TŌGŌ
A social digital workplace
by NTT DATA

Initial setup - Installation Guide

NTT Data, November 2023

2023

Last updated: November 10th, 2023

CONTENTS

TŌGŌ INITIAL SETUP

> TŌGŌ is a Social Digital Workplace solution that extends Microsoft 365 capabilities to create an easy-to-use and integrated social experience on top of the Microsoft 365 out-of-the-box suite.

Goal

The purpose of this document is to provide advice on how to successfully set up a TŌGŌ Fast Track, to help TŌGŌ customers with a streamlined process that defines the required activities and to assist you during the setup process.

Organization

The structure of this document is based on the following sections:

TŌGŌ Setup Process Overview

Provides an overview of the NTT DATA setup process, identifying the main activities and the expected results for each activity.

Planning

This section describes the activities in the planning phase of the setup process. And details the recommended approach for this phase.

Preparation

Describes the required environment preparation for a TŌGŌ setup, including the required subscription acquisition in Microsoft 365 and Azure, and the required Microsoft Entra ID setup to support access to the required tenant.

Deployment

Describes the step-by-step process required to deploy a TŌGŌ Fast Track.

Intended Audience

This document is intended to support the following audiences:

Project Managers

Assists the project manager in comprehending the necessary steps for a successful deployment of TŌGŌ, including all preparation procedures.

Operations Managers

Enables the Operations Manager to plan a TŌGŌ deployment and determine the necessary cloud infrastructure platform requirements.

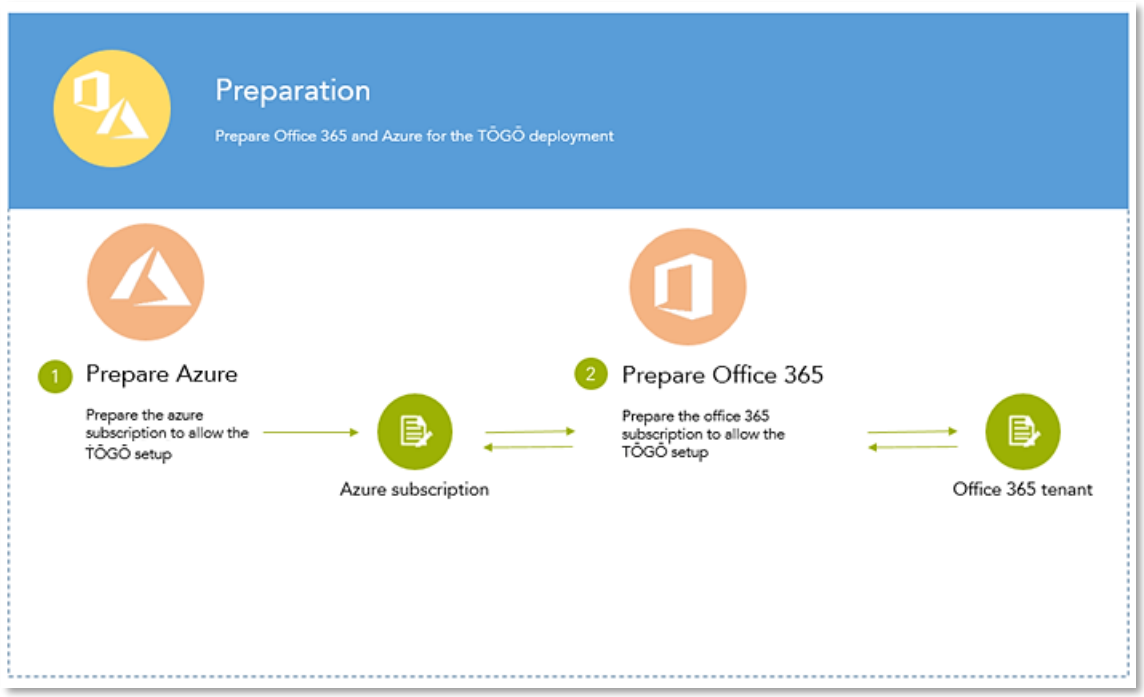
Operations Team Members

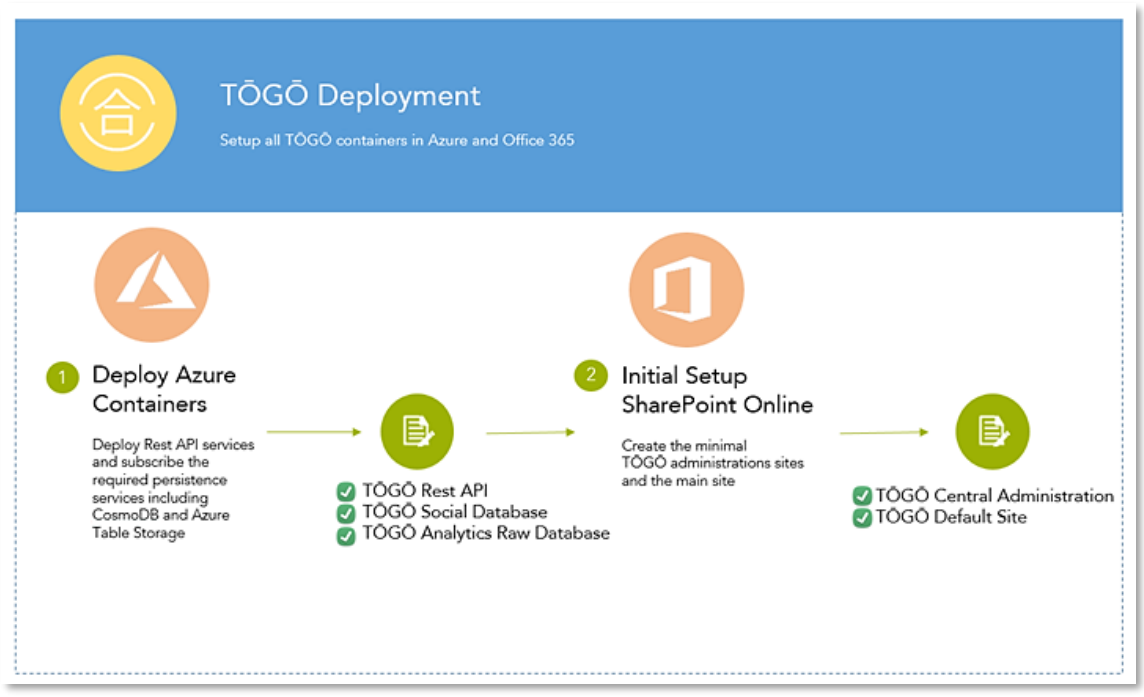
Enables operations team members to set up a TŌGŌ by following the necessary steps described in this document, after obtaining an overview of the mandatory deployment process.

Process overview

TŌGŌ is a system based on multiple interconnected containers. Each container has its own deployment process and necessary infrastructure. NTT DATA has created a process based on a series of activities to help our customers successfully set up a TŌGŌ Fast Track.

The following diagram outlines the activities required for a successful TŌGŌ setup.





Planning

Work with the IT department to perform the Technical Assessment and Plan the TŌGŌ deployment.

Technical Assessment

This activity allows the current as-is situation to be identified from a technical perspective prior to the deployment of TŌGŌ and will identify the preparatory actions required to set up TŌGŌ on the current as-is technical environment.

Deployment Planning

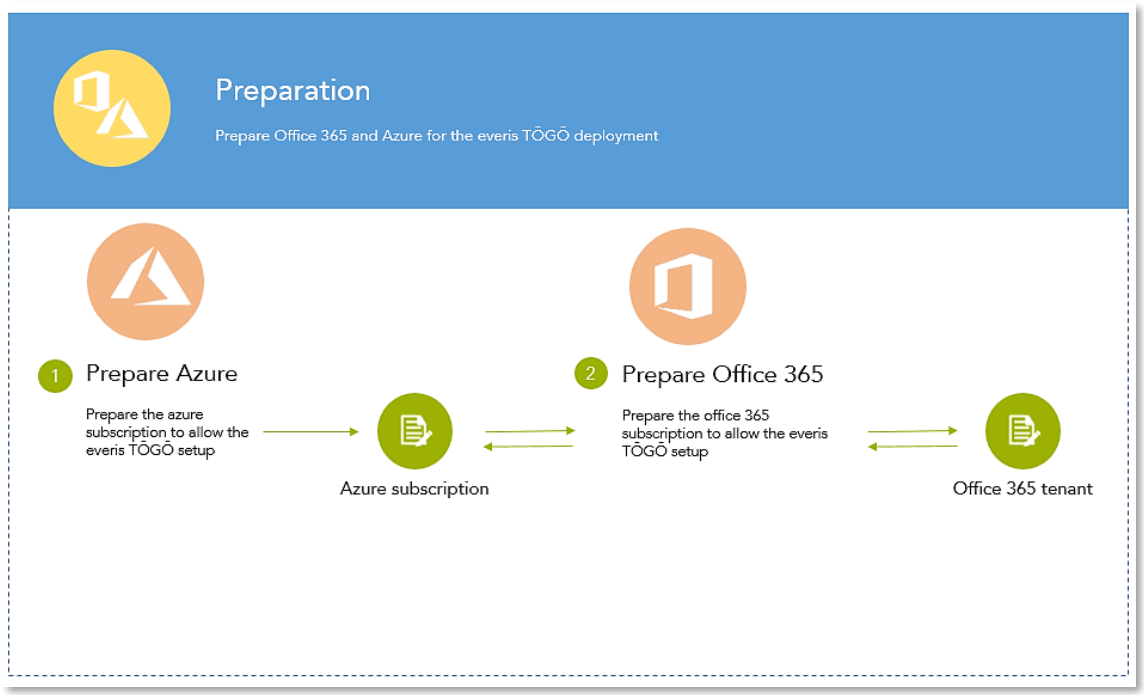
Using the technical assessment as input, you will need to start creating a deployment plan that includes the required deployment actions. Each action will include the due date and responsible.

TECHNICAL ASSESMENT	
	GOAL Identify the organizational, environmental, and technical information that will be required for a successful TŌGŌ setup process.
	INPUT **Technical Assessment Questionnaire** A questionnaire designed to help you to identify what technical infrastructure and support will be required for the TŌGŌ setup.
	TASKS **Technical Assessment** Conduct a series of meetings and workshops to gather all the necessary information for the TŌGŌ deployment and assess the current status. This information can be used as input to create a deployment plan. **Technical Assessment Report Review & Approval** Once the technical assessment report is completed, the IT department must review and iterate it as needed, and once it meets the IT department's needs, approve it.
	DELIVERABLES **Technical Assessment Report** A document that contains the Technical Assessment Report. This report contains the as-is situation before the TŌGŌ deployment and identifies the necessary preparatory actions that will be required before the TŌGŌ deployment.

DEPLOYMENT PLANNING	
	GOAL Create a deployment plan with an agreed upon timeline, stakeholders responsible for executing the TŌGŌ deployment, including preparation and deployment tasks.
	INPUT **Technical Assessment Report** A document that contains the Technical Assessment Report. This report contains the as-is situation prior to the TŌGŌ deployment and identifies the required preparatory actions that will be required prior to the TŌGŌ deployment.
	TASKS **Create the Deployment Plan** Once you have identified all the required tasks, you will need to identify task dependencies, task owners, and stakeholders. Meet with key people to get estimates and agree on a timeline based on their team assignments to start building the plan. **Review & Approve the Deployment Plan** Once the deployment plan is complete, share it with all stakeholders and owners for review and approval if no further iterations of the document are required.
	DELIVERABLES **Deployment plan** A document containing the deployment plan agreed upon by all stakeholders that will guide the next two phases.

Preparation

Perform the preparation actions as per the deployment plan schedule. These actions are specific to each environment and will be handled as per the deployment plan schedule.



Prepare Azure

Execute the necessary steps to create an Azure subscription that permits deployment of TŌGŌ Azure PaaS services.

Prepare Microsoft 365

Execute the necessary steps to configure a Microsoft 365 tenant that enables TŌGŌ SharePoint deployment.

PREPARE AZURE	
	GOAL Ensure a properly configured Azure subscription, including synchronization of Microsoft 365 Tenant Microsoft Entra IDs.
	INPUT **Deployment Plan** A document that lists the required Azure preparation tasks and the schedule for performing these tasks.
	TASKS **Execute deployment plan Azure preparation tasks** Perform the Azure preparation tasks identified in the deployment plan. **Control and monitor** Depending on the Deployment Plan, complexity control and monitoring are key tasks for a successful deployment. During this phase, we plan to identify technical and coordination issues that may need to be addressed during the deployment and adjust the Deployment Plan schedule accordingly.
	DELIVERABLES **Azure Subscription** An Azure Subscription setup as required for the TŌGŌ Fast Track deployment.

PREPARE MICROSOFT 365	
	GOAL Create the minimal infrastructure in Microsoft 365 SharePoint Online to enable TŌGŌ deployment.
	INPUT **Deployment Plan** A document that outlines the required Microsoft 365 preparation tasks and the timeline for completing those tasks.
	TASKS **Execute the Microsoft 365 preparation tasks in the deployment plan** Perform the Microsoft 365 preparation tasks identified in the deployment plan. **Control and monitor** Depending on the deployment plan, complexity control and monitoring is a key task for a successful deployment. During this phase, we plan to identify technical and coordination issues that can be adjusted as we go along and adjust the Deployment Plan schedule accordingly.
	DELIVERABLES **Microsoft 365 Tenant** A Microsoft 365 tenant with the required service subscriptions and environment ready to set up TŌGŌ.

Deployment

The following section describes the steps required to complete a TŌGŌ Fast Track installation.

Preparation

Before starting

Before you begin, you will need:

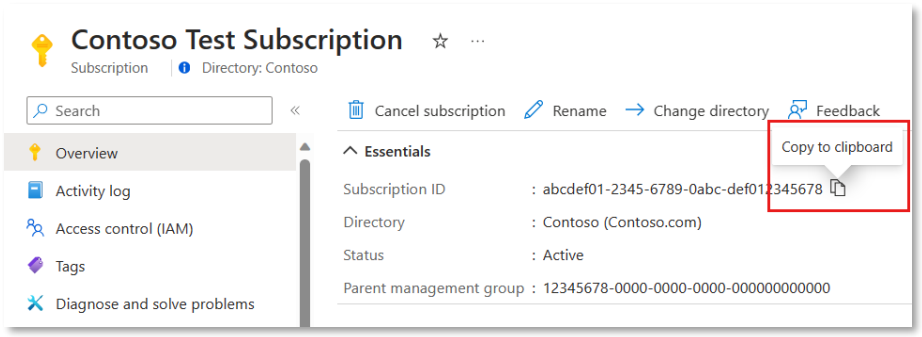
1. The TŌGŌ Deployment Package.
2. A computer with:
    a. Internet access to the Azure subscription and Microsoft 365 tenant subscription.
    b. PowerShell installed.
    c. A Microsoft 365 tenant with an app catalog site created to allow you to deploy the TŌGŌ SPFx solutions.

Note: the recommended name for the application catalog site is **AppCatalog**.

Get the Azure subscription ID

Follow these steps to retrieve the ID for a subscription in the Azure portal.

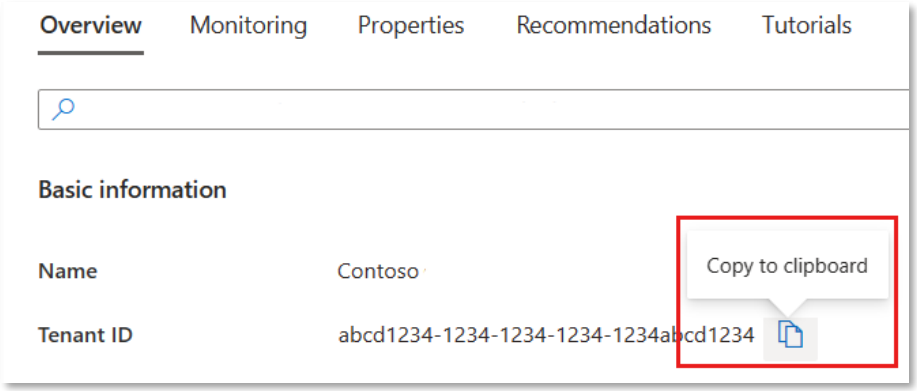
1. Login to Azure portal with an Azure subscription administration user.
2. Click on the **Subscriptions** menu item.
3. Find the **Subscription ID** in the **Overview** section.



Get the Azure tenant ID

Follow these steps to retrieve the ID for a Microsoft Entra tenant in the Azure portal.

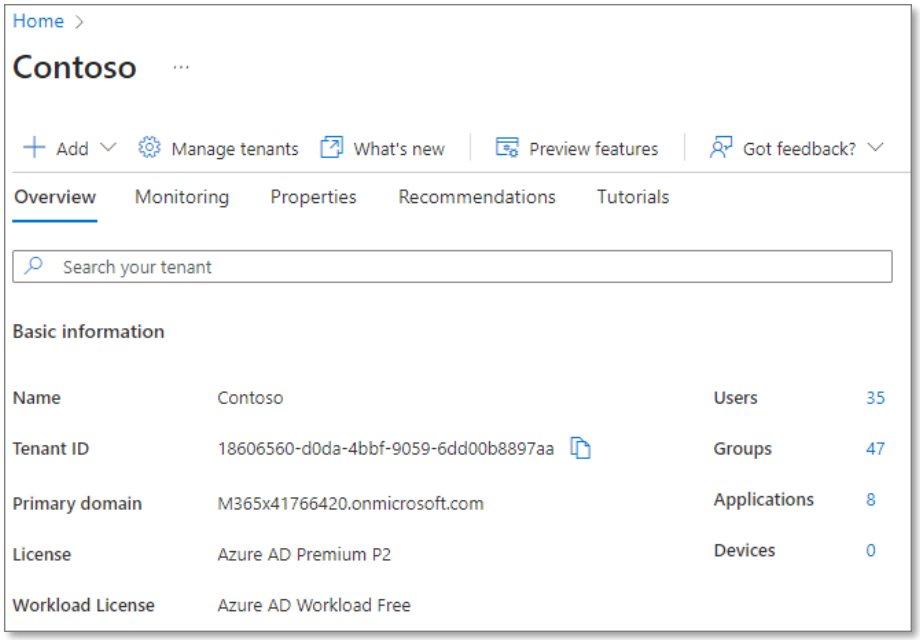
1. Login to Azure portal with a subscription administration user.
2. Click on the **Microsoft Entra ID** menu item.
3. Find the **Tenant ID** in the **Overview** section.



Get the Microsoft 365 tenant ID

Following are the steps required to obtain the Azure subscription id:

1. Login to Microsoft Entra admin center with a Microsoft 365 subscription administration user.
2. Find the **Tenant ID** in the **Overview** section.



Install additional PowerShell modules

All scripts must be run with PowerShell 5.0 version on the local machine.

The following are the PowerShell modules required during the installation process:

- Node.js
- Bicep
- PnP PowerShell
- Azure Az
- Azure CLI
- Azure AD
- Azure Bot Service
- M365 CLI
- Moline
- Open SSL

Here is how to check that the modules are present, and how to install them on the local machine that will be used during the installation. Although the installation will check that everything is correct, it is recommended that you carry out these steps first.

Node.js

To check if it exists, we will use

```
node -v
```

If it does not exist, we need to download and install it from here Node.js. We recommend that you download the LTS version

Bicep

To check if it exists, we will use

```
az bicep version
```

> Note: the minimum required version is 0.14.46

If it does not exist, we need to download and install it from here Bicep. If you are not using the Windows x64 version, you can find all versions here.

PnP PowerShell

To check if it exists, we will use

```
Get-Module -ListAvailable -Name PnP.PowerShell
```

> Note: the required version is 1.12.0

If it does not exist, we will use to install it

```
Install-Module PnP.PowerShell -RequiredVersion 1.12.0 -Force
```

Azure Az

To check if it exists, we will use

```
Get-InstalledModule -Name Az
```

> Note: the required version is 9.3.0

If it does not exist, we will use to install it

```
Install-Module -Name Az -RequiredVersion 9.3.0 -Force
```

Azure CLI

To check if it exists, we will use

```
az version
```

> Note: the minimum required version is 2.40.0

If it does not exist, we need to download and install it from here Azure CLI. If you are not using the Windows version or for more information on how to install it, you can access here.

Azure AD

To check if it exists, we will use

```
Get-Module -ListAvailable -Name AzureAD
```

If it does not exist, we will use to install it

```
Install-Module AzureAD -AllowClobber
```

Azure Bot Service

To check if it exists, we will use

```
Get-InstalledModule -Name Az.BotService
```

> Note: the minimum required version is 0.30.0

If it does not exist, we will use to install it

```
Install-Module -Name Az.BotService -RequiredVersion 0.3.0 -Force
```

M365 CLI

To check if it exists, we will use

```
m365 version
```

> Note: the minimum required version is 5.8.0

If it does not exist, we will use to install it

```
npm i -g @pnp/cli-microsoft365
```

MSOnline

To check if it exists, we will use

```
Get-Module -ListAvailable -Name MSOnline
```

If it does not exist, we will use to install it

```
Install-Module MSOnline
```

Open SSL

To check if it exists, we will use

```
openssl version
```

If it does not exist, we will use to install it

```
Install-Module -Name OpenSSL
```

Set SharePoint term store administrator

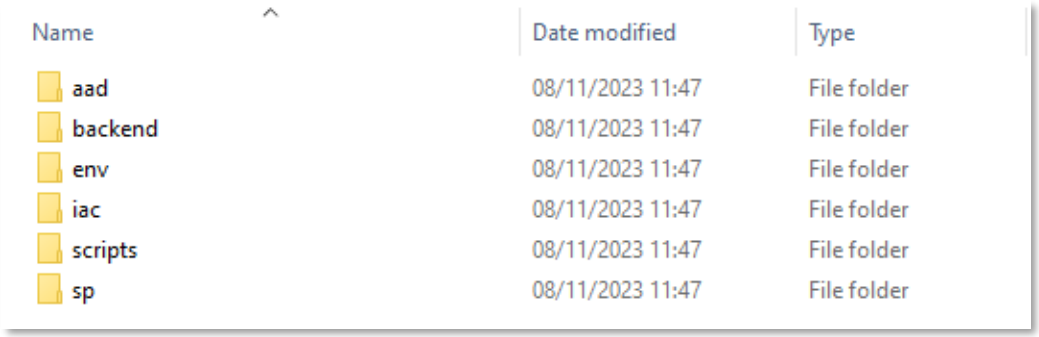
Follow these steps to add the user who will be used during the installation as the term store administrator.

1. In the SharePoint admin center, under **Content services**, select Term store.
2. In the tree view pane on the left, select the taxonomy.
3. In the **Term store** page, for **Admins**, select **Edit**. The **Edit term store admin** panel appears. Enter the names or email addresses of the people who you want to add as term store admins. Select **Save**.

Install

Deployment Package Structure

The TŌGŌ Deployment Package contains the following files and folders.

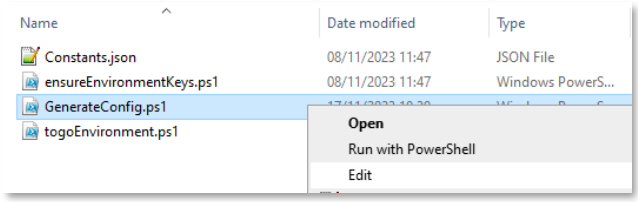


- **aad**, folder containing the support scripts and files required during the deployment process to create the app registrations in the Microsoft Entra ID.
- **backend**, folder containing the support scripts and files required during the deployment process to set the binaries for the TŌGŌ app services.
- **env**, folder containing the support scripts and files required during the deployment process to manage the environment constants.
- **iac**, folder containing the support scripts and files required during the deployment process to provision the TŌGŌ resources.
- **scripts**, folder containing the support scripts and files required to start the deployment process.
- **sp**, folder containing the support scripts and files required during the deployment process to configure the SharePoint resources and upload the TŌGŌ solution to the app catalog.

Configure environment constants

In this first step, you will configure the environment constants associated with your Microsoft 365 and Azure subscription.

We will do this with the help of the **GenerateConfig.ps1** file, located in the **env** folder, using the Windows PowerShell ISE, as shown next. Once the data is entered, it is necessary to save the file and run the script.



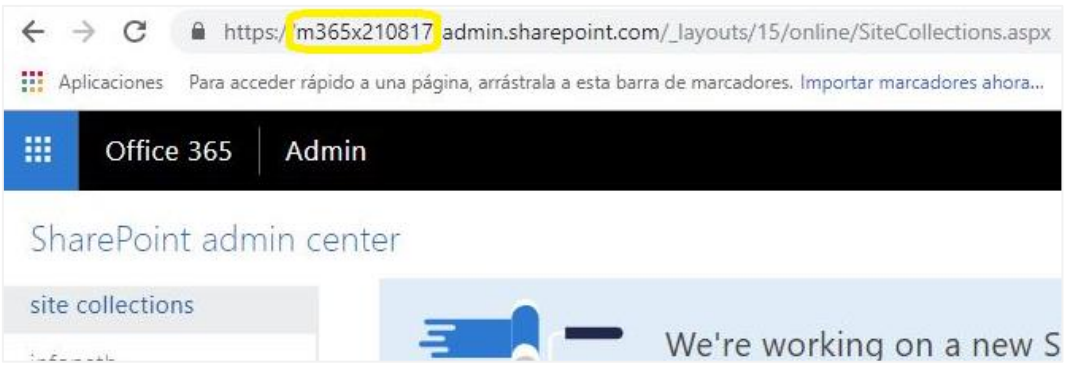
GenerateConfig.ps1

```
param (
    [Parameter(Mandatory = $true, HelpMessage = "Tenant name, which corresponds to the prefix of the SharePoint URL. Eg: For 'https://everitogopre.s
    [string]$tenantName,
    [Parameter(Mandatory = $true, HelpMessage = "Azure Subscription identifier")]
    [string]$azSubscriptionId,
    [Parameter(Mandatory = $true, HelpMessage = "O365 Tenant identifier")]
    [string]$o365TenantId,
    [Parameter(Mandatory = $true, HelpMessage = "Azure Tenant identifier")]
    [string]$azTenantId,
    [Parameter(Mandatory = $true, HelpMessage = "O365 Administrator's email")]
    [string]$o365AdminMail,
    [Parameter(Mandatory = $true, HelpMessage = "Azure Administrator's email")]
    [string]$azureAdminMail,
    [Parameter(Mandatory = $true, HelpMessage = "Managed path. Posible values: sites, teams")]
    [string]$managedPath,
    [Parameter(Mandatory = $true, HelpMessage = "Resource group name")]
    [string]$rgName,
    [Parameter(Mandatory = $true, HelpMessage = "Resource group location")]
    [string]$rgLocation,
    [Parameter(Mandatory = $true, HelpMessage = "Open Api Resource group name")]
    [string]$openApiRgName,
    [Parameter(Mandatory = $true, HelpMessage = "Open Api Resource group location")]
    [string]$openApiRgLocation,
    [Parameter(Mandatory = $true, HelpMessage = "Custom audience targetting. If true, Togo audience targetting will be applied, otherwise SharePoint
    [string]$customAudience,
    [Parameter(Mandatory = $true, HelpMessage = "Overwrite addons. If true, Togo addons will be overwritten, otherwise addons will not be modified")
    [string]$overwriteAddonDefinitions,
    [Parameter(Mandatory = $true, HelpMessage = "Togo User Profile Synchronization. If true, Togo user profile linked with external system, otherwis
    [bool]$togoUPEnabled,
    [Parameter(Mandatory = $true, HelpMessage = "Togo User Profile Synchronization. If true, Togo user profile will be sync in SharePoint user profi
    [bool]$togoSharePointUPSyncEnabled,
    [Parameter(Mandatory = $true, HelpMessage = "Generate self-signed certificate. If true, the process will generate self-signed certificate, other
    [string]$generateSelfSignedCertificate,
    [Parameter(Mandatory = $true, HelpMessage = "Togo version you want to install")]
    [string]$togoVersion
)

$environmentConfig = [ordered]@{
    tenantPropTogoVersionValue = $togoVersion;
    tenantPropTogoLocalesValue = '["en-US", "es-ES"]';
    managedPath = $managedPath;
    envM365TenantId = $o365TenantId;
    envM365TenantName = $tenantName;
```

When this script is run, you will be prompted for the following parameters:

tenantName

Specify the **name of the tenant** that matches the prefix of the SharePoint URL.



azSubscriptionId

Specify the **Subscription ID** obtained in the *Get the Azure subscription ID* section.

o365TenantId

Specify the **Tenant ID** obtained in the *Get the Microsoft 365 tenant ID* section.

azTenantId

Specify the **Tenant ID** obtained in the *Get the Azure tenant ID* section.

o365AdminMail

The email address of the Microsoft 365 tenant administrator. It is recommended that this user be the Global Admin in Microsoft 365, because this user must have administrator permissions to the Term Store, edit permissions to the SharePoint App Catalog, and permissions to the Microsoft Entra ID to create enterprise registrations.

azureAdminMail

The email address of the Azure tenant administrator. This user must have permissions to create resources and register providers within the subscription.

managedPath

This managed path is used to create the TŌGŌ site. The possible values are:

sites, https://<tenantname>.sharepoint.com/sites/togo

teams, https://<tenantname>.sharepoint.com/teams/togo

which are the managed paths that SharePoint Online provides by default.

rgName

This is the name of the resource group where all TŌGŌ resources will be installed. For example, *RG_TOGO_PRO*.

rgLocation

This is the name of the location where all TŌGŌ resources will be installed. For example, *West Europe*.

openApiRgName

This is the name of the resource group where TŌGŌ Open API resources will be installed. For example, *RG_TOGO_PRO*.

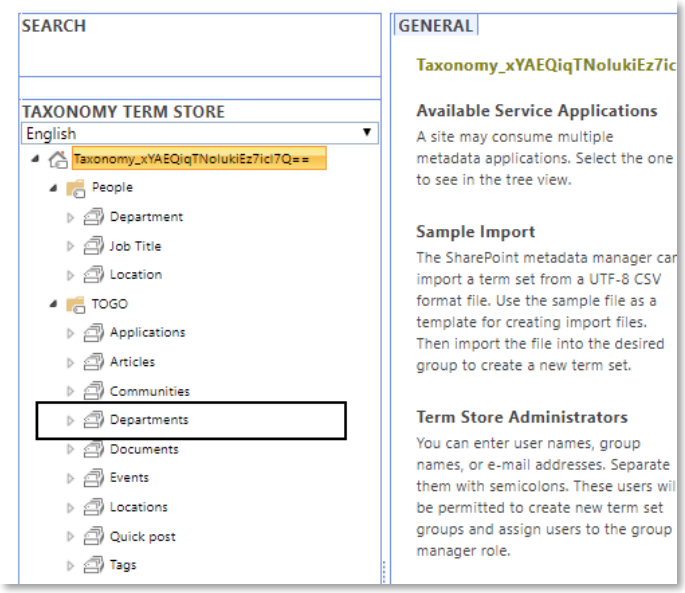
openApiRgLocation

This is the name of the location where TŌGŌ Open API resources will be installed. For example, *West Europe*.

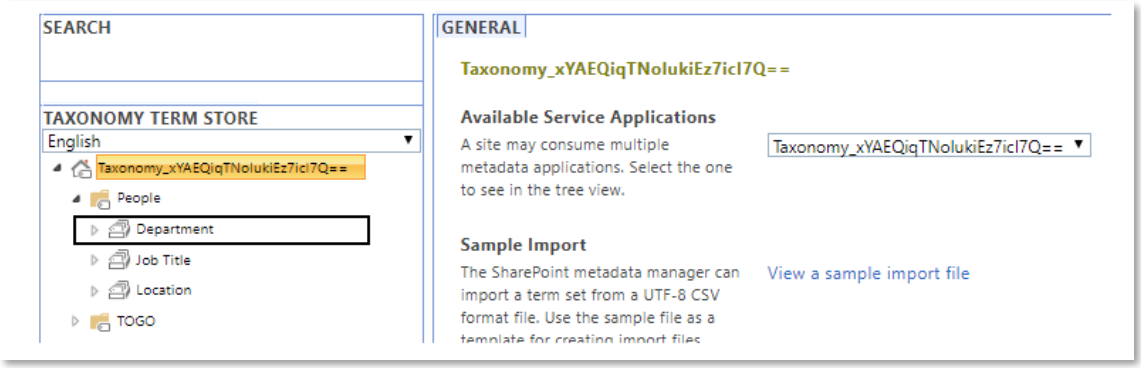
customAudience

This parameter indicates whether you want to use custom audience targeting. Possible values are true or false.

If **$true**, TŌGŌ audience targeting is used. The installation process creates a custom term set to manage the departments and locations used in the Audience Targeting and Departments and Locations filter values.



If **$false**, it uses SharePoint OOTB.



overwriteAddonDefinitions

This parameter indicates whether the addon configuration should be overwritten. Possible values are true or false. If **$true**, TŌGŌ addons are overwritten. If **$false**, TŌGŌ addons are not changed.

togoUPEnabled

As part of TŌGŌ Audience, we allow you to decide whether to synchronize the user's profile with Microsoft Entra ID or with an external system. Possible values are true or false. If **$true**, TŌGŌ user profile will be linked with external system. If **$false**, we will use synchronization with Microsoft Entra ID.

togoSharePointUPSyncEnabled

As part of TŌGŌ Audiences, we allow you to decide whether you want to synchronize the user's profile with the SharePoint Online user profile. Possible values are true or false. If **$true**, TŌGŌ user profile will be sync in SharePoint User Profile. If **$false**, it will not.

generateSelfSignedCertificate

During the installation process, we need a certificate for some Azure resources. Possible values are true or false. If **$true**, the process will generate self-signed certificate. If **$false**, it will be required to upload one to Key Vault.

togoVersion

Specify the version of TŌGŌ you want to install. For example, *1.9.0*

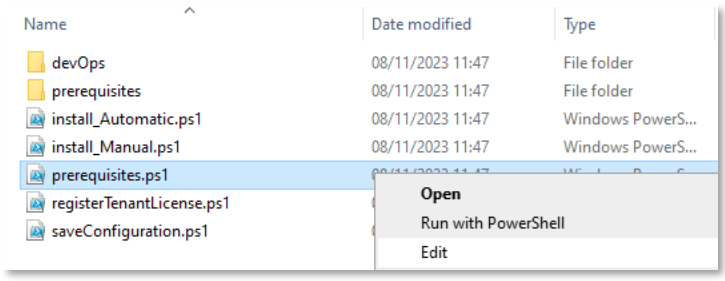
Prerequisites

In the next step, we will install all the prerequisites needed to deploy TŌGŌ. This step requires the presence of the Azure and Microsoft 365 administrators that were specified in the generation of the constant file during the previous step, as they will be asked for their credentials during the process.

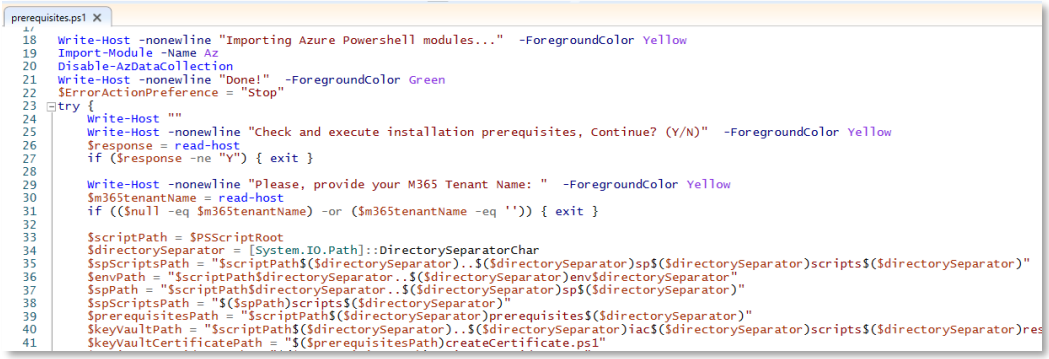
> Note: it is recommended that this user be a Global Admin in Microsoft365. Review section **o365AdminMail** of the previous step.

> Note: it is recommended that this user be an Admin in the Azure subscription. Review section **azureAdminMail** of the previous step.

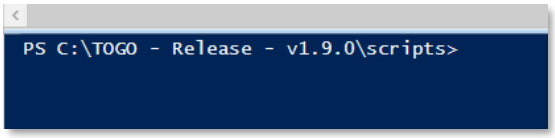
We will do this by using the **prerequisites.ps1** file, which is in the **scripts** folder, and by using the Windows PowerShell ISE, as shown next.



prerequisites.ps1



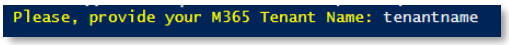
This is necessary to verify that the execution folder is the same as the folder that contains the prerequisites.ps1 script.



When everything is ready and checked, you **can run the script**. You need to follow the steps that appear in the console. There are several cases, such as entering user credentials, accessing a SharePoint configuration page, or simply waiting for the step to complete. The steps are detailed below.

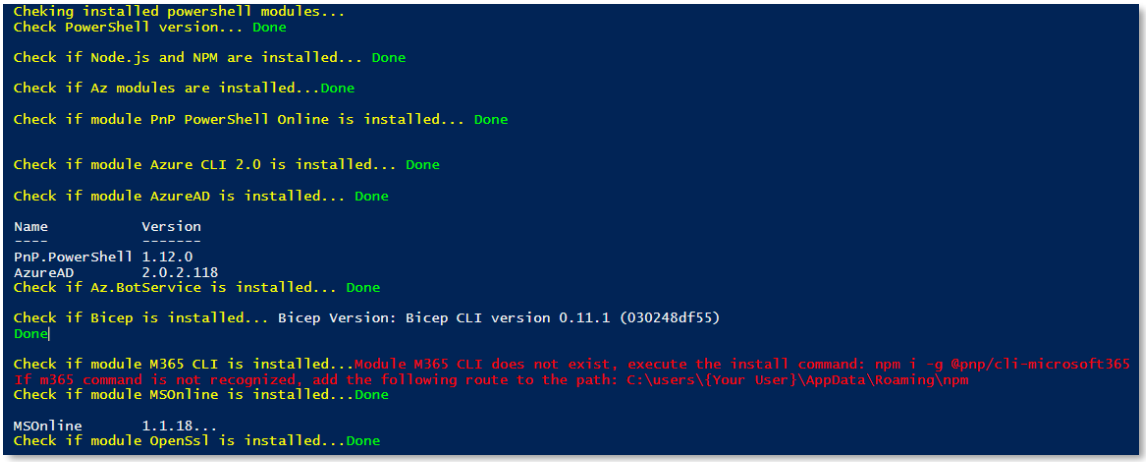
Tenant name

The first parameter the script asks for is the tenant's name, that matches the prefix of the SharePoint URL.



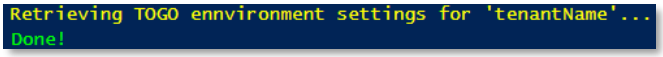
Check PowerShell modules

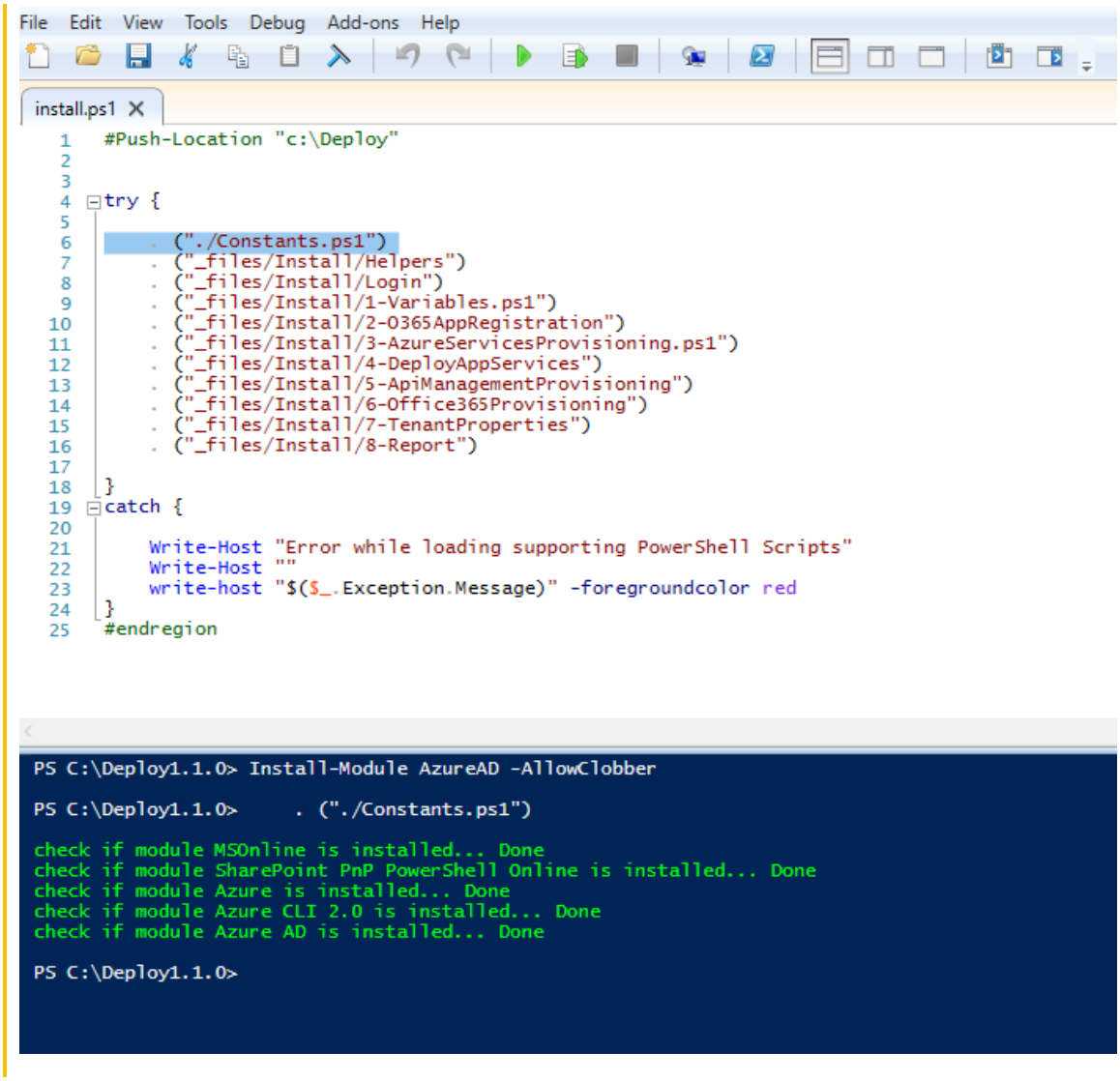
The first step is to verify that all the PowerShell modules are installed correctly. If any modules are missing, the script tells you what to do to install them. You can also refer to section Install additional PowerShell modules.



Load environment constants

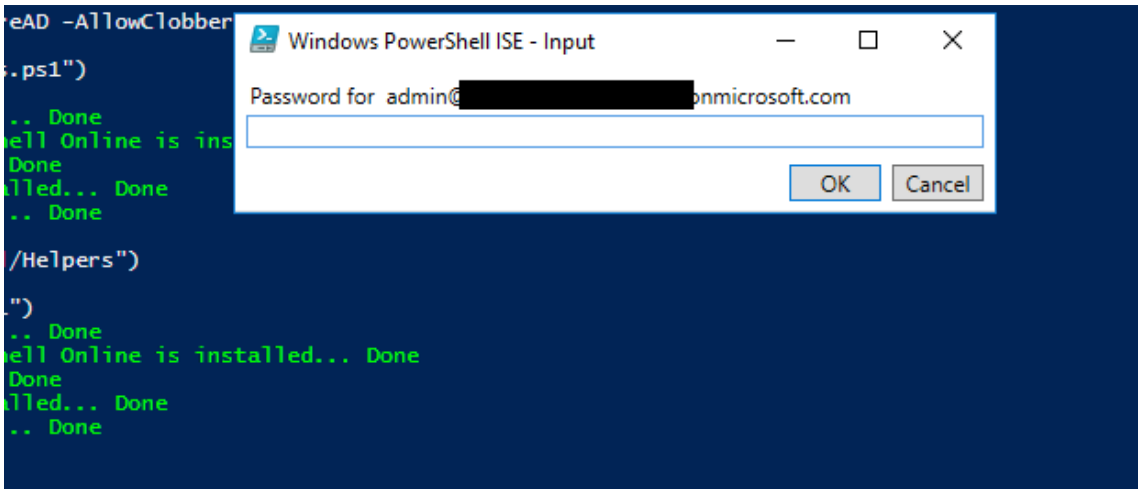
Next, the script initializes the environment constants with the tenant name. It will first try to retrieve the configuration stored in Azure Storage. If it does not exist, it will retrieve the constant file that we created in the previous steps.



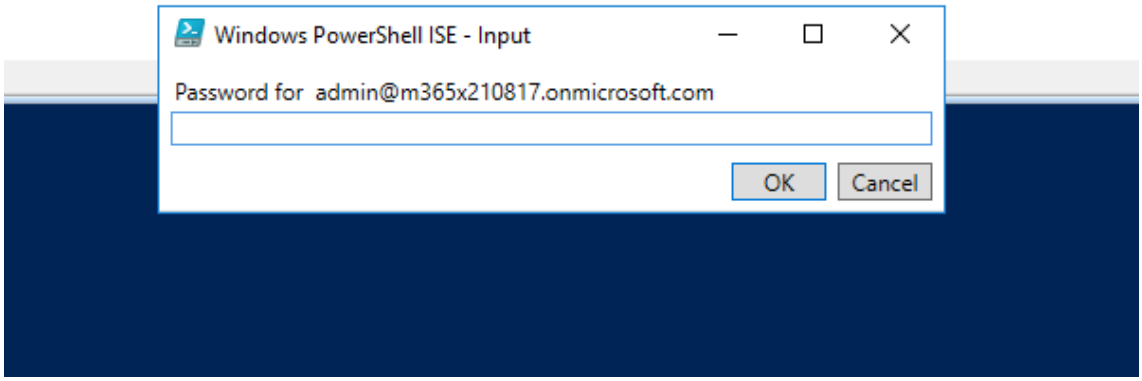


- Execute command line . ("_files/Install/Constants.ps1"):
- Execute command line . ("_files/Install/Helpers.ps1"):
- Execute command line . ("_files/Install/Login.ps1")

In this step the script will prompt for the password of the Azure admin Account

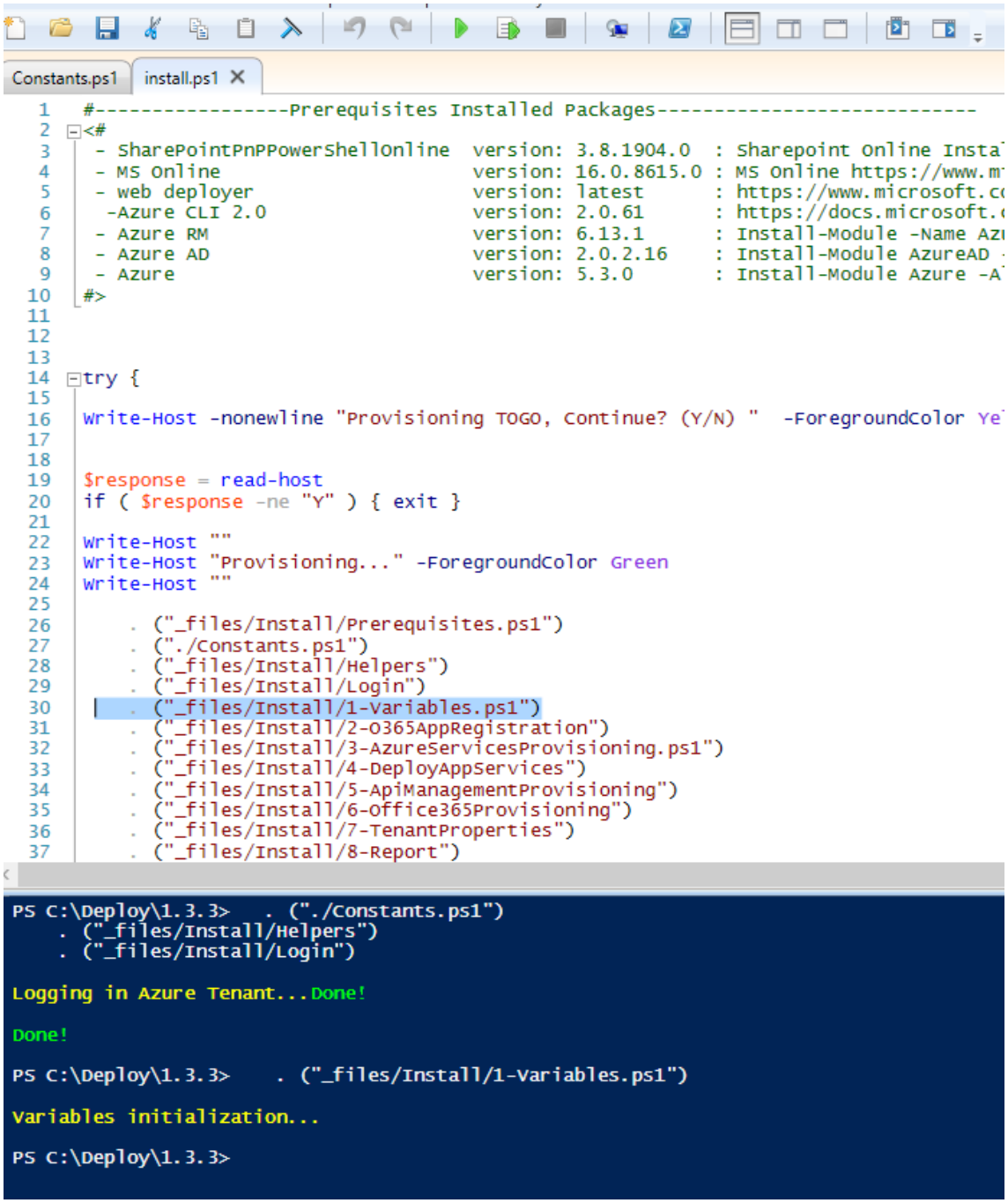


And the password of the O365 Tenant Admin

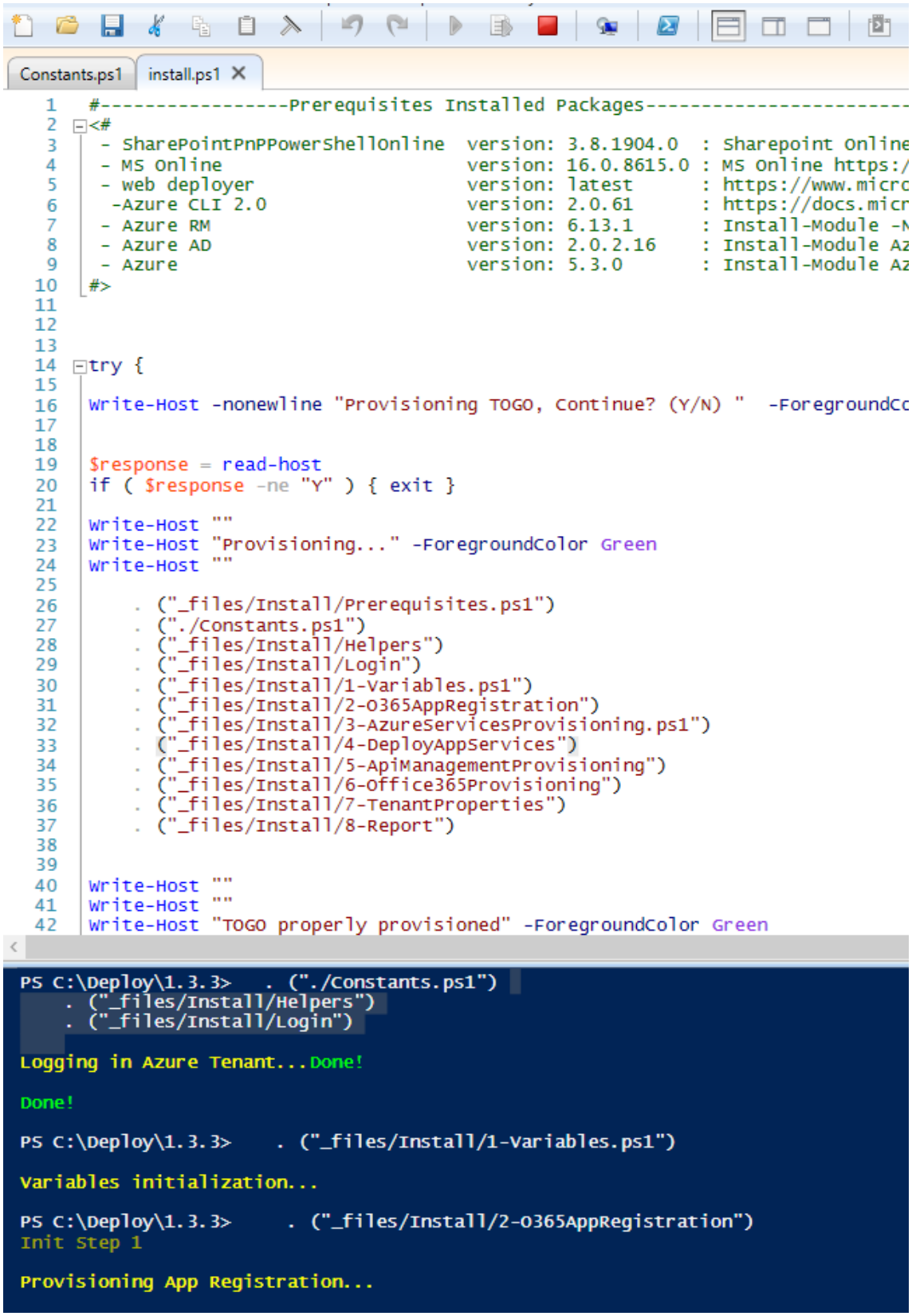


If the credentials are correct the script output the message " Done!"

- Execute command line `. ("_files/Install/1-Variables.ps1")`

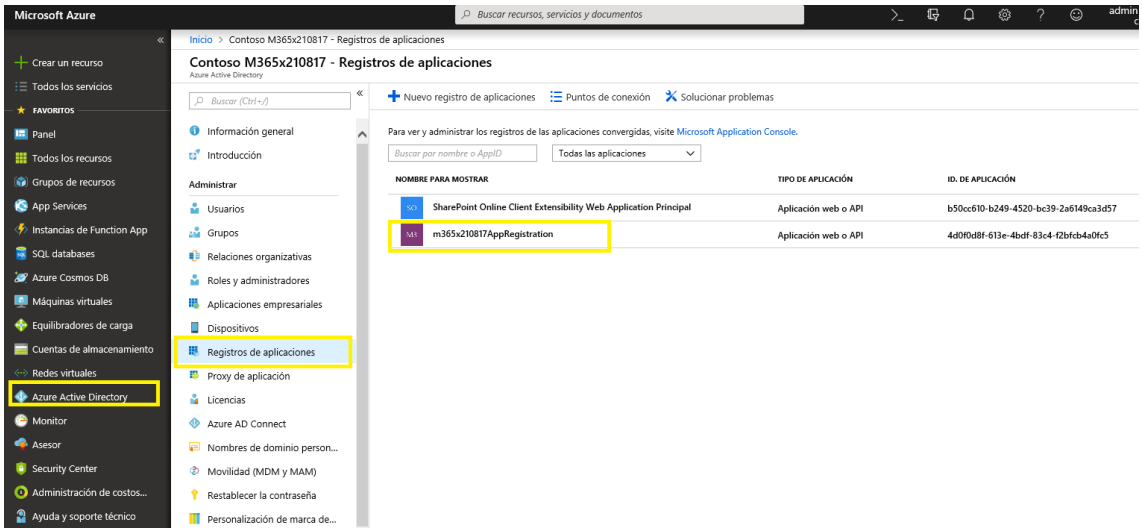


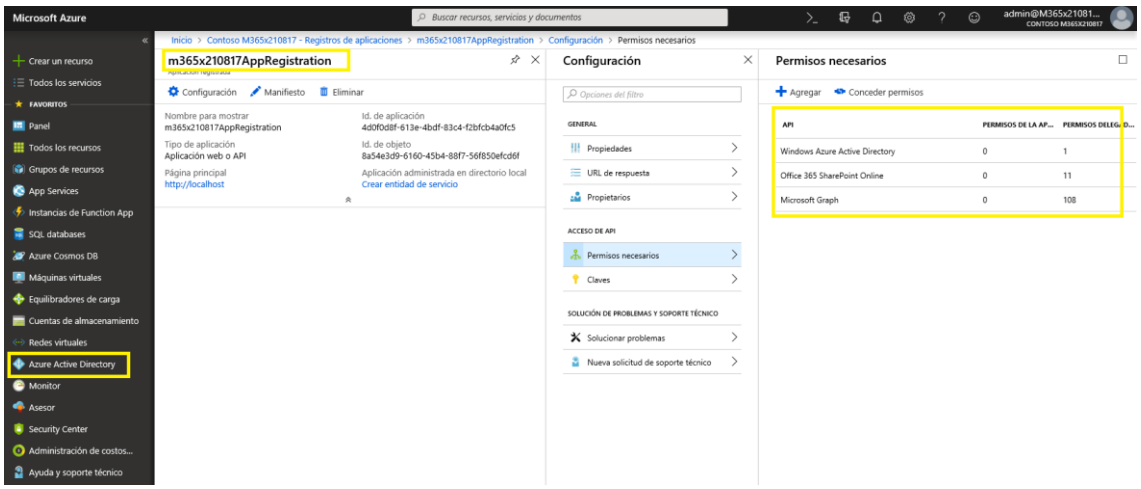
- Execute command line `. ("_files/Install/2-O365AppRegistration")`



To check if the registration process done you can navigate to https://azure.portal.com

Login with the O365 Tenant Admin credentials and go to the Azure Active directory > Application Registration ><TenantName>AppRegistration





> If the app have the same permissions showing in the above image, the process has done

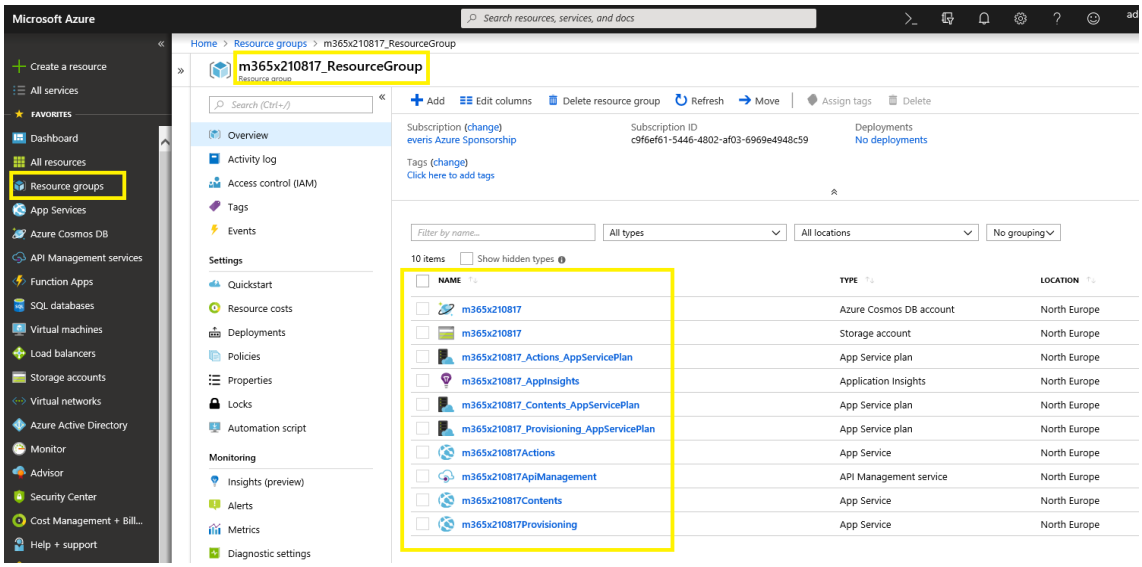
- Execute command line `. ("_files/Install/3-AzureServicesProvisioning.ps1")`

Constants.ps1 install.ps1

```
#-----------------Prerequisites Installed Packages---------------------------
<#
 - SharePointPnPPowerShellOnline  version: 3.8.1904.0  : Sharepoint Online Insta
 - MS Online                      version: 16.0.8615.0 : MS Online https://www.m
 - web deployer                   version: latest      : https://www.microsoft.c
  -Azure CLI 2.0                  version: 2.0.61      : https://docs.microsoft.
 - Azure RM                       version: 6.13.1      : Install-Module -Name Az
 - Azure AD                       version: 2.0.2.16    : Install-Module AzureAD
 - Azure                          version: 5.3.0       : Install-Module Azure -A
#>


try {

Write-Host -nonewline "Provisioning TOGO, Continue? (Y/N) "  -ForegroundColor Ye

$response = read-host
if ( $response -ne "Y" ) { exit }

Write-Host ""
Write-Host "Provisioning..." -ForegroundColor Green
Write-Host ""

    . ("_files/Install/Prerequisites.ps1")
    . ("./Constants.ps1")
    . ("_files/Install/Helpers")
    . ("_files/Install/Login")
    . ("_files/Install/1-Variables.ps1")
    . ("_files/Install/2-O365AppRegistration")
    . ("_files/Install/3-AzureServicesProvisioning.ps1")
    . ("_files/Install/4-DeployAppServices")
    . ("_files/Install/5-ApiManagementProvisioning")
    . ("_files/Install/6-Office365Provisioning")
    . ("_files/Install/7-TenantProperties")
    . ("_files/Install/8-Report")


Write-Host ""
Write-Host ""
Write-Host "TOGO properly provisioned" -ForegroundColor Green
```

```
Provisioning App Registration...Done!


Init Step 2

Create TOGO Users AD Group ...

Done!


PS C:\Deploy\1.3.3>     . ("_files/Install/3-AzureServicesProvisioning.ps1")
Init Step 3
Provisioning Resource Group...
Done!

Init Step 4

Provisioning Storge Table...
```

> When the script execution is done, you can go to azure portal and check if all resources had been created



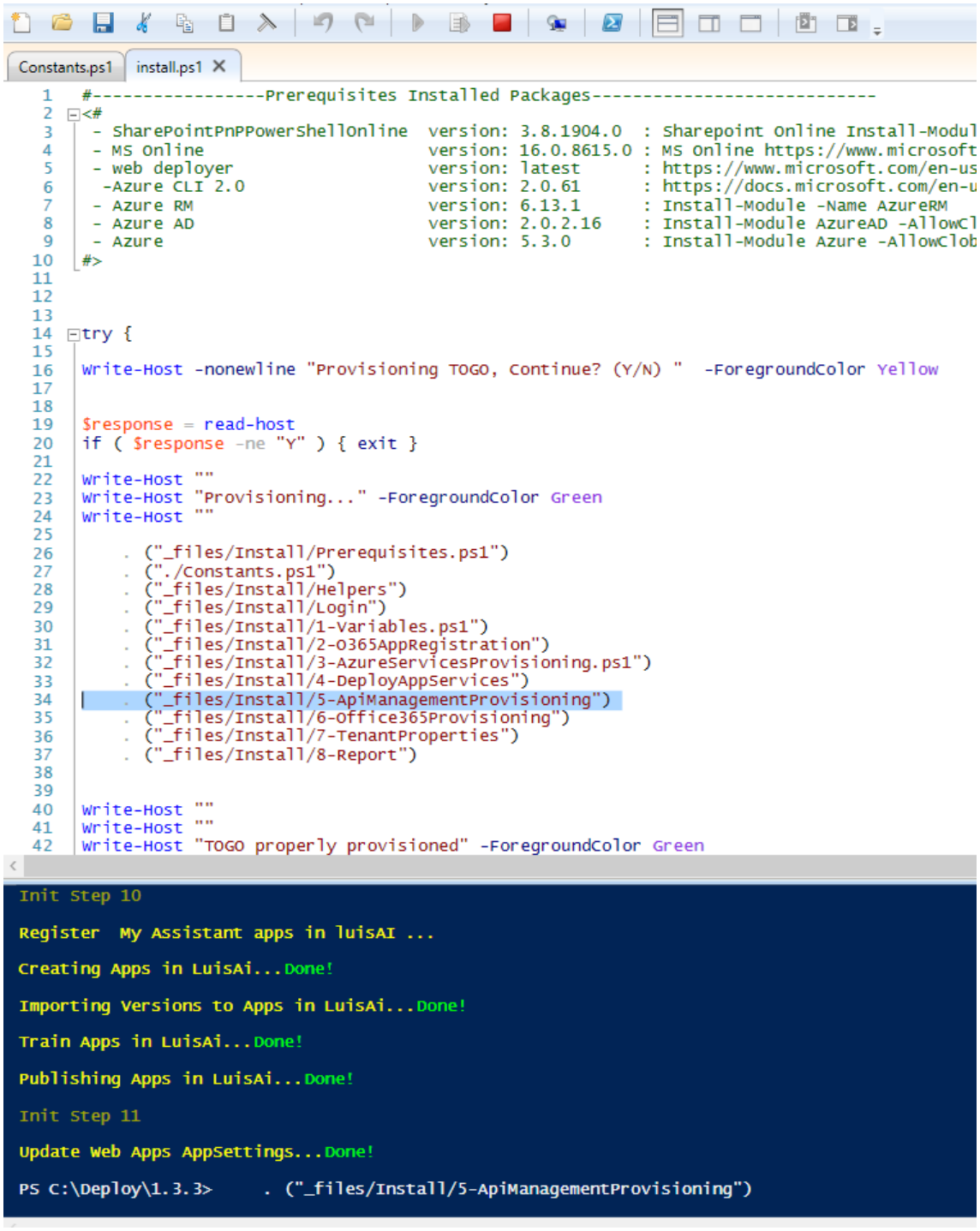
- Execute command line `. ("_files/Install/4-DeployAppServices")`

Constants.ps1 install.ps1

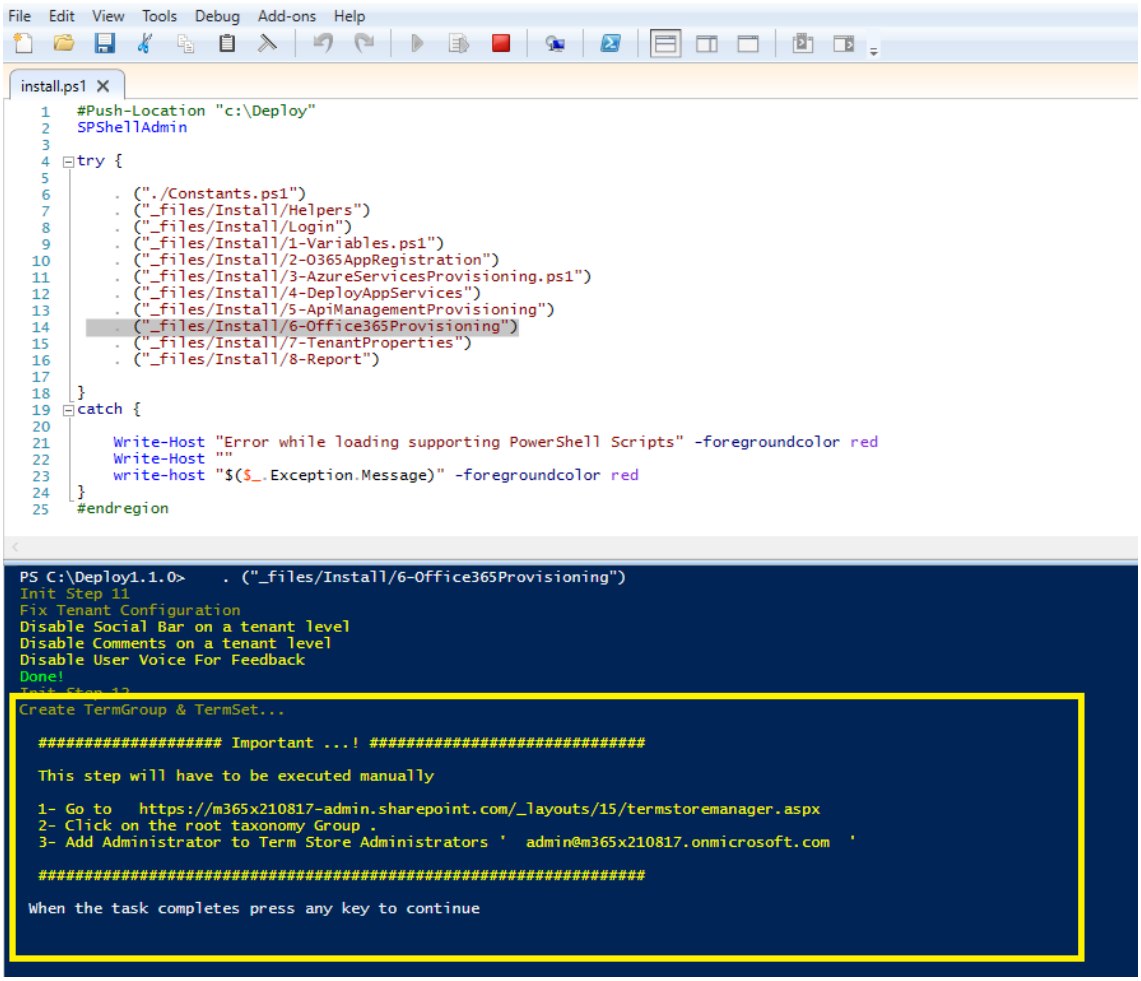
```
#-----------------Prerequisites Installed Packages----------------------------
<#
 - SharePointPnPPowerShellOnline  version: 3.8.1904.0  : Sharepoint Online Ins
 - MS Online                      version: 16.0.8615.0 : MS Online https://www
 - web deployer                   version: latest      : https://www.microsoft
  -Azure CLI 2.0                  version: 2.0.61      : https://docs.microsof
 - Azure RM                       version: 6.13.1      : Install-Module -Name
 - Azure AD                       version: 2.0.2.16    : Install-Module AzureA
 - Azure                          version: 5.3.0       : Install-Module Azure
#>


try {

Write-Host -nonewline "Provisioning TOGO, Continue? (Y/N) "  -ForegroundColor


$response = read-host
if ( $response -ne "Y" ) { exit }

Write-Host ""
Write-Host "Provisioning..." -ForegroundColor Green
Write-Host ""

    . ("_files/Install/Prerequisites.ps1")
    . ("./Constants.ps1")
    . ("_files/Install/Helpers")
    . ("_files/Install/Login")
    . ("_files/Install/1-Variables.ps1")
    . ("_files/Install/2-O365AppRegistration")
    . ("_files/Install/3-AzureServicesProvisioning.ps1")
    . ("_files/Install/4-DeployAppServices")
    . ("_files/Install/5-ApiManagementProvisioning")
    . ("_files/Install/6-Office365Provisioning")
    . ("_files/Install/7-TenantProperties")
    . ("_files/Install/8-Report")


Write-Host ""
Write-Host ""
Write-Host "TOGO properly provisioned" -ForegroundColor Green
```

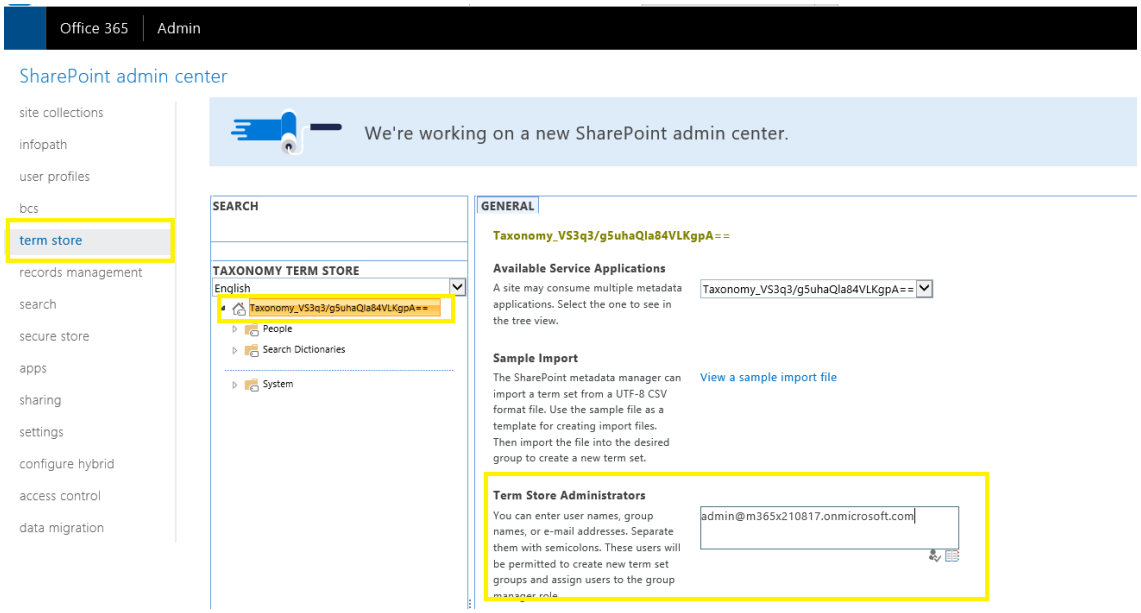
```
Init Step 10

Register  My Assistant apps in luisAI ...

Creating Apps in LuisAi...Done!

Importing Versions to Apps in LuisAi...Done!

Train Apps in LuisAi...Done!

Publishing Apps in LuisAi...Done!

Init Step 11

Update Web Apps AppSettings...Done!

PS C:\Deploy\1.3.3>  . ("_files/Install/3-AzureServicesProvisioning.ps1")
```

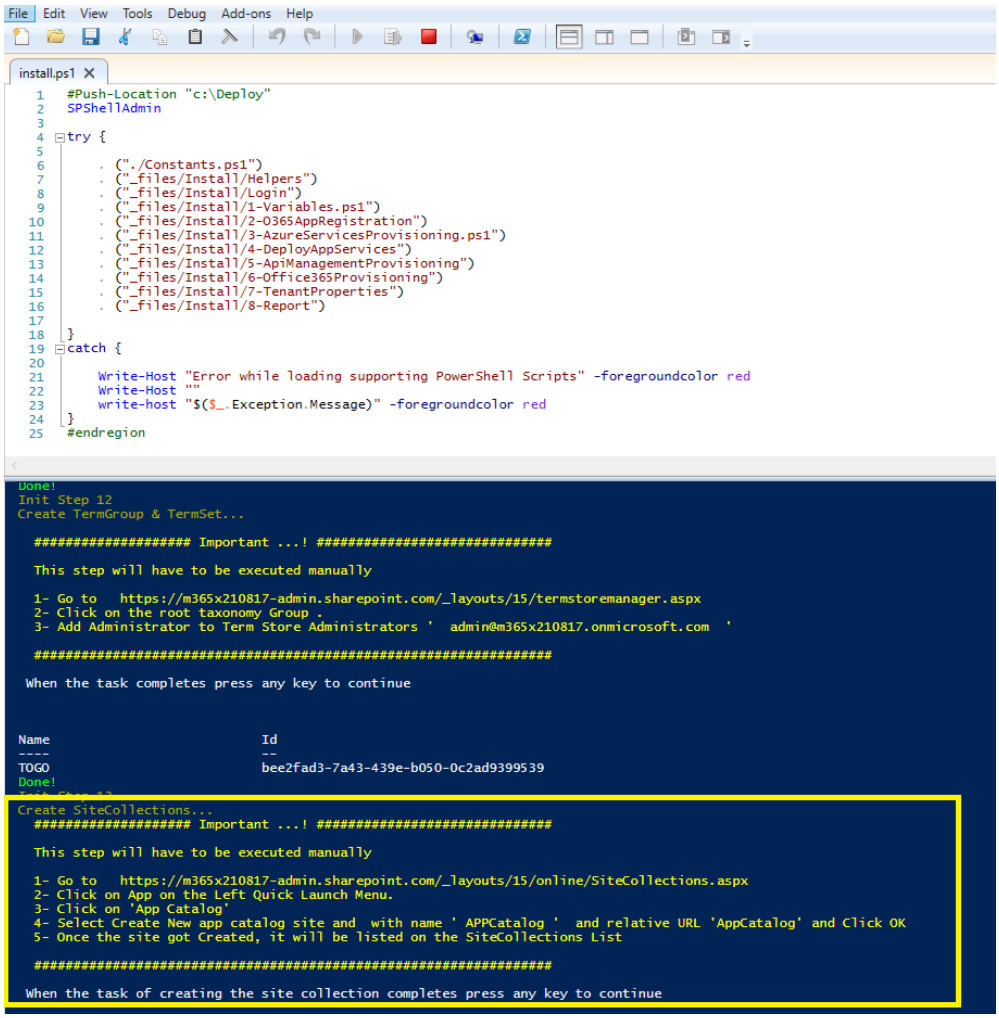
- Execute command line `.("_files/Install/5-ApiManagementProvisioning")`

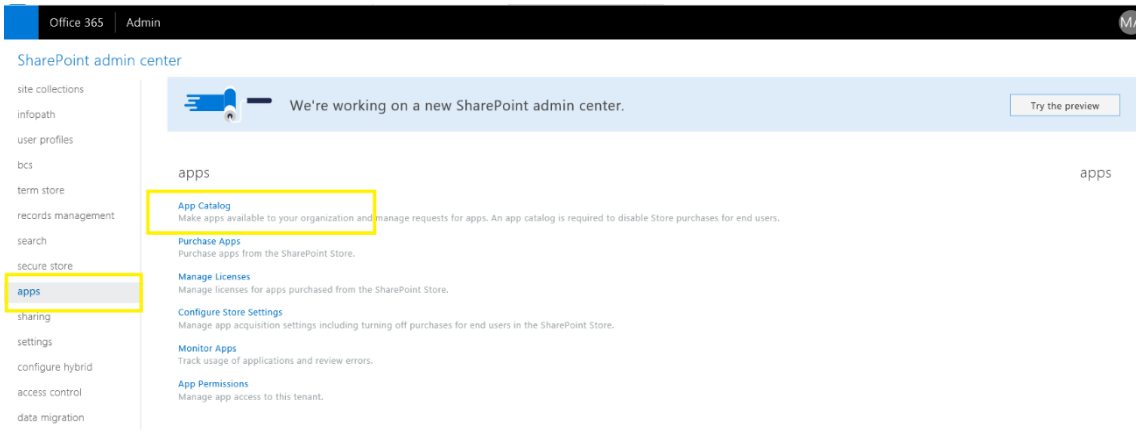


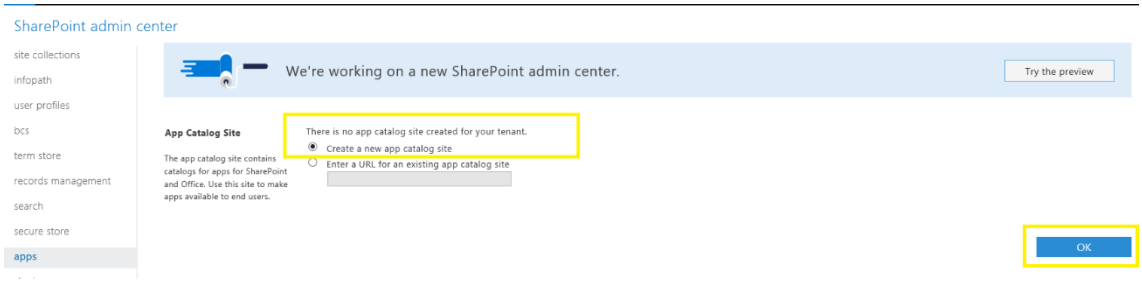
- Execute command line `. ("_files/Install/6-Office365Provisioning")`

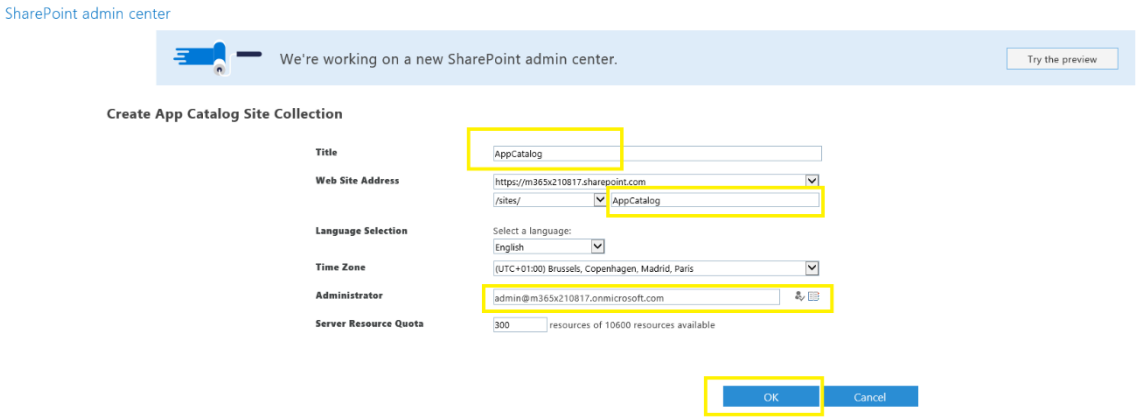


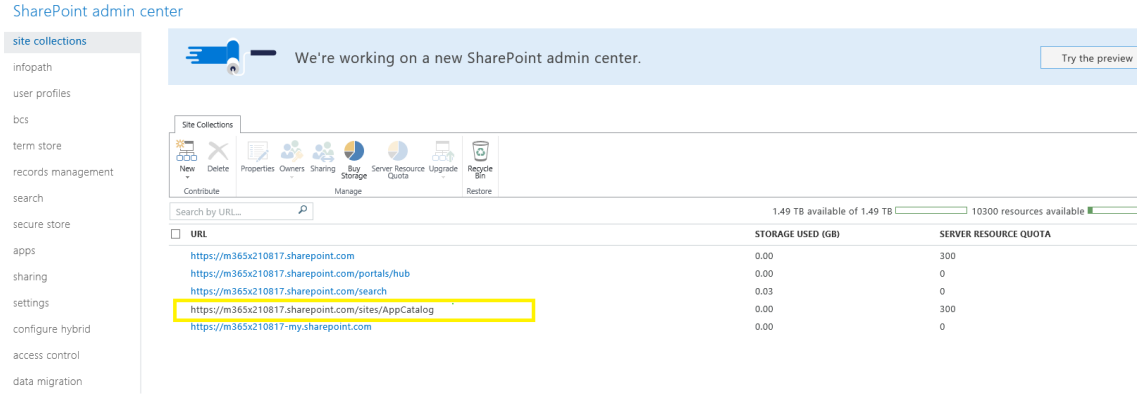


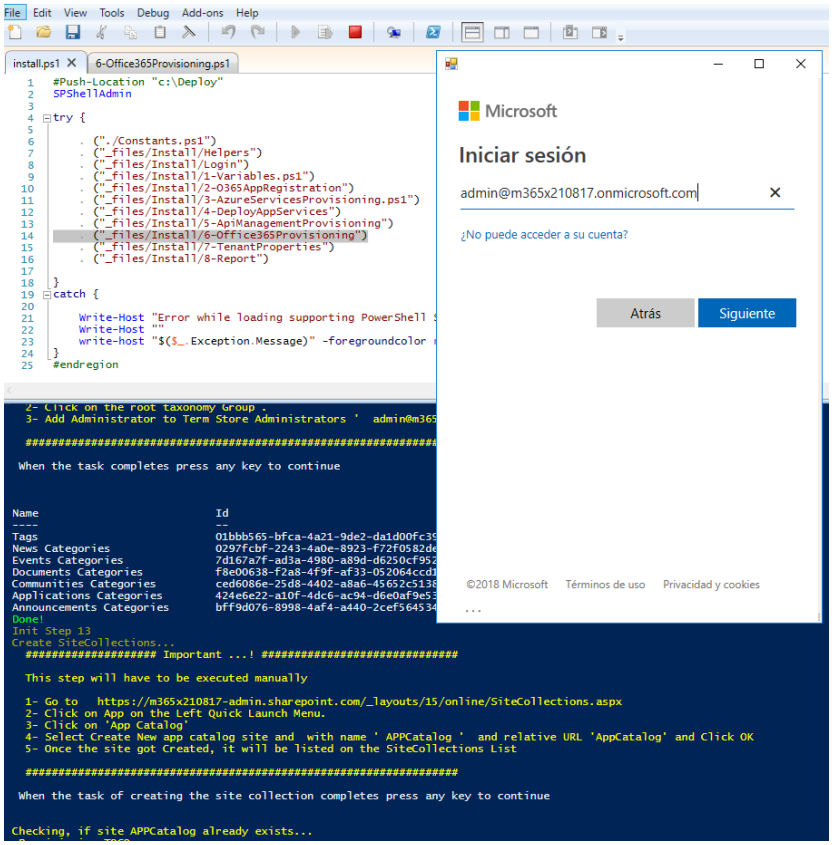


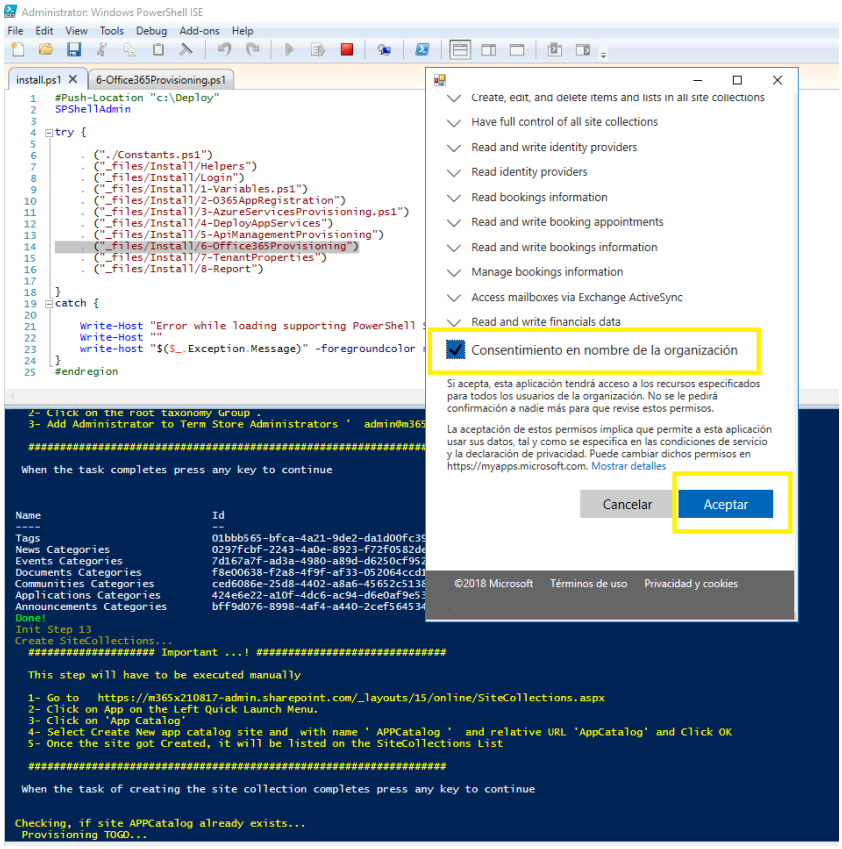


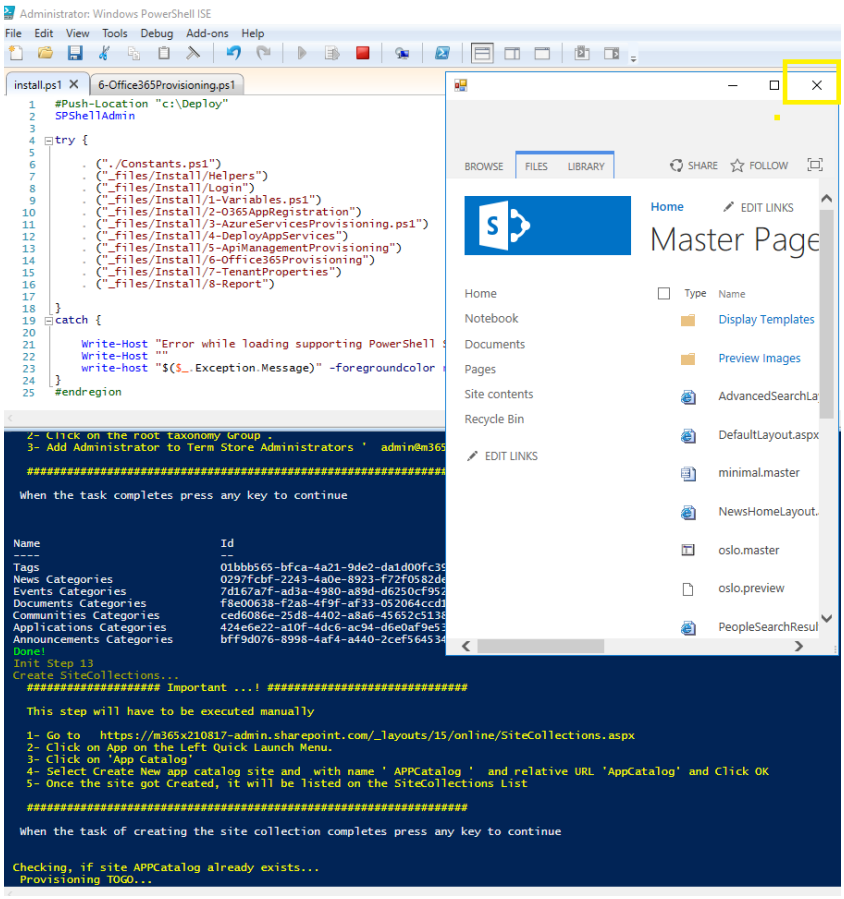


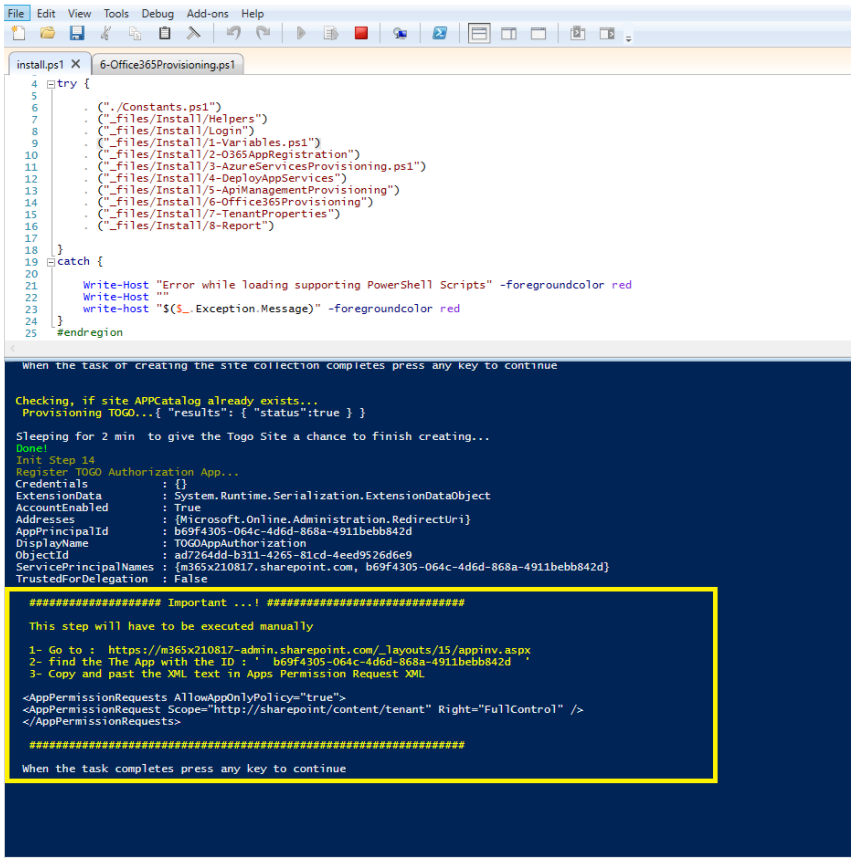


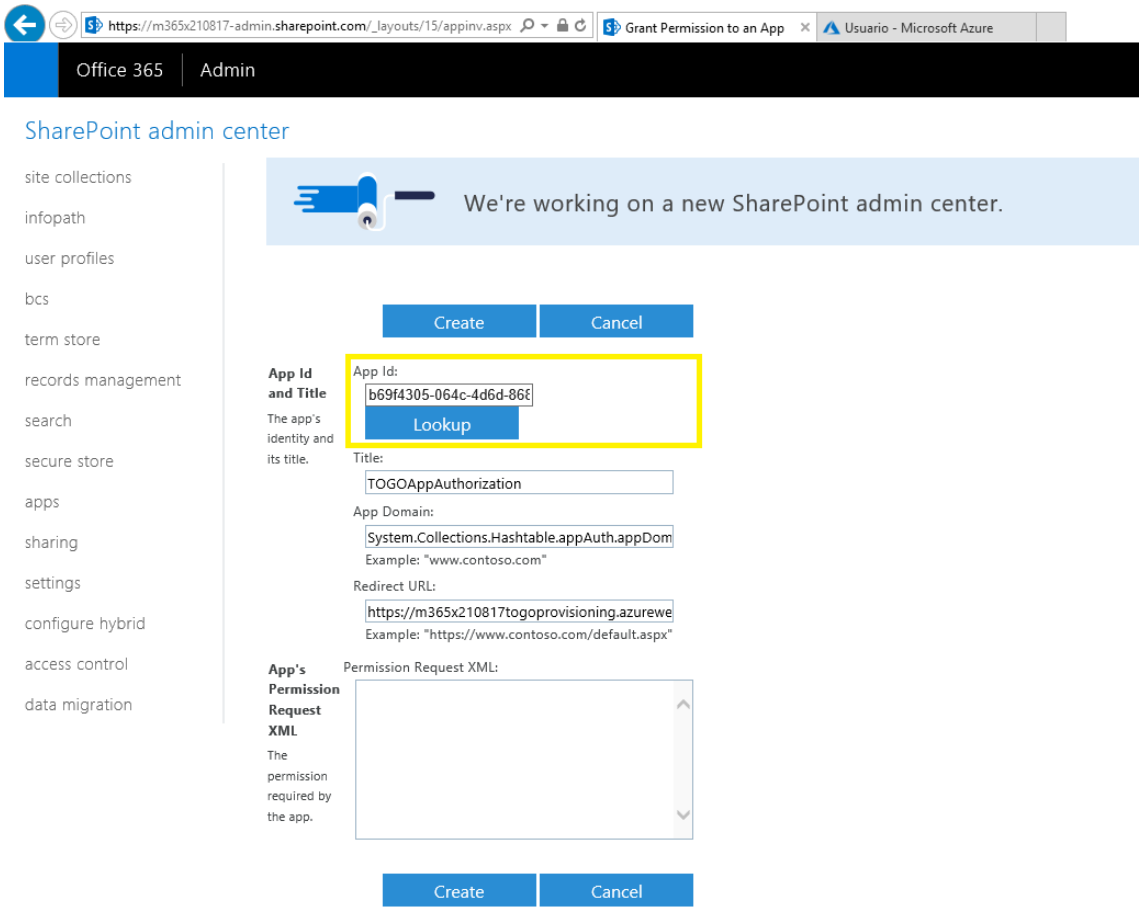


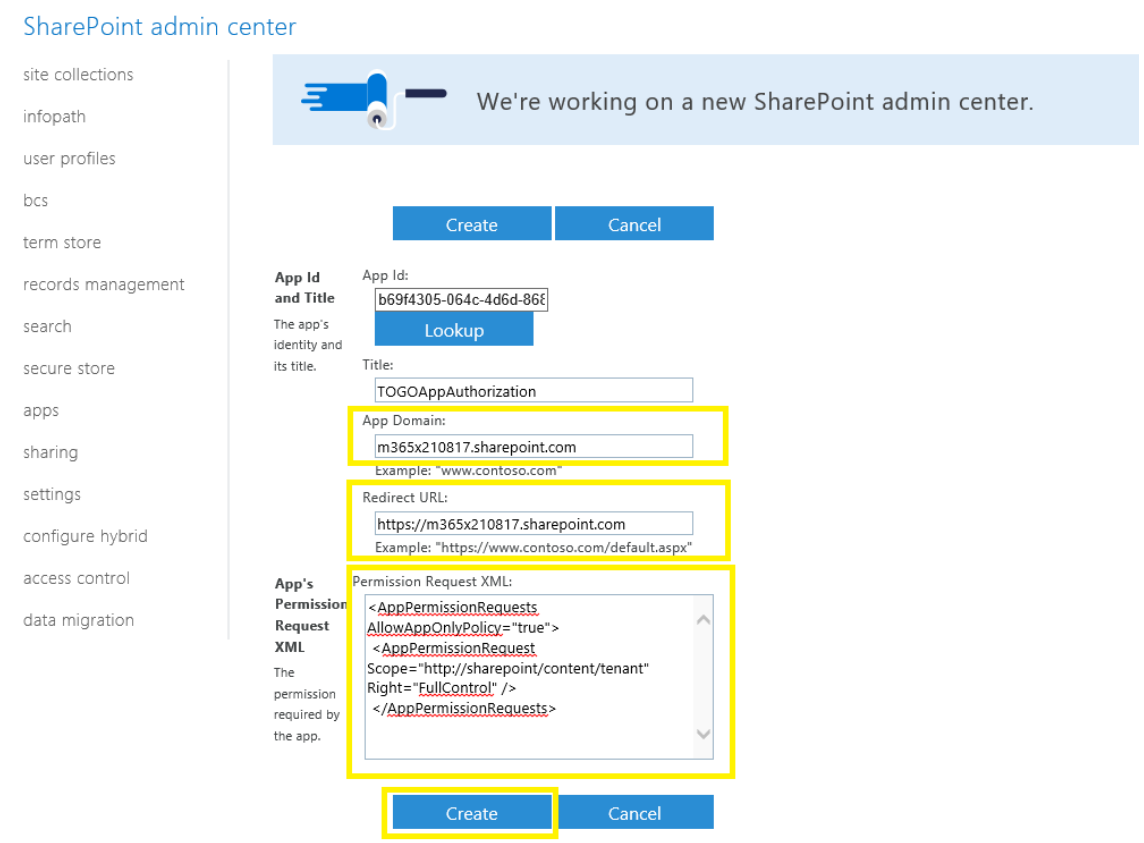


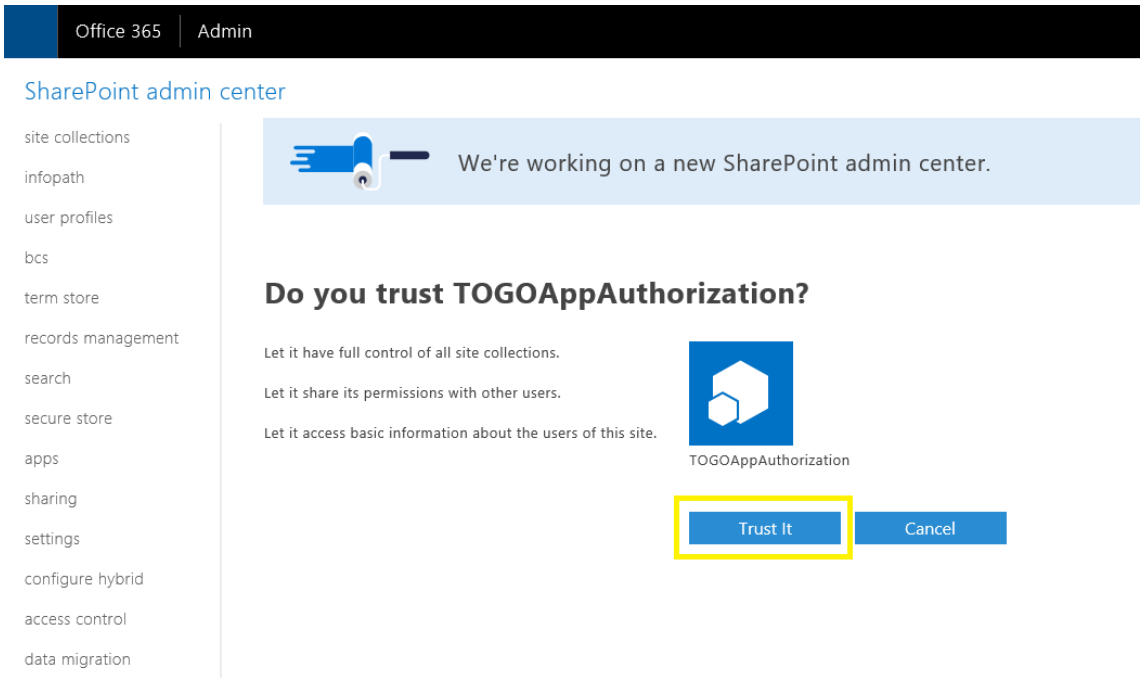


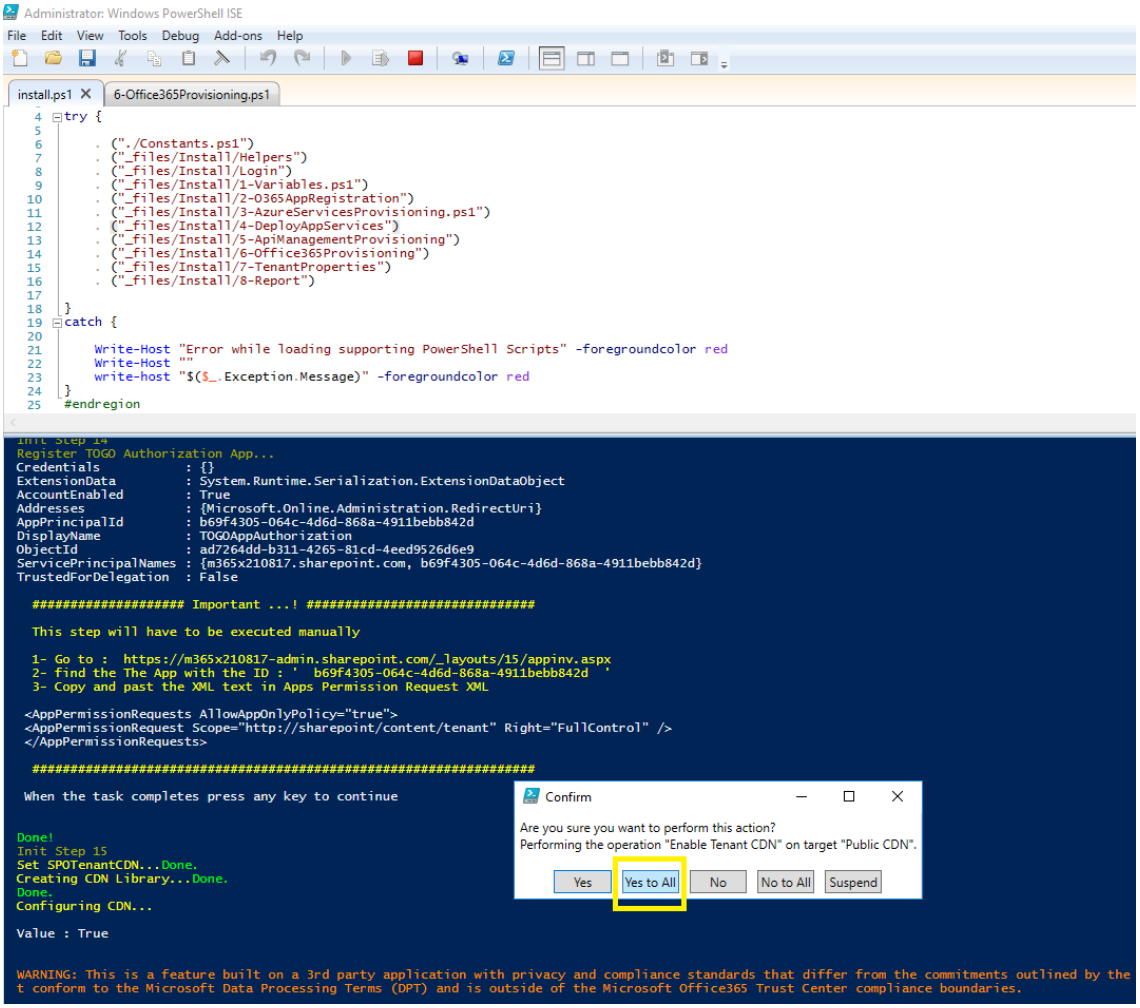


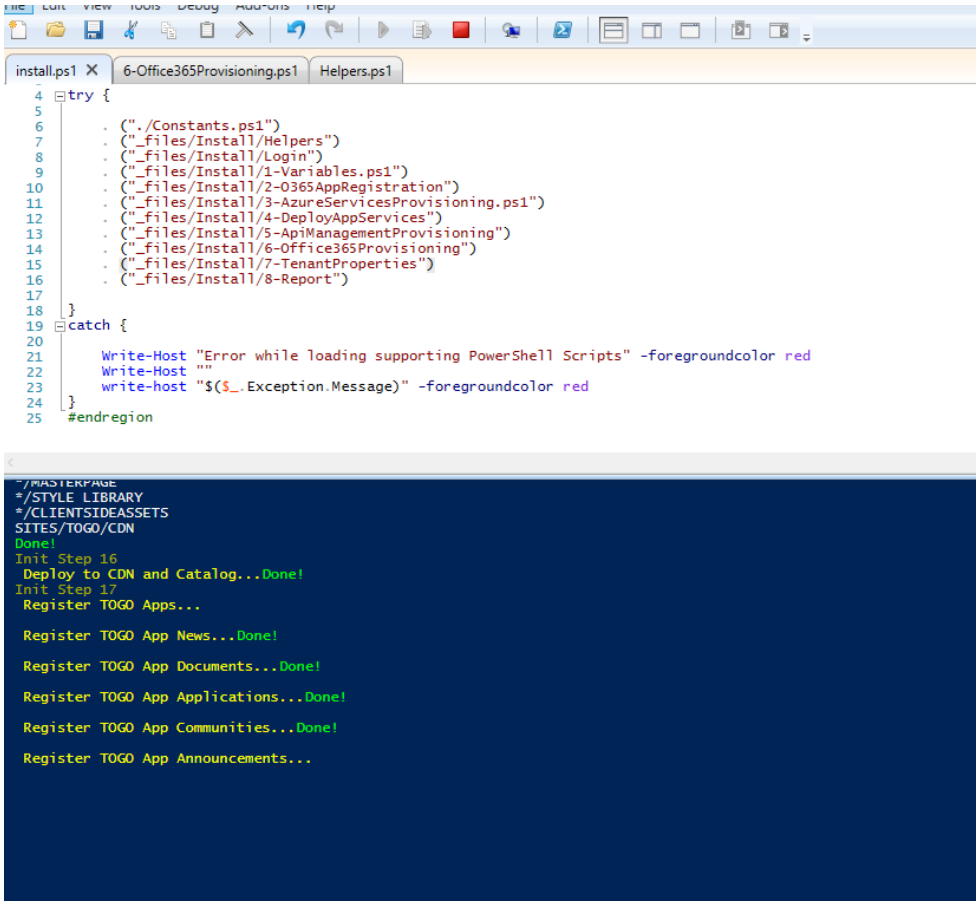




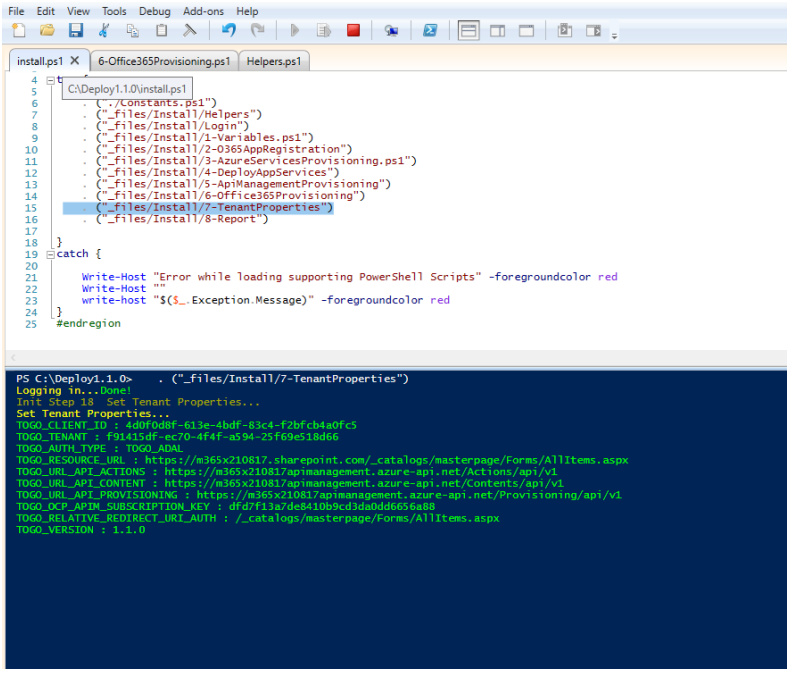




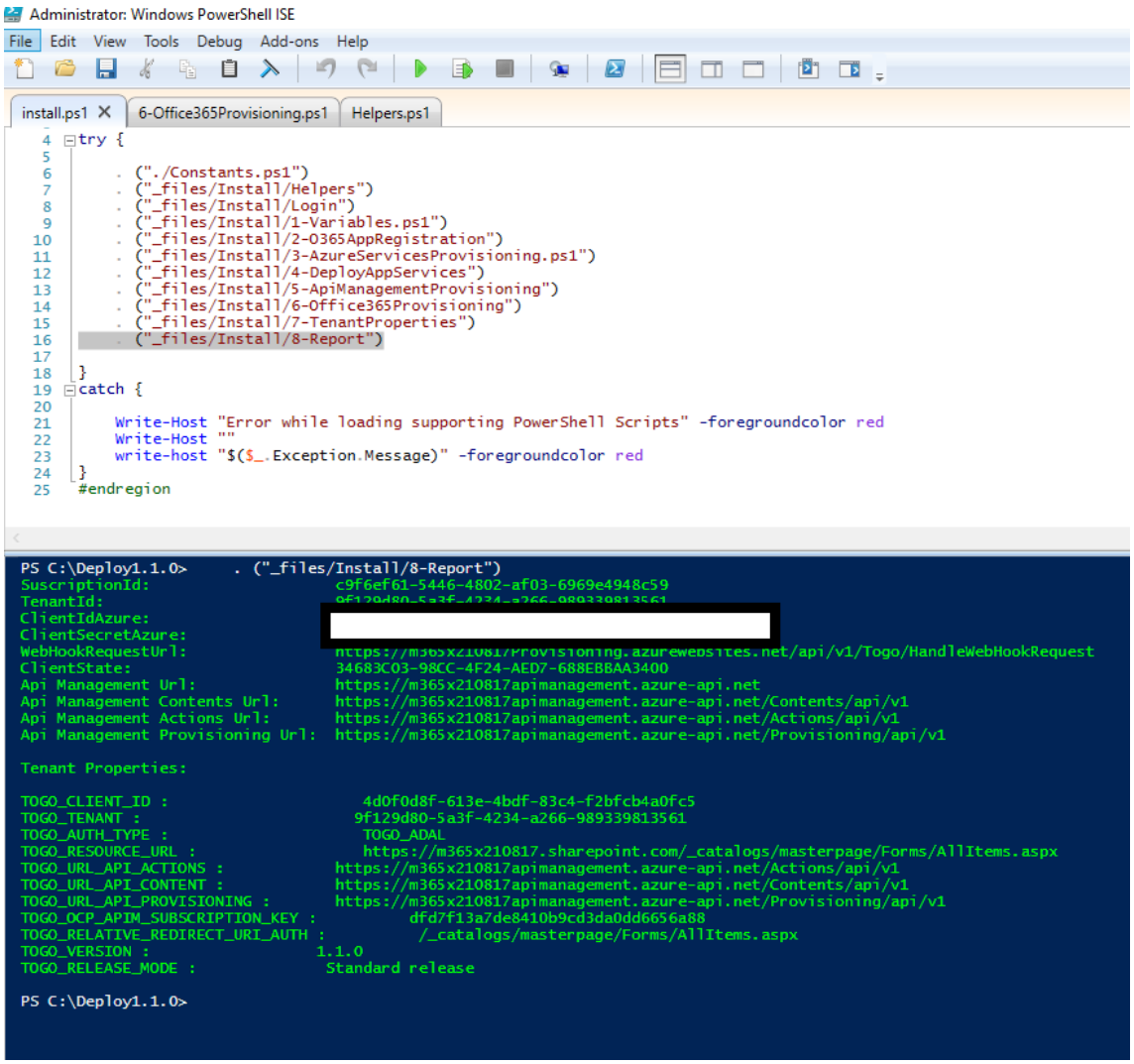




- Execute command line `. ("_files/Install/7-TenantProperties")`



- Execute command line . ("_files/Install/8-Report")



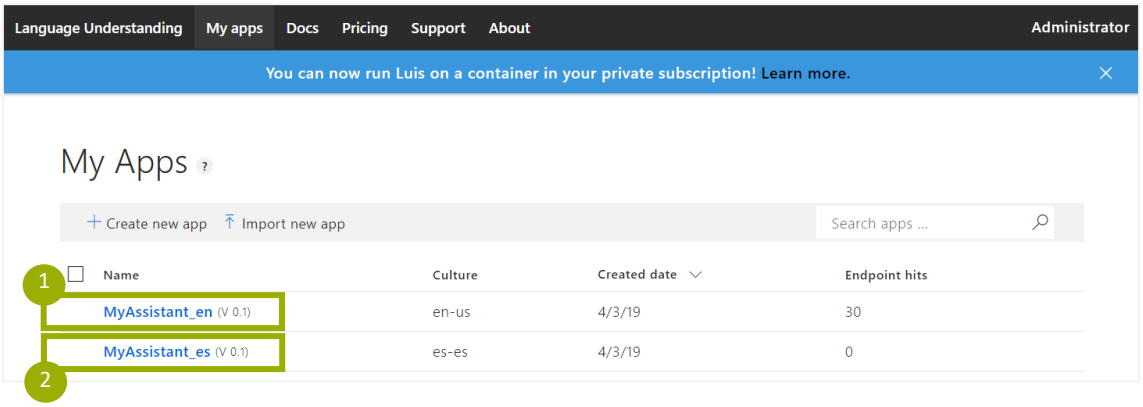
Post deployment actions

Link luis.ai applications with the Azure Subscription

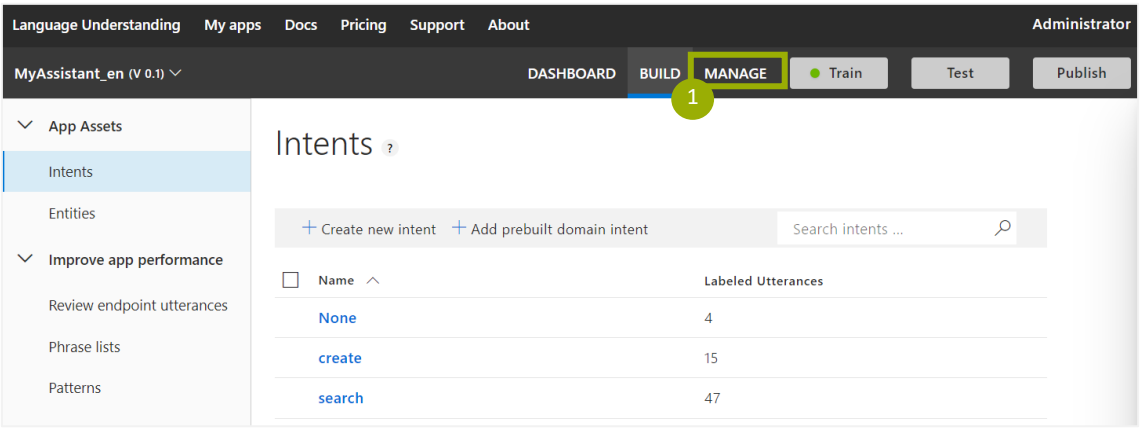
Once the deployment is finished is necessary to change the out-of-the-box configuration of the luis.ai applications.

Following are the steps required:

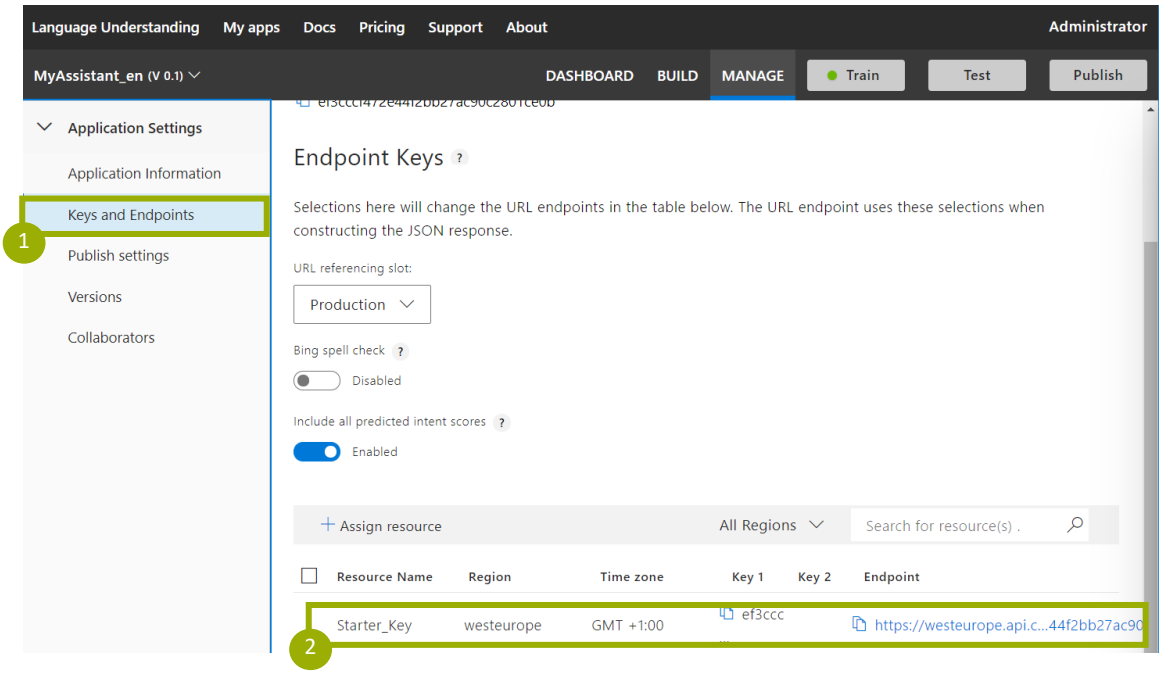
1. Navigate to https://eu.luis.ai
2. Sing in with an Azure Subscription Administrator user
3. Select an application



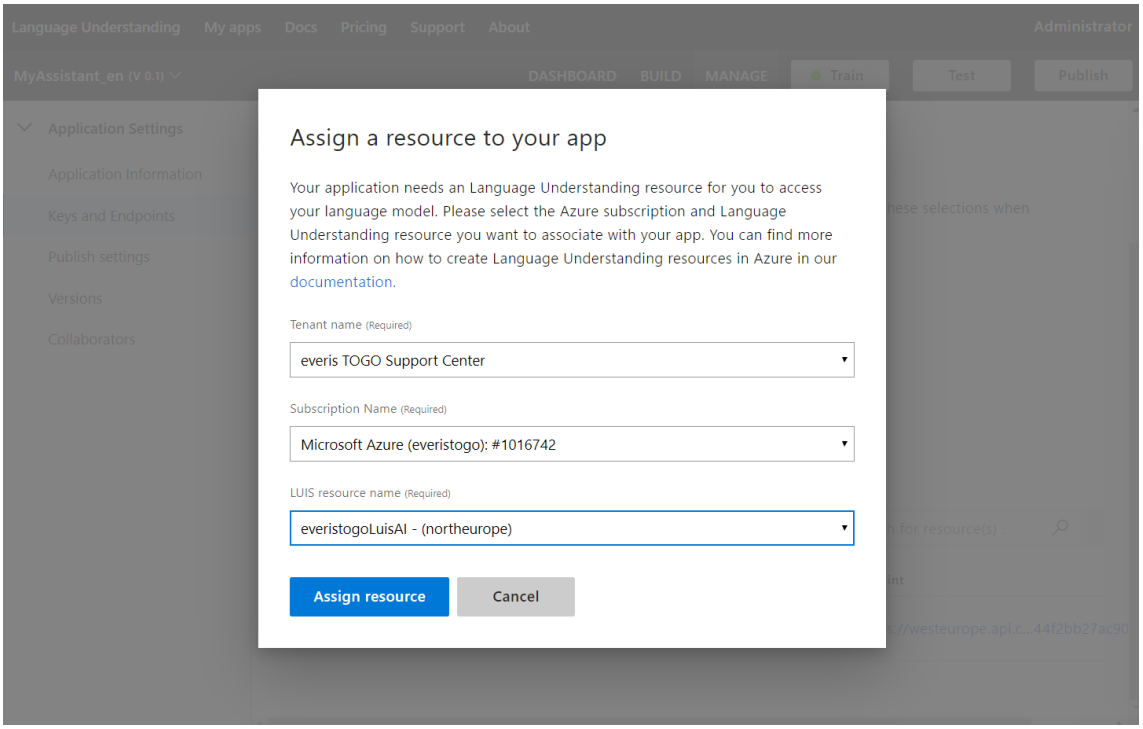
4. Click on manage



5. Click on Keys and EndPoints and check the associated subscription, it will be configured with an Starter_Key by default



6. Click on add resource and select the Azure tenant name, Subscription and everistogLuisAI that will be automatically created and click on assign resource

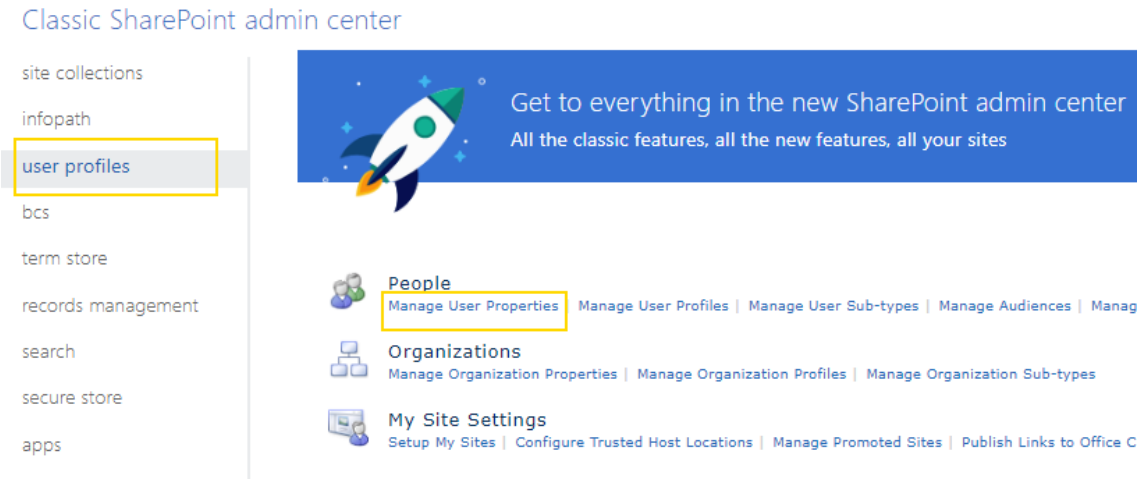


User profile custom Properties (Audience Targeting)

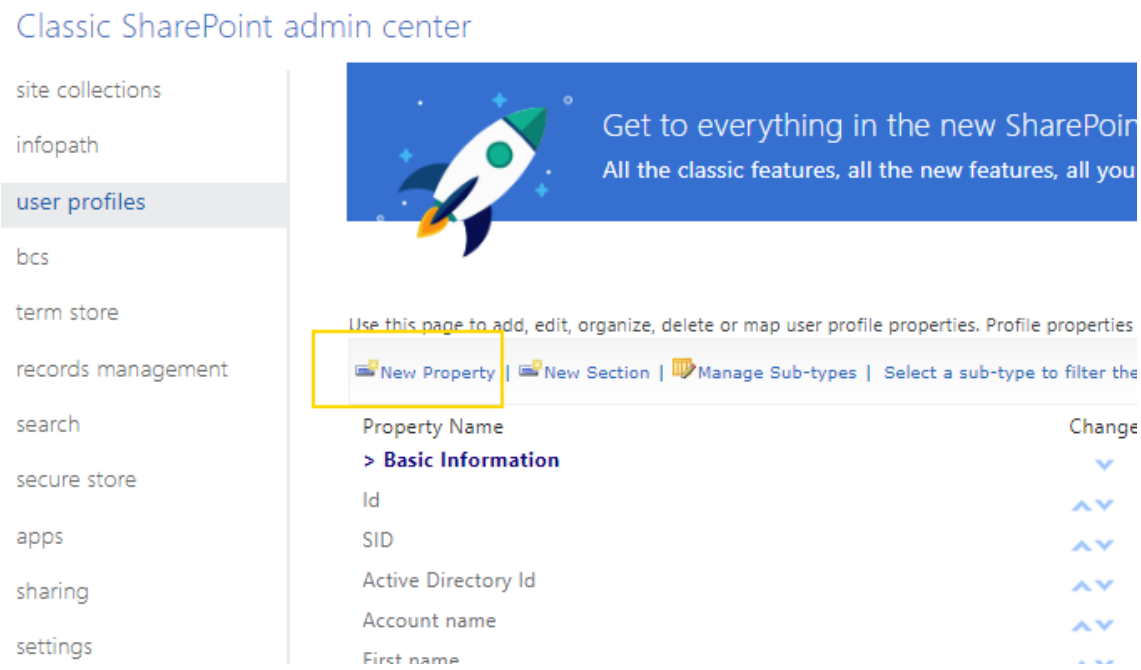
Following there is the description of the manual steps needed to create the two User Profile properties required to the Audience targeting business logic.

At the time of writing this guide the only way to create User Profile properties is through the classic experience of the SharePoint Admin Center.

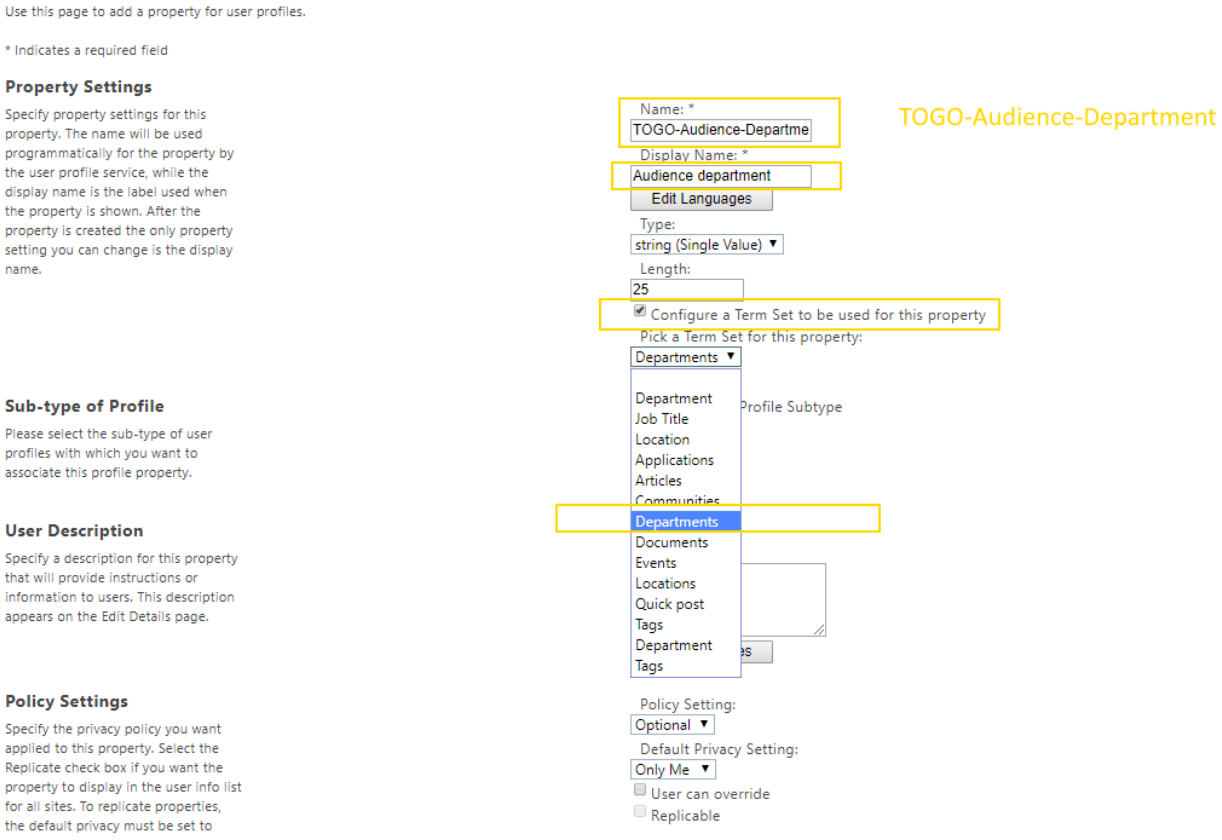
From the user profiles option, first select “Manage User Properties”



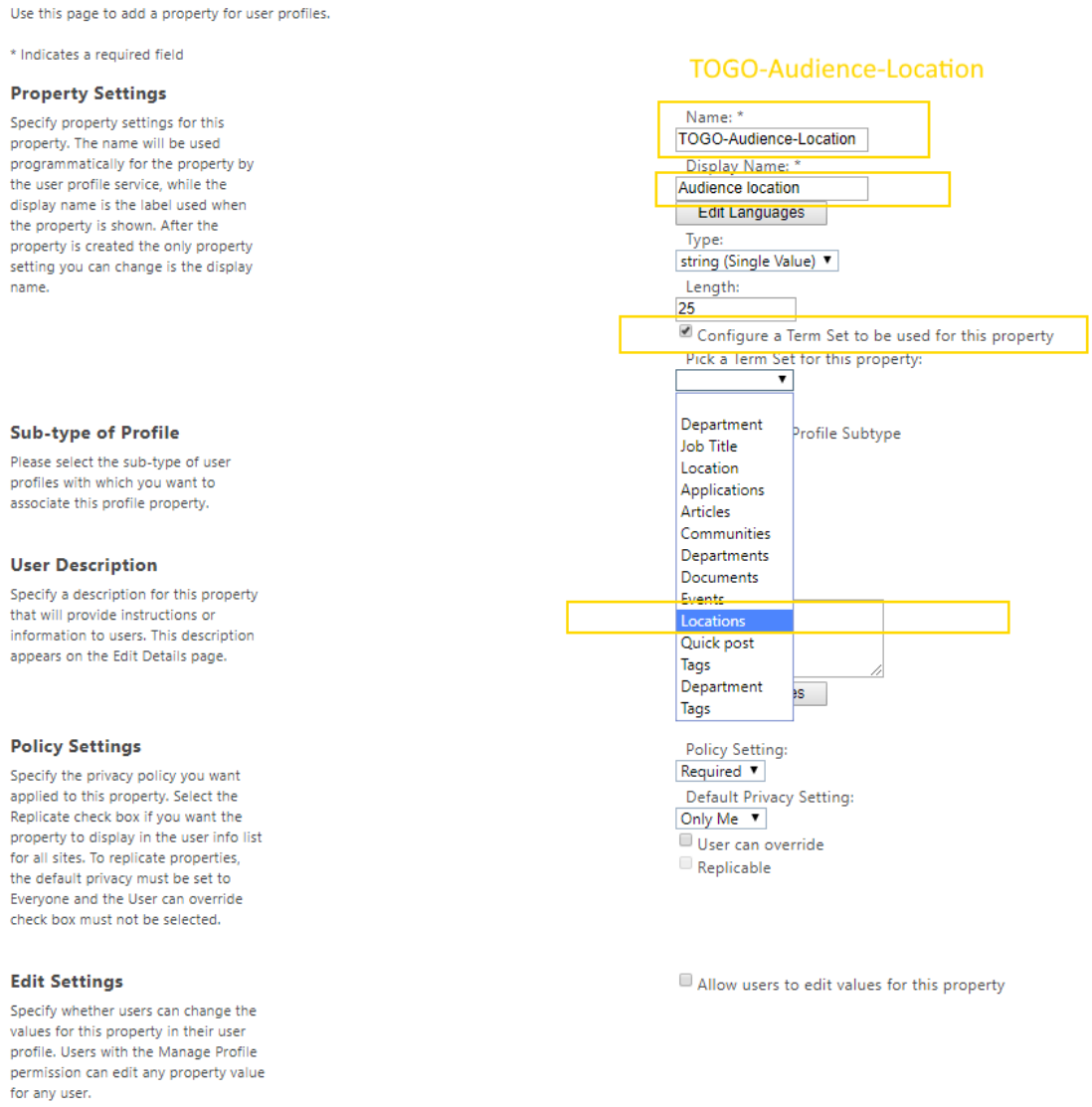
For each property we need, you need to create a “New property”



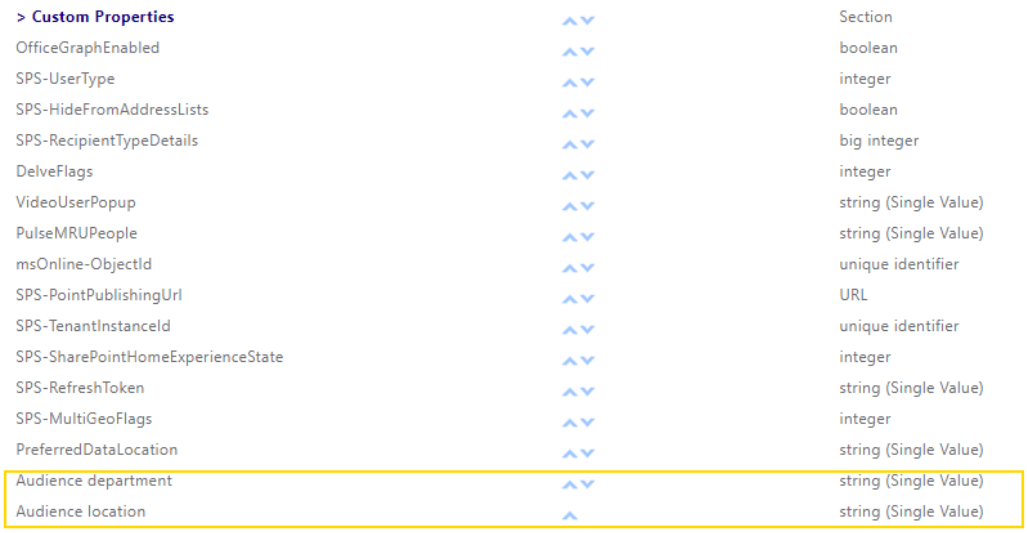
See below the details needed to create the user property involved in the department audience targeting: TOGO-Audience-Department



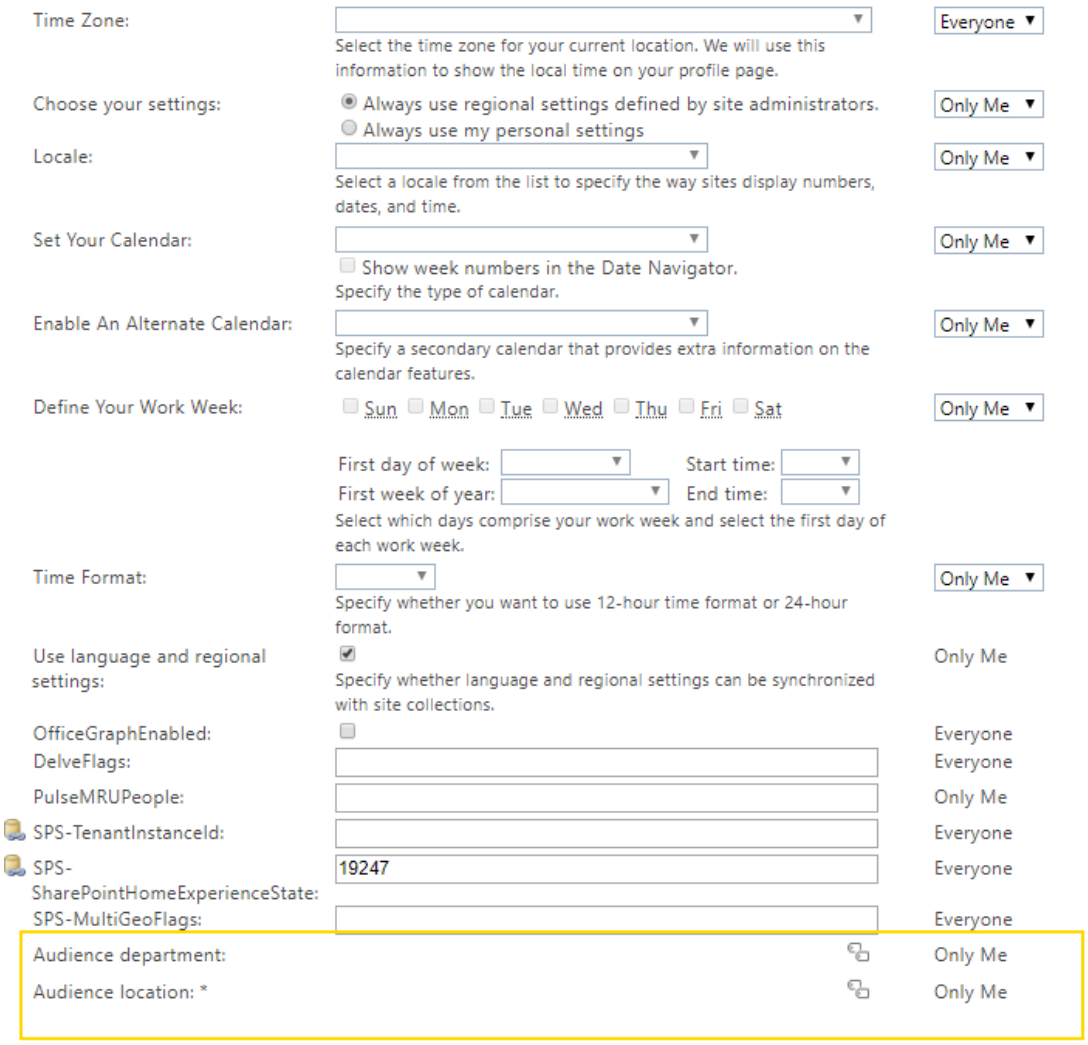
See below the details needed to create the user property involved in the location audience targeting: TOGO-Audience-Location



Once created the two required user properties, you should view them in the “Custom properties” section as depicted below:



From now on, the new two user properties will be displayed at the end of any user's property profile page and both properties should be associated to the corresponding termset.



Verification

Open your preferred browser

For internet explorer you need to configure it as depicted below