M365 Provisioning Tool

Azure Configuration and

Deployment Process

Contents

[Version 2](#_Toc126048653)

[Overview 2](#_Toc126048654)

[Main Components 2](#_Toc126048655)

[Asset Summary 2](#_Toc126048656)

[Service Accounts (Used for Logic App API Connections) 2](#_Toc126048657)

[SharePoint Site Dependencies 3](#_Toc126048658)

[Resource Groups 3](#_Toc126048659)

[Logic Apps 3](#_Toc126048660)

[Managed Identities (System-Assigned) 3](#_Toc126048661)

[Azure Functions 3](#_Toc126048662)

[Storage Account 5](#_Toc126048663)

[Key Vault 5](#_Toc126048664)

[Deployment Process 5](#_Toc126048665)

[Subscription Setup 6](#_Toc126048666)

[Create Resource Group 6](#_Toc126048667)

[Create Azure Logic App 6](#_Toc126048668)

[Create Azure Function App 6](#_Toc126048669)

[Create Azure Key Vault 6](#_Toc126048670)

[Configure Resource Group 6](#_Toc126048671)

[Provide Access Manage Resource Group for Solution Administrators 6](#_Toc126048672)

[Configure Logic App 6](#_Toc126048673)

[Enable System Assigned Managed Identity (of Logic App) 6](#_Toc126048674)

[Configure Function App 7](#_Toc126048675)

[Enable System Assigned Managed Identity (of Function App) 7](#_Toc126048676)

[Enable Azure AD Authentication (To secure Azure Function Triggering) 7](#_Toc126048677)

[Grant Permissions for the Azure Logic App (Managed Identity) to Trigger Function 8](#_Toc126048678)

[Create a Self-Signed Certificate (To allow PowerShell to authenticate to SharePoint Online) 8](#_Toc126048679)

[Configure Azure AD Application Registration (Grant API Permissions) 8](#_Toc126048680)

[Specify the Required Modules to Load 9](#_Toc126048681)

[Create Functions 9](#_Toc126048682)

[Modify Function App to use PowerShell Code Runtime Version 7.0 9](#_Toc126048683)

[Set Function App Configurations 10](#_Toc126048684)

[Configure Key Vault 10](#_Toc126048685)

[Add Certificate to Key Vault 10](#_Toc126048686)

[Get Certificate URL to use in Function All Configuration 10](#_Toc126048687)

[Deploy Code to Function App 10](#_Toc126048688)

[Add Script to Function 11](#_Toc126048689)

[Deploy Code to Logic App 11](#_Toc126048690)

[Deploy Logic App Code from a Template 11](#_Toc126048691)

# Version

|  |  |  |
| --- | --- | --- |
| Version | Release Data | Notes |
| 1.0 | 31/01/2023 | Initial Version |
| 2.0 | 03/02/2023 | SA Gov Deployment Update   * Actual resource names * Minor process updates |

# References

|  |  |
| --- | --- |
| Name | Comment |
| SA Gov Azure Cloud Naming standard - draft v0.5 | Azure Resources Naming Standards |
| https://azure.microsoft.com/en-au/explore/global-infrastructure/geographies/#overview | Azure Geography Reference |

# Overview

This document details the provisioning and configuration of Azure components to support the M365 Provisioning process that has an accompanying Power Platform solution and SharePoint Site.

## Main Components

**Resource Group** is a logical container for all our assets, essentially just grouping them together as a solution.

**Logic App** offer automated monitoring of our SharePoint environment data, with logic to trigger provisioning script passing through associated data required. Only the logic app is authorised to call the Azure Function Apps, with authentication automatically managed.

**Function App** offers containers to store our script and runtime environment to run the scripts when required. This app runs under a managed identity which is authorised to call specific Graph and SharePoint APIs in the environment.

**Key Vault** offers a secure place to store the certificate so that our function app can also access it during runtime.

# Asset Summary

## Service Accounts (Used for Logic App API Connections)

|  |  |  |  |
| --- | --- | --- | --- |
| UPN | PW | Licences | Permissions |
| DPC-OCIO.PowerPlatformSPProvisioningTool  @sagov.onmicrosoft.com |  | Office 365 E3 Dev - SharePoint, Flow, Powerapps, Exchange mailbox | Site Collection Admin for Solution SharePoint Site |

## SharePoint Site Dependencies

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Used For | URL |
| Requests | List | Request Config | (Production Environment)  <https://sagov.sharepoint.com/sites/DPC-SAG_SPOProvisioningTool/Lists/requests> |
| Template\_exports | Library | Template Export Files | (Production Environment)  <https://sagov.sharepoint.com/sites/DPC-SAG_SPOProvisioningTool/template_exports> |

## Subscription

|  |  |
| --- | --- |
| Name | Notes |
| SAG XG DPC-ICT Tenancy Management Functions | OCIO Managed |

## Resource Groups

|  |  |
| --- | --- |
| Name | Region |
| SAGSPOProvisioningTool | Australia-East |

## Logic Apps

|  |  |
| --- | --- |
| Name | Settings |
| sag-m365p-logic-ause-001 | Basics   * Resource Group: As created above * Logic App Name: As to left * Publish: Workflow * Region: Australia-East * Enable Log Analytics: No * Plan: Consumption * Zone Redundancy: Disabled   API Connections   * SharePoint: Connection using Service Account * Function App: Connection using System Managed Identity of Logic App |

## Managed Identities (System-Assigned)

|  |  |  |
| --- | --- | --- |
| Name | Object ID | Permissions |
| Logic App - sag-m365p-logic-ause-001 | 88b9f024-359e-4d2c-bf0b-5ebb0a8f7b38 | Azure Function - Reader |
| Azure Function - sag-m365p-func-ause-001 | 597bc4ea-570f-4d6b-ad7f-f588dd7330ec | App ID API Permissions |

## Azure Functions

|  |  |
| --- | --- |
| Name | Settings |
| sag-m365p-func-ause-001 | Basics   * Resources Group: As created above * Publish: Code * Runtime: PowerShell Core * Version: 7.2 * Region: Australia Southeast * Operating System: Windows * Plan: Consumption   Hosting   * Storage Account: Create new   Networking   * Default   Monitoring   * Enable Application Insights: No   ---  Function (1)   * Development Environment: Develop in Portal * Template: HTTP Trigger * Name: M365ProvisionGroup * Authorisation: Anonymous (Function is secured via other means)   Function (2)   * Development Environment: Develop in Portal * Template: HTTP Trigger * Name: M365ProvisionTeams * Authorisation: Anonymous (Function is secured via other means)   Function (3)   * Development Environment: Develop in Portal * Template: HTTP Trigger * Name: M365ProvisionSharePoint * Authorisation: Anonymous (Function is secured via other means)   ---  Authentication   * Identity Provider: Microsoft * Default settings * Unauthenticated Requests: HTTP 401 Unauthorised * App (Client) ID - 0e3267eb-50ea-482e-86d2-72bee5cf927e * API Permissions (Application)   + SharePoint – Sites.FullControl.All   + Graph – Group.ReadWrite.All   + Graph – User.Read.All   ---  Configuration / Application Settings  Name: ClientID  Value: Azure AD App/client id (App (Client) ID)  Name: Certificate  Value: @Microsoft.KeyVault(SecretUri=<CertificateURI>)  Name: WEBSITE\_LOAD\_USER\_PROFILE  Value: 1 |

## Storage Account

|  |
| --- |
| Name |
| sagm365pstause001 |

## Key Vault

|  |  |
| --- | --- |
| Name | Settings |
| sag-m365p-kv-ause-001 | Basics   * Subscription: As Configured * Resource Group: As Configured * Key Vault Name: As Standard * Region: Australia-Southeast * Pricing Tier: Standard * Purge Protection: Enabled, 90 days   Access Policy (Create)   * Permissions: Secret > Get * Principle: Function App Managed Identity ID Guid |

# Deployment Process

The Following tasks are performed in the Azure Portal.

<https://portal.azure.com/>

Global Azure Administrator Rights Required For

* Initial Subscription & Resource Group setup
  + Permissions to managed and create components with this Resource Group can then be assigned
* API Permissions for Azure App Registration
  + Performing the Grant Admin Content App

## Subscription Setup

* This solution requires an active solution
* It can utilise an existing shared or new subscription just for this solution assets

## Create Resource Group

1. Create a new Resource Group
   1. Using settings as per Asset List

## Create Azure Logic App

1. Create a new Logic App
   1. Using settings as per Asset List

## Create Azure Function App

1. Create a new Function App
   1. Using setting as in Asset List

## Create Azure Key Vault

1. Create a new Key Vault
   1. Using settings as in Asset List

## Configure Resource Group

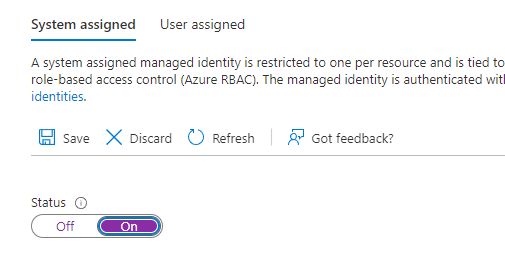
### Provide Access Manage Resource Group for Solution Administrators

1. Open Resource Group
2. Open Access Control (IAM) blade
3. Click Add > Add Role Assignment
4. Select Contributor Role, Next
5. For Members, Assign User, Select Member
6. Add, Review + Assign

## Configure Logic App

### Enable System Assigned Managed Identity (of Logic App)

1. Enabled System assigned Managed Identity in Identity tab
   1. Save Setting once turned on



\*\* This is effectively an identify which we give access to call the Function App

## Configure Function App

### Enable System Assigned Managed Identity (of Function App)

1. Open Function App
2. Open Identity blade
3. Enable System Assigned Managed Identity (Status = On)
4. Save, Yes
5. Make note of Object ID

### Enable Azure AD Authentication (To secure Azure Function Triggering)

1. Open Function App
2. Open Authentication blade
3. Click Add Identity Provider
4. Select Microsoft
5. Choose to return status code 401 Unauthorized for unauthenticated requests
6. Save
7. Once Created, click Edit icon in Identity Provider section
8. Remove “v2.0” from the end of Issuer URL
9. In the Allowed Token Audiences list, copy the GUID visible on the first row (after api://) and add the GUID to its own row immediately below the API URL

Graphical user interface, text, application, email

Description automatically generated

### Grant Permissions for the Azure Logic App (Managed Identity) to Trigger Function

1. Open Function App
2. Open Access Control (IAM) blade
3. Click Add role Assignment to begin create a new Role Assignment
4. Select Reader as the role
5. Go to the Members Tab
6. Choose to assign the role to a Managed Identity
7. Click on Select Members
   1. Ensure correct Azure Subscription is selected, select Logic App from Managed Identity dropdown
8. Select the Logic App created
9. Proceed with Review + Assign to finish.

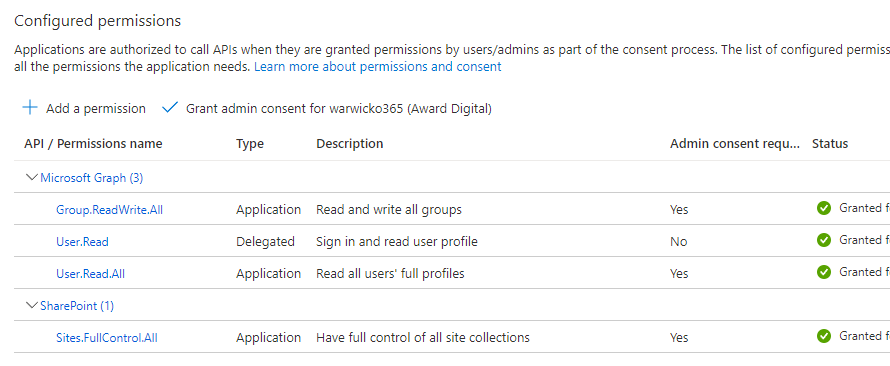
### Create a Self-Signed Certificate (To allow PowerShell to authenticate to SharePoint Online)

* Requires PnP PowerShell Modules for script cmdlets - <https://pnp.github.io/powershell/index.html>

1. Open PowerShell script Create-SelfSignedCertificate.ps1
2. Update commonName and password variables as required
3. Run all lines in script
4. Certificate files will be generated into the current directory location

### Configure Azure AD Application Registration (Grant API Permissions)

1. Open Function App
2. Open Authentication blade
3. In the Identity Provider list, copy the App(Client) ID guid to be used
4. Click the Identity Provider link in the first column to open Azure AD application registration
5. In the application registration settings, first go to the API Permissions blade
   1. Click on Add a permission
   2. Select SharePoint
   3. Select Application permissions
   4. Select Site.FullControl.All
   5. Click on Add Permissions
   6. Add SharePoint and Graph Permissions as in Asset List Above
   7. Click Grand admin consent for organisations
      1. \*\* This step required Global Tenant Admin permissions to enable – Request this to be performed if required



1. Go to Certificated & Secrets blade
   1. Open Certificates tab
   2. Click on Upload certificate
   3. Select the .cer file we generated with the PowerShell script

Graphical user interface, text

Description automatically generated

### Specify the Required Modules to Load

1. Open Function App created
2. Open App files blade
3. Access host.json
   1. Open associated script provided and paste into code window (replacing)
4. Access profile.ps1
   1. Open associated script provided and paste into code window (replacing)
5. Access requirements.psd1
   1. Open associated script provided and paste into code window (replacing)

### Create Functions

1. Open Function App created
2. Open Functions blade
3. Create 3 new functions
   1. Using settings as per Asset List

### Modify Function App to use PowerShell Code Runtime Version 7.0

\*\* Due to incompatibility of PnP.PowerShell and PowerShell Core 7.2, a downgrade to 7.0 is required - <https://github.com/pnp/powershell/issues/2136>

1. Log into Azure PowerShell Prompt (Either using Cloud Shell, or VS Code with Azure Account extension
2. Open / Run Commands in script Set-AzureFunctionRuntime7.0.ps1
   1. Modify variable to match current environment
      1. SubscriptionID
      2. ResourceGroupName
      3. AppName (Function App)
      4. TenantID
   2. Step through command and confirm downgrade using commands at the verify step

### Set Function App Configurations

\*\* Function App Configuration has depenancies on Key Vault configuration – Perform the certificate upload to get Certificate URI to be used in this section

1. Open Function App
2. Open Configuration blade
3. Add the following application settings as per Asset List
   1. ClientId
   2. Certificate
   3. WEBSITE\_LOAD\_USER\_PROFILE
4. Confirm Key Value Certificate Access is shown in Source

Graphical user interface, text, application

Description automatically generated

## Configure Key Vault

### Add Certificate to Key Vault

1. Open Key Vault
2. Open Certificates blade
3. Click on Generate/Import button
4. Select Import
5. Give a descriptive name to the vault certificate (M365-Provisioning-Certificate)
6. Upload the .pfx certificate file generated earlier
7. Enter in the same password as what you used when generating the certificate

### Get Certificate URL to use in Function All Configuration

1. Click on the certificate in Key Vault
2. click on current Certificate Version
3. Copy the Certificate Identifier (URL)
   1. The version (GUID) isn’t required at the end of the URL, this way the latest version of the certificate will always be used

## Deploy Code to Function App

### Add Script to Function

1. For Each function
   1. Open Code + Test
   2. Accessing the run.ps1
   3. Open associated script provided and paste into code window
   4. Save Script

## Deploy Code to Logic App

### Deploy Logic App Code from a Template

1. Open Logic App
2. Click Edit
3. Select Blank Template
4. Click code view
5. Copy and Paste json code from **logicapp\_codeview\_export.json** template file
6. Update Logic App
   1. Delete SharePoint API Connections and re-add using Service Account with access to require SharePoint Site
   2. Fix All Connections in Logic App
   3. Modify All Azure Function Connections,
      1. Reselect required Function App a function
      2. Setup Managed Identity and Corresponding Audience GUID from the Function App Managed Identity