Offline Migration from 'Azure Database for PostgreSQL Single Server' to 'Azure Database for PostgreSQL Flexible Server'

Contents

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

1. How does it work?

The migration tool is a hosted solution where we deploy a VM on Azure and automatically setup the all the infrastructure needed for doing an offline migration.
The migration tool uses Azure database migration service (DMS) to drive the offline migration.

It automates all the steps that are needed to do an offline 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.

It is suitable for databases which can afford downtime and doesn't have to adhere to logical replication limitations.

2. How can it be consumed?

The offline migration service is currently exposed through a wizard-based Azure portal experience. 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 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.

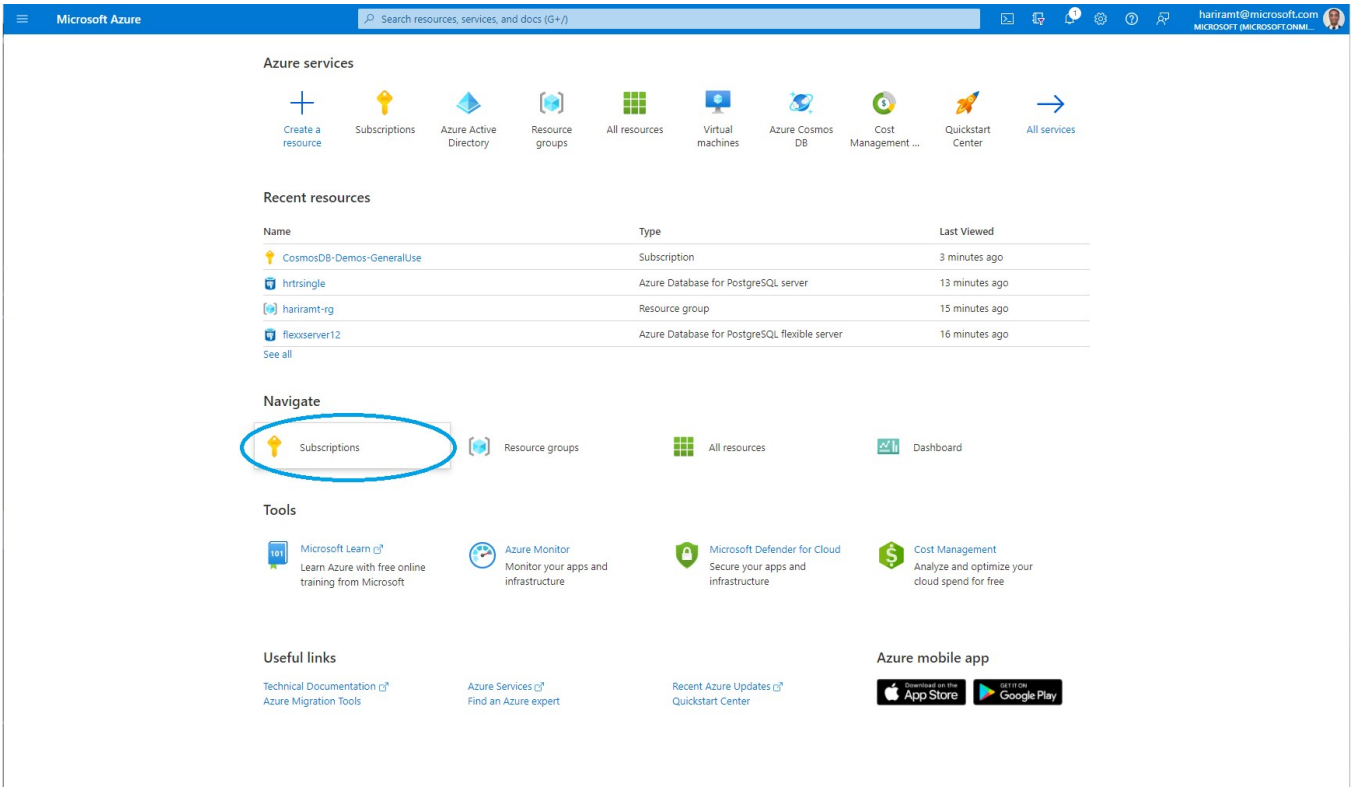
3. Current Limitations

The following are the current limitations of this migration service:

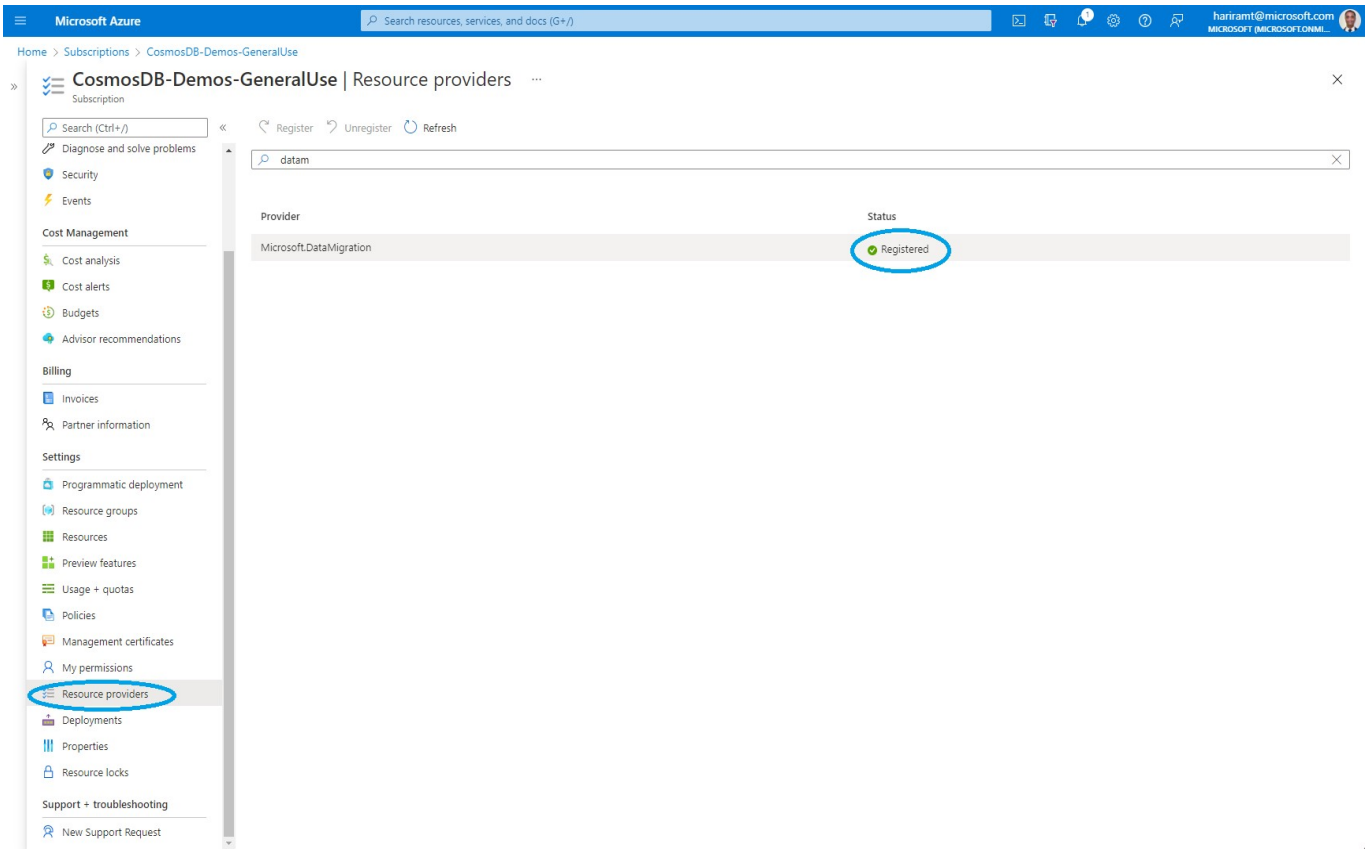
- There will be a downtime to the application during migration. The duration of the downtime will depend upon the size of the data being migrated. Duration estimates are available here.
- Known issues/migration limitations with online migrations apply to this migration solution as well.
- You can migrate 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 an 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.
- 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 migrated. This must be done by the customers after migration and before pointing their application to the flexible server.
- This solution only migrates user databases and not system databases such as **template_0**, **template_1, azure_maintenance, and azure_sys**.

4. 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.

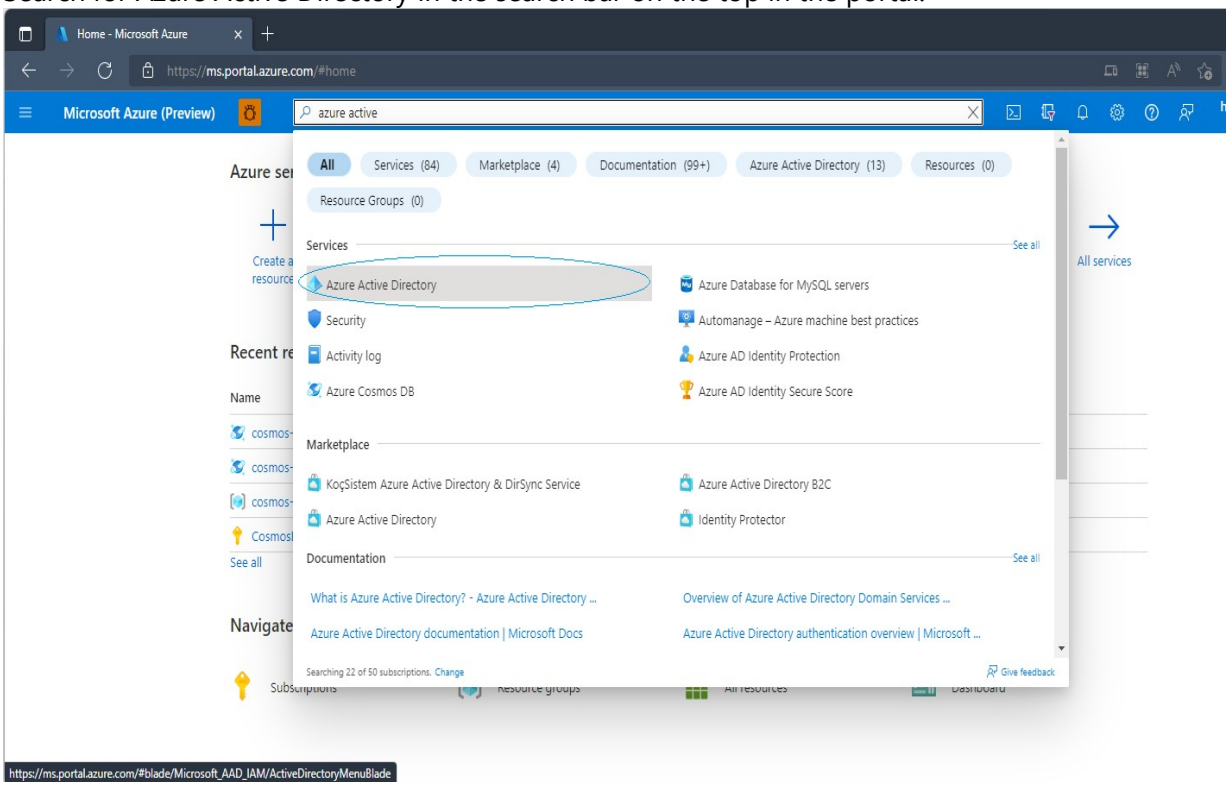


- **Source server pre-requisites** - This automated migration solution uses Azure DMS to do an offline migration from the source to target.

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.

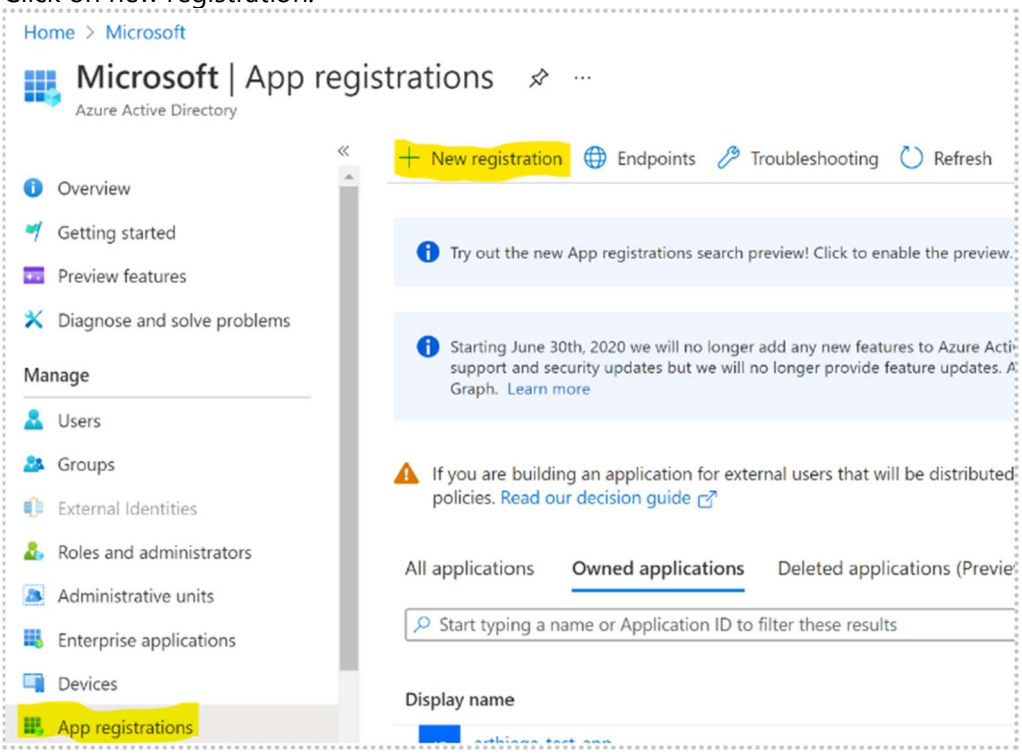
- ✓ You can enable it manually and reboot the source server when possible.
- ✓ You can have this automated solution enable logical replication via Azure CLI command in the source server.
  We will cover this aspect later in this document.

- **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.

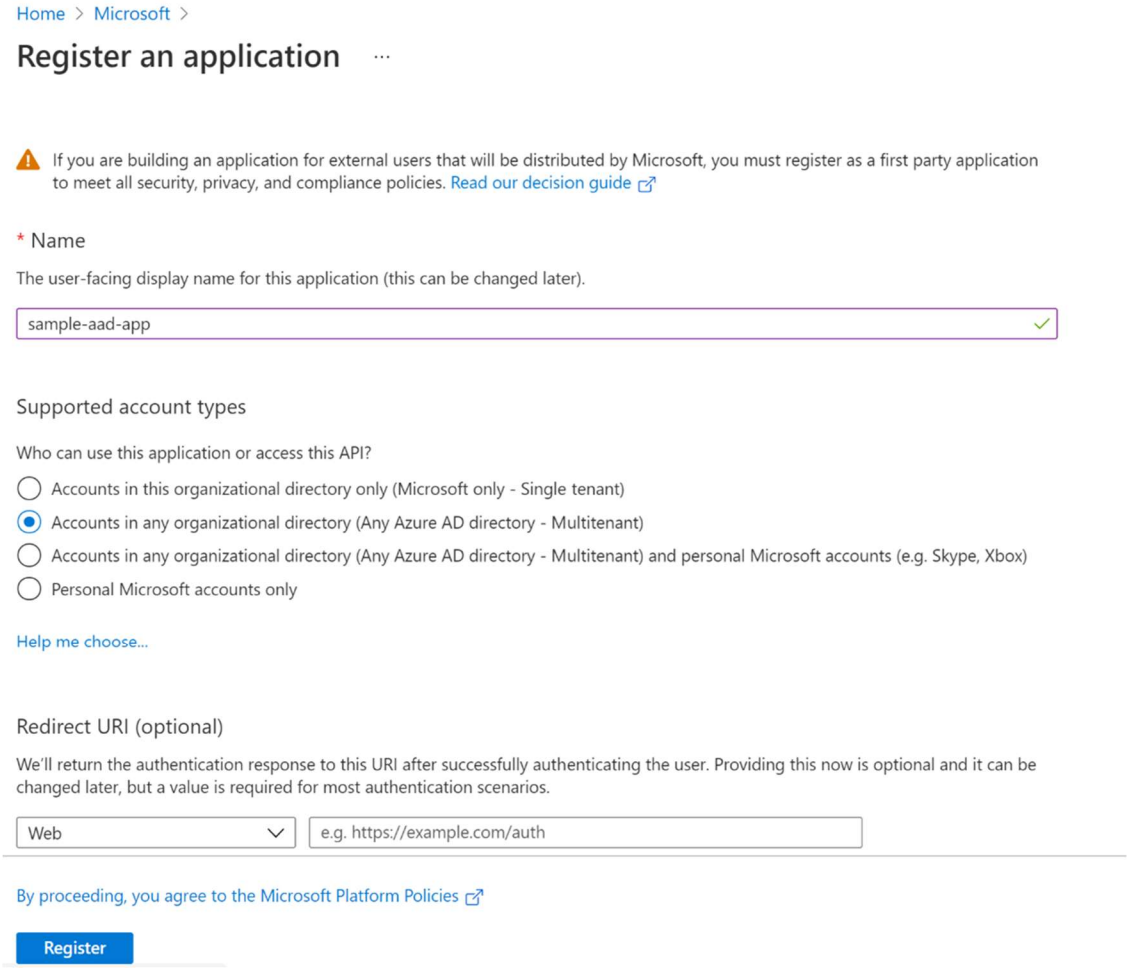


  - ✓ Within the Azure Active Directory portal, under manage on the left, choose App Registrations.

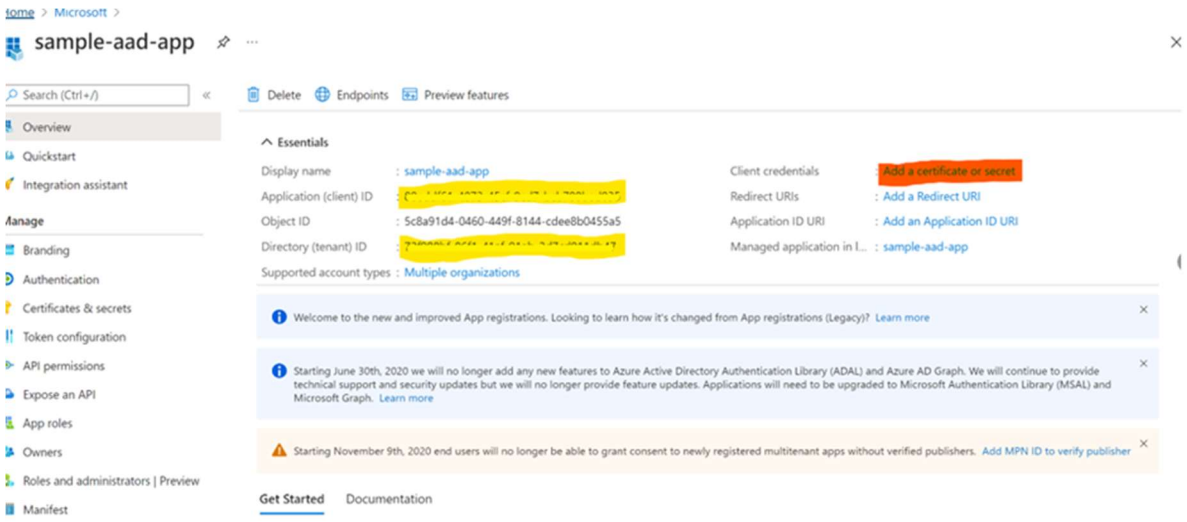
✓ Click on new registration.



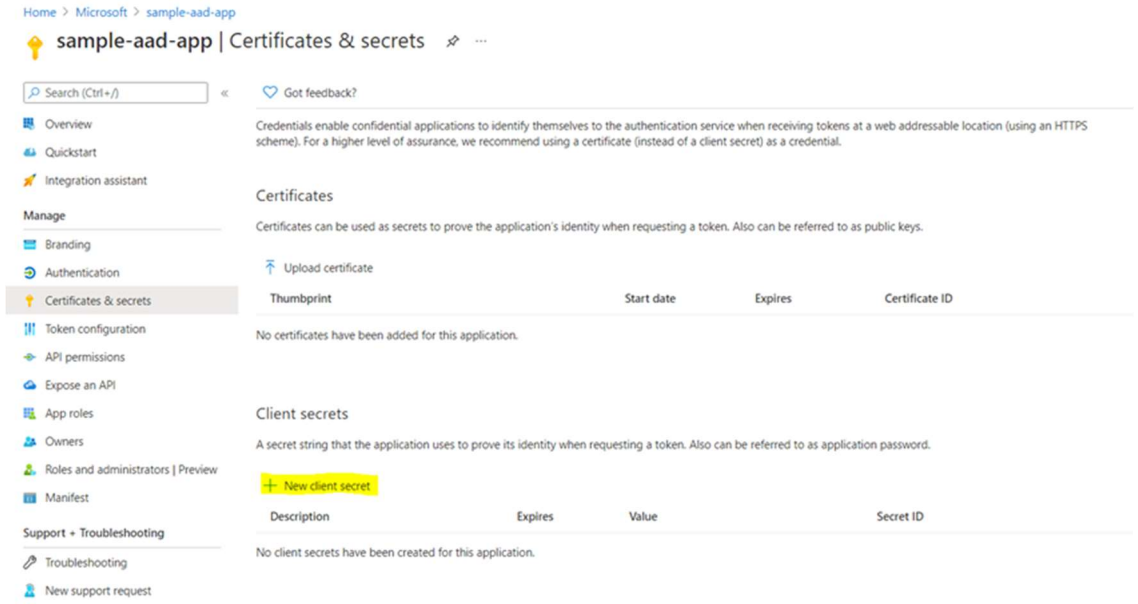
✓ Give the app registration a name, choose an option that suits your needs for account types and click register



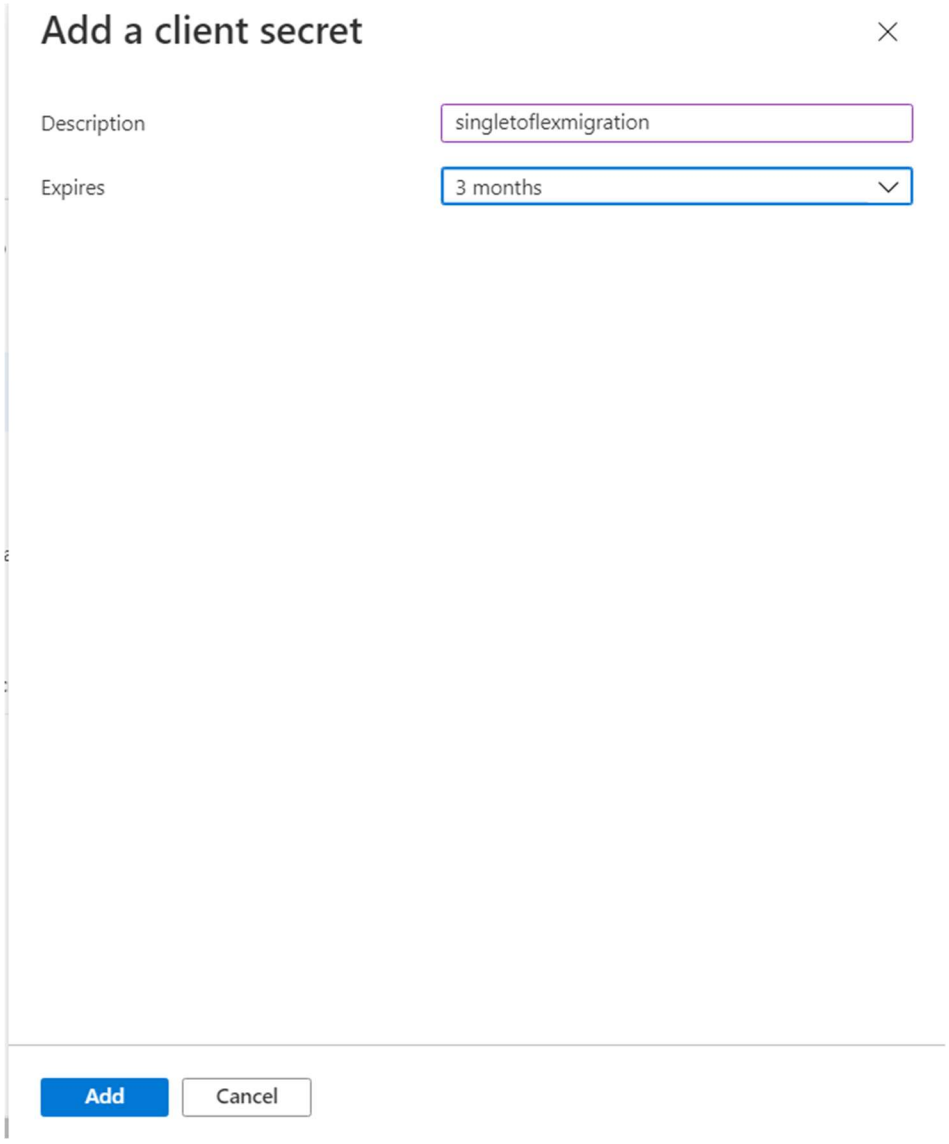
✓ Once the app is created, click on **Add a certificate or secret**.
If executing the migration using the CLI, you can copy the **client ID** and **tenant ID** required for later steps in the migration.



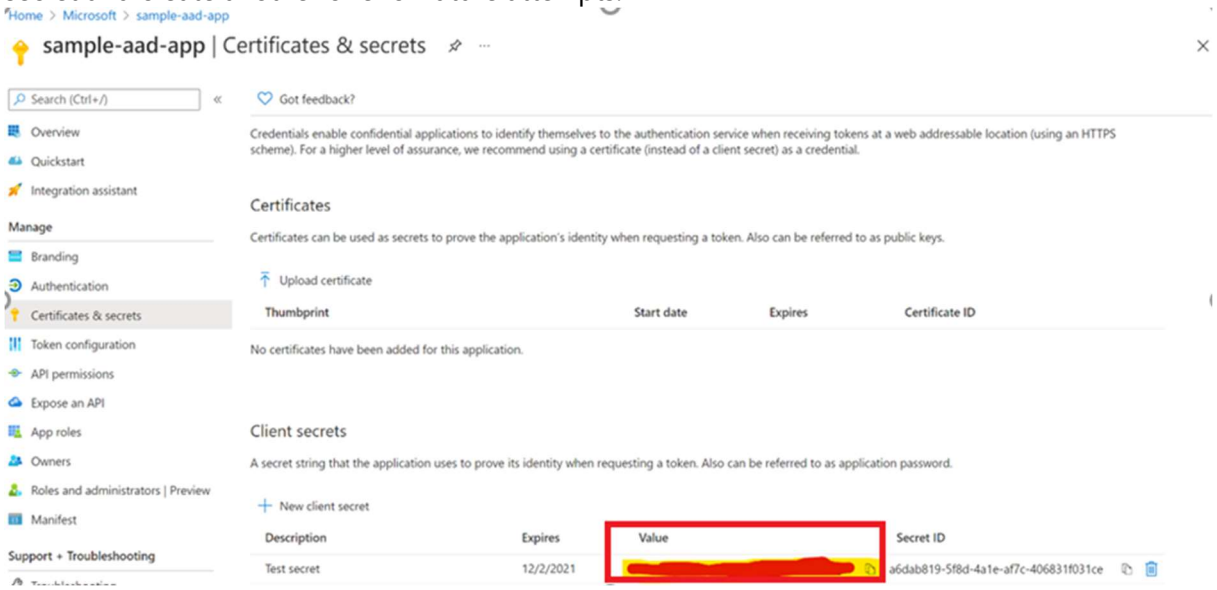
✓ In the next screen, click on **New client secret**.



✓ 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, you may choose three months and click **add**.

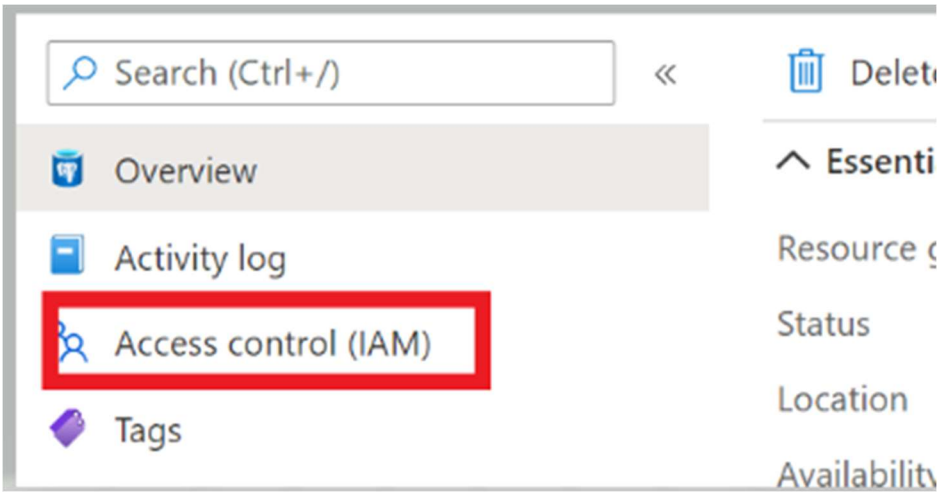


✓ In the next screen, copy the Value column (highlighted in the pic below) which has the details of the Azure Active Directory App secret.
This can be copied only while creation. If you miss copying this secret, you will need to delete this secret and create another one for future attempts.

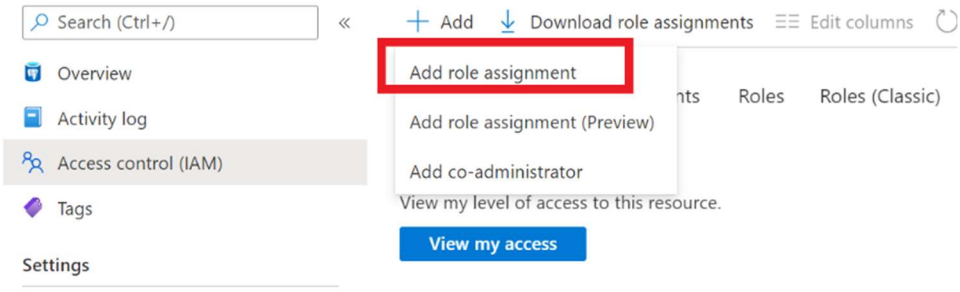


- ✓ Once Azure Active Directory App is created, you will need to add contributor privileges for this Azure Active Directory app to the following resources:
  - **REQUIRED**: Source single server you are migrating from.
  - **REQUIRED**: Target flexible server you are migrating into.
  - **REQUIRED**: 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.

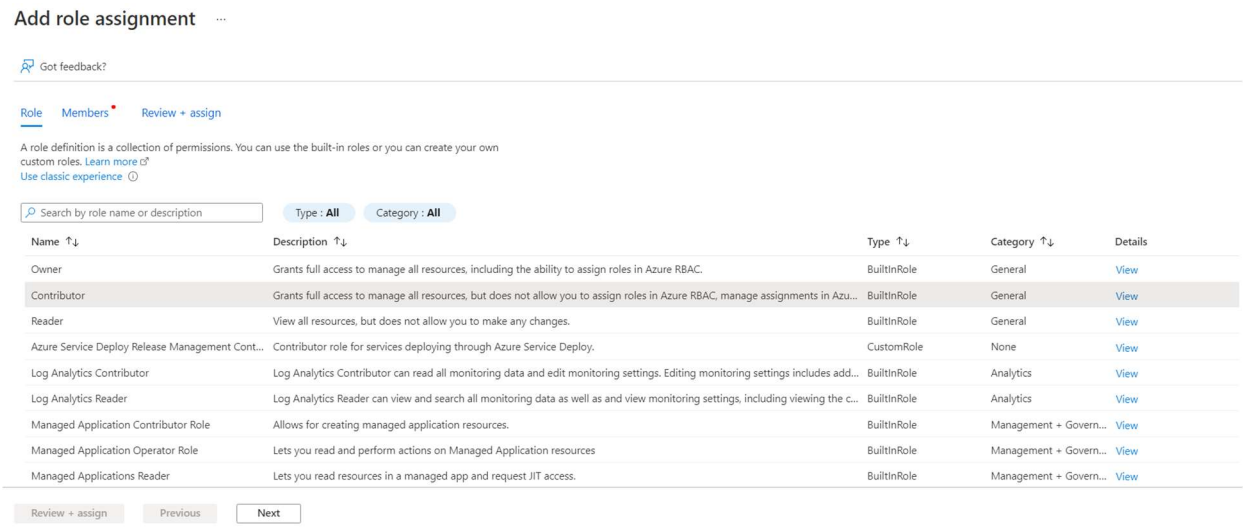


Click **Add** and choose **Add role assignment**.

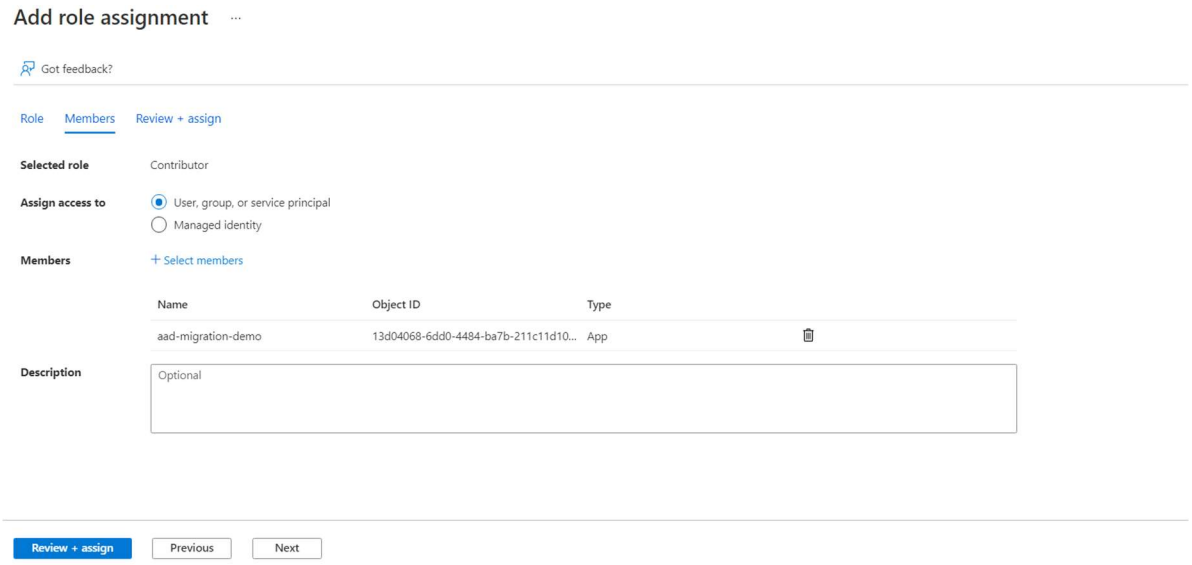


Note: The Add role assignment capability is only enabled for users in the subscription with role type as **Owners**. Users with other roles do not have permission to add role assignments.

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.
Important: Please repeat the steps for adding contributor privileges for the source single server instance and the migration resource group (By default this is the target flexible server resource group).

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

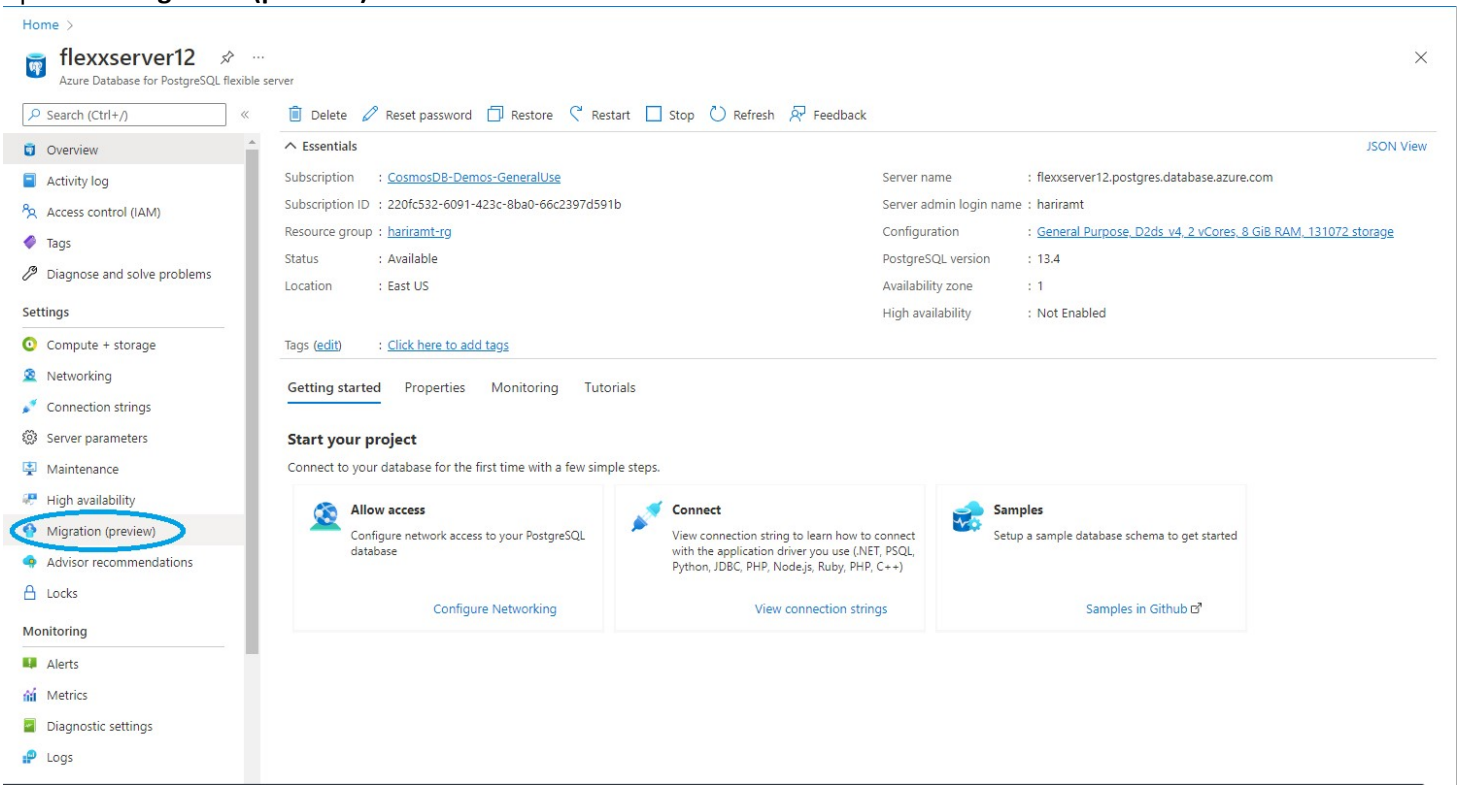
5. Initiate Offline Migration using the Azure Portal

Before we begin, please ensure you have satisfied all the pre-requisites listed in this here, which are necessary to get started with the automated migration service.
The automated migration service comes with a simple, wizard-based portal experience to create a migration from single server to flexible server.

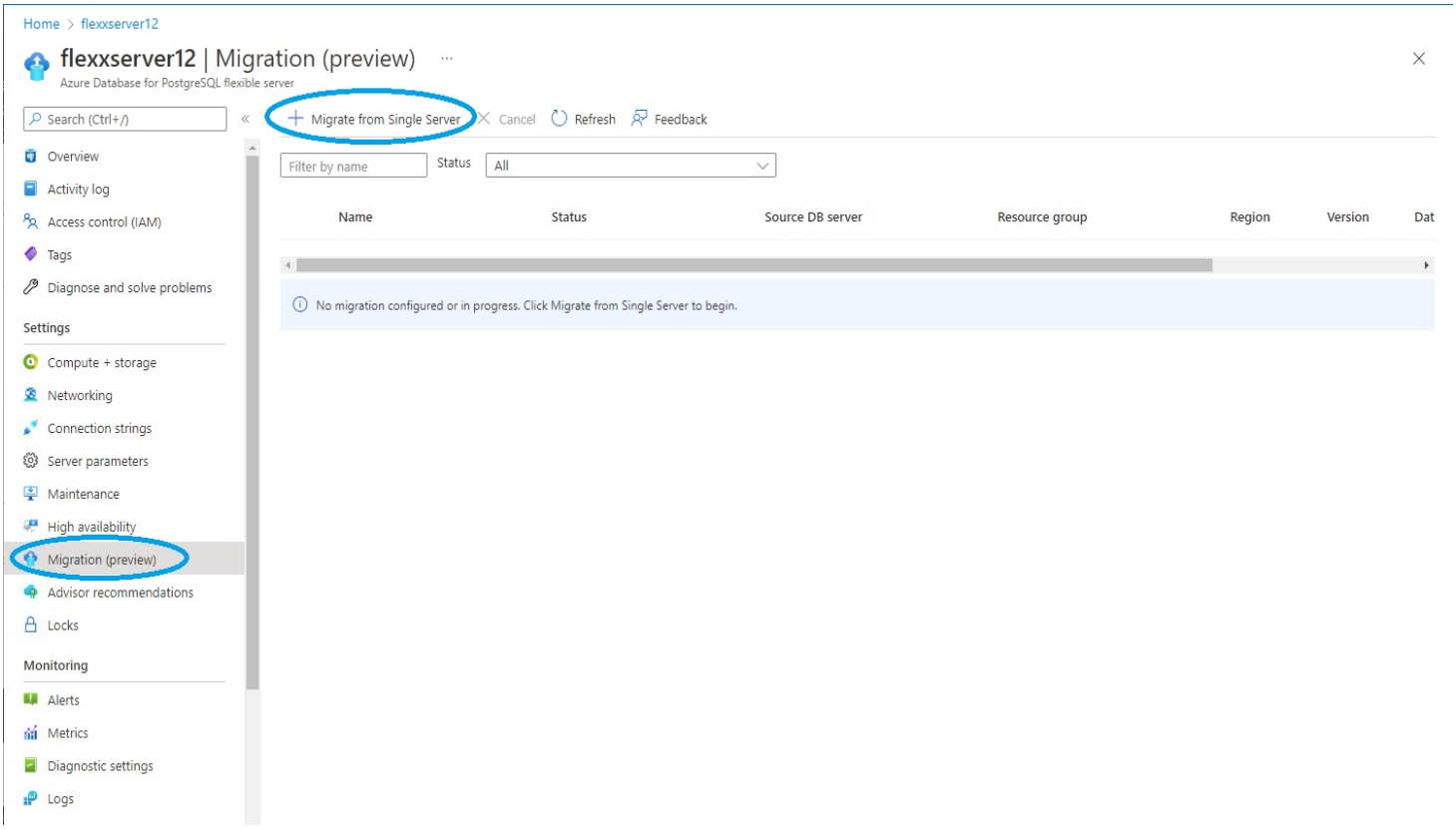
5.1 Sign into the Azure portal

Open your web browser and go to the portal and sign in with your credentials. The default view is your service dashboard.

If you don't have an Azure subscription, create a free Azure account. If you haven't created an Azure database for PostgreSQL flexible server, go ahead and create one using this link.
If you already have a Flexible server instance in place, find and navigate to your flexible server instance.
Once you are in the **Overview** tab of your flexible server, use the left navigation window and scroll down to the option of **Migration (preview)** and click on it.

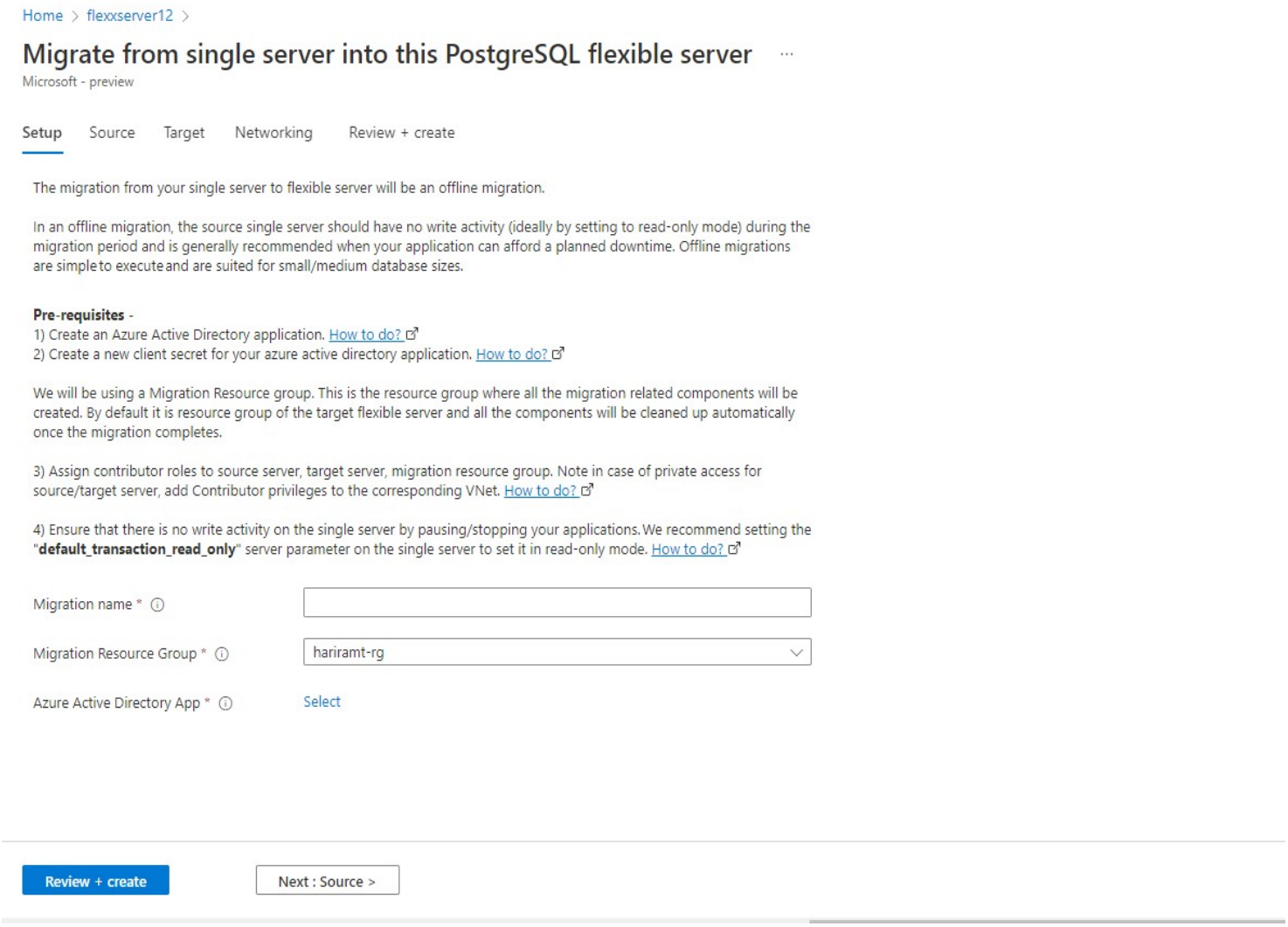


If this is the first time you are using the migration service, you will see an empty grid with a message to click the button **Migrate from Single Server** to start a migration.



If you have already created migrations to your flexible server, you should see the list of migrations that were attempted to this flexible server from single servers.
Click on the **Migrate from Single Server** option. You'll be taken through a wizard-based setup to create a migration to this flexible server from any single server.

5.2 Setup Tab



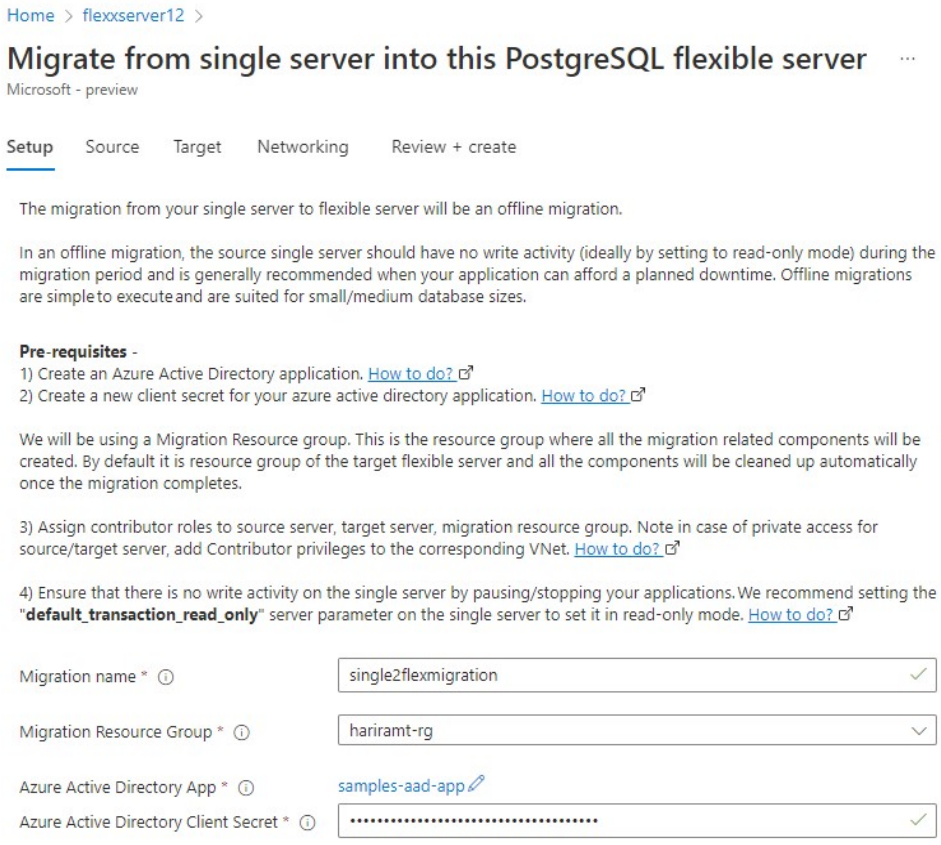
In case the pre-requisites haven't been satisfied yet, please ensure that they are taken care of by following the steps in Section 4.

In the Setup tab,

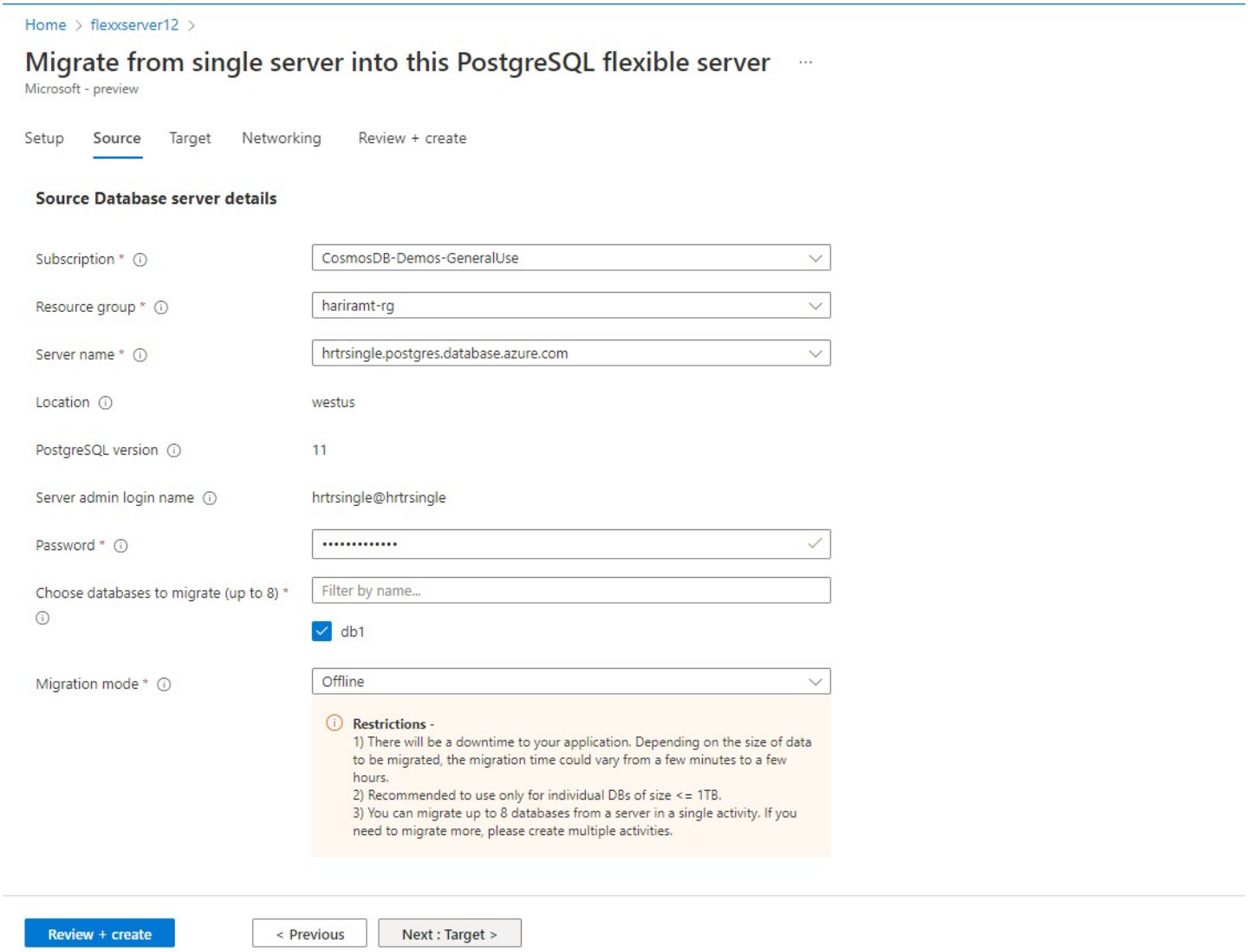
- The **Migration name** field accepts only alphanumeric characters and does not accept any special characters except **'-'**.
  The name can't start with a **'-'** and should be unique for a target server. No 2 migrations to the same flexible server can have the same name.
- The **Migration resource group** is where all the migration-related components will be created by the migration service.
  By default, it's resource group of the target flexible server and all the components will be cleaned up automatically once the migration completes.

If you want to create a temporary resource group for migration-related purposes, create a resource group and select the same from the dropdown.

- For the **Azure Active Directory App**, click the **select** option and pick the app that was created as a part of the pre-requisite step.
  Once the AAD App is chosen, paste the client secret that was generated for the AAD app to the **Azure Active Directory Client Secret** field.



5.3 Source Tab



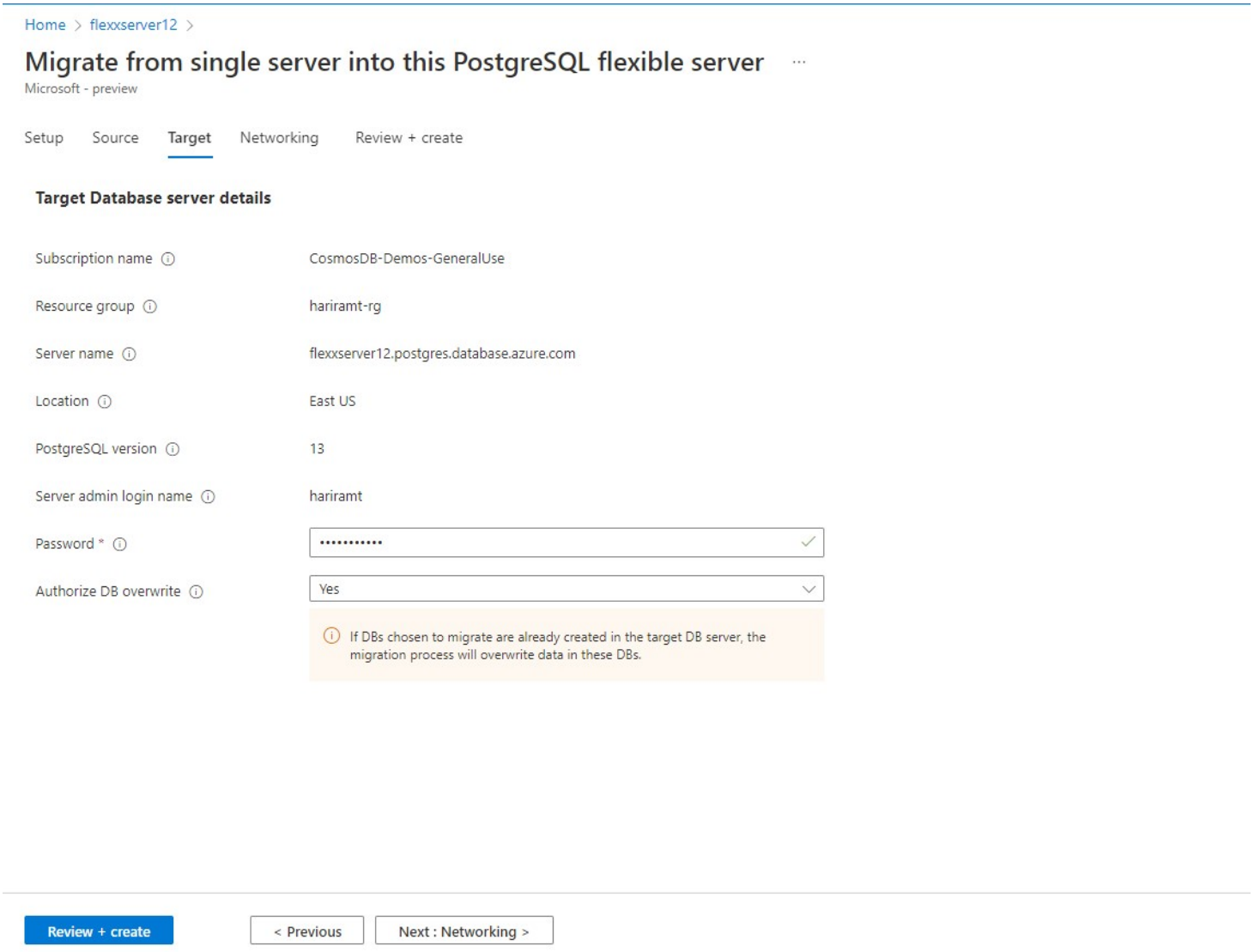
In this tab, you'll be asked to give details related to the source single server.
As soon as you pick the **Subscription** and **Resource Group**, the dropdown for server names will have the list of single servers under that resource group.
Choose any single server from the drop down.

Once the single server is chosen, the fields such as **Location, PostgreSQL version, Server admin login name** are automatically pre-populated.
The server admin login name is the admin username that was used to create the single server.
You should also see the list of databases inside the single server populated as shown in the above pic.

Enter the corresponding password for the admin user and pick the list of databases that you want to migrate.
A max of eight databases can be chosen per migration. Once the list of databases is selected, click on the **Next** Button

Since this is an offline migration, the '**Migration mode**' selection is offline by default. In case you want to use Online migration instead, please contact Microsoft support.

5.4 Target tab



This tab displays metadata of the flexible server like the **Subscription**, **Resource Group**, **Server name**, **Location**, and **PostgreSQL version**.
It displays **server admin login name** which is the username that was used during the creation of the flexible server.

Enter the corresponding password for the admin user.

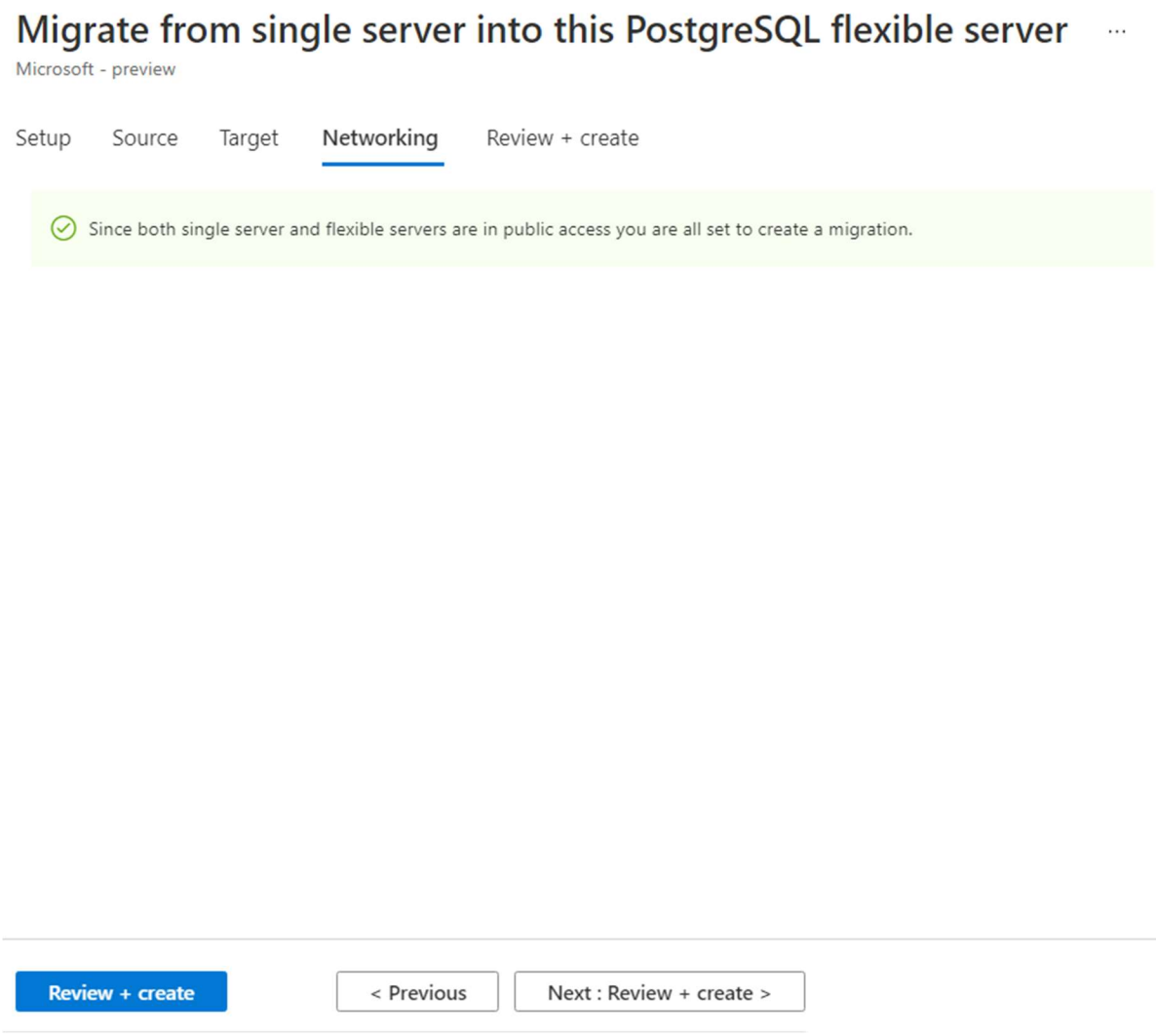
Choose an option **yes/no** for **Authorize DB overwrite**.
If you set the option to **Yes**, you give this migration service permission to overwrite existing data in case when a database that is being migrated to flexible server is already present.
If set to **No**, it goes into a waiting state and asks you for permission either to overwrite the data or to cancel the migration.

Click on the **Next** button

5.5 Networking tab

The content on the Networking tab depends on the networking topology of your source and target servers.

- If both source and target servers are in public access, then you are going to see the below message

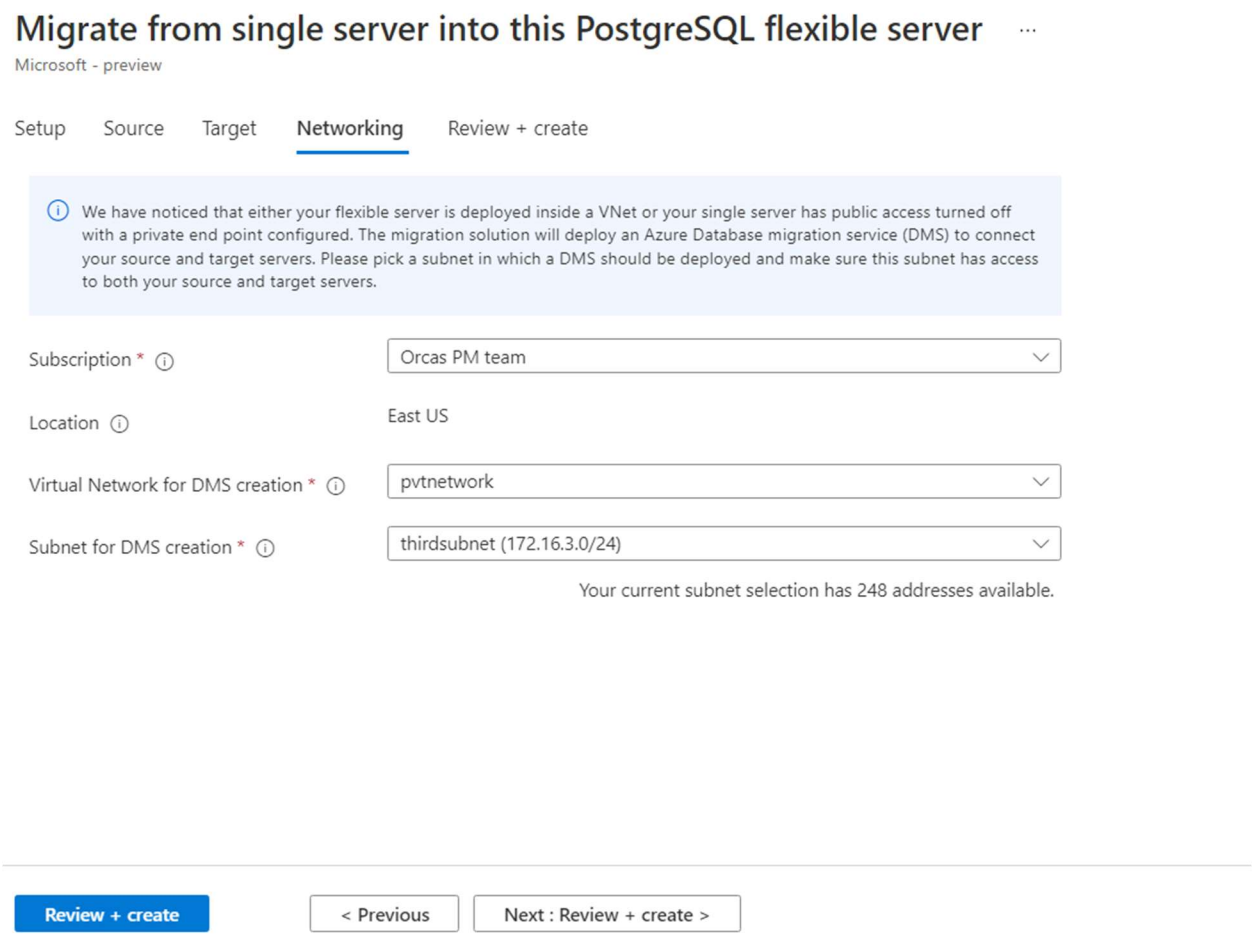


In this case, you need not do anything and can just click on the **Next** button.

If either the source or target server is in private access, then the content of the networking tab is going to be different. Let us try to understand what private access means for a single server and flexible server

- **Single Server Private Access** - **Deny public network access** set to **Yes** and a private end point configured
- **Flexible Server Private Access** - When flexible server is deployed inside a VNet.

If either source or target is private access, then the networking tab looks like the following



All the fields will be automatically pre-populated with subnet details. This is the subnet in which the migration service will deploy Azure DMS to move data between the source and target.

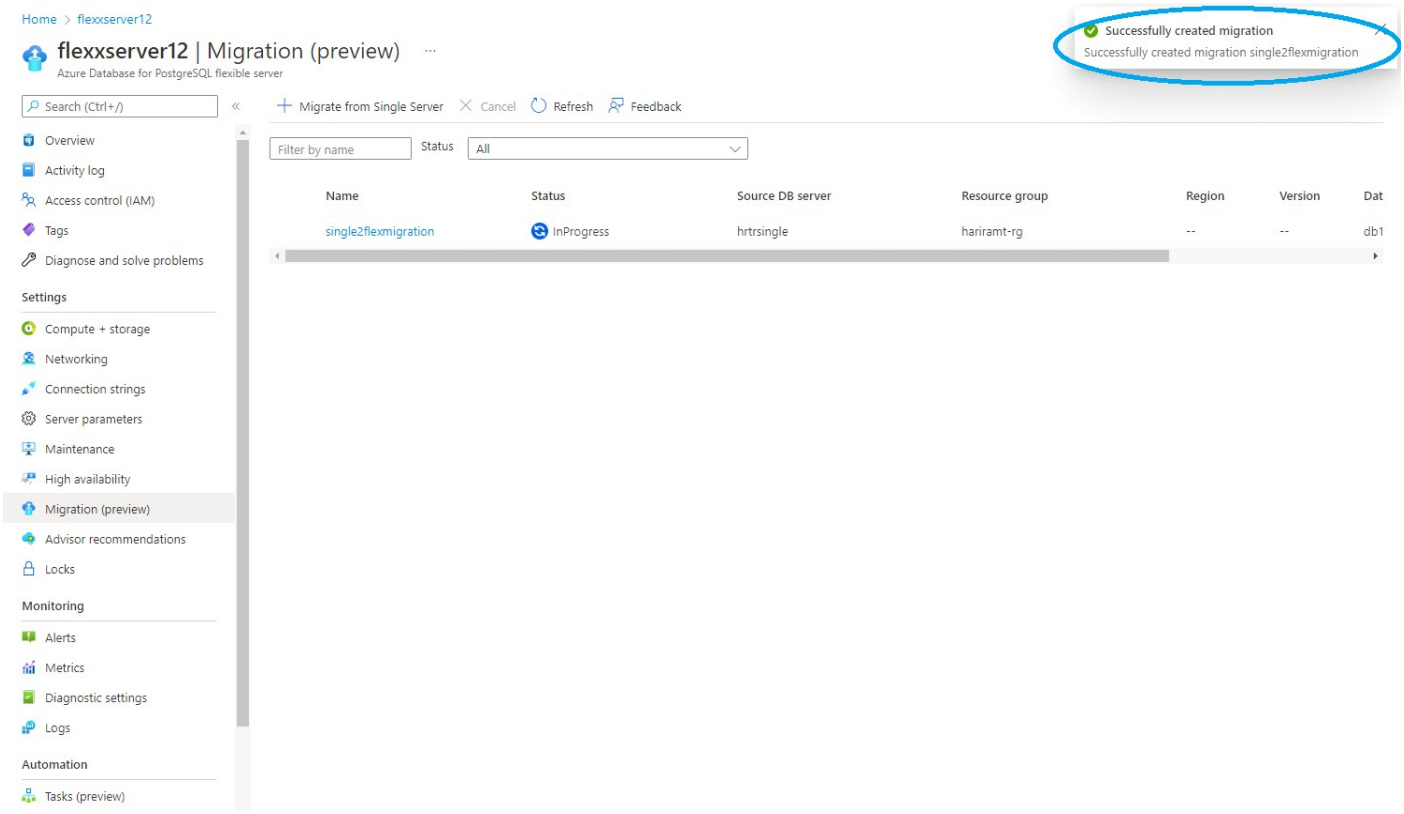
- If source has private access and target has public access, then the pre-populated subnet details would be that of the same subnet for which the private end point is configured.
- If the source has public access and target has private access, then the pre-populated subnet details would be that of the subnet inside the same VNet inside which the flexible server is deployed. It will not be the same subnet as that of flexible server since flexible server needs delegated subnets
- If both source and target are in private access, then the pre-populated subnet details would be that of the subnet inside the same VNet inside which the flexible server is deployed.

You can go ahead with the suggested subnet or choose a different subnet. But make sure that the selected subnet can connect to both the source and target servers.

After picking a subnet, click on **Next** button

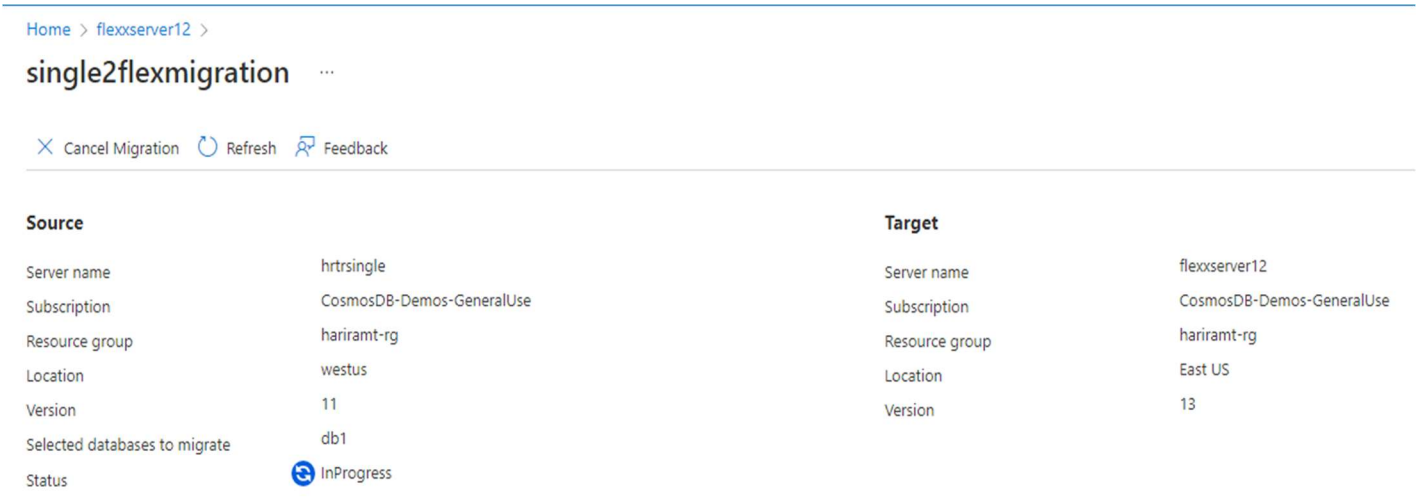
5.6 Review + Create tab

This tab gives a summary of all the details given by the user for creating the migration. Review the details and click on the **Create** button.
After clicking on the **Create** button, you should see a notification saying the migration was successfully created in a few seconds.
You should be automatically redirected to **Migrations Preview** page of flexible server that will have a new entry of the recently created migration.



The grid displaying the migrations has various columns including **Name**, **Status**, **Source server name**, **Region**, **Version**, **Database names**, and the **Migration start time**.
By default, the grid shows the list of migrations in the decreasing order of migration start time. In other words, the more recently created migrations appear on top of the grid.

You can use the refresh button to refresh the status of the migrations. You can click on the migration name in the grid to see the details of that migration.

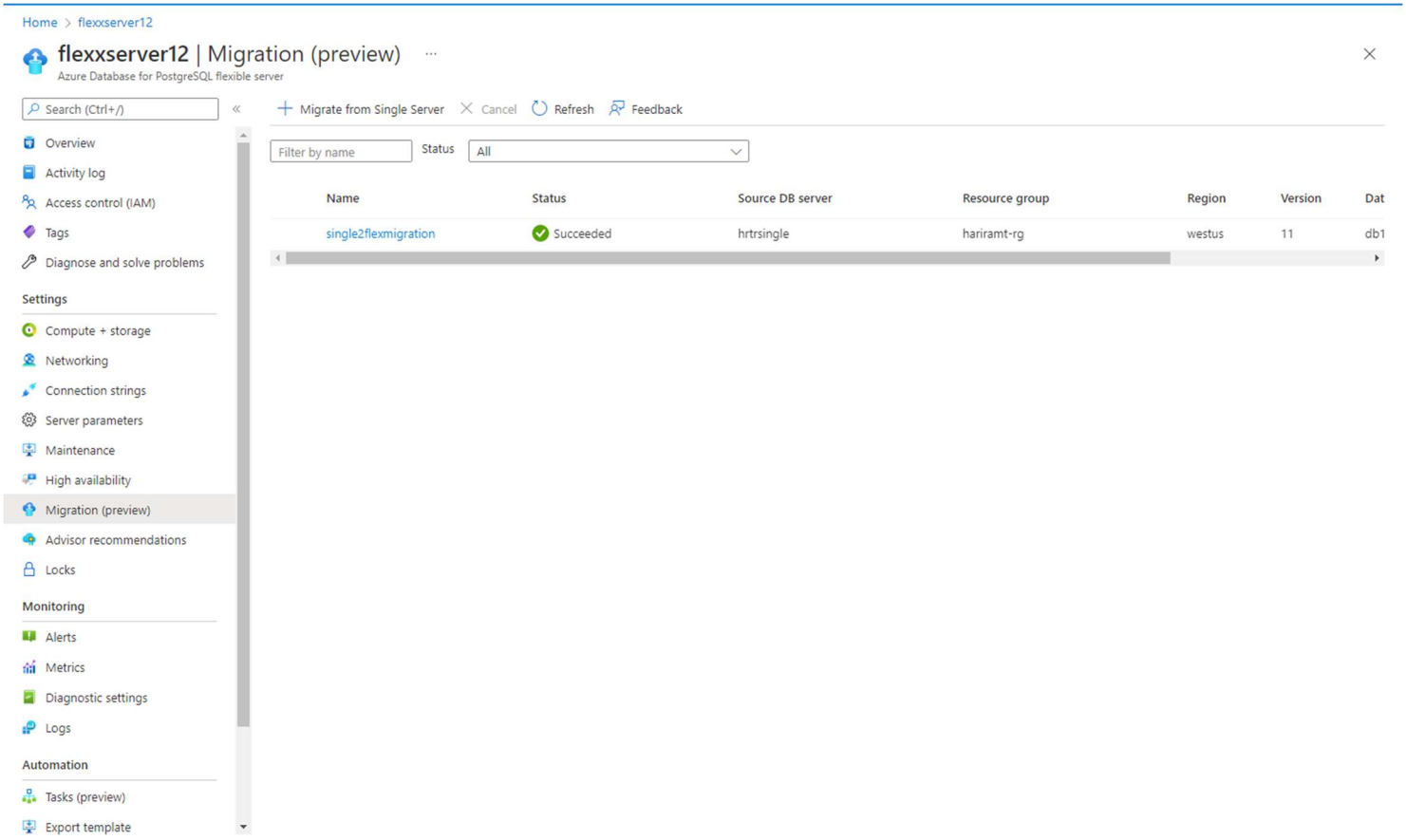


As soon as a migration is created, it moves into the state of **InProgress**. During this state, a bunch of pre-requisites steps will be carried out by the migration service.
This includes creating an Azure DMS, ensuring DMS has access to both servers by adding its IP to the list of firewall rules and a few other maintenance tasks.
In general, it takes around 8 to 10 minutes for the migration to move from the **InProgress** state to **Migrating Data** state.

The total time taken for the migration will depend on the size of the database. The table below gives a rough estimate for various sizes.

Database Size	Approximate Time Taken (HH:MM:SS)
1 GB	00:01:01
5 GB	00:04:20
10 GB	00:08:29
50 GB	00:46:01
100 GB	06:11:36
500 GB	09:28:13
1000 GB	09:25:54

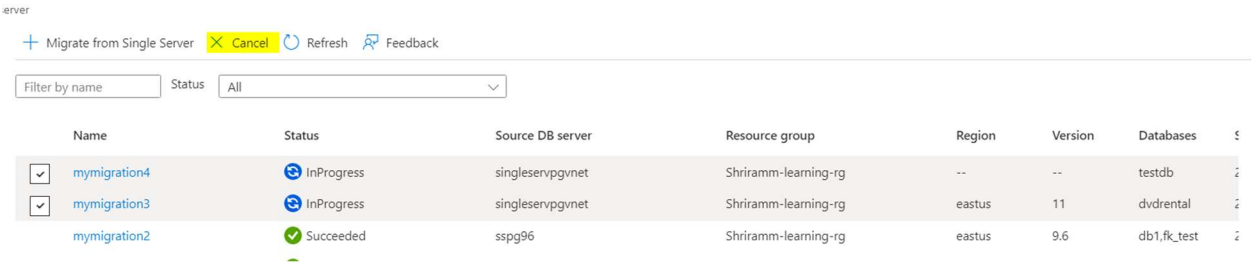
Once the Offline migration is successfully complete, the migration moves to 'Succeeded' state.



After verifying the data in your target server, you can make changes to your application and point to the flexible server.

You also have the option to cancel any ongoing migrations. To cancel a migration, it must be in an **InProgress** state. You can't cancel a migration that has either **Succeeded** or **Failed**.

You can choose multiple ongoing migrations at once and can cancel them.



Please note that '**cancel migration'** stops only further migration activity on your target server.
It will not drop or roll back any changes on your target server that is already done by the migration attempt.

6. 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 concludes.
- Post data validations and making your application point to flexible server, you can consider deleting your single server.