Major Version Upgrade in Private Preview

Azure Database for PostgreSQL – Flexible Server

Doc version: 1.0 – 6/27/22

Updated 1.1 -10/31/2022

This private preview is offered to select customers under NDA.

Overview

Flexible server PostgreSQL supports Postgres versions 11, 12,13, and 14.

Postgres releases a new major version containing new features about once a year. Each major version receives bug fixes every three months in the form of minor releases. Minor version upgrades (e.g. from 12.**5** to 12.**9**) never change the internal storage format and are always compatible with other minor versions of the same major version. The service periodically updates the minor versions during customer's maintenance window.

Major version upgrades (e.g. from **12**.9 to **13**.2) are more complicated as internal storage format is subject to change, new features are introduced, internal schema changes etc. Customers usually have to create a new server with a higher version and perform dump and restore of data from the source server.

Flexible Server Postgres has now introduced an in-place major version upgrade feature that performs an offline in-place upgrade of the server with a push of a button. This means, there will be a short down time (usually around 10 minutes) while the upgrade happens and once completed, you can continue to access the same server with the same endpoint but now upgraded to a higher version.

The time to complete the operation does not depend on the size of data, but rather the number of objects in the database. Secondly, you can skip versions and go to a higher version directly. For example, you can upgrade from PG 11 to PG 14 . The service takes an implicit backup prior to attempting the upgrade operation. In the event of any failures, the server will be reverted to the older version using the backup.

NOTE: It is strongly recommended to perform a PITR of your production server and try the upgrade in that test server. That way, you can identify any application incompatibilities and address them before upgrading your production server.

User Experience

Major version upgrades need to be explicitly triggered by users with appropriate roles (like Owner).

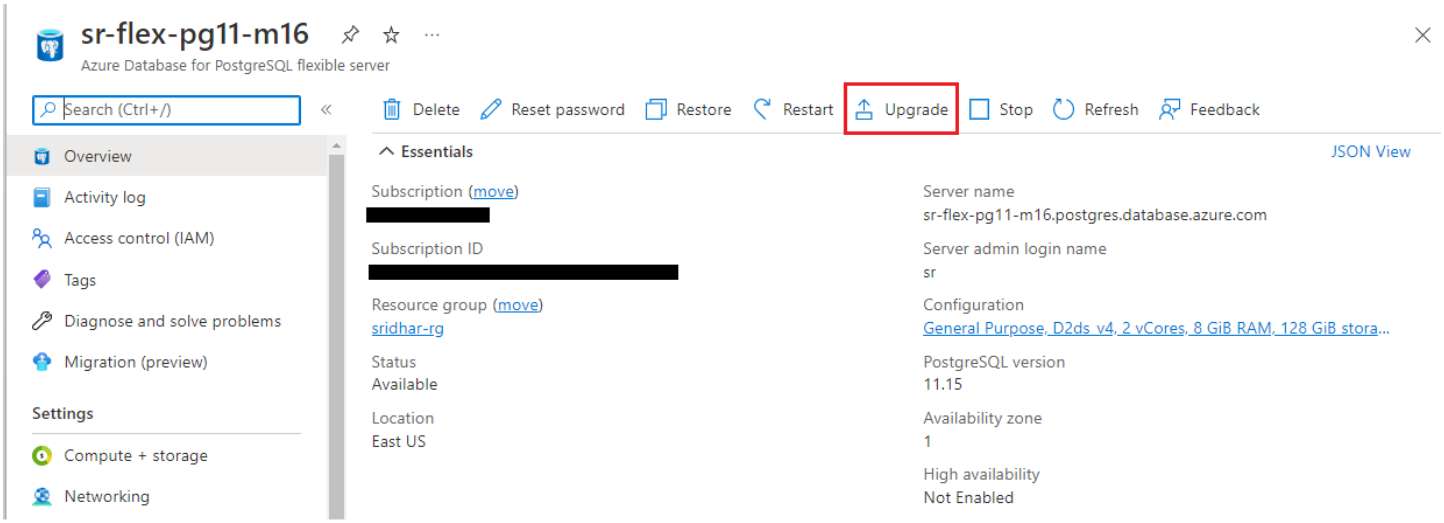

Figure 1: New Upgrade button in Overview blade

Clicking on the *Upgrade* button shows a blade allowing users to select the major Postgres version to upgrade to.

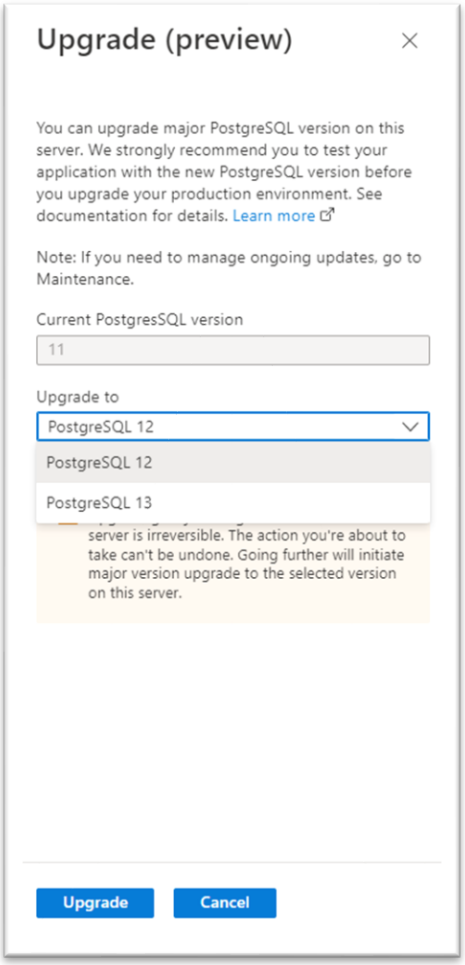

Figure 2: Upgrade blade allows to select version and start upgrade process

The upgrade task is initiated.

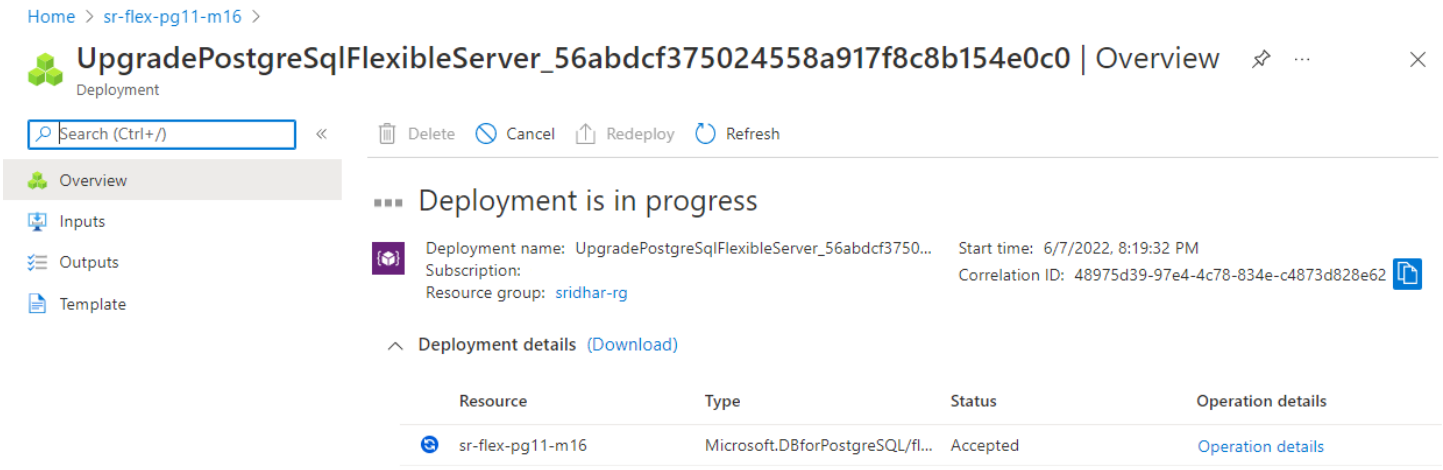

Figure 3: Upgrade process initiated as a deployment

Once the upgrade is in progress, users will have to wait for the process to be completed. You can start using the server once the server is back up.

Processes

- Before performing the upgrade, Flexible server takes an implicit snapshot of the server and reverts back in the event of any failure during the upgrade process.
- You can only choose a higher major version. Flexible server automatically deploys with the latest supported minor version.
- Once upgraded, there are no automated ways to change to the previous version. You can choose to perform a PITR to the time prior to the upgrade.
- The service checks for upgradability and aborts if any incompatibles are found. A detailed error message will be shown.
    - Example message: "The major version upgrade failed precheck. Upgrading extension postgis from source version 12 to target version 13 is not supported. Upgrading extension timescaledb from source version 12 to target version 13 is not supported. Upgrading with replication slots is not supported: found inactive logical replication slot regression slot. "
- The upgrade operation requires some free space on your disk. If your capacity usage is beyond 80% of your provisioned size, please consider increasing the storage before attempting to perform upgrades.
- If your server is configured with HA, during the upgrade operation, it will first disable HA (drops the standby), performs the upgrade, and re-enables HA.
- Perform ANALYZE operation post upgrade process to update the statistics.
- PITR operations
    - If you perform a PITR post the upgrade, then it will restore to the latest upgraded version.
    - If you choose a time prior to the upgrade, then the restored server will be based on the older version.
    - If you choose the time during the upgrade, the PITR operation will error out.

Limitations

- Major version upgrade will not progress if you have any of the following Postgres extensions. If you have any of these extensions, please drop them first and re-enable post upgrade. Note that this is the current private preview limitation
    - Non-supported extensions: TimescaleDB, dblink, postgres_fdw, pgaudit
    - PG 11 extensions not supported: orafce, postgis*, address_standardizer*

- Postgres extensions are not automatically upgraded during this process and most extensions share the same version between different PG major versions, however for the ones that do need upgrade you will need to drop and re-create the extension.

  You can verify the versions using below commands.
  - `select * from pg_extensions;` and
  - `select * FROM pg_available_extension_versions;` queries.
  - You can also verify azure.extensions.
- Servers configured with logical replication slots are not supported in private preview.
- We recommend you to perform upgrades using General Purpose or Memory Optimized. Burstable SKUs would work, but that is still being tested.

Contact

For any feedback or if you run into any issues, please contact AskAzureforPostgres-MVU@microsoft.com.