Azure Entra ID Configuration Guide

Document Metadata	
Title	Azure Entra ID Configuration Guide
Type	Guide
Classification	Public
Abstract	This document provides detailed instructions on how to configure the Azure Entra ID application for Single Sign-On (SSO) integration with our system. The guide covers the process from application creation to assigning the necessary permissions. Screenshots are included at each step to ensure clarity and ease of follow-through.

Introduction

Purpose of this Guide

The purpose of this guide is to facilitate the configuration of Azure Entra ID for SSO integration with our system, ensuring a seamless authentication experience for users.

Overview of Single Sign-On (SSO) with Azure Entra ID

Single Sign-On (SSO) is an authentication process that allows users to access multiple applications and services using one set of login credentials (such as a username and password). This eliminates the need for users to log in separately to each system, simplifying the user experience and enhancing productivity.

How SSO Works with Azure Entra ID

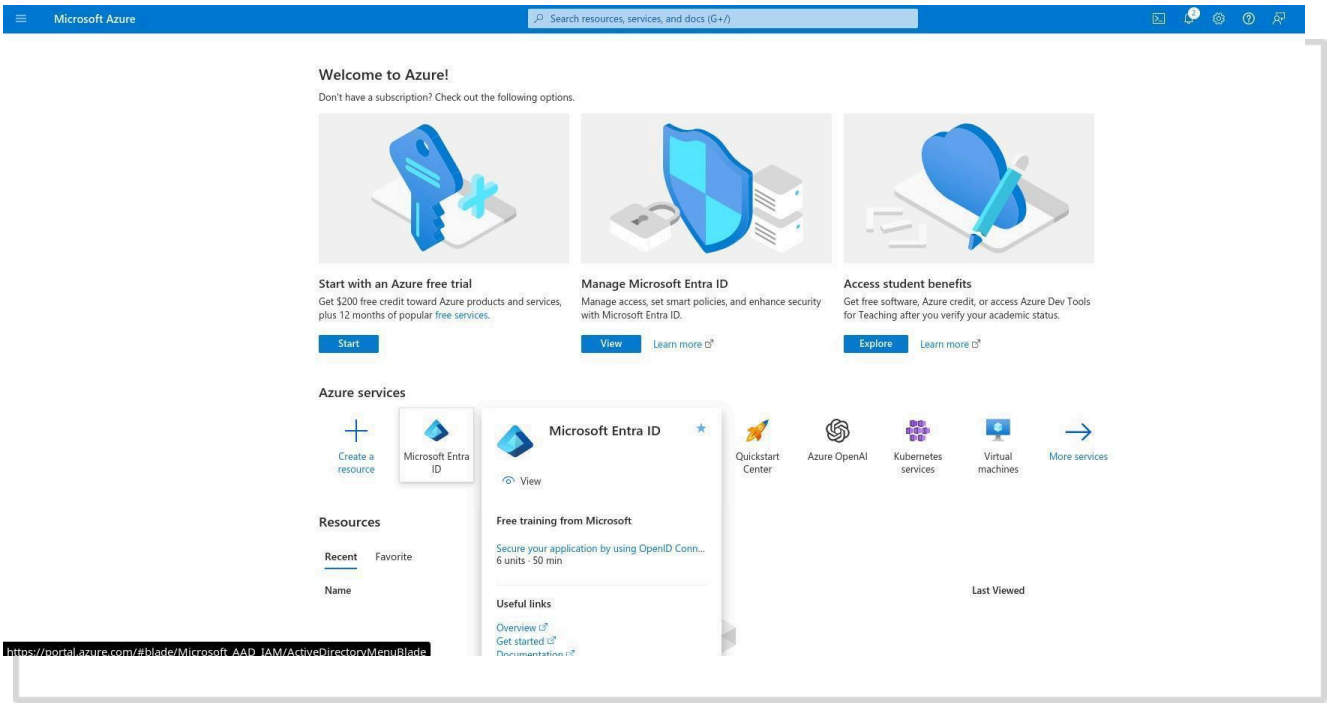
Azure Entra ID, formerly known as Azure Active Directory (Azure AD), is Microsoft's cloud-based identity and access management service. In the context of SSO, Azure Entra ID serves as a trusted central authentication point that provides access control and identity management across cloud and on-premise applications.

Configuration Steps on Azure

Detailed step-by-step guide for configuring the Azure Entra ID application, supported by screenshots. Each step should be clearly described and include:

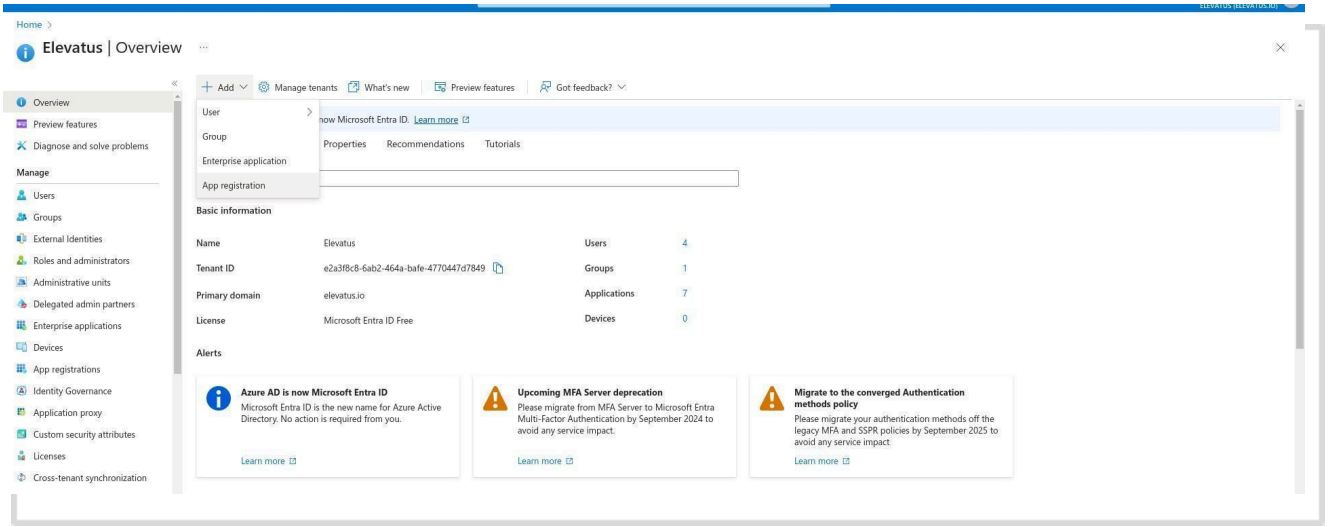
Step 1: Accessing Entra ID

Navigate to the Azure portal and log in with your credentials. Search for Microsoft Entra ID in the Azure Services section and click on it.



Step 2: Create a New Application

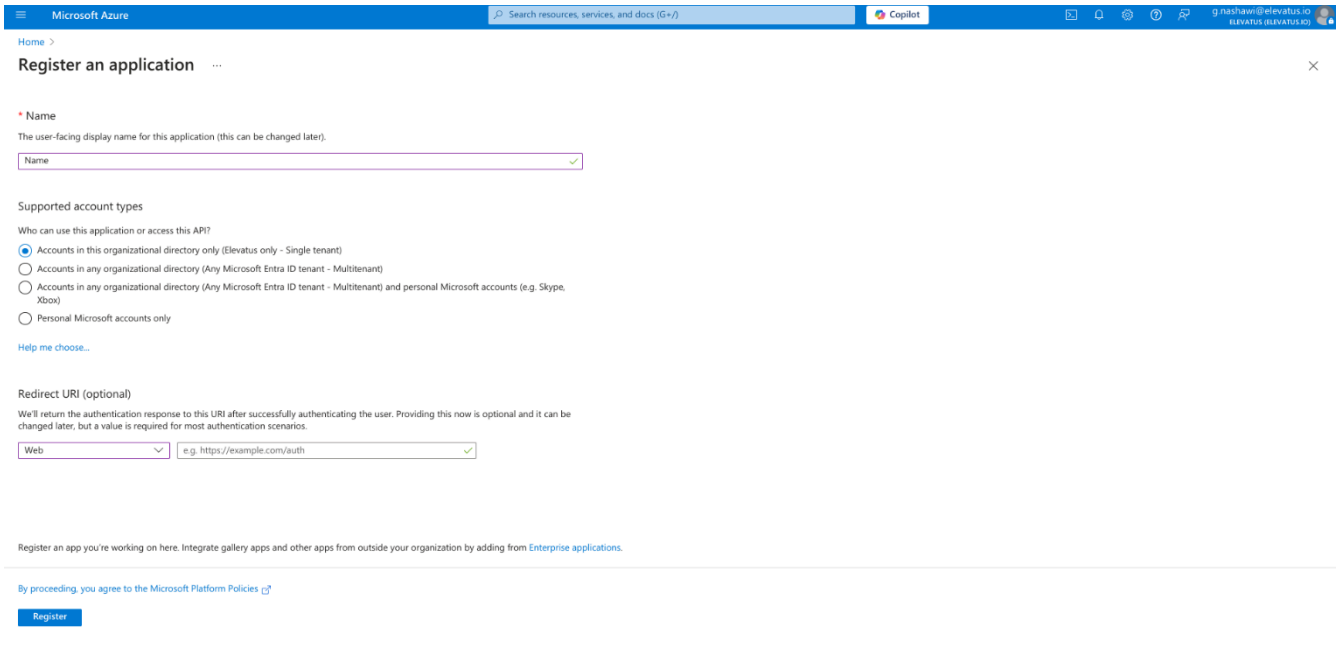
Once you have entered the Entra ID directory, you'll find the + Add button. Select it and navigate to "App registration".

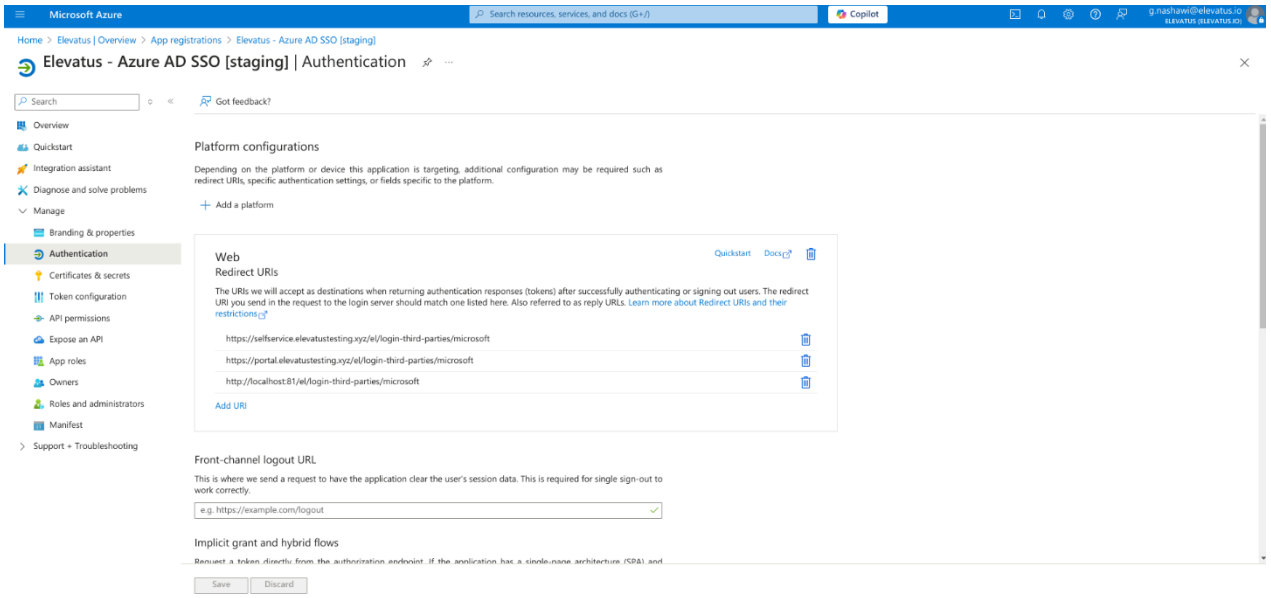


Enter a name for your application, and select "Accounts in this organizational directory only (Single Tenant)". Make sure a redirect URI is set up as well, by selecting the "Web" from the dropdown, and adding the following URLs depending on the Elevatus data center being used:

Dev Datacenter	https://dev.elevatus.ai/accounts/login-third-party/microsoft
KSA Datacenter	https://app.elevatus.ai/accounts/login-third-party/microsoft

During registering an application, you'll be able to add only one Redirect URI, however, you can add more later on after the app is created.





Note: The application only accepts a single redirect URI during creation. Additional URIs can be added afterward.

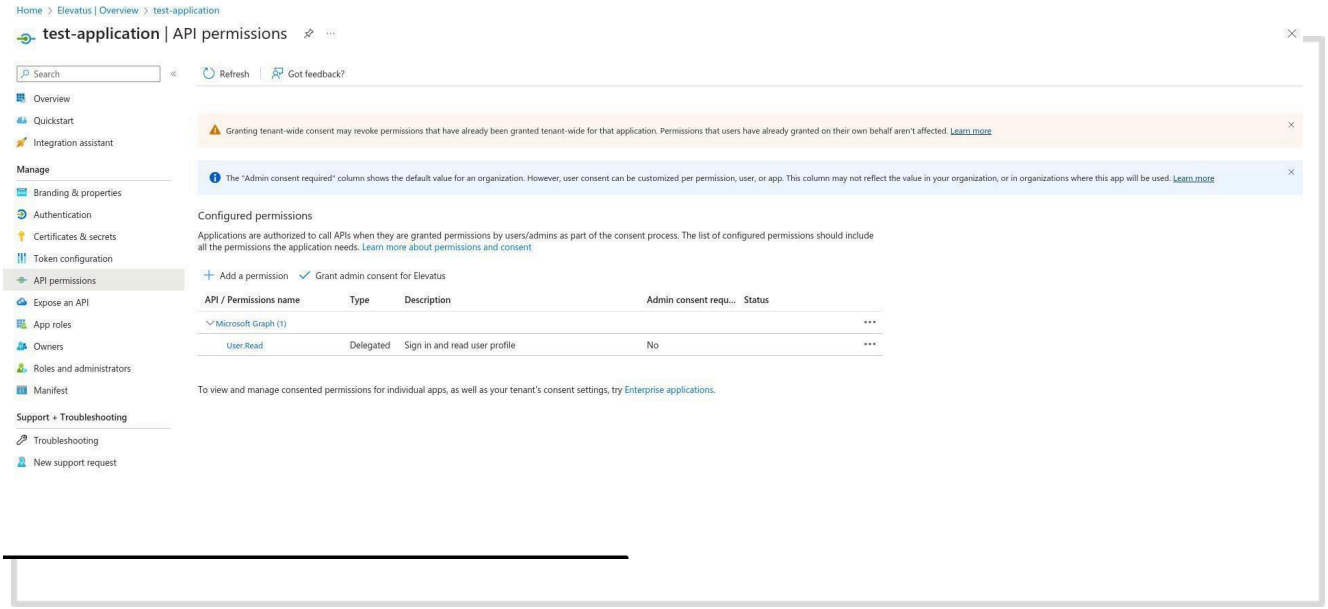
Step 3: Assigning Permissions

In order for the app to work properly, there are specific permissions that need to be enabled as a baseline, which allow our app to communicate with the web app you are creating.

Required Permissions

```
Directory.Read.Al
l Group.Read.All
GroupMember.Read.All
Member.Read.Hidden
RoleManagement.Read.All
RoleManagement.Read.Directory
User.Read
User.Read.All
User.ReadBasic.All
```

To begin, open the newly created app and navigate to "API Permissions" in the sidebar

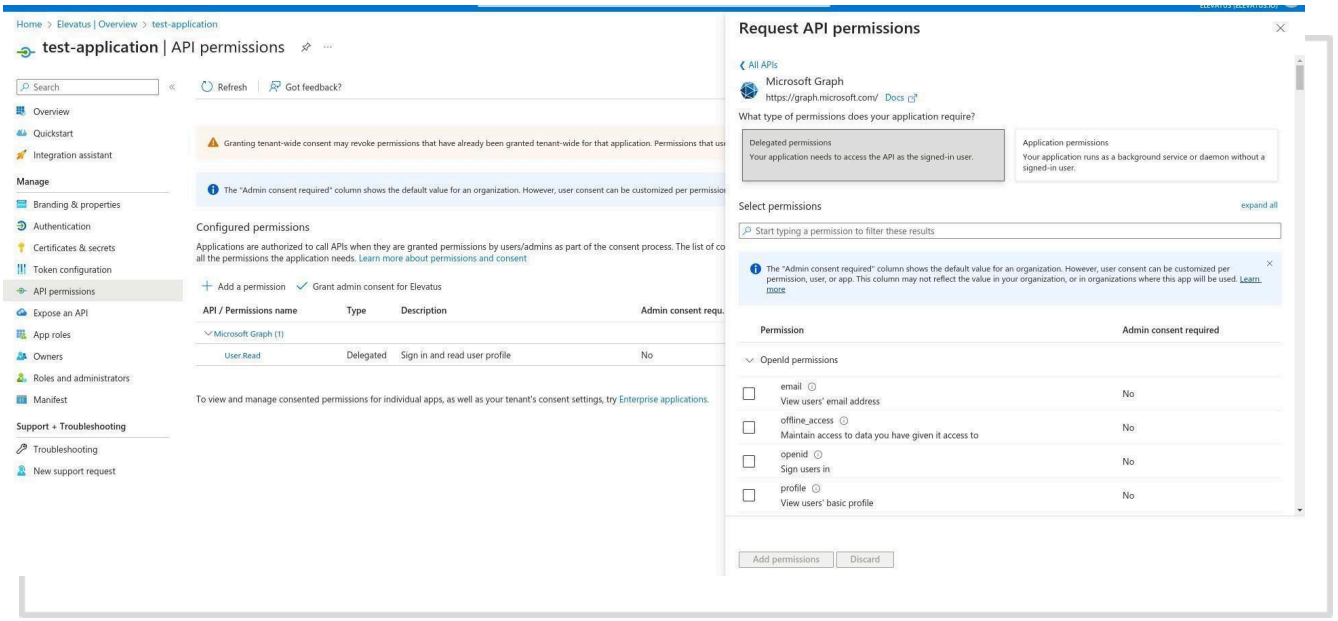


Click on "+ Add a permission", then select the Microsoft Graph API.

There are two types of permissions:

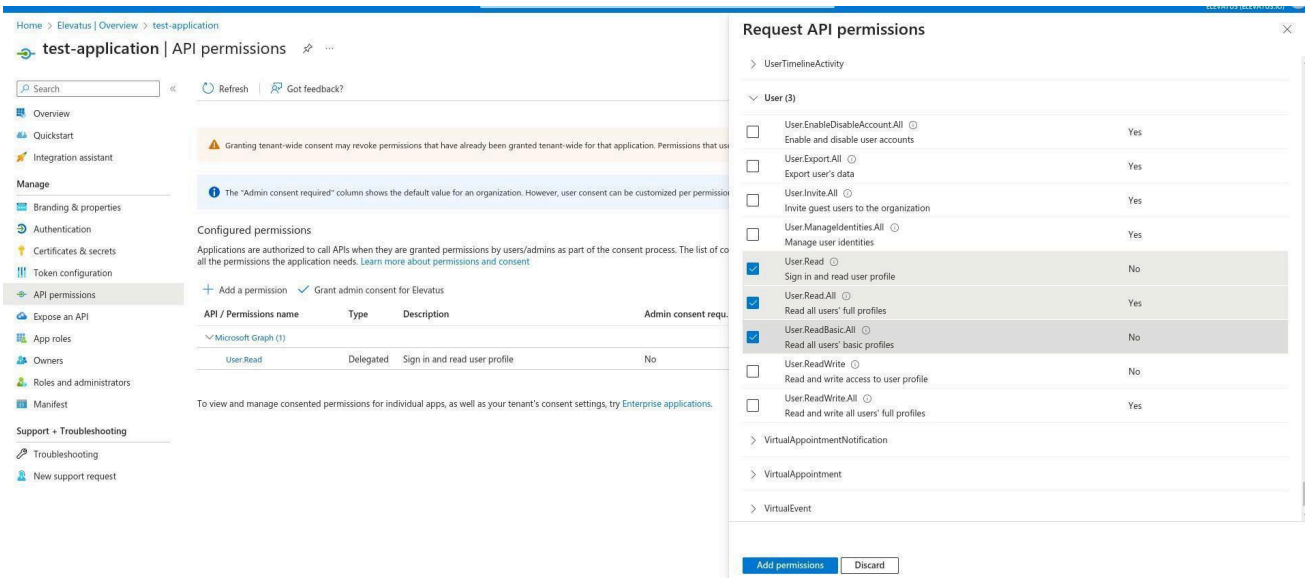
- **Delegated Permissions:** allow an app to act on behalf of a signed-in user with restricted access.
- **Application Permissions:** grant the app independent access to resources without user involvement.

The required permissions should be added to both **Application Permissions, and Delegated Permissions.**



Note: The **User.Read** permission can be found and identified under delegated permissions only.

You can type in the search bar to search for the permissions mentioned above in the list, similar to the screenshot below:



Granting Access

Finally, we need to click on "Grant admin Consent for Directory Name}", which will register those permissions and as such, allow us to be able to connect to it.

Step 4: Assigning Groups (optional)

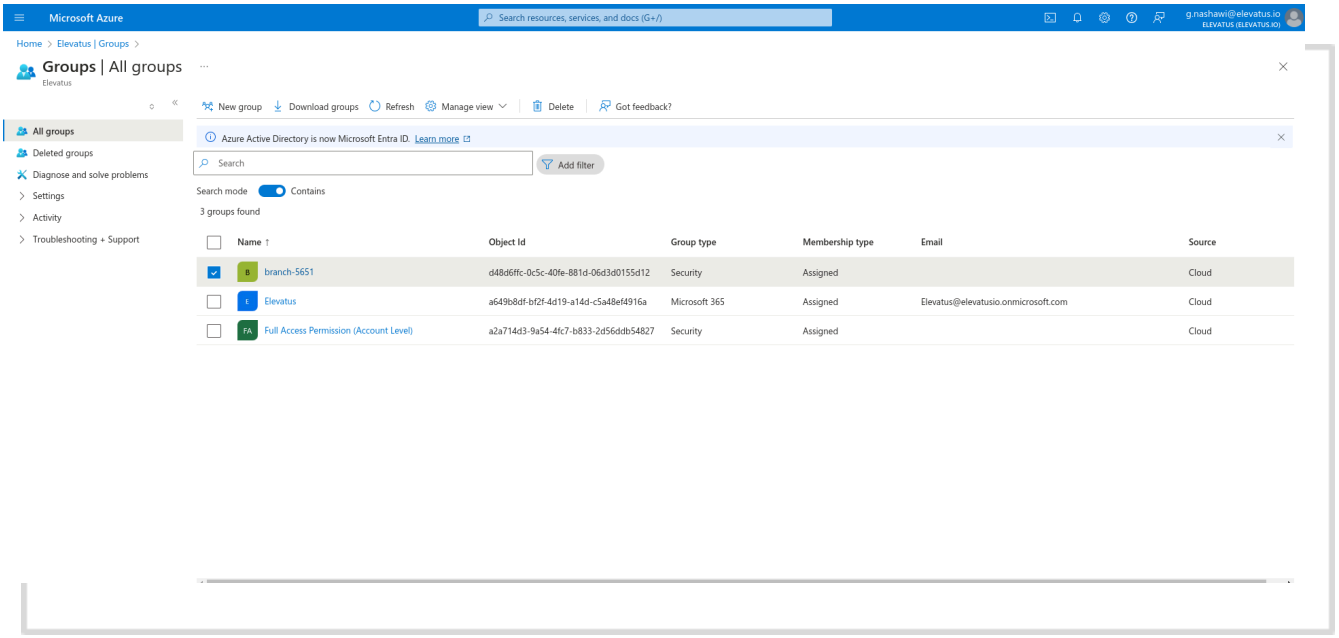
This step is optional and involves assigned users to groups. It is particularly important to reflect the correct structure of the directory, so that if you have multiple branches, then adding them as follows to a group, will help us auto-sync them into the correct branch.

Create a branch-designated group

In Elevatus, all branches must have codes generated for it. The code is the unique identifier in the system of the customer, it is how the customer refers to this entity or resource inside their internal systems. In technical terms, it is referred to as the Foreign Key.

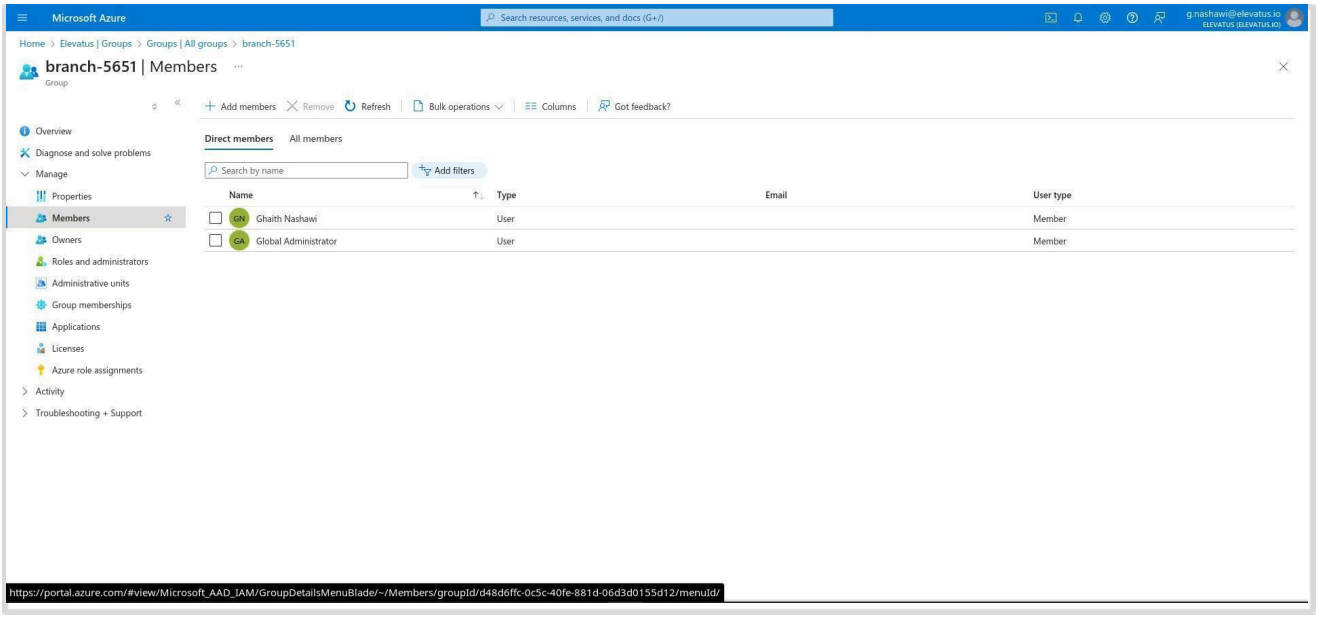
Create an access group according to the naming convention:

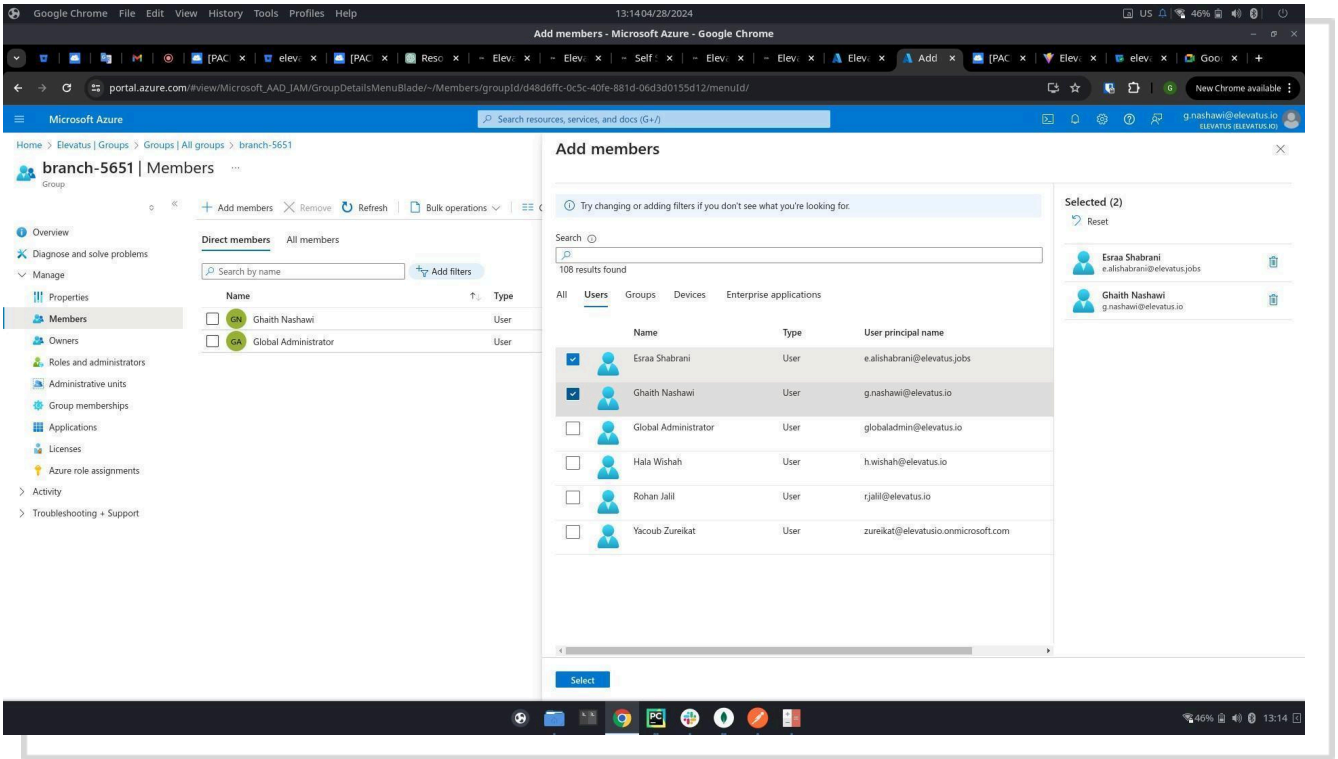
- branch-{code}



Collect members into the group

Select the members you'd like to add to the newly-created group.





And that completes the branch segregation. Nothing further is required on this end.

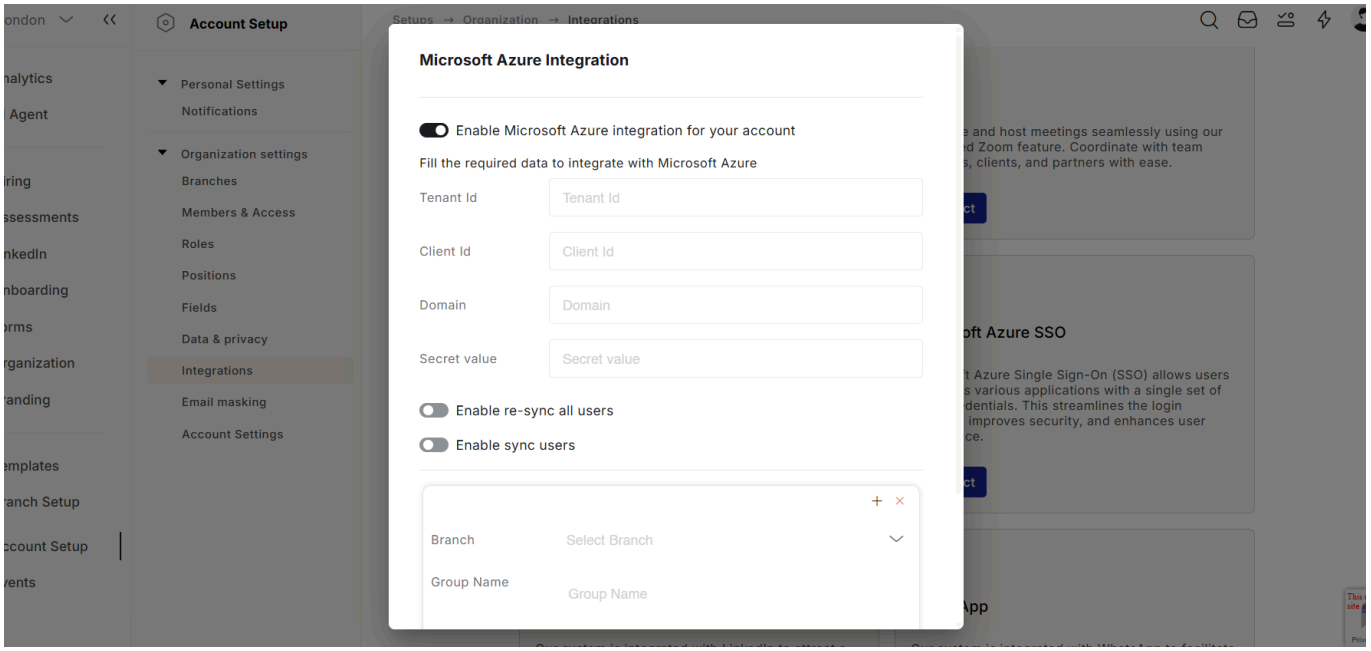
Configuration Steps on Elevatus

After the app is registered successfully on Azure, we now need to configure it on Elevatus. The process is seamless and efficient.

Under the account setup In the **Integrations** page, navigate to Microsoft

Azure SSO and click on **Connect**. You'll be presented with the following

form to be filled:



The following details must be entered:

- Tenant ID of the created app
- Client ID of the created app
- Secret value of the created app
- Domain of the Entra ID directory Finally,

click save, and the process is complete.

Testing and Validation

You can always create a test app to check whether the connection is working or not. You have the ability to rotate credentials or revoke access at any time.

Appendices

Appendix A - Revision History

Version	Date	Editor	Approver	Change Description
1	Apr 23, 2024	Yanal Kashou	Yanal Kashou	Composed initial document
2	Apr 28, 2024	Yanal Kashou	Yanal Kashou	Added updates on branch synchronization