Migrate PostgreSQL Single Server to Flexible Server using Azure portal

Contents

This article shows you how to create a migration from your Azure database for PostgreSQL single server to flexible server using our automated migration service through Azure portal.

Pre-requisites

Make sure to take care of the pre-requisites listed in this document, which are necessary to get started with the automated migration service.

Portal Experience

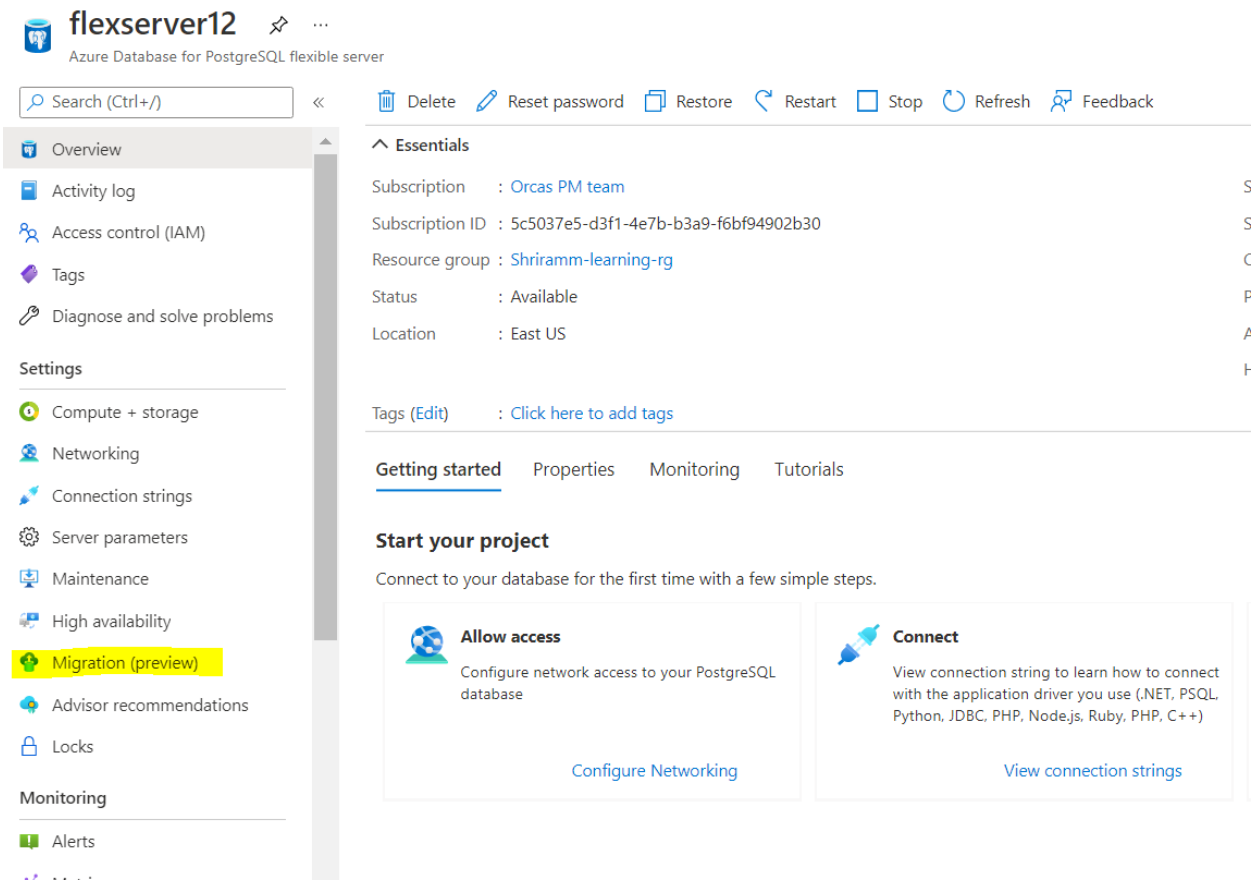
The automated migration service comes with a simple, wizard-based portal experience to create a migration from single server to flexible server.

Sign into the Azure portal

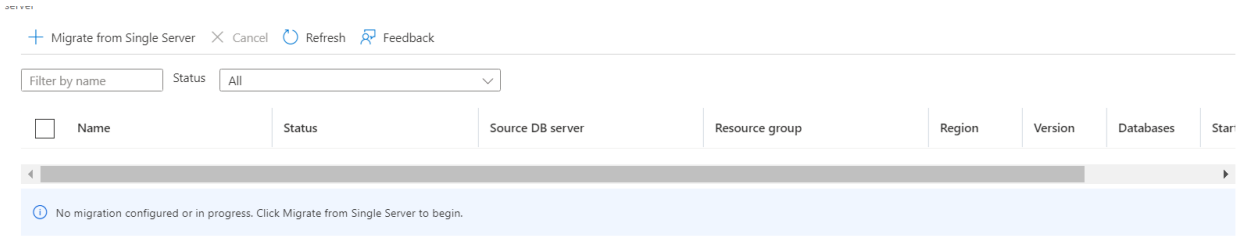
Open your web browser and go to the portal. Enter your credentials to sign into the portal. The default view is your service dashboard. If you don't have an Azure subscription, create a free Azure account

If you haven't created an Azure database for PostgreSQL flexible server, go ahead and create one using this link. Else, 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.



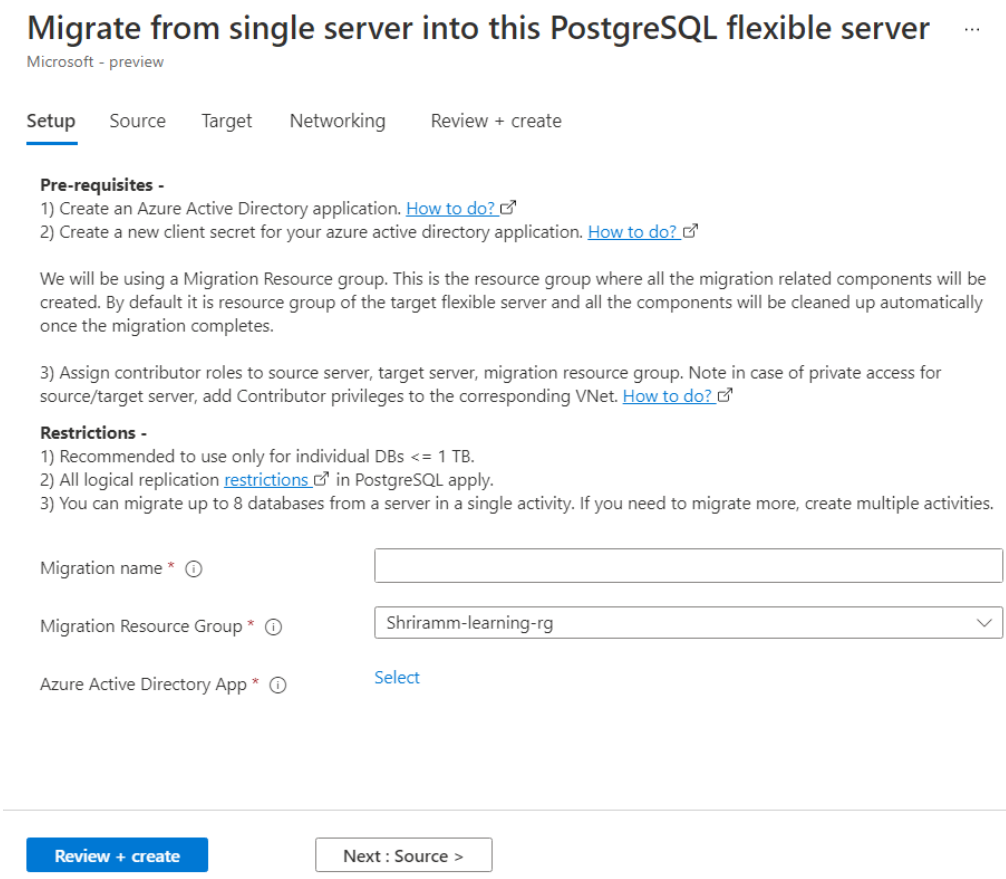
If this is the first time you are using the migration service, you will see an empty grid with a message to click the button **Migrate from Single Server** to start a migration.



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.

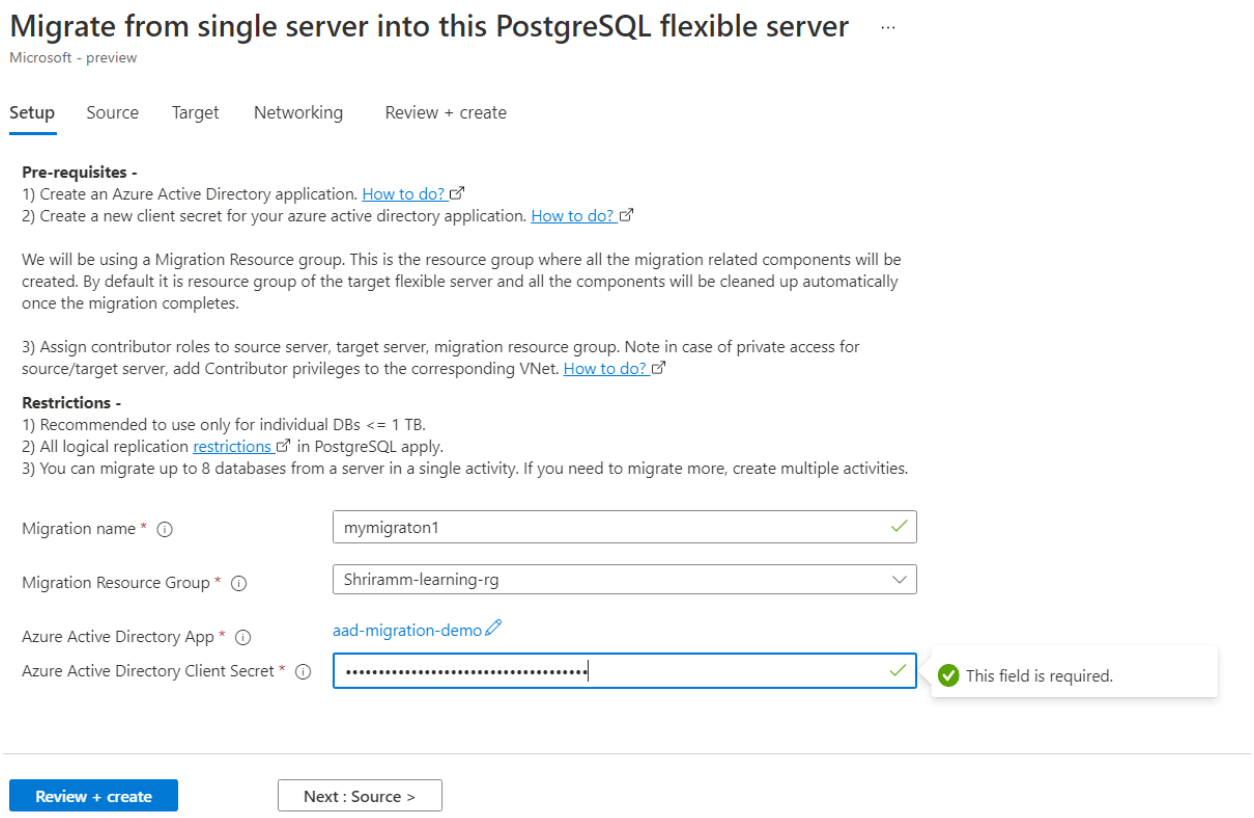
Click on the **Migrate from Single Server** option. You'll be taken through a wizard-based setup to create a migration to this flexible server from any single server.

Setup Tab



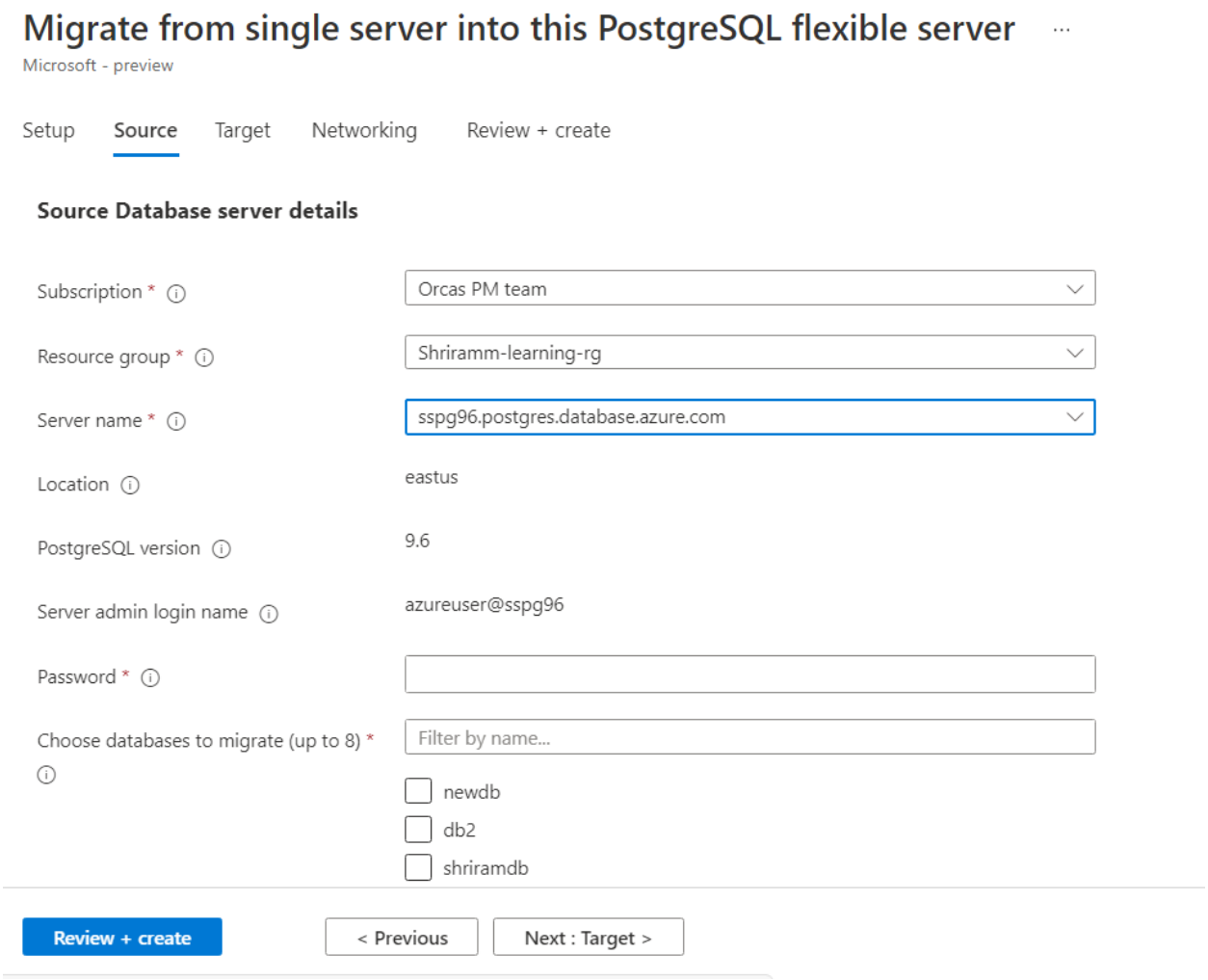
There is a **pre-requisites** section, the details of which are documented in detail here. There is also a **restriction** section that lists down the restrictions that are applicable to this migration service

- 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.



Click on the **Next** button.

Source tab

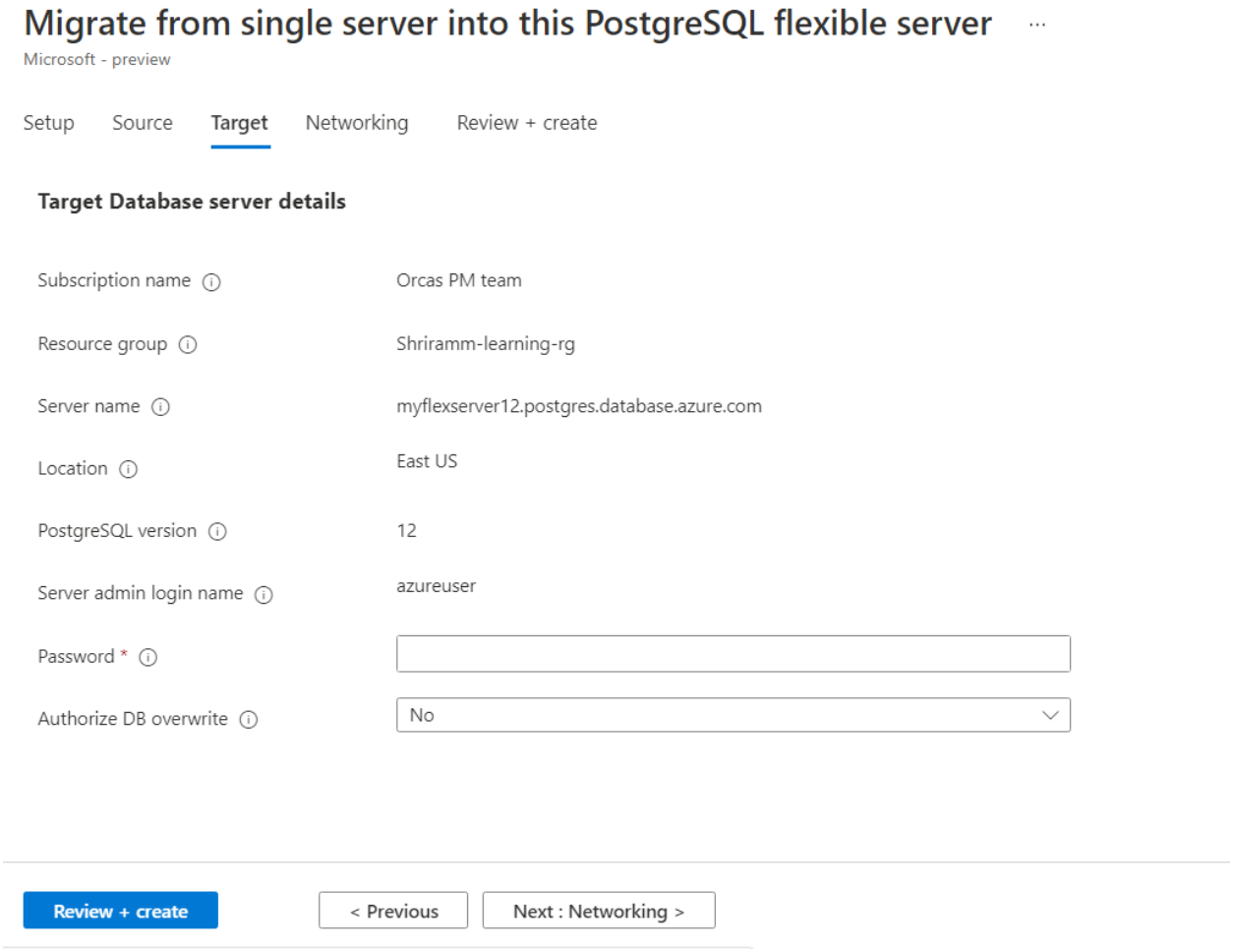


In this tab, you'll be asked to give details related to the source single server. As soon as you pick the **Subscription** and **Resource Group**, the dropdown for server names will have the list of single servers under that resource group. Choose any single server from the drop down.

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

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.

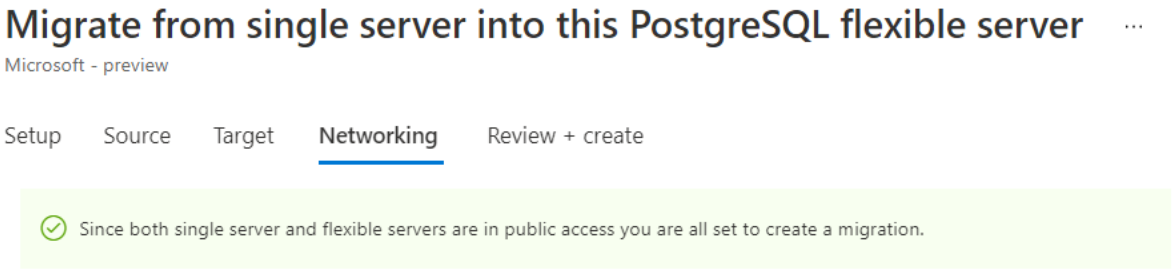
Choose an option **yes/no** for **Authorize DB overwrite**. If you set the option to **Yes**, you give this migration service permission to overwrite existing data in case when a database that is being migrated to flexible server is already present. If set to **No**, it goes into a waiting state and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

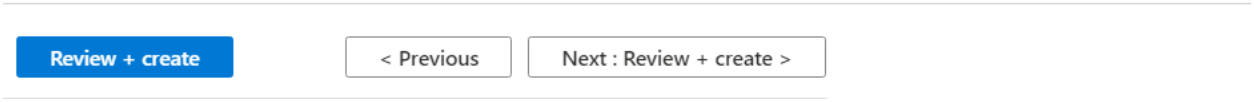
Click on the **Next** button

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

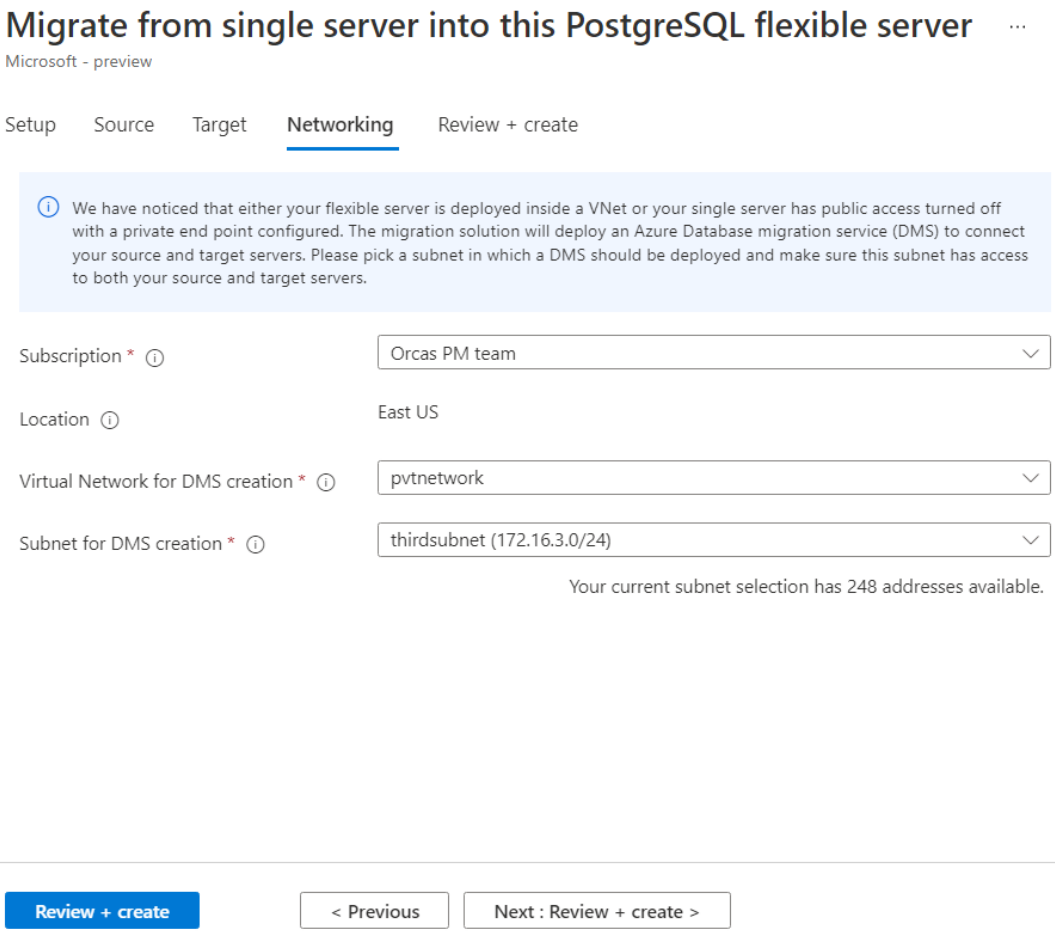




In this case, you need not do anything and can just click on **Next** button.

- 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 does private access mean for 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 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.

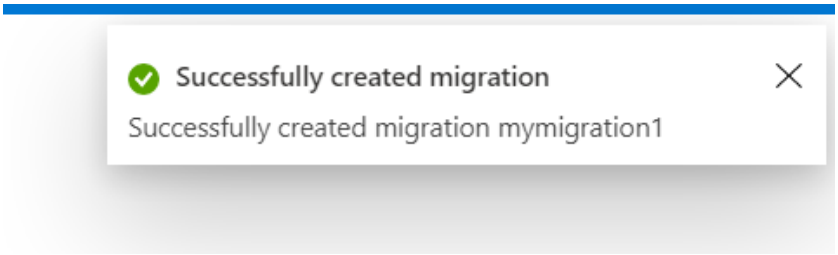
You can go ahead with the suggested subnet or choose a different subnet. But make sure that the selected subnet can connect to both the source and target servers.

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

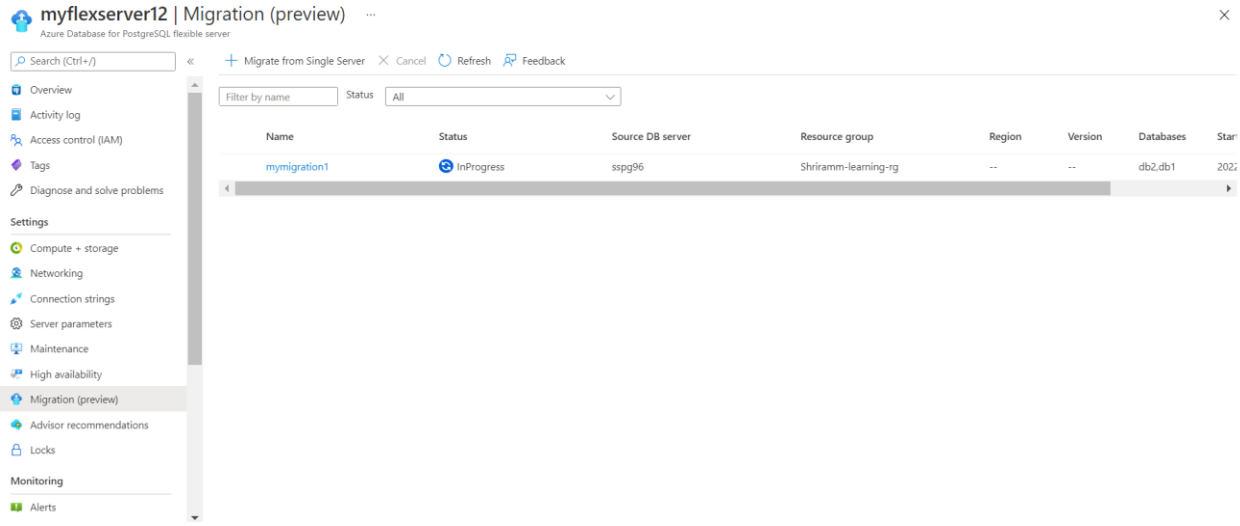
Review + Create tab

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



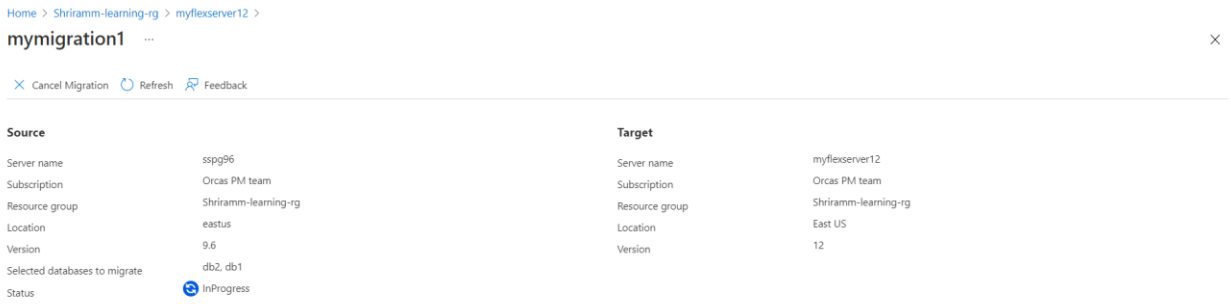
You should be automatically redirected to **Migrations Preview** page of flexible server that will have a new entry of the recently created migration



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.

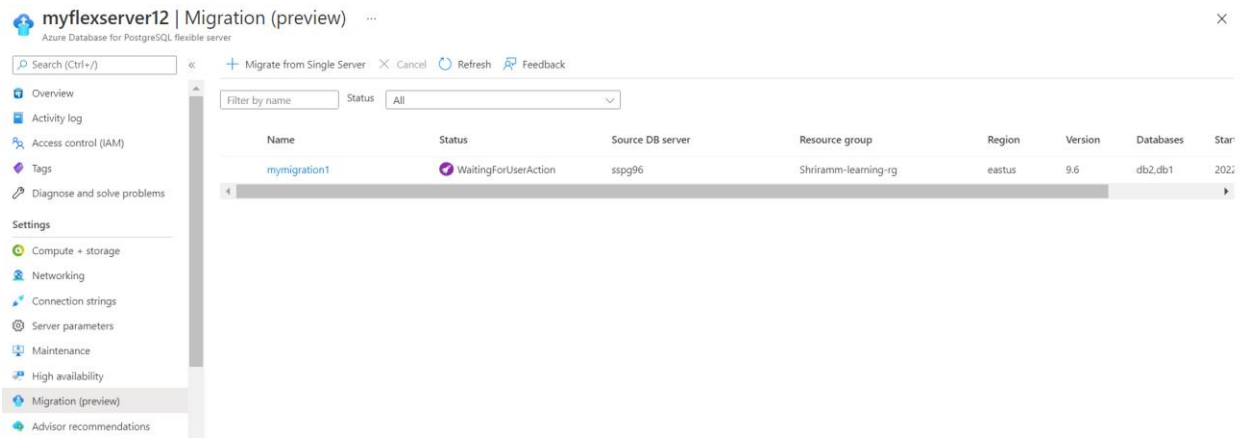
You can use the refresh button to refresh the status of the migrations.

You can click on the migration name in the grid to see the details of that migration.

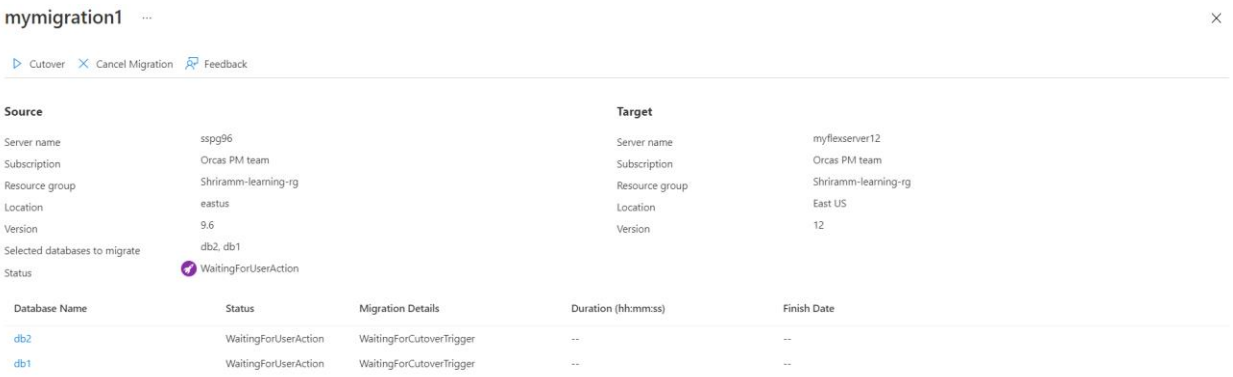


As soon as a migration is created, it moves into the state of **InProgress**. During this state, a bunch of pre-requisites steps will be carried out by the migration service. This includes creating an Azure DMS, ensuring DMS has access to both servers by adding its IP to the list of firewall rules and 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.

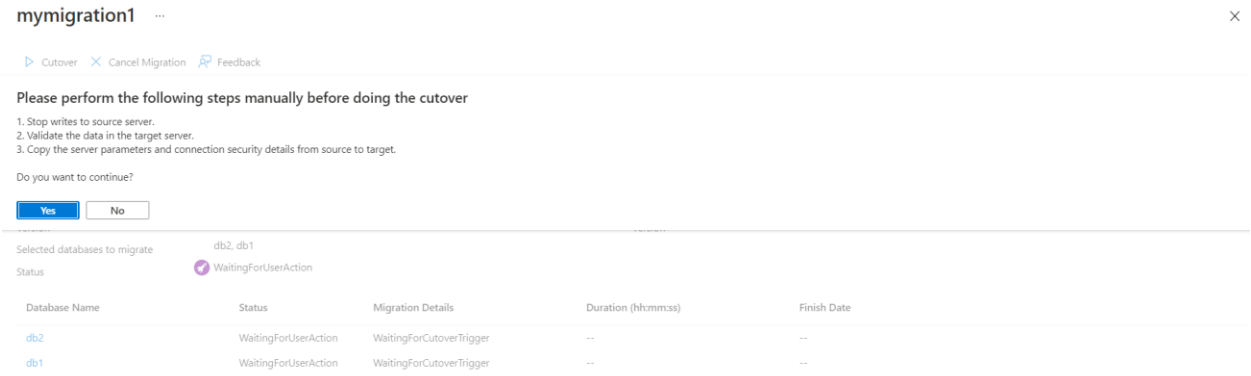
Migrating Data is the state where the dump and restore of schema happens followed by movement of data from source to target server. Once this is completed, the migration moves to the state of **WaitingForUserAction** and substate of **WaitingForCutOver**.



You can click on the migration name to go into the migration details page and should see the substate of **WaitingForCutover**.

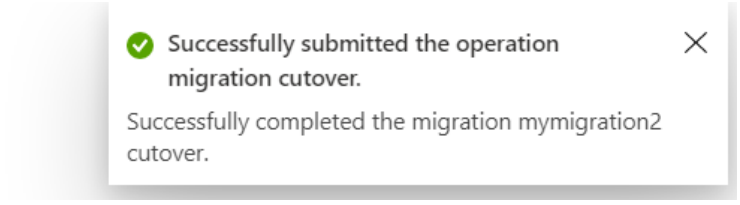


The migration moves into the substate **WaitingForCutOver** once it has finished the dump and restore of the databases of your single server. At this stage, the ongoing writes at your source server will be replicated to the target flexible server using the logical replication feature of PostgreSQL. Click on the **Cutover** button and you should see the following message



Make sure to stop writes on your server and click on the **Yes** button to start cutover.

In a few seconds after starting cutover, you should see the following notification

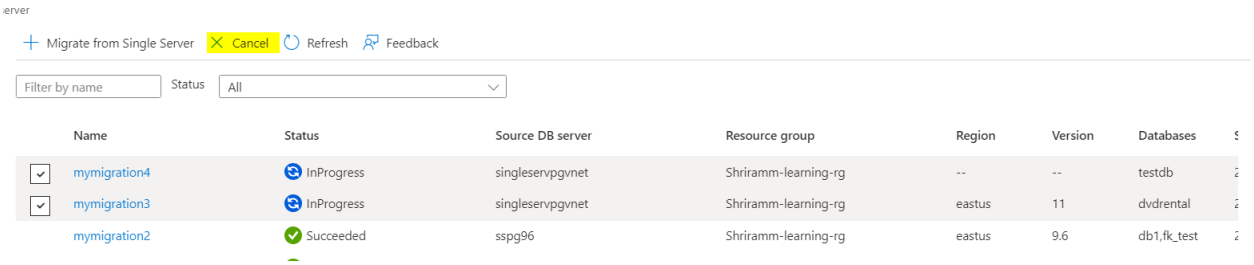


Once the cutover is complete, the migration moves to **Succeeded** state and migration of schema data from your single server to flexible server is now complete. You can use the refresh button in the page to check if the cutover was successful.

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

You also have the option to cancel any ongoing migrations. For a migration to be canceled, it has to be in **InProgress** or **WaitingForUserAction** 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.



Note that **cancel migration** just stops any more further migration activity on your target server. It will not drop or roll back any changes on your target server that were done by the migration attempts.

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.
- 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.