Managing your own Teams Recorder Bot for Apresa

Introduction

To record Teams calls, the Teams Recorder Bot is deployed in the cloud (Azure). This bot creates call recordings and sends these recordings to Apresa. This recorder bot can be installed and managed by yourself (described in this document), or you could rent access to an existing bot managed by someone else. Both options have their advantages.

Advantages of **managing your own** Teams Recorder Bot

- You can control settings of the Azure deployment, such as geographical server location
- If you have existing Azure deployments, the bot becomes part of the same infrastructure, allowing a single control interface
- You have control over timing of updates and other servicing of the bot
- The deployment is not shared by other companies for additional safety

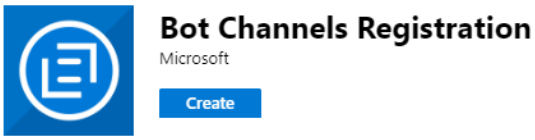
Advantages of **renting the use** of a Teams Recorder Bot

- No complicated setup or management, this is performed by an external party
- Sharing a deployment with other companies can reduce cost

This document describes how to create and manage your own recorder bot.

Bot Registration

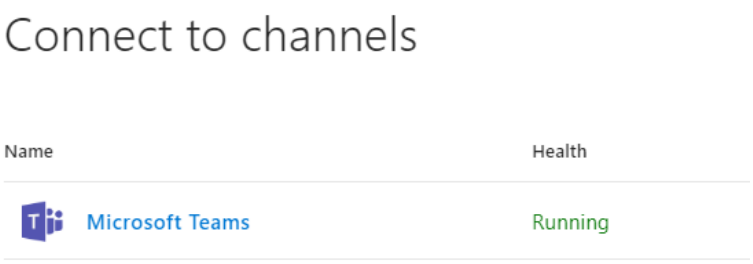
- Login to Azure
- Create a new resource of type Bot Channels Registration



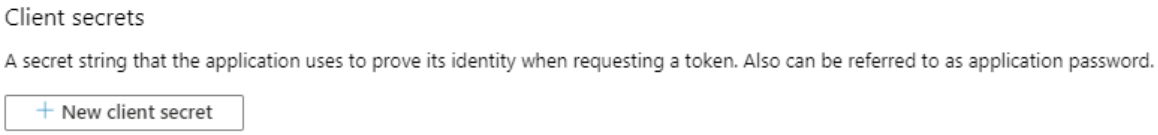
    - Bot handle: Choose a unique identifier containing only alphanumeric characters without spaces
    - Resource group: Select or create one
    - Messaging endpoint: this can be left empty currently
    - After filling in other settings as desired, click Create.
- Wait until the bot registration is created
- Open the new Bot Channel Registration
    - Open Channels
        - Add a Microsoft Teams channel
        - In the Calling tab, enable calling, and as webhook (for calling) fill in:

          https://<bot-domain>/api/calling

        - Save the changes



- Open Settings
  - Copy the displayed App ID (we will need this in another step)
  - Click on Manage next to the App ID
  - Click New client secret



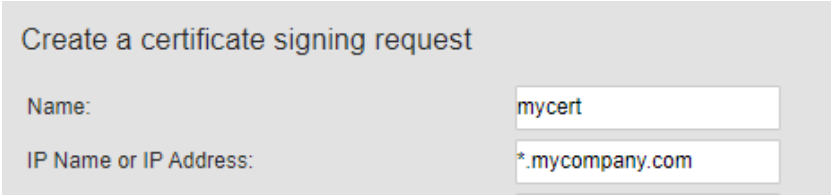
  - Copy the secret value (this is the App secret we will need in another step)
- Open API permissions (this option is visible after the previous step)
  - Click Add a permission
  - Select Microsoft Graph
  - Select Application permissions
  - Select Calls.AccessMedia.All and Calls.JoinGroupCall.All, and add these two permissions

  These permission will need to be consented by an administrator separately for each company that uses the recorder bot.

Certificates

The Teams recording bot requires the use of a subdomain for which you have a valid certificate. You can use an existing or create a new wildcard certificate (*.mycompany.com). The certificate is not allowed to be self-signed, but must be signed by a CA. You will also need the private key belonging to the certificate. If you have an existing valid signed wildcard certificate, you can use this. Otherwise you could create one using Apresa (or any other capable certificate tool).

- In Tools, choose Certificates
- In the Certificate signing requests tab, click Add
  - Name: This value is shown on the certificates page to identify a signing request in the database. It has no meaning for the created signing request itself.
  - IP Name: fill in for example: *.company.com (fill in your own company domain name). The star dot (*.) in front makes it a wildcard certificate.



  - Complete the other fields as desired and click OK

The certificate signing request can then be sent to a certificate authority for validation. After receiving a certificate from the CA, you will need to combine this with the private key that is stored on Apresa.

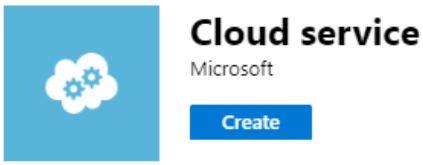
- Connect to Apresa using SSH and SFTP
- Upload the certificate and (if applicable) the intermediate certificate using SFTP
- Use this command to create a .pfx file:

  openssl pkcs12 -export -out bot.pfx -in domain.crt -in intermediate.crt -in /var/apresa/certificates/xxx.xxx.key

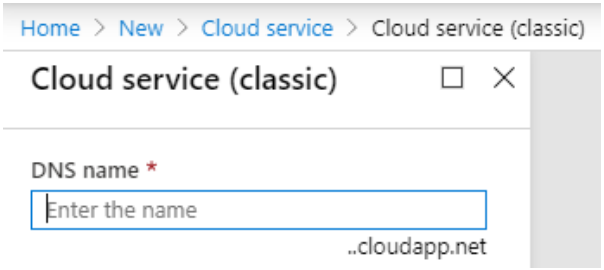
  filling in the right file names for the certificate(s). To find out the filename for the private key, use this command: ls -l /var/apresa/certificates/
  When prompted for a password for the .pfx, type a new password.
- Download the bot.pfx using SFTP
  This file together with the password will be needed in a later step.

Creating the Bot cloud service

- Login to Azure



- Create a new resource of type **Cloud service** (classic).
  - DNS name: choose a unique identifier. This will be used to create a subdomain of cloudapp.net. This is an alternative access point, and not the domain you will need a certificate for. Later we will create a DNS CNAME pointing to this cloudapp.net subdomain from your own subdomain.



  - After filling in other settings as desired, click Create.
- Wait for the service to be created
- Open the created service
  - Open Certificates
  - Click Upload
  - Upload the .pfx file (containing the signed certificate and private key of the domain), and input the password of the .pfx file

Domain configuration

The recorder bot will reside on a subdomain, for example teamsbot.company.com is possible. You can choose freely the subdomain, but it has to be on a domain that you own, that you have a private key and certificate for, as described in the section Certificate.

- Open the DNS configuration control panel of your provider

- Create a new CNAME entry, directing your subdomain to the cloudapp.net subdomain created in the previous step in Azure. For example, a CNAME pointing from teamsbot.company.com to xyz.cloudapp.net.

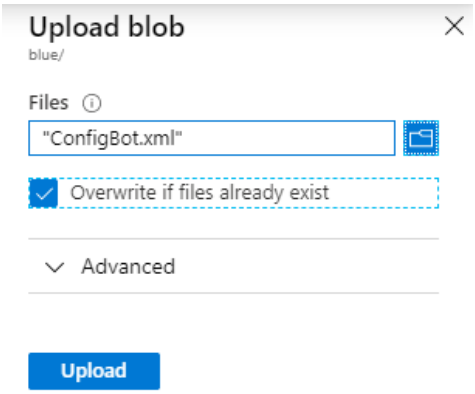
Configuration

The bot is configured using a file named BotConfig.xml. Included in the software package is a file sample BotConfig.xml. Open the file using a text editor, and adjust it for your configuration.

- tenants: The configuration file contains a section for each tenant. If there is only one company recorded using the bot, then only one tenant entry will be present.
- tenant.id: this is Microsoft tenant ID. It can be seen in the logs, and also can be retrieved by the tenant in the PowerShell.
- tenant.name: the name of the tenant (for easier identification)
- tenant.server: the domain name of the Apresa server. Do not add the https:// prefix, this will be added automatically. The server must be on public internet and accessable with https.
- tenant.serverAuth: this is the Apresa API authentication code, which the bot will use to upload recordings to Apresa
- tenant.serverTenant: Uploaded recordings can be placed automatically in a certain Tenant in Apresa. This is an optional setting.
- tenant.channelLimit: Optionally the tenant can be restricted to a maximum number of similtaneous recordings. If this option is absent, no restriction is applied for this tenant (except on the global level).
- service.auth: This is the authentication code used by the manager of the bot, for example to retrieve log files

The recorder bot needs a place to load settings and save settings. This will need to be in a blob container in Azure.

- Login to Azure
- Create or open an existing Storage account (classic) resource, that the recorder bot may use to load and save settings
- Go to Access keys, and copy the Connection string. We will need this to configure the recorder bot.
- In the same storage account, under the Blob service, go to Containers
- Open or create the Container for use by the bot
- Upload BotConfig.xml



The bot will load but not alter BotConfig.xml. It will create and update the file BotData.json instead, and store this in this blob container.

After updating the configuration file BotConfig.xml in Azure, to apply the changes to the bot, there is a command in the Admin Teams Bot Control Panel to let the bot reload its configuration.

Customizing the Bot package

- Open the file ServiceConfiguration.Cloud.cscfg in a text editor
- Fill in the following fields:

- ServiceDnsName: this should be something like xyz.cloudapp.net (xyz is what was chosen during creation of the bot cloud service)
- ServiceCNAME: this must be the subdomain for which you have a signed certificate, for example: teamsbot.company.com
- AadAppId: the App ID created during Bot registration
- AadAppSecret: the App Secret created during Bot registration
- ConfigStorage: Fill in the Azure storage account Connection string that the bot may use to load and save settings.
- ConfigContainer: the name of the Blob container to use.
- Certificate thumbprint: To find out the thumbprint, in Windows open the .crt file of the domain, then on the Details tab, copy the Thumbprint.

Deploying the bot service

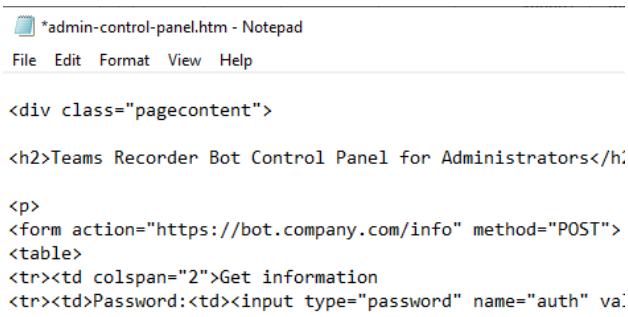
- Login to Azure
- Open the Bot cloud service
- Select Production
- Click the Update button in the Overview screen
  - Upload the provided .cspkg (unmodified) and .cscfg file (adapted)
  - Enable "Deploy even if one or more roles contain a single instance"
  - Click OK to start deployment

Teams Bot Control Panels

The control panels are HTML files that you can open in a browser, to control the bot.

The HTML files need to be customized first before they can be used.

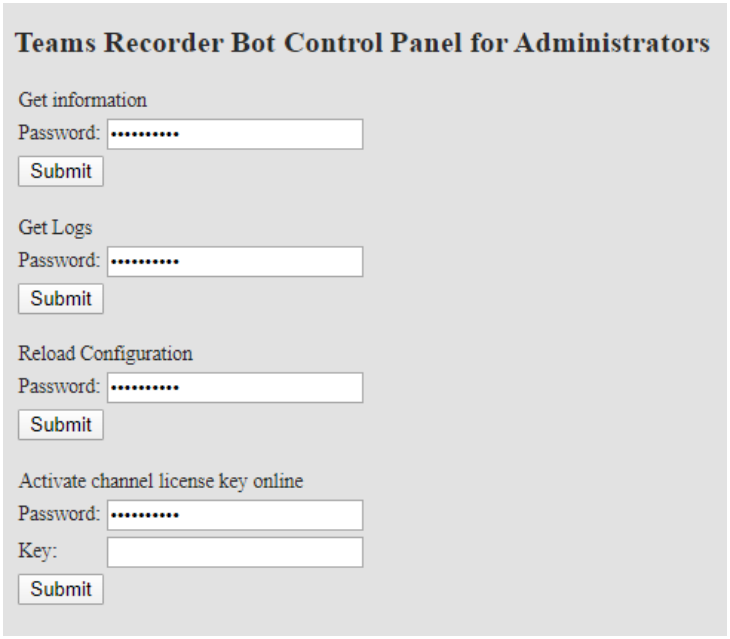
- Open the HTML in a text editor
- Do a search and replace, searching for https://localhost and replace this with https://bot.company.com (fill in the domain name of your bot)



To use the control panel:

- Open the .htm file in a browser
- Fill in the password field and other required fields for the action you want to take, and click Submit
- The browser will try to contact the Teams Bot, and show its reply

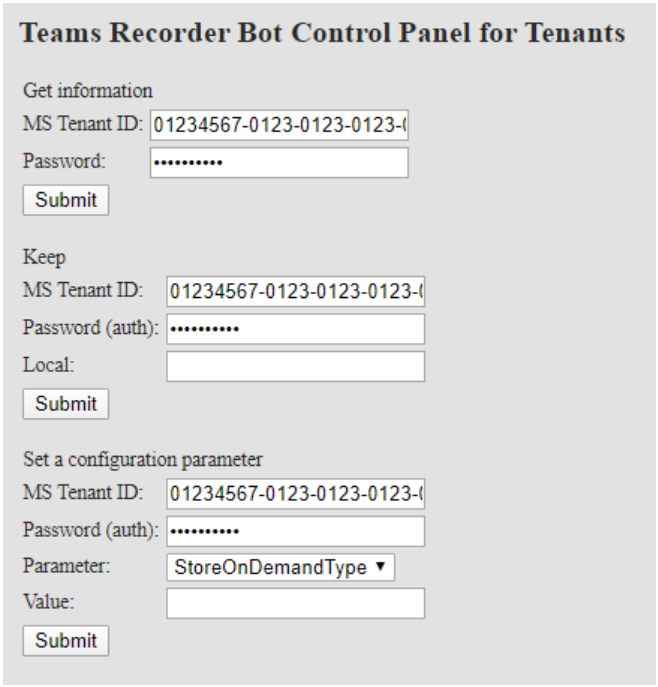
The admin control panel is to be used by the manager of the bot.



The output response is in JSON format, except for the logs. The information output includes information about currently active calls.


```
{
  "botVersion": "1.0.1.0",
  "licensedChannels": 11,
  "calls": [],
  "tenants": [
```


The tenant control panel could be used by one or more tenants (companies that are recorded using this bot).



The actions that are available in the tenant control panel correspond to the available actions in the Tenant API (see separate document). The control panel is one possible way to access the Tenant API. It can also be used directly by a HTTPS client.

Bot Channel Licensing

The bot is limited to a number of simultaneous recordings by the number of installed licensed channels. To add channels:

- Open the admin Teams Bot control panel (see previous section)
- Fill in the control panel password, and license key, and click Submit
- The bot will show the updated number of licensed channels

The licenses are tied to the domain name of the bot.

Updating the bot

When a new version of the bot is available, the bot can be updated as follows:

- Login to Azure
- Open the Bot cloud service
- Select Staging
- Click the Update button in the Overview screen
    - Upload the new .cspkg file together with the existing adapted .cscfg file
    - Enable “Deploy even if one or more roles contain a single instance”
    - Click OK to start deployment
- When Staging has started, click Swap
  When the swap is completed, the new version is running in the Production environment, and the old version in the Staging environment.
- Delete the Staging deployment.

Calls that are active during updating will probably not be recorded. The idea to update Staging and then swap, is to avoid down-time. If this is not a problem, alternatively, you could also update directly to Production, and do not swap afterwards.