O365 Client Credentials flow - Email Authentication

Contents

Introduction

As Microsoft has started to turn of Basic authentication for email protocols for tenants in favor of Modern Authentication (OAuth2) we have implemented one form of this for ECS email using an interactive flow, see also: O365 On Behalf Of - Email Authentication

This was done, because it was the quickest way for us to prevent users being unable to use email anymore, but it is not the optimal solution.
We now have implemented the client credentials flow, which requires no user interaction and uses Microsoft Graph for email handling instead of STMP and IMAP.

The benefit for administrators/users is that there only needs to be one client secret provided in the Azure App which then needs to be configured in ECS once and then **all the tenant's mailboxes** can be used in ECS by just adding them to the ECS configuration.
If there needs to be a restriction on allowed mailboxes, this can be controlled entirely by the Azure admin. For more information about this, please see: Limiting application permissions to specific Exchange Online mailboxes

Also, the expiration is controllable on Azure side and can be set to a maximum of two years (at the time of writing), meaning that when it works, the ECS configuration only needs to be updated once every two years when a new secret needs to be issued.
In comparison, the interactive flow needs a quarterly update, involving user interaction and (most likely) a MFA input - per mailbox!

All mailboxes are accessible

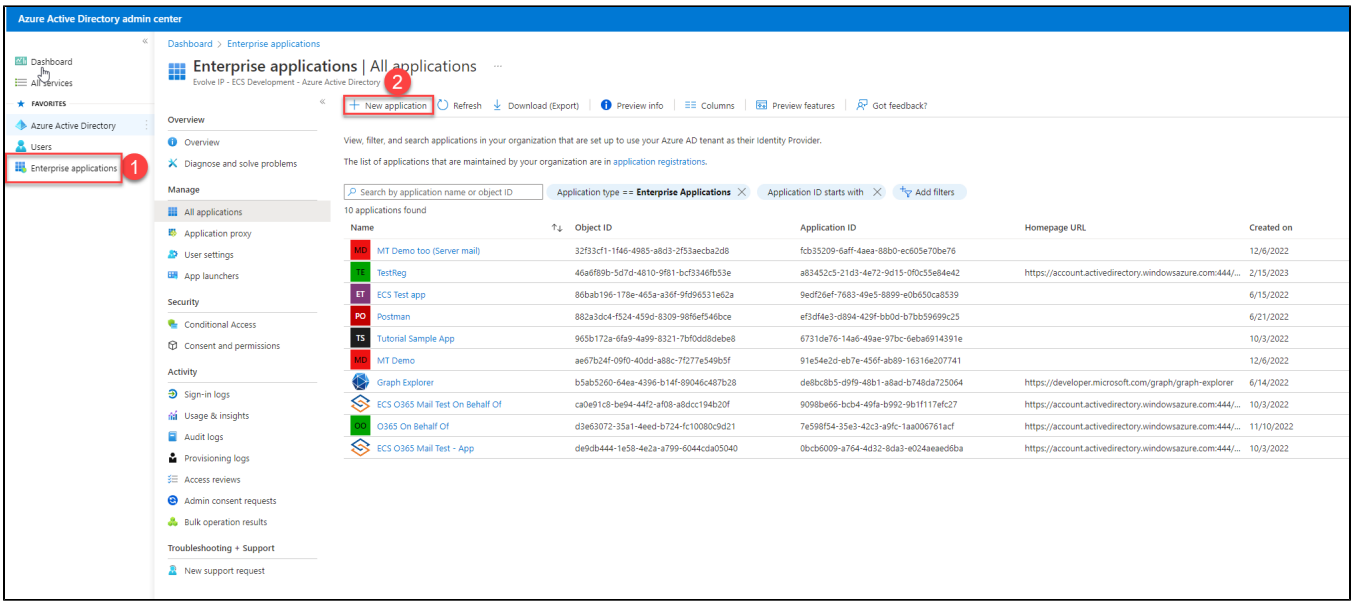
We emphasize again that all the tenant's mailboxes are available by default, you probably want to limit access in Azure to only the mailboxes you actually need in ECS.
For more information about this, please see: Limiting application permissions to specific Exchange Online mailboxes

For Azure administrators

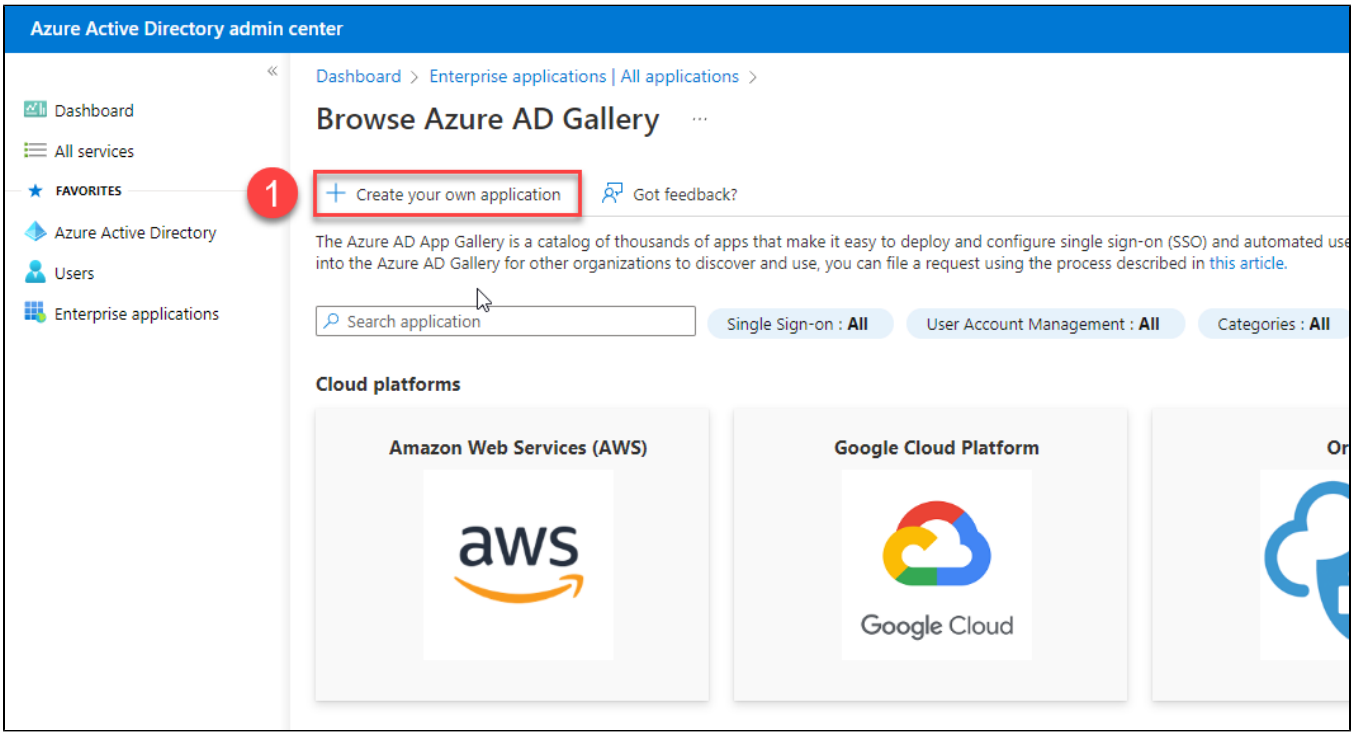
In Azure Active Directory you need to have an application that supports mail using MS Graph with a client secret available.

Application creation

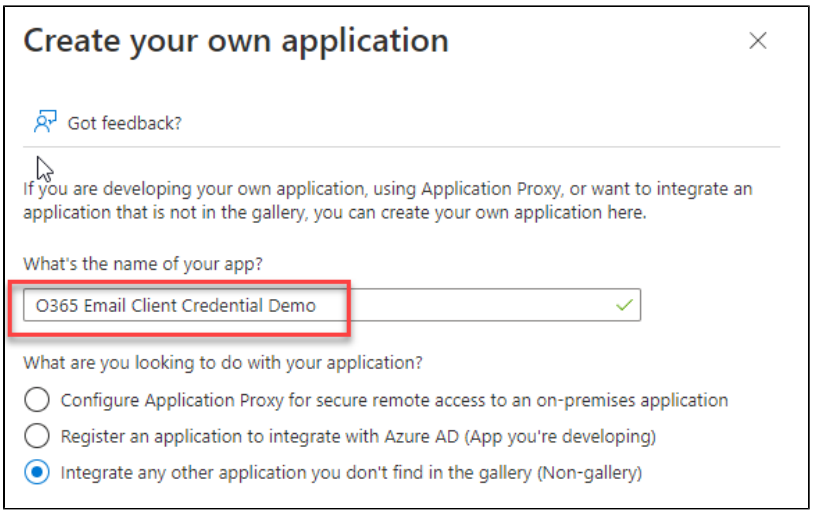
If you do not have the application set up yet, create one (else, proceed to add permissions or adding secrets as needed):



1. Go to "Enterprise applications"
2. Select "+ New application"



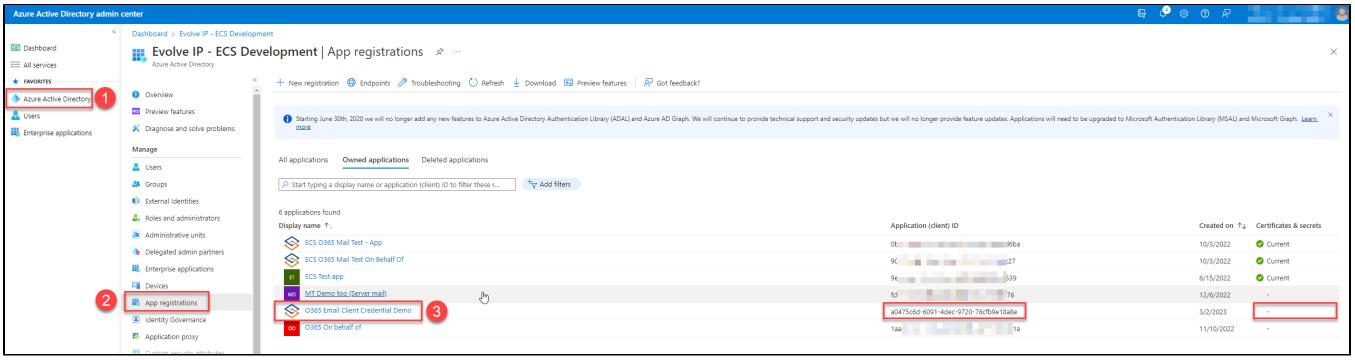
1. Then select "+ Create your own application"



Give it a logical name so you or others can easily identify it in the future and select the correct application type, for us it is "Integrate any other... ".

Application registration and permissions

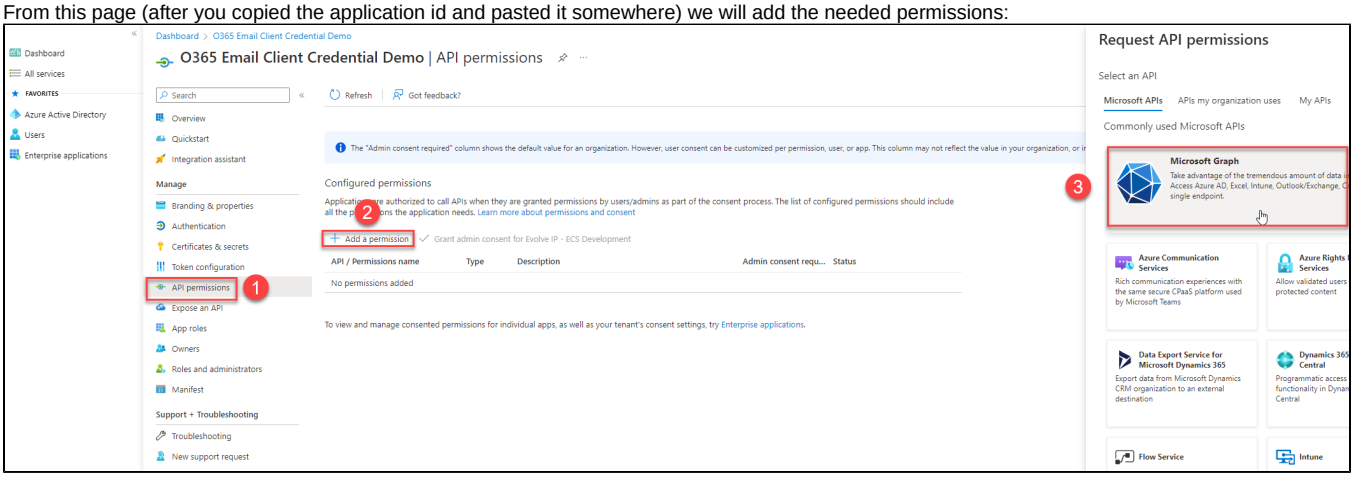
Now go to the application registration page.



1. Select "Azure Active Directory" in the main menu
2. Select "App registrations" in the sub menu
3. As you can see here, the application does not have a secret yet. (far right column)
   Now select the application by clicking the name, this will bring you to the application overview page.

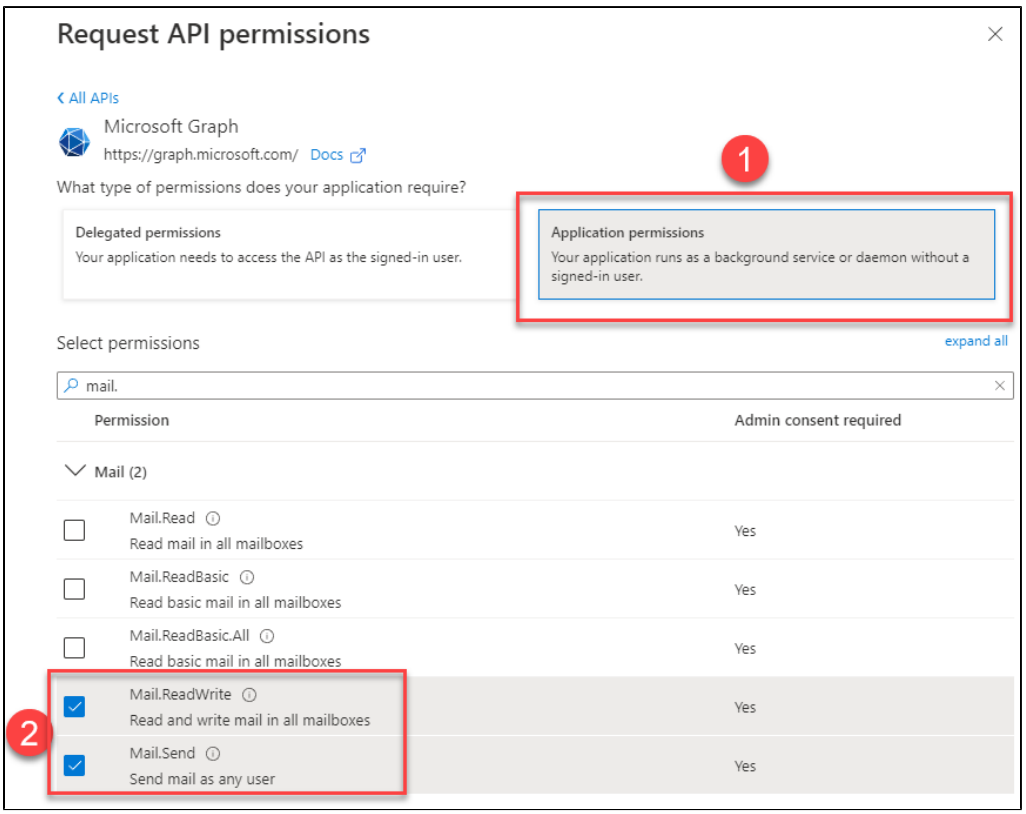
> **Application id**
>
> Also take note of the application id, this is part of the configuration we need. You can copy it now for later use. or from the application overview page which might be easier since it has copy buttons next to the id.
> In our example this is "a0475c6d-6091-4dec-9720-76cfb9e18a8e"

From this page (after you copied the application id and pasted it somewhere) we will add the needed permissions:



1. Select "API permissions"
2. Select "+ Add a permission"
3. Select "Microsoft Graph"

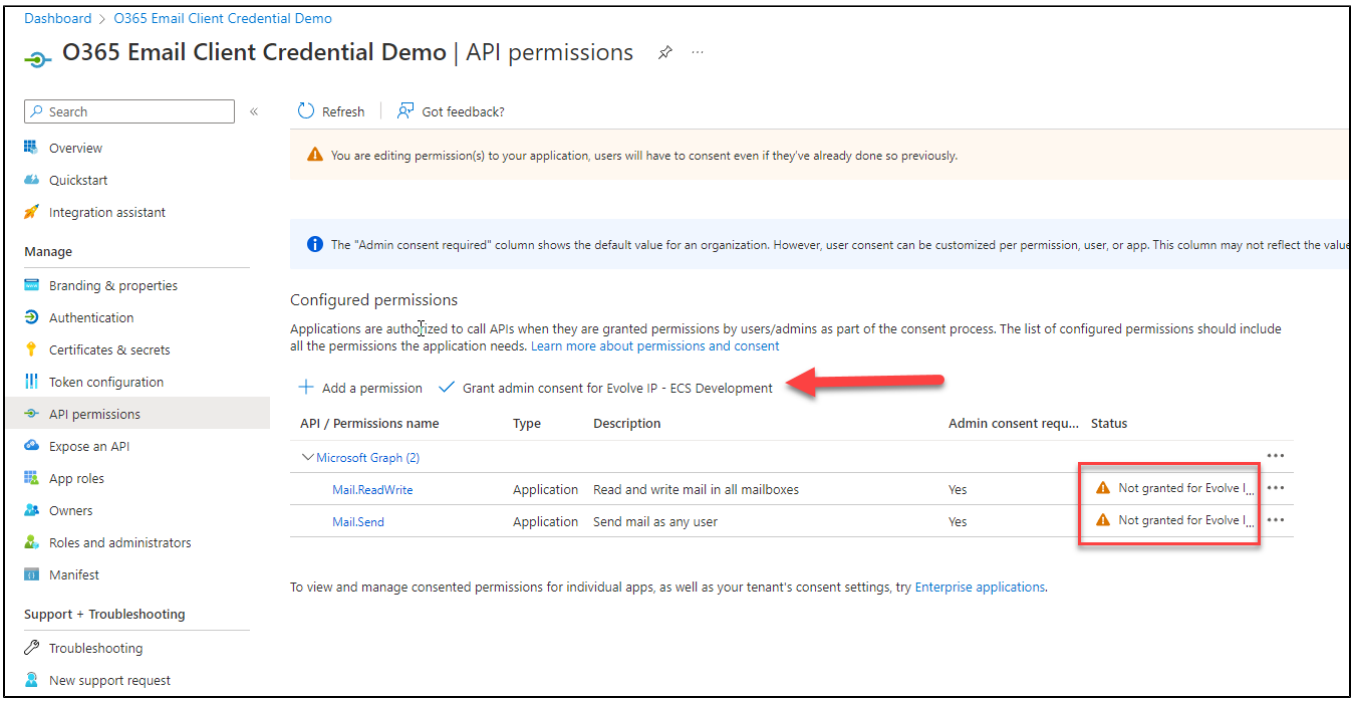
On the API sidebar:



1. Select "Application permissions"

> **Search**
>
> For the next step note that you can do a partially search for "mail." in the search box as show in the screenshot.

2. Select both:
    a. Mail.ReadWrite
    b. Mail.Send

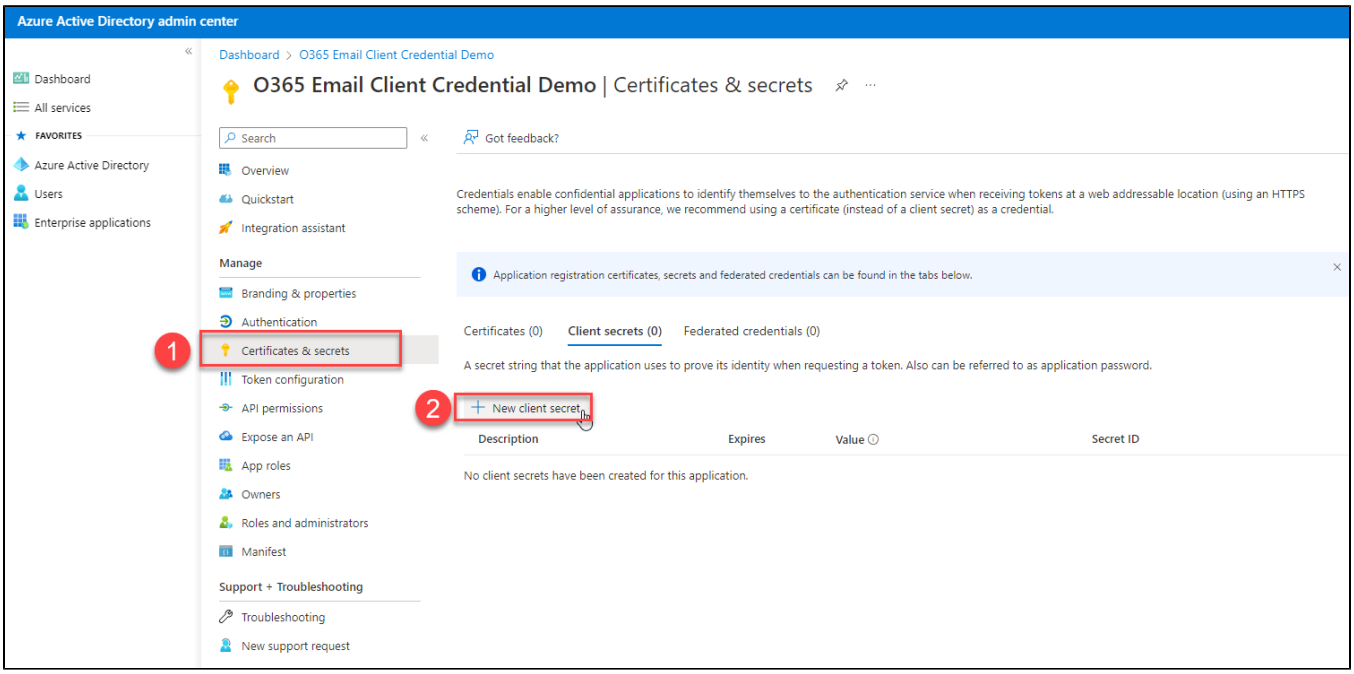


When you confirm your permissions you should see they are added, but not have not been granted access yet.
You can grant access here by clicking on the "Grant admin consent for..." title.

Application secret

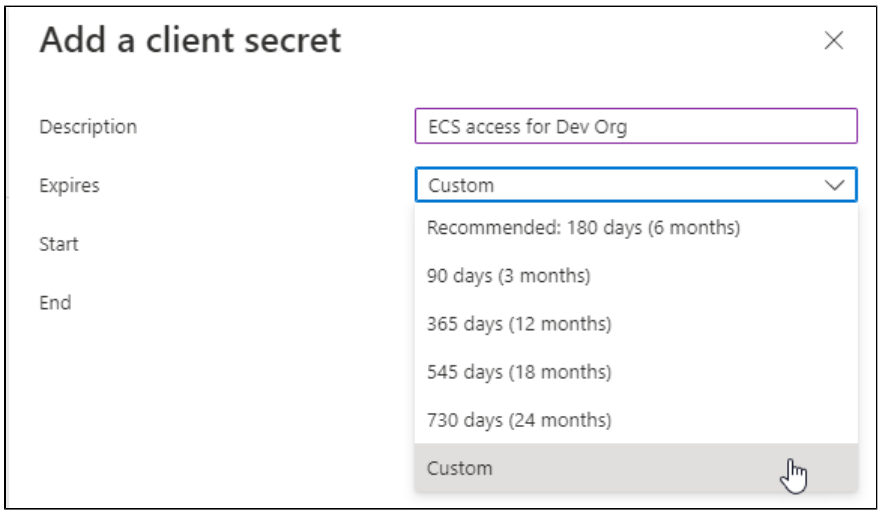
We now are ready to create a secret we can use in ECS to get access to the mailboxes.

> **Expiration**
>
> If the secret is about to expire or is already expired, return here to create a new secret. You will of course need to update ECS configuration afterwards as well in order to use the new key.



1. In the submenu select "Certificates and secrets"
2. Select "+ New client secret"

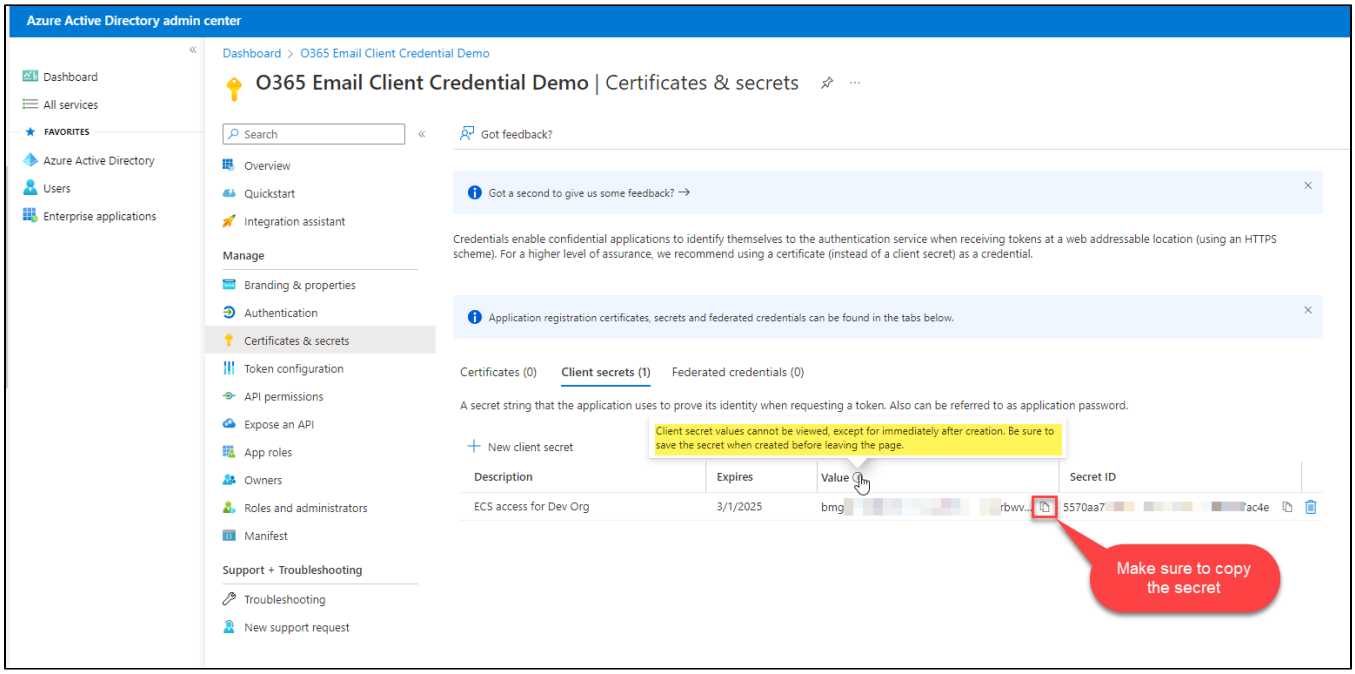
On the client secret sidebar:



- Give it a logical description so you or others can easily identify it in the future
- Select the desired expiration, using the "custom " option you can also set a start date if desired

When the secret is generated you will see its value on the page.

> **Only chance to copy**
>
> Be aware that this is the **only** chance to copy the secret's value, if you navigate away and return it is no longer visible or copyable and you will need to generate a new secret!



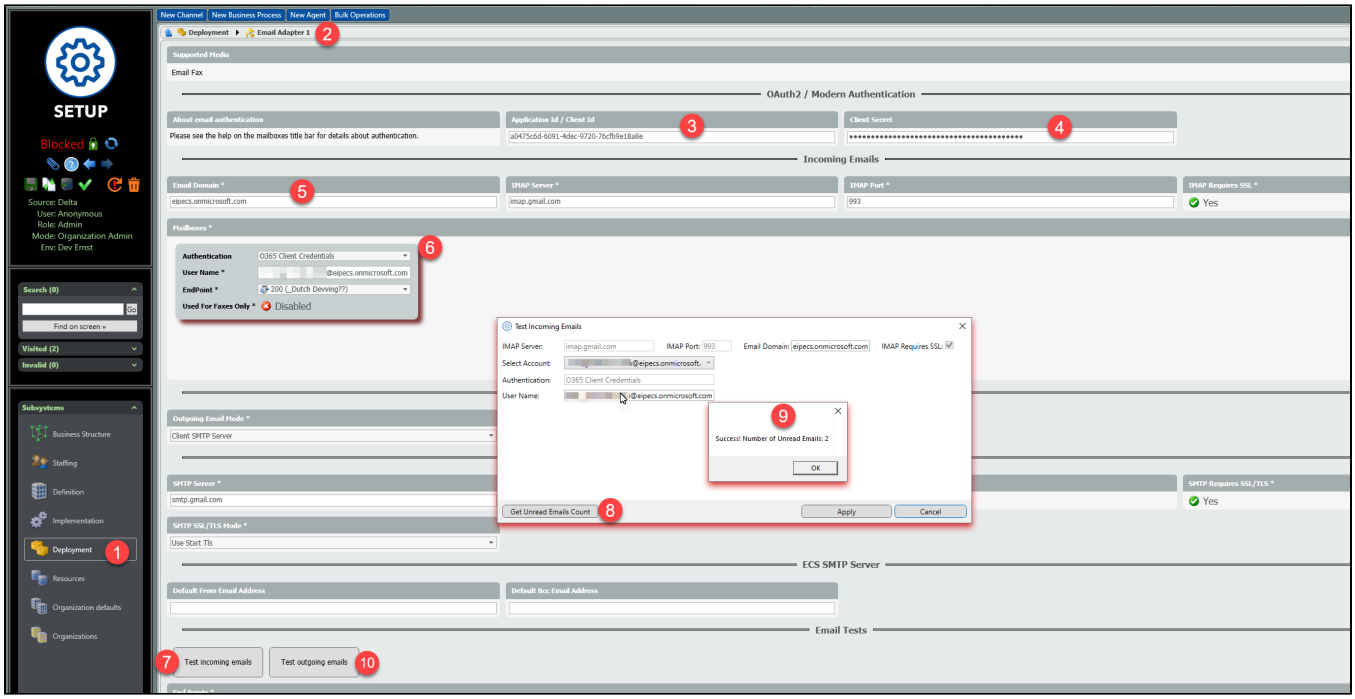
- Use the copy button to copy the secret's **value** - the ID is not relevant.
- Paste it somewhere for later reference

For ECS organization administrators/implementers

When the Azure part is set up correctly, the mailboxes can be configured.

Configure a mailbox

This step is much simpler than the interactive flow, open the organization's setup and then navigate to the email adapter:



1. Select "Deployment"
2. Select the email adapter (selection itself not shown in the screenshot, only the result, but it is under deployment - software services - email adapter)
3. Fill out the application id of your Azure application, in out example it is "a0475c6d-6091-4dec-9720-76cfb9e18a8e"
4. Paste the secret of your Azure application. In case you forgot to copy it earlier, you will need to create a new secret.
5. Fill out the domain, this is used as tenant identifier for logging in in Azure, which should normally match the domain of your user.

> **Other users**
>
> Note that you can have other users, but in that case they need to have been imported into your Azure Active Directory

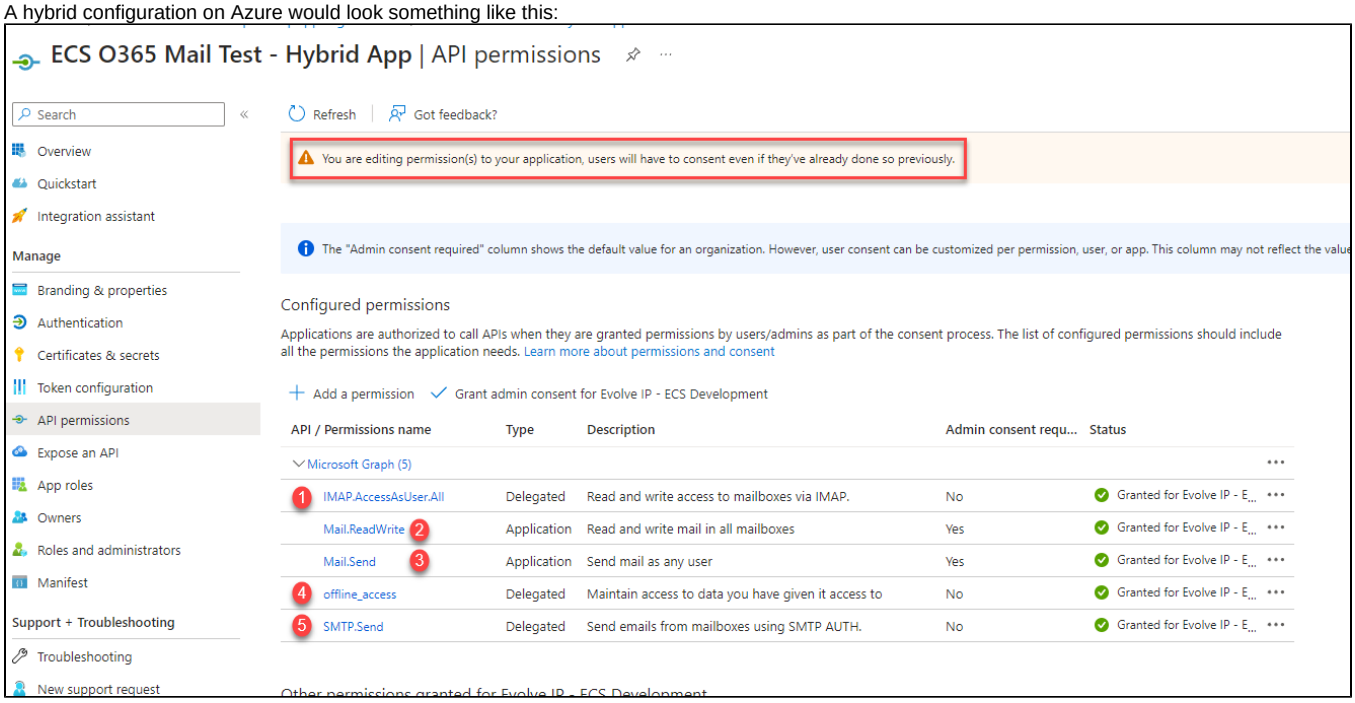
6. Create a mailbox using "O365 Client Credentials" and the username of the mailbox you want to use
7. Select the "Test incoming emails" button
8. Select "Get unread emails count"
9. If configured correctly, you should see this window.
   Note that it can show 0 (zero), that depends on actual unread emails in the mailbox, but that will still mean the configuration works
10. Optionally you can also send out a test email

If the configuration works you can optionally add more mailboxes and then deploy the configuration.

Migrating from On behalf Of to Client Credentials / Hybrid application

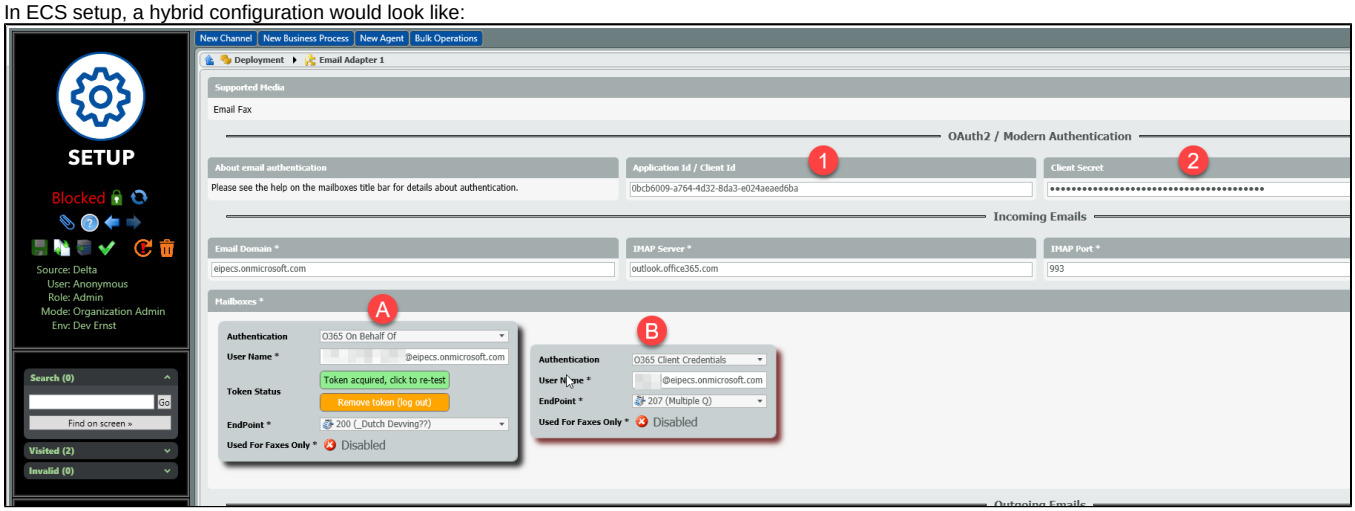
If you already are using the interactive flow, it should be fairly easy to migrate/upgrade to the client credentials flow.
Since the application is already present, you can add the permissions and secrets as described above and as long as you do not remove the original API permissions you are able to use both types in ECS.

A hybrid configuration on Azure would look something like this:



- The permissions 1, 4 & 5 are set for the "on behalf of" interactive flow as can be seen by the Type "Delegated".
- Permissions 2 & 3 are for the client credentials flow, Type "Application".
- ⚠️ Note the warning above, on changes it may be necessary for users to give consent again, meaning acquiring a new token and following the popup guidelines

In ECS setup, a hybrid configuration would look like:



- The application id (1) remains the same.
- The new client secret is added (2).
- Mailbox A is using a token, but note that if all settings are correct we could just change authentication type.
- Mailbox B is using the client credentials flow.