Migrate from Azure Database for PostgreSQL Single Server to Flexible Server

Contents

Migrate from Azure Database for PostgreSQL Single Server to Flexible Server 1

How does it work? 1

How can it be consumed? 1

What happens behind the scenes? 2

Current Limitations 2

Pre-requisites 2

- Create a migration using Azure portal 9
- Create a migration using Azure CLI 9

Post Migration 10

This document contains details of an automated solution to migrate schema and data from an Azure Database for PostgreSQL – Single-Server instance to Azure Database for PostgreSQL – Flexible Server with minimal downtime to the application. It requires you to create a target flexible server instance and to take care of few pre-requisites before you try a migration. All the details related to pre-requisites are covered in the later sections of this document. This solution migrates only schema and data. Other server components such as server parameters, connection security details, users, and roles, tags must be manually configured in the target flexible server.

How does it work?

The migration service is a hosted solution where we deploy a VM on Azure and automatically setup the all the infrastructure needed for doing an online migration. The migration service uses Azure database migration service (DMS) which internally uses the logical replication of the PostgreSQL engine to support online migration.

It automates all the steps that are needed to do an online migration such as setting up of DMS, creating the database in the target server, migrating schema, handling of foreign keys and triggers, adding firewall rules at both source and target to allow DMS access them, etc, and thus simplifying the process of migration.

How can it be consumed?

The migration service is currently exposed through wizard-based Azure portal experience and via easy-to-use Azure CLI commands. You can create migrations, list migrations, display migration details, modify state of the migration, and delete migrations. The details of how to perform these actions are covered in detail in later section of this document.

The migration service supports a variety of configurations:

- 1 : 1. For example, For example, a standalone migration of database(s) from a single server instance into a flexible server instance.

- 1 : Many. For example, migration of database(s) from a single server instance into multiple flexible server instances.
- Many : 1. For example, database migrations from multiple single servers into a Flexible Server instance.
- Migrations across versions can also be carried out hassle-free.
  For example, database(s) from Single Server PostgreSQL version 11 can be migrated to a Flexible Server PostgreSQL version 13.

What happens behind the scenes?

The migration service is built on top of Azure DMS and the following steps are automated

- Creation of a Azure DMS in the region of the target flexible server
- Creation of a DMS project with both source and target types as Azure database for PostgreSQL
- Creation of DMS activity of type online migration which aims to migrate the databases specified by the user from source to target.

Upon successful cutover of the DMS activity, the data and schema of your single server databases will be migrated to your flexible server.

Current Limitations

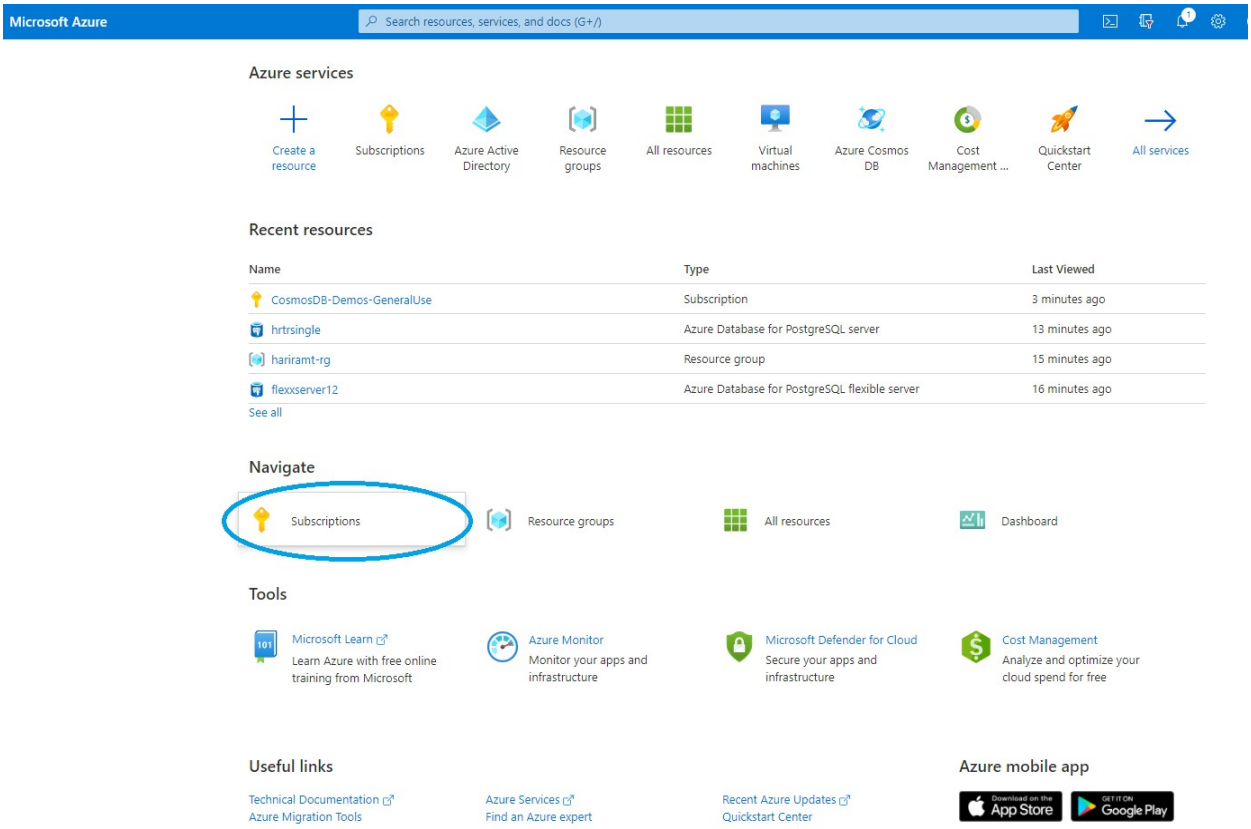
The following are the current limitations of this migration service

- All DMS based limitations apply to this migration solution as well.
- This is a logical replication solution that will use resources in the source single-server instance. Plan to scale resources (CPU, memory, and storage IOPS) accordingly.
- All limitations applicable to logical replication are applicable to the solution as well.
- You can migrate up to eight databases per server in a single migration attempt. If you have more than eight databases to migrate, you can create multiple migration attempts. Migrating more than eight databases in parallel from a source single-server instance may put extra load on the source server.
- An Azure Active Directory App (AAD App) with appropriate privileges is required for this automated solution to work. The solution cannot be used without an Azure Active Directory App.
- There will be a short downtime to the application during cutover. The amount of downtime depends on other post-migration tasks that may need to be performed after attempting the cutover.
- This solution migrates data and schema for the database. It does not migrate other managed service features such as server parameters, connection security details, firewall rules, users, roles and permissions. In other words, everything except data and schema must be manually configured in the target server.
- It does not validate the data being moved. This must be done by the customers before pointing their application to the flexible server.
- This solution only migrates user databases and not system databases like **template_0**, **template_1, azure_maintenance, and azure_sys**.

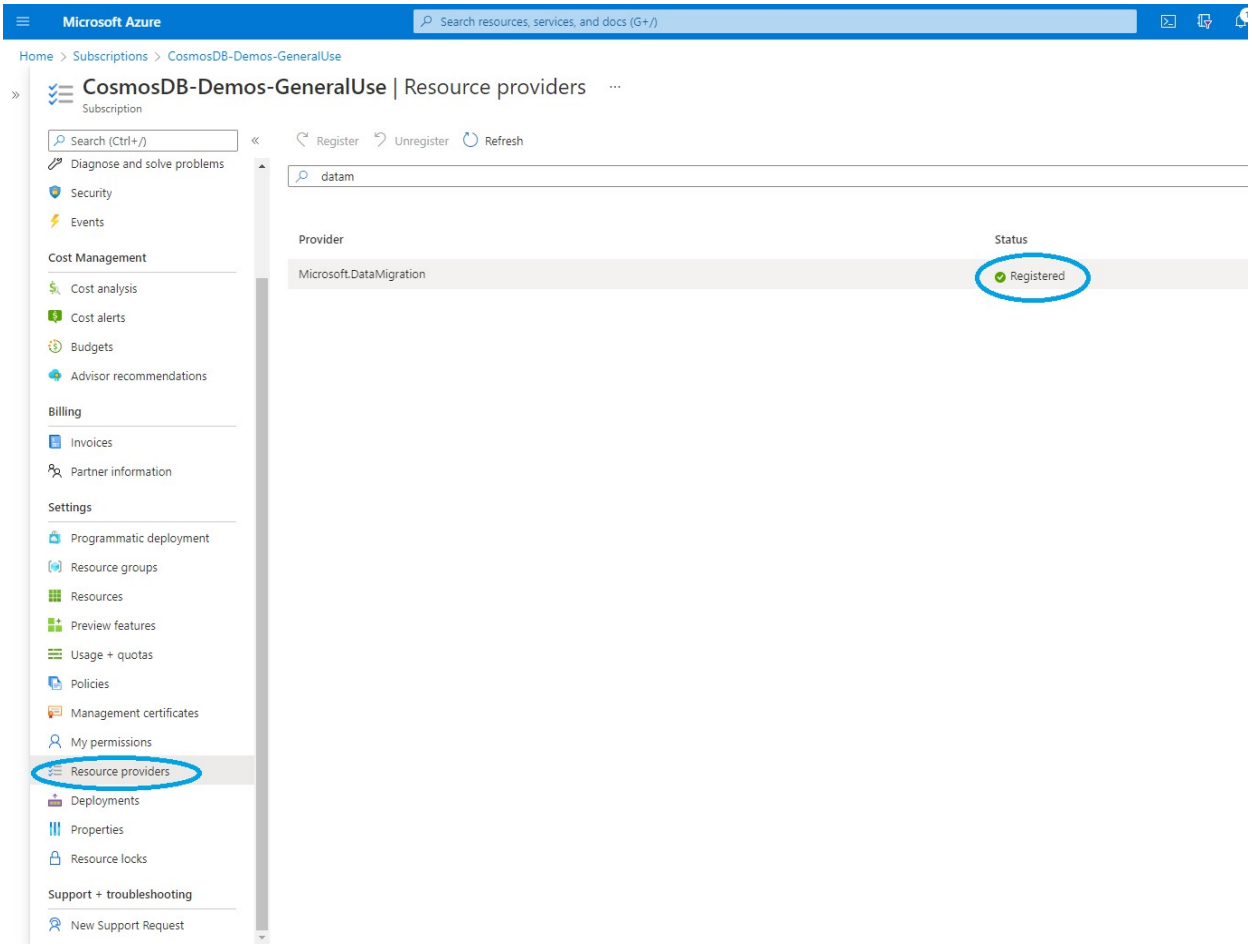
Pre-requisites

- **Target server creation**
  - ✓ You need to create the target PostgreSQL flexible server before starting the migration. Use the creation QuickStart guide to create one.
- **Resource provider pre-requisites –** The 'Microsoft.DataMigration' Resource provider has to be enabled in the subscription.

✓ Go to the Azure portal homepage and navigate to your ‘Subscriptions’ as shown below.

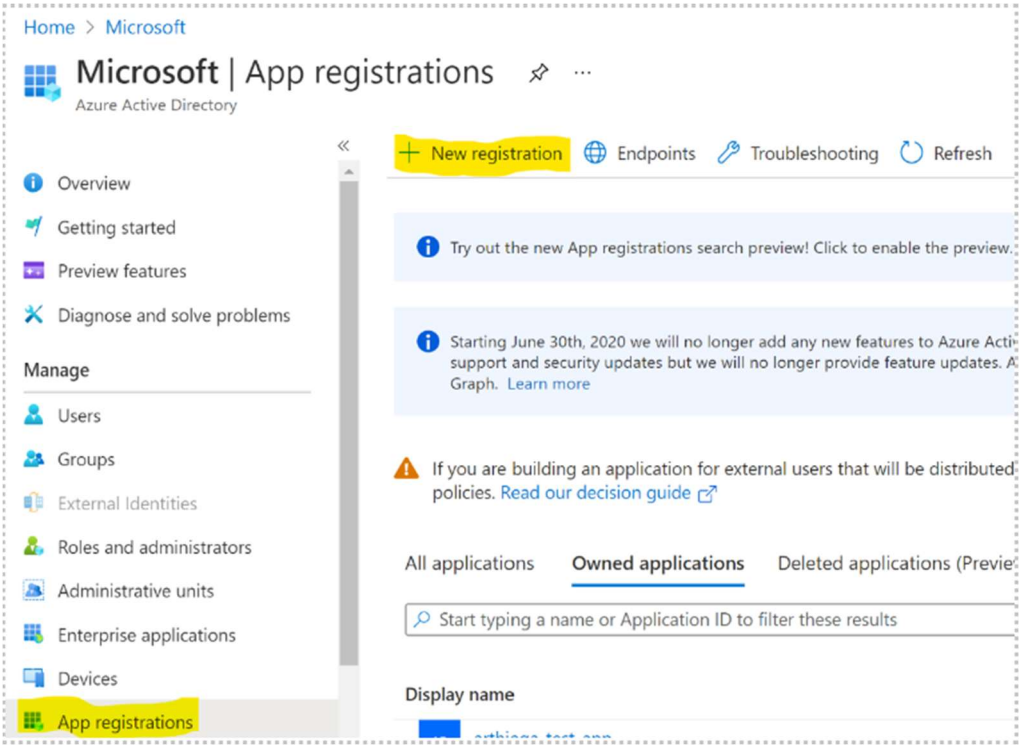


✓ Once you have selected your subscription, choose ‘Resource Providers’ from the menu on the left and register ‘Microsoft.DataMigration’ as shown below.

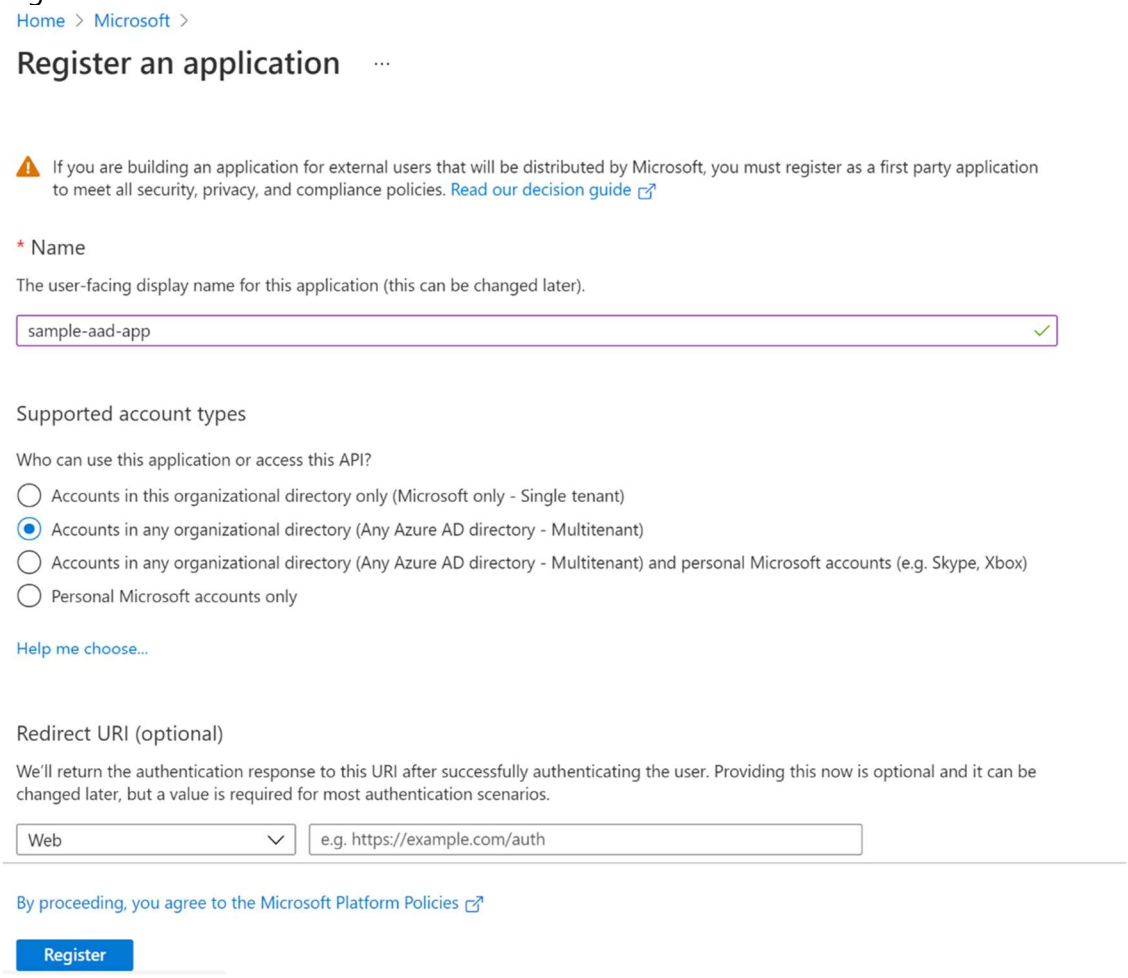


- **Source server pre-requisites** - This automated migration solution uses Azure DMS to do an online migration from the source to target. As a result, you must enable logical replication pre-requisites in the source DB server. Enabling logical replication will require a server reboot for the change to take effect. You have a couple of options to enable logical replication in the source.
    - ✓ You can enable it yourself and reboot the source server when possible.
    - ✓ You can have this automated solution enable logical replication via Azure CLI command in the source server. We will cover this aspect later in this document.

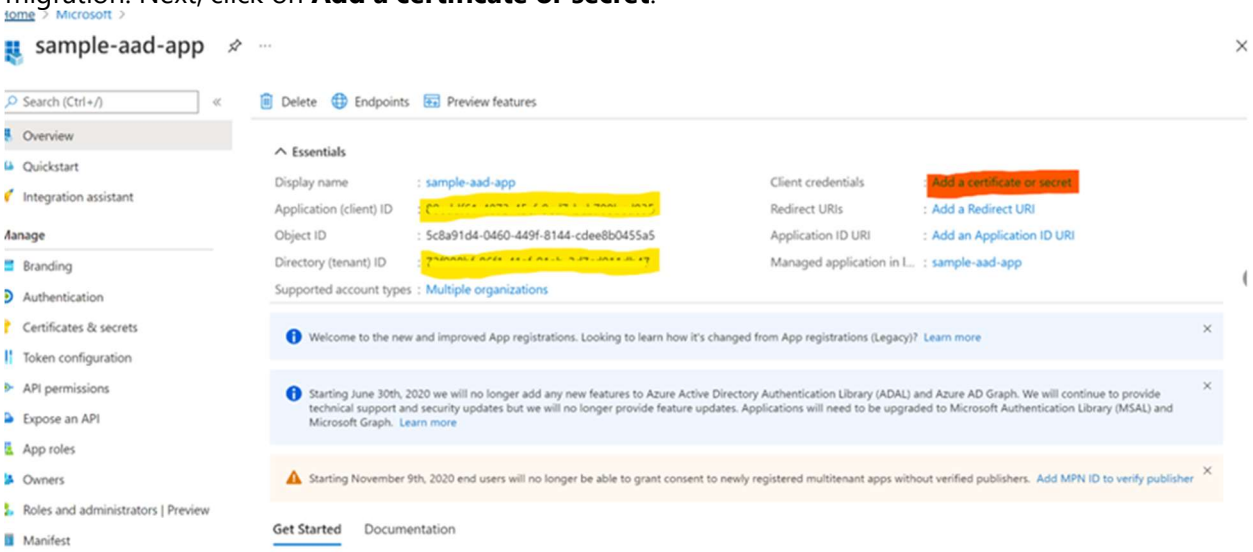
- **Azure Active Directory App set up** - One of the most important components of this automated solution is the creation of Azure Active Directory app (AAD App) which helps in role-based access control. This automation service needs access to both the source and target servers. Access to these resources is restricted by the roles assigned to the Azure Active Directory App. The AAD app instance once created, can be used to manage multiple migrations. To get started, create a new Azure Active Directory Enterprise App by doing the following.
    - ✓ Search for Azure Active Directory in the search bar on the top in the portal.
    - ✓ Within the Azure Active Directory portal, under manage on the left, choose App Registrations.
    - ✓ Click on new registration.



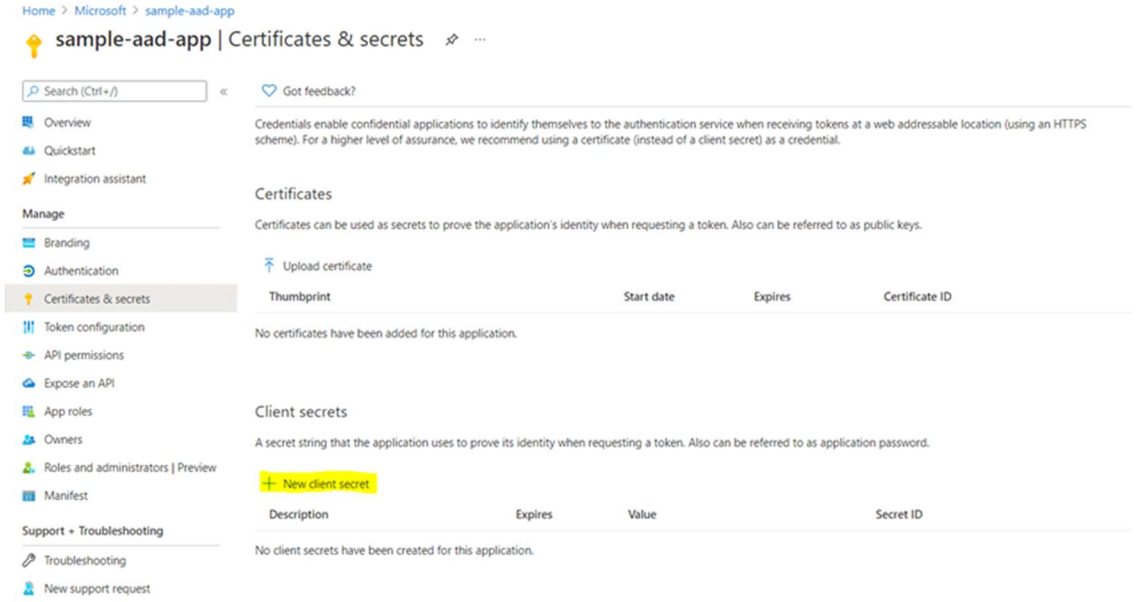
✓ Give the app registration a name, choose an option that suits your needs for account types and click register



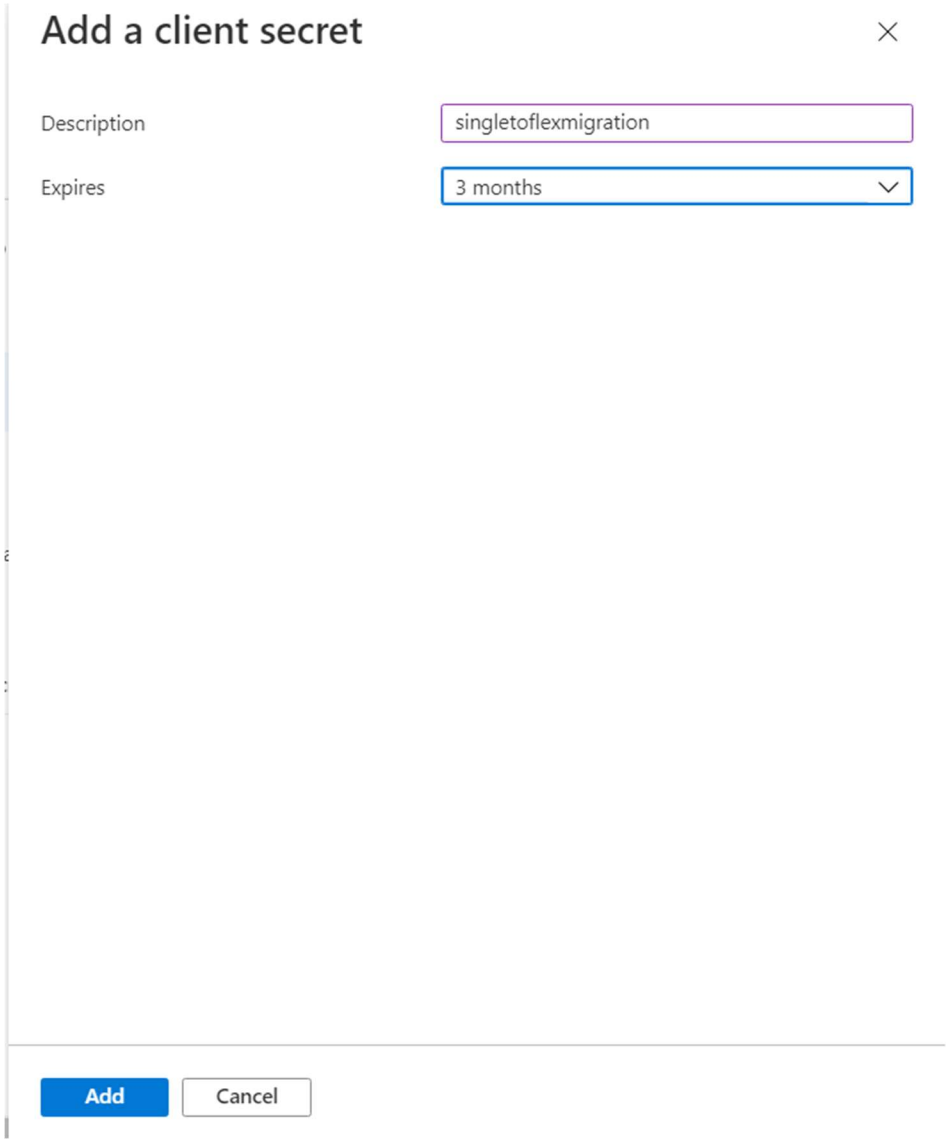
✓ Once the app is created, you can copy the client ID and tenant ID required for later steps in the migration. Next, click on **Add a certificate or secret**.



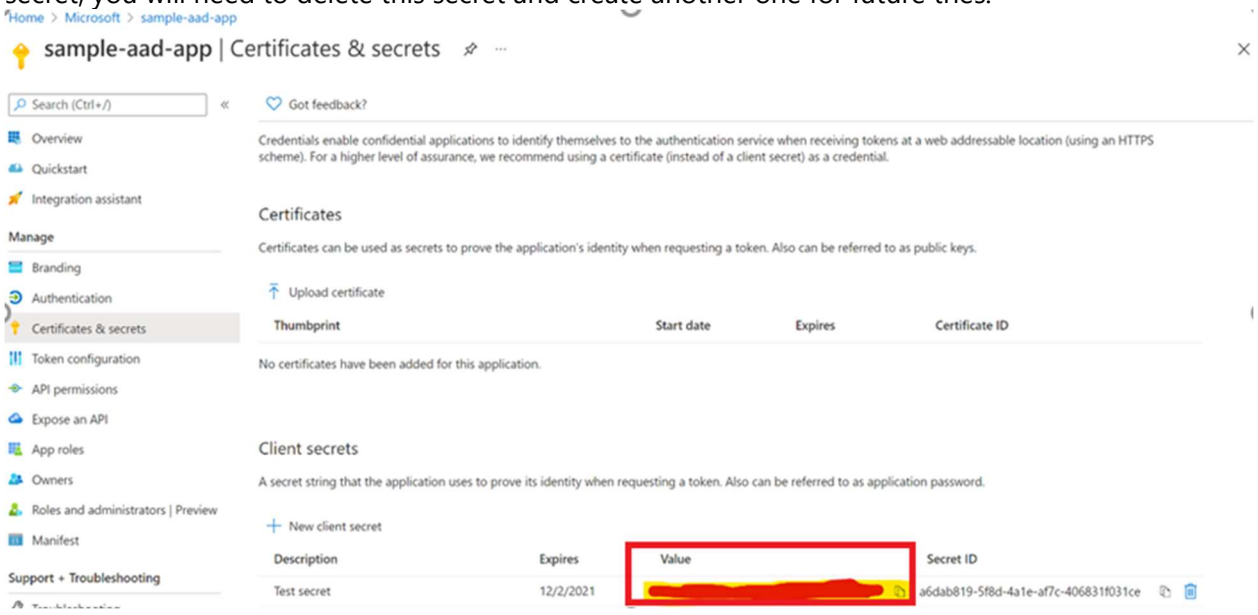
✓ In the next screen, click on **New client secret**.



✓ In the fan-out blade that opens, add a description, and select the drop-down to pick the life span of your Azure Active Directory App. Once all the migrations are complete, the **Azure Active Directory App which was created for Role Based Access Control can be deleted**. The default option is six months. If you do not need Azure Active Directory App for six months, choose three months and click **add**.

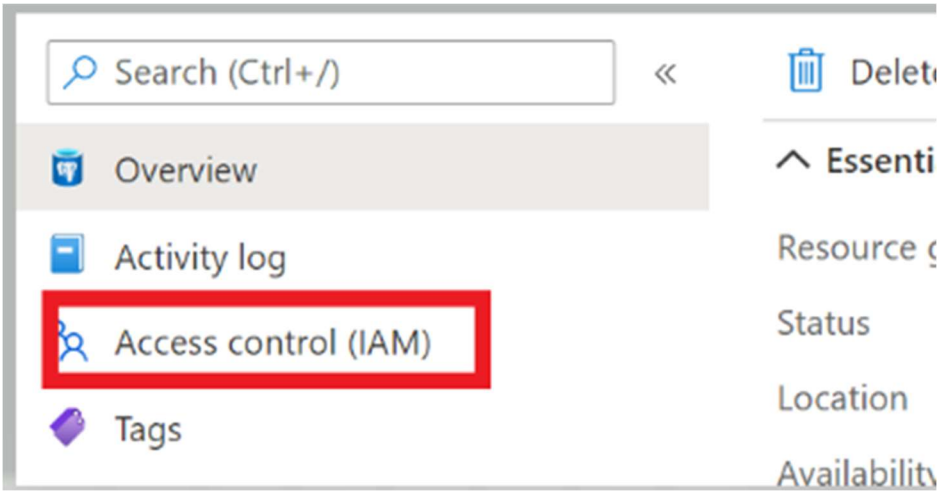


✓ In the next screen, copy the Value column (highlighted in the below pic) which has the details of the Azure Active Directory App secret. This can be copied only while creation. If you miss copying this secret, you will need to delete this secret and create another one for future tries.

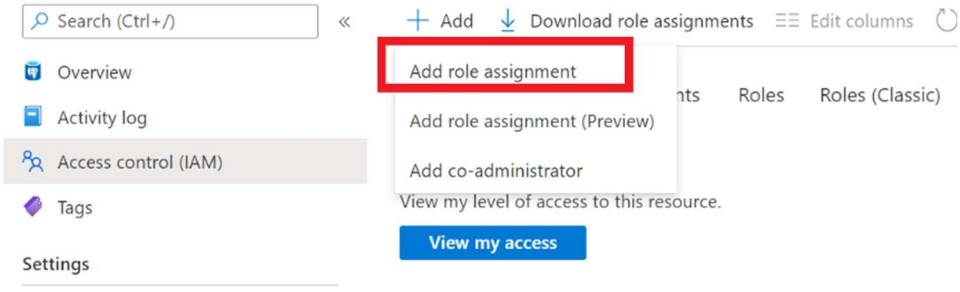


- ✓ Once Azure Active Directory App is created, you will need to add contributor privileges for this Azure Active Directory app to the following resources:
  - **REQUIRED**: Source single server you are migrating from.
  - **REQUIRED**: Target flexible server you are migrating into.
  - **REQUIRED**: Resource group for the migration (By default this is the target flexible server resource group). (Or) If you are using a temporary resource group to create the migration infrastructure, the Azure Active Directory App will require contributor privileges to this resource group as well.
  - **NOTE:**
    - If the source or the target happens to be inside a VNet, then the Azure Active Directory App will require contributor privileges to corresponding VNet.
    - If the source and the target happen to be in different VNets, then the Azure Active Directory app will require contributor privileges to both the source and target VNets

Let us look at how to add contributor privileges to an Azure resource. For the target flexible server, do the following: - Select the target flexible server in the Azure portal. - Click on Access Control (IAM) on the top left.

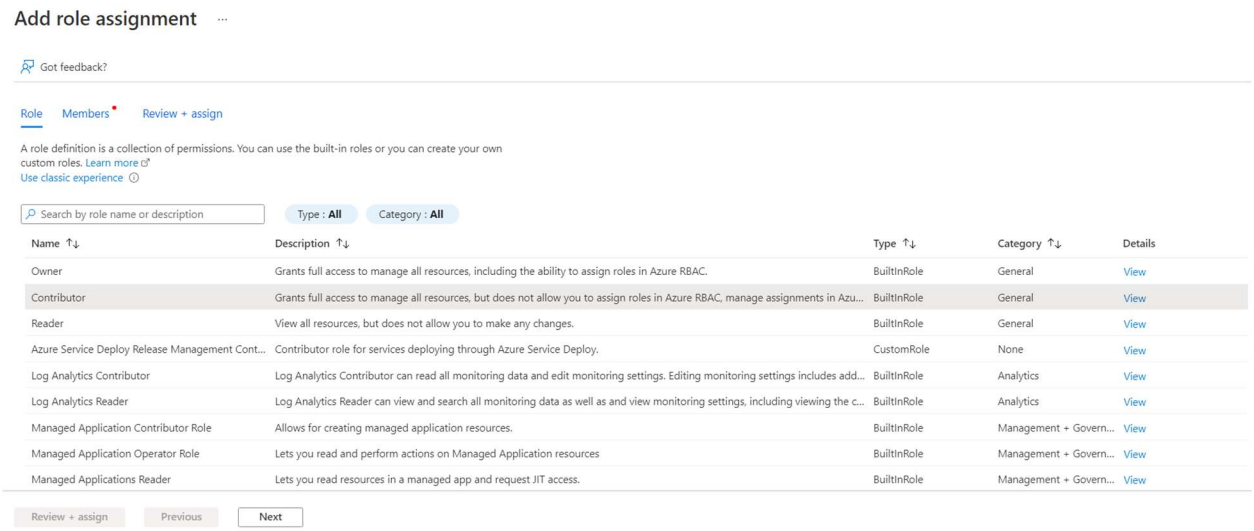


Click **Add** and choose **Add role assignment**.

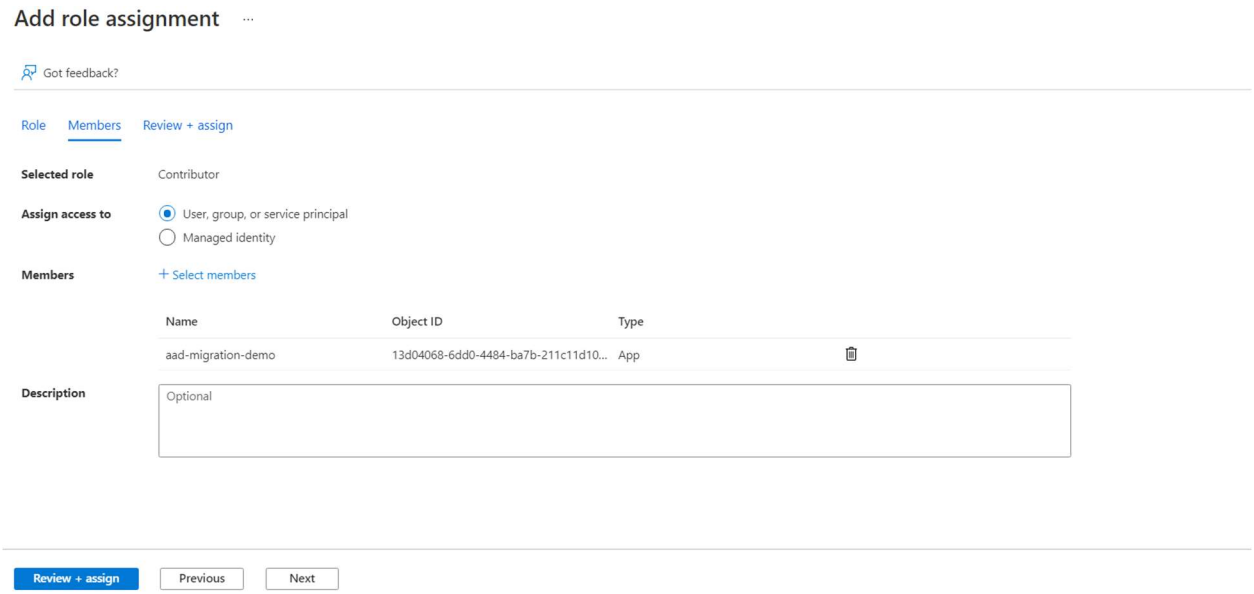


Note: The Add role assignment capability is only enabled for users in the subscription with role type as **Owners**. Users with other roles do not have permission to add role assignments.

Under the Role tab, click on **Contributor** and click **Next** button



Under the Members tab, keep the default option of **Assign access to** to User, group or service principal and click **Select Members**. Search for your Azure Active Directory App and click on **Select**. - Click on **Review and Assign**



The Azure Active Directory App now has contributor privileges to the target flexible server instance. Repeat process for all the required resources.

Once all these pre-requisites are taken care of, you are now ready to start the migration process.

- Create a migration using Azure portal
- Create a migration using Azure CLI

Post Migration

- Note that all the resources created by this migration solution will be automatically cleaned up irrespective of whether the migration has **succeeded/failed/cancelled**. There is no action required from your end. This includes clearing up of the source's logical slots.
- If your migration has failed and if you want to retry the migration, then you need to create a new migration with a different name and try running it again. For now, there is no option of retry on a failed migration.
- If you have more than eight databases on your single server and want to migrate all of them, it is recommended to create multiple migrations between the same single server and flexible server with each migration migrating a set of eight databases each.
- For security reasons, it is highly recommended to delete the Azure Active Directory app once the migration completes.
- Post data validations and making your application point to flexible server, you can consider deleting your single server.