

Fundraising and Engagement

Deployment Guide

**Published**: May 12, 2021  
**Release Version**: v1.2

Contents

[Overview 4](#_Toc71557439)

[Getting Started 5](#_Toc71557440)

[Intended Audience 5](#_Toc71557441)

[Prerequisites 5](#_Toc71557442)

[Deploying Fundraising and Engagement for Dynamics 365 Sales Enterprise 8](#_Toc71557443)

[Acquire from AppSource 8](#_Toc71557444)

[Configuring Application User in Dynamics 9](#_Toc71557445)

[Provisioning the ARM Template 12](#_Toc71557446)

[Azure Components Created 12](#_Toc71557447)

[Deploying the ARM Template 13](#_Toc71557448)

[Adding the Azure SQL User Accounts 17](#_Toc71557449)

[Deploying the Fundraising and Engagement components to Azure 20](#_Toc71557450)

[Clone the Solution 20](#_Toc71557451)

[Build the Solution 22](#_Toc71557452)

[Deploy to Azure 23](#_Toc71557453)

[Populate the Azure SQL Database Schema 30](#_Toc71557454)

[Post Deployment Tasks (Dynamics 365) 31](#_Toc71557455)

[Assigning Security Roles 31](#_Toc71557456)

[Configuration Record Prerequisite 32](#_Toc71557457)

[Assigning Configuration Settings to Dynamics 365 Users 34](#_Toc71557458)

[Sample Data Import 35](#_Toc71557459)

[Updating to a newer version 37](#_Toc71557460)

[Prerequisites 37](#_Toc71557461)

[Updating the Fundraising and Engagement Dynamics solution through AppSource 37](#_Toc71557462)

[Updating Fundraising and Engagement Azure services 37](#_Toc71557463)

[Troubleshooting 43](#_Toc71557464)

[Failure Occurs During ARM Template Deploy 43](#_Toc71557465)

[Incorrect Parameter Value Entered when Updating to a New Version of the ARM Template 43](#_Toc71557466)

[Updating Visual Studio 2019 Community Edition (or Higher) 44](#_Toc71557467)

[Modifying the ARM Template 44](#_Toc71557468)

[Naming and Unique Name Concepts 44](#_Toc71557469)

[Hitting Maximum Path Length Limit When Deploying Azure Components 44](#_Toc71557470)

[Learn More About Azure and Extending Dynamics 365 45](#_Toc71557471)

# Overview

Fundraising and Engagement, built on Dynamics 365 Sales and part of [Microsoft Cloud for Nonprofit](https://aka.ms/MicrosoftCloudforNonprofit), is composed of two parts: Power Platform application installed through AppSource, and accompanying services running in Azure. The Azure part requires a single tenant Azure environment and is necessary for features such as recurring donation processing, Bank ACH batching, and automated financial summaries. The Azure services are deployed through a provided Azure [ARM template](https://aka.ms/fe-armtemplate).

This guide covers the installation of Power Platform and Azure, the configuration of the application, and general instructions for upgrading an existing installation to a newer version.

# Getting Started

This guide is intended to help implementors navigate the initial setup of a new Azure environment that is supporting a new instance of Dynamics 365 Customer Engagement which already has the Fundraising and Engagement managed solution installed and configured. Take a moment to review the requirements prior to moving ahead. The process requires that the user take note of the results of the creation of the Azure environment as they are needed to complete the setup process.

## Intended Audience

This guide assumes the user implementing the Fundraising and Engagement Azure environment has a solid understanding of Azure, Web Applications, and has completed AZ-900 or comparable accreditation at a minimum.

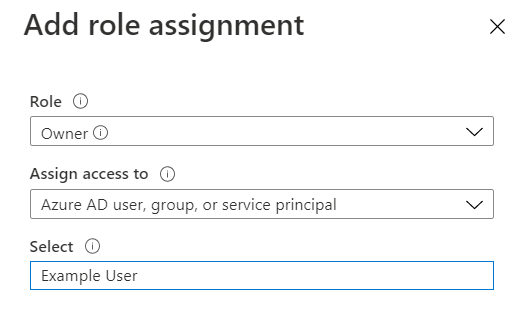
## Prerequisites

The following prerequisites are required to deploy the template and create the Azure components.

|  |
| --- |
| Note: The ARM template is not capable of rolling back components. This means if the ARM template fails as a result of the following prerequisites not being met, the user implementing the template must manually remove the created components before reattempting the deployment. |

### Azure Role for Template Deployment

The template must be deployed by a user that has the ‘Owner’ role assignment on the target subscription (see [Azure role assignments](https://docs.microsoft.com/en-us/azure/role-based-access-control/role-assignments-portal)). During the creation process, several steps assign access privileges and associate roles to application user accounts. Without the ‘Owner’ role, the ability to manage this process automatically via the template cannot occur.



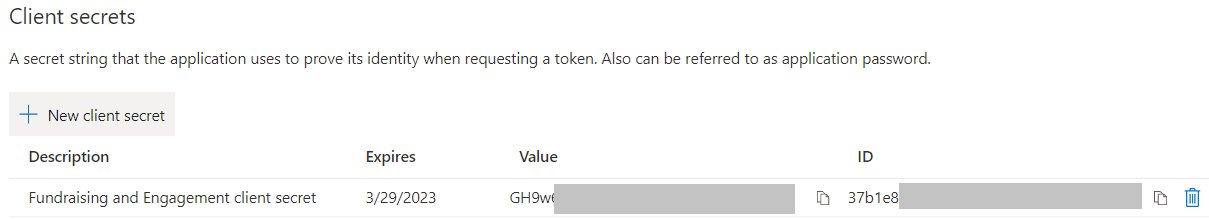
### Azure Active Directory Application Registration

An App registration must be created to allow access to the Power Platform environment with Fundraising and Engagement from the Azure services within the tenant. We will refer to it as ‘Fundraising and Engagement App registration ‘ in this guide. The App registration can be created in [Microsoft Azure Portal](https://portal.azure.com/) under Azure Active Directory > App registrations > New registration.

Graphical user interface, text, application, email

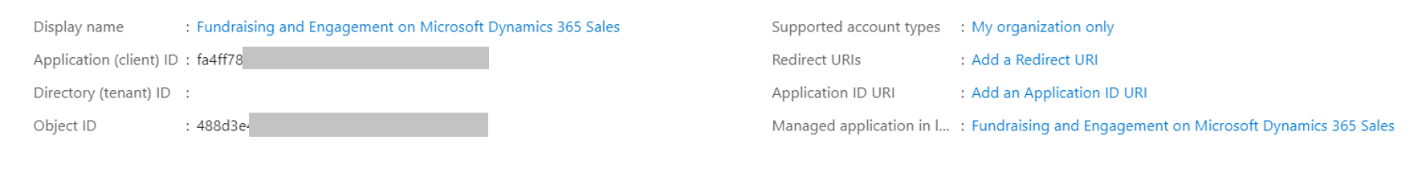
Description automatically generated

Once the App registration is set up, create and record a new client secret under ‘Certificates & secrets ‘ (see [Client Secrets](https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app#add-a-client-secret) in Azure documentation).



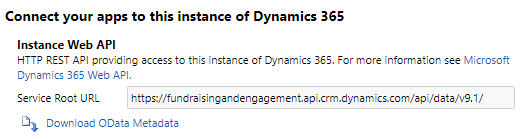
|  |
| --- |
| Note: The application client secret, once generated, cannot be retrieved after leaving the page. Be sure to record the value as it is needed in the completion of the template deployment. |

In addition to the client secret value, the ‘Application (client) ID’ of the app registration will also be required later in the process.



### Dynamics 365 Customer Engagement Instance Information

The Azure components configuration requires the Dynamics Customer Engagement API URL. You can find it in your Dynamics environment in settings under Advanced Settings -> Customizations -> Developer Resources.



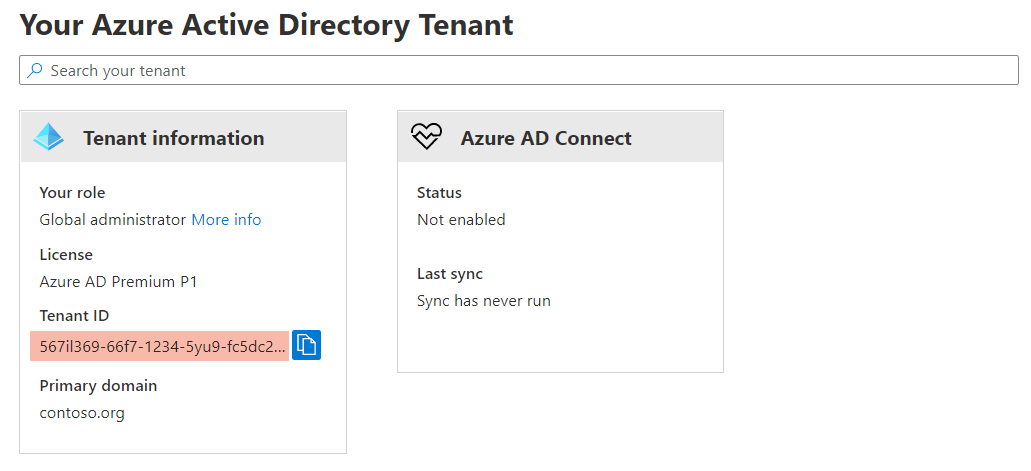
The value required will have the format https://<instancenameasdisplayed>.api.crmx.dynamics.com.

### Azure and Dynamics Tenant

The Azure environment you are creating the components in must be within the same Office 365 Azure Active Directory controlled tenant. Due to the Azure environment relying on server to server authentication by means of the application record in Azure Active Directory, the user must be visible to your Fundraising and Engagement for Dynamics 365 Sales solution instance.

### Retrieve Your Azure Directory Tenant ID

The Azure Active Directory Tenant Identifier is required during the setup of the Azure components. For the server-to-server authentication to work as expected, both the application identifier and the tenant identifier are required. The tenant identifier in Azure can be found by navigating to ‘Azure Active Directory’ and viewing the ‘Tenant ID’ from the overview pane.



### Configure SQL Server Management Studio (SSMS)

SSMS is required to access the Azure SQL database that is created post completion of the ARM template deployment. Get the latest version of [SQL Server Management Studio (SSMS)](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms).

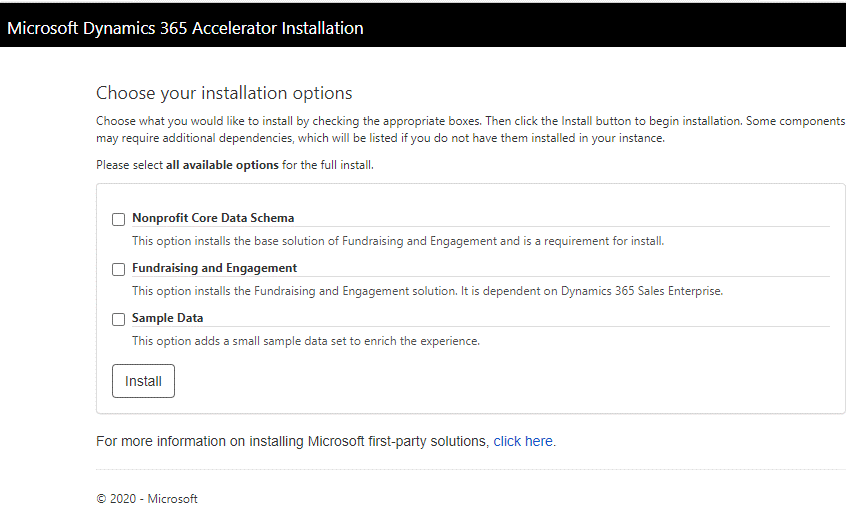
# Deploying Fundraising and Engagement for Dynamics 365 Sales Enterprise

## Acquire from AppSource

1. You can navigate to the [Fundraising and Engagement listing on AppSource](https://appsource.microsoft.com/en-us/product/dynamics-365/msnfp.msftnonprofitfundraisingandengagement). Or from the AppSource catalogue, navigate to Fundraising and Engagement by first going to the AppSource home page then searching for ‘Fundraising and Engagement.’ Once located, select ‘GET IT NOW’.



1. Follow the prompted guide to complete the process ensuring that the user completing the AppSource process has the ‘System Administrator’ role assigned.
2. You will be prompted to login to your tenant. From there select the Dynamics 365 instance you want to deploy Fundraising and Engagement. Follow the remaining prompted steps, such as the privacy policy and terms and conditions, both of which must be accepted for the solution to be installed.



The Nonprofit Core Data Schema is required for Fundraising and Engagement.

|  |
| --- |
| Note: To inspect the latest version of the Common Data Model (CDM) for Nonprofits and how it may impact those organizations who have already deployed the CDM, please review the latest release on [GitHub](https://github.com/microsoft/Industry-Accelerator-Nonprofit/releases). |

1. The Fundraising and Engagement application will appear once installed. With the application visible, this indicates a successful deployment.

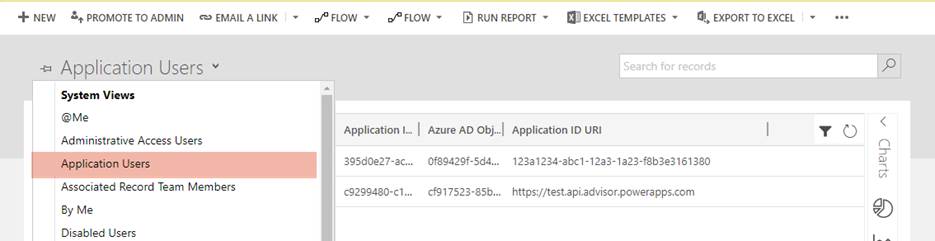
Once Fundraising and Engagement is successfully deployed, the user needs to set up the Configuration Record for the Dynamics components to work properly. Step by step instructions on how to set up the Configuration Record can be found in the [User Guide](https://aka.ms/feuserguide).

## Configuring Application User in Dynamics

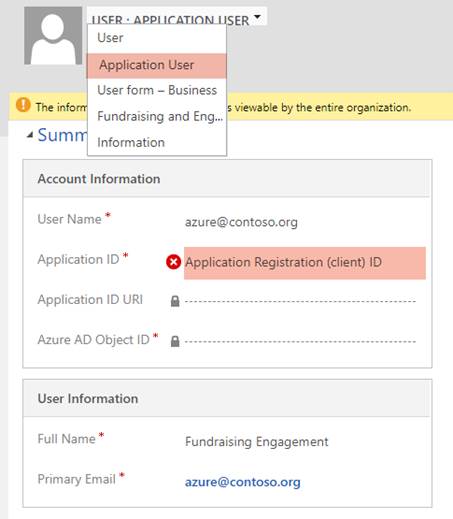
After Fundraising and Engagement is installed, an application user needs to be configured so that the Azure part can access the Dynamics environment.

### Create Application User and Assign Roles

1. Open settings for your Dynamics environment, go to Advanced Settings > Security > Users. Once there, change the view from the default ‘All Users’ to ‘Application Users.’



1. Click ‘+ New ‘ to create a new application user. Switch to the ‘Application User’ view and enter the ‘Application (client) ID’ value from [prerequisites](#_Azure_Active_Directory) to the ‘Application ID’ field. Although the user name, full name, and email address are required, the application does not validate those fields so any valid values can be entered there. Save the result with the Save button or the save icon in the bottom right corner.



1. Once the user is created, go to ‘Manage Roles’ and assign the ‘FundraisingandEngagement: Azure App User’ role to the user.

For additional details on configuring your ‘Application Users’ please review documentation on [setting up non interactive users as application users in Dynamics 365](https://docs.microsoft.com/en-us/power-platform/admin/create-users-assign-online-security-roles#create-an-application-user). For more details on administering privileges in Dynamics 365, see [Security Roles and Privileges](https://docs.microsoft.com/en-us/power-platform/admin/security-roles-privileges).

#### Assign Field Security Profile to Application User

After the user is created, it needs to be assigned the ‘Fundraising and Engagement Azure App User ‘ field security profile.

1. Open settings for your Dynamics environment by going to Advanced Settings > Security > Field Security Profiles.

Graphical user interface, text, application

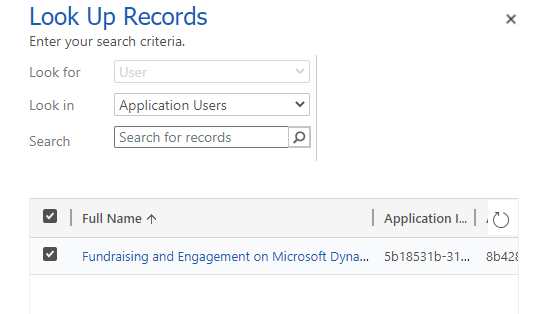
Description automatically generated

1. Open the ‘Fundraising and Engagement Azure App User ‘ profile and navigate to Users > Add.

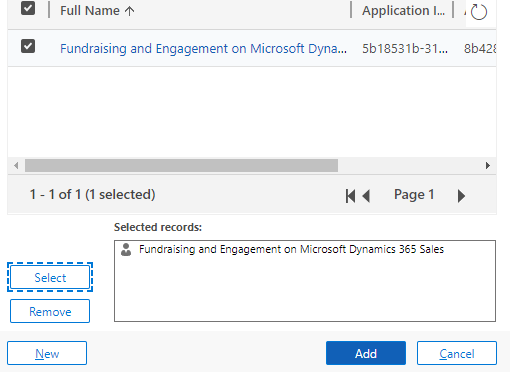
Graphical user interface, text, application, email

Description automatically generated

1. Select Application Users in the ‘Look in’ field, locate the application user you created in the previous steps and select it by checking its checkbox.



1. Click Select so that the user appears in ‘Selected records’ and confirm with the Add button.



# Provisioning the ARM Template

In this step, you will create the Azure components required by Fundraising and Engagement using a supplied [ARM template](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview).

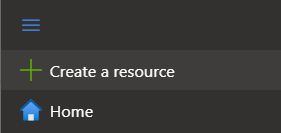
## Azure Components Created

The following Azure components are created during the deployment of the ARM template.

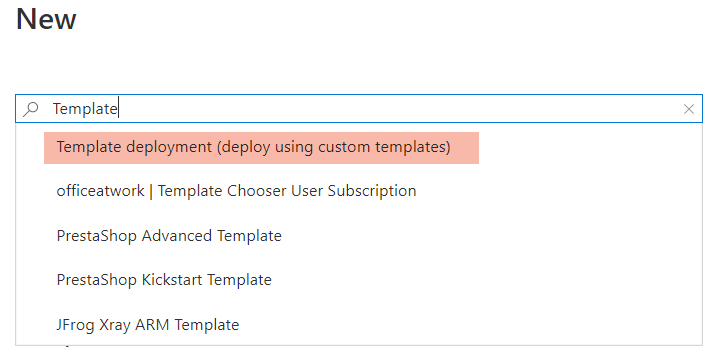
|  |  |  |
| --- | --- | --- |
| Component | Type | Purpose |
| API | App Service | The API used to send data from Dynamics 365 Customer Engagement to Azure. |
| Background Services | App Service | The Function processes to manage rollup queries and calculations, this primarily writes values back to Dynamics 365 Customer Engagement. |
| Bank Run | App Service | The application which generates and delivers the import file for the specified bank using the specified value. |
| Recurring Donation Engine | App Service | A process that manages the recurring gift process and generates the results in the form of Transactions that are written back to Dynamics 365 Customer Engagement. |
| App Service Plan | App Service Plan | The service plan created during the creation of the Azure environment. This plan can be edited after creation. |
| API Insights | Application Insights | The corresponding Application Insights instance that corresponds to the created API. |
| Background Services Insights | Application Insights | The corresponding Application Insights instance that corresponds to the created Background Services. |
| Bank Run Insights | Application Insights | The corresponding Application Insights instance which corresponds to the created Bank Run. |
| Recurring Donation Engine Insights | Application Insights | The corresponding Application Insights instance which corresponds to the created Recurring Donation Engine. |
| Vault | Key Vault | The secure storage that brokers sensitive information such as SQL passwords between Azure Applications. |
| Azure SQL Database | SQL Database | The Azure SQL Database which stores data passed to it from Dynamics. |
| Azure SQL Server | SQL Server | The Azure SQL server that is created by default in order to facilitate the Azure SQL Database. |
| Logging Storage | Storage Account | The storage account which holds the logging information from the recurring donation engine. |

## Deploying the ARM Template

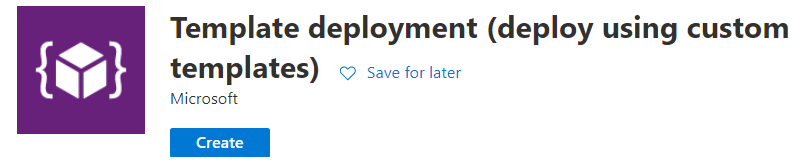
1. Login into your Azure Portal as the user with ‘Owner’ privileges for the subscription. The guide assumes you have accessed your Azure environment via <https://portal.azure.com>.
2. Create a new resource.



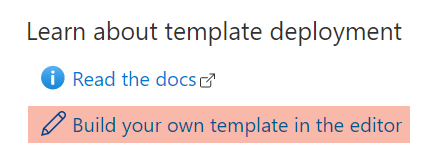
1. In the search box, enter ‘Template’ and select ‘Template deployment (deploy using custom templates).’



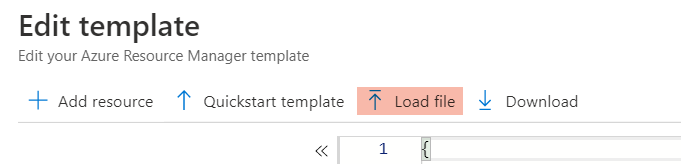
1. Select, ‘Create.’



1. Select ‘Build your own template.’



1. Select ‘Load file’ from within the ‘Edit template’ section and select the ARM template available [here at GitHub](https://aka.ms/fe-armtemplate).



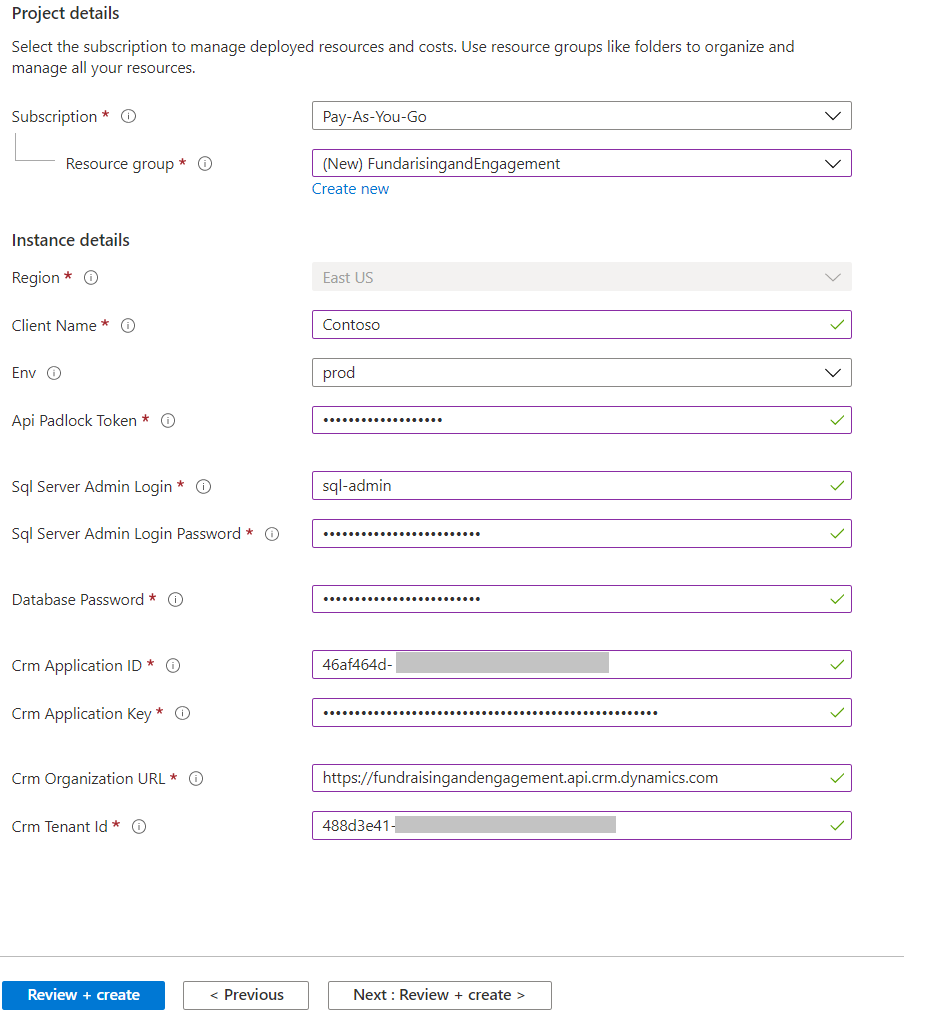
1. Select ‘Save’ to progress to the user inputs.
2. The template will require all inputs to be completed before the deployment. The specific inputs are:
   1. Basics section

|  |  |  |
| --- | --- | --- |
| **Setting** | **Description** | **Example** |
| *Subscription* | Select from an existing subscription, although multiple subscriptions may be visible, only those that the user deploying the template as the ‘Owner’ role set on can be used. | *Pay-As-You-Go* |
| *Resource Group* | Enter the resource group to be used, enter a descriptive name as all the Azure components will be assigned to this resource group and identifying those resources that are needed for Fundraising and Engagement will be beneficial to a demonstrator.  The resource group is assumed to already exist before deployment. | *FundraisingandEngagement* |
| *Location* | The resource locations for these components. All resources are assigned to specific geographic locations based on the group. Locations can be moved post deployment. | *(US) Central US* |

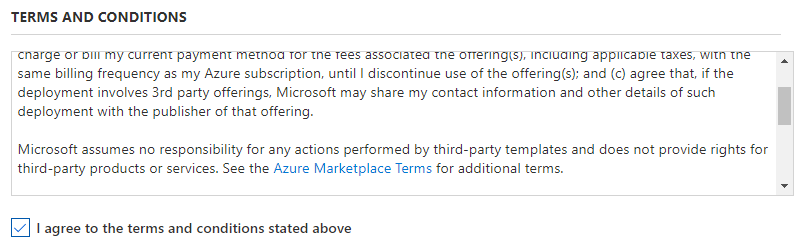
* 1. Settings section

|  |  |  |
| --- | --- | --- |
| **Setting** | **Description** | **Example** |
| *Client Name* | The short name for the client. This value acts as the suffix prefix applied to the components created during this process forming a unique name for each application.  Because the value is used as the prefix of resource names, some limitations apply. E.g., the value can contain only numbers and letters and the maximum length is limited. | *Contoso* |
| *Env* | Select from Dev, QA, UAT, Test and Prod. Select the option that best suits the purpose of this deployment. This acts as a prefix to the naming of the created applications. | *Prod* |
| *API Padlock Token* | **Deprecated.** The hashed padlock token value generated according to the Generating a Padlock Token section in this guide. | *Should be empty* |
| *Sql Server Admin Login* | The SQL Server Administrator account that will be used when creating the Azure SQL Server. It will be important to record these details securely as they are required to complete additional configuration. | *sql-admin* |
| *Sql Server Admin Login Password* | The SQL Server Administrator password that will be used when creating the Azure SQL Server. It will be important to record these details securely as they are required to complete additional configuration. | *An example password* |
| *Database Password* | The SQL Database user password that will be stored as a secret in the secure key vault. | *An example password* |
| *App Registration Client Id* | The ‘application (client) id’ obtained from the Fundraising and Engagement App registration in Azure Active Directory during the [prerequisite](#_Azure_Active_Directory) steps. | *39b994e7-6430-4969-9…* |
| *App Registration Client Secret* | The client secret obtained from the Fundraising and Engagement App registration in Azure Active Directory during the [prerequisite](#_Azure_Active_Directory) steps. | *W+GLdvquGb& …* |
| *Crm Organization URL* | The Dynamics 365 for Sales Enterprise API URL. | *https://fundraisingandengagement.api.crm.dynamics.com* |
| *Crm Tenant Id* | The Azure Active Directory identifier. | *488d3e41-7005-4a64-8…* |

1. Using the above example inputs, the template will look like this:



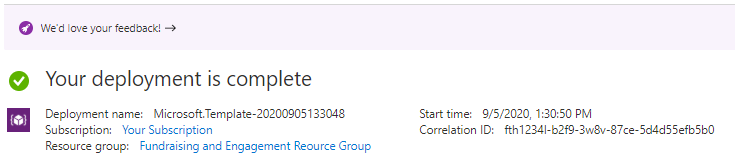
1. To start the process, select ‘I agree to the terms and conditions stated above.’



1. Proceed to create.
2. Review the completed process. Any errors presented during the deployment of the template will result in an administrator needing to delete the components, correct the input error, and re-run the template.
3. The process will create several resources, including the following:

|  |  |
| --- | --- |
| **Azure Resource** | **Example Name (Using the Above Example)** |
| Service Plan | Contoso-AppServicePlan |
| SQL Server | Contoso-sqlserver-prod |
| SQL Database | Contoso-db-prod |
| SQL Database Username | Contoso\_prod\_user |
| Secure Key Vault | Contoso-vault-prod |
| Azure Storage Account | Contosostorageprod |
| Azure Function | Contoso-BackgroundServices-prod |
| Azure Web Application (API) | Contoso-ApiApp-prod |
| Azure Web Job (Recurring Donation Engine) | Contoso-RecurringDonationApp-prod |

1. The deployment will alert the user to the success of the deployment. A successful deployment presents the following:



If the deployment fails for some reason, it can be retried with the ‘Redeploy ‘ button:

Graphical user interface, text, application

Description automatically generated

## Adding the Azure SQL User Accounts

Once the template has successfully completed the creation of all Azure components, the application SQL user account needs to be set up within the Azure SQL Database.

|  |
| --- |
| Note: SQL Server Management Studio is recommended as some of these steps must be performed against the ‘master’ database of the SQL Azure Server. You can download the latest version of [SQL Server Management studio](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms). To enable access from SQL Server Management Studio to the SQL Azure Server, follow these steps on how to set firewall rules using the [Azure portal](https://docs.microsoft.com/en-us/azure/azure-sql/database/firewall-configure#use-the-azure-portal-to-manage-server-level-ip-firewall-rules). |

### Connect to SQL Server

Open SQL Server Management Studio and connect to the Azure database. The name of the server to connect to has the format <client\_name>-sqlserver-<env>.database.windows.net and can be found under ‘Server name ‘ in the detail of SQL server provisioned in the target resource group by the ARM template.

Text

Description automatically generated with medium confidence

Use the SQL Server admin credentials provided as parameters of the ARM template for login and password.

Graphical user interface, text, application, email

Description automatically generated

SQL Server Management Studio may ask you to enable access exception in the database firewall if needed. Firewall exceptions can also be [configured](https://docs.microsoft.com/en-us/azure/azure-sql/database/firewall-configure#use-the-azure-portal-to-manage-server-level-ip-firewall-rules) directly in Azure Portal if Studio is not able to do so automatically.

Graphical user interface, text, application

Description automatically generated

### Create Login in Master Database

Open a query window for the master database (select New Query in the context menu for Databases > System Databases > master).

Diagram

Description automatically generated

The login name you create needs to follow the naming convention ‘<Client Name>\_<Env>\_user’ where <Client Name> and <Env> are values of the ARM template parameters. Using the examples listed above, the SQL login name will be *Contoso\_prod\_user.*

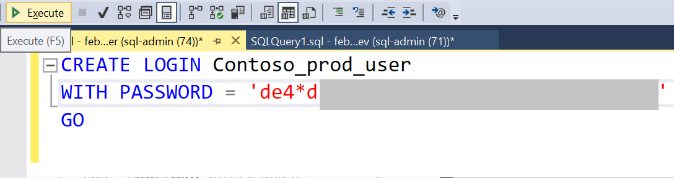
Paste the following script to the query window, replacing ‘SQL\_Login\_Name’ with the login name above and password with the Database Password parameter of the ARM template.

CREATE LOGIN SQL\_Login\_Name

WITH PASSWORD = 'Password as entered during ARM Template configuration'

GO

Execute the script.



### Create User in Application Database

Open another query window for the application database located under Databases and named <Client Name>-db-<Env> (e.g. contoso-db-prod).

Graphical user interface, text, application, email

Description automatically generated

Paste and execute the following script, again replacing all occurrences of ‘SQL\_Login\_Name’ with the login name above:

CREATE USER SQL\_Login\_Name

FOR LOGIN SQL\_Login\_Name

WITH DEFAULT\_SCHEMA=dbo

GO

EXEC sp\_addrolemember N'db\_owner', N'SQL\_Login\_Name'

GO

# Deploying the Fundraising and Engagement components to Azure

These steps assume the Azure environment is now ready to be populated by the applications that power the Fundraising and Engagement solution. Each component created must be in place before completing the steps outlined in this section.

The following prerequisites are required to deploy the template and create the Azure components:

* Visual Studio 2019 Community Edition or Higher
* Access to the latest [GitHub repository for Fundraising and Engagement](https://github.com/microsoft/fundraising-and-engagement)
* The latest updates to [Visual Studio 2019 Community Edition](https://visualstudio.microsoft.com/vs/community/) or higher.

|  |
| --- |
| Note: This guide assumes the user populating the Azure environment has a basic understanding of Visual Studio. Visual Studio 2019 Community Edition or higher is the only supported version. Download the latest version of [Visual Studio 2019](https://visualstudio.microsoft.com/downloads/), a free development tool. |

## Clone the Solution

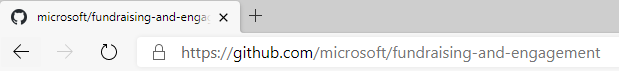
The goal is to clone the latest GitHub repository to your local Visual Studio instance. This guide covers only the initial clone.

|  |
| --- |
| Note: If you have already deployed the Fundraising and Engagement Azure components using this guide, connect to [GitHub](https://github.com/microsoft/Industry-Accelerator-Nonprofit/releases) to retrieve the latest solution. Always use the most up to date version of the solution when deploying to Azure. |

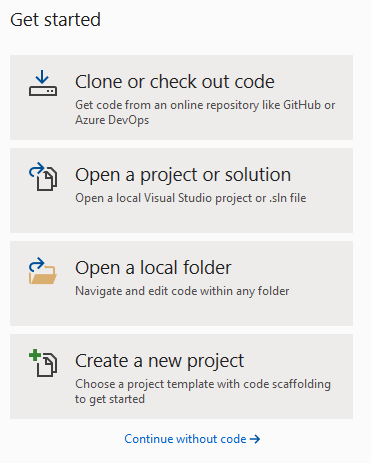
1. Retrieve the latest GitHub repository URL by locating the most recent repository and selecting ‘<> Code.’



Copy the URL that now displays in the browser window.

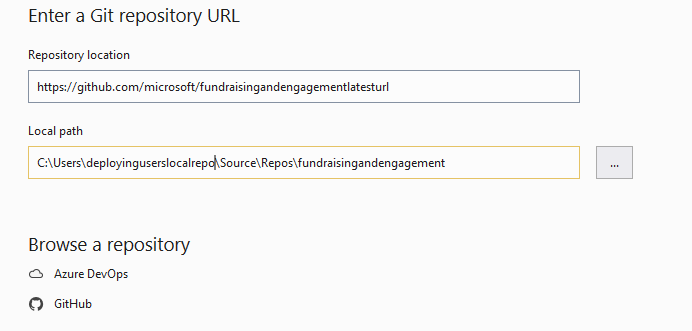


1. Open Visual Studio and select ‘Clone or check out code.’

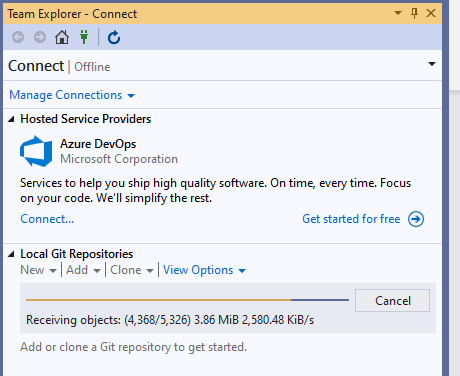


1. Enter the previously copied GitHub URL. You now have an option to change the local path.

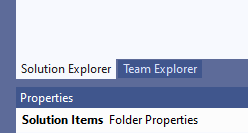
|  |
| --- |
| Note: If you have already deployed the Fundraising and Engagement Azure components using this guide, to avoid a conflict, you may want to change the local path to indicate the most recently cloned release from GitHub. |



1. The solution will indicate it is cloning successfully by presenting the progress bar.



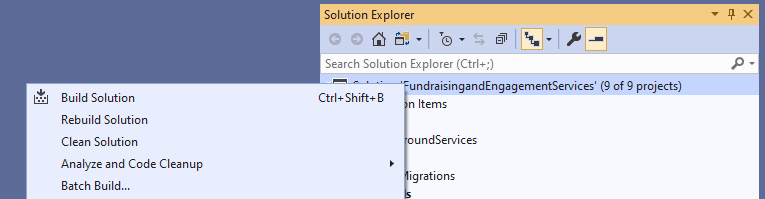
1. Change the view to default on the ‘Team Explorer’ tab.
2. Open the services solution by double clicking on ‘FundraisingandEngagementServices.sln.’
3. Navigate back to the solution by selecting the ‘Solution Explorer’ tab.



## Build the Solution

The solution needs to be built locally prior to deploying to Azure. The process of building the solution to deploy does not require running it locally (only the build step is required).

To build the solution, right click on the solution and select ‘Build Solution.’



There are nine build steps. The user can confirm that the build was successful by verifying the output:

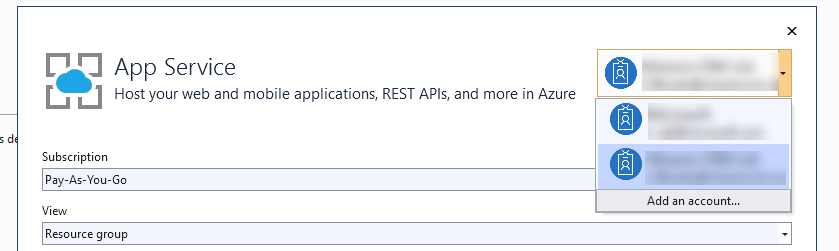
Sample screenshot of step #9 showing successful verification

## Deploy to Azure

Deploying the solution to the previously created Azure components requires that the user deploying the solution has access to those that were previously created. This guide assumes the user creating the Azure components is using the same access credentials as the user that then deploys the Fundraising and Engagement solution.

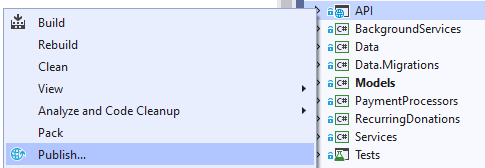
There are three components that require deploying to Azure: the API, RecurringDonations, and the BackgroundServices. Follow these steps to deploy each Azure application.

Note: Visual Studio 2019 Community Edition supports having multiple profiles for Azure deployment. If you already have some profiles, make sure the profile in use matches the desired target for deployment:

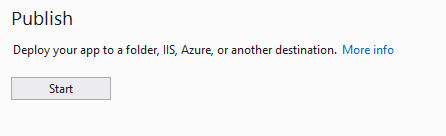


### API

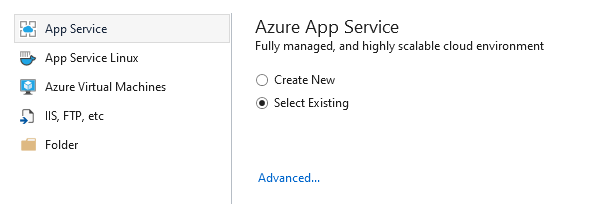
1. Right click on the API Visual Studio folder and select ‘Publish.’



Select ‘Start.’



1. Select ‘App Service’ (this is the default) and ensure that ‘Select Existing’ is checked.



1. Select ‘Create’ and ensure the profile in use is the profile that has ‘contributor’ rights to the Azure deployment. Select the subscriptions, change the view to ‘Resource Group,’ then locate the ‘API’ created during the Azure deployment steps.



1. Select ‘OK’ then ‘Publish.’

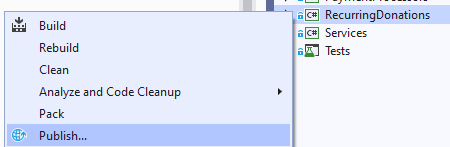


1. Verify the success by ensuring the ‘Output’ section displays the successful deployment.

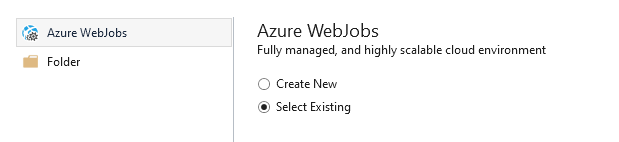


### Recurring Donation

1. Right click on the ‘RecurringDonations’ Visual Studio folder and select ‘Publish.’



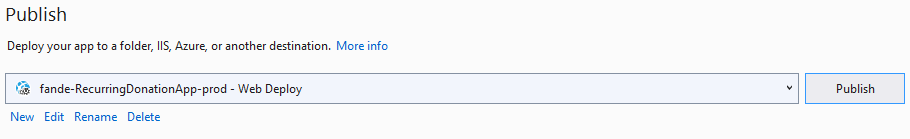
1. Select ‘Azure WebJobs’ then ‘Select Existing.’



1. Select ‘Finish’ and ensure the profile in use is the profile that has ‘contributor’ rights to the Azure deployment. Select the subscriptions, change the view to ‘Resource Group,’ then locate the ‘Recurring Donation App’ created during the Azure deployment steps.



1. Select ‘OK’ then ‘Publish.’



1. Verify success by ensuring the ‘Output’ section displays the successful deployment.

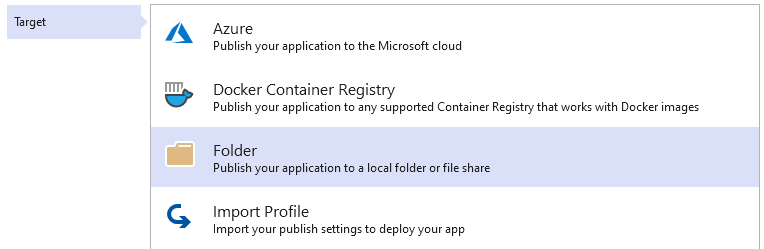


### Background Services

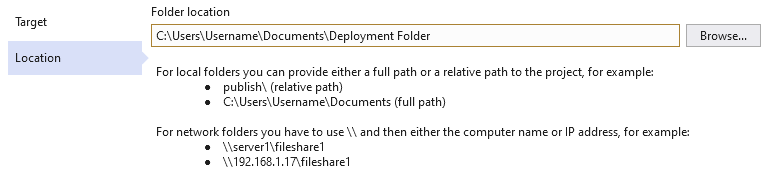
1. Right click on the ‘BackgroundServices’ folder in Visual Studio folder and select ‘Publish.’



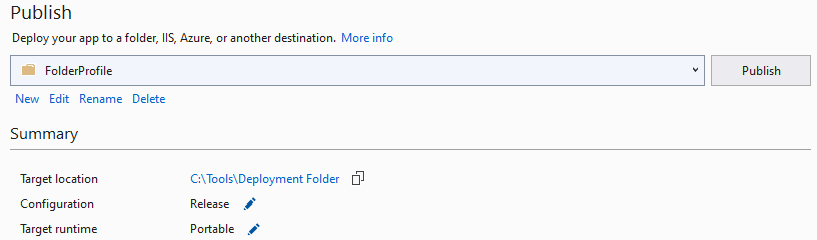
1. Select ‘Folder’ then ‘Next.’



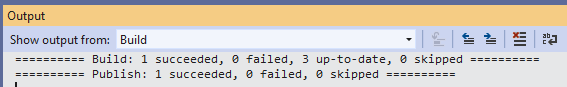
1. Select a local folder repository that you have permission to write to e.g. C:\Users\Username\Documents\Deployment Folder.



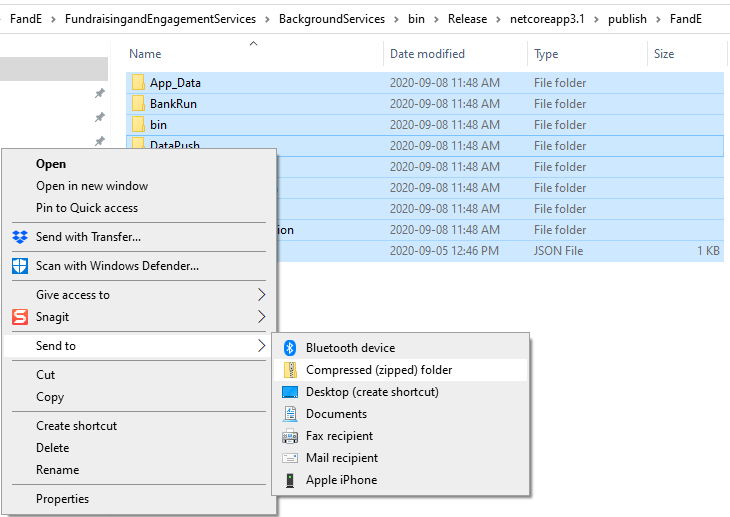
1. Select ‘Publish.’



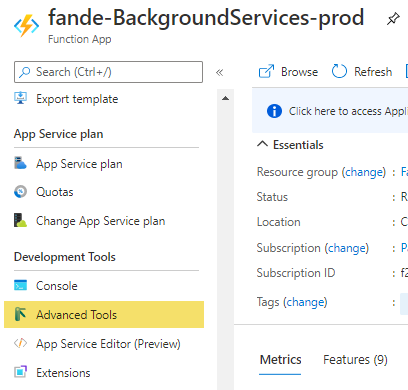
1. Confirm the build was successful.



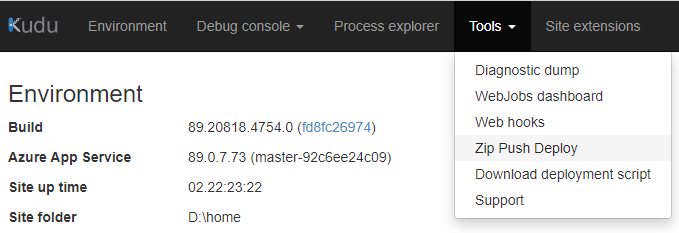
1. Navigate to the deployment folder, select all folders in the deployment path, and add them to a zip file.



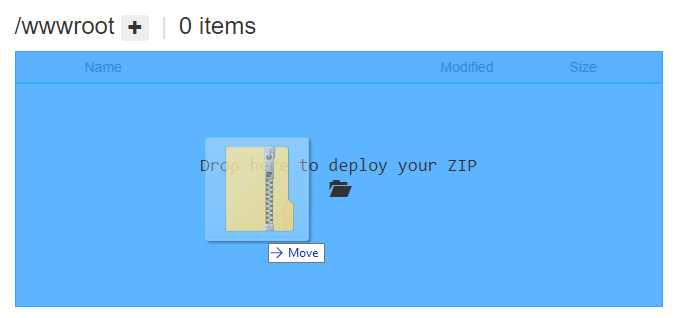
1. Navigate to the originally created function app within Azure, select ‘Advanced Tools,’ then select ‘Go.’



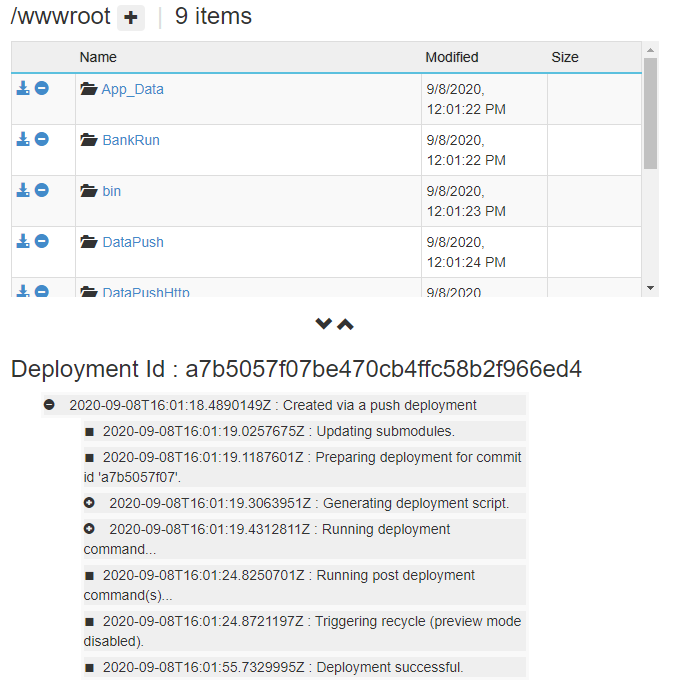
1. From the explorer tools, select ‘Tools’ then ‘Zip Push Deploy.’



1. Drag and drop the zip file directly to the open browser window.



1. The process will complete and display a successful deployment message.



1. Your deployment is complete. All three Azure components are now deployed.

## Populate the Azure SQL Database Schema

To populate the newly created database, open a new query window for the application database in Microsoft SQL Server Management Studio the same way as described in section ‘Create User in Application Database’ of this guide.

Graphical user interface, text, application, email

Description automatically generated

Locate the file named migrations.sql in the asset files associated with the [GitHub release](https://github.com/microsoft/fundraising-and-engagement/releases) you are deploying. Paste the contents of the file to the query window and press Execute.

Graphical user interface, text, application

Description automatically generated

Successful result should look like this:

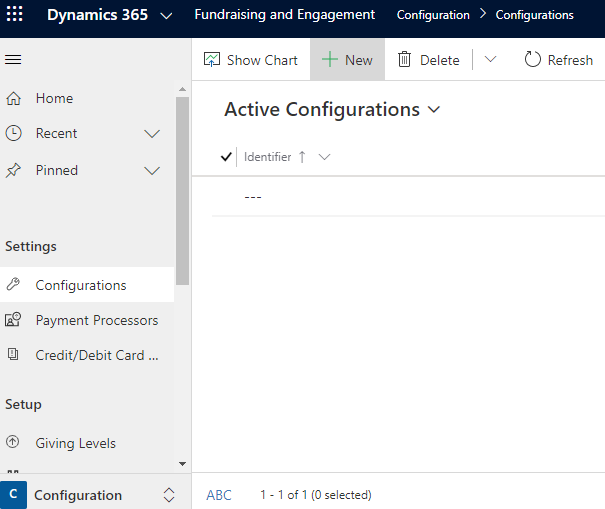
Text

Description automatically generated with medium confidence

# Post Deployment Tasks (Dynamics 365)

Additional key steps are required after an administrator has set up the Fundraising and Engagement Dynamics 365 environment. At least one configuration record populated with the necessary Azure information is required and all system users must have a configuration record assigned to their user record prior to use.

The configuration record can be created and edited in the Fundraising and Engagement application. Go to the ‘Configuration’ area at the bottom left of the navigation pane and select ‘Configurations.’



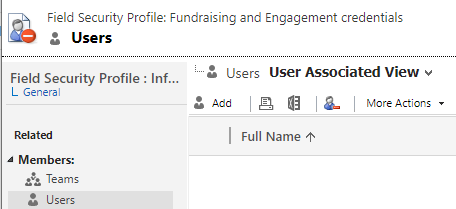
## Assigning Security Roles

Some configuration is necessary before all features of Fundraising and Engagement can be used. The Configuration record in particular needs to be created. If this is not done by a user with the System Administrator role, the user needs both the appropriate role and field security profile:

* The ‘FundraisingandEngagement: System Administrator’ role.



* The ‘Fundraising and Engagement credentials’ field security profile (Managed in Advanced settings > Security > Field Security Profiles).



## Configuration Record Prerequisite

The Configuration Record requires several specific Azure-dependent configuration settings to be set prior to use. These five configuration settings have been listed here along with how to retrieve the value from Azure. Note that the labels of the following options match the label of the configuration record in Fundraising and Engagement.

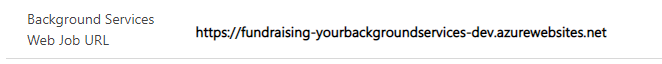
1. **Azure Web API URL and API Padlock Token**

Azure Web API URL and API Padlock Token **are deprecated** since version 1.2.3.0 and may be removed in future versions. Both values should remain empty.

Graphical user interface, application

Description automatically generated

1. **Background Services Web Job URL**



1. Locate the Azure Background Functions application.
2. Under Overview, view the URL.
3. Paste the URL into the ‘Background Services Web Job URL’ field as is – the full URL must read: <https://fundraising-yourbackgroundservices-dev.azurewebsites.net>.
4. **Bank Run Security Key**



1. Locate the Azure Background Functions application.
2. Under Functions, select Functions.
3. Navigate to and select the ‘BankRun’ function.
4. Once the function opens in the new window, select ‘Overview.’
5. Select ‘Get Function Url.’
6. In the URL, copy the entire value after the ‘?code=’
7. Update the ‘Bank Run Security Key’ value with that of the copied URL value. The result should look like the following (your exact value will be different): SdPgNdTKyUMEZe2BCHsdkcmonm4t8ngfnwmc2o5Pu99g78ks7RzuiHnmODbECfPSQ==

See also the [Azure documentation on obtaining Function Keys](https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-http-webhook-trigger?tabs=csharp#obtaining-keys).

1. **Yearly Giving Security Key**



1. Locate the Azure Background Functions application.
2. Under Functions, select Functions.
3. Navigate to and select the ‘YearlyGivingFromEntity’ function.
4. Once the function opens in the new window, select ‘Overview.’
5. Select ‘Get Function Url.’
6. In the URL, copy the entire value after the ‘?code=’
7. Update the ‘Yearly Giving Security Key’ value with that of the copied URL value. The result should look like the following (your exact value will be different): a5RSdPgNdTKyUMEZe2BCHsdkcmonm4t8ngfnwmc2o5Pu99g78ks7RzuiHnmODbECfPSQ==
8. **Event Receipting Security Key**

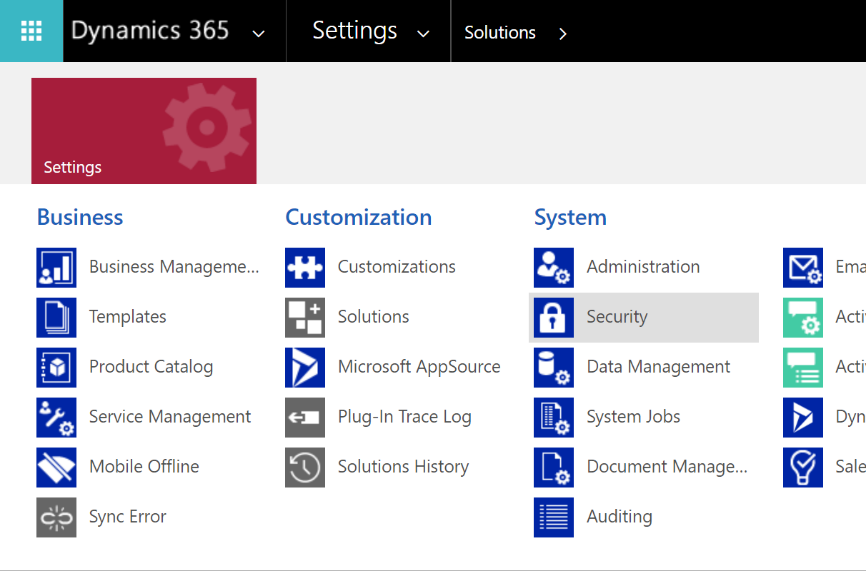


1. Locate the Azure Background Functions application.
2. Under Functions, select Functions.
3. Navigate to and select the ‘EventReceipting’ function.
4. Once the function opens in the new window, select ‘Overview.’
5. Select ‘Get Function Url.’
6. In the URL, copy the entire value after the ‘?code=’
7. Update the ‘Event Receipting Security Key’ value with that of the copied URL value. The result should look like the following (your exact value will be different): a5RSdPgNdTKyUMEZe2BCHsdkcmonm4t8ngfnwmc2o5Pu99g78ks7RzuiHnmODbECfPSQ==

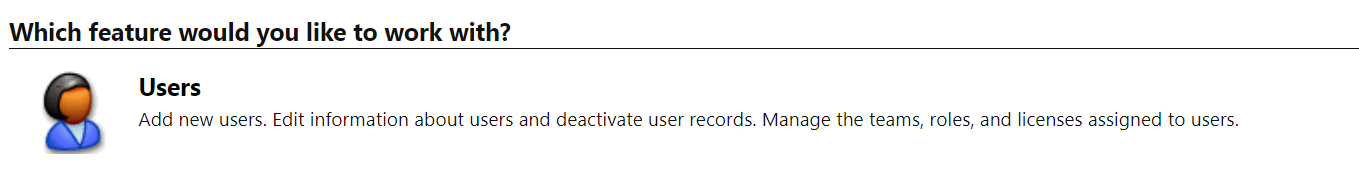
## Assigning Configuration Settings to Dynamics 365 Users

All users that need to interact with the Fundraising and Engagement business logic must have a configuration record assigned, **including the application user** [configured above](#_Configuring_Your_Application). Users are limited to a single configuration record assigned at any one time. To associate a configuration record to a user in Dynamics, log in to Dynamics 365 as a user that has the System Administrator role and complete the following steps.

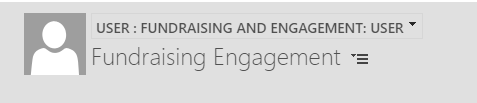
1. Navigate to ‘Advanced Settings.’
2. Navigate to System and then select ‘Security.’



1. Select Users.



1. Retrieve and open the user record you want to assign the configuration record to.
2. Select the ‘Fundraising and Engagement: User’ view.



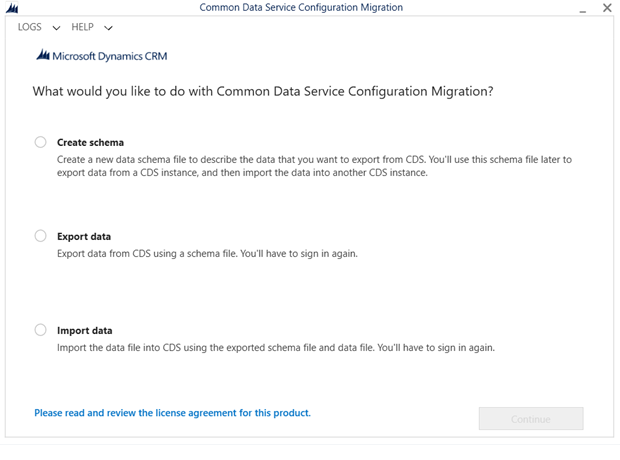
1. Use the lookup to set the correct configuration record in the ‘Configuration’ field and then hit save and close.



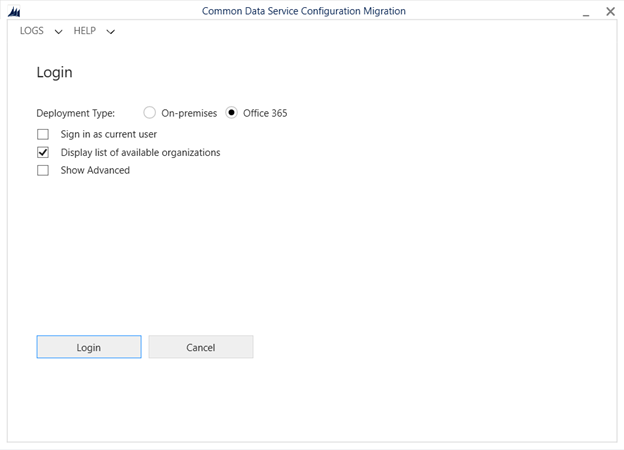
## Sample Data Import

To import sample data into Fundraising and Engagement, complete the following steps.

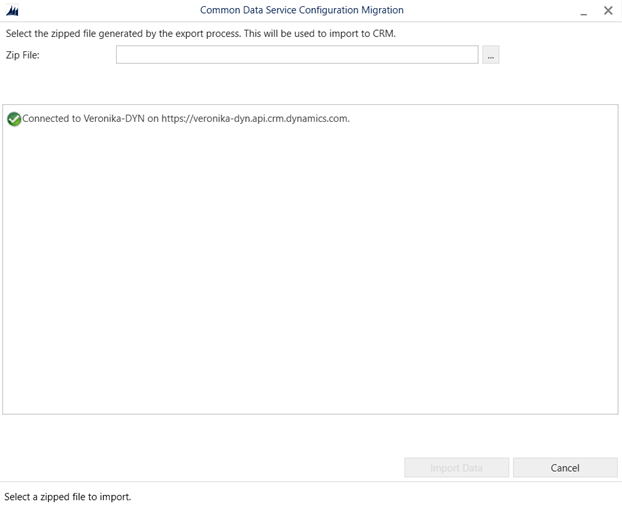
1. Download a sample dataset from the Fundraising and Engagement GitHub repository. There is ‘FundraisingAndEngagement.Sample.Data.zip’ file in the ‘Samples’folder.
2. Make sure a correctly configured configuration record exists in the application prior to data import, as well as the association with the user account.
3. Deploy the sample data set using the Data Migration utility. This tool is part of the Power Apps developer tools. It can be downloaded from [developer tools and resources](https://docs.microsoft.com/en-us/powerapps/developer/data-platform/developer-tools).
4. Run Data Migration utility.



1. Select the ‘Import Data’ option.
2. Check the ‘Display list of available organizations’ option and log in.



1. Select your environment from the list of available organizations and click Login.
2. After successful Login you will be prompt to select sample data file to import



1. Locate the downloaded FundraisingAndEngagement.Sample.Data.zip and click Import Data.
2. Wait until the tool finishes the import.
3. Once the tool finishes the import, check if the data is in the Azure database.

# Updating to a newer version

New Fundraising and Engagement releases bring new functionality, important bug fixes, and critical security, privacy, and accessibility enhancements.

The health and stability of your Fundraising and Engagement product is our top priority.

As new security, privacy, and accessibility enhancements are developed for Fundraising and Engagement, organization must take action to upgrade their environment. Failure to take the actions detailed below could result in an avoidable security risk.

## Prerequisites

Please visit [Fundraising and Engagement Releases](https://github.com/microsoft/fundraising-and-engagement/releases) on GitHub before upgrading your environment to the latest release version. While this documentation covers the update process in general, GitHub releases contain additional, specific version update instructions if such instructions exist. When updating to the most recent release from an older release, it may be necessary to execute all update actions for each intermediate release in between.

## Updating the Fundraising and Engagement Dynamics solution through AppSource

To update to the newest version, follow these steps:

1. Navigate to your environment in the [Power Platform admin center](https://admin.powerplatform.microsoft.com/environments) and open the environment hosting Fundraising and Engagement.
2. Select Resources > Dynamics 365 apps.

Graphical user interface, application

Description automatically generated

1. If there is a newer version to update to, you should see ‘Update available’ next to Fundraising and Engagement.

Graphical user interface, application

Description automatically generated

1. Select the available update, and confirm installation with the ‘Update’ button.
2. Check [Fundraising and Engagement Releases](https://github.com/microsoft/fundraising-and-engagement/releases) on GitHub for the release you updated to and follow any instructions relevant for AppSource installation.

## Updating Fundraising and Engagement Azure services

Not every release requires updates to Fundraising and Engagement Azure services. Some releases only require updates to the Dynamics solution through AppSource, as outlined above. Updates to Fundraising and Engagement Azure services is required whenever the most recent release (or intermediate releases since your last update of Azure services) on the Fundraising and Engagement Releases page list changes in Azure services or mentions required update actions for Azure.

The upgrade procedure has several steps that are similar to the installation instructions above, and includes required redeployment of the ARM template, redeployment of Azure services, and application of database migrations.

### Apply the Latest ARM Template

1. Follow the instructions in the section titled Deploying the ARM Template to apply the latest version of the ARM template via [https://portal.azure.com](https://portal.azure.com/). Follow the instructions in the ARM Template Deployment Guide up to the step where you are presented with the screen where template parameters need to be entered .

Graphical user interface, application, email

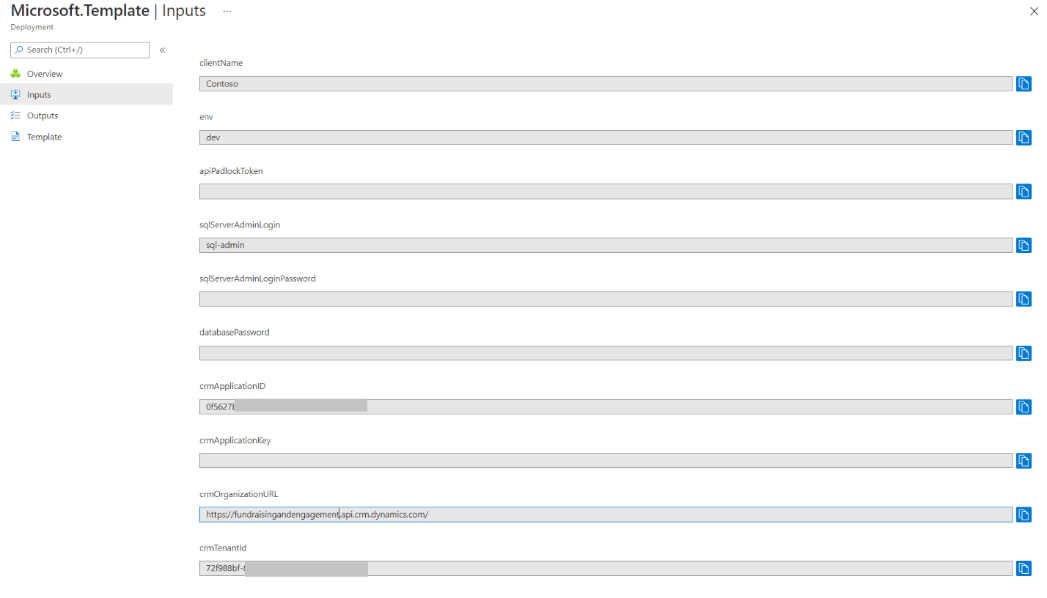
Description automatically generated

1. Before applying the template, it is important to enter exactly the same values as they were entered during the previous deployment. You can retrieve the values that were entered by going to the resource group you are deploying to in Azure Portal, and selecting Deployments.

Graphical user interface, text, application

Description automatically generated

Select your last successful deployment and go to Inputs.



1. Sensitive values such as passwords are not visible in the Inputs area of a previous section:
2. **API Padlock Token**: API Padlock Token **is deprecated** since version 1.2.3.0. The value can remain empty.
3. **Sql Server Admin Password**: If you don’t have the original admin password, the recommended procedure is to reset it to a new one.  
     
   Go to the resource group where the Fundraising and Engagement Azure services are deployed in [Azure Portal](https://portal.azure.com/). Open details of the SQL server there – its name will be in the format <name>-sqlserver-<env>, e.g. contoso-sqlserver-prod.  
     
   Open the SQL server detail, click the ‘Reset password ‘ button, enter a new strong password, and save it**.**

Graphical user interface, application, Word

Description automatically generated

Use the new password as the value for the SQL Server Admin Password parameter of the ARM template.

1. **Database Password**: If you don’t have the original database password, it can be retrieved from Azure Key Vault. Go to the resource group where the Fundraising and Engagement Azure services are deployed in [Azure Portal](https://portal.azure.com/). Open details of the Key vault – its name will be in the format <name>-vault-<env>, e.g. contoso-vault-prod.  
     
   Grant yourself access to read secrets in the vault by following the instructions in [Assign a Key Vault access policy](https://docs.microsoft.com/en-us/azure/key-vault/general/assign-access-policy-portal), and choosing ‘Secret Management ‘ as the template and your user account as the principal.

Graphical user interface, text, application, email

Description automatically generated

After adding the policy, confirm with the Save button. Now you can navigate to Secrets in the vault menu and open the ‘ConnectionSecrets—PaymentContextPassword’ secret to reveal the database password value there.

Graphical user interface, application

Description automatically generated

Use the secret value as input for the Database Password parameter of the ARM template. **Remove the Key Vault access policy you created afterward.**

1. **App Registration Client Secret:** If you don’t have the original client secret, the recommended procedure is to generate a new one.  
     
   In [Azure Portal](https://portal.azure.com/), navigate to Azure Active Directory > App registrations and open the app registration used for Fundraising and Engagement. Locate the correct app registration by its ID – the value of App Registration Client Id from the ARM template input. Under the app registration, navigate to Certificates & secrets and create a new secret (see also section Azure Active Directory Application Registration in this guide).

Graphical user interface, text, application

Description automatically generated

Use the value of the newly generated secret as an input for the App Registration Client Secret parameter of the ARM template. **It is strongly recommended that you delete the previous secret** (with the trash bin icon next to the secret).

1. Once all inputs are filled in, click ‘Create’ to start the deployment.

Graphical user interface, application

Description automatically generated

1. After a successful deployment of the template, the Configuration record needs to be updated. Navigate to the Configuration area > Configurations. For all Configuration records, **clear the values of Azure Web API URL and API Padlock Token** fields and save the change.

Graphical user interface, application

Description automatically generated

1. The Background Services Configurations section in the Configuration record contains security keys for triggering Azure functions from the Dynamics part of Fundraising and Engagement. Under normal circumstances, the update should not change the keys and no action is needed. However, in some cases (e.g., if the function was removed on the Azure side manually), the keys may change after the application of the ARM template. In such case, follow the instructions from section Configuration Record Prerequisites to set the correct values again.

### Redeploy Azure Services and Apply Database Migrations

This step is identical to the Deploying the Fundraising and Engagement components to Azure in this guide for deploying from scratch, including both the deployment of services to Azure (Deploy to Azure) and applying database migrations (Populate the Azure SQL Database Schema).

# Troubleshooting

## Failure Occurs During ARM Template Deploy

The ARM template is not capable of rolling back components. This means if the ARM template fails as a result of the prerequisites not being met, the user implementing the template must manually remove the created components prior to reattempting the deployment.

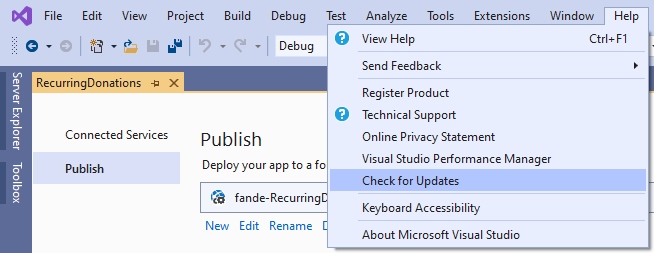
## Incorrect Parameter Value Entered when Updating to a New Version of the ARM Template

When deploying a new version of the ARM template during the update procedure, it is necessary to enter values exactly according to instructions. The list below outlines what problems can be caused by incorrect values and how to address them:

* **Client Name** and **Env**: Entering Client Name or Env value different from the initial installation can result in the creation of a second set of Azure resources (e.g., two SQL Servers instead of one, with the associated cost) in the target resource group. If this happens, the second set of resources needs to be deleted manually and the ARM template redeployed again.
* **Sql Server Admin Password or Login**: Entering a value different from the actual admin password can result in failure during deployment (e.g., with error message ‘Invalid value given for parameter Login ‘). Follow the instructions for resetting the admin password in the Updating Fundraising and Engagement Azure services section if this happens.  
    
  Deployment may also fail if the password entered is too weak with an error similar to   
  Password validation failed. The password does not meet policy requirements because it is not complex enough.   
    
  Retry with a stronger password (ideally a randomly generated string with letters, numbers and special characters with a length of more than 20 characters).
* **App Registration Client Id or Client Secret**: If any of these values are entered incorrectly, Azure services will fail to connect to Dynamics at runtime and errors will be visible in Azure services logs. This can be fixed by redeploying the ARM template with correct values. If you no longer have the original value of App Registration Client Secret, follow the steps in Updating Fundraising and Engagement Azure services to generate a new one.

## Updating Visual Studio 2019 Community Edition (or Higher)

Ensure your installation has the latest updates. This guide relies on the integrated publishing processes from Visual Studio to Azure components. As a result, they may need to be frequently updated for Azure components to be deployed. To update Visual Studio, select ‘Help’ then ‘Check for updates.’ Follow prompts to update the installation if required.



## Modifying the ARM Template

The ARM template has been configured and tested to deploy a single Azure environment per Dynamics 365 instance. For implementors wishing to change the deployment process or alter how the template delivers the application and configuration, they do so at their own risk. Support currently covers a single deployment per Dynamics 365 instance architecture.

## Naming and Unique Name Concepts

A failure during deployment of the template can occur due to the naming conventions used during the template setup. Resource names are externally addressable and must be unique. The template has been configured to generate unique names for the created applications but in the unlikely event that a name’s space for a component within the template is already in use, the deployment will fail. If this occurs, the implementor must modify the organization name to create a unique naming convention.

Please see details on best practices for [Azure naming conventions](https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/naming-and-tagging).

## Hitting Maximum Path Length Limit When Deploying Azure Components

If you are using Windows, the build or deploy of Azure components in Visual Studio may fail if path lengths of files created during the build reach the maximum Windows [path length limit](https://docs.microsoft.com/en-us/windows/win32/fileio/naming-a-file#maximum-path-length-limitation). To avoid this issue, please move the root of the fundraising-and-engagement git repo to a different location on filesystem with shorter path.

# Learn More About Azure and Extending Dynamics 365

This document assumes users provisioning Fundraising and Engagement Azure environments have a solid understanding of both the Azure components that will be created and configured as well as the Dynamics 365 environment which will utilize this functionality.

The following courses and certification paths are recommended prior to utilizing this guide:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Educate** | **Accredit** | **Discussion** |
| Azure Fundamentals | [Learning](https://docs.microsoft.com/en-us/learn/paths/azure-fundamentals/) Path | [Exam](https://docs.microsoft.com/en-us/learn/certifications/exams/ai-900) | [Community](https://powerusers.microsoft.com/t5/PowerApps-Community/ct-p/PowerApps1) |
| Dynamics 365 Fundamentals | [Learning](https://docs.microsoft.com/en-us/learn/paths/dynamics-365-fundamentals/) Path | [Exam](https://docs.microsoft.com/en-us/learn/certifications/exams/mb-900) | [Community](https://powerusers.microsoft.com/t5/Microsoft-Flow-Community/ct-p/FlowCommunity) |