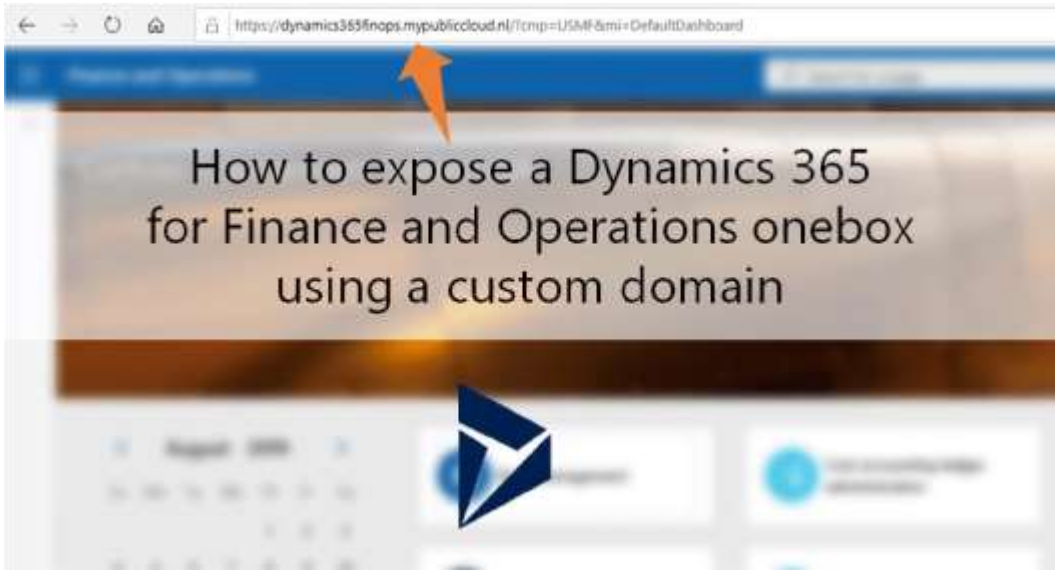
Tutorial: Expose Dynamics 365 for Finance and Operations onebox using a custom domain

by Michiel | Aug 23, 2019 | Dynamics 365 for Finance and Operations | 14 comments



In this tuturial you'll learn how to expose a Dynamics 365 for Finance and Operations onebox (VHD edition) to the public Internet using a custom domain name. For example, the client URL becomes
"dyn365demo.mydomain.com" in stead of default URL
"usnconeboxax1aos.cloud.onebox.dynamics.com".

"Why don't you just deploy a D365FO Cloud Hosted Environment to Azure using LCS?" – I hear you say 🙂 .
Yep, that's the most easy (and supported) way to expose a D365FO onebox to the Internet.

However, there are reasons why you would want to follow my tutorial below:

- Deploy and run the onebox in an on-prem (preferably DMZ) or other Cloud environment (like AWS).
- Keeping full control over the Azure deployment processes and governance.
  For example, being able to deploy only necessary resources using ARM or Blueprint templates.
- Limiting the remote access to your LAN in stead of the Internet.
- Sharing your development onebox with colleagues or the Internet.
- Changing the default *.dynamics.com client URL to your own corporate domain, like mydomain.com.

- Last but not least: just because you can! 😀

In other words, if you're looking for an easy and Microsoft-supported way to deploy a onebox (dev/demo) to Azure and expose it to the Internet, stop reading and follow my other tutorial which guides you through all the required steps in LCS. Otherwise, keep reading 🙂 .

Requirements

You'll need to have:

- A working and running onebox (VHD edition) of Dynamics 365 for Finance and Operations running in a hosted environment (Cloud or on-prem) of your choice.
- Don't have one? Check my tutorial.
- Your onebox needs to be exposed on port 80 and 443 to the Internet, for example using NAT port forwarding.
- Preferably host your onebox in a DMZ or behind a reverse proxy solution!
- Global Administrator privileges to your Azure AD / Office 365 tenant, because you'll need to create an Azure AD app registration.
- Learn how to create a tenant if you're not having one.
- DNS admin privileges on a public routable domain name (i.e. mycompany.com), because you'll need to create one or more DNS record(s).

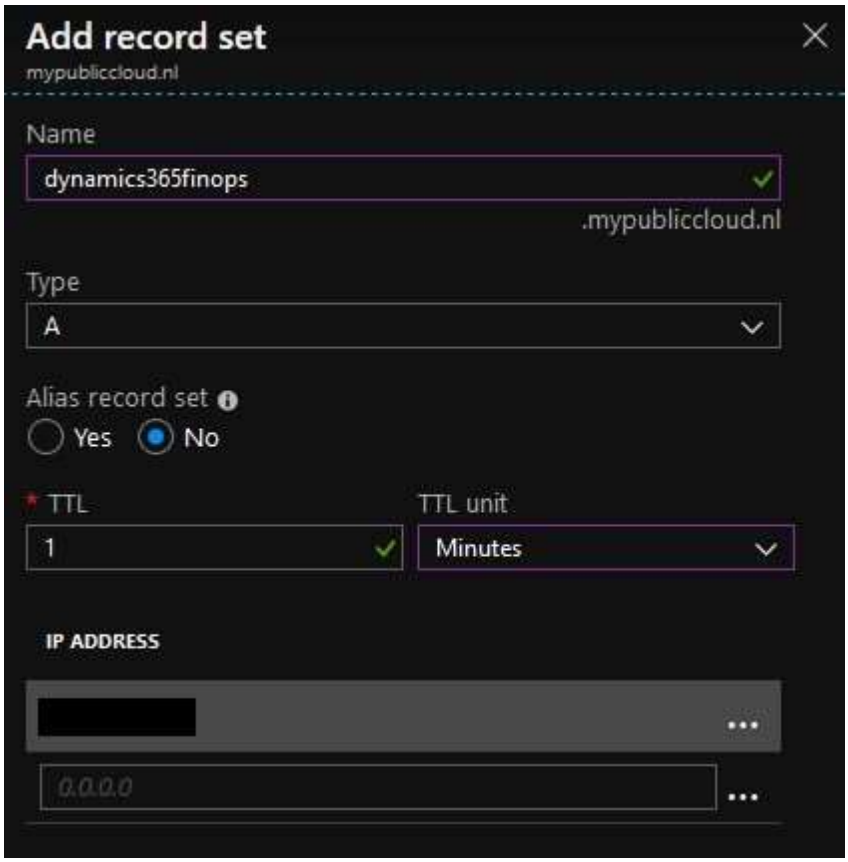
Step 1 – Prepare domain

- First, decide which (sub)domain URL to which you want to expose your onebox to, and make a note of it. In my tutorial it's configured as: dynamics365finops.mypubliccloud.nl

- Preferably use a root domain name from your Azure AD / Office 365 tenant which is connected to your onebox for log-in. I am referring to the domain name which you've entered in the Admin provisioning tool (step 3 in my VHD download tutorial).

- Sign in to the DNS configuration panel of your domain provider and create a DNS record (type A), pointing to the public exposed IP address of your onebox. Contact your domain provider if you don't know where to find the configuration panel.

  Below you'll find an example in where I've used DNS zones in Azure.



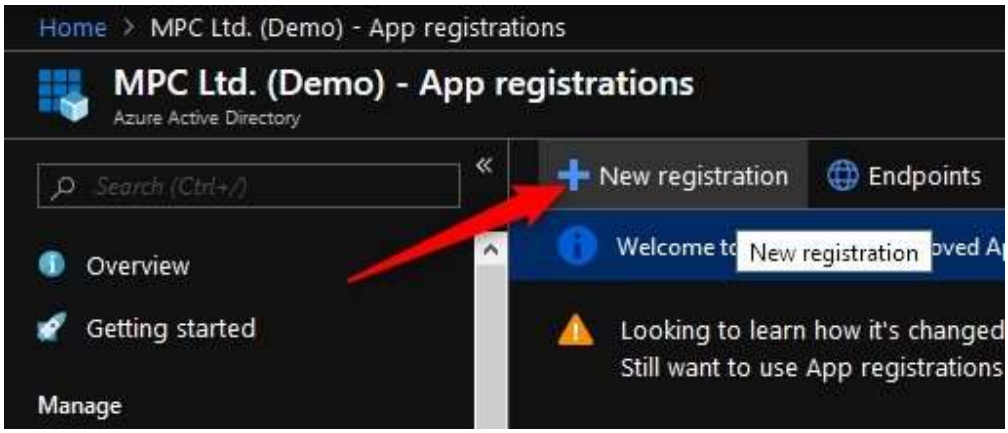
Step 2 – Configure Azure AD

In this step you will create an Azure AD application registration. This is required to let Azure AD trust your custom domain name for application usage, like you will do with Dynamics 365. Skipping or misconfiguring this step will break any attempts to sign in to the Dynamics 365 client.

- Sign in to the Azure Portal using an account with Global Administrator privileges and confirm if the portal is signed in to the appropriate tenant at the top left. This should be the tenant which holds the admin account you've entered in the Admin Provisioning Tool on the onebox desktop.

- Navigate to the Azure Active Directory blade > App registrations.

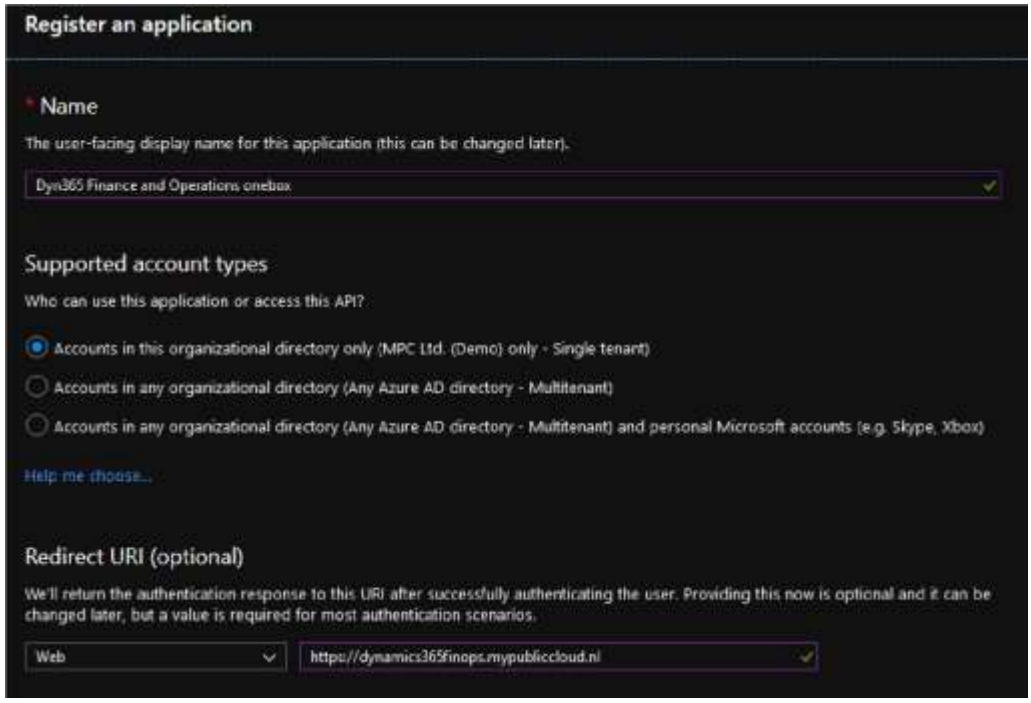
- Click + New registration at the top.



- Give it a descriptive name, and make sure to set the Redirect URI to the URL of your onebox.

In my case, it is: https://dynamics365finops.mypubliccloud.nl . Also, set the supported account types to Accounts in this organizational directory only (Single tenant).

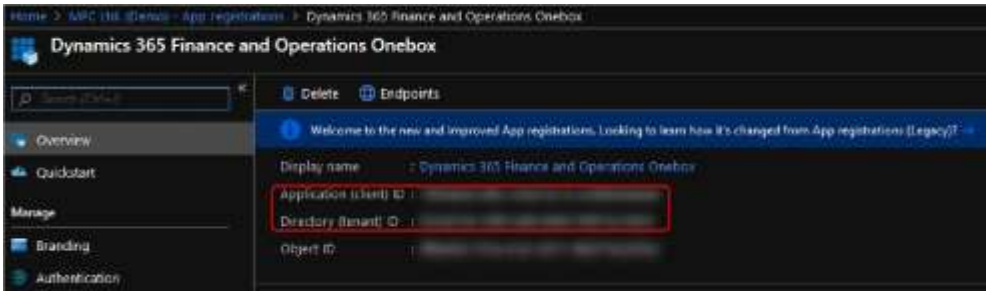
Example:



Finally, click Register at the bottom of the form.

- After registering the app, a menu menu will appear which allows you to configure the app registration.

- At first, make a note of the following data on the Overview panel. You'll need these ID's to modify a few config files on your onebox later in this tutorial:
- Application (client) ID
- Directory (tenant) ID

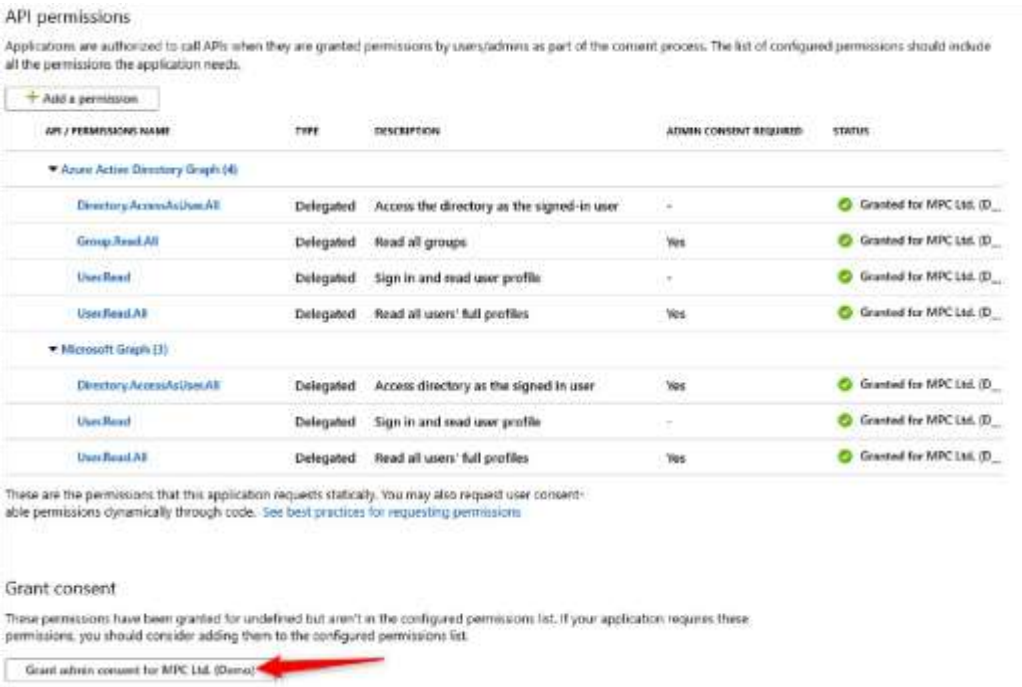


- Switch to the Authentication panel, and make sure the settings match the ones you've configured during creation.

- Switch to the API permissions panel and make sure to add the following permissions:
- Azure Active Directory Graph
- Directory.AccessAsUser.All
- Group.Read.All
- User.Read
- User.Read.All

- Microsoft Graph
- Directory.AccessAsUser.All
- User.Read
- User.Read.All

- Make sure to hit the Grant admin consent button afterwards.



- Keep this configuration panel open (because you need to complete the configuration at step 6).

Step 3 – Configure IIS config files

In this step you'll modify a few IIS config files on your onebox to change the default client URL to your custom domain.

- RDP to your onebox and sign in using the default Administrator account.

User: Administrator

Default password: pass@word1

- Don't forget to change the default password, especially when you're exposing your onebox to the Internet.

- Open Notepad (or comparable editor such as Notepad++) with Run as admin privileges, and open the following files in directory: C:\AOSService\webroot
- web.config
- wif.config
- wif.services.config

- In web.config make the following modifications:
- Search for Aad.Realm and change the spn id (containing alot of zero’s) to the Application (client) ID you’ve written down in previous step.

  Example:
  <add key="Aad.Realm" value="spn:12f3d12d-df2c-123d-1c23ac1f2b34a5a6" />

- Replace the default client URL for your own custom URL at the following keys, which can be found in a random order through the config file:
- Infrastructure.FullyQualifiedDomainName
- Infrastructure.Hostname
- Infrastructure.HostUrl
- SoapServicesUrl

- Make sure to preserve all prefixes and suffixes at each modified value.

  Examples:
  <add key="Infrastructure.FullyQualifiedDomainName"
  value="dynamics365finops.mypubliccloud.nl" /> <add
  key="Infrastructure.HostName"
  value="dynamics365finops.mypubliccloud.nl" />
  <add key="Infrastructure.HostUrl"
  value="https://dynamics365finops.mypubliccloud.nl/" /> <add
  key="Infrastructure.SoapServicesUrl"
  value="https://dynamics365finops.mypubliccloud.nl/" />

- Check if the directory tenant ID on line 22 (key TenantDomainGUID) matches the Directory (tenant) ID you’ve written down in previous step. If it’s not, rerun the Admin Provisioning Tool on the desktop and enter a valid Azure AD account which is listed in your tenant.

- In wif.config make the following modifications:
- On line 8 you’ll find a key named add value=”spn:000……..” which refers to a Microsoft-managed directory tenant ID.
- Change the ID (after spn:) to the Application (client ID) you’ve written down in previous step.

Example:

<add value="spn:12f3d12d-df2c-123d-1c23-ac1f2b34a5a6" />

- In wif.services.config make the following modifications:
- On line 5 and 6 you'll need to change the tenant name (onmicrosoft.com), tenant id (spn) and client URL (on 2 places).

Original code:

<wsFederation passiveRedirectEnabled="true"

issuer="https://login.windows.net/contosoax7.onmicrosoft.c om/wsfed"

realm="spn:00000015-0000-0000-c000-000000000000"

reply="https://usnconeboxax1aos.cloud.onebox.dynamics.com/ " requireHttps="true"

/> <cookieHandler requireSsl="true"

domain="usnconeboxax1aos.cloud.onebox.dynamics.com" path="/" />

Example modification:

<wsFederation passiveRedirectEnabled="true"

issuer="https://login.windows.net/mypubliccloud.onmicrosof

t.com/wsfed" realm="spn:12f3d12d-df2c-123d-1c23-ac1f2b34a5a6" reply="https://dynamics365finops.mypubliccloud.nl/"

requireHttps="true" /> <cookieHandler requireSsl="true"

domain="dynamics365finops.mypubliccloud.nl" path="/" />

- Save all files.

- Update 27-9-2019: Added config modification below to enable usage of Office integration and Data Management module (i.e. export data entities). Source: erconsult.eu blogpost

- Using an editor (like Notepad) with Run as admin privileges, open the following file: C:\Program Files (x86)\Microsoft SDKs\Azure\Storage Emulator\AzureStorageEmulator.exe.CONFIG

- You should find three lines of code which include the default URL http://127.0.0.1:1000X/ where X increments from 0 to 2.
  In my environment these URL's were located at line 9, 10 and 11. Replace the IP-address 127.0.0.1 at all of the 3 URL entries for your new Dynamics client URL.

Example: http://127.0.0.1:10001/ becomes –>

http://dynamics365finops.mypubliccloud.nl:10001/ Don’t forget to

retain the trailing slash at the end of the URL.

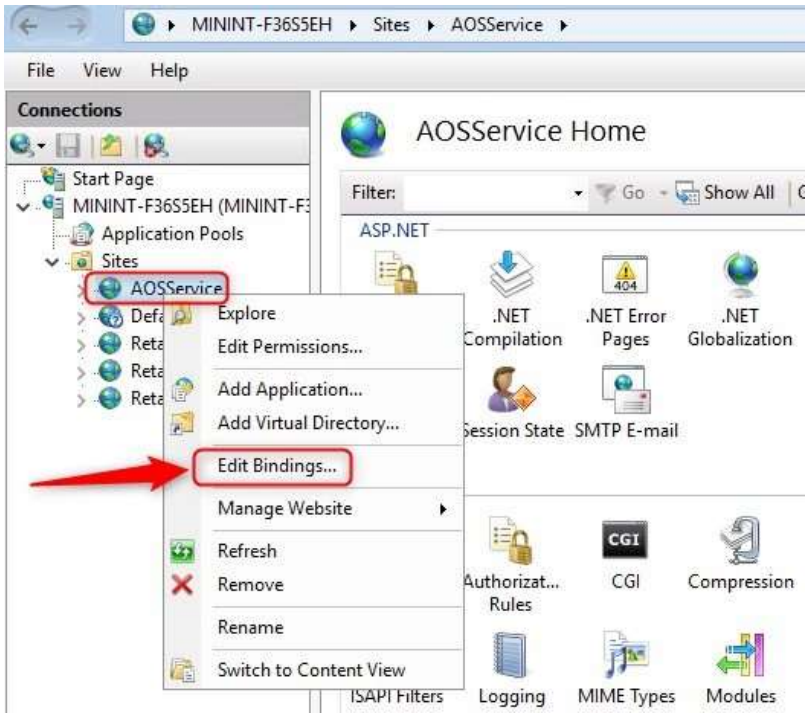
- Now open the following config file:
  C:\AOSService\webroot\web.CONFIG
  and search for key: AzureStorage.StorageConnectionString

- Replace the default encoded connection value (after value=” and between the quotes) for:
  UseDevelopmentStorage=true;DevelopmentStorageProxyUri=http://
  /dynamics365finops.mypubliccloud.nl
  and make sure to modify the URL to your new Dynamics client URL.
- Go to the Windows Firewall (i.e. by running wf.msc) and make sure to open inbound TCP & UDP ports 10000, 10001 and 10002. Also, if you’re exposing the onebox to the WAN, make sure port forwarding is also in place.

- Stop and start the Azure Storage Emulator. You can find the tool within the start menu of Windows which will tell you how to start and stop the emulator.
  Also, run a iisreset /noforce command to reload the IIS webserver.

Step 4 – Add IIS binding

- Go to the IIS manager and add a binding to the website AOSService, pointing to your custom domain URL. Make sure to tick the SNI checkbox and select a random SSL certificate.
  Don’t change or remove the default binding.



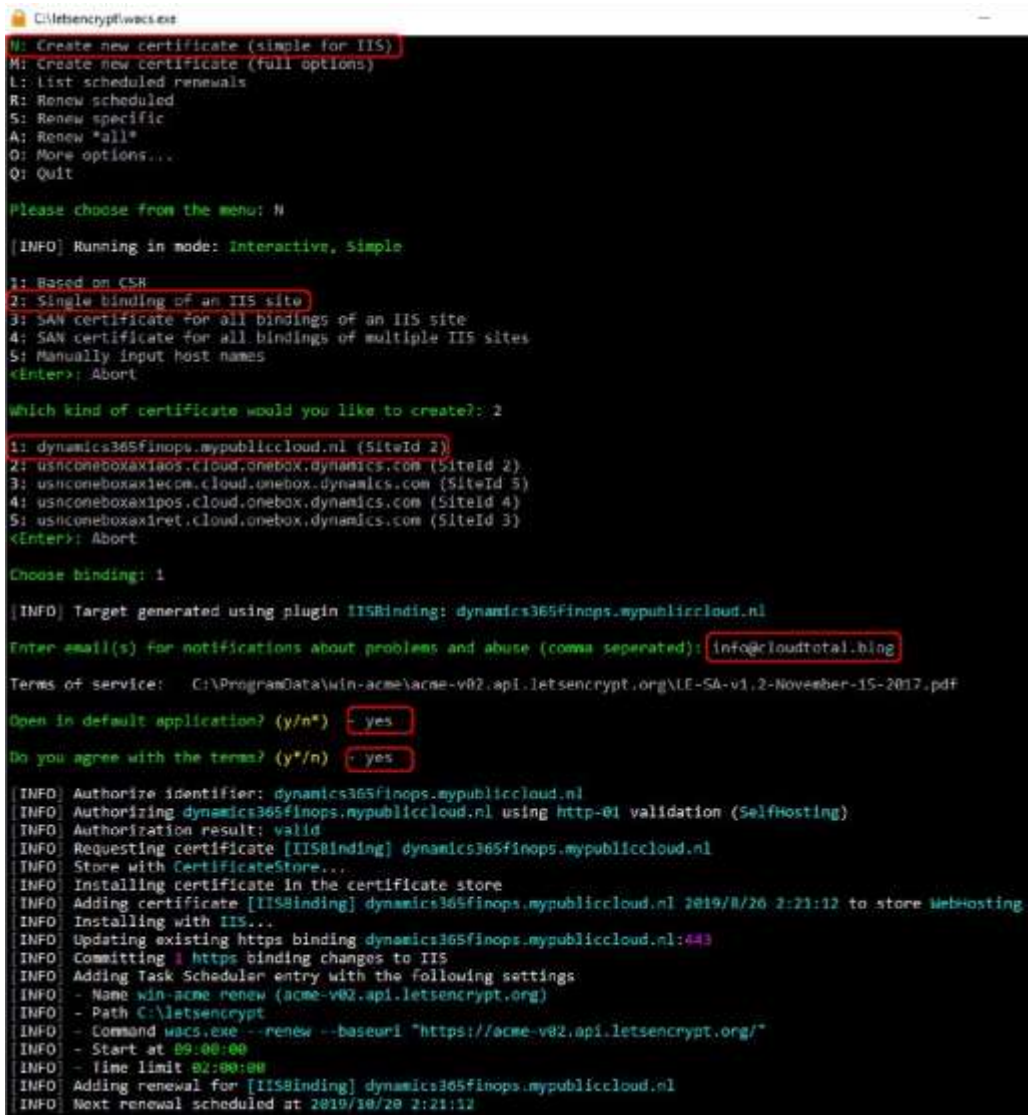
Step 5 – Create SSL certificate

- If you already have a valid signed web SSL certificate which is applicable for your D365 client URL, for example a wildcard *.domain.com, you can install it on the onebox VM and skip this step.

- If you don’t have such certificate, follow steps below to use LetsEncrypt to generate a valid certificate.

- In this tutorial we use WACS to leverage the LetsEncrypt certificate creation service. It’s easy and free to use, and also auto-configures IIS and scheduled tasks for periodic renewal.

- Example:



As you can see WACS has modified the AOSService binding and added the newly created SSL certificate. It also added a task in Windows Task Scheduler for auto renewal 😀 .

Step 6 – Authorize aadclient certificate in Azure AD

- Update 25-9-2019: Added this step to enable import users from Azure AD

- While you're still logged into the VM (onebox), open a PowerShell window with Run as administrator privileges.

- Execute the following command to export the (default deployed) SSL certificate which D365FO will use to authenticate to your Azure AD tenant on your behalf (for example to allow importing users)

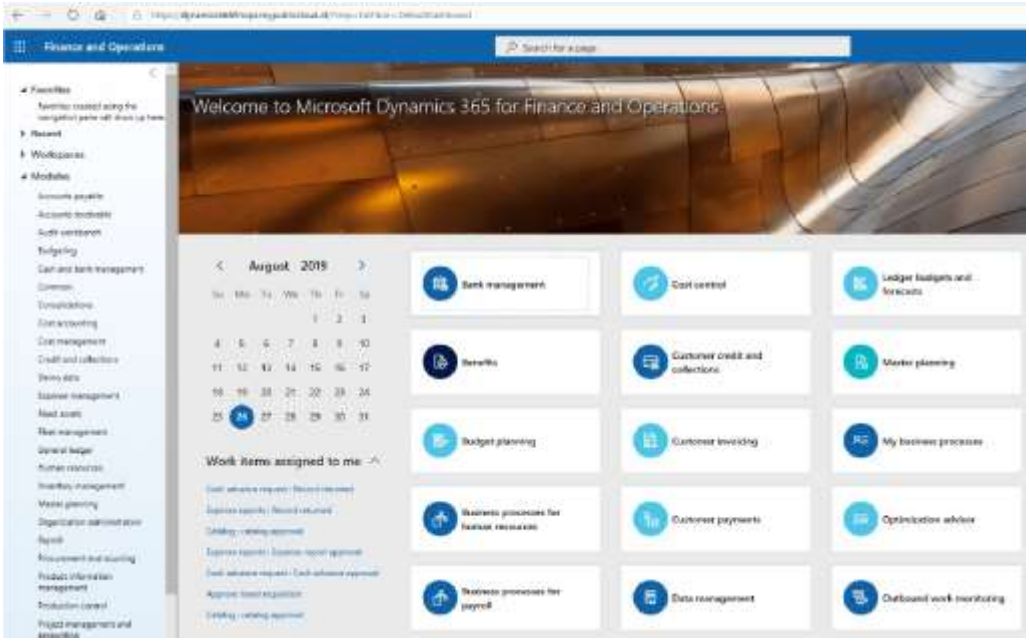
Get-ChildItem -Recurse -Path cert:\LocalMachine\Root ` | Where-Object {$_.Subject -like 'CN=aadclient.erp.ppe.dynamic
s.com'} ` | Export-Certificate -FilePath 'C:\aadclient.cer'

- Copy the CER file (created in root of volume C) to your computer.

- Go back to the Azure AD app registration portal (step 2), go to menu item Certificates & secrets and upload the CER file using the Upload certificate button.

Step 7 – Testing

- Open up a random browser and navigate to the new client URL. In this tutorial I've configured:
  https://dynamics365finops.mypubliccloud.nl/
- The Azure AD / Office 365 login prompt should appear.
  Log in using the same credentials as you have used in the Admin Provisioning Tool.
  The Dynamics 365 for Finance and Operations client should appear as displayed below.



- Try to import an Azure AD user via the Users panel (System Administration, page: SysUserInfoPage). If any error occurs when clicking Import users, verify the Azure AD app registration configuration.

  You can also check the Windows Event Viewer logs:
  Applications and Services Logs > Microsoft > Dynamics > MicrosoftDynamics-AX-SystemRuntime > Operational.
  Filter for any errors. If there are errors, they should contain diagnostic info about the authentication to Azure AD and Graph.

You're done! 😉