## Introduction

The Single to Flexible server migration tool is designed to move databases from Azure Database for PostgreSQL Single server to Flexible server. We launched the preview version of the tool also known as the **Migration (Preview) feature** in Flexible server, back in August 2022 for customers to try out migrations. The preview version used an infrastructure based on Azure Database Migration Service (DMS). We have aggregated feedback from customers and have come up with an improved version of the migration tool which will use a new infrastructure to perform migrations in a more easy, robust and resilient way. This document will help you understand the functioning of the new version of the tool along with its limitations and the pre-requisites that need to be taken care of before using the tool.

Currently, the new version of the tool **only supports Offline mode** of migration. Offline mode presents a simple and hassle-free way to migrate your databases but might incur downtime to your server depending on the size of the databases and the way in which data is distributed across tables in your database.

How does it work?

The new version of the migration tool is a hosted solution where we spin up a purpose-built docker container in the target Flexible server VM and drive the incoming migrations. This docker container will be spun up on-demand when a migration is initiated from a single server and will be decommissioned as soon as the migration is completed. The migration container will use a new binary called **pgcopydb** which provides a fast and efficient way of copying databases from one server to another. Though pgcopydb uses the traditional pg_dump and pg_restore for schema migration, it implements its own data migration mechanism which involves multi-process streaming parts from source to target. Also, pgcopydb bypasses pg_restore way of index building and drives that internally in a way that all indexes can be built concurrently. So, the data migration process is much quicker with pgcopydb. Following is the process diagram of the new version of the migration tool.
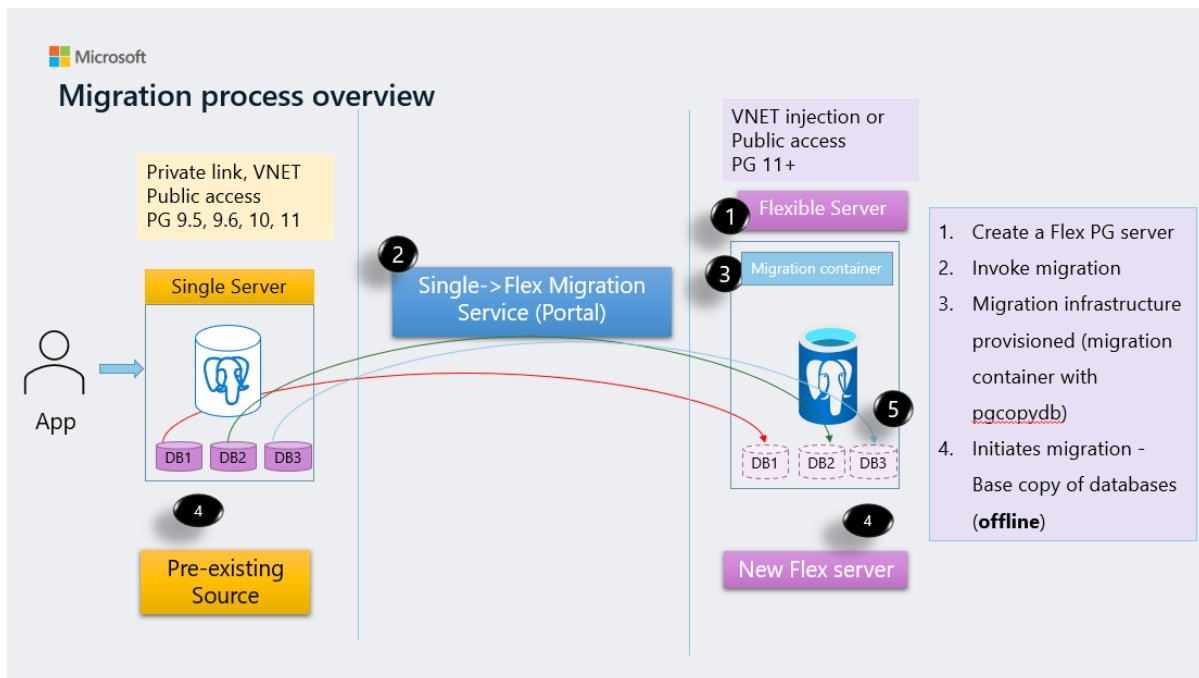
The following table shows the approximate time for performing offline migrations for databases of various sizes using the new version of the tool. Note that the migration was performed on the **General-Purpose** Flexible server with **Standard_D4s_v3 SKU (4VCore)**, **4TB Storage and 6400 IOPs**.

Database Size	Approximate Time Taken (HH:MM: SS)
1GB	00:01:00
5GB	00:03:00
10GB	00:08:00
100GB	01:45:00
500GB	
1000GB	

The below table has data on the time taken to perform offline migration with General-Purpose flexible server with **Standard_D8s_v3 SKU (8VCore), 4TB storage and 7500 IOPs.**

Database Size	Approximate Time Taken (HH:MM: SS)
1GB	00:00:30
5GB	00:02:00
10GB	00:04:18
100GB	01:15:00
500GB	
1000GB	

Please note that these numbers are the average time taken to complete offline migration for a database with a given size with multiple different schemas and with varying data distribution.

From the above data, it is very clear that with a higher compute on Flexible server, the migration will complete faster. It is a good practice to create a flexible server with a higher SKU and complete the migrations from single server in a quick way. Post successful migrations, you can tune the SKU of your flexible server to meet the application requirements.

Limitations

- You can have only one active migration to your flexible server.
- Only a max of 8 databases can be included in one migration attempt from single to flexible server. If you have more than 8 databases, you must wait for the first migration to be completed before initiating another migration for the rest of the databases. Support for migration of more than 8 databases will be introduced in a later version of the tool.
- Only **offline** mode is supported as of now. Online mode will be introduced later in a later version of migration tool.
- The source and target server must be in the same Azure region. Cross region migrations will not work in this version of the migration tool.
- Collation mismatches between single and flexible server are not handled by the tool. If there is a collation used in single server that does not exist in flexible server, the migration will fail.
- The migration tool does not migrate users and roles. You should consider creating users and roles in your flexible server after the completion of data migration.

Best Practices for offline migration

- Offline migrations will require a downtime of your application. So, plan for a time slot when the application can take a maintenance window.
- Before initiating the migration, stop all the applications that connect to your single server.
- Create a flexible server with a General-Purpose or Memory Optimized compute tier. Avoid Burstable SKUs since it might take longer time to complete the migration due to non-availability of dedicated CPUs.
- Pick a higher SKU size for flexible server. Go for a minimum of 8VCore or higher to complete the offline migration quickly.
- Do not include HA or geo redundancy option while creating flexible server. You can always enable it with zero downtime once the migration from single server is completed.
- Once the migration is completed, please verify the data on your flexible server and make sure it is an exact copy of the single server.
- Post verification, enable HA/ backup options as needed on your flexible server.
- Change the SKU of the flexible server to match the application needs. Note that this change will need a database server restart.
- Migrate users and roles from single to flexible servers. This can be done in a manual way by creating users on flexible servers and providing them with suitable privileges or by using this Python script which automates the process.
- Make changes to your application to point the connection strings to flexible server.
- Monitor the database performance closely to see if performance tuning is required.

How can customers consume it?

The new version of the migration tool can be consumed from Azure Portal. Once you share your subscription details, we will enable the new version of the migration tool on your subscription.

Pre-requisites

Here is the list of pre-requisites to get started with the migration tool.

- Create a flexible server with higher SKU for migration purposes.
- Enable all extensions used in single server on flexible server.
- Enable network connectivity from target flexible server to single server.
    - If both single and flexible servers are public access, there is no action needed from your end to establish connectivity. The migration tool automatically allow-lists the IP of flexible server in single server.
    - If single server is public access and flexible server is inside a VNet, please allow connections from the VNet of flexible server to your single server by adding a VNet rule or service end point.
    - If the single server is behind a private end point and flexible server is public access, there is no way to establish connectivity between the servers. The migrations will fail, and this scenario is not supported by the new version of the migration tool.
    - If the single server is behind a private end point and flexible server is inside a VNet, ensure the VNets associated with private end point and flexible server are connected. This will involve VNet peering if VNets of source and target are different. If both are in the same VNet, make sure there is no network security group (NSG) rule that is blocking the connectivity between the flexible server and single server.

Portal Experience

Once the pre-requisite steps are taken care of, you can perform the following steps to create an offline migration using the new version of the tool.

- Go to your Azure Database for PostgreSQL Flexible Server. Scroll down to **Migration (preview)** option and select it.
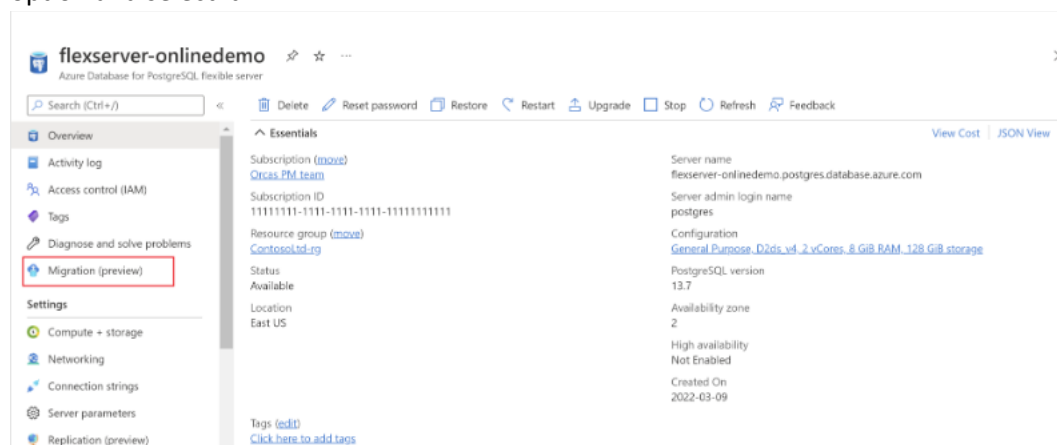
- Select the **Migrate from Single Server** button to start a migration from Single Server to Flexible Server.
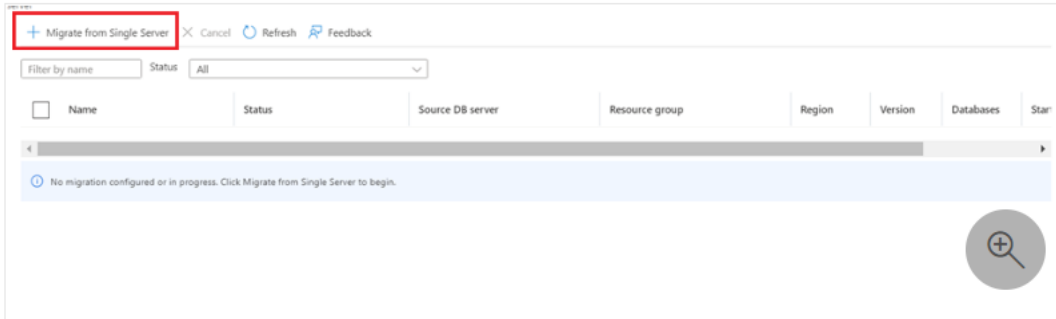
- This opens a wizard with a series of tabs. First is the **setup** tab which prompts for a migration name. Please ensure that all extensions used in single server are enabled on the flexible server.

Enter a valid migration name and select **Next** button.

- The next is the **Source** tab.

Migrate from single server into this PostgreSQL flexible server
Microsoft - preview

Setup   Source   Target   Review + create

Source Database server details

Subscription * ⓘ	Orcas PM team
Resource group * ⓘ	Shriramm-learning-rg
Server name * ⓘ	canary-test-sspg.postgres.database.azure.com
Location ⓘ	eastus2euap
PostgreSQL version ⓘ	11
Server admin login name ⓘ	azureuser@canary-test-sspg
Password * ⓘ	•••••••• ✓
Choose databases to migrate (up to 8) * ⓘ	Filter by name...

☐ postgres
☑ mydatabase1
☑ dvdrental
☑ adventureworks

[Review + create]   [< Previous]   [Next : Target >]

Pick the subscription and resource group of your single server and provide the password for the admin user of your single server. Pick up to a maximum of 8 databases for the migration and click on the **Next** button.

- The next is the **Target** tab.

Migrate from single server into this PostgreSQL flexible server
Microsoft - preview

Setup   Source   Target   Review + create

Target Database server details

Subscription * ⓘ	Orcas PM team
Resource group * ⓘ	Shriramm-learning-rg
Server name * ⓘ	canary-test-14.postgres.database.azure.com
Location ⓘ	eastus2euap
PostgreSQL version ⓘ	14
Server admin login name ⓘ	azureuser
Password * ⓘ	•••••••• ✓
Authorize DB overwrite ⓘ	No

[Review + create]   [< Previous]   [Next : Review + create >]

Please provide the password for the admin user of flexible server and the rest of the details are already filled out. Click on the **Next** button.

- The final tab is **Review and Create.**

Review the information and click on the **Create** button.

Monitoring Migrations

After you select the Create button, a notification appears in a few seconds to say that the migration was successfully created. You should automatically be redirected to **the Migration (Preview)** page of Flexible Server. It should have a new entry for the recently created migration.

You can also select the migration name in the grid to see the details of that migration.

Use the refresh button to get the latest status of the migration. Over time, the migration will succeed or fail with appropriate errors.

Support

Please reach out to us at AskAzureDBforPGS2F@microsoft.com for any kind of support needed on migrations from single server to flexible server.