|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | | |
|  |  |  |  | |
| Microsoft Teams Security and Administration Masterclass |  | Automating Team Creation |  |  |
|  |  |  |  |
|  |  |  | Instructor Led Lab |
|  |  |  |  | |

# Module overview

|  |  |
| --- | --- |
| **Introduction** | Contoso want to create a proof of concept Teams approval and creation application based on a SharePoint List, Flow and Microsoft Graph API. This will allow them to control the creation and proliferation of Teams during their migration project. The lab below provides details of how to approach such a task. |
| **Objectives** | After completing this lab, you will be able to:   * Create App registrations in Azure AD * Create custom connectors in MS Flow * Create SharePoint sites and lists * Create an approval flow |
| **Prerequisites (if applicable)** | An Office 365 Enterprise tenant with the appropriate PowerApps Licensing  Confidential, Internal and External Group classifications have been defined for the tenant. |
| **Estimated time to complete this lab** | 3-4 hours |

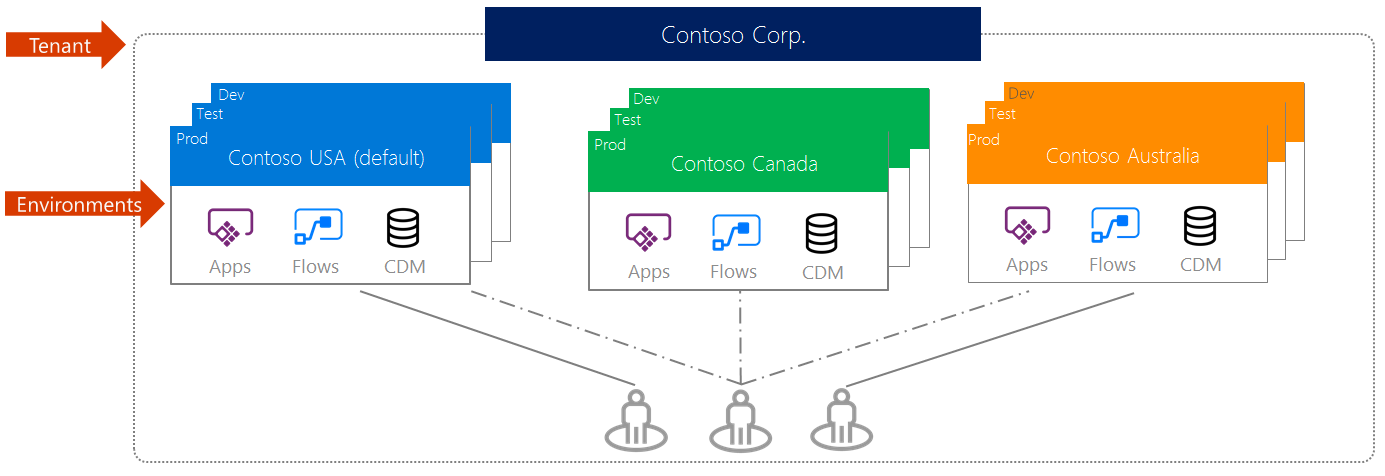
# 

# Create a trial environment for MS Flow

An **environment** is a space to store, manage, and share your organization’s business data, apps, and flows. This concept is valid for PowerApps, Flow and Dynamics. They also serve as containers to separate apps that may have different roles, security requirements, or target audiences.

You can create environments for different purpose. A Trial environment is for trying out the environment and database with Common Data Service experience. It expires after certain period.

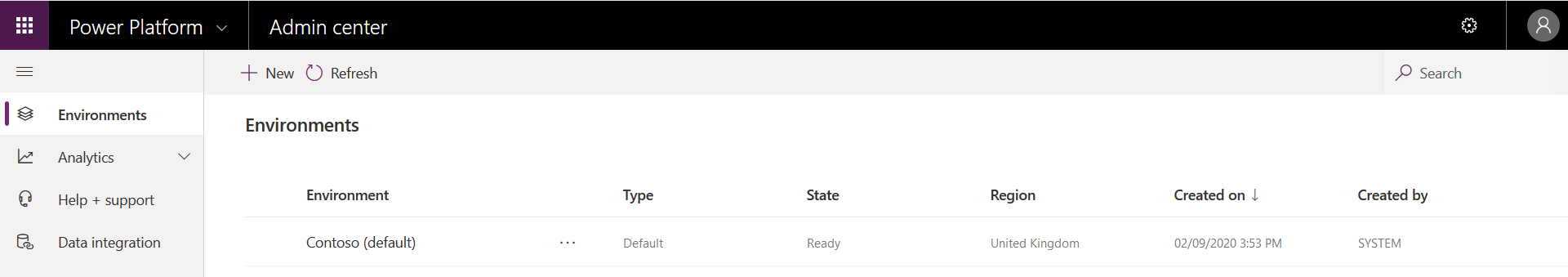
## If you haven’t created a demo environment, please go back to the prerequisites document.



1. Navigate to **Microsoft PowerPlatform Admin Portal** (log in as a trial tenant admin if necessary):

<https://admin.powerplatform.microsoft.com/environments>

1. Click + **New** to create a new environment:



1. Set the following environment settings:

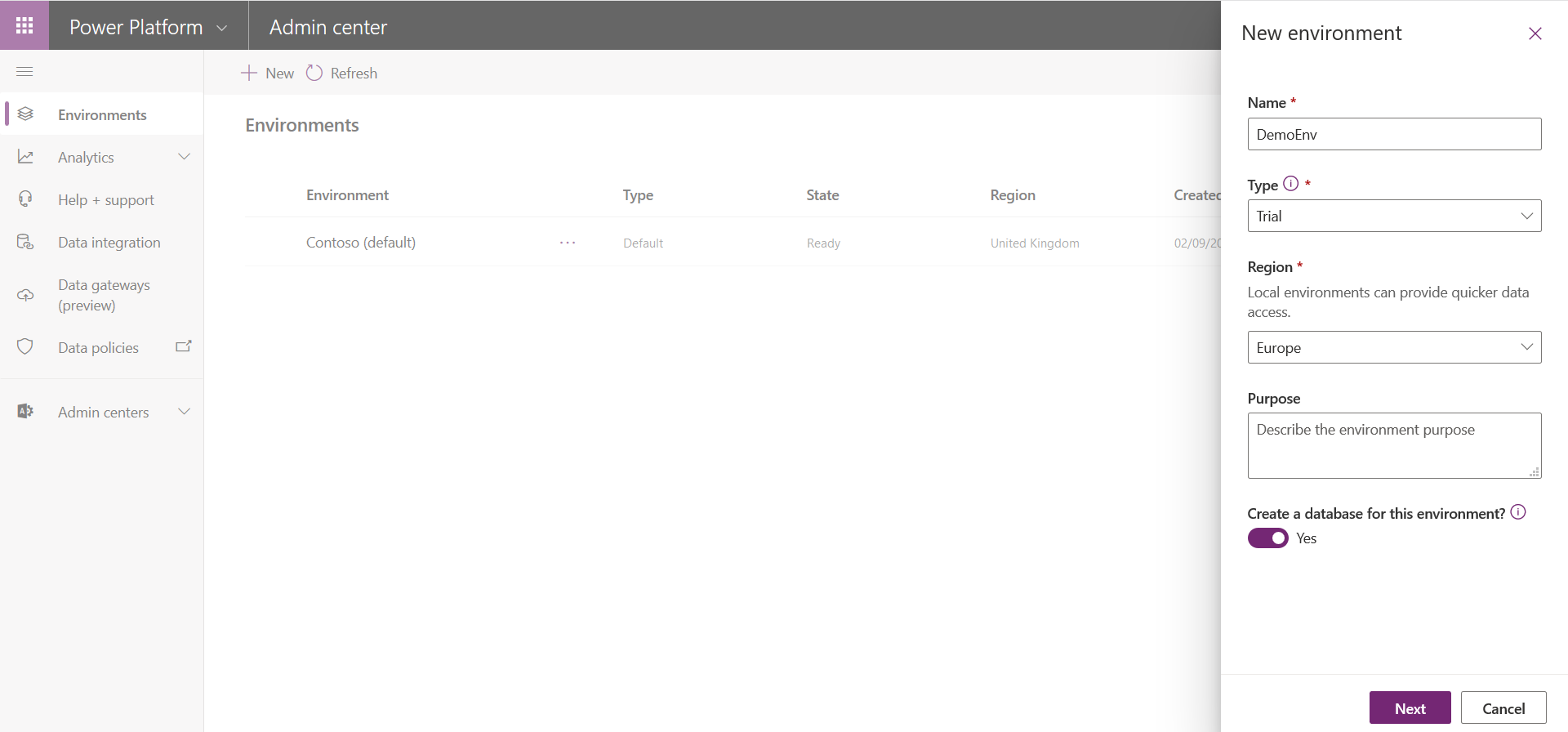
**Environment name:** DemoEnv

**Region:** Europe (Please use Europe for the lab)

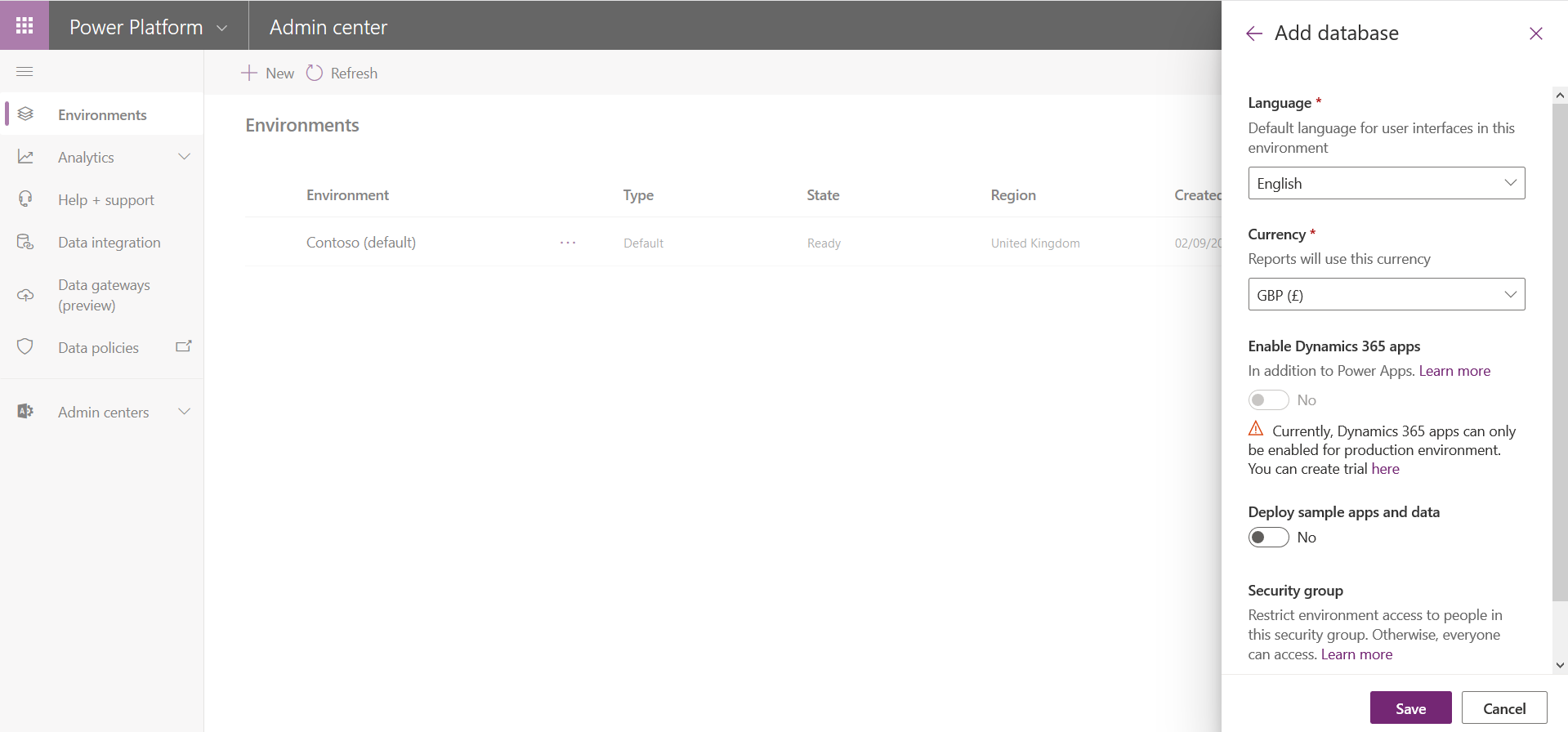
**Environment type:** Trial

**Create a database for this environment:** Yes

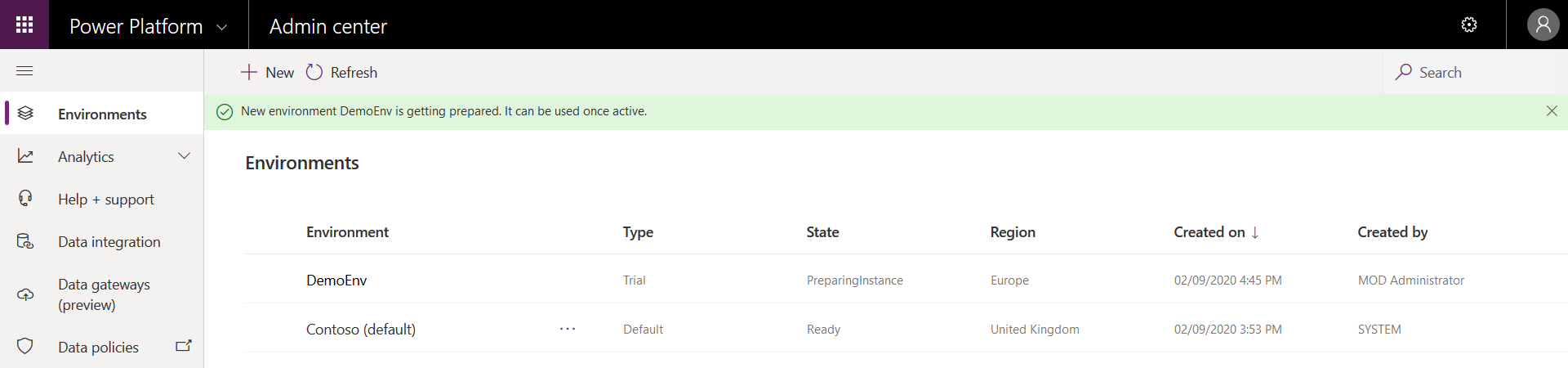
**Note: USE EUROPE for the Demo Environment.**



1. Press Next, leave the following screen’s settings at their defaults and select Save.



1. Make sure that the environment is provisioned:

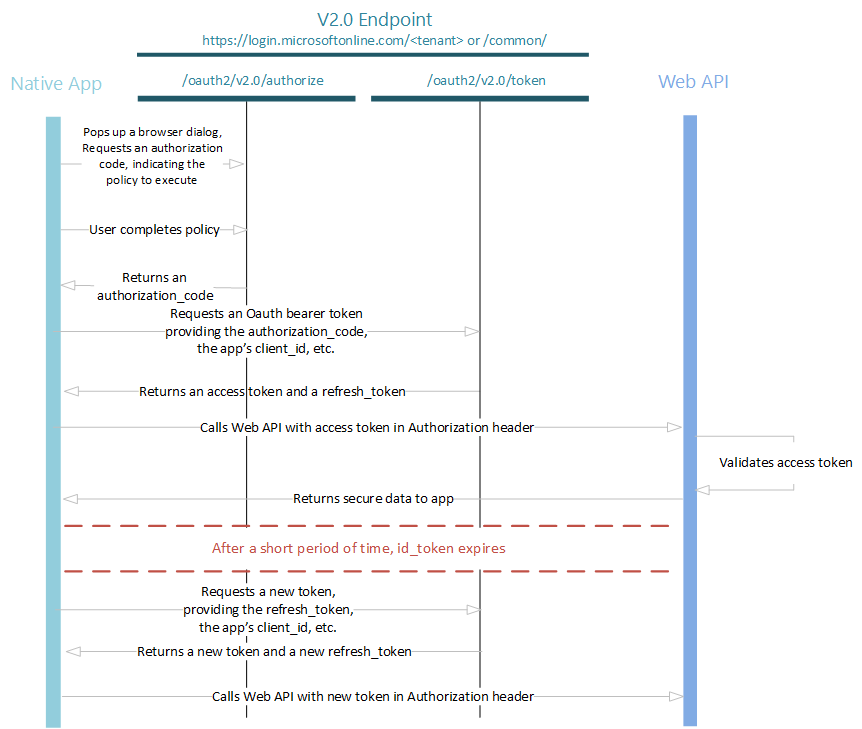


1. While the database spins up, proceed to the next task…

# Create App registration in Azure AD

1. To build an app that uses a **work account (Azure AD)** to sign-in, you'll first need to register the app with Microsoft. We will then build a custom connector that makes requests to **Microsoft Graph API**.

The **OAuth 2.0 authorization** code grant can be used in apps that are installed on a device to gain access to protected resources, such as web APIs. Using the app model v2.0 's implementation of OAuth 2.0, you can add sign in and provide API access to your mobile and desktop apps.

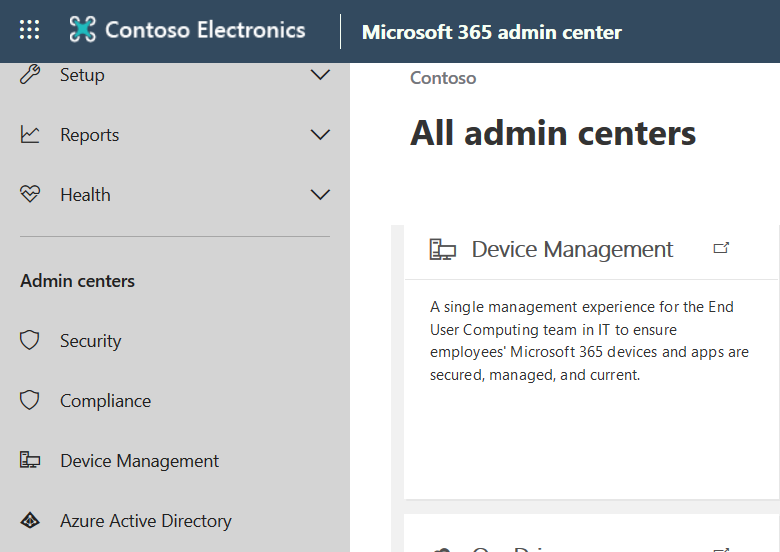


**Recommended reading:** <https://docs.microsoft.com/en-us/azure/active-directory/develop/v1-protocols-oauth-code>

1. Navigate to **Office 365 Admin Portal** (log in as a trial tenant admin if necessary):

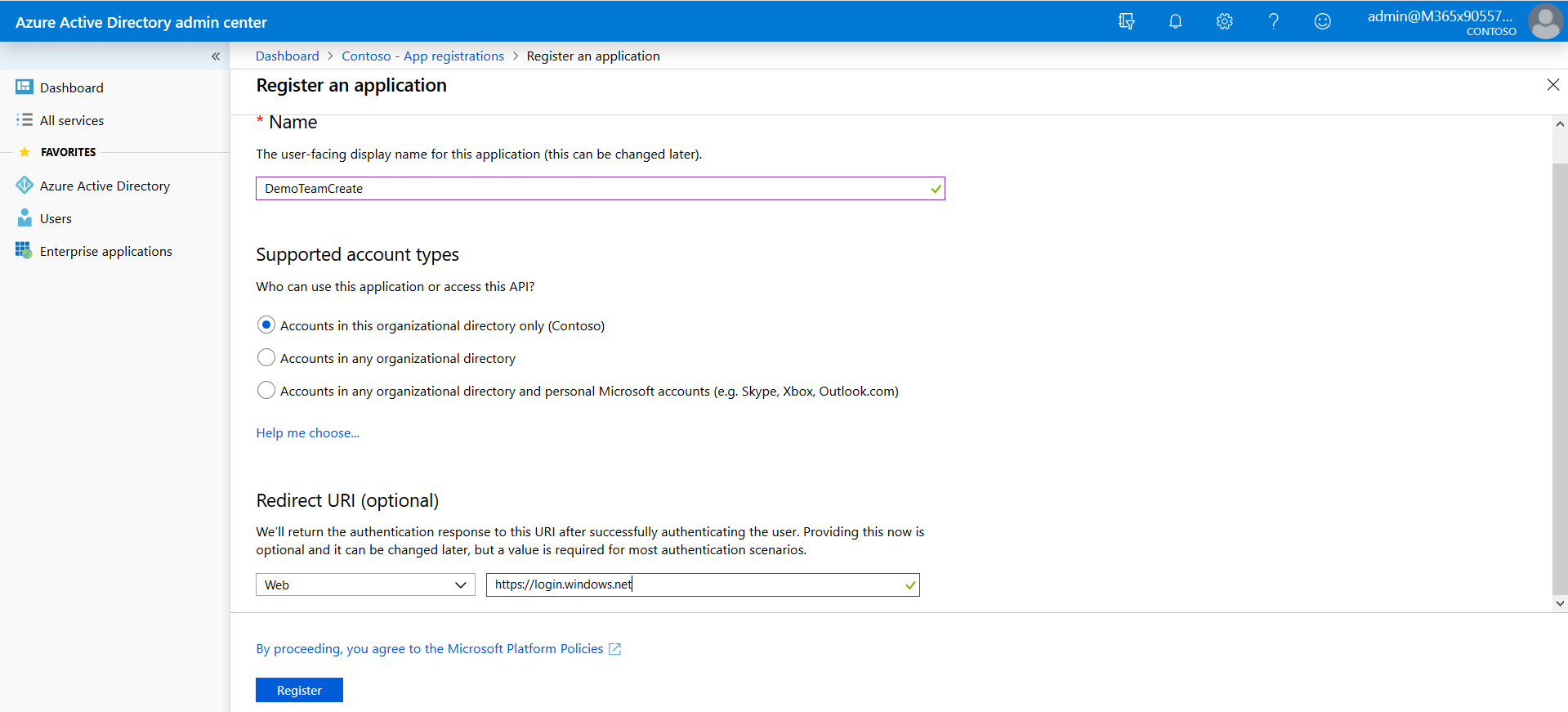
<https://admin.microsoft.com>

1. Navigate to **Admin Centers -> Azure Active Directory**:



1. Navigate to **App registrations** blade and create a new app registration:

*Note: The guide is written around the new App registration interface. Please use this.*

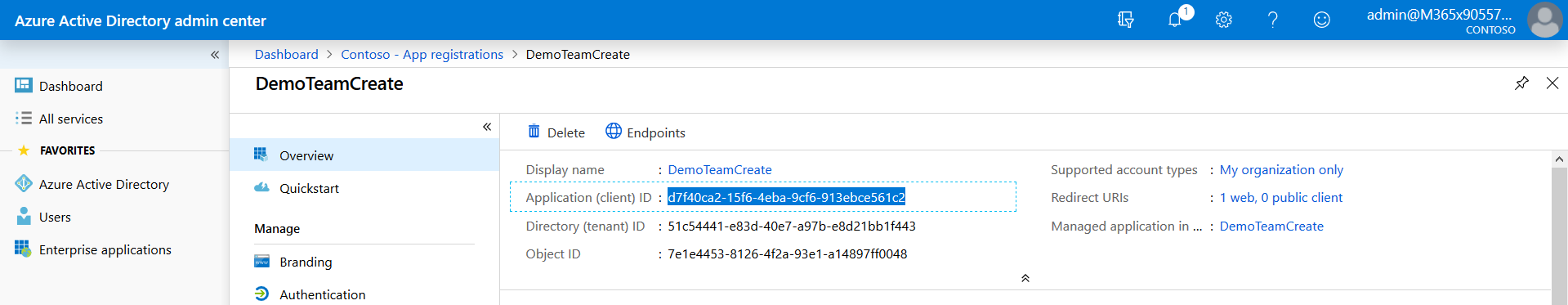


1. Specify the following App registration settings:

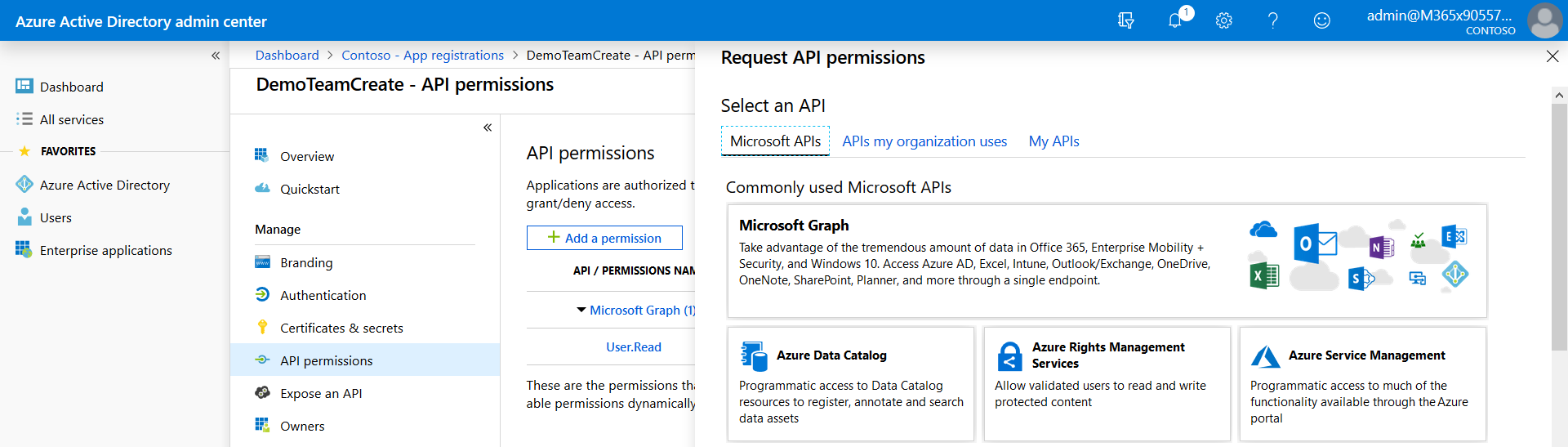
**Name:** DemoTeamCreate

**Redirect URL:** any valid “dummy” URL can be used as a placeholder. We will create the real one later: Use <https://login.windows.net>

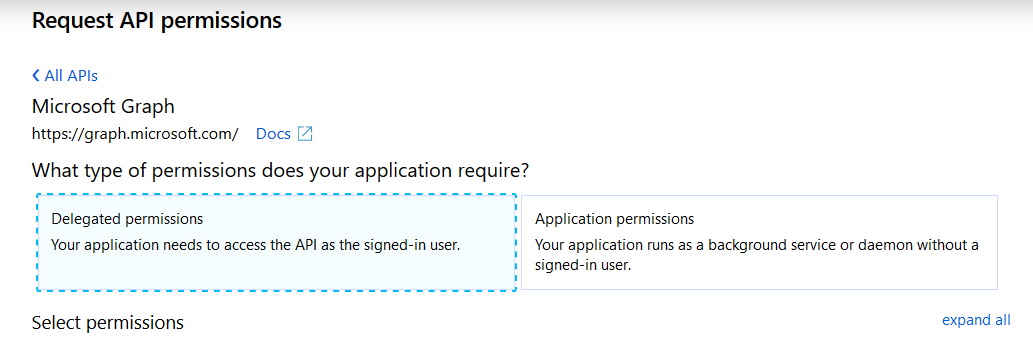
1. You will see the blade that show your App registration settings, **Application ID** (also called client Id) is your application registration identifier, **copy this value to your notes** - you will need it later:



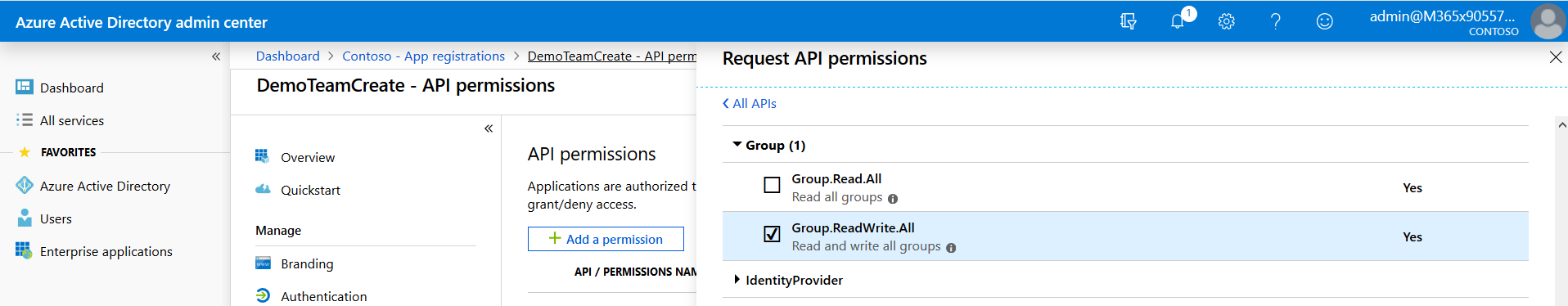
1. Click **View API Permissions**
2. Select Microsoft Graph



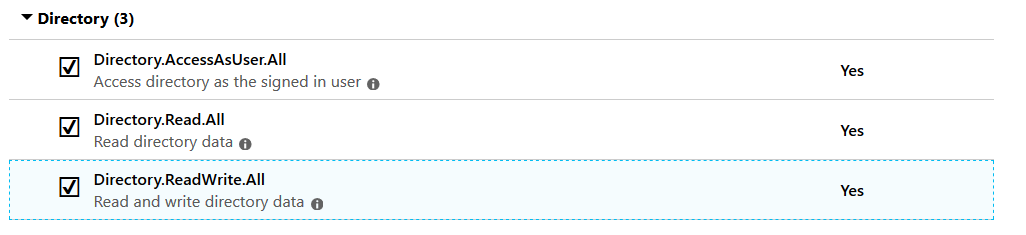
1. Our application will need permissions to utilize Graph API to create Team objects, click **Add Delegated Permissions** and select **Microsoft Graph**:



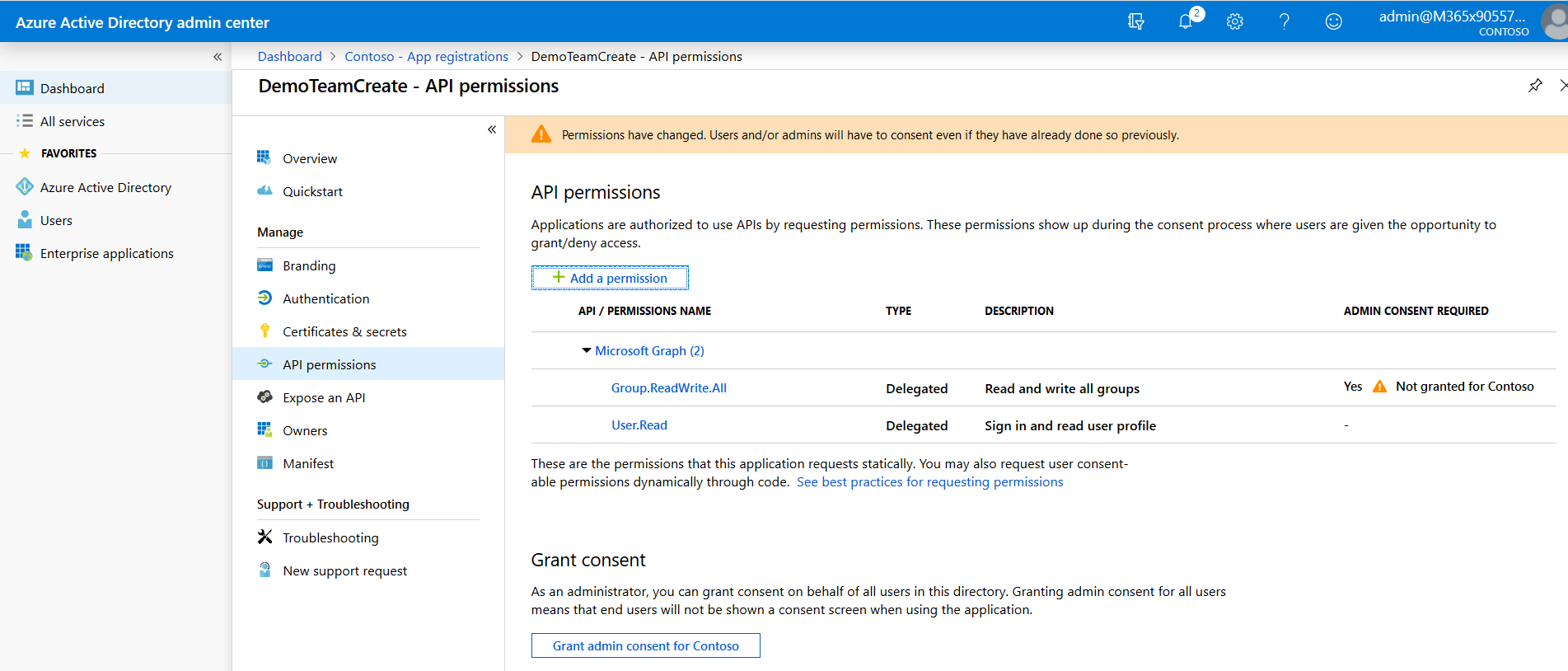
1. Select **Read and Write all groups** in the **Delegated permissions** section:



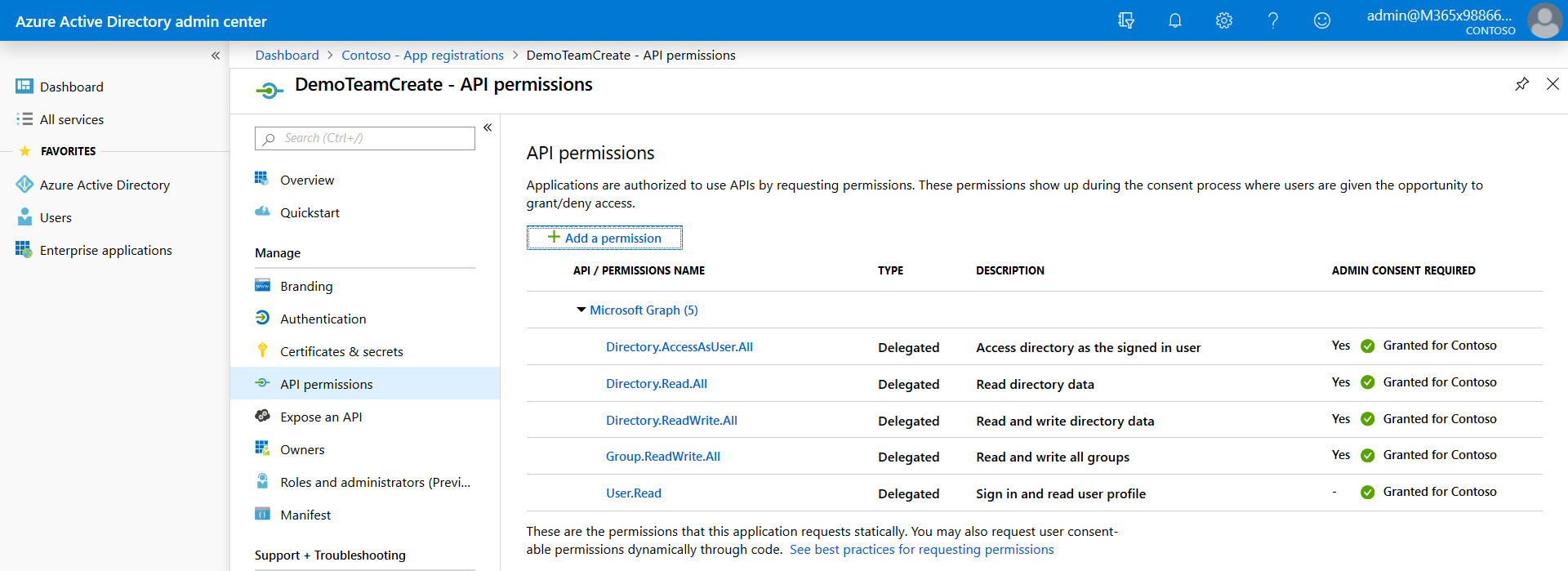
1. Then add the following Directory permissions.



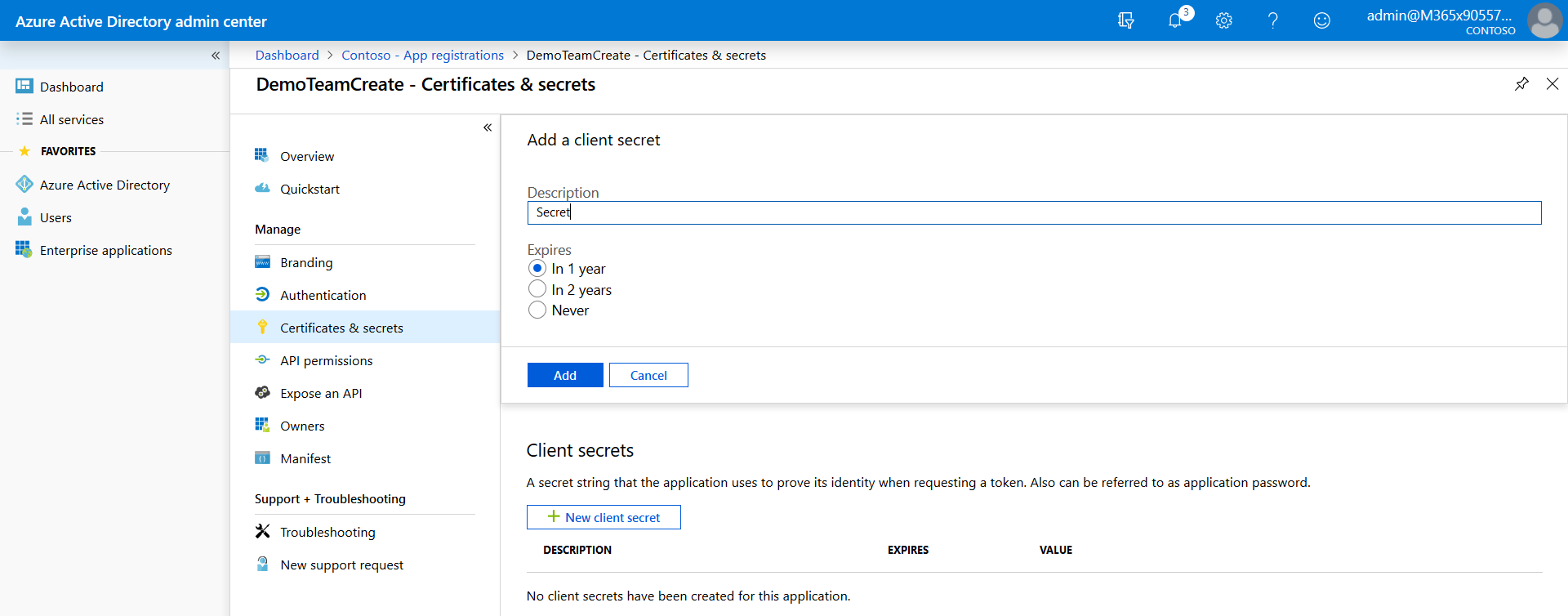
1. Click **Grant admin consent for Contoso** to grant permissions to your application:



1. Your API permissions should match the screenshot provided below:



1. The next thing that we need is a secret key for this App registration. Navigate to **Certificates and Secrets** blade, name the key by typing in the Description box, select a 1-year expiry and **Add** it:



1. **Copy the key value** to your notes alongside with Application Id, we will need it later.
2. Don’t close this browser window, we’ll be back soon.

# 2 Create a Custom Connector

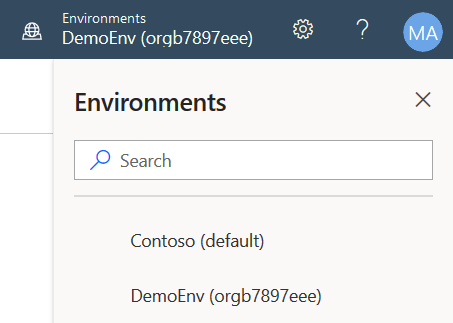
1. A **Custom connector** will allow you to wrap an API call into a Flow action block and reuse it later in your environment. We will create a connector that calls the Microsoft Graph API and creates a Microsoft Team object.
2. Navigate to Office 365 home page as an Administrator and then navigate to **Flow** app:

<https://office.com>

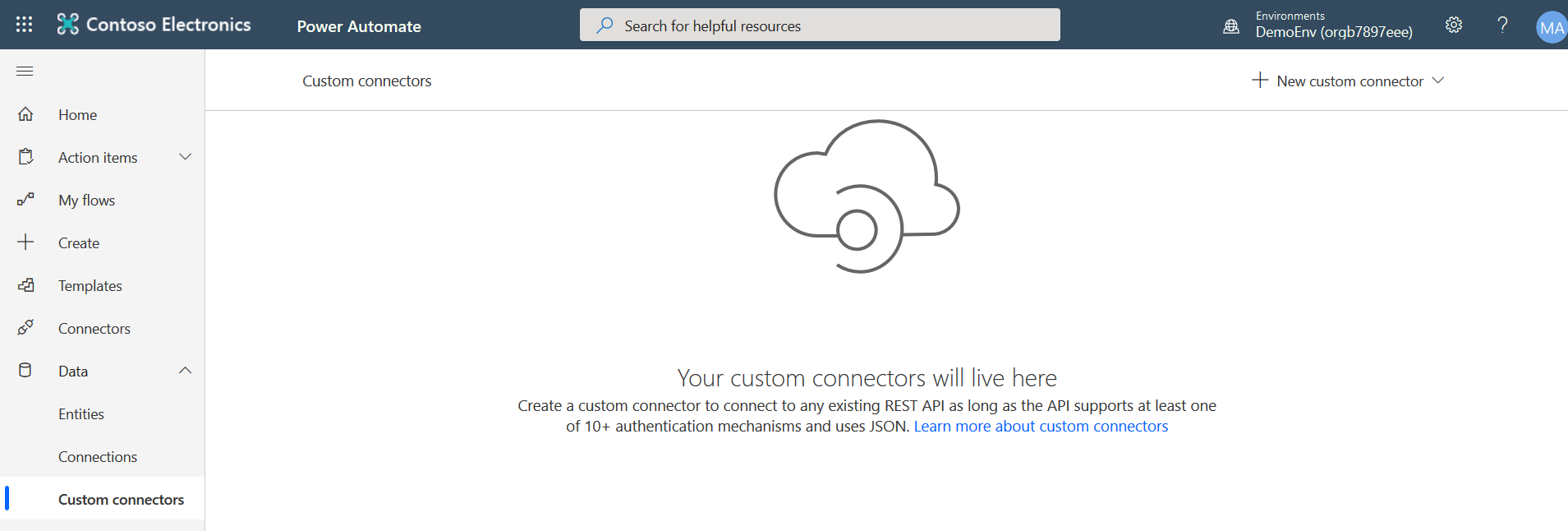
A screenshot of a cell phone

Description generated with high confidence

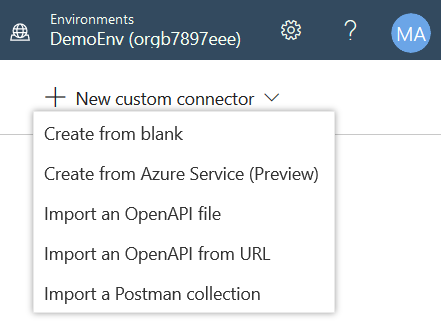
1. Switch to your **DemoEnv** environment:



1. Click the **Data tab** select **Custom Connectors**.



1. Then click **New custom connector** **-> Create from blank**:



1. Name it DemoTeamCreate

A screenshot of a cell phone

Description generated with very high confidence

1. On the first step you need to name your custom connector, provide description and specify the following settings:

**Scheme:** HTTPS

**Host:** graph.microsoft.com

**Base URL:** /

A screenshot of a cell phone

Description generated with very high confidence

1. Proceed to the next step (Press the Security Link).
2. At this step we configure how this custom connector gets authorization to use **Graph API**.   
   Choose **OAuth2** and specify the following settings:

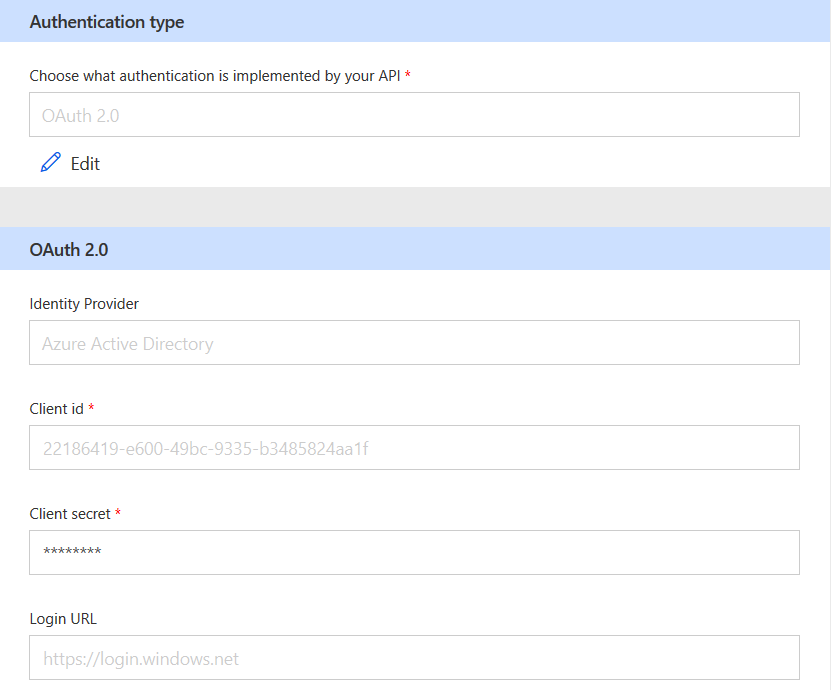
**Identity Provider:** Azure Active Directory

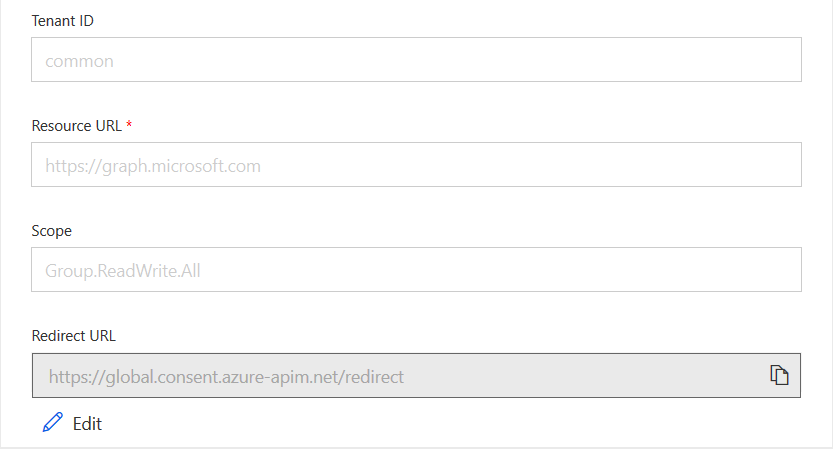
**Client Id:** *<Application Id from the previous>*

**Client secret**: *<Application secret key from the previous task>*

**Resource URL:**<https://graph.microsoft.com>

**Scope:** Group.ReadWrite.All





1. Proceed to the next step. Select Definition.
2. Create a New Action and specify basic settings for it:

**Summary:** DemoTeamCreate

**Description:** Connector for a team creation

**Operation ID:** DemoTeamCreate

Example:

A screenshot of a social media post

Description generated with very high confidence

1. Navigate to Graph API reference in a differentbrowser tab and review it:

<https://docs.microsoft.com/en-gb/graph/api/team-post?view=graph-rest-beta>

We are going to use Example 2 that will result in a new team being created, including a pre-defined owner.

1. In Request section click **Import from sample** (refer to screenshot below)and specify the following:

*POST https://graph.microsoft.com/beta/teams*

*Content-Type: application/json*

*{*

*"template@odata.bind": "https://graph.microsoft.com/beta/teamsTemplates('standard')",*

*"displayName": "My Sample Team",*

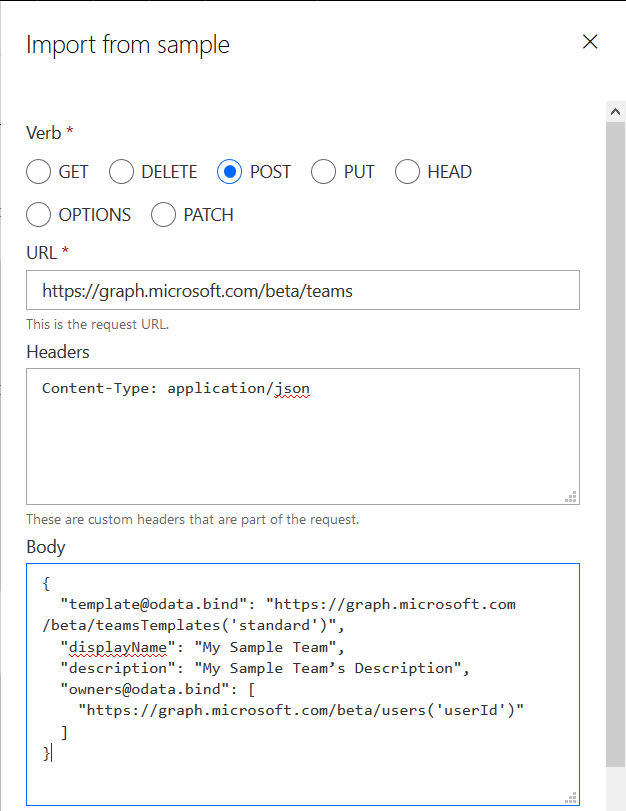
*"description": "My Sample Team’s Description",*

*"owners@odata.bind": [*

*"https://graph.microsoft.com/beta/users('userId')"*

*]*

*}*



1. Click **Import**
2. In **Response** session, click **default** response and Import from Sample, copy response headers and body from the **Graph API reference** below and click **Import:**

*HTTP/1.1 202 Accepted*

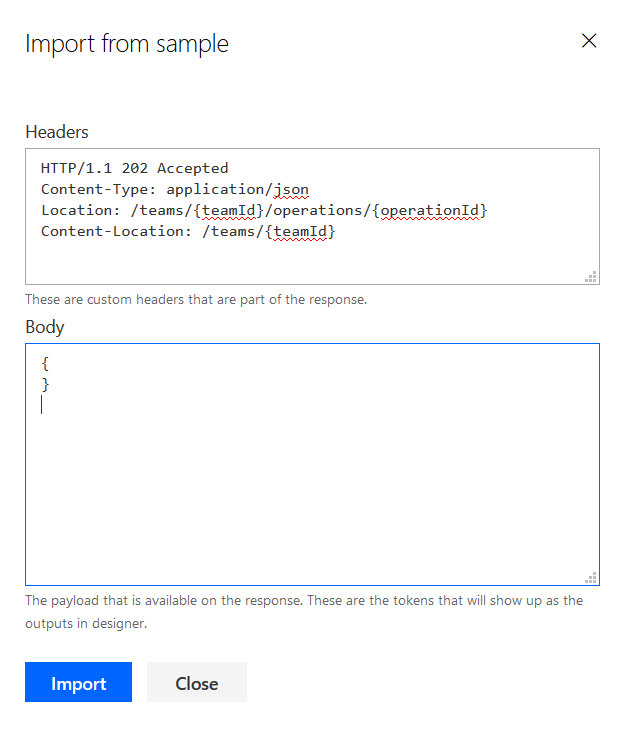
*Content-Type: application/json*

*Location /teams/{teamId}/operations/{operationId}*

*Content-Location: /teams/{teamId}*

*{*

*}*





Note: In the documentation you might see Location**:** in the Header information and not Location (without :), see the above screenshot. During the lab creation is was found that the API call returns Location without a (:) and the Teams Object ID we need for the optional part of this lab is buried in the returned syntax. <https://docs.microsoft.com/en-gb/graph/api/team-post?view=graph-rest-beta>

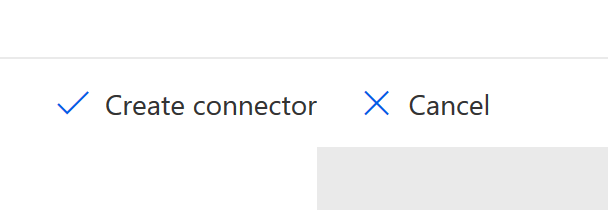
Things may have changed We might need to verify this before we build the second (optional) connector.

1. Check the **Validation** section and get sure that we’re ready to go!

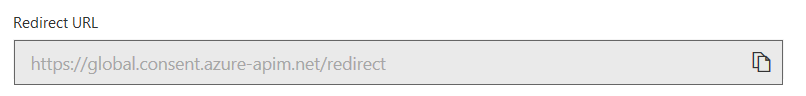
A screenshot of a cell phone

Description generated with very high confidence

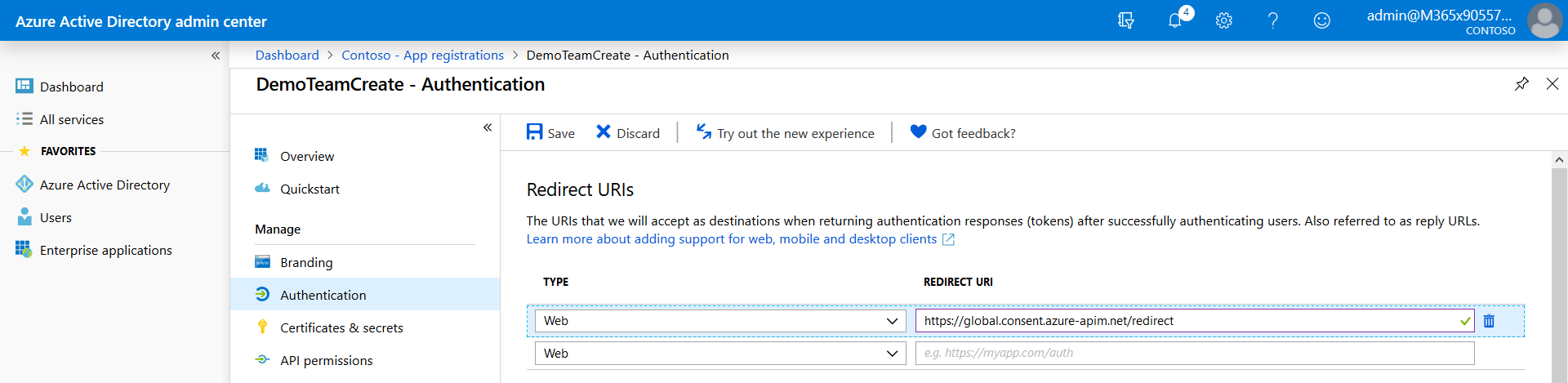
1. Now we are ready to create a custom connector, click **Create connector** in the top right corner:



1. Navigate back to custom connector’s **Security** tab and copy the generated **Redirect URL**:



1. In **Azure AD portal** navigate to **App registration** that we use and open the **Authentication** blade. Then copy **Redirect URL** from connector as a **Redirect URL** in App registration and **Save**. Delete the dummy entry. Then save the entry.

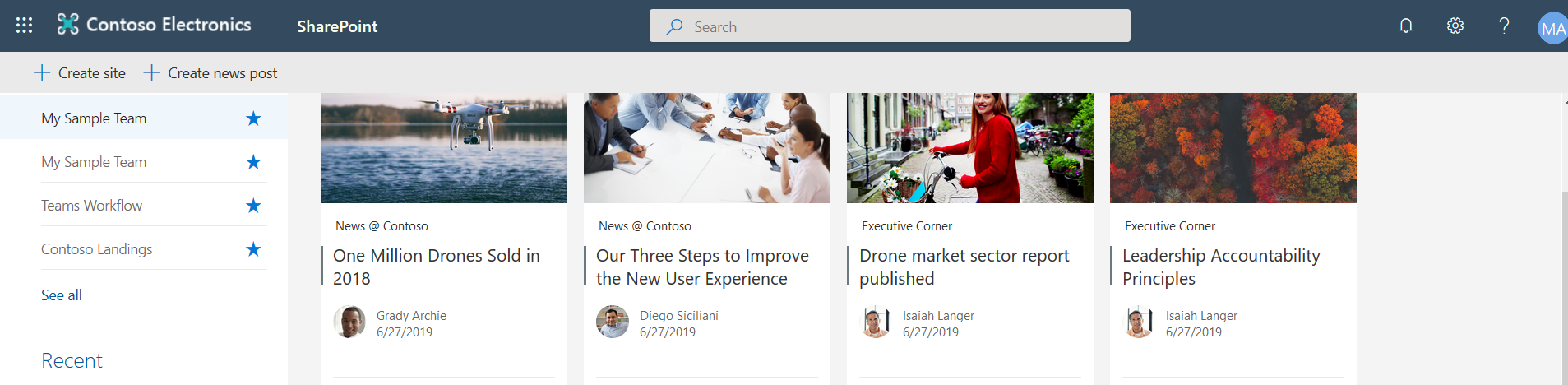


# Create a list in SharePoint Online

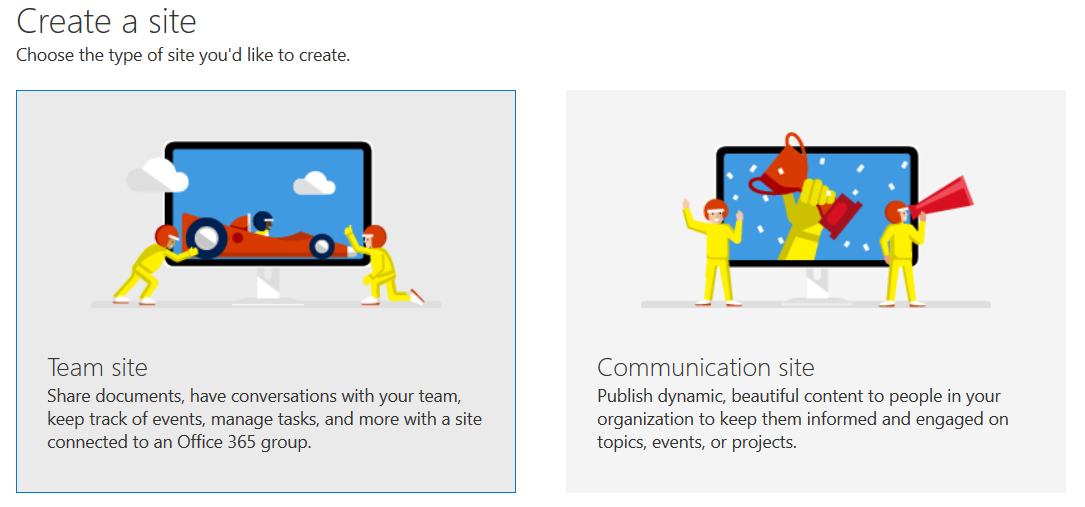
1. Let’s create a storage for our Teams requests.   
   Navigate to **SharePoint Online home page** (log in as a trial tenant admin if necessary):

<https://M365x988662.sharepoint.com/_layouts/15/sharepoint.aspx>

Note: Don’t take any new feature tours.



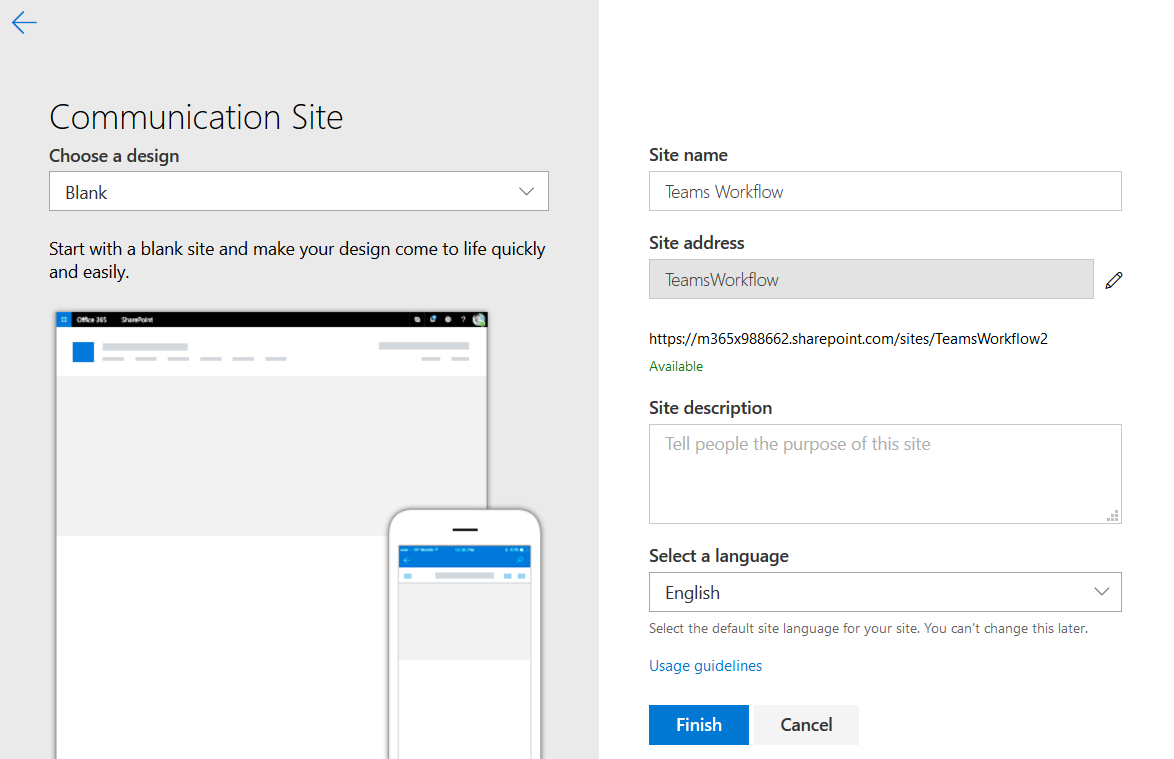
1. Click **Create site** and choose **Communication site** option:



1. Specify the new **Communication site’s** properties and choose a default design option:

**Site name:** Teams Workflow

**Choose a design:** Blank



1. Wait till the site is provisioned, then navigate to (if you’re not redirected):

<https://m365xXXXXXX.sharepoint.com/sites/TeamsWorkflow>

1. For the test purposes we need to make this site accessible by the other users.  
   Click **Share site** link in the top right corner:

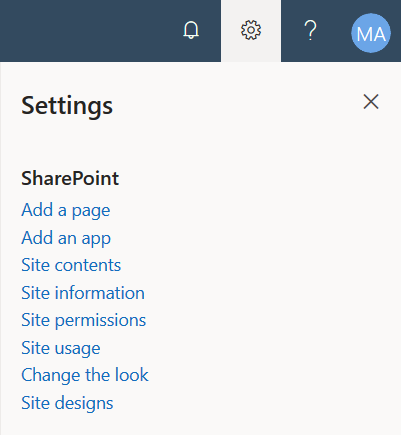
A screenshot of a cell phone

Description generated with very high confidence

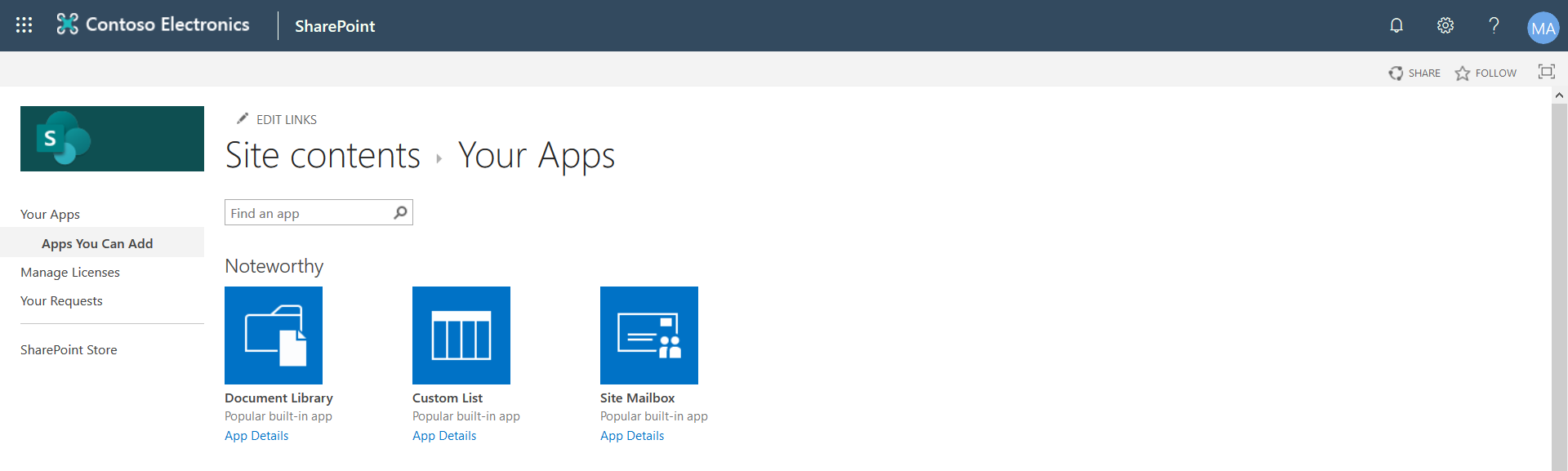
1. In **Share ‘Team Requests’ form** specify **Alex Wilbur** then press the **Share** button:



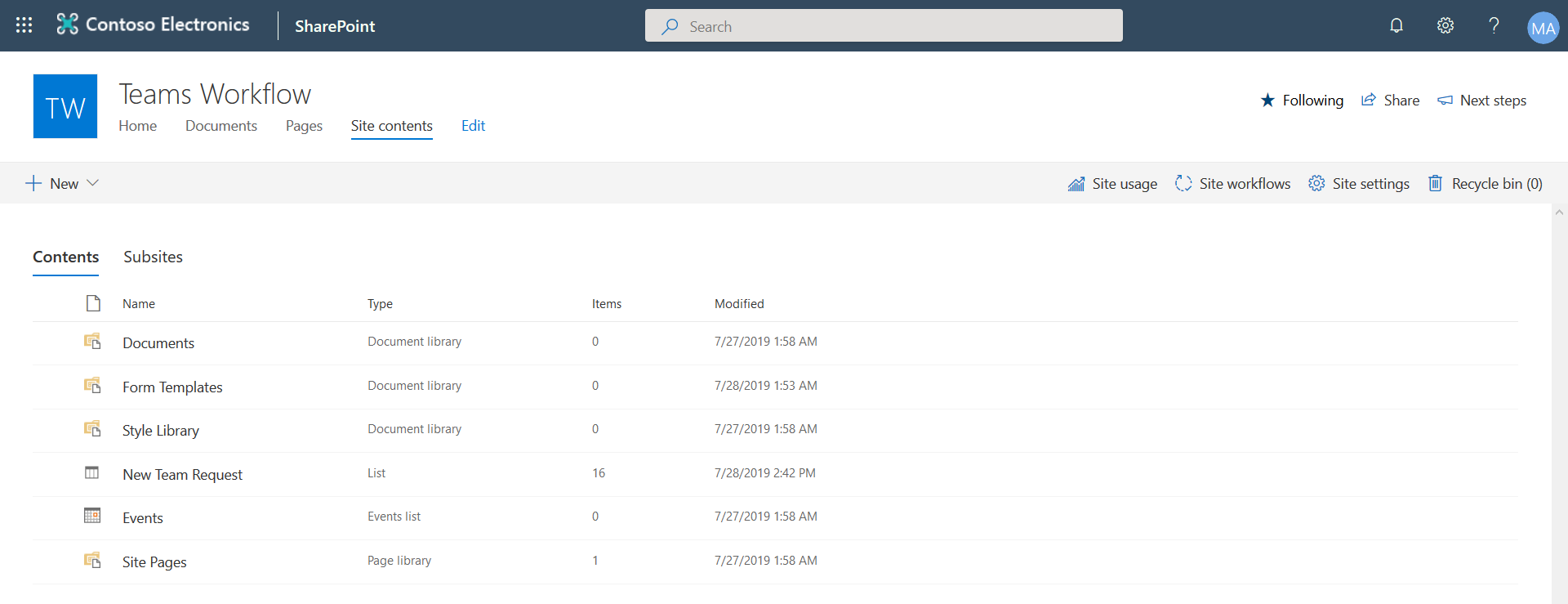
1. Now we need to create a **list** for Team requests. Click on the **gear icon** in the top right corner and select **Add an app** option:



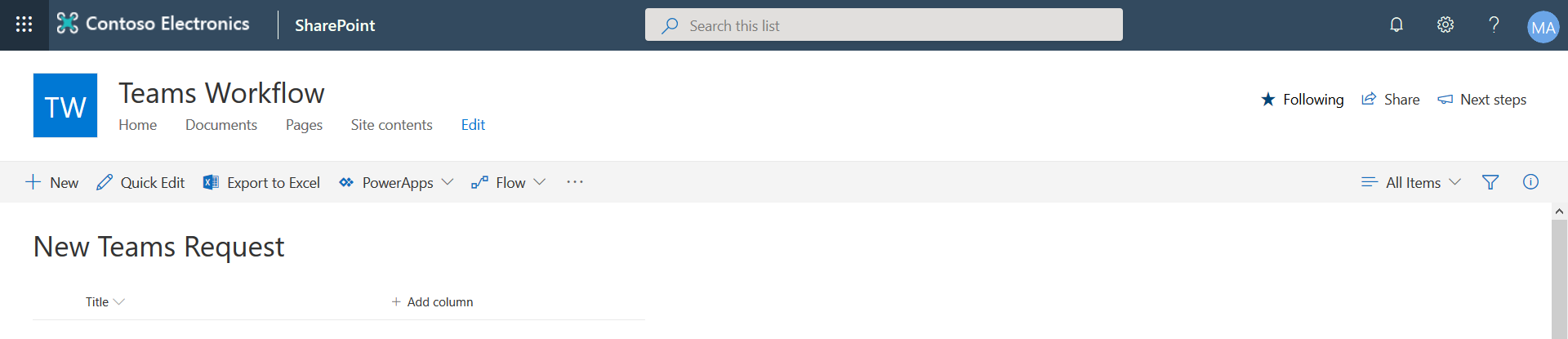
1. Choose **Custom list** template and specify list name (e.g. New Team Request):



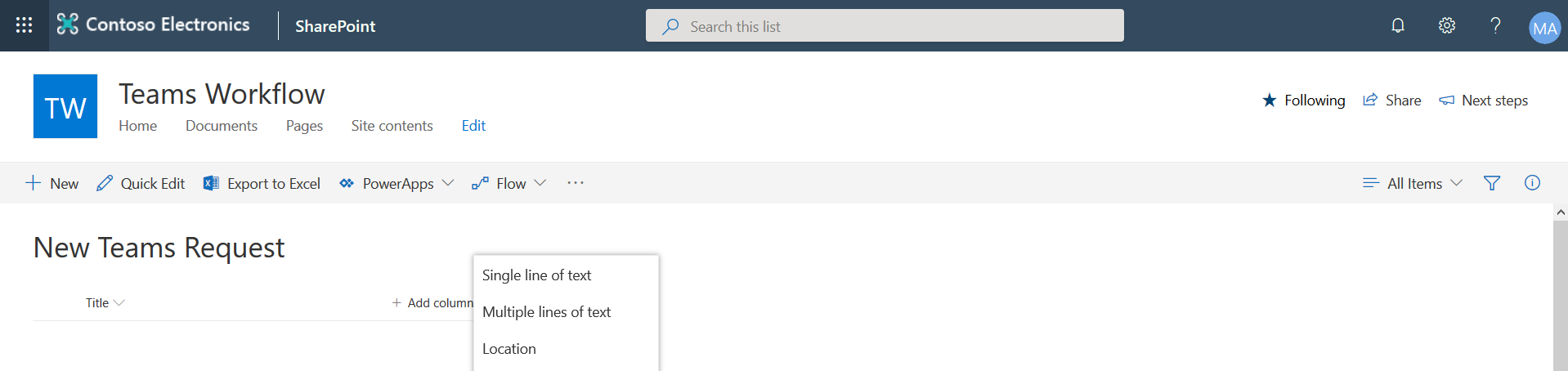
1. You can find a new list created in **Site contents**:



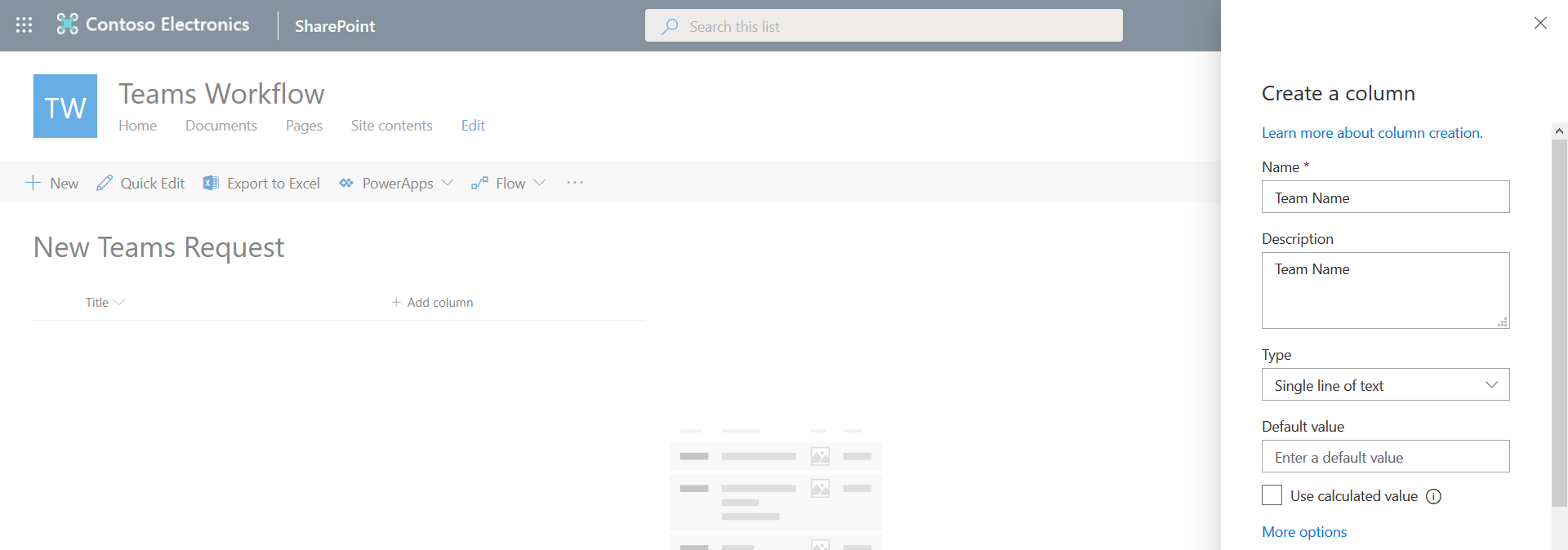
1. Click the **New** **Team Requests** list to navigate to a default list view page.



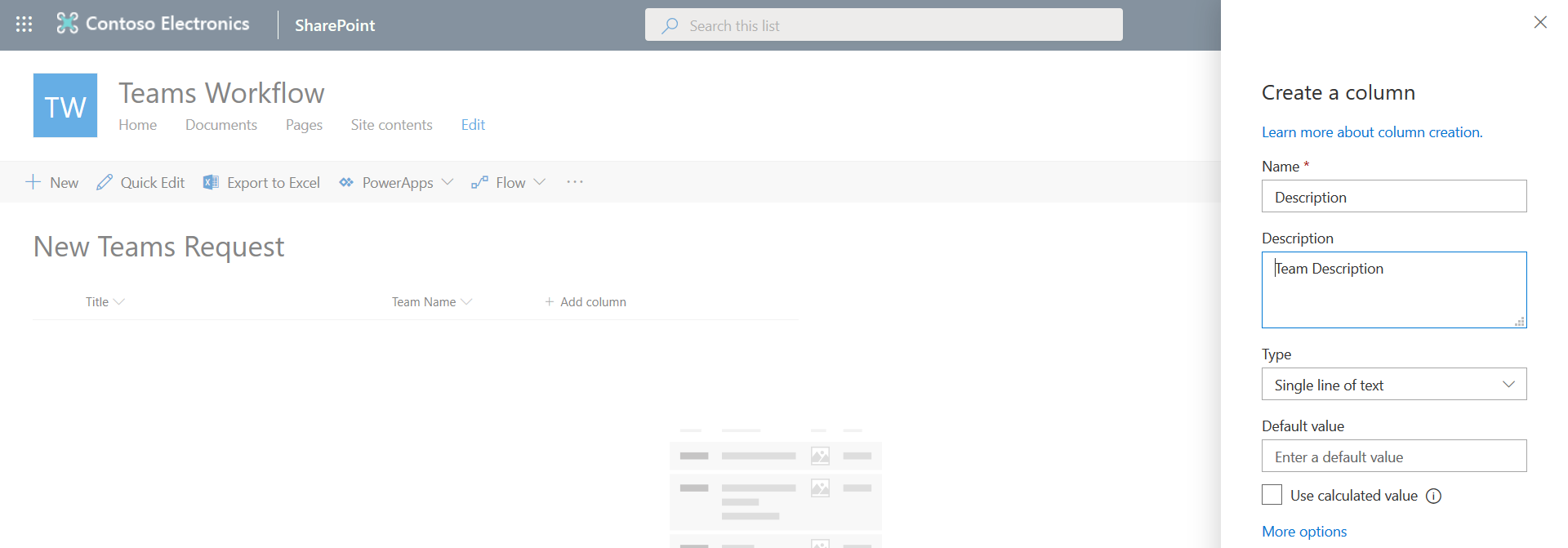
1. Start by Adding a new column. Press the Add column button.



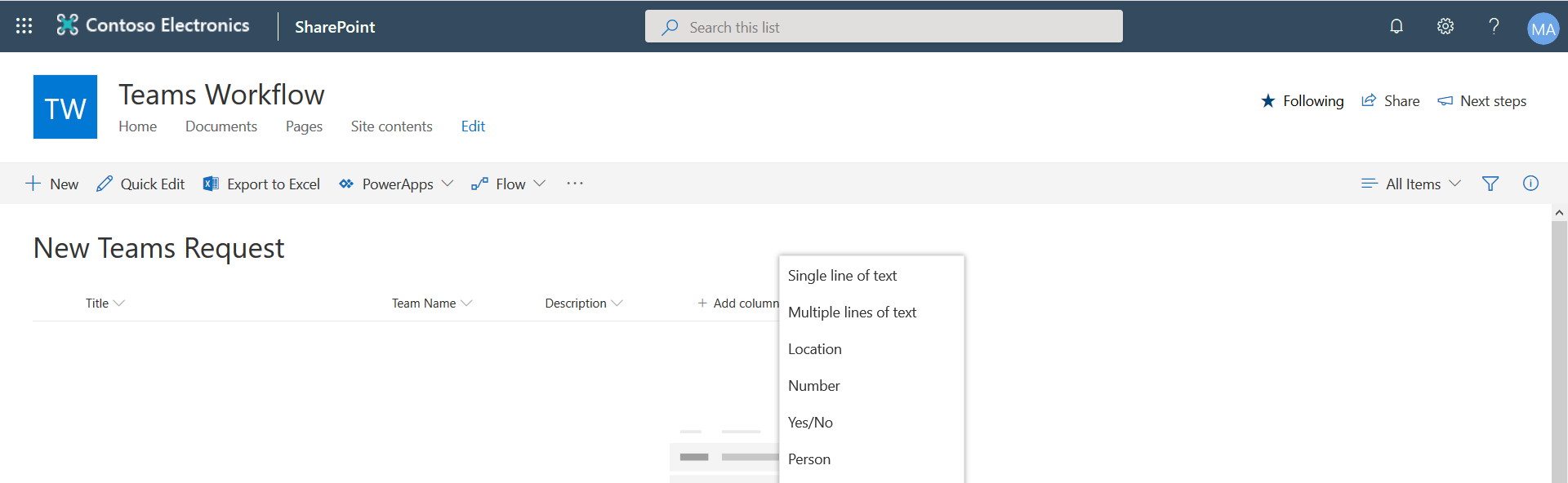
1. Select Single line of text and call the Column Team Name and give it a Description.



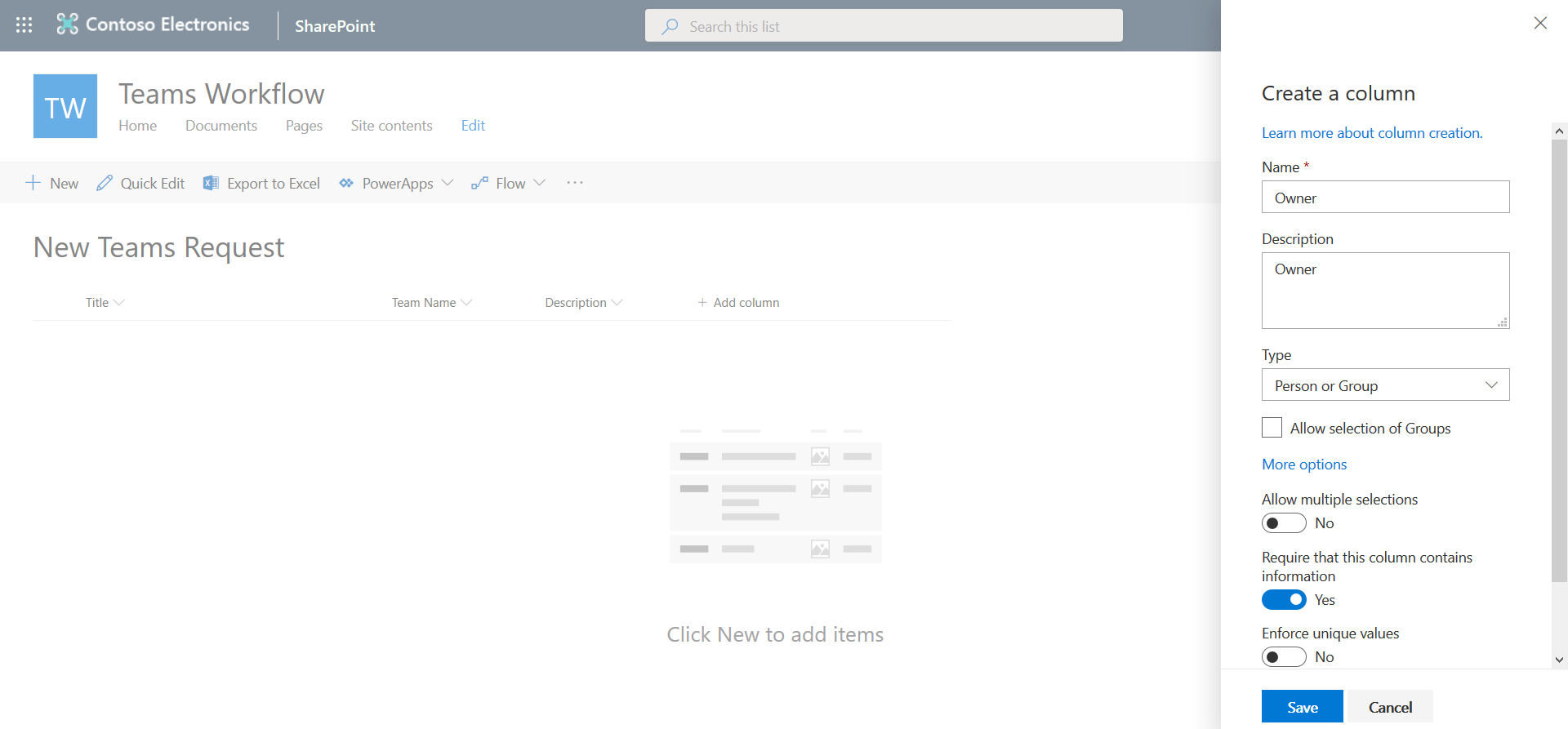
1. Under More options, change the Require this column contains information to yes. Then press save.
2. Add another column. Type = Single line of text. Call it Description.



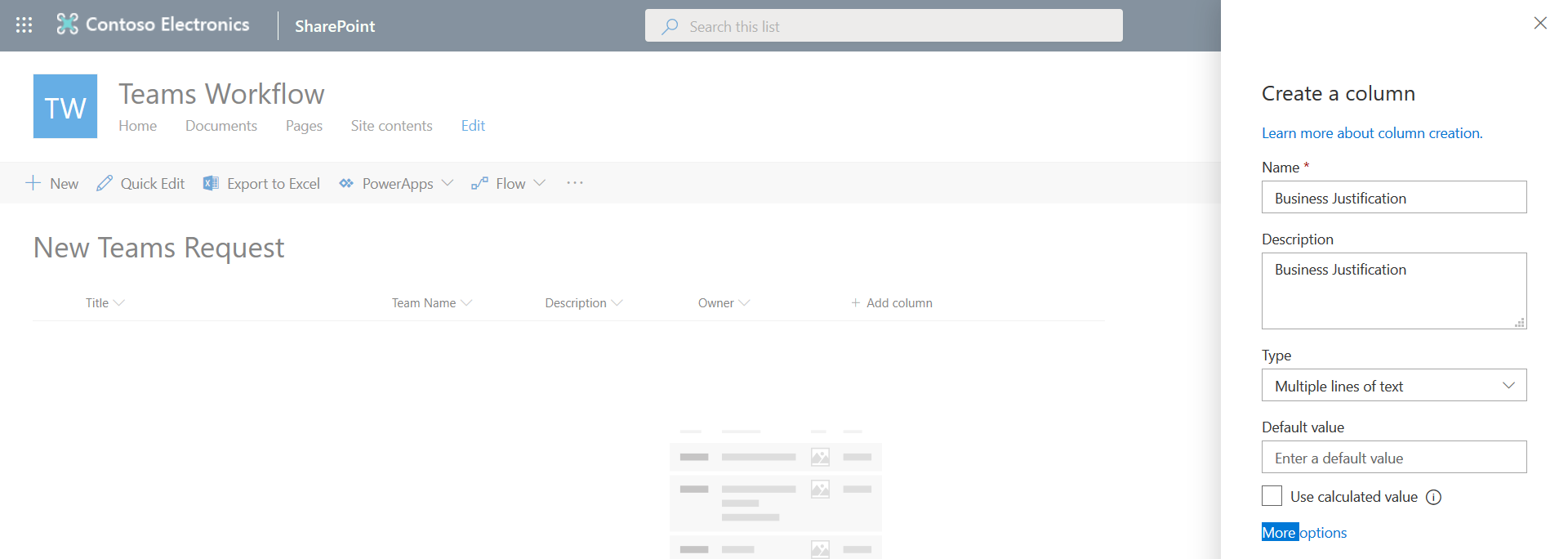
1. Under More options, change the Require this column contains information to yes. Then press save.
2. Add the next column. This time select Type = Person



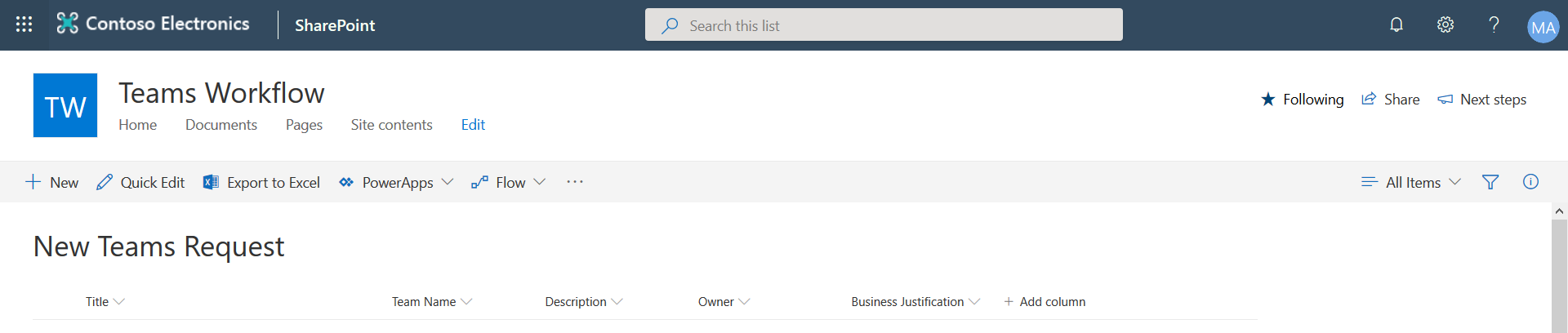
1. Fill in the Column details as shown below, then Save it.



1. Add a column Type =Multiple lines of text. Call it Business Justification and give it a Description.



1. Under More options, change the Require this column contains information to yes. Then press save.
2. Your List should look like the following screenshot.



# Create an approval Flow

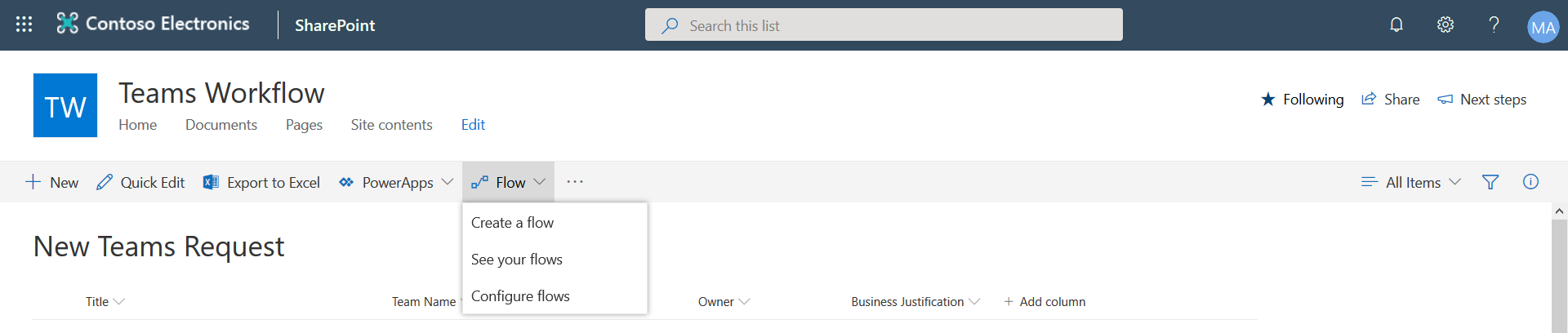
Note: The final flow (depending on how much of the lab you decide to complete) will look something like the following:

A screenshot of a social media post

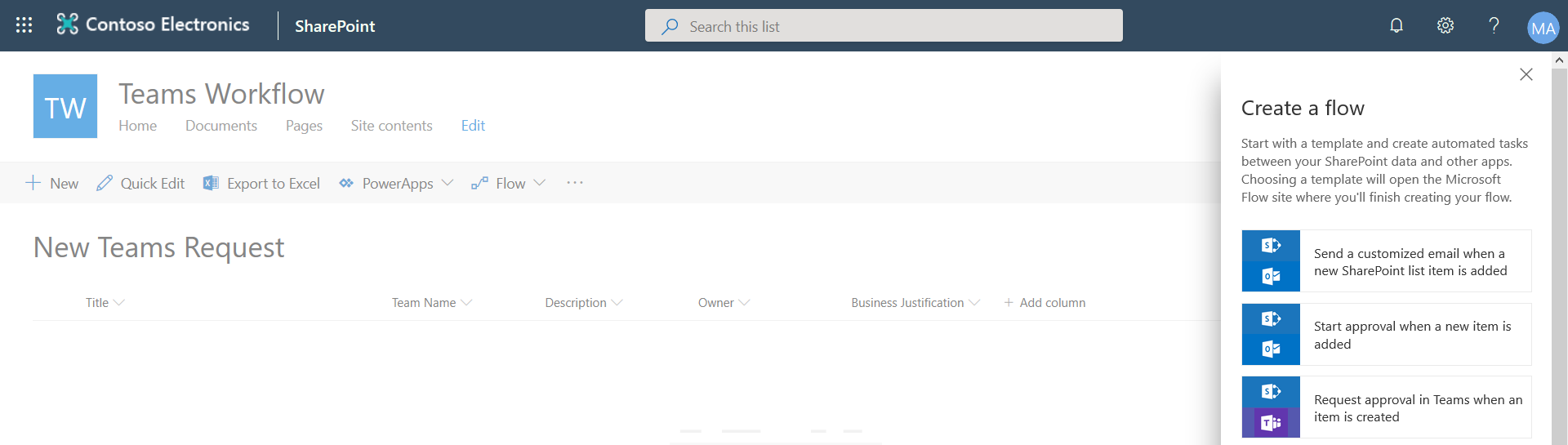
Description automatically generated

1. Let’s create a simple approval flow based on the existing list structure.

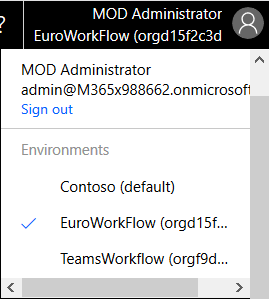
From your SharePoint list, go to **Flow -> Create a flow**:

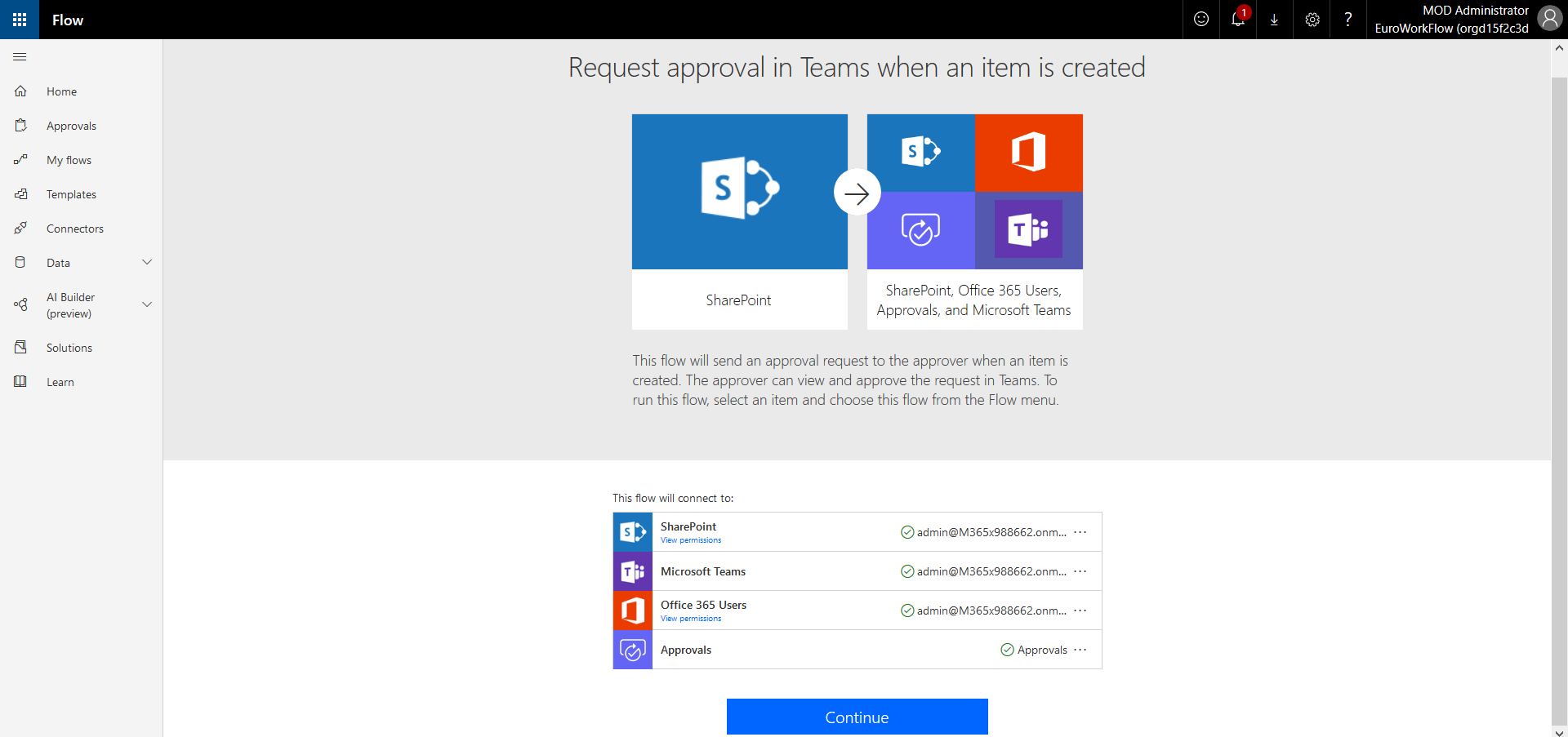


1. To simplify the creation of the Flow, we can utilise an existing approval workflow template. Choose **Request approval in Microsoft Teams when a SharePoint item is created.**



1. Important! Make sure that you are in **Demo Env** environment. You can switch to the Demo Env from the top right-hand side of the screen. See below.

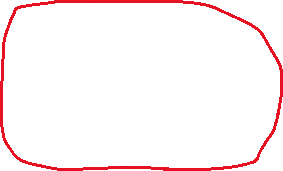




1. Press Continue. Then check if service connections are “green”.
2. On Flow editor page you will see the following. On the right-hand side of the template delete “Send an email” and “Terminate” items. We don’t need this part of the template for the lab. Fill in the different items of the flow as described below.

A screenshot of a social media post

Description automatically generated

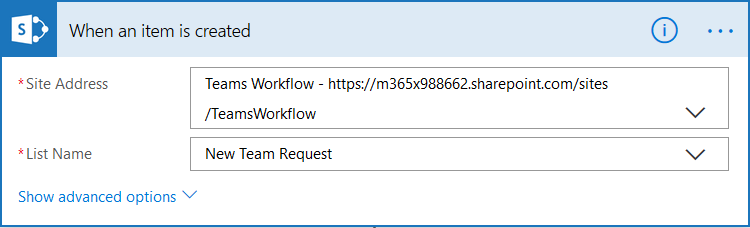


1. After you have delated the above, your Flow should look like the screenshot below.

A screenshot of a cell phone

Description automatically generated

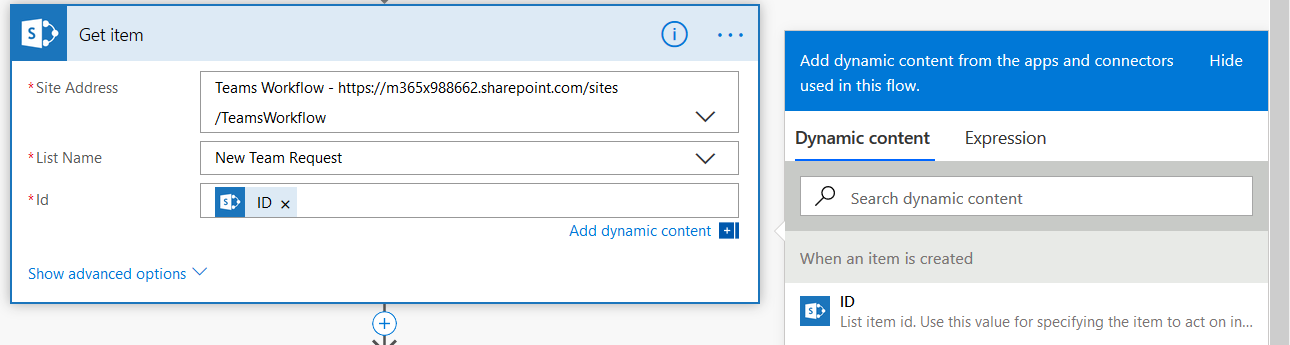
1. Now start to fill in the individual components of the flow as described below.
2. Start with the “When an item is created” entity. Refer to the screenshot below.



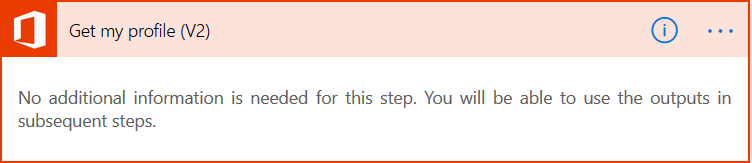
1. Get item



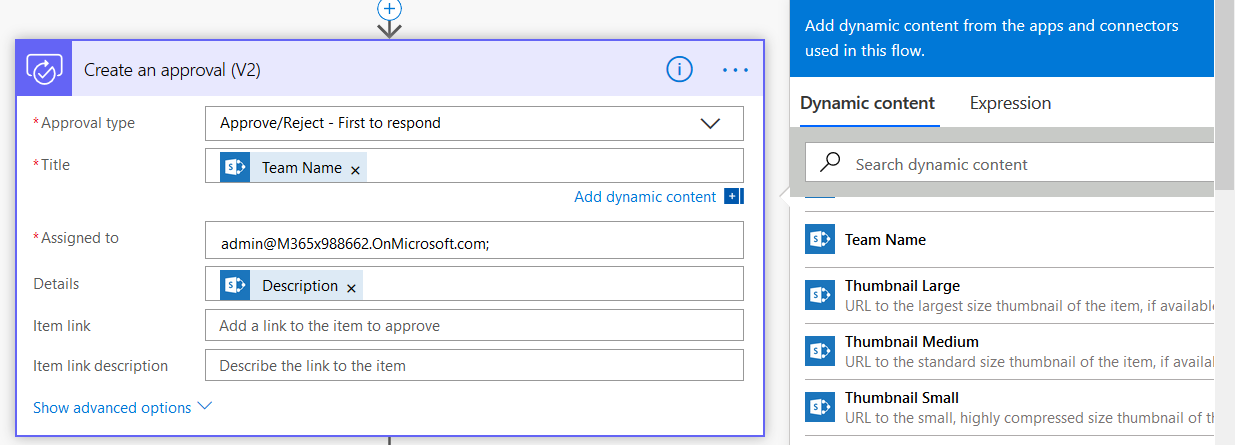
Note: Id is the List item id. It can be found from the dynamic content option. You will use Dynamic content elements to help you fill out the Flow items. If you are new to Flow you will need to spend some time becoming familiar with the configuration interface.



1. Get my profile

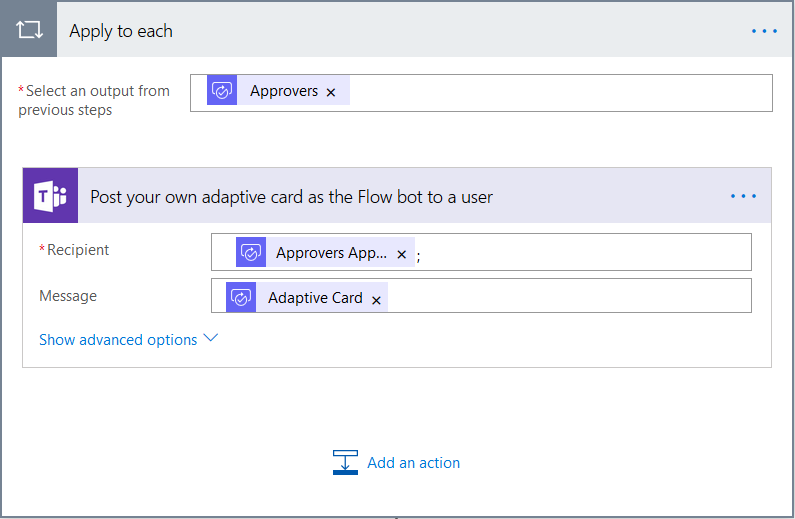


1. Create an approval (V2)

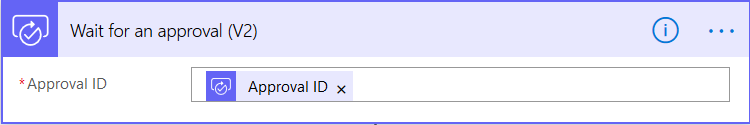


Note: Use Dynamic content to find and add the Team Name and Description. These map on to the entries in Team Name and Description columns in your SharePoint list. The assigned to person (use the Admin) will approve or reject the request.

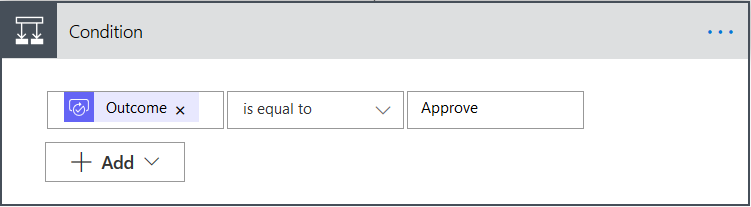
1. Apply to each



1. Wait for an approval (V2)



1. Condition



Note: This is the If/Else statement that will branch the Flow based upon whether the Flow if Approved/Rejected.

1. It used to be possible under If Yes to press a + sign and add a new action. This has recently changed. It is now necessary to press the + sign below the “Apply to each response to approve” element. See below.

A screenshot of a social media post

Description automatically generated



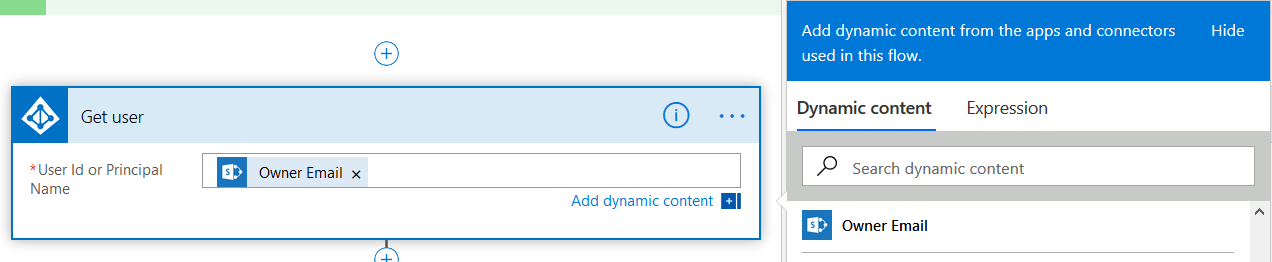
1. This brings up the Choose an action box. Type in Azure to surface the Azure actions. Click the Azure icon and then find and select Get user.

A screenshot of a social media post

Description automatically generated



1. You will end up with the Get user action inserted into your Flow. Then use the dynamic content to add the Owner email content. This is the email address of the Team owner you specified in the SharePoint list.



Note: The Get user action will return the Object id of the Team owner. We need this for the next step.

Note: Your Flow should now look like the screenshot below.

A screenshot of a social media post

Description automatically generated

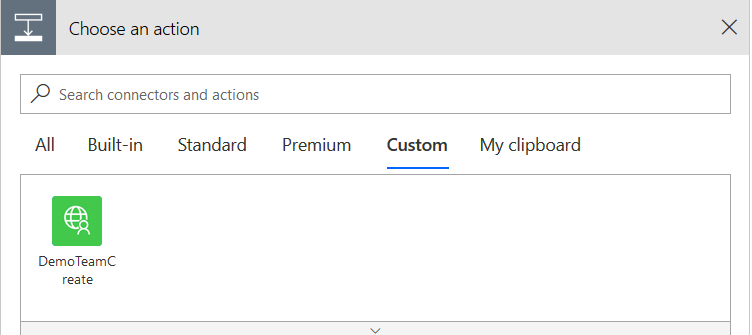


The “Apply to each response to approve” element is the last part of the completed Flow and needs to be moved to sit below the “Get user” box. You can easily do this by dragging and dropping the “Apply to each response to approve” element downwards. The key to success is to drop the box between the “Get user” element and the “Add an action” link. The screenshot shows the resulting Flow.

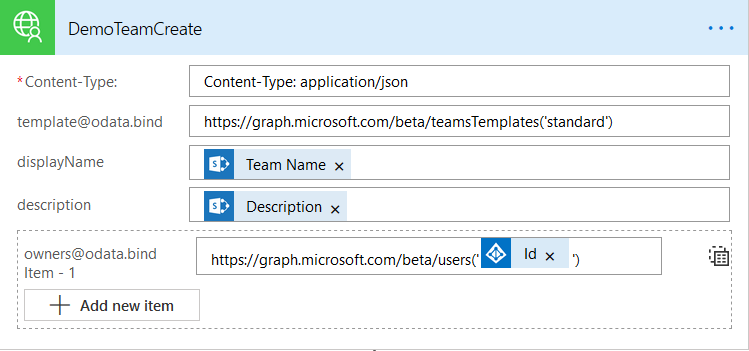
A screenshot of a social media post

Description automatically generated

1. Now add a new Step/Action underneath the Get user action. Select Custom and then click DemoTeam Create. You will need to authenticate. Use your Admin credentials.



1. DemoTeamCreate



Fill out the Custom Connector as shown above (cut and paste content is below):

Content-Type: application/json

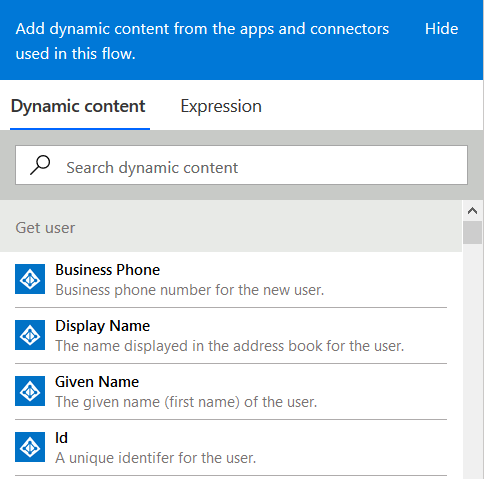
template@odata.bind - <https://graph.microsoft.com/beta/teamsTemplates('standard')>

This is the standard Team definition template. More information can be found here:

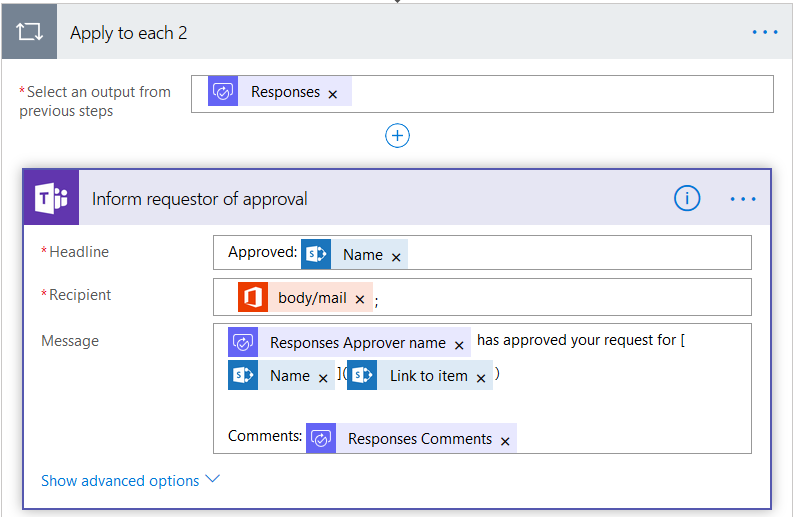
<https://docs.microsoft.com/en-us/MicrosoftTeams/get-started-with-teams-templates>

owners@odata.bind - <https://graph.microsoft.com/beta/users('userId')>

Replace ‘userId’ with dynamic content. In this lab we will use the Id of the Get user action.

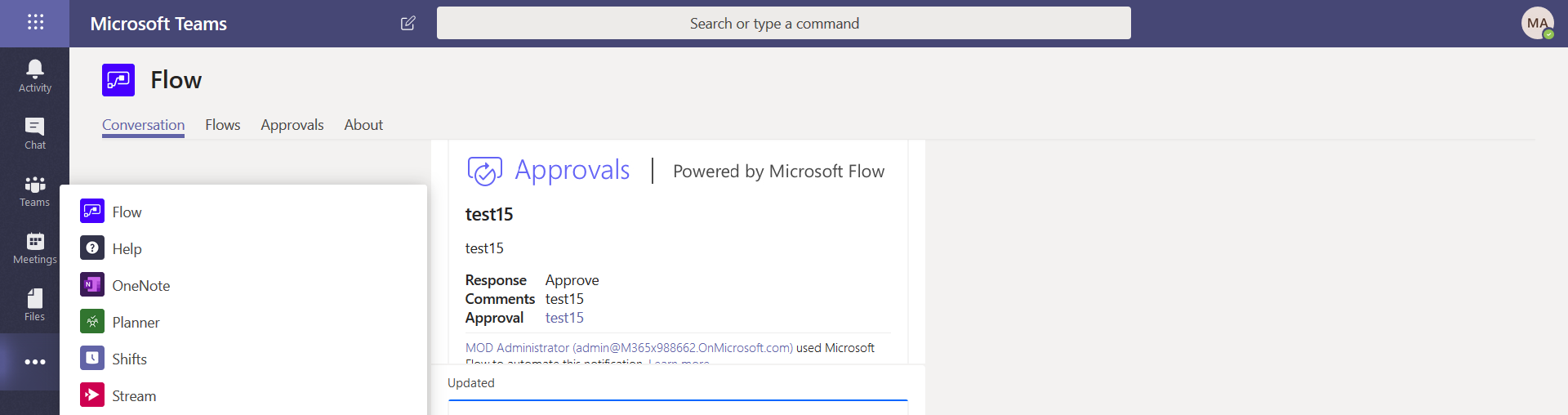


1. Apply to each

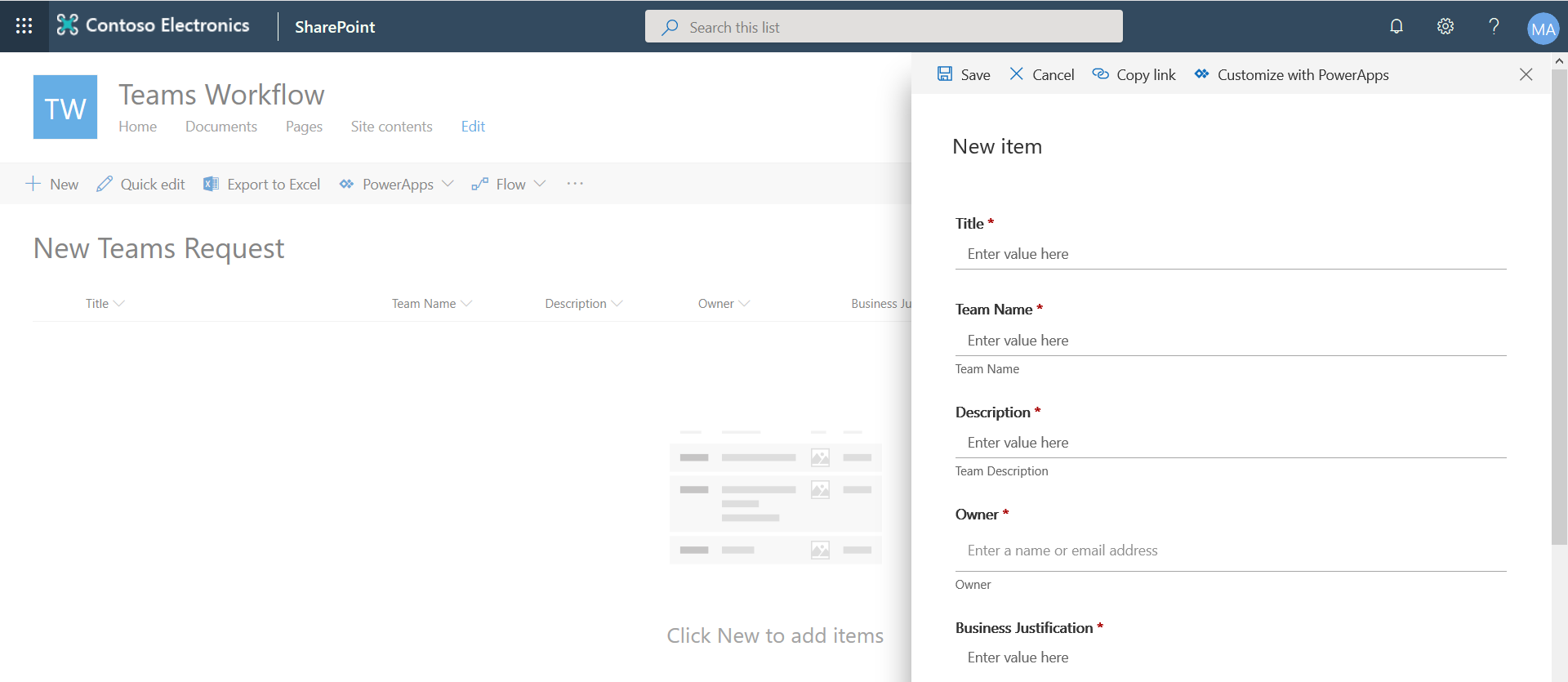


Note: Use Dynamic content as required to fill in the action. This will send a card to Teams to inform the requestor that the Flow has been completed.

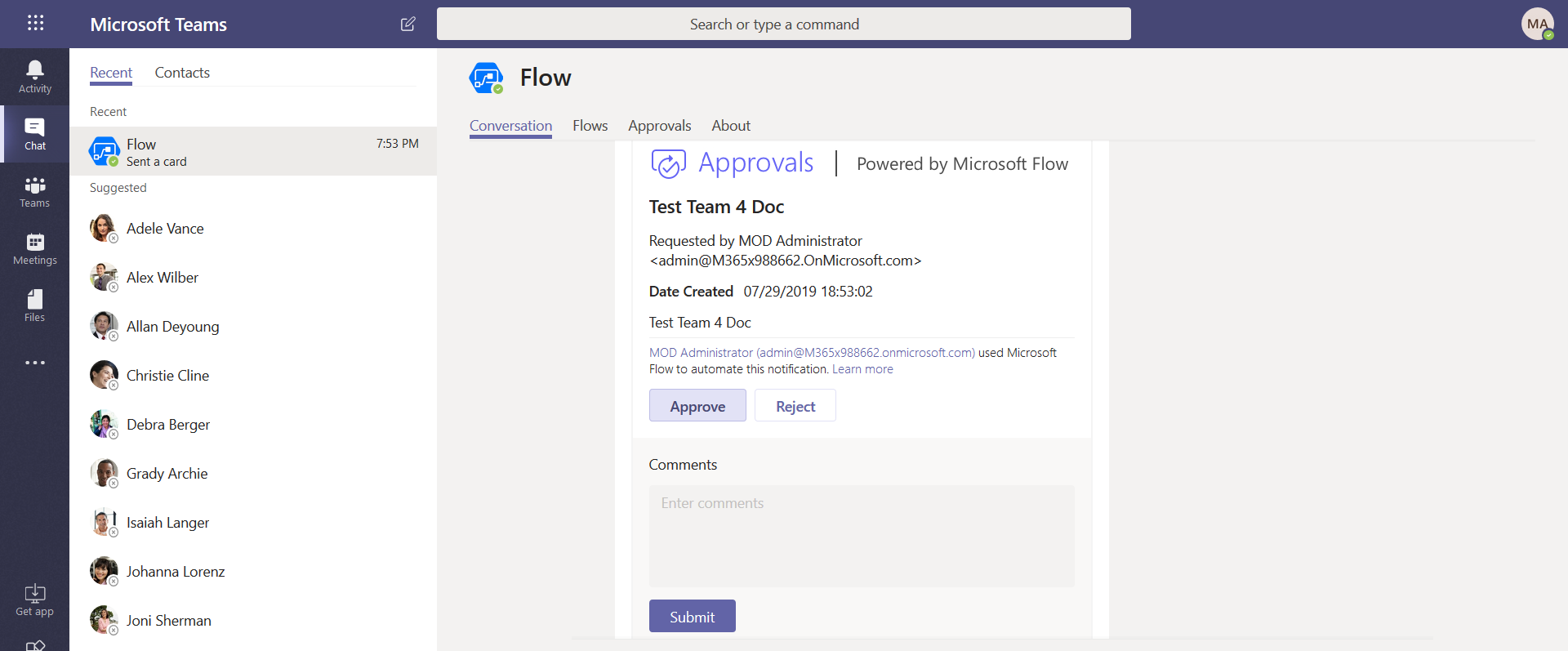
1. Save the Flow.
2. Log onto the Admins Team account and install the Flow as personal app



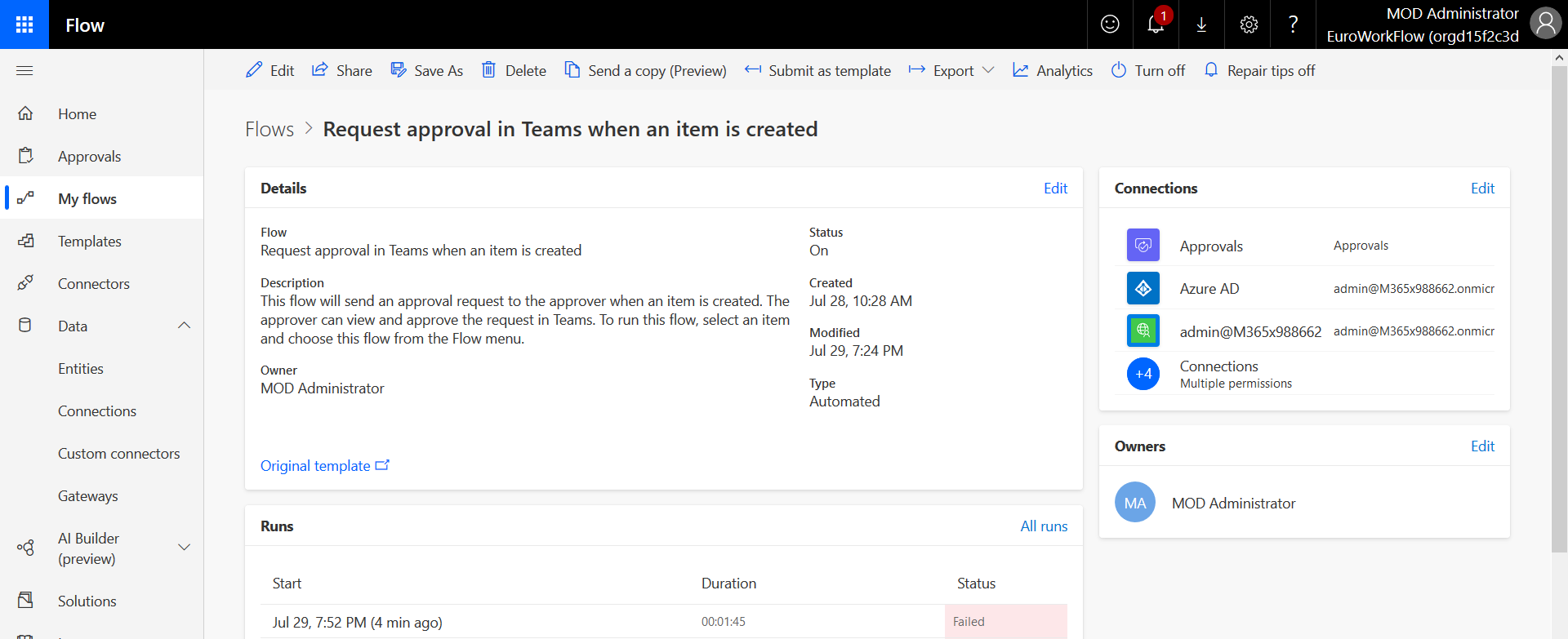
1. Then using the admin account navigate back to the SharePoint list and create a new entry. Add a Team Name, Description, Owner (Adele Vance) and Business Justification.



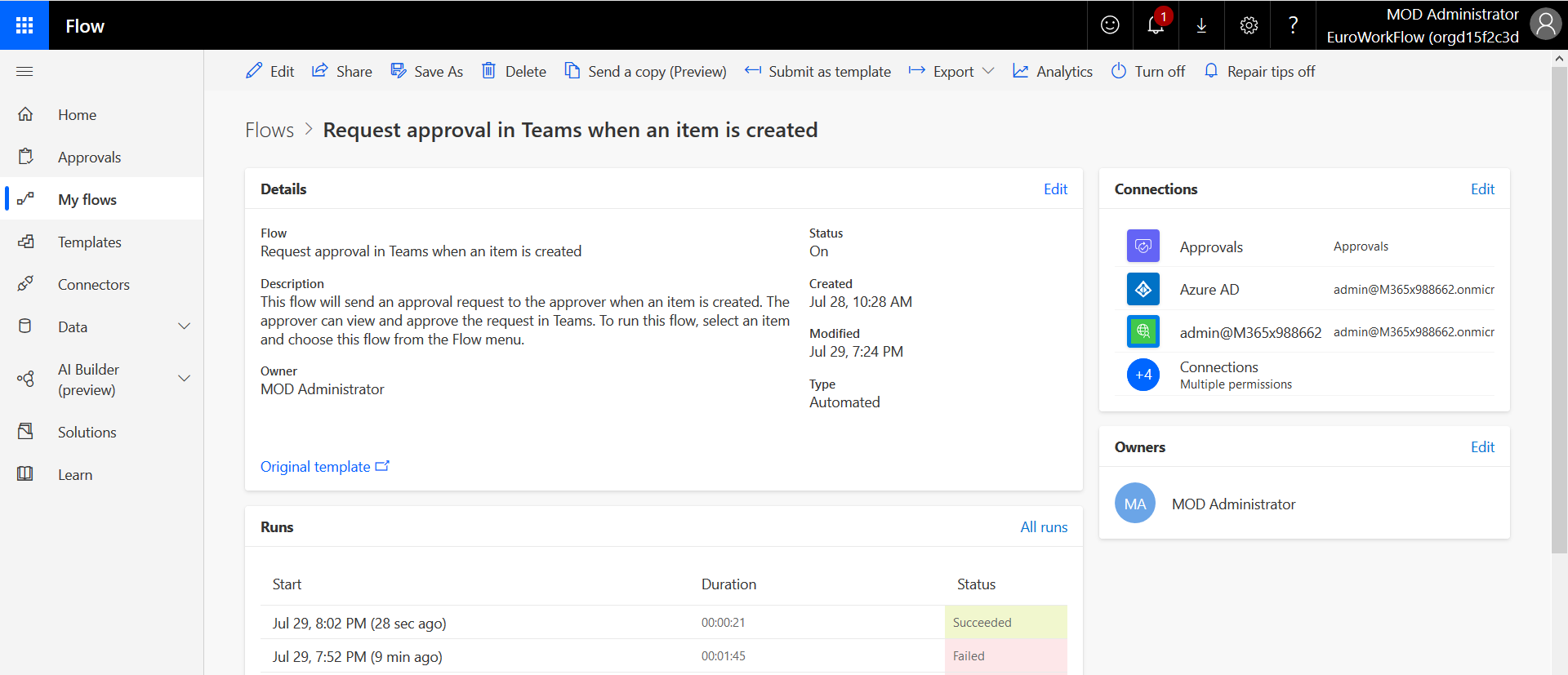
1. Save it and then go back to your Admin’s Teams client and wait for the Approval Card to arrive. Approve.



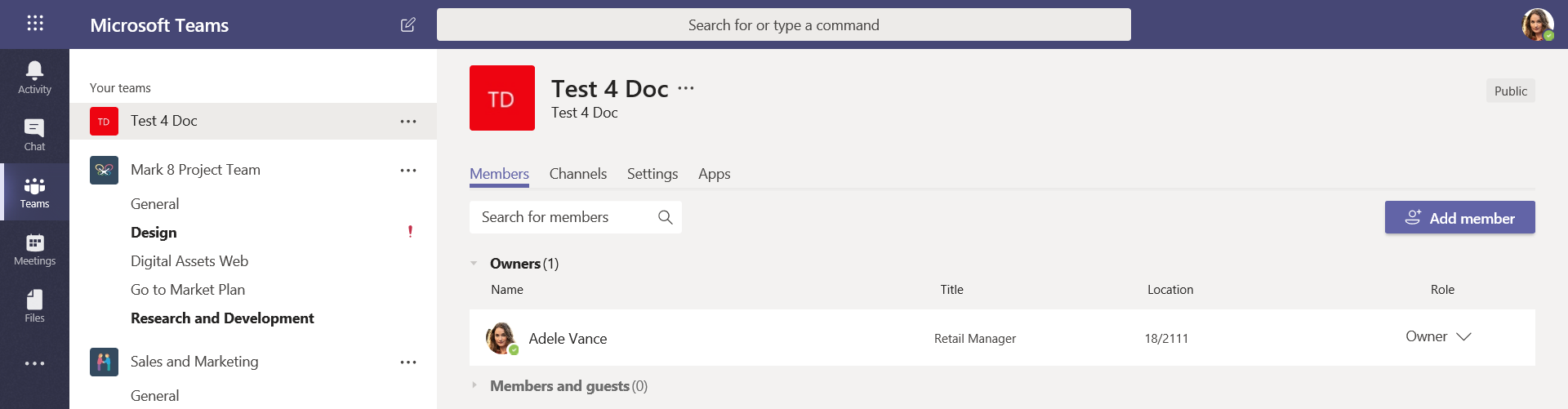
1. Add Comment and then submit.
2. Navigate to your Flow Portal and check if the Flow has succeeded or failed.



1. Create and new list item and repeat the test.
2. Approve the Flow in Team and check to see if it has succeeded.



1. Logon to Adele’s Teams client and verify if she is the owner of the new Team.



# 5. Adding additional attributes to the Workflow

During testing Contoso have observed that the standard Teams template they used for the above Flow creation configured Teams with their Visibility set to Public. This isn’t going to be acceptable to some parts of Contoso’s business who want to establish Private Teams that will not appear in searches. Furthermore, Contoso want to be able to classify each Team as either Confidential, Internal or External to provide some guidance to their users as to what type of company information should be shared within any given Team. They also want to have the ability to enable or disable Guest access as part of their automated Workflow, specifically the ability to allow Guest access for Teams with the classification of External and blocked Guest access for the Internal and Confidential classifications.

The results of the previous Governance labs demonstrated that these capabilities are available to Teams/Groups, so Contoso want to now create a Flow that delivers these features programmatically.

Note: as this is an advanced lab the steps provided below will only provide the core information needed to create the additional steps to meet Contoso’s new requirement for:

* Teams Visibility = Public/Private
* Classification = Confidential/Internal/External
* Guest Access = True/False

**Note:**

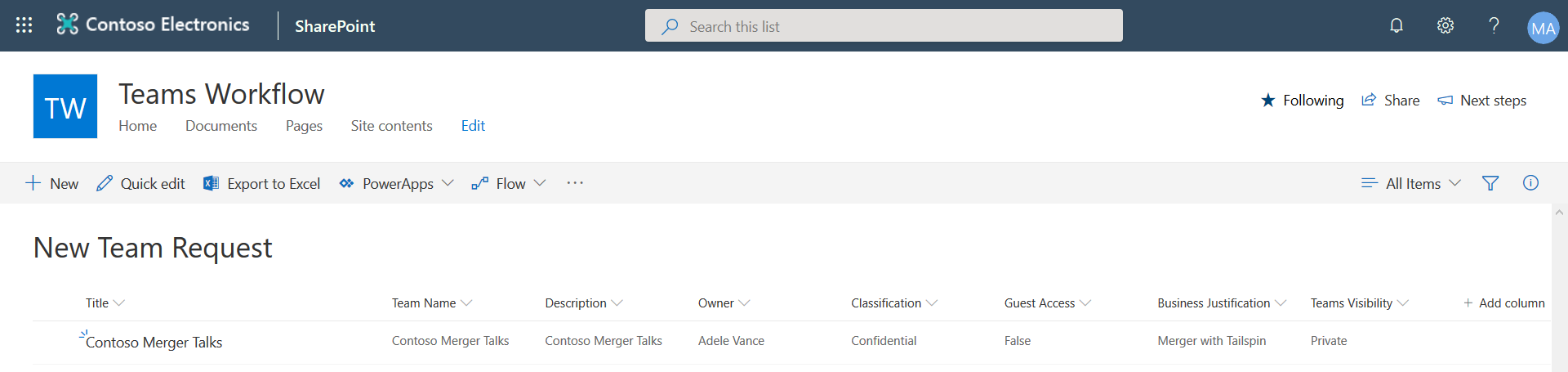
**This lab currently uses Azure AD site classification which will be eventually superseded by Office 365 Sensitivity Labels. These are currently in public preview. When Sensitivity Labels go GA, this lab module will be updated to include the newer classification methodology that also includes Guest Access and Managed Device controls. In the meantime, more information on Sensitivity Labels can be found here:**

<https://docs.microsoft.com/en-gb/microsoft-365/compliance/sensitivity-labels>

<https://docs.microsoft.com/en-us/MicrosoftTeams/sensitivity-labels>

## Lab

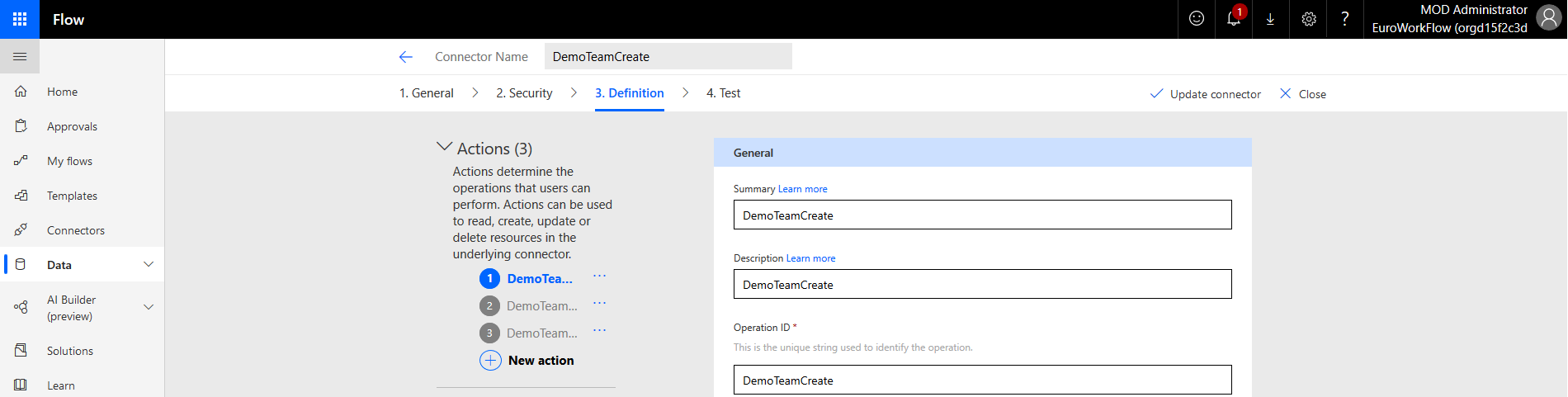
1. Navigate to your SharePoint list and add new columns for the new Teams attributes you want to include in the Flow.



Note: to support the Microsoft Graph API calls your Flow you need to ensure your new columns will be populated with the correct syntax options, hence you need to select the appropriate settings option so that:

* Teams Visibility = Public/Private
* Classification = Confidential/Internal/External
* Guest Access = True/False

2. Next, edit your Custom Connector. We are going to add two new Actions. Each one will have its own unique Operation ID – DemoTeamClass and DemoTeamGuest.



Note: The new Actions will make Microsoft Graph API call using the same App Registration you created earlier in the lab. No changes are needed in AAD to support these new Actions. The Requests and Responses for each Action are provided in the next steps.

3. Add the Request and Response content that’s provided below for DemoTeamClass.

PATCH [https://graph.microsoft.com/v1.0/groups/{id}](https://graph.microsoft.com/v1.0/groups/%7bid%7d)

Content-Type: application/json

***Request:*** *Paste the following:*

{

"classification": "Internal",

"visibility": "Public"

}

***Response:*** *Paste the following:*

HTTP/1.1 204 No Content

4. Add the Request and Response content that’s provide below for DemoTeamGuest.

POST https://graph.microsoft.com/v1.0/groups/{id}/Settings

Content-Type: application/json

***Request:*** *Paste the following:*

{

"displayName": "Group.Unified.Guest",

"templateId": "08d542b9-071f-4e16-94b0-74abb372e3d9",

"values": [

{

"name": "AllowToAddGuests",

"value": "True"

}

]

}

***Response:*** *Paste the following:*

cache-control: private

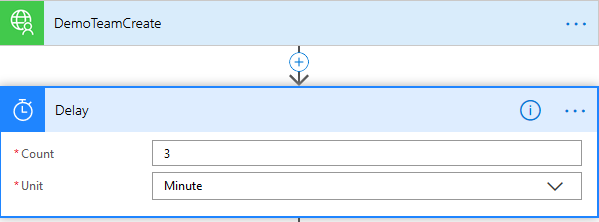
client-request-id: bce96b4d-6c85-44d6-aef4-13de97ccf937

content-type: application/json;odata.metadata=minimal;odata.streaming=true;IEEE754Compatible=false;charset=utf-8

request-id: bce96b4d-6c85-44d6-aef4-13de97ccf937

5. Be sure to Update the Connector

6. Now we are going to edit the Flow. After the DemoTeamCreate action add a Delay action.



Note: We are doing this so that the previous Team creation step has time to complete. If you make an immediate API call to update the underlying Group configuration it is very likely you will encounter errors.

7. Add a new action. Select your Custom Connector and choose the DemoTeamClass option. Then configure the step as shown below.

Note: To identify (id) your new Team, you need to use a simple substring function and the dynamic content for the Location attribute returned from the previous DemoTeamCreate API call.

The returned Location information will contain something like:

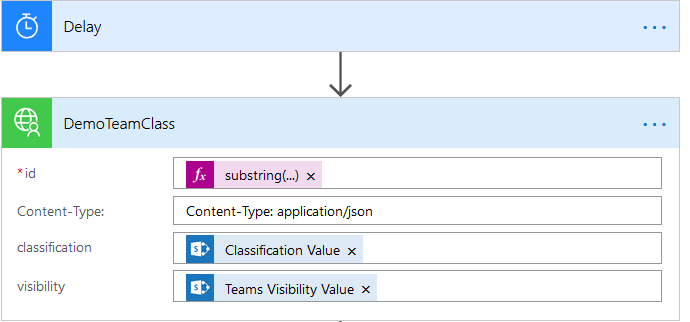
/teams('17eb901a-b8e7-4c6f-be27-c569d78b903d')/operations('48cb9edf-ca3e-4b07-9591-f5190ece96a0')

Where 17eb901a-b8e7-4c6f-be27-c569d78b903d is the new Teams id.

Instructions on how to create the Expression syntax are provided below:

In the Expression window type in substring() and then from the Dynamic Content window add the Location attribute between the brackets e.g. (Location), this will then transform to outputs('DemoTeamCreate')['headers']['Location']), you then add 8,36 which are the Teams id starting position (characters from the start) and length (characters). The finished expression will look like:

**substring(outputs('DemoTeamCreate')['headers']['Location'],8,36)**



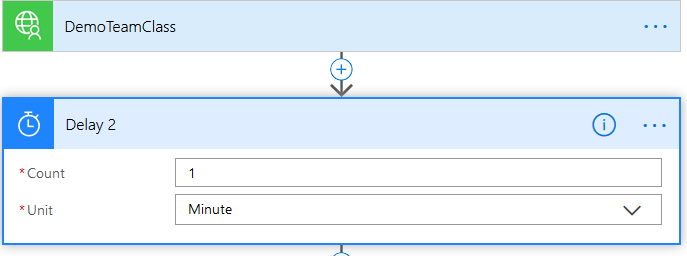
Use the Location attribute on the Expression

Machine generated alternative text:
Dynamic content 
Expression 
x nCreate ' ) [ ' headers ' ' Location ' 
Update 
DemoTeamCreate 
Body 
Content-Location: 
Content-Location: 
Content-Type: 
Content-Type: 
HTTP/I.I 
HTTP/I .1 
Location 
Location 

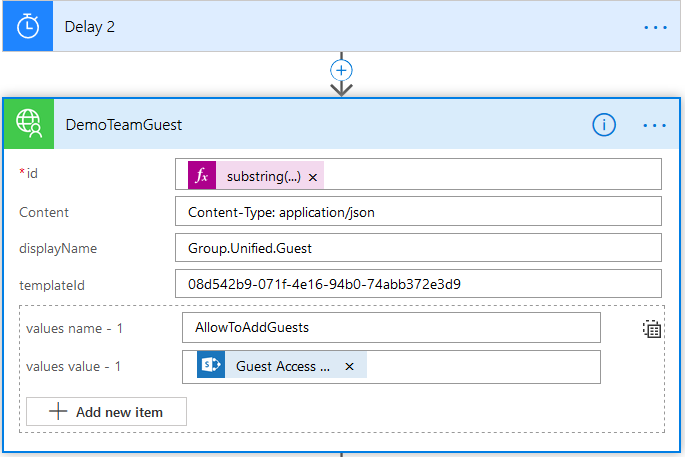
Insert the Location in the substring function and add the character position and length.

Machine generated alternative text:
DemoTeamCreate 
DemoTeamClass 
fid 
Content-type: 
classification 
Add an expression to do basic things like access, 
convert, and compare values. Learn more 
Dynamic content Expression 
Hide 
.fx substring(outputs( 'DemoTeamCreate ' ) [ ' heade 
Add dynamic content 
Content-type: application/json 
s Classification Value x 
Update 
String functions 
concat(text_l, text_2?, 
Combines any number of strings together 
See more 

8. Add another delay just to let the change propagate.



9. Now add the DemoTeamsGuest custom connector.

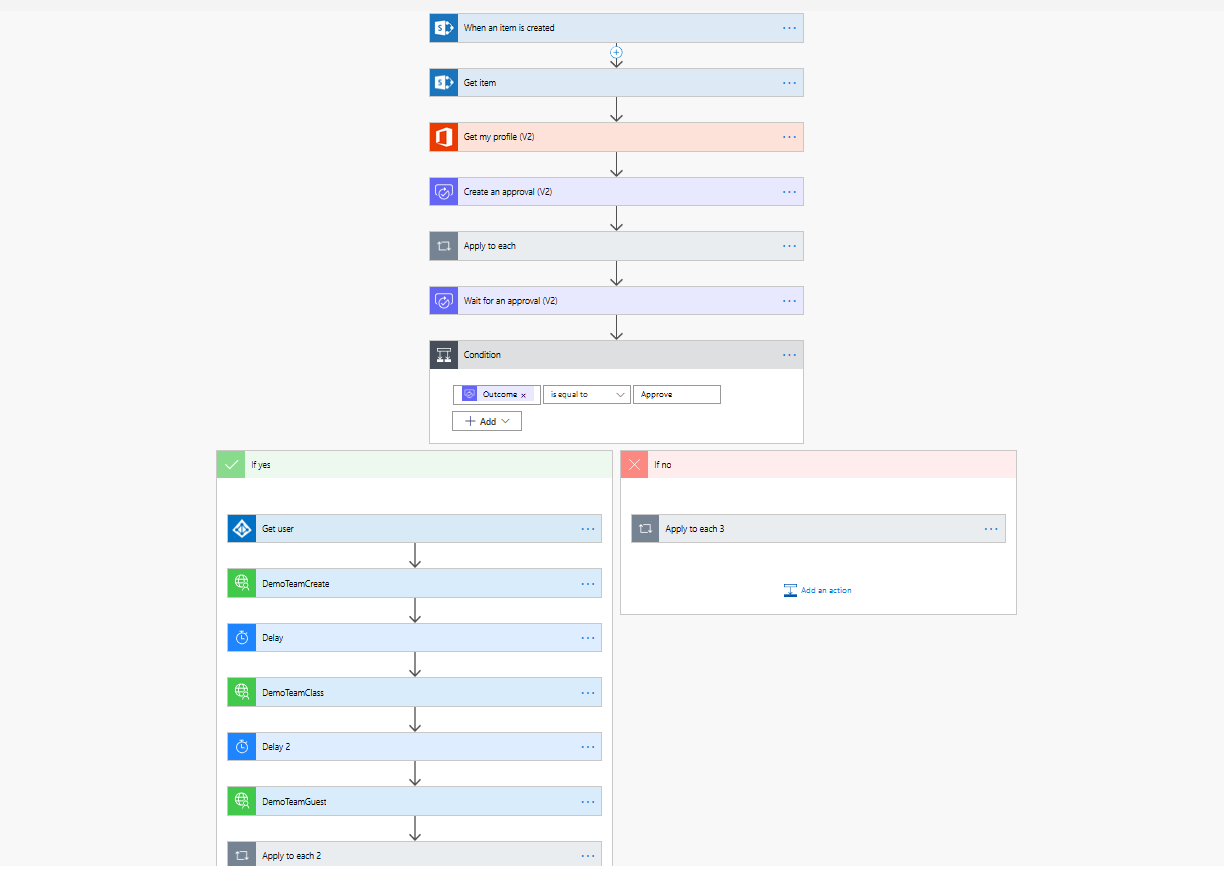


Note: The Expression with the substring function is the same as you used for DemoTeamClass. The Content, displayName, teamplateId and values name - 1 is static content.

* Content-Type: application/json
* Group.Unified.Guest
* 08d542b9-071f-4e16-94b0-74abb372e3d9
* AllowToAddGuests

The dynamic content “Guest Access” is derived from the SharePoint list.

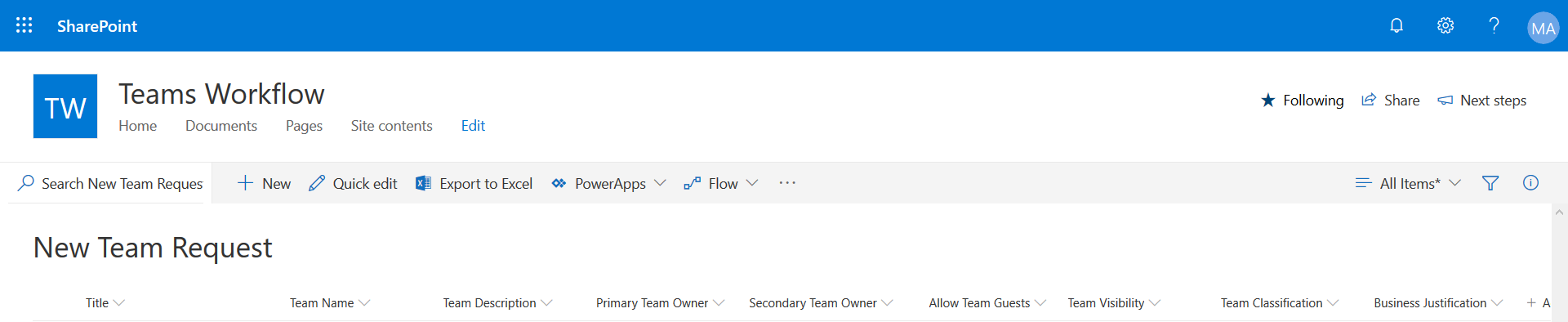
10. Once you have saved the content, your high-level flow should look like the one provided below:



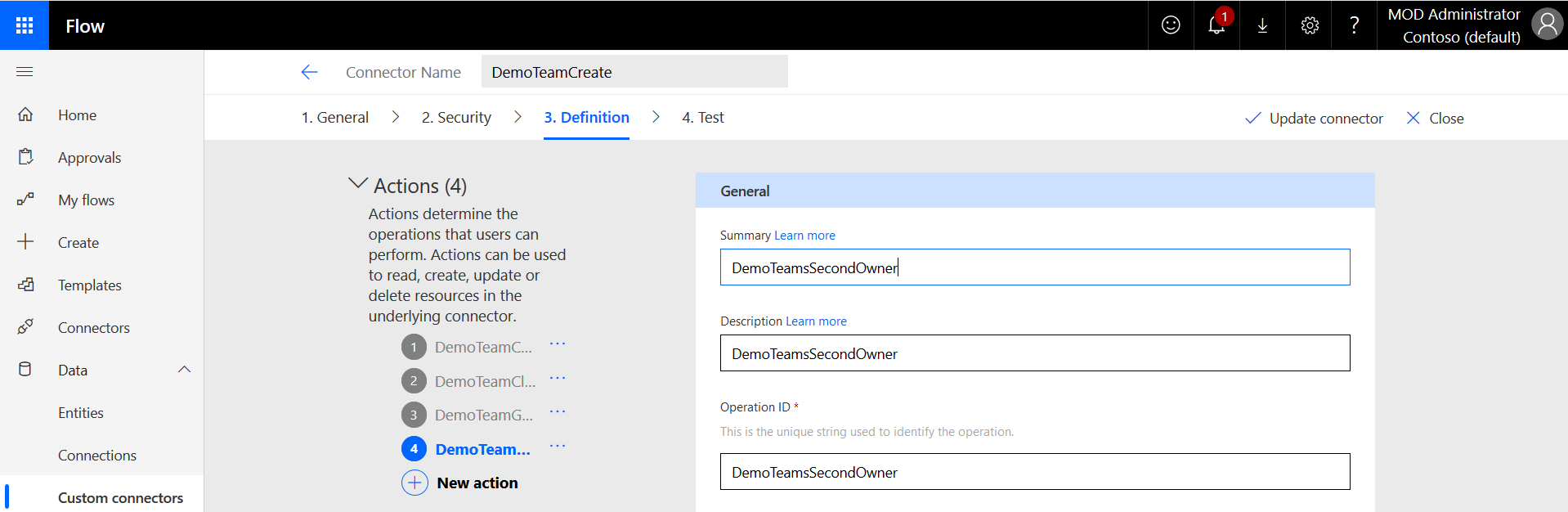
11. Now create a new SharePoint list entry and test.

# 6. Creating a 2nd Owner for your Teams

1. Update your SPO List to add a Secondary Team Owner (make it a required parameter)



2. Create a new Action called DemoTeamSecondaryOwner and use the following API call to update your new Team with the second owner.



Request:

POST https://graph.microsoft.com/beta/groups/{id}/owners/$ref

Content-type: application/json

Content-length: 30

{

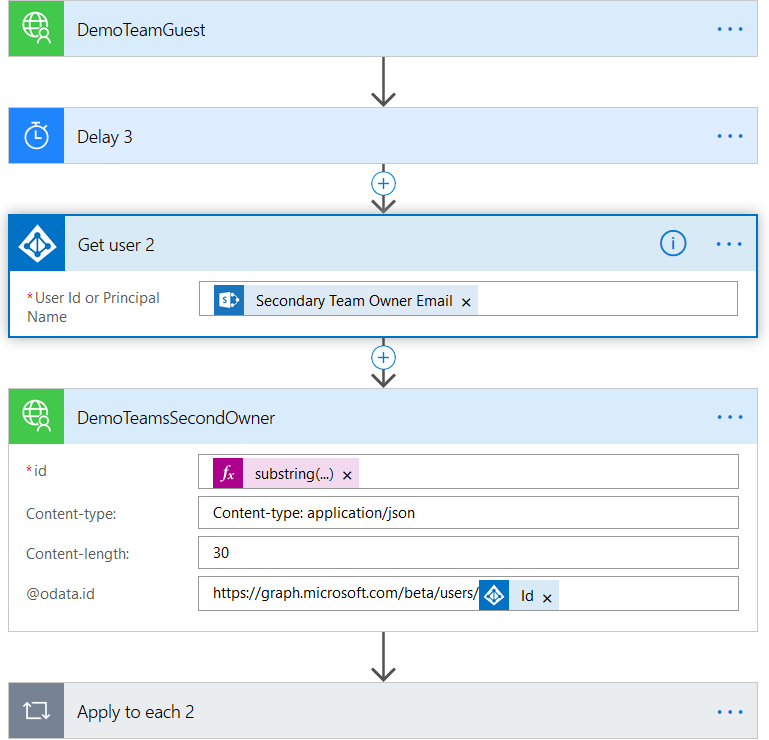
"@odata.id": "https://graph.microsoft.com/beta/users/{id}"

}

Response:

HTTP/1.1 204 No Content

3. Add new Actions within your Flow to get the Object ID of your second owner and then call the new Custom Connector action to update your Team. Save and test.



# 7. Optionally Create a PowerApp from your SharePoint list

It’s very easy to front-end your Flow with a PowerApp. The simplest way to approach this is to create the App from the SharePoint list.

The following screenshot will get you going. The additional steps are intuitive, however, if you have any questions please speak to a proctor.

A screenshot of a computer

Description automatically generated

# 8. Optionally add Team Members and Channels to the Flow.

This optional section of the lab is aimed at more advanced lab users who want to utilise the learning from the steps outlined above to create additional Connectors and Flow steps. The challenge is to add members and channels to the Teams creation process.

# If something goes wrong…

1. In order to check, what happened to your Flow, navigate to **My flows**, click on the Flow (e.g. **Start approval when a new item is added**) and you will see flow details:

A screenshot of a cell phone

Description generated with very high confidence

1. In Run history you may click on any of the flows to see its execution details:

A screenshot of a social media post

Description generated with very high confidence

1. Select any failed action and review the error details.

# This is the end of the lab.