KPMG

cutting through complexity

Saviynt Azure Integration & O365 License Provisioning & Prerequisites

<Client>
KPMG Build

July 2022
Version: 1.0

Table of Contents

1 Document Control

1.1 Document Review & Feedback

An updated version of this document has been created and will be reviewed by the team members and stakeholders listed below. The feedback obtained from their review will be incorporated.

Manjunath Madiraju	07/20/2022	Architect	07/20/2022
Architect	Date	<Title>	Date
Security Architect	Date	<Title>	Date
<Title>	Date	<Title>	Date

1.2 Document Acceptance

Representative Approvers

By signing this document, you confirm that you have read, reviewed, and approved the contents of this deliverable.

AGREED TO AND ACCEPTED BY:	**AGREED TO AND ACCEPTED BY:**
Client Project Management	Vendor Project Management
By: _____	By: _____
Name: _____	Name: _____
Title: _____	Title: _____
Date: _____	Date: _____

1.3 Modification History

Change Date	Author	Version Modified	Description of Changes
July 20, 2022	Manjunath Madiraju	1.0	Initial Creation

2 Document Purpose

Enabling access to technology resources in a secure and efficient manner is at the core of a strong cyber security program. An organization must provide its workforce (employees, contractors and business partners) with the required access to securely enable business operations and collaboration. The purpose of this document is to provide Azure and Powershell Integration Prerequisites for <Client>.

3 Introduction

The **Azure AD** connector is used for importing (reconciliation) Azure AD users and the **REST** connector is used for provisioning and de-provisioning license to Azure AD users which automatically enables the remote mailbox on Office 365. The Microsoft Graph API is used for integration between **EIC** and **Azure AD**.

You can use EIC to provision and de-provision licenses of Azure AD users on Office 365 using the REST connector. Azure AD is the repository of all users and is connected to Office 365, which is a cloud mailbox service. To enable the users to use the Office 365 as a mail server, you just need to provision license of those users from EIC to Azure AD. This automatically assigns the licenses to Office 365 users. The Office 365 mailbox uses Azure AD to fetch the users. When you create an account in Azure AD, the user is automatically created. However, the mailbox is not enabled on Exchange.

Supported Features:

The Azure AD connector supports the following features:

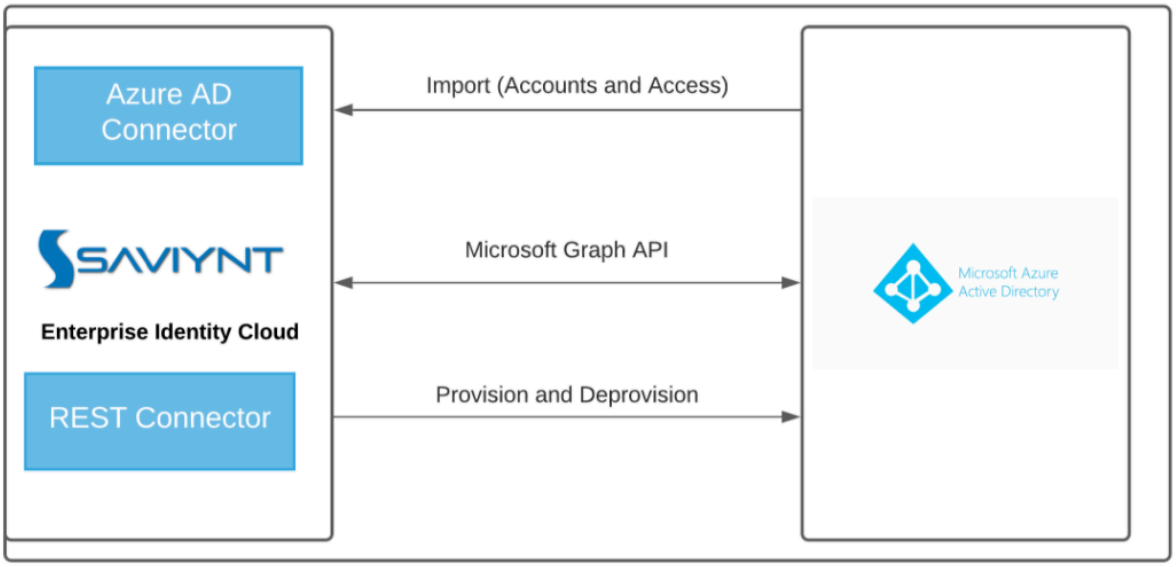
- Data import (reconciliation):
  - Ability to perform incremental imports for accounts and groups.
  - Ability to run parallel import of accounts and access.
  - Ability to import entitlements, accounts, permissions, groups, directory roles, applications, Service Principal, and subscriptions.
  - Ability to import the following using the APIs:
    - Entitlement2 Mapping for Subscription and SKU
    - Entitlement2_Privilege for Application Instance into EIC
  - Ability to import Microsoft Teams and Channels created under each team.
  - Ability to import Member and Guest permissions to understand the type of access provided over the Team.
    Four new entitlement types added are: Teams, Channels, Member Permissions, and Guest Permissions to support Microsoft Teams and Channels.
- Provisioning and deprovisioning:
  - Ability to create provisioning tasks for applications and specific entitlements. After the request is created and approved by the manager, you can create a trigger of the Provisioning Job (WSRETRYJOB), and provision the task for Azure AD.
  - Powerful provisioning and deprovisioning mechanism by creation of rules for automated access provisioning during Identity lifecycle events, such as new hire (birth-right access), promotion, re-hire, and termination.
  - Supports only provisioning default access for the application instance and provisioning all the service plans for the SKUs.
  - Ability to provision Members and Owners as Privileges for Group and Team Entitlement Types
  - Ability to handle Azure AD group management and REST based provisioning in one security system. (Available with Release v5.5.2 and later)
  - Ability to provision membership to distribution groups by extending the Win-PS connector capability.
- Saviynt Analytics provides a dashboard containing a pictorial graph of violations that happened. The detective analytics control is a powerful tool that provides the number of violations performed.

Based on your discretion, it allows you to accept a violation till a specific date or reject the violation.

- Provision to filter out the orphan accounts without any user in a Dashboard, which can be created by selecting **Admin > Analytics > Dashboard.**
- Enables security by using the OAuth client credential grant, which is also known as two-legged OAuth. It uses the identity of an application for authentication instead of user ID and password.
- Ability to create, update, and delete groups from the **ARS** module.
- Ability to add and delete accounts (owner and members) through the **ARS** module of **EIC**.
- Provides visibility on who has access to the **Subscription** entitlement type.
  For more information on visibility about privileges and provisioning on the **Subscription** entitlement type, see Azure Connector.
- From Release v2021.0 onwards, the connector supports **multithreaded** access import for the following objects:
  - Full import of groups, group memberships, and group owners.
  - Full or incremental import of Microsoft Teams and Channels.
- From Release v2021.0 onwards, the connector supports integration with multi-region Azure AD environments like Azure Germany and Azure AD for US Government instead of only Azure AD (global service) and Azure AD China environments.

4 Connector Architecture

The following architecture diagram illustrates the **Azure AD** Connector architecture and Saviynt's communication with **Azure AD**. The right-side depicts the directory services supported by **Azure AD**, and the left-side depicts **EIC**. The **Azure AD** connector is used for importing (reconciliation) of data and the **REST** connector is used for provisioning and deprovisioning of data from **EIC** to **Azure AD**. The Microsoft Graph API is used for integration between **EIC** and **Azure AD**.



5 Azure AD Prerequisites

The following section contains the steps required in Azure AD, in order for Saviynt to connect Azure AD and reconcile the Azure user accounts and groups into Saviynt.

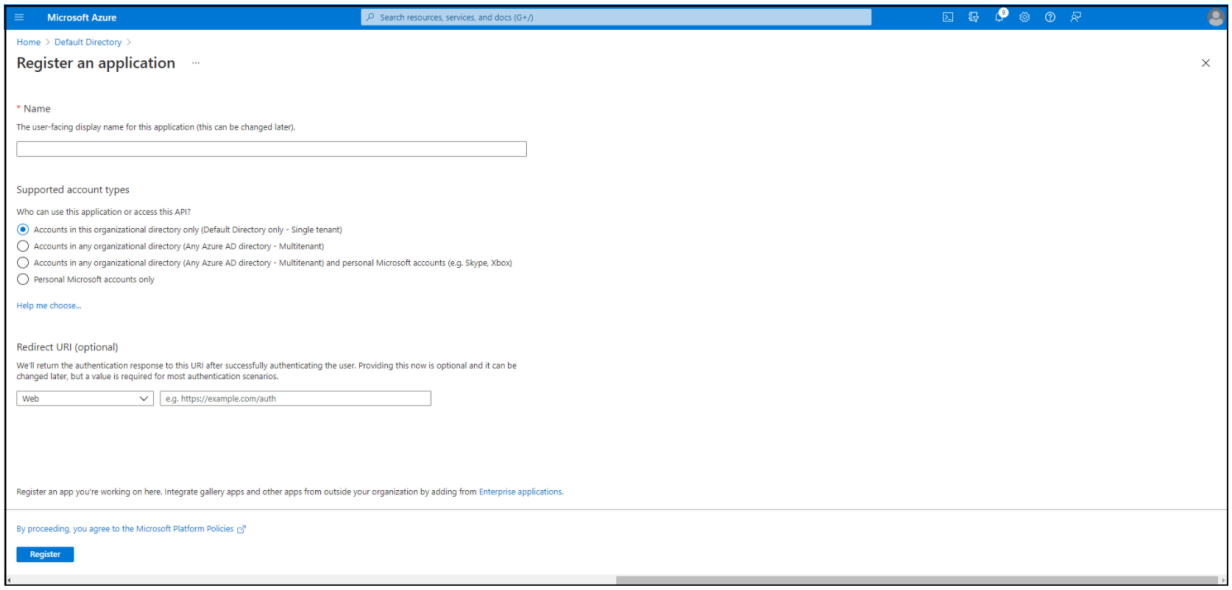
5.1 Register an Application in Azure AD

To register an application in Azure AD, perform the following steps:

Note:

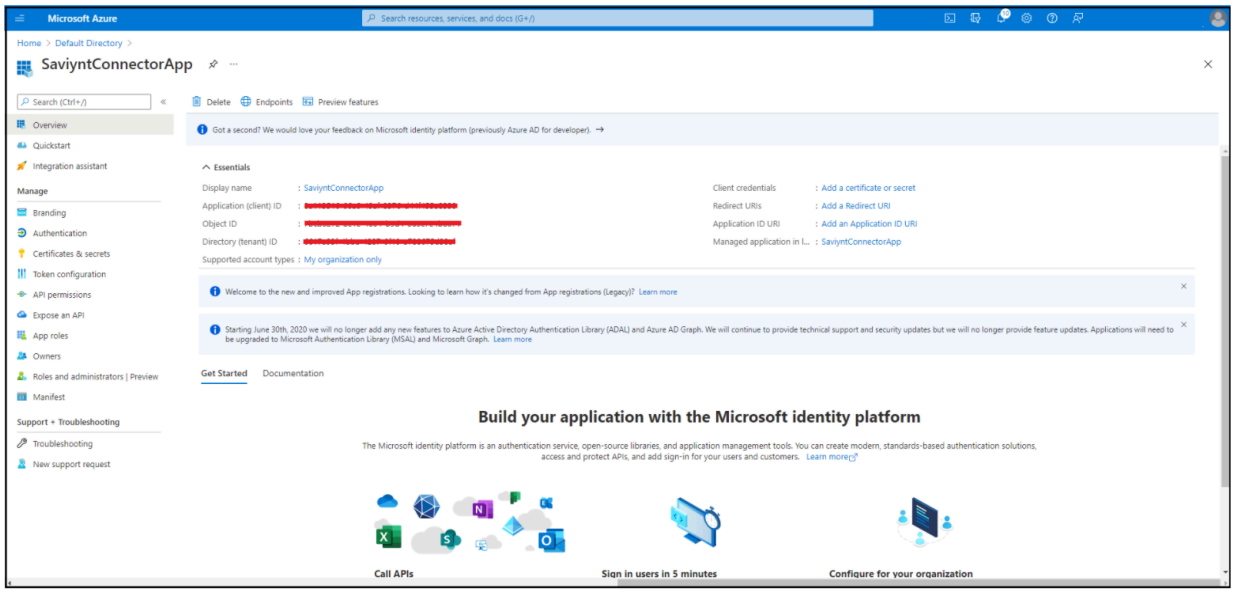
- Perform the steps in this section for the following connectors:
  - Azure AD Connector
  - Azure Connector
  - Office 365 Connector
- Azure Active Directory Graph is deprecated by Microsoft. Therefore, you cannot add Azure Active Directory Graph permissions for new applications registered in Azure AD. If you are creating an application in Azure AD for a release earlier than Release v2021.0, contact the Saviynt Support team by raising a Freshdesk ticket for guidance.

1. Log in to the Azure Portal using Azure Admin credentials to access the Azure AD associated with the tenant.
2. Select **Azure Active Directory** on the **Azure Home** page.
3. In the left pane, under **Manage**, select **App registrations**. The **App registrations** page is displayed.
4. Click **New registration**. The **Register an application** page is displayed.
5. Under **Manage**, select **App registrations** and then click **New registration**.

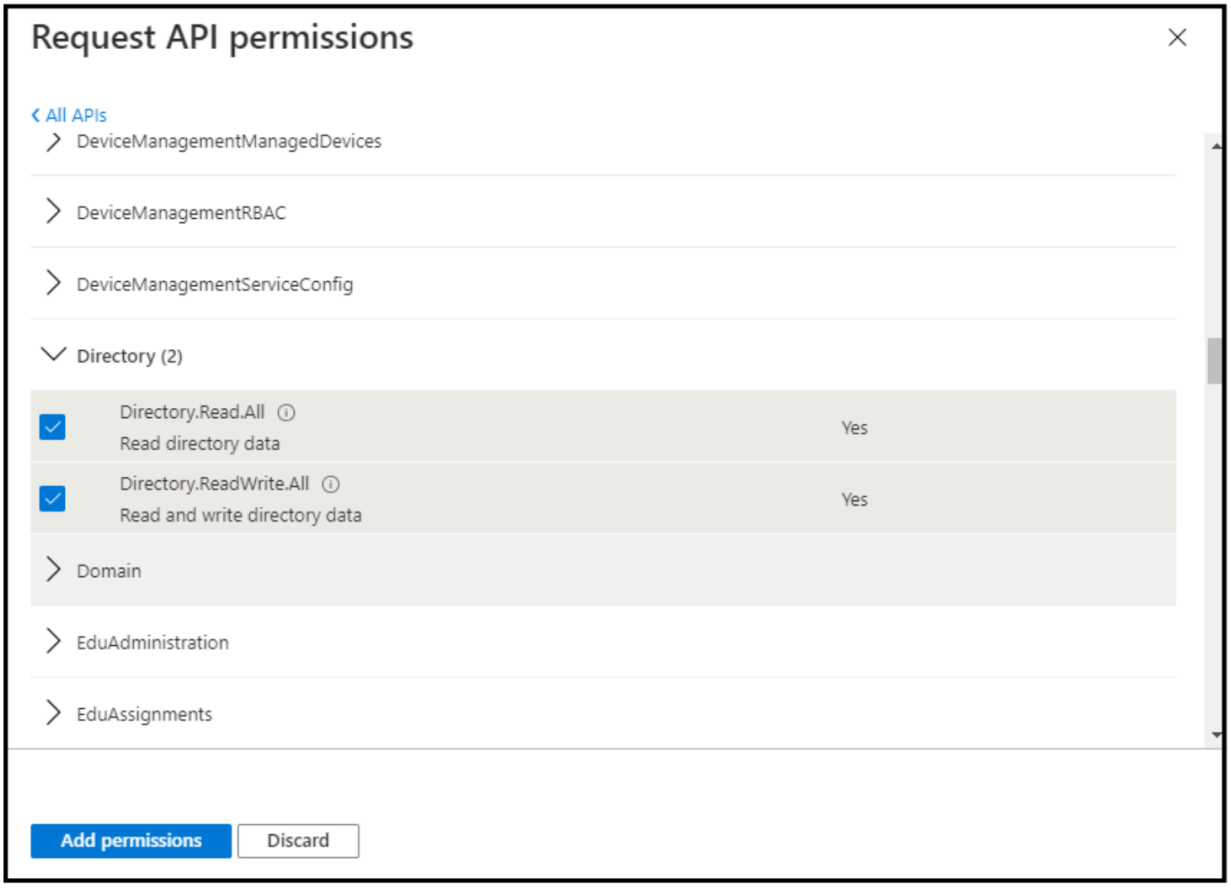


6. On the **Register an application** page, do the following:
   - **Name**: Specify the name for the application (For example, **SaviyntConnectorApp**).
   - **Supported account types**: Select **Accounts in this organizational directory only (Default Directory only - Single tenant)**. This is the default option and maps to Azure AD only single-tenant.
   - **Redirect URI (optional)**: Leave the field empty.

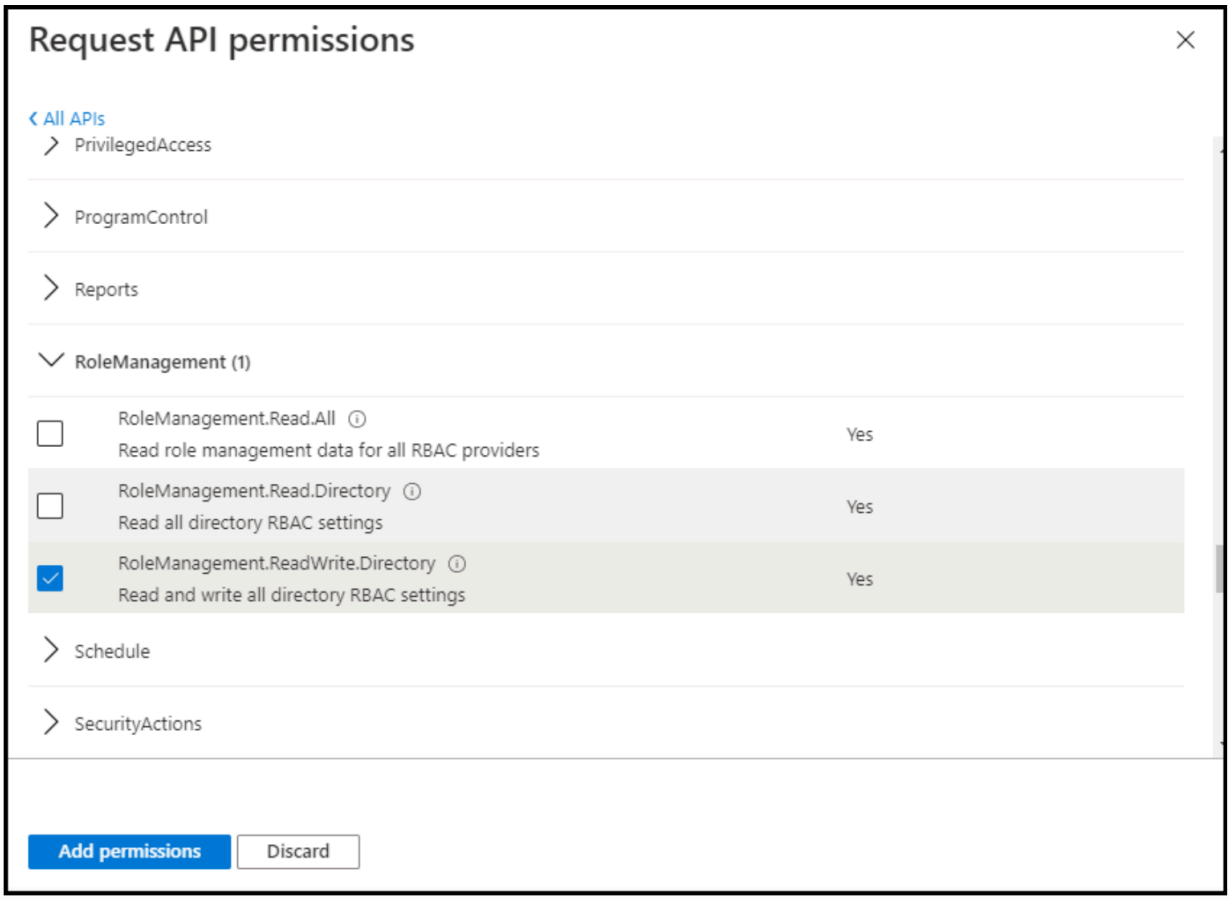
- Click **Register** to create your application in Azure AD. A page with details of the newly created application is displayed. The value displayed for the **Application (client) ID** uniquely identifies your application in the Microsoft identity platform.



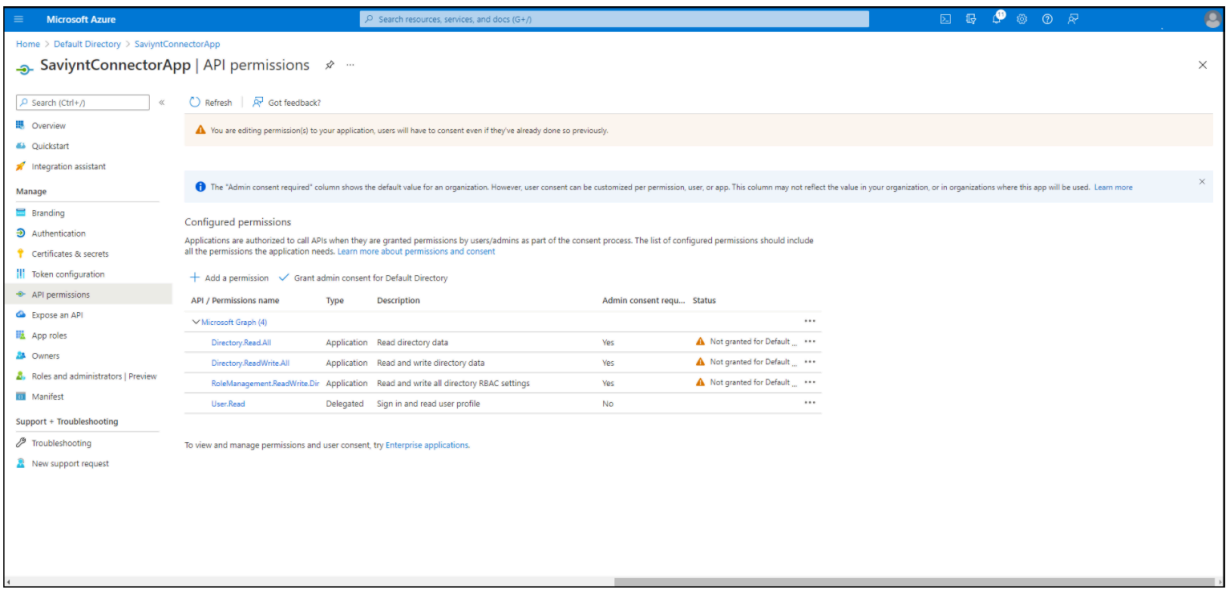
7. Configure API permissions for the application you created:
    - In the left pane, under **Manage**, click **API permissions**.
    - On the **API permissions** page, do the following:
        - Click **Add a permission**. The **Request API permissions** pane is displayed.
        - Select the **Microsoft APIs** tab, and then click **Microsoft Graph**.
        - Click **Application permissions**.
        - Scroll down, expand **Directory** and then select the following:
            - **Directory.Read.All**: This permission is required to perform reconciliation operations.
            - **Directory.ReadWrite.All**: This permission is required to perform provisioning operations.



- Scroll down, expand **RoleManagement**, and then select **RoleManagement.ReadWrite.Directory** for directory role provisioning and deprovisioning.

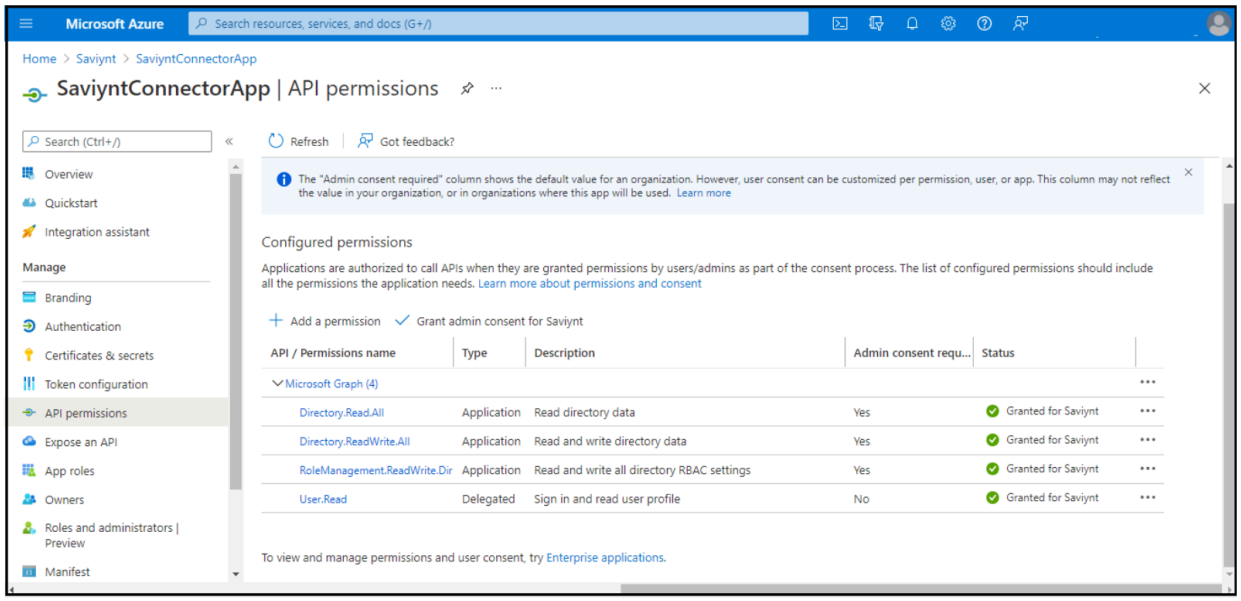


- Click **Add permissions**. The added permissions are displayed on the **API permissions** page.

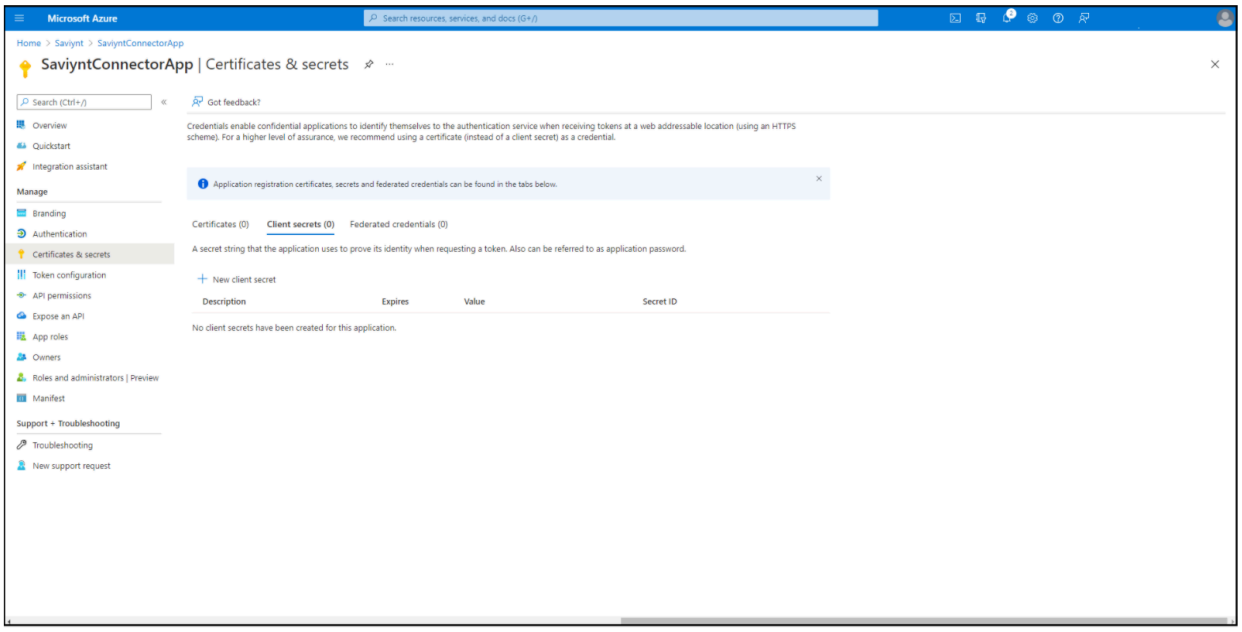


- Review the permissions added and then click **Grant admin consent for Default Directory**. A message is displayed asking for confirmation to grant admin consent. **Note**: If you are not currently logged in as a user with admin rights, the option to grant admin consent is disabled on the **API Permissions** page.

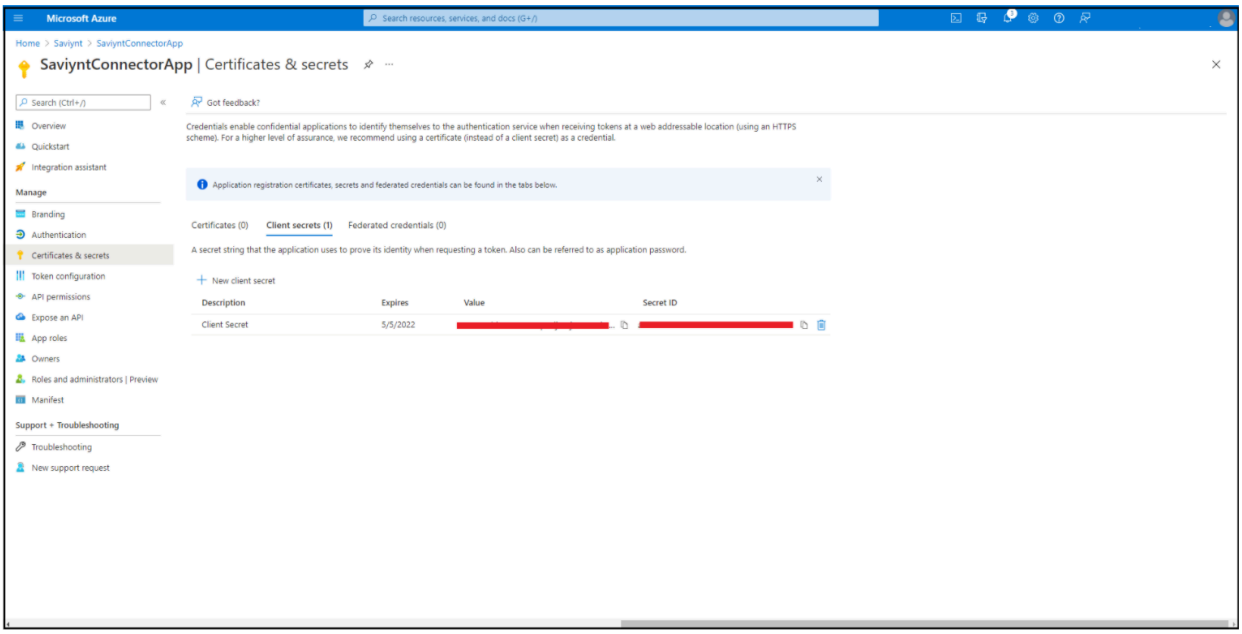
- Click **Yes** to grant consent for the requested permissions for all accounts in the Default Directory.



8. Create a client secret for the application that you registered:
   - In the left pane, under **Manage**, select **Certificates & secrets**.

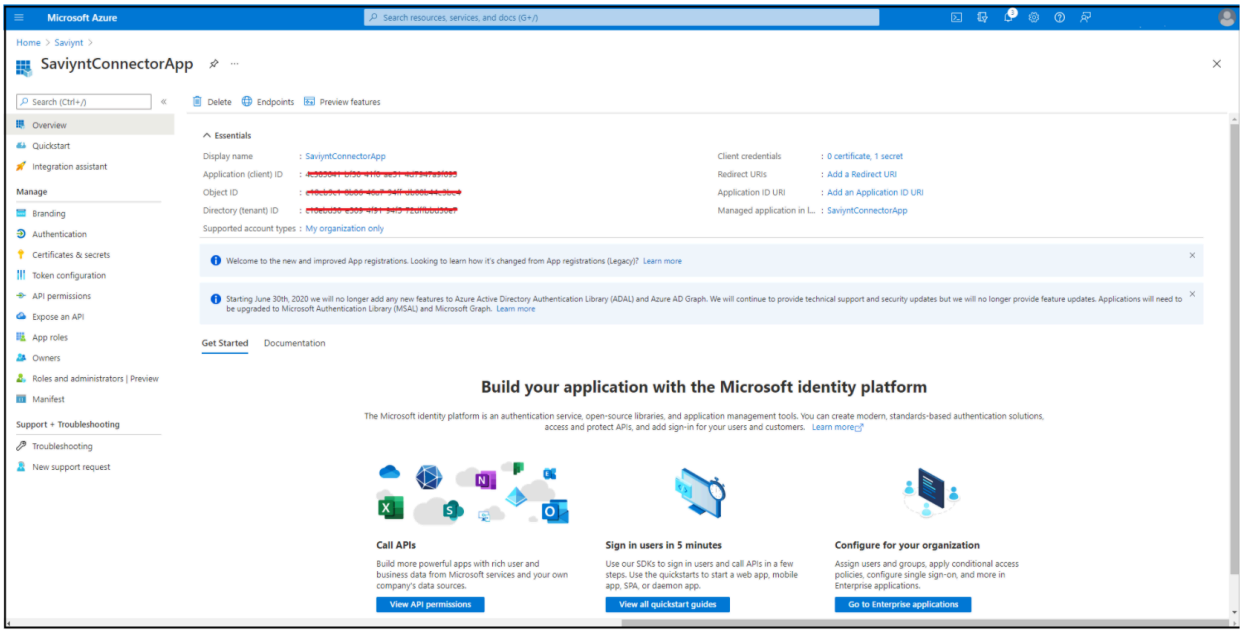


- On the **Certificates & secrets** page, do the following:
  - Click **New client secret**.
  - Add a description for your client secret.
  - Select a duration.
  - Click **Add**.
    The Secret key is generated and displayed in the Client secret section.



- Click the **Copy to clipboard** icon to record the secret's value for use in your client application code and share it with the Saviynt Support team.
  Note: This secret value is never displayed again after you leave this page.

On the left pane, go to **Overview** and get the **Client ID** and **Tenant ID**. You must share them with the Saviynt Support team.



5.2 Enabling an Azure AD Application to Deprovision Users

To enable Saviynt to perform user deprovision operation in Azure AD, perform the following steps:

Note: Perform the steps in this section for the Azure Connector.

1. Install the latest version of MSOnline PowerShell Module.
2. Connect to AzureAD PowerShell module and execute the following commands:

3. Connect-msolservice #Enter Admin credentials of the Azure portal
4. $webApp = Get-MsolServicePrincipal –AppPrincipalId “<ClientId of Azure AD Application>”

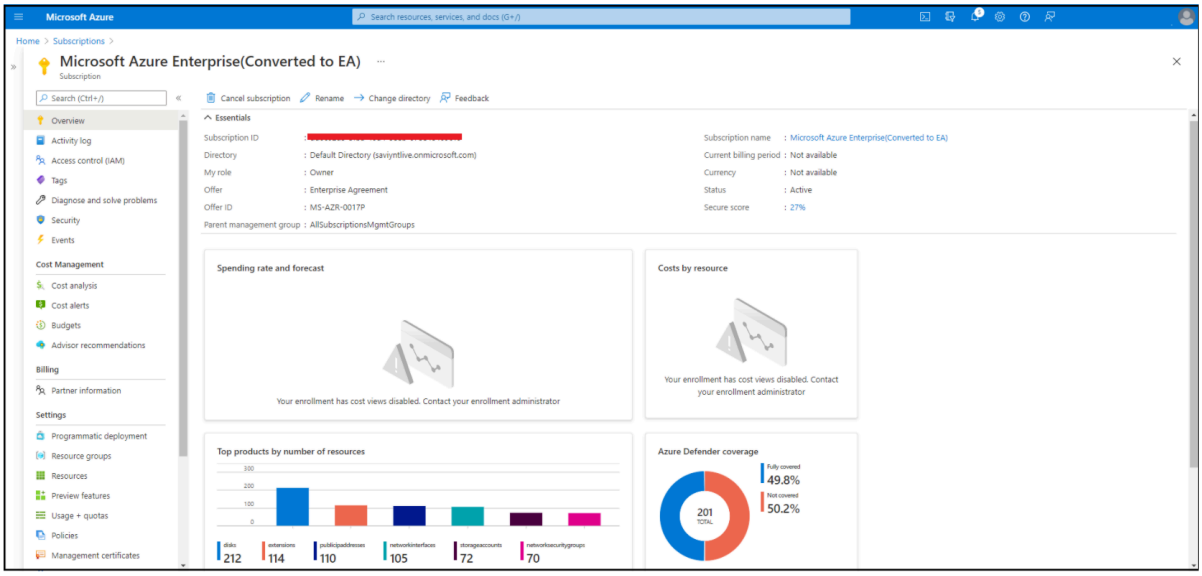
Add-MsolRoleMember -RoleName "Company Administrator" -RoleMemberType ServicePrincipal -RoleMemberObjectId $webApp.ObjectId

5.3 Granting User Access Administrator Role to Azure Subscription

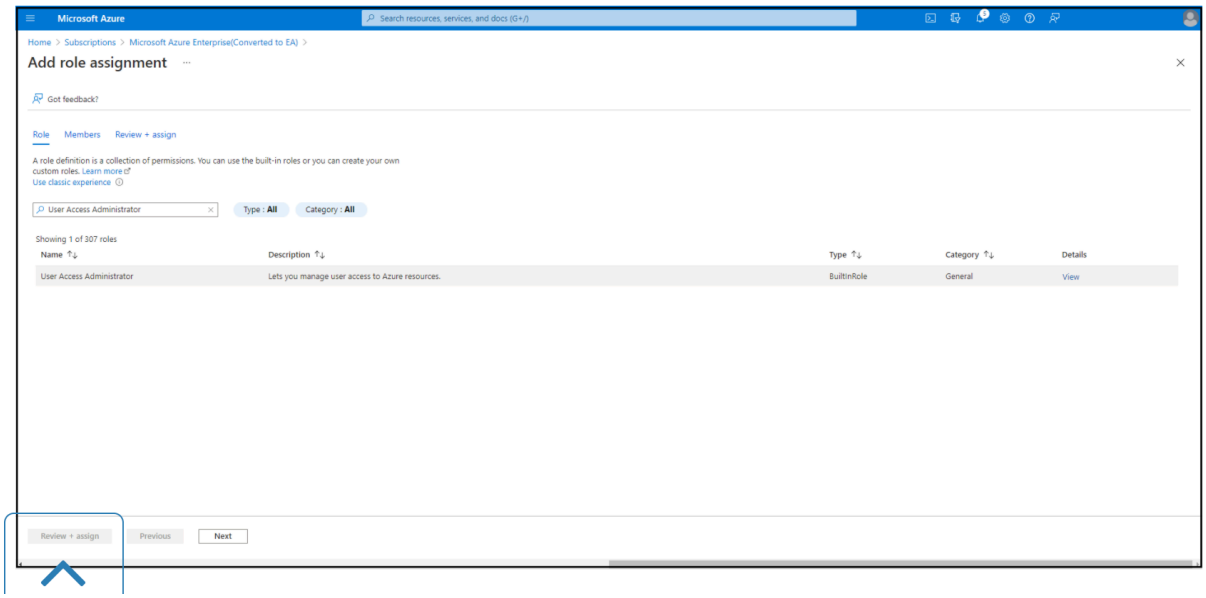
To grant user access to the subscription, perform the following steps:

Note: Perform the steps in this section for the Azure Connector.

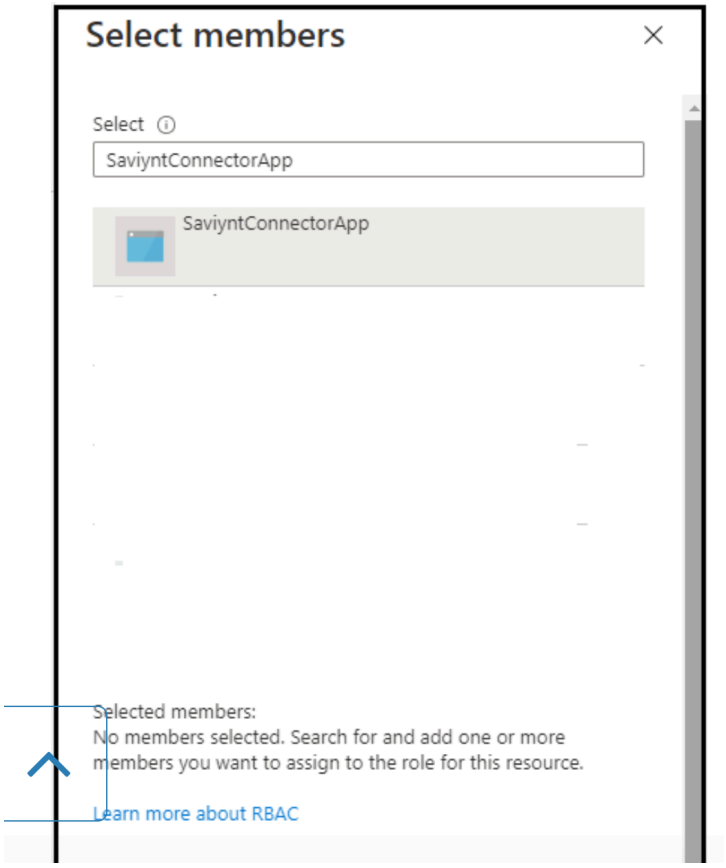
1. Log in to the Azure Portal using Azure Admin credentials.
2. Select **Subscriptions** on the **Azure Home** page. The **Subscriptions** page is displayed.
3. Click the subscription you want to use. The following shows an example subscription.



4. In the left pane, click **Access control (IAM)**.
5. Click **Add > Add role assignment**. The **Add role assignment** page is displayed.
6. Search for the **User Access Administrator** role.

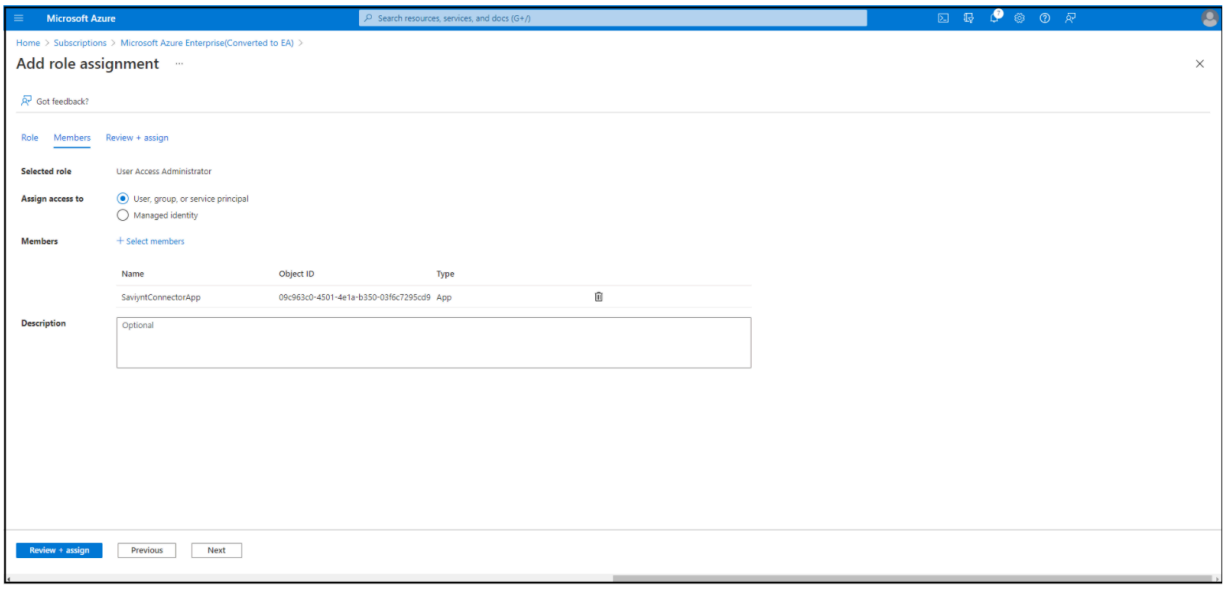


7. Select **User Access Administrator** in the **Name** column and click **Next**.
8. On the **Members** tab, do the following:
    - Select **User, group, or service principal**.
    - Click **Select members** and search for the application you created in Azure AD.



- Click **Select** to add the Azure AD application to the Members list.

- Specify the description for this role assignment.
- Click **Next**.



9. Click **Review + assign** to assign the role. After a few moments, the user is assigned the **User Access Administrator** role.
10. Go to **Overview** and get the **Subscription ID**. You must share them with the Saviynt Support team.

5.4 Assigning Permissions to the Subscription for Visibility and Governance

To assign permission for visibility and governance to the Subscription that you want to import into Saviynt, perform the following steps:

Note: Perform the steps in this section for the Azure AD Connector.

1. Log in to the Azure Portal.
2. Select **Subscriptions** on the **Azure Home** page. The **Subscriptions** page is displayed.
3. Click the subscription you want to import.
4. In the left pane, click **Access control (IAM)**.
5. Click **Add > Add role assignment**. The **Add role assignment** page is displayed.
6. Search for the **Reader** role.
7. Select **Reader** in the **Name** column and click **Next**.
8. On the **Members** tab, do the following:
    - Select **User, group, or service principal**.
    - Click **Select members** and search for the application you created in Azure AD.
    - Click **Select** to add the Azure AD application to the Members list.
    - Specify the description for this role assignment.
    - Click **Next**.

9. Click **Review + assign** to assign the role. After a few moments, the user is assigned the **User Access Administrator** role.
10. Repeat steps 3 through for all the subscriptions that need to be imported.

5.5 Data to be shared with Saviynt Team for Connection

The following Azure data to be shared with the Saviynt Team /POC:

Parameters	Description
CLIENT ID	Represents a unique identifier of the application within Azure AD tenant.
CLIENT SECRET	Represents secret access key of the application created in Azure Active Directory.
TENANT ID	Represents a unique identifier of a dedicated instance in Azure AD service that an organization receives and owns when it signs up for a Microsoft cloud service.
SUBSCRIPTION ID	Represents a unique identifier of Azure subscription which grants you access to Azure services and to the Azure Resource Management Portal.

6.0 O365 License provisioning

Until now, licenses could only be assigned at the individual user level, which can make large-scale management difficult. For example, to add or remove user licenses based on organizational changes, such as users joining or leaving the organization or a department, an administrator often must write a complex PowerShell script. This script makes individual calls to the cloud service.

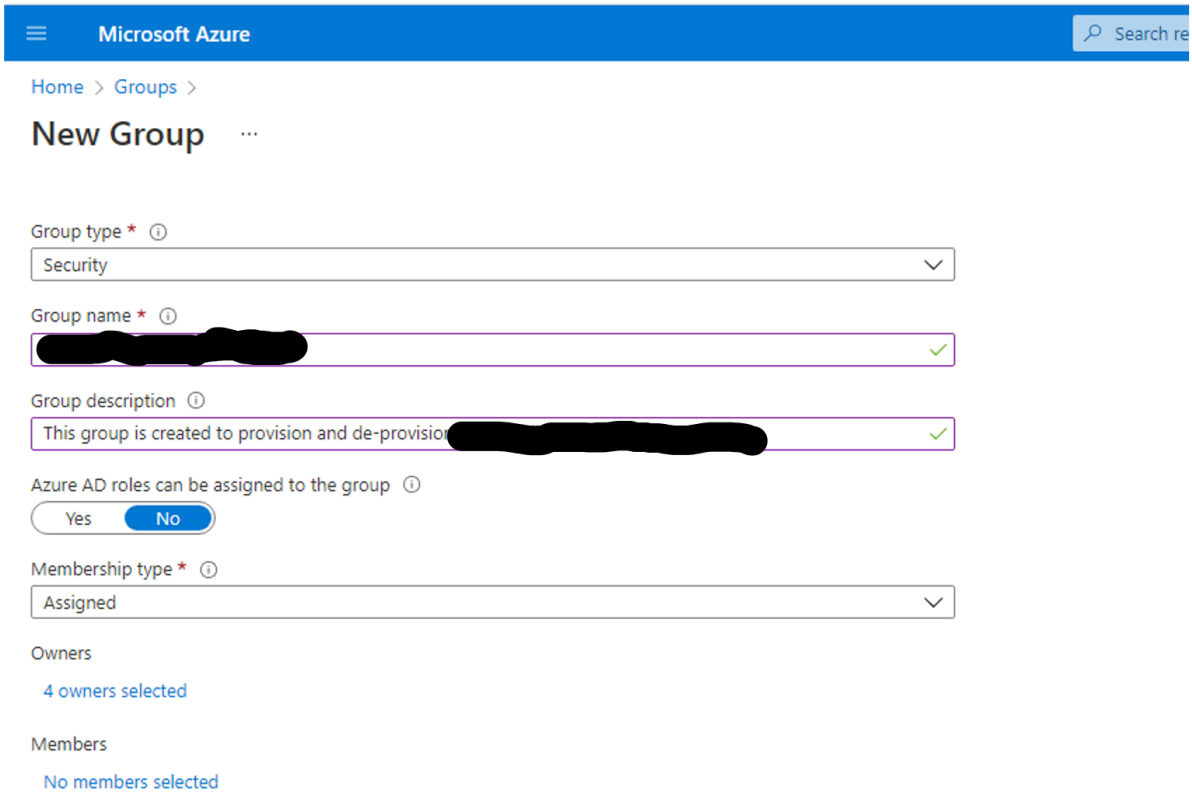
To address those challenges, Azure AD now includes group-based licensing. You can assign one or more product licenses to a group. Azure AD ensures that the licenses are assigned to all members of the group. Any new members who join the group are assigned the appropriate licenses. When they leave the group, those licenses are removed. This licensing management eliminates the need for automating license management via PowerShell to reflect changes in the organization and departmental structure on a per-user basis.

Azure AD groups should be created using below similar template.

- **AAD-O365License-<LicenseType>-<Scope>**

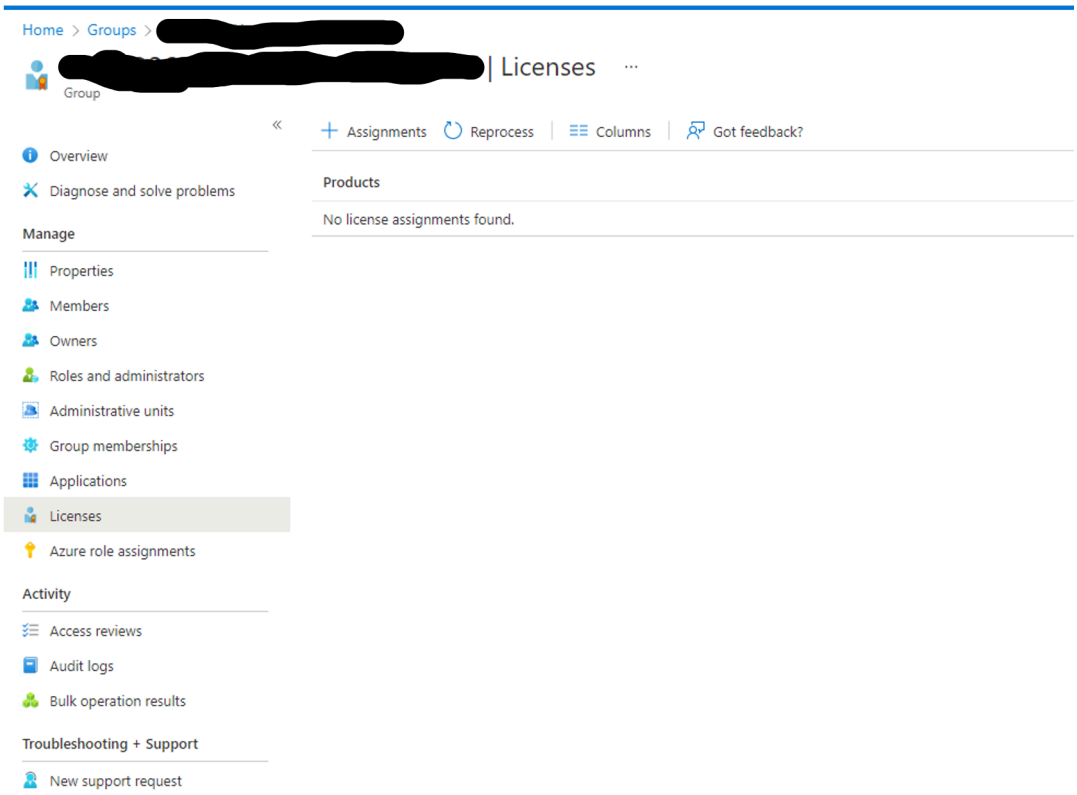
Group Name	Group Description	Group Type
AAD-O365License-E3-Location/Dept/EmpType/Other-param	This group is created to provision and de-provision O365 E3 license for Location/Dept/Other-param users.	Security
AAD-O365License-F3-Location/Dept/EmpType/Other-param	This group is created to provision and de-provision O365 F3 license for Location/Dept/Other-param users.	Security
AAD-O365License-P2-Location/Dept/EmpType/Other-param	This group is created to provision and de-provision O365 P2 license for users.	Security

Steps to create new AAD group:

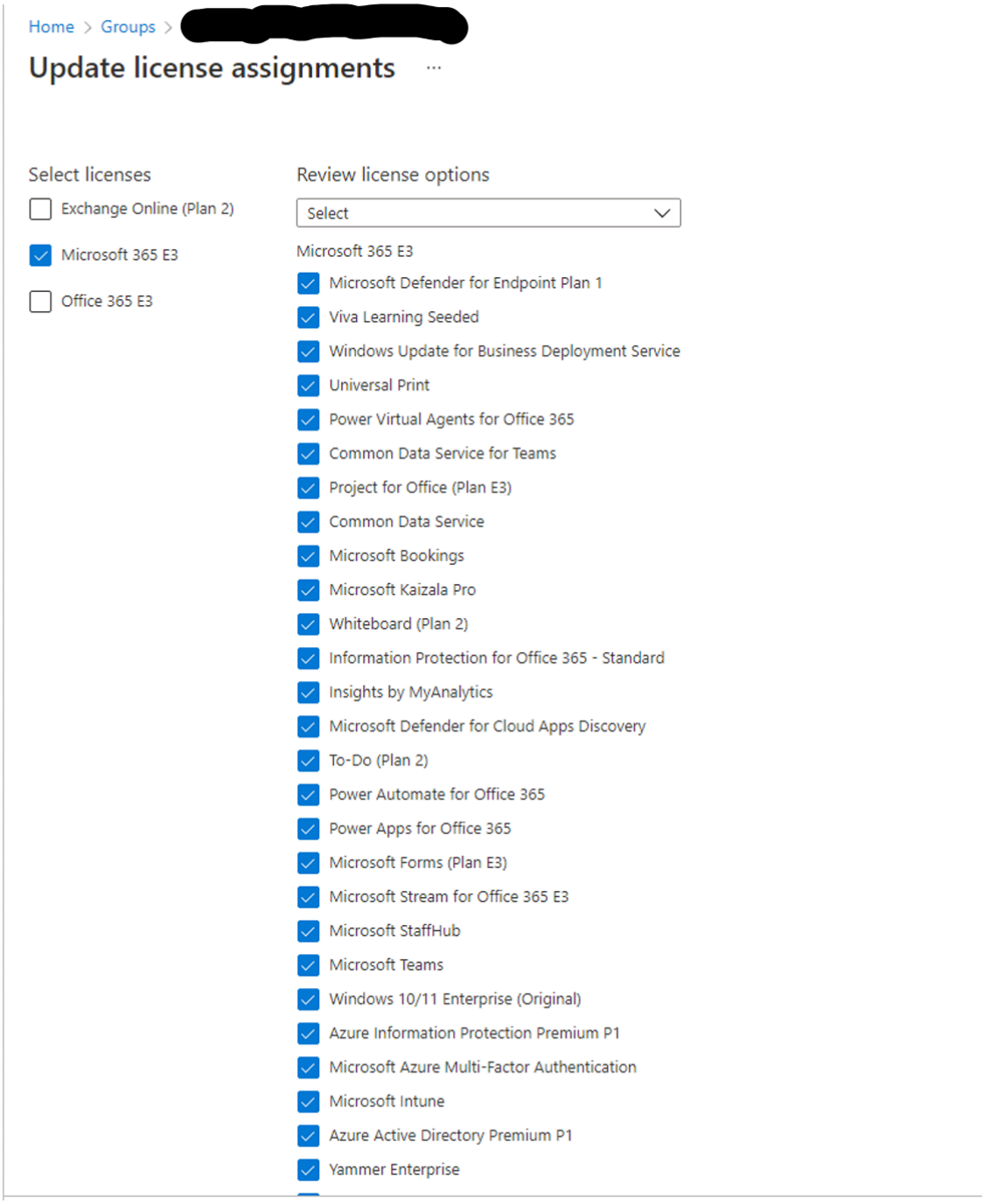


Assign required licenses for newly created Azure AD groups

Select the Azure AD group → Navigate to licenses



Click on Assignments – Select the licenses and sub-licenses appropriate for the Azure AD group.



6.1 How to migrate users with individual licenses to groups for licensing

You may have existing licenses deployed to users in the organizations via direct assignment; that is, using PowerShell scripts or other tools to assign individual user licenses. Before you begin using group-based

licensing to manage licenses in your organization, you can use this migration plan to seamlessly replace existing solutions with group-based licensing.

The most important thing to keep in mind is that you should avoid a situation where migrating to group-based licensing will result in users temporarily losing their currently assigned licenses. Any process that may result in removal of licenses should be avoided to remove the risk of users losing access to services and their data.

6.1.1 Recommended migration process

1. You have existing automation (for example, PowerShell) managing license assignment and removal for users. Leave it running as is.
2. Create a new licensing group (or decide which existing groups to use) and make sure that all required users are added as members.
3. Assign the required licenses to those groups; your goal should be to reflect the same licensing state your existing automation (for example, PowerShell) is applying to those users.
4. Verify that licenses have been applied to all users in those groups. This application can be done by checking the processing state on each group and by checking Audit Logs.
   - You can spot check individual users by looking at their license details. You will see that they have the same licenses assigned “directly” and “inherited” from groups.
   - You can run a PowerShell script to verify how licenses are assigned to users.
   - When the same product license is assigned to the user both directly and through a group, only one license is consumed by the user. Hence no additional licenses are required to perform migration.
5. Verify that no license assignments failed by checking each group for users in error state. For more information, see Identifying and resolving license problems for a group.

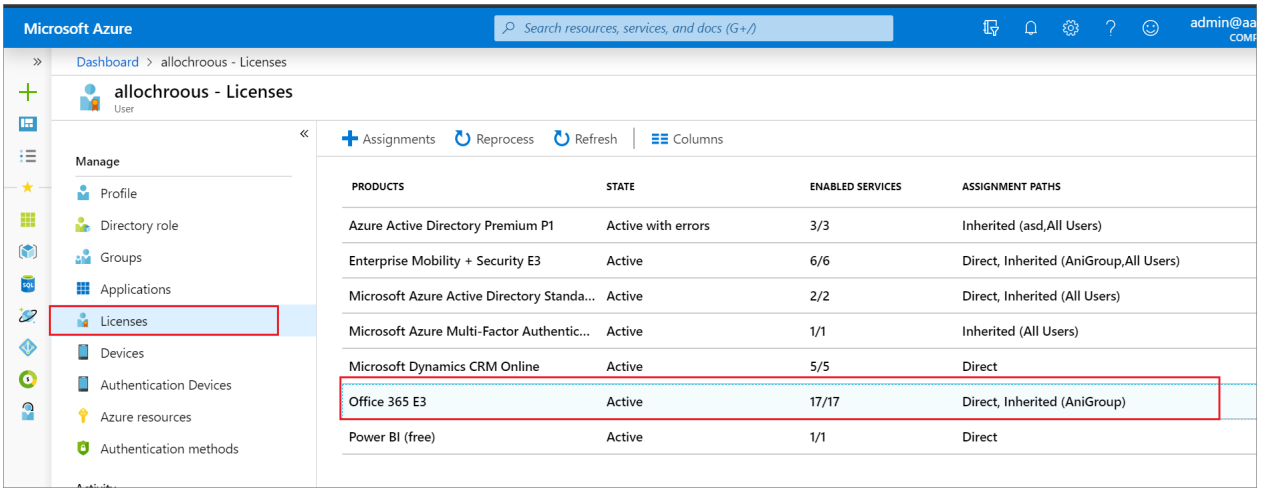
Consider removing the original direct assignments. We recommend that you do it gradually, and monitor the outcome on a subset of users first. If you could leave the original direct assignments on users, but when the users leave their licensed groups they retain the directly assigned licenses, which might not be what you want.

6.1.2 An example

An organization has 1,000 users. All users require Office 365 Enterprise E3 licenses. Currently the organization has a PowerShell script running on premises, adding and removing licenses from users as they come and go. However, the organization wants to replace the script with group-based licensing so licenses can be managed automatically by Azure AD.

Here is what the migration process could look like:

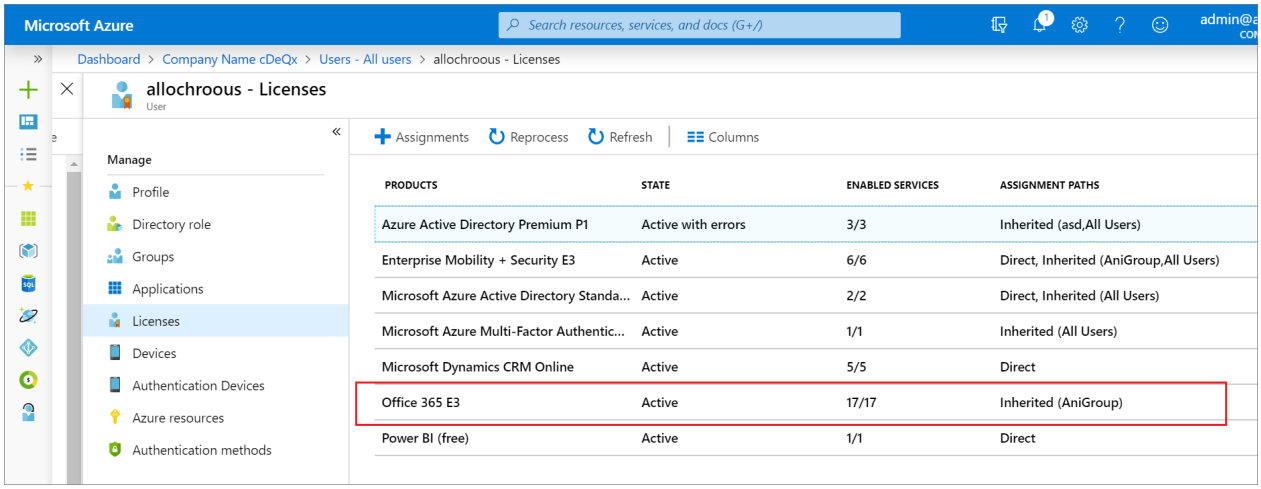
1. Using the Azure portal, assign the Office 365 E3 license to the **All users** group in Azure AD.
2. Confirm that license assignment has completed for all users. Go to the overview page for the group, select **Licenses**, and check the processing status at the top of the **Licenses** blade.
   - Look for “Latest license changes have been applied to all users" to confirm processing has completed.
   - Look for a notification on top about any users for whom licenses may have not been successfully assigned. Did we run out of licenses for some users? Do some users have conflicting license plans that prevent them from inheriting group licenses?
3. Spot check some users to verify that they have both the direct and group licenses applied. Go to the profile page for a user, select **Licenses**, and examine the state of licenses.
   - This is the expected user state during migration:



This confirms that the user has both direct and inherited licenses. We see that Office 365 E3 is assigned.

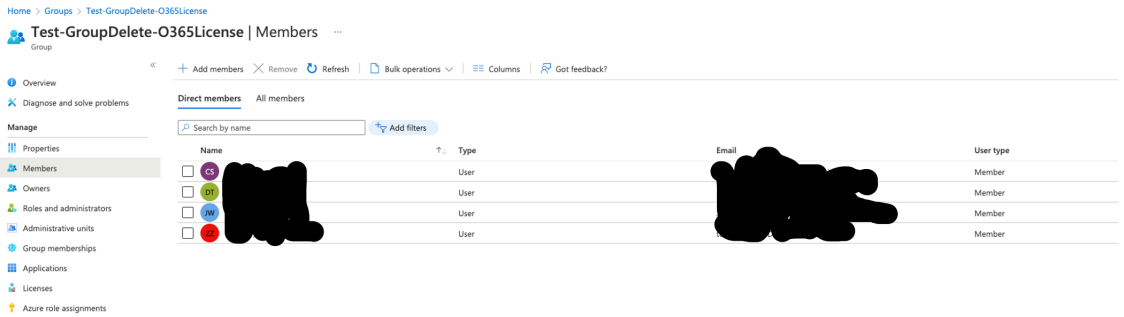
- Select each license to see which services are enabled. To verify that the direct and group licenses enable exactly the same services for the user, select **Assignments**.

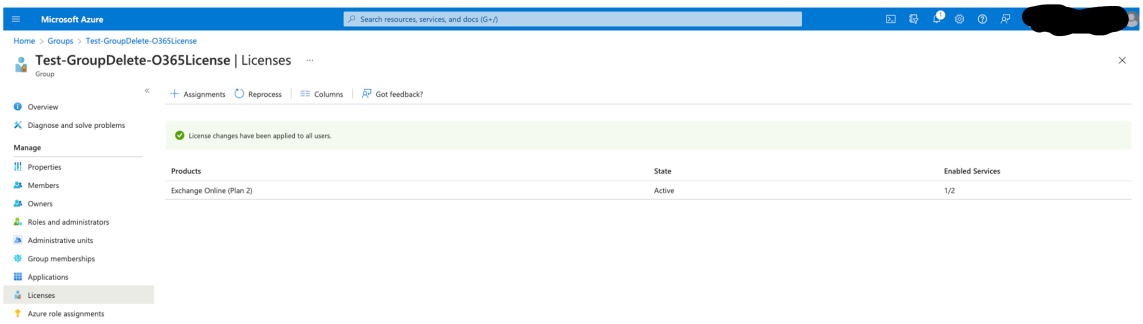
4. After confirming that both direct and group licenses are equivalent, you can start removing direct licenses from users. You can test this by removing them for individual users in the portal and then run automation scripts to have them removed in bulk. Here is an example of the same user with the direct licenses removed through the portal. Notice that the license state remains unchanged, but we no longer see direct assignments.



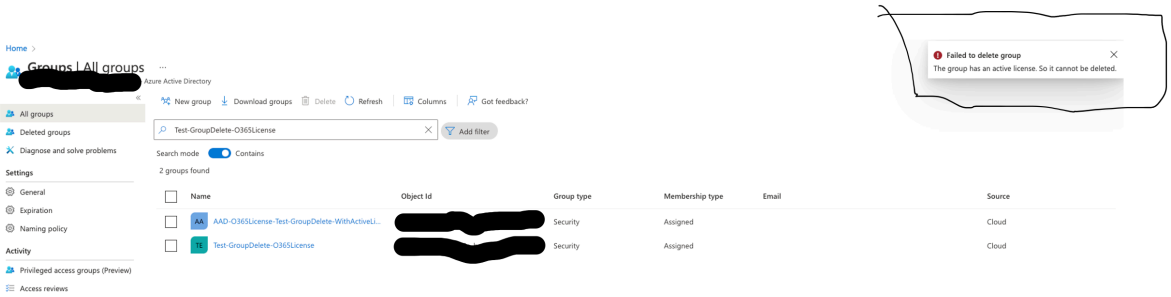
6.2 Deleting a group with an assigned license

It is not possible to delete a group with an active license assigned. An administrator could delete a group not realizing that it will cause licenses to be removed from users - for this reason we require any licenses to be removed from the group first, before it can be deleted.





When trying to delete a group in the Azure portal you may see an error notification like this:



Go to the **Licenses** tab on the group and see if there are any licenses assigned. If yes, remove those licenses and try to delete the group again.

You may see similar errors when trying to delete the group through PowerShell or Graph API. If you are using a group synced from on-premises, Azure AD Connect may also report errors if it is failing to delete the group in Azure AD. In all such cases, make sure to check if there are any licenses assigned to the group, and remove them first.

6.3 Usage location

Some Microsoft services are not available in all locations. Before a license can be assigned to a user, the administrator should specify the **Usage location** property on the user. In the Azure portal, you can specify usage location in **User** > **Profile** > **Settings**.

For group license assignment, any users without a usage location specified inherit the location of the directory. If you have users in multiple locations, make sure to reflect that correctly in your user resources before adding users to groups with licenses.

Note

Group license assignment will never modify an existing usage location value on a user. We recommend that you always set usage location as part of your user creation flow in Azure AD (for example, via AAD Connect configuration) - that will ensure the result of license assignment is always correct, and users do not receive services in locations that are not allowed.

Contact us

Manjunath Madiraju
Manager, Technical Architect

Email mmadiraju@kpmg.com

Ken Dunbar
Engagement Director

Email kbdunbar@kpmg.com

© 2021 [legal member firm name], a [jurisdiction] [legal structure] and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative ("KPMG International"), a Swiss entity. All rights reserved.

The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavour to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

The KPMG name, logo and "cutting through complexity" are registered trademarks or trademarks of KPMG International Cooperative ("KPMG International").