Fresh upgrade guide

Upgrade to v4.3

Updated 21 September 2022

FRESH

Content and Cloud Ltd.
contentandcloud.com | **freshintranet.com**

1 Prerequisites

1.1 Version of Fresh

Make sure the previous version of Fresh (at least Fresh 4.2.0) is installed in your tenant.

You can check which version of Fresh you are on in the App catalog following the steps described in 9.1 Check which version of Fresh you are on

If you do not have the latest version of Fresh deployed in your tenant, ask the Service Owner for the previous upgrade packages.

1.2 Install PnP.PowerShell module

The PnP.PowerShell module provide commands which are required by the upgrade script.

If not already installed, navigate to the following URL and install the latest PnP.PowerShell version.

https://pnp.github.io/powershell/

1.3 Azure CLI

Only if the provisioning module is deployed in your tenant, you will need to install Azure CLI to deploy the resources to Azure:
https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?view=azure-cli-latest

1.4 Set PowerShell execution policy

An execution policy is part of the PowerShell security strategy. Execution policies determine whether you can load configuration files, such as your PowerShell profile, or run scripts. In order to run scripts, you will need to set this policy and then, after the deployment is completed, set it back to the original value.

Open a new instance of the SharePoint Online Management Shell as an administrator (right-click, 'Run as Administrator). Enter the following command:

```
Get-ExecutionPolicy
```

Take note of the current execution policy as it should be returned to this value after the deployment is complete.

Enter the following command:

```
Set-ExecutionPolicy "Unrestricted"
```

When prompted to change the execution policy type "Yes"

This will allow PowerShell to execute scripts including those which have not been signed by a trusted publisher

2 Deployment account

The deployment steps in this document must be actioned using an account that has the required attributes. The deployment account must:

- The account should have Cloud authentication (i.e. not Federated authentication).
    - The deployment scripts will run faster if MFA is disabled for this user.
- Be a SharePoint administrator. This is provided by either of the following administrator roles:
    - 'Global administrator'
    - 'Customized administrator' with 'SharePoint administrator' rights
      ['Office 365 admin center' > USERS > Active Users > **select a user** > EDIT USER ROLES]
- Be a SharePoint term store administrator
  ['Office 365 admin center' > ADMIN CENTERS > SharePoint > term store > Term Store Administrators]
- Have English as the primary language
  ['About me' > 'Update profile' > 'Additional information' > 'Edit' > '...' > 'Language and Region']
- Have permissions to add solutions in the App Catalog.
  ['App catalog' > 'Site settings'].
- If the Provisioning module is deployed, the deployment account must have contributed permissions in the Resource group in Azure

Recommendation: Create a dedicated deployment account.

- Create a cloud-only account on the *onmicrosoft* domain associated with the tenant
- Do not assign a license
- Configure the account as a SharePoint administrator, term store administrator, and App catalog administrator.
- Once the deployment is complete disable sign-on for the account

Note that an account with higher privileges will be needed to approve SharePoint API permission requests and to grant admin consent for the permission that app registrations need.

3 PnP.PowerShell authentication

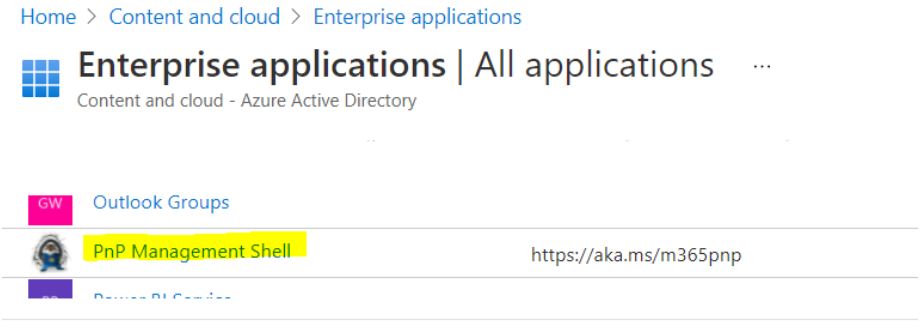
PnP.PowerShell allows you to authenticate with credentials to your tenant. However, due to changes in the underlying SDKs this PowerShell commands require to register an Azure AD Application which will allow to authenticate.

This action will create an Enterprise Application in the Azure Active Directory that will allow the PnP PowerShell script to run the commands in the tenant.

A **global administrator** is required to consent the permissions required by the enterprise application.

If this has been already done in the environment, it doesn't need to be done again.
The account that is used in the deployment needs to be added in the Enterprise Application.

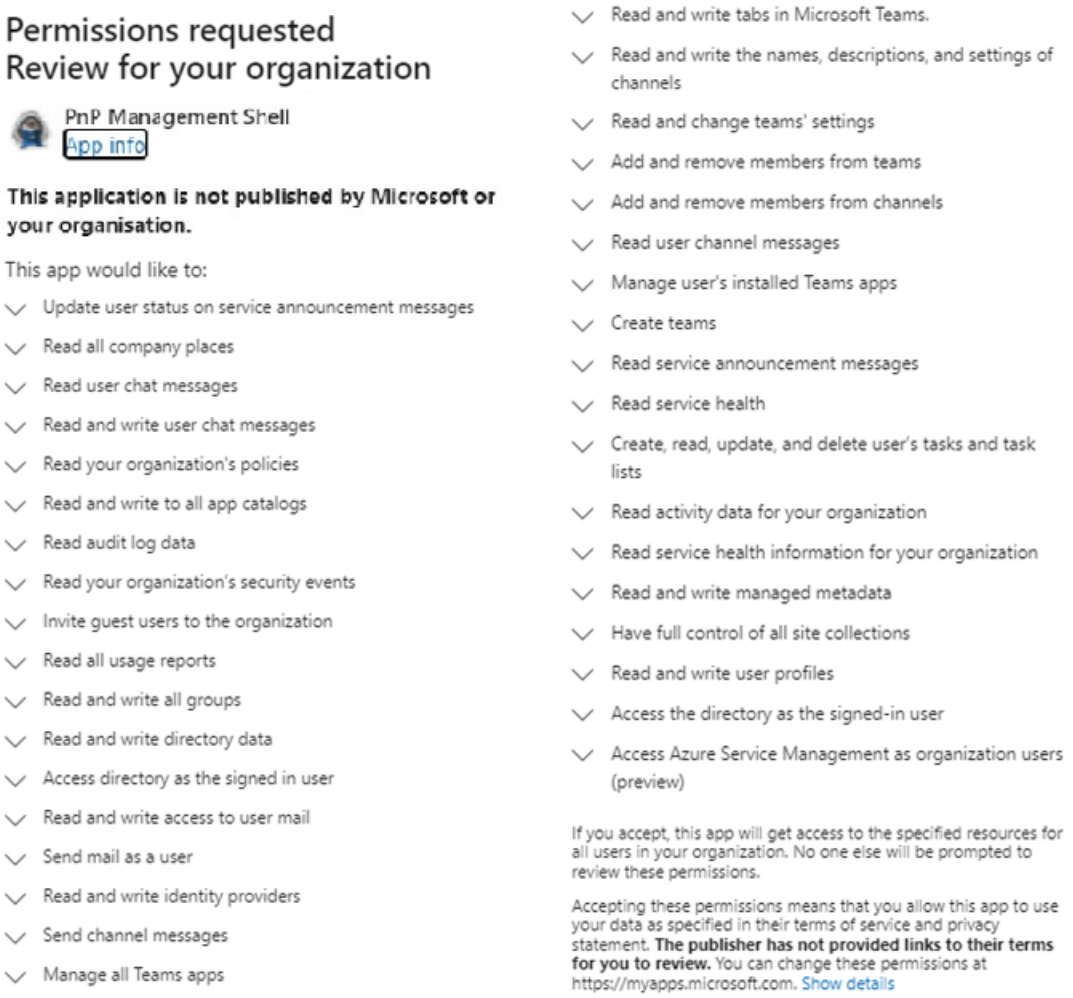
This is the Enterprise Application that will be created in the Azure Active Directory



You can read more about the PnP.PowerShell authentication in the following page: https://pnp.github.io/powershell/articles/authentication

3.1.1 Run the script

1. Open a new Windows PowerShell console as an administrator (right-click, 'Run as Administrator).
2. Run one of these commands
    - If the user executing the script is Global admin, enter the following command: `Register-PnPManagementShellAccess`
    - If the user executing the script is not Global admin enter the following command: `Register-PnPManagementShellAccess -ShowConsentUrl`. This command provides you with an URL you can share with a person with appropriate access rights to provide consent for the organization.
3. The following pop-up appears showing the permissions that need to be consent for a Global admin (or an administrator that can consent Azure AD Applications):



The Enterprise Application can be removed after the deployment it there are any concerns about security.

4 Configure deployment environment script

NOTE: Do not edit any of the deployment scripts beyond what is specifically mentioned here

Copy the provided deployment scripts folder locally.

At the root of the provided Scripts folder, find the folder called Config.

You should have a file for every environment. Use `dev.ps1`, for the development environment or duplicate it and rename it with the corresponding environment name (like uat.ps1 or prod.ps1).

If you do not have a config file from a previous deployment, open the provided file in a text editor such as VS Code or Notepad and update the following variables:

SharePoint settings:

- `$global:tenantUrl` The value assigned should be the URL of the root site collection and should end with ".sharepoint.com". The root site collection will not be a deployment target – this value is only used to build other URLs. Make sure the URL has no trailing slash.
- `$global:tenantAdminUrl` The value assigned should be the URL of the SharePoint Admin Centre. This value should end with "-admin.sharepoint.com"
- `$global:defaultSiteOwner` The value assigned should be the username of the account under which the deployment script will run.
- `$global:termStoreLanguages` Array with the codes of the languages.

Authentication settings:

- `$global:useWebLogin` Indicate if web login should be used. This should be true for Multi-factor authentication. If it is set to false, your credentials should be stored in a file.
- `$global:storedCredentialsPath` Only if `$useWebLogin` is set to false. Indicate the path of the credentials file. After editing and saving this configuration file, execute the script `Tools/Save-Credentials.ps1` to generate the credentials file.

Azure settings (not needed for a deployment without Azure components):

- `$global:subscriptionId` You can get the subscription id from https://portal.azure.com >Cost Management + Billing > Subscriptions.
- `$global:tenantDomain` Domain of the tenant. https://portal.azure.com > Azure Active Directory. (not needed for a deployment without Azure components)
- `$global:azureAssetPrefix` Azure Assets Prefix that was used in the Fresh deployment
- `$global:resourceGroupName` Resource group where the Fresh artefacts are located excluding the azureAssetPrefix
- `$global:resourceGroupLocation` Location of the resource group, like "uksouth".

Provisioning settings (Not needed for a deployment without Provisioning module):

- `$global:provisioningStorageAccountName` the name of the storage account used for provisioning
    - Go to portal.azure.com
    - In the top search bar, write "storage" and then click "Storage account".
    - Get the name of the storage account used in the provisioning process
- `$global:provisioningFunctionAppName` the name of the function used in the provisioning.
    - Go to portal.azure.com
    - In the top search bar, write "function app" and then click "Function app".
    - Get the name of the function for provisioning.

5 Configure instances script

If you have the file `Files\Instances.psm1` from a previous deployment, you can overwrite the provided file with it in the Files folder.

If you do not have the file `Files\Instances.psm1` from a previous deployment, edit the provided one and make sure all the Fresh instances are listed in the $instances array. For every instance, at least the properties InstanceStamp and InstancePrefix should be populated. For environments with only one instance the provided file should be good to use.

6 Microsoft 365 CDN issue

This only applies if the Microsoft 365 CDN is activated in your tenant.

To get general information about the Microsoft 365 CDN, read this article: Use the Office 365 Content Delivery Network (CDN) with SharePoint Online.

We have found an issue with the Microsoft CDN that only affects some users directly after deployment is done, where the error message "Something went wrong" is displayed to the user. This is related to the synchronization between the updated SPFx packages in the App Catalog and the CDN. The error message is removed if the user does a hard refresh in the browser (Control + F5).

This seems to be caused by the Microsoft 365 CDN taking extra time to refresh the reference to the new files that has been deployed and this cause an error in the pages.

To avoid this, we are deactivating the CDN in the tenant as the first step in the deployment. We recommend this action to avoid the errors. We are actively researching about this problem and when we have more information, and the issue is solved we will contact you to activate the CDN again. The script to enable the CDN in the tenant is included in the Tools folder: `Enable-Office365Cdn.ps1`

If you feel comfortable deploying the upgrade with the CDN activated, this step can be skipped

To disable the CDN (public and private types):

6.1.1 Execute script

4. Update the first line of the `Environment.ps1` file to point to the corresponding environment file.
5. Open a new instance of the SharePoint Online Management Shell as an administrator (right-click, 'Run as Administrator).
6. Navigate to the local location of the Upgrade deployment scripts folder. E.g., cd C:\Deployment\Scripts\Tools
7. Enter the following command: `.\Disable-Office365Cdn.ps1`
8. If you are using WebLogin, when prompted, enter the username and password.

7 Deployment

7.1 Tenant updates

The upgrade script will perform the following steps:

a. Update the following packages in the app catalog
   - candc-fresh-globalstructure.sppkg
   - candc-fresh-contentwebparts.sppkg
   - candc-fresh-intranetwebparts.sppkg
   - candc-fresh-coredwp.sppkg
   - candc-fresh-governanceadmin.sppkg
   - candc-fresh-usercentricwebparts.sppkg
   - candc-fresh-toolsadmin.sppkg
   - candc-fresh-corelibrary.sppkg
   - candc-fresh-dashboard.sppkg
   - candc-fresh-coreweb.sppkg
   - candc-fresh-media.sppkg
   - candc-fresh-governance.sppkg

b. Depending on the modules that are deployed, these packages will be updated as well
   - candc-fresh-provisioningwebparts.sppkg
   - candc-fresh-appchecklist.sppkg
   - candc-fresh-webanalytics.sppkg

c. Create new managed properties
   - FreshMobileOnly
   - FreshDesktopOnly
   - FreshWhiteText

7.1.1 Prepare the scripts

1. Update the first line of the `Environment.ps1` file to point to the corresponding environment file.
2. Open the `Upgrade\Upgrade-Fresh4.3-deployapps.ps1` file
3. Edit the line 25 to indicate the optional modules deployed in your environment:
   - Add `-Provisioning` parameter if the Provisioning is deployed in the environment
   - Add `-WebAnalytics` parameter if the Analytics solution is deployed in the environment
   - Add `-CheckList` parameter if the Checklist solution is deployed in the environment

   The line 25 should be like this (in case the 4 packages are deployed in your tenant):
   `Update-Solutions -Provisioning -WebAnalytics –CheckList`

If none of these modules are installed in the environment the line 25 should be: `Update-Solutions`

7.1.2 Execute script

1. Open a new Windows PowerShell console as an administrator (right-click, 'Run as Administrator).
2. Navigate to the local location of the `Upgrade` deployment scripts folder. E.g., `cd C:\Deployment\Scripts\Upgrade`
3. Enter the following command: `.\Upgrade-Fresh4.3-deployapps.ps1`
4. If you are using WebLogin, when prompted, enter the username and password.

Note that after the script is complete it is expected to see the version **4.3.0.0** in the Fresh packages in the app catalog.

If there is an error in step 2, with this message: **Set-PnPSearchConfiguration: Internal Server Error**, this is because there are some issues in SharePoint Search. We can perform this step manually, but first section 7.2 needs to be completed. Once the steps in 7.2 are completed, the manual steps to complete can be found in section 9.5.

7.2 Fresh site collections updates

For this step, the correct Instance file is needed in the Files folder. See section 5 of this document.

The upgrade script will perform the following steps in every site:

a. Update the "Fresh - Media features" app.

b. If the Tools list exists in the site, add new columns to the Tools list:
   - Mobile only
   - Desktop only
   - Use white text

7.2.1 Execute script

This script takes around 3 seconds per site.

1. Open a new Windows PowerShell console as an administrator (right-click, 'Run as Administrator).
2. Navigate to the local location of the `Upgrade` deployment scripts folder. E.g. `cd C:\Deployment\Scripts\Upgrade`
3. Enter the following command: `.\Upgrade-Fresh4.3-updatesites.ps1`
4. If you are using WebLogin, when prompted, enter the username and password.

7.3 Provisioning module

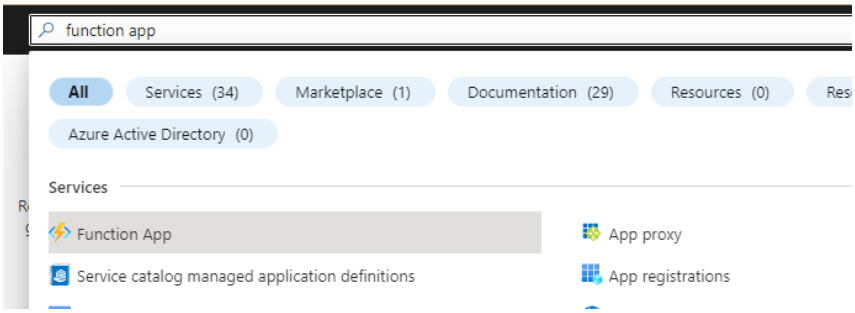
Follow these steps only if the provisioning module is deployed in your tenant.

The upgrade script will perform the following steps:

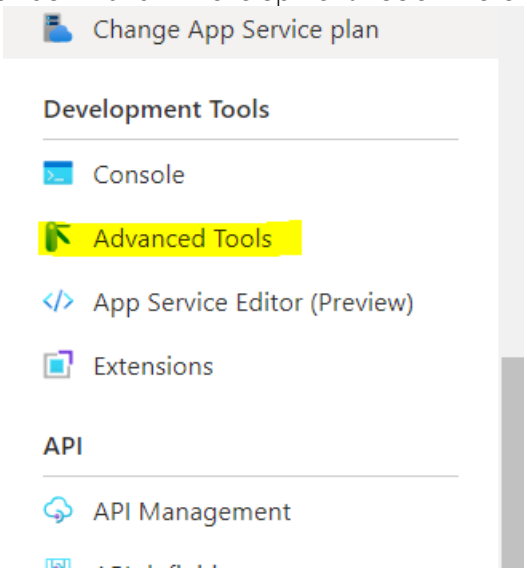
- Update the provisioning module to use the latest .NET Core framework
- Upload the "Communities site" template to properly configure webpart in homepage

7.3.1 Create a back-up of the templates

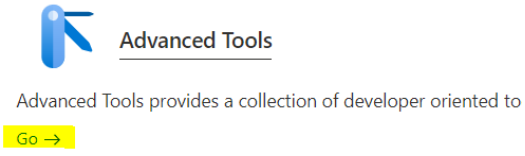
1. Go to the Azure portal: https://portal.azure.com
2. In the top search box, write "Function app" and click in "Function App" in the opening panel.



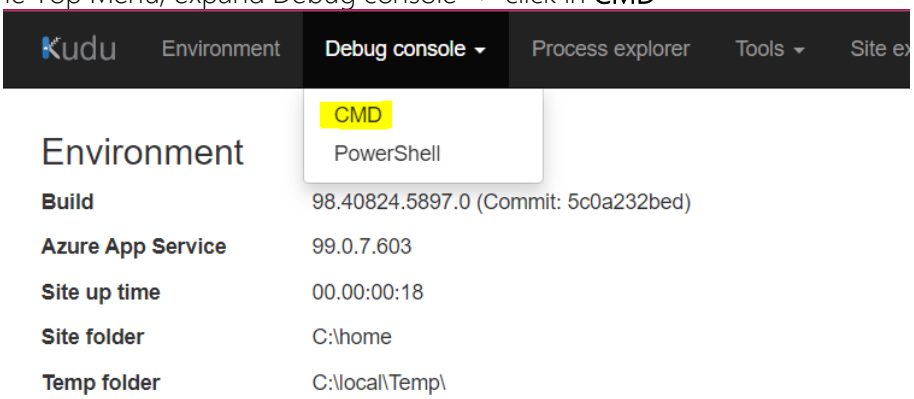
3. Select the provisioning function app. It finishes with "-provisioningapp"
4. Scroll down and in Development Tools -> click in Advanced Tools



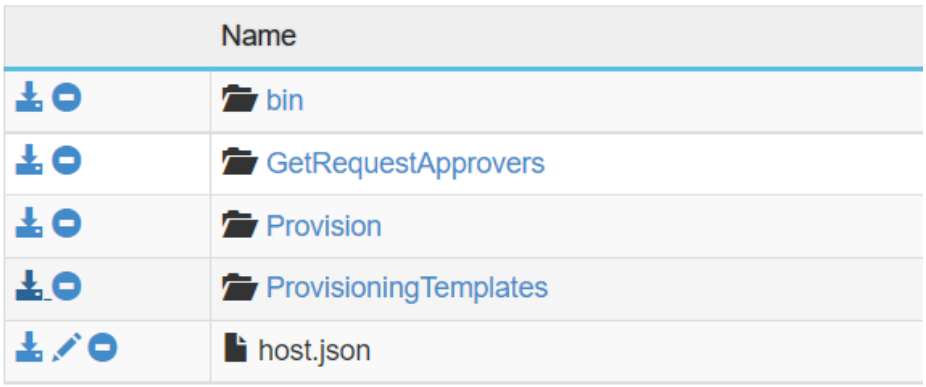
5. Click in Go



6. In the Top Menu, expand Debug console -> click in **CMD**



7. Navigate to **site** and then **wwwroot**



8. Download the **ProvisioningTemplates** folder

This action will download the **ProvisioningTemplates.zip** file with all the provisioning templates. Please keep this file in a safe location just in case some of the files need to be restored.

7.3.2 Execute script

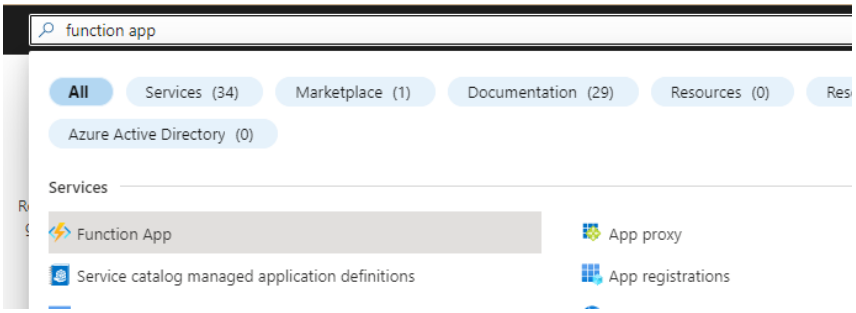
1. Open a new Windows PowerShell console as an administrator (right-click, 'Run as Administrator).
2. Navigate to the local location of the `Upgrade` deployment scripts folder. E.g. `cd C:\Deployment\Scripts\Upgrade`
3. Enter the following command: `.\Upgrade-Fresh4.3-provisioning.ps1`
4. If you are using WebLogin, when prompted, enter the username and password.

Note that this step can be done manually. You can upload the libraries and the templates into the Azure storage, as it is described in section 9.4.

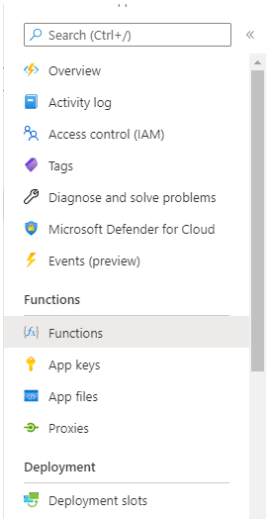
7.3.3 Manual steps – update Flow

After the provisioning module has been updated, the provisioning Flow needs to be edited to update the URL of the function app.

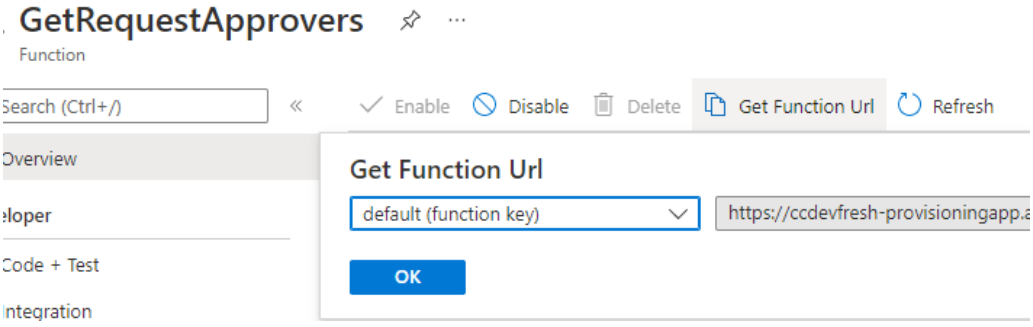
1. Get the function URL
    a. Go to the Azure portal: https://portal.azure.com
    b. In the top search box, write "Function app" and click in "Function App" in the opening panel.



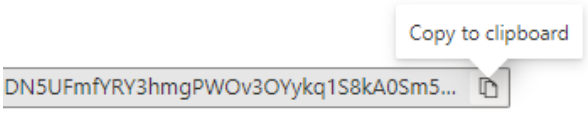
    c. Select the provisioning function app. It finishes with "-provisioningapp"
    d. Click "Functions" form the lateral navigation



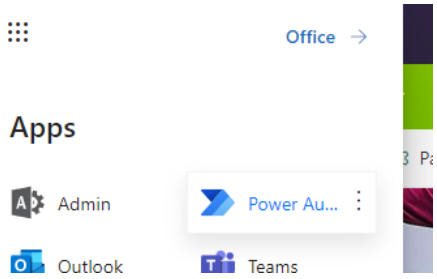
    e. Click GetRequestApprovers
    f. Click "Get Function Url" (it might take a few seconds to get enabled)



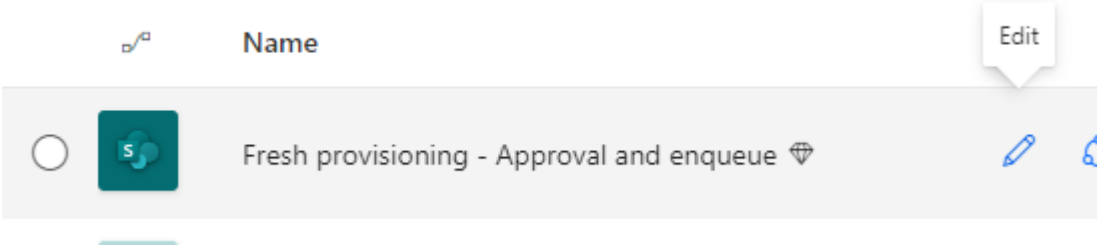
g. Copy the URL by clicking in the icon at the end of the text box



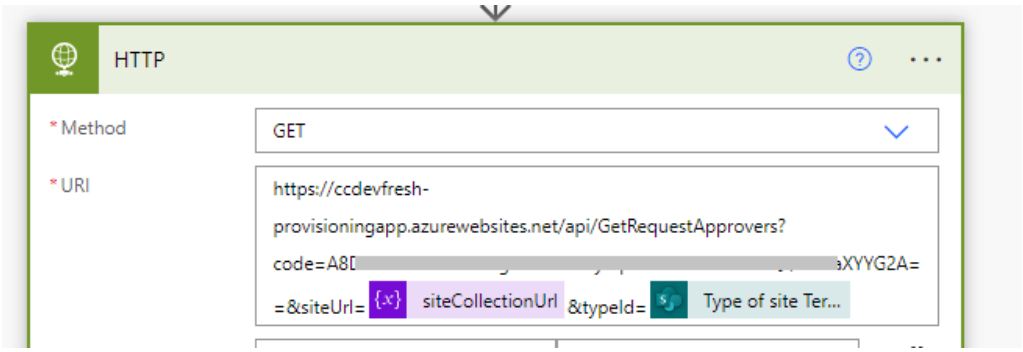
2. Update the URL in the flow
   a. Go to Power Automate



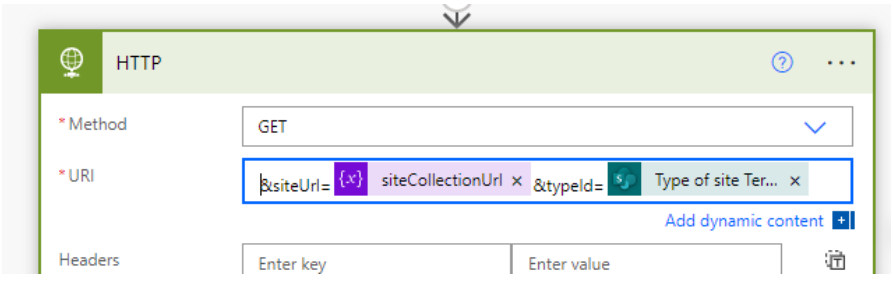
   b. Click "My flows" from the lateral navigation
   c. Click in the pencil button to edit the "Fresh provisioning - Approval and enqueue" flow



   d. Click in the green box called "HTTP" to expand it



   e. Remove all the text that appears before "&siteUrl"



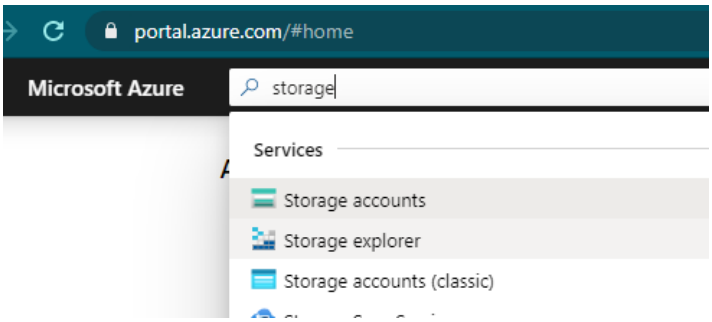
   f. Paste the copied URL in step 1.g at the beginning of the text box

g. Save the flow

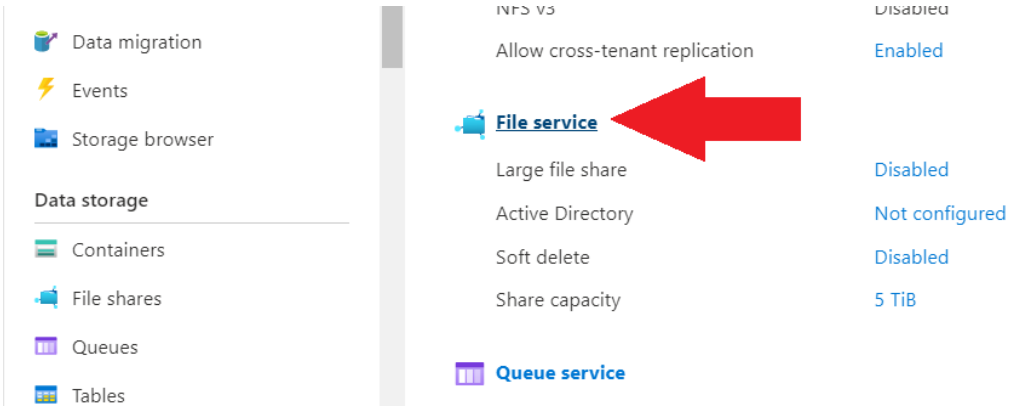
7.3.4 Manual steps – update templates

Due to a bug in the latest version of PnP applying templates with the old PnP schema, the schema of the templates needs to be updated.

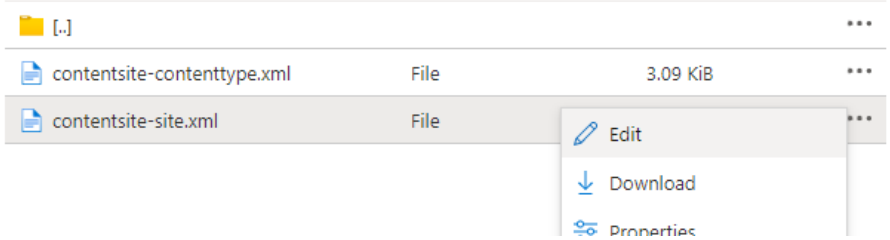
1. Go to the Azure portal: https://portal.azure.com
2. In the top search box, write "storage" and then click in "Storage Accounts":



3. Click in the Fresh provisioning storage account
4. Click "File shares":



5. Click the provisioning File shares.
6. Click on the "site" folder. Then the "wwwroot" folder. Then click the "ProvisioningTemplates" folder.
7. Note that the ProvisioningTemplates folder contains a subfolder for every entity with the PnP templates. For every subfolder:
    a. Check if any of the files ends with "-site.xml".
    b. Click on the ellipsis of the file that ends with "-site.xml" and select Edit



    c. Update the year and month in the second line to 2021/03
    `<pnp:Provisioning`

```
xmlns:pnp="http://schemas.dev.office.com/PnP/2021/03/ProvisioningSchema
">
```

d. Click Save

7.4 NEW Site template module (OPTIONAL)

This module is meant to be used in implementations of Fresh with only one instance.

This module requires that the Instances file (`Files\Instances.psm1`) has the properties TopNavigationTermSetGuid, FooterNavigationTermSetGuid and FreshTaxParameters defining all the term set ids and taxonomy column names

```
InstanceStamp                = "fresh"
InstancePrefix               = "fresh-" # Prefix is mandatory to differenciate Fre
TopNavigationTermSetGuid     = "FF3BFC6D-63F7-4B8C-B8A3-F0E422403D00"
FooterNavigationTermSetGuid  = "FF090690-00E9-429B-B88D-0C26BF2F4A99"
NavigationFilePath           = "$PSScriptRoot\..\Files\Navigation\taxonomy.xml"
```

```
FreshTaxParameters           = @{
  GroupName             = "Fresh"
  GroupGuid             = "6DD67EE8-351D-4A19-8502-7A143D731CBC"
  FreshTax1FieldName    = "Location"
  FreshTax1TermSetGuid  = "FFD99857-61E1-48B4-8DF8-8D66EE64C601"
  FreshTax2FieldName    = "Department"
  FreshTax2TermSetGuid  = "FFAF7768-0D89-411D-923F-5B3C181C5602"
  FreshTax3FieldName    = "Function"
  FreshTax3TermSetGuid  = "FF8B7FB8-8D6C-4FB5-9C4B-EFE4C0FD0B03"
  FreshTax4FieldName    = "Topic"
  FreshTax4TermSetGuid  = "FFE3F6BA-F8BF-48B7-BA86-AC84DA04C904"
  FreshTax5FieldName    = "Audience"
  FreshTax5TermSetGuid  = "FF4846B2-2B74-4902-9AD5-97C19210A205"
  FreshTax6FieldName    = "Language"
  FreshTax6TermSetGuid  = "FF7CFCE7-EAE6-4C60-8D46-851DD1234F06"
  FreshTax7FieldName    = "Type of content"
  FreshTax7TermSetGuid  = "FFD3AF59-F51E-4CD5-9071-609359012C07"
  FreshTax8FieldName    = "Action type"
  FreshTax8TermSetGuid  = "FF3CFFF5-1738-46B7-A743-BFA84C66EB08"
}
```

Fresh provides a new module that adds some Site templates (previously known as Site designs) to your tenant. You can read more about site templates in the SharePoint site template and site script overview.

7.4.1 Prepare the scripts

1. Open the `Deploy-SiteTemplates\Deploy.ps1` file
2. Edit the line 19 to indicate the site templates to deploy:
   - Add `"Apps"` to deploy a site template that install the Fresh apps in the site.
   - Add `"Navigation"` to deploy a site template that add the Fresh top navigation and footer to the site.

- Add `"ContentTypes"` to deploy a site template that add the Fresh columns and content types to the site.

3. Save the file.

7.4.2 Execute script

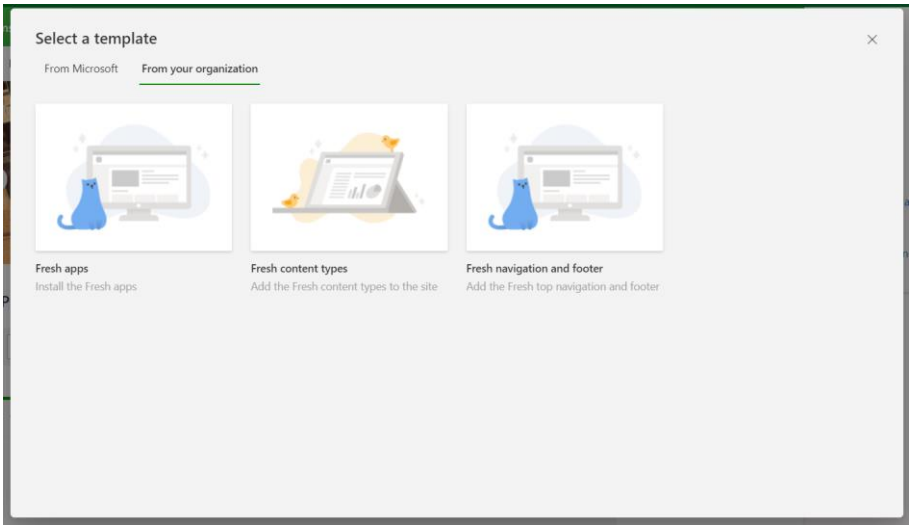
1. Open a new Windows PowerShell console as an administrator (right-click, 'Run as Administrator).
2. Navigate to the local location of the `Deploy-SiteTemplates` deployment scripts folder. E.g., `cd C:\Deployment\Scripts\Deploy-SiteTemplates`
3. Enter the following command: `.\Deploy.ps1`
4. If you are using WebLogin, when prompted, enter the username and password.

7.4.3 Test the deployment

After the deployment has been completed successfully new site templates should be available to be applied for the site collections in the tenant.

To check this:

1. Go to a site collection
2. Click in the cog and choose "Apply a site template"
3. In "From your organization" tab, the following site templates should be available
    a. Fresh apps
    b. Fresh content types
    c. Fresh navigation and footer



8 Rollback plan

To come back to the previous version, please follow these steps

1. Find the previous version of the packages and deploy them to the App catalog.

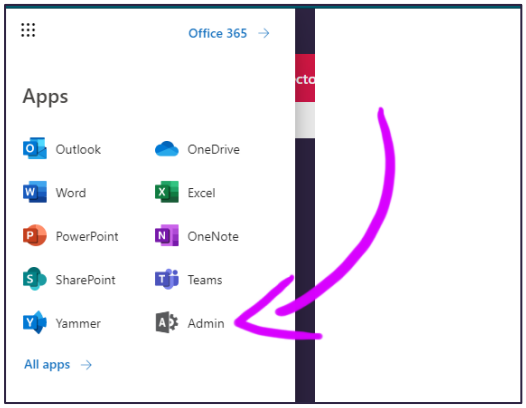
9 Troubleshooting

9.1 Check which version of Fresh you are on

You can check the version of Fresh in the App Catalog. Please follow these steps:

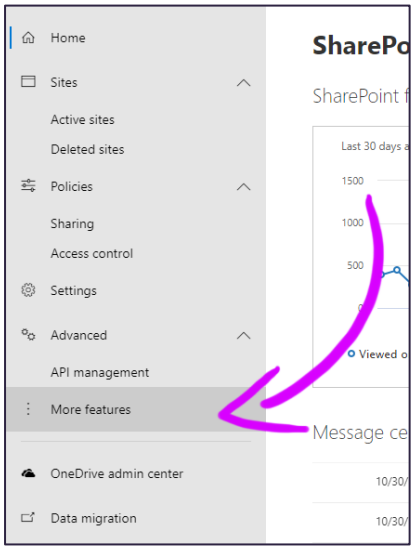
1. To go to the app catalog:

    a. Click the waffle and then click Admin

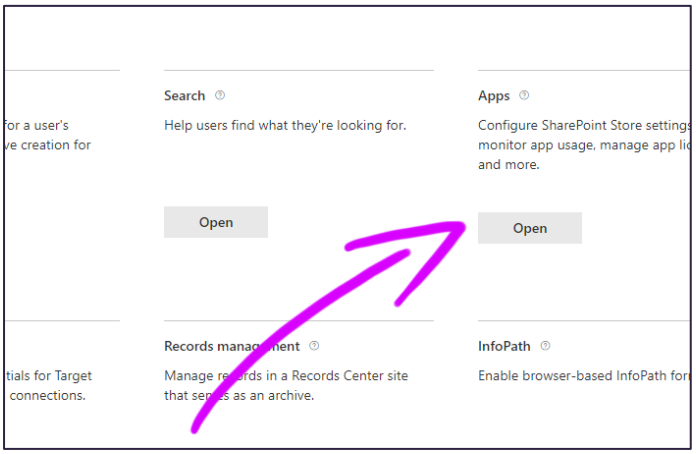


    b. In the left panel, expand "Show all" and click "SharePoint" under "Admin centers"

    c. In the left panel, click "More features"

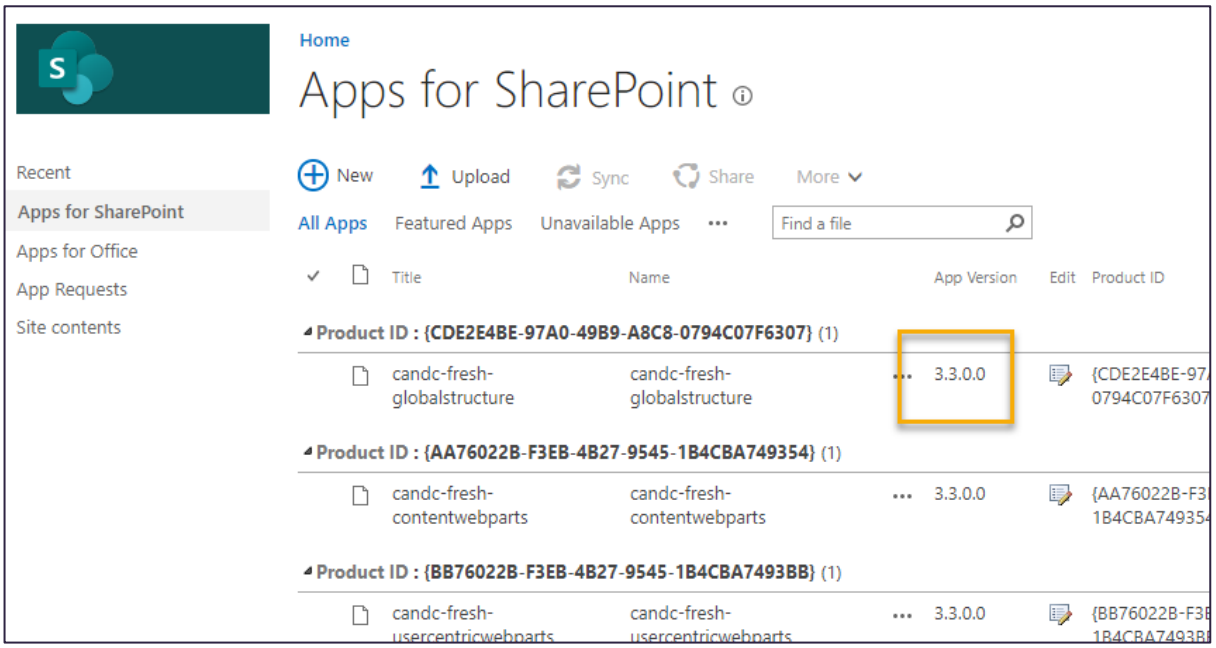


    d. Click the "Open" button under "Apps"



e. Click the link "App catalog".
f. In the left menu, click "Apps for SharePoint"

2. Check the "App version" column of any of the Fresh solutions

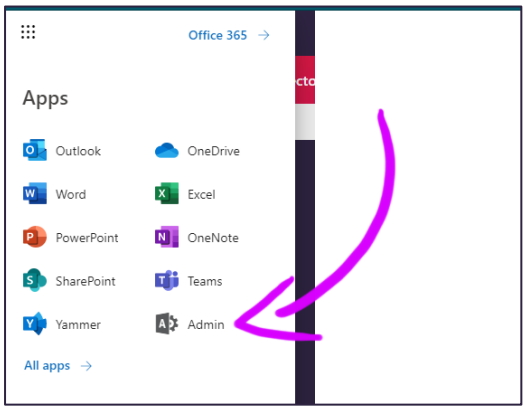


9.2 Manually upload the packages

If for some reason there is a problem uploading the packages to the App Catalog, this step can be done manually following these steps:

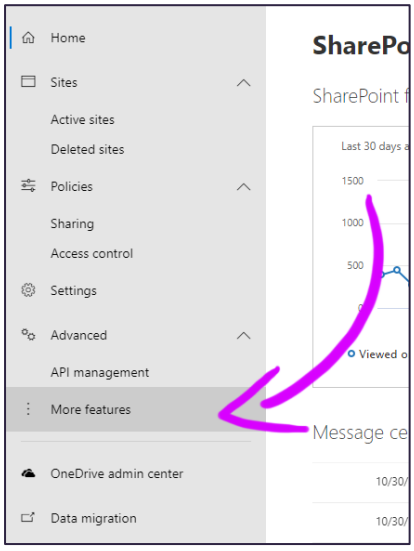
1. Go to the app catalog:

    a. Click the waffle and then click Admin

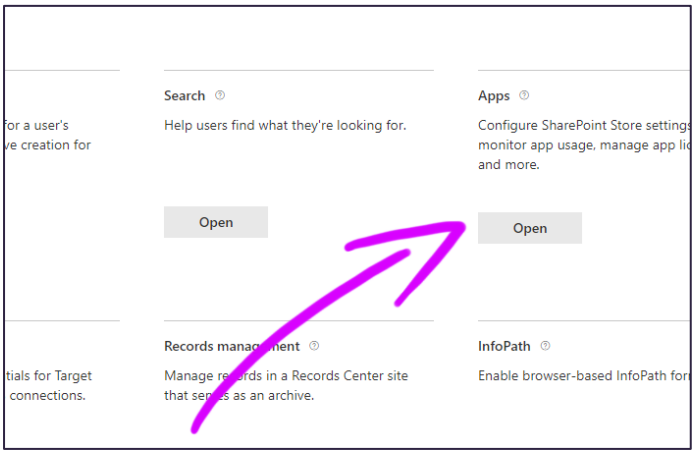


    b. In the left panel, expand "Show all" and click "SharePoint" under "Admin centers"

    c. In the left panel, click "More features"

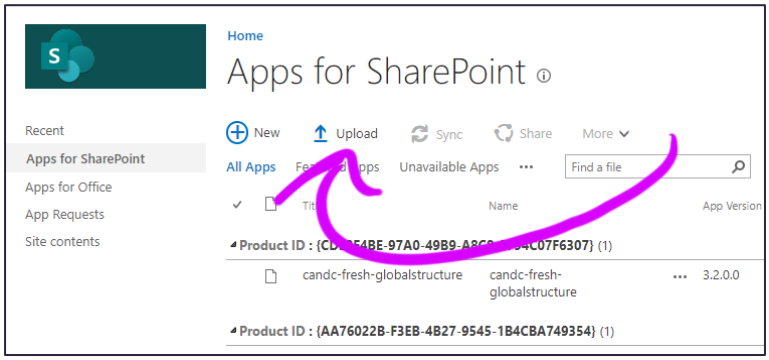


    d. Click the "Open" button under "Apps"



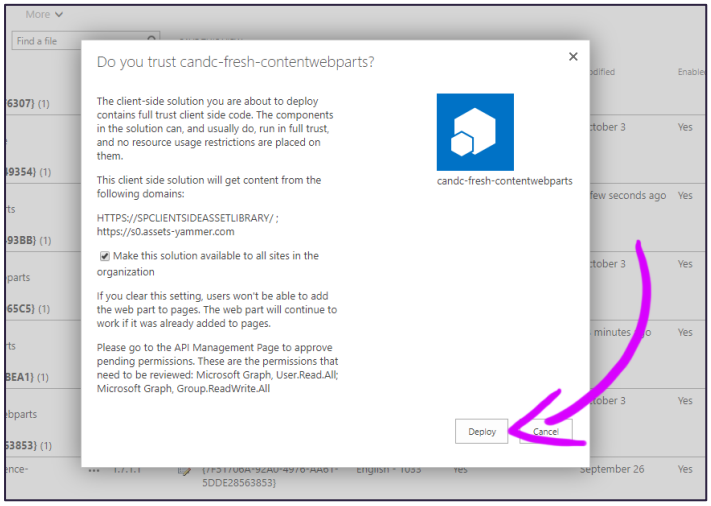
e. Click the link "App catalog".
f. In the left menu, click "Apps for SharePoint"

2. Upload the 6 solutions provided in the folder Scripts\SolutionPackages.
For every solution:
a. Click the "Upload button" at the top of the page.



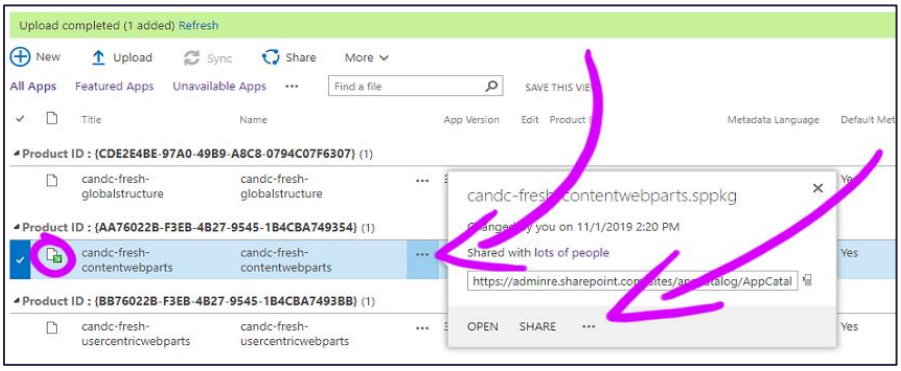
b. In the pop-up, select the solution and click "OK".

c. When prompted, click always "Deploy".



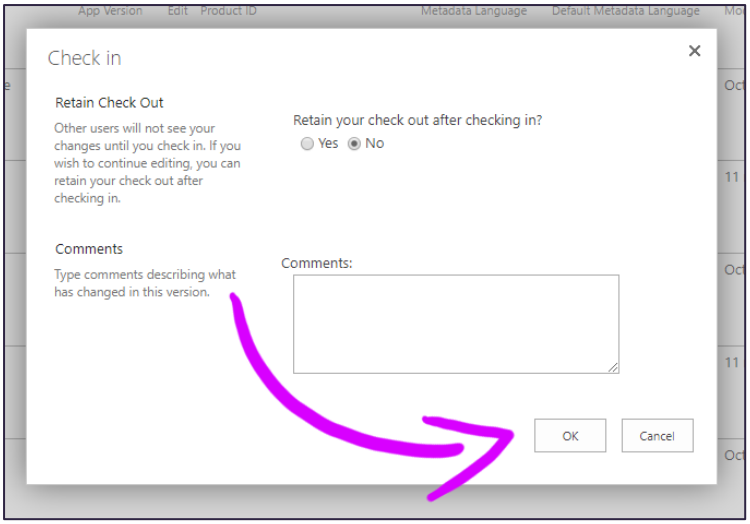
d. If the solution appears with a green box with an arrow inside (check out status), you need to check it in. To do that:

i. Open the contextual menu by clicking in the ellipsis "...". In the dialog that appears, click the ellipsis "...".



ii. In the menu select "Advanced" and then "Check-in".

iii. In the pop-up, click "OK"

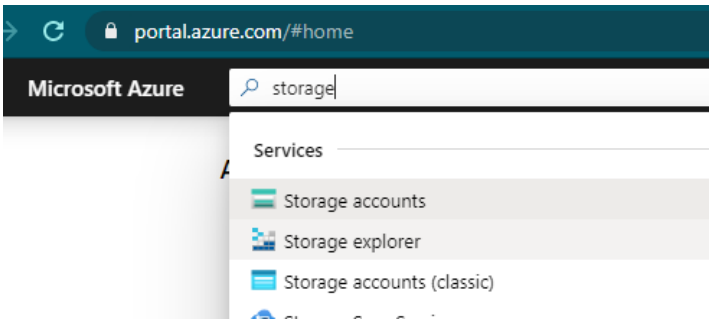


9.3 Access to the Provisioning files

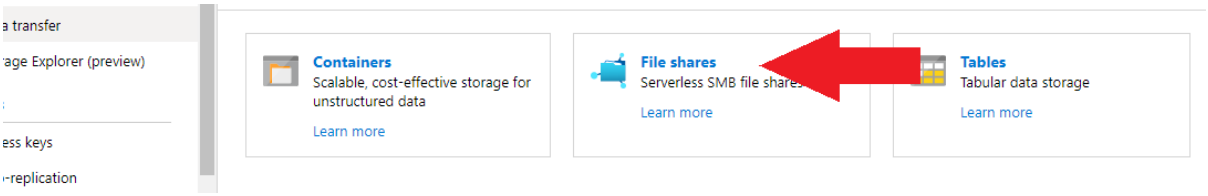
This section describes how to access files to the Provisioning engine through the Azure portal. This method does not require that you install any additional tool on your computer. Alternatively, you could consider using Azure Storage Explorer.

Please follow these steps to access to the Provisioning engine files through the Azure portal:

8. Log in to the Azure portal: https://portal.azure.com/
9. In the top search box write "storage" and then click in "Storage Accounts":



10. Click in the provisioning storage account
11. Click "File shares":



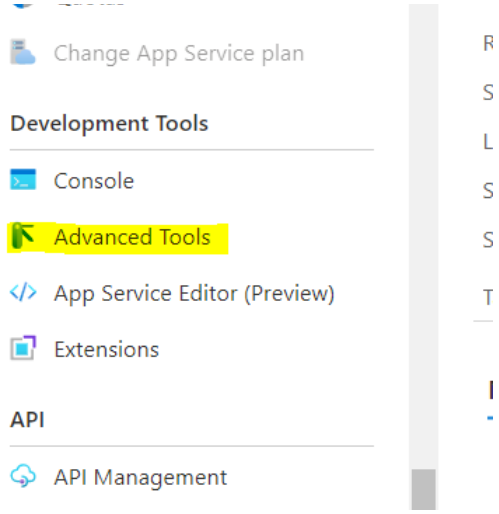
12. Click the first item in the list.
13. Click on the "site" folder. Then the "wwwroot" folder.
    a. Note that the `bin` folder contains the libraries (dll) that are needed to execute the provisioning engine.
    b. Note that the `ProvisioningTemplates` folder contains the PnP templates that can be customized.

9.4 Manually upload the Provisioning files

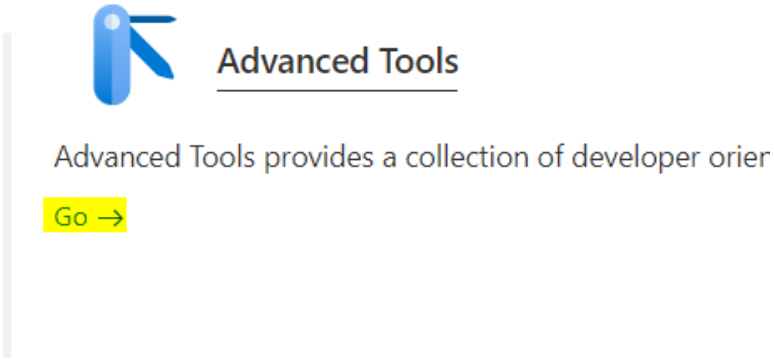
If the script to update the Fresh provisioning engine cannot be execute due proxy restrictions, we can do the steps manually

9.4.1 Manually upload the new function files

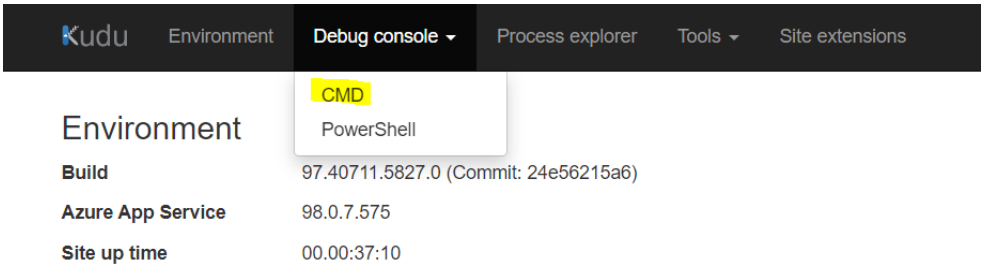
- Go to the Provisioning app function in Azure
- In the Quick Launch menu go to Development Tools -> Advance Tools



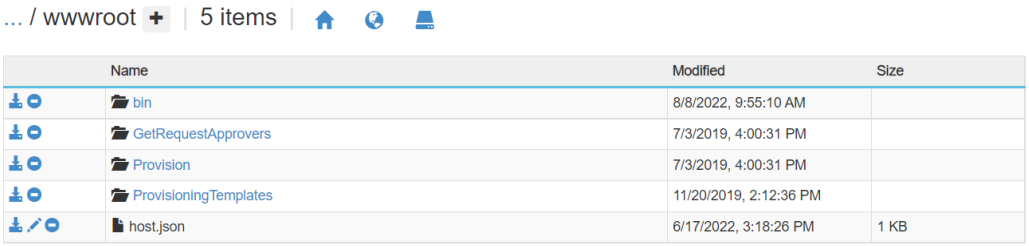
- Click in Go



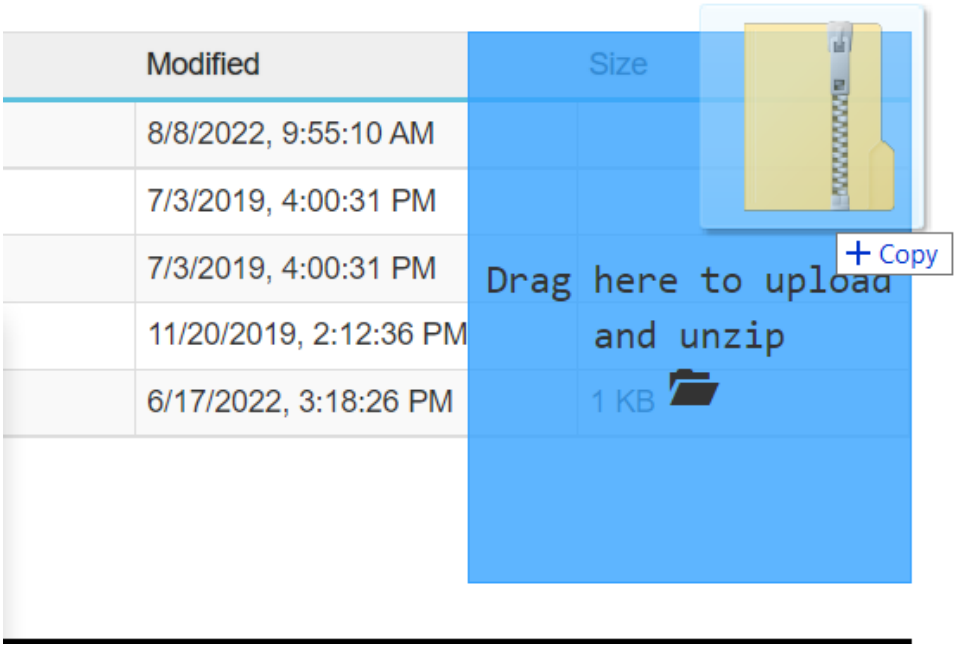
- In the Top Menu, click in Debug console -> CMD



- Go to the wwwroot folder (site -> wwwroot)



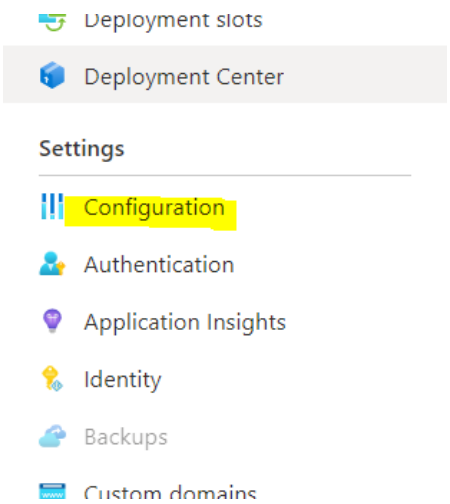
- Drag the zip file CandC.ProvisioningFunctions.zip (in the Fresh upgrade package, in the folder Upgrade -> 4.3) onto the right side of the screen (in the Size column of the grid). Doing this, the zip file will be automatically unzipped, and the files placed in the right folders. If the zip is dragged in the main area on the screen, the zip will not be unzipped.



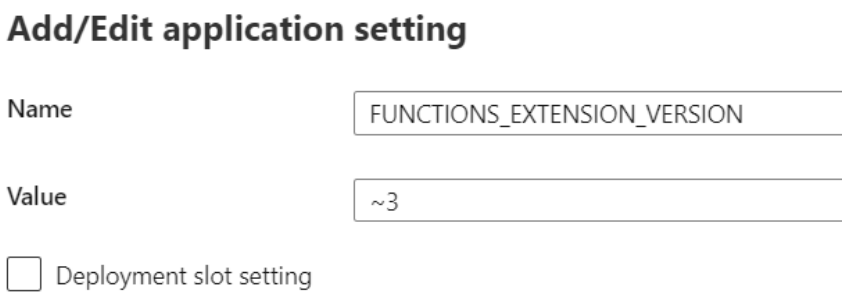
- The function files will be updated with this process

9.4.2 Upgrade Framework settings for the function app

- Go to the Provisioning app function in Azure
- In Settings click in Configuration



- Click in FUNCTIONS_EXTENSION_VERSION and set the variable to: ~3. It should look like this:



- Save the changes in the configuration.

9.4.3 Upgrade the community site template

The PnP template that contains the community site settings has been updated in this upgrade. These templates need to be overwritten in the storage. Check the section 9.3 to access to the storage. In the `wwwroot` folder, click in `ProvisioningTemplates` and overwrite the following file:

In the `CC.Community.PnP` overwrite the community-site.xml file (this file is located in the Fresh upgrade package, in the folder Upgrade / 4.3 / CC.Community.PnP folder.

9.5 Create Search managed properties manually

If there is a problem importing the search managed properties (in this case in the step 2 of the Upgrade-Fresh4.3-deployapps.ps1 script this can be because SharePoint is having issues with Search (usually there will be a message in the MS Health Centre related with Search).

If it is possible, it will always be better to wait until the service is restored, but if the deployment needs to be completed, the Search managed properties can be created manually.

To create the Search managed properties manually we need to have run the Upgrade-Fresh4.3-updatesites.ps1 and make sure that the following fields have been added to the Tools list:

- Use white text
- Mobile only
- Desktop only

9.5.1 Add or modify an item in the Tool list

To trigger the search engine and make sure the crawl properties are created, first thing that need to be done is add a value in each of the fields that have been created in the Tool list

We can add a new item or update an existing one, but we need to make sure that we add a value in each of these fields (it is enough to do it in one item)

- Use white text (internal name: CandC_WhiteText)
- Mobile only (internal name: CandC_MobileOnly)
- Desktop only (internal name: CandC_DesktopOnly)

After that, at least 15 minutes wait will be needed, to make sure the crawl properties are created.

9.5.2 Create and map the Search managed properties

Now we need to create the new Search managed properties and map the crawl properties

1. Go to the SharePoint Admin Centre
2. Go to Search -> Manage Search Schema
3. Clicking in New Managed Property create these 3 ones
    a. FreshWhiteText
        i. Type: Yes/No
        ii. Queryable
        iii. Retrievable
        iv. Safe
        v. Token Normalization
        vi. Mapped to: ows_CandC_WhiteText

PROPERTY NAME	TYPE	MULTI	QUERY	SEARCH	RETRIEVE	REFINE	SORT	SAFE	MAPPED CRAWLED PROPERTIES	ALIASES
FreshWhiteText	Yes/No	-	Query	-	Retrieve	-	-	Safe	ows_CandC_WhiteText	

b. FreshMobileOnly
    i. Type: Yes/No
    ii. Queryable
    iii. Retrievable
    iv. Safe
    v. Token Normalization
    vi. Mapped to: ows_CandC_MobileOnly

PROPERTY NAME	TYPE	MULTI	QUERY	SEARCH	RETRIEVE	REFINE	SORT	SAFE	MAPPED CRAWLED PROPERTIES	ALIASES
FreshMobileOnly	Yes/No	-	Query	-	Retrieve	-	-	Safe	ows_CandC_MobileOnly	

a. FreshDesktopOnly
    vii. Type: Yes/No
    viii. Queryable
    ix. Retrievable
    x. Safe
    xi. Token Normalization
    xii. Mapped to: ows_CandC_DesktopOnly

PROPERTY NAME	TYPE	MULTI	QUERY	SEARCH	RETRIEVE	REFINE	SORT	SAFE	MAPPED CRAWLED PROPERTIES	ALIASES
FreshDesktopOnly	Yes/No	-	Query	-	Retrieve	-	-	Safe	ows_CandC_DesktopOnly	