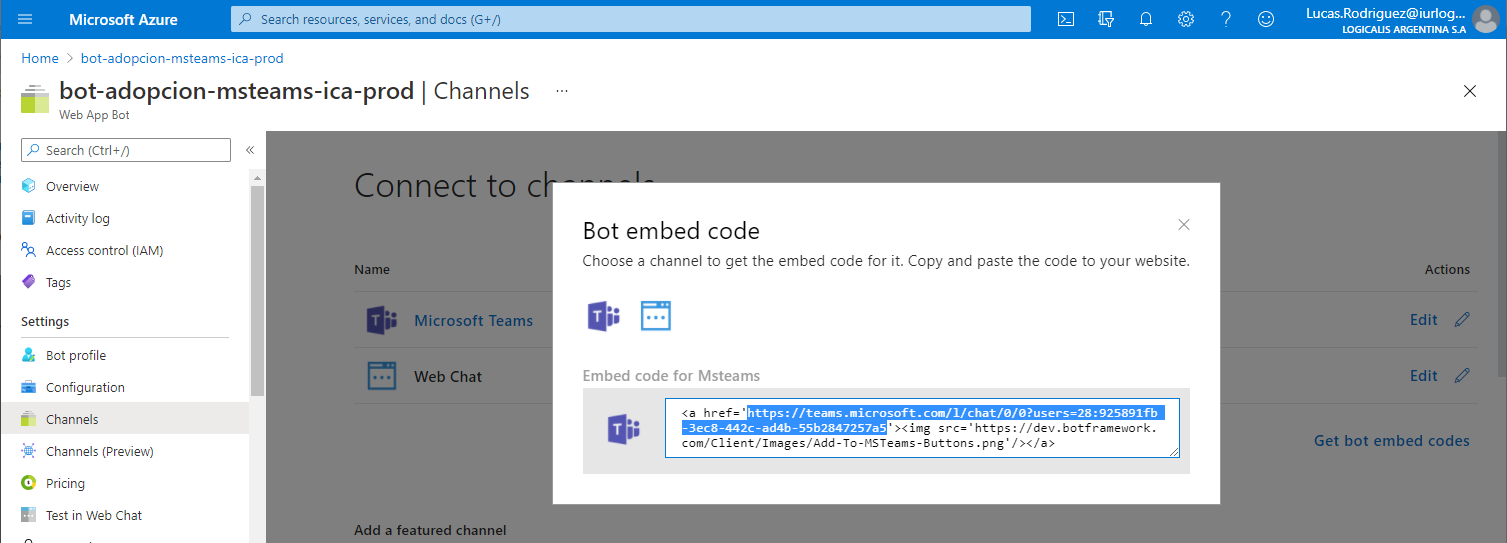
* Aceptar los términos y condiciones (*Agree)*:



* Recargar la pantalla y presionar sobre *Get bot embed codes*



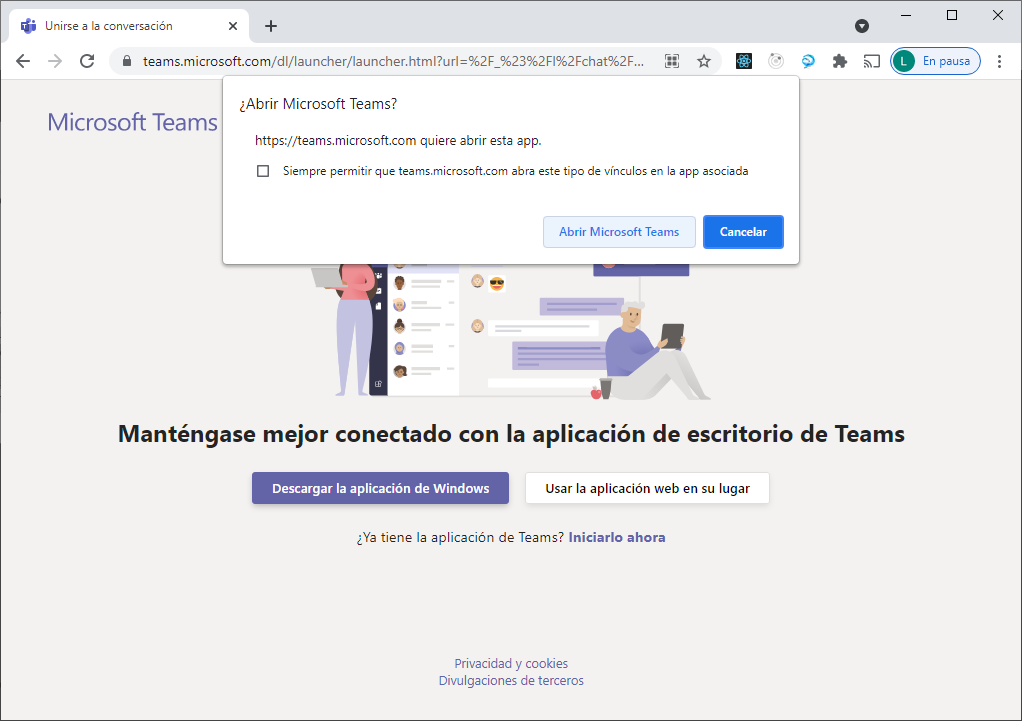
* Ahí se encontrará el link con el cual se podrá agregar el bot en Ms Teams simplemente abriéndolo con el browser:



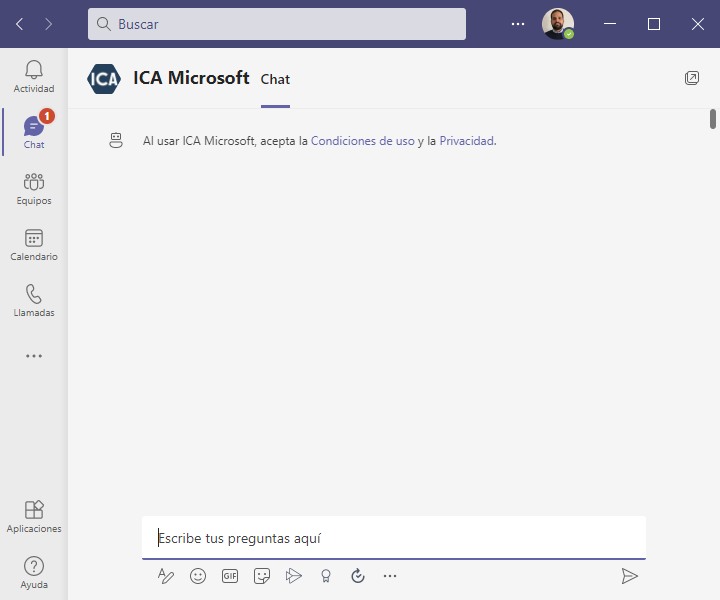
* En este caso será:

<https://teams.microsoft.com/l/chat/0/0?users=28:925891fb-3ec8-442c-ad4b-55b2847257a5>

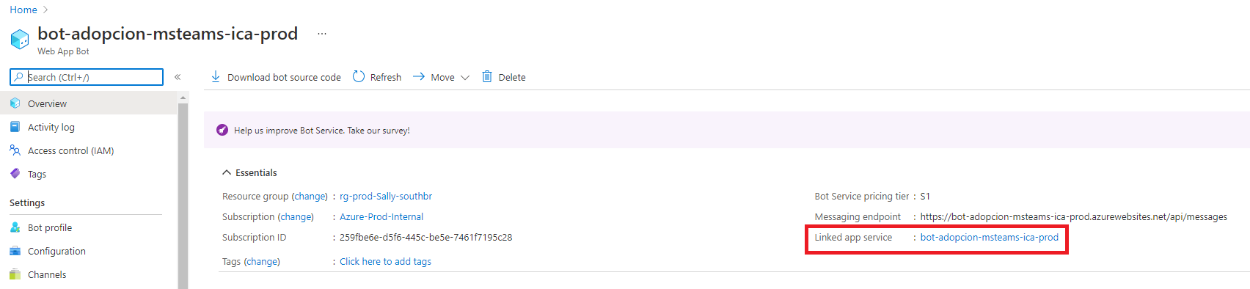
* Seleccionar *Abrir Microsoft Teams*



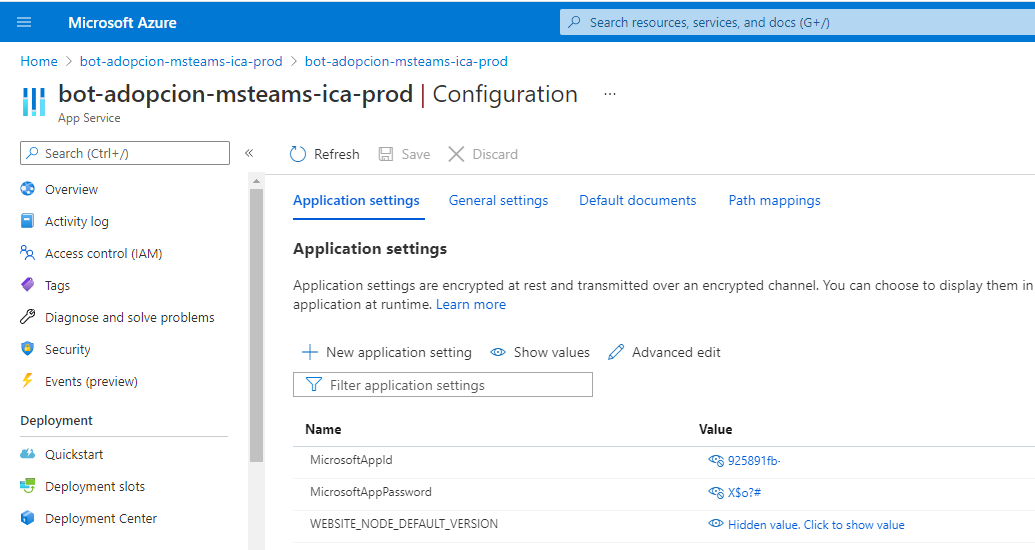
* Se abrirá Ms Teams y será redirigido a la conversación **la cual podrá utilizar una vez que termine el proceso completo.**



* Volver a la pantalla del portal, donde nos encontrábamos anteriormente, dirigirse a *Overview* y hacer click sobre *Linked app service*



* Una vez en el App service dirigirse a *Configuration* y copiar los valores de MicrosoftAppId y MicrosoftAppPassword



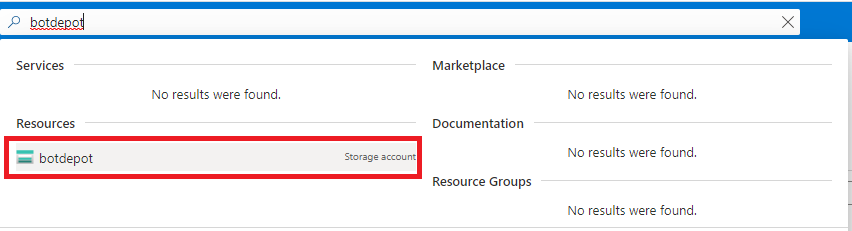
* Guardarlos de la siguiente forma para usar más adelante:

"MicrosoftAppId": "c026e097-XXXX",

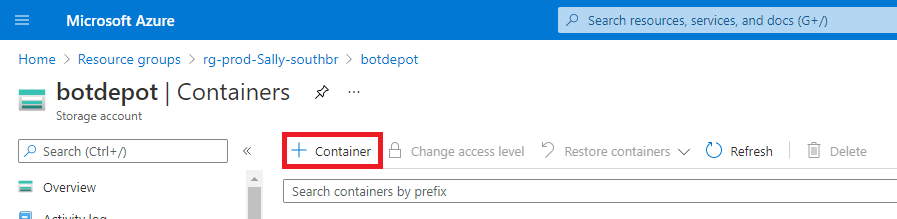
"MicrosoftAppPassword": "p;!FAa=XXXX",

## Generar Container de *Blob Storage* para imágenes de marca

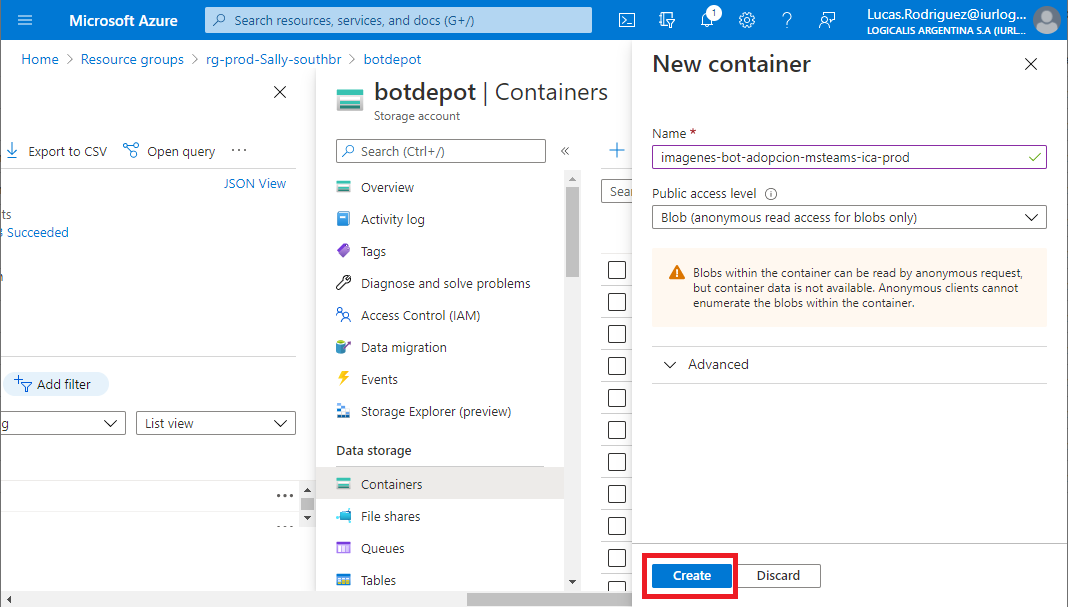
* En el buscador escribir “*botdepot*” y seleccionar el *Storage account*:



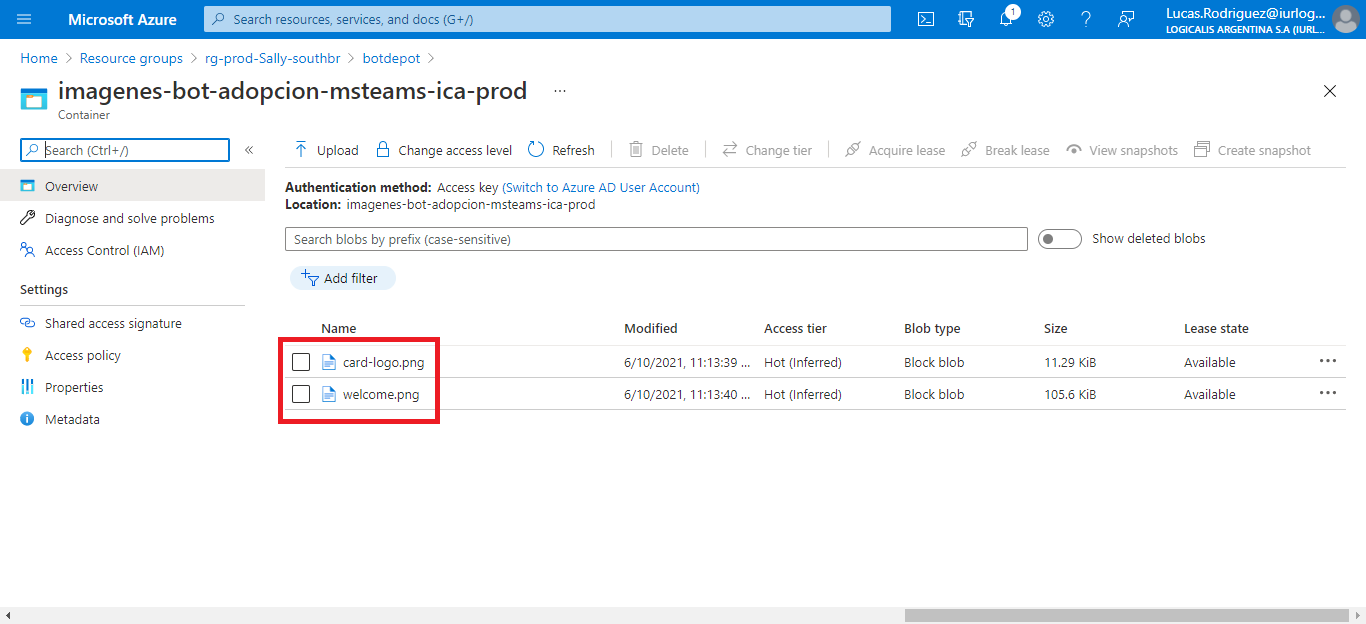
* Seleccionar *Container* para generar uno nuevo



* Completar el nombre, elegir la opción *Blob* en *Public Access level* y presionar *Crear*



* Dentro del Blob se deben subir dos imágenes llamadas *card-logo.png* y *welcome.png* las que se utilizaran en los cards del Bot.

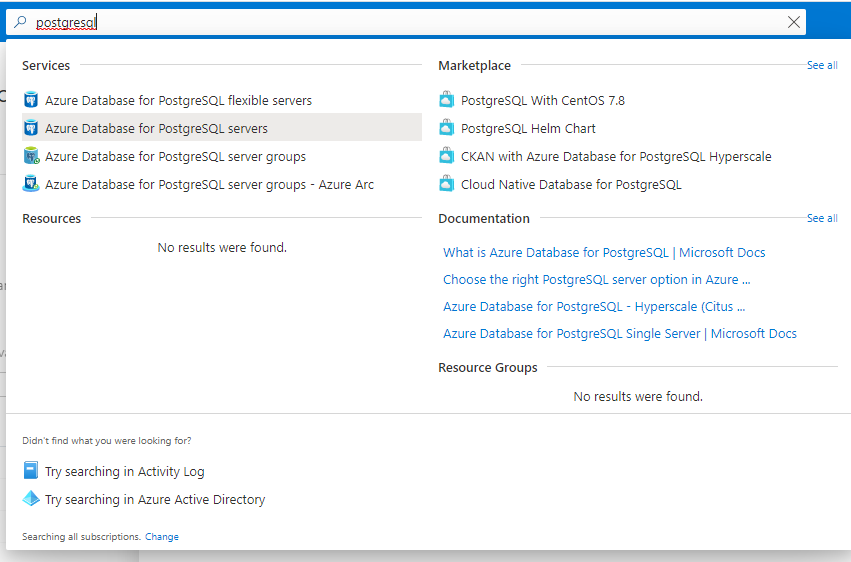


* Guardar el path del container de la siguiente forma para usar más adelante:

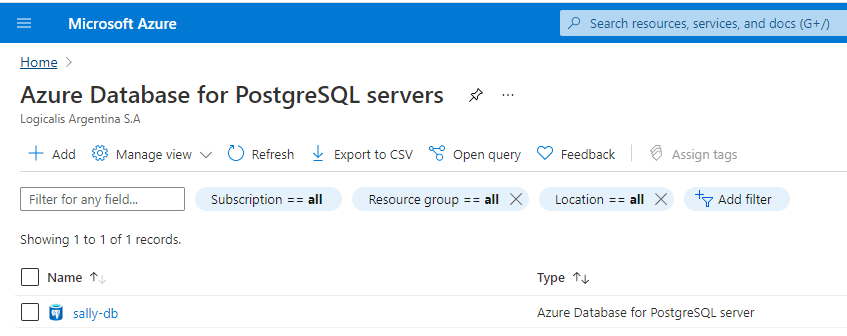
"BlobImagesUrl": "https://botdepot.blob.core.windows.net/<NOMBRE\_CONTAINER>",

## Obtener las credenciales de PostgreSQL

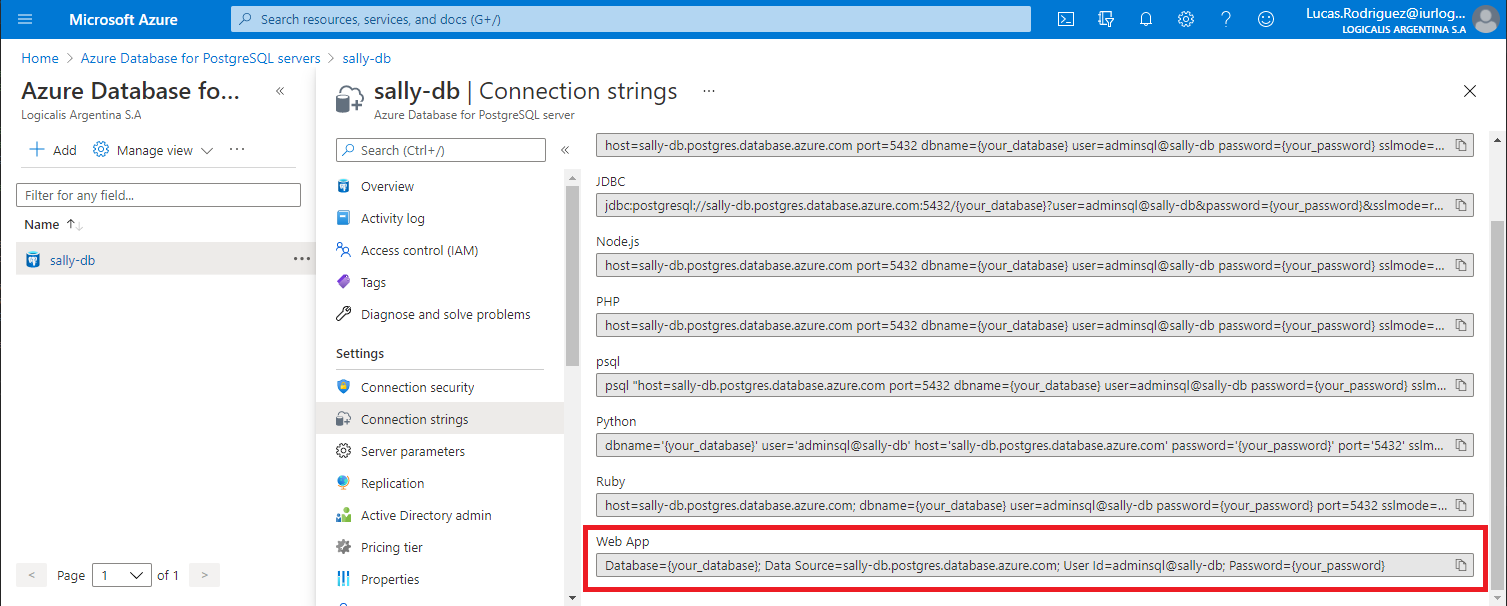
* En el buscador escribir “*postgresql*” y seleccionar la opción *Azure Database for PostgreSQL servers*



* Hacer click sobre *sally-db*



* Dirigirse a la opción *Connection string* y copiar la opción *Web App*



* Para esté proyecto el valor deberá ser:

"HarryContext": "Host=sally-db.postgres.database.azure.com;Database=bot-adopcion-msteams-prod-db;Username=adminsql@sally-db;Password=admin123&"

* Copiarlo junto con las demás credenciales.

# Configuración entorno Local

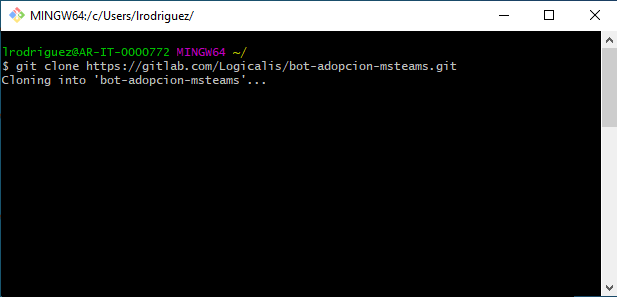
## Instalar Microsoft Visual Studio

* Descargar desde el siguiente link: <https://visualstudio.microsoft.com/es/downloads/>
* Completar el proceso de instalación
* Autenticarse en Visual Studio con la cuenta de Azure con la cual se generó el App Service.

## Repositorio de Git

* Clonar el proyecto de forma local con el siguiente comando utilizando [*Git Bash*](https://git-scm.com/downloads):

git clone <https://gitlab.com/Logicalis/bot-adopcion-msteams>



* Una vez descargado ejecute el *archivo QnABot.csproj* el cual cargará el proyecto en el Visual Studio.

## Configuración de Servicios

* Abrir el archivo appsettings.json el cual deberá completar con los valores obtenidos anteriormente.

Ejemplo:

{

"MicrosoftAppId": "c026e097-XXXXX",

"MicrosoftAppPassword": "p;!FAa=XXXX",

"QnAKnowledgebaseId": "2fb232a6-abbc-45cc-b4ec-7a01e3acf1e2",

"QnAEndpointKey": "05f10486-bd11-4fd6-bfa6-fedd7e86baa0",

"QnAEndpointHostName": "https://bot-adopcion-msteams-qna-testing.azurewebsites.net/qnamaker",

"ConnectionStrings": {

"HarryContext": "Host=sally-db.postgres.database.azure.com;Database=harry-msteams-demo-db;Username=adminsql@sally-db;Password=xxxx"

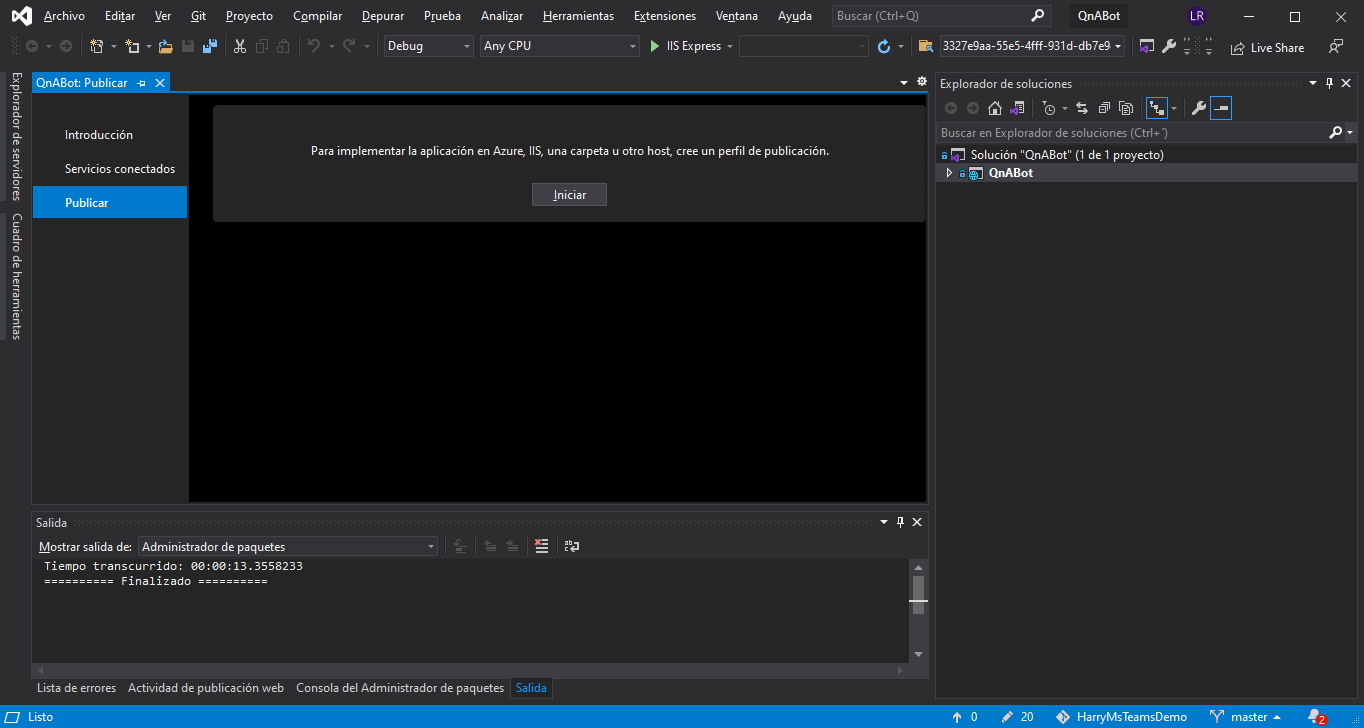
},

"BlobImagesUrl": "https://botdepot.blob.core.windows.net/imagenes-bot-adopcion-msteams-ica-prod"

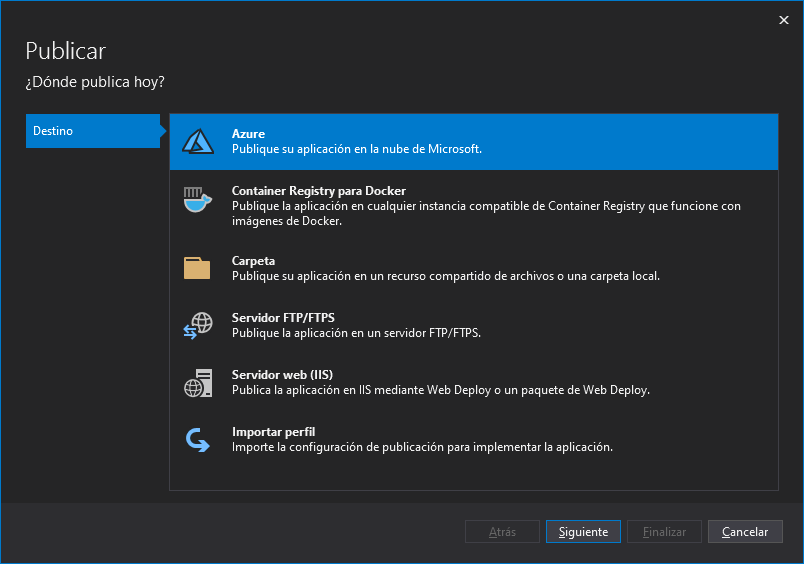
}

## Despliegue en producción

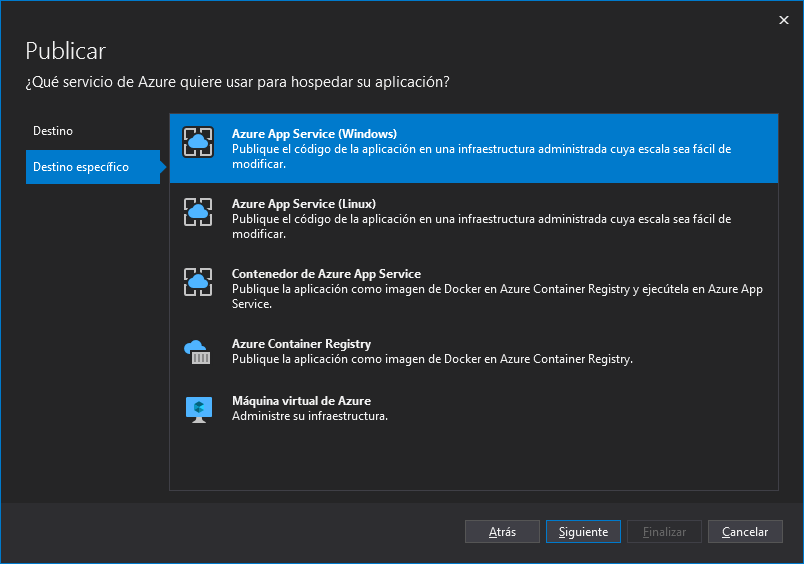
* En Visual Studio, en el menú dirigirse a Compilar > Publicar QnABot
* Esto abrirá una ventana donde deberá indicar el App Service en el cual se realizará el despliegue.



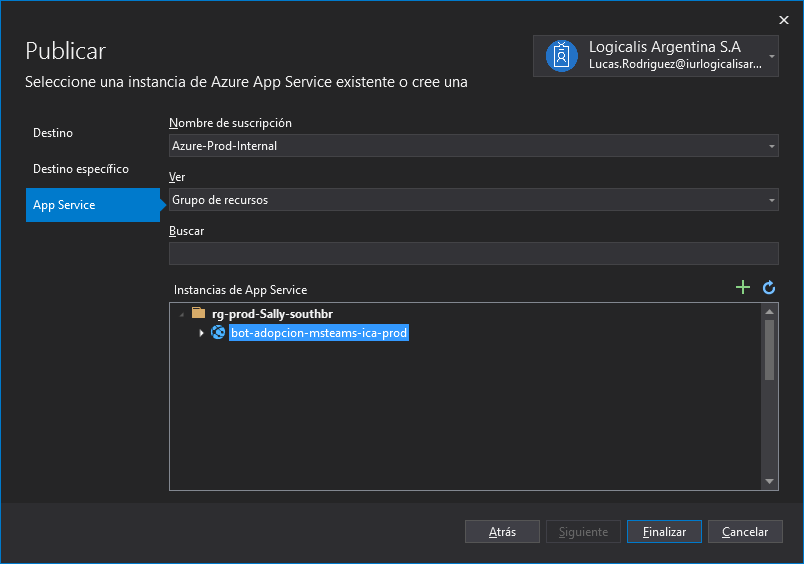
* Presionar *Iniciar*.



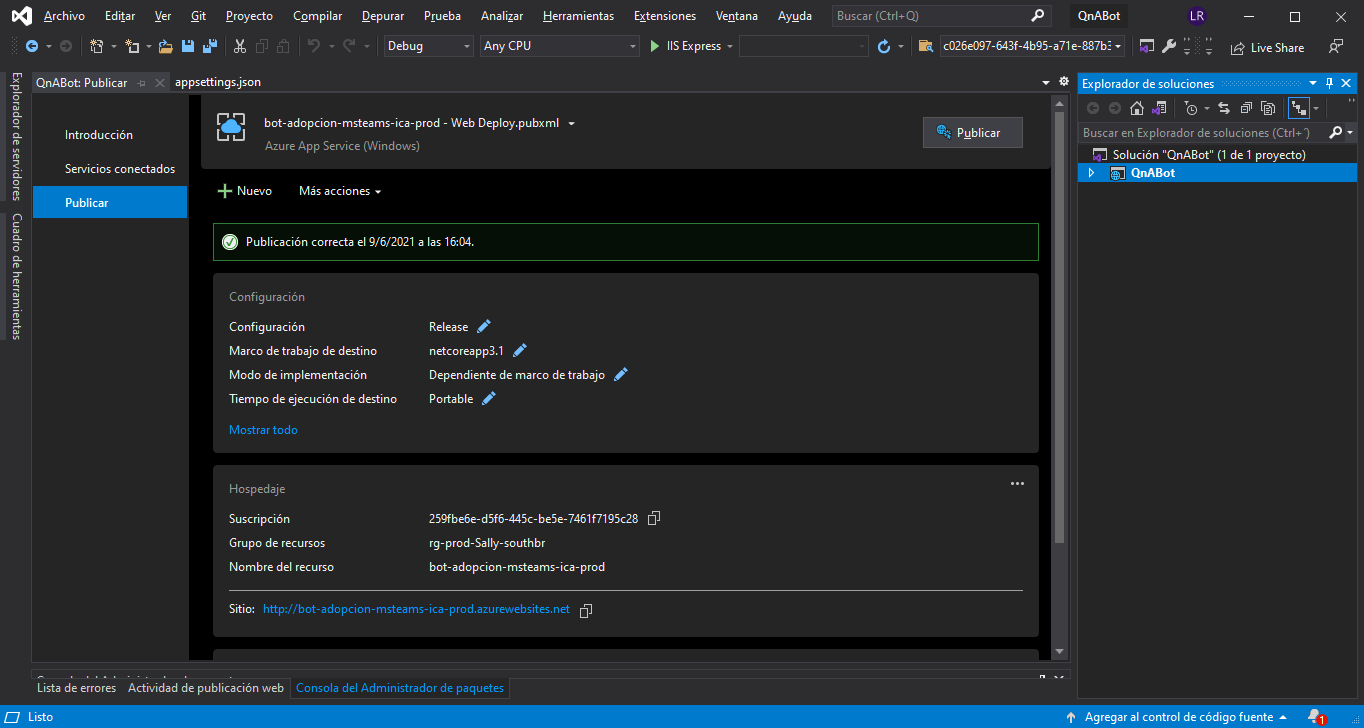
* Seleccionar *Azure* y presionar *Siguiente*



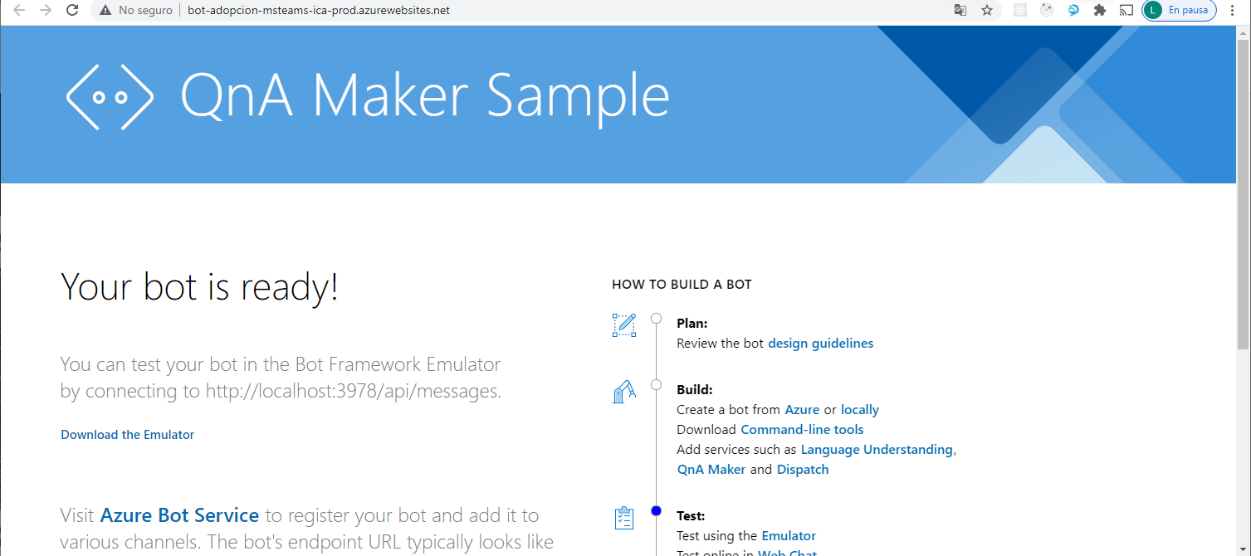
* Seleccionar *Azure App Service (Windows)* y *Siguiente*



* Seleccionar el App Service donde se hará el despliegue, por ejemplo, en este caso será: *bot-adopcion-msteams-ica-prod* y presionar *Finalizar*



* Una vez configurado el App Service se presionará *Publicar*, el proceso tomará unos segundos, una vez finalizado el despliegue se abrirá el browser con la siguiente pantalla:



* Ahora sí estará listo para probar desde **Ms Teams**:

