Overview

PostgreSql Database Migration Recommendation

Azure Database for PostgreSQL is a relational database service in the Microsoft cloud based on the PostgreSQL. Azure Database for PostgreSQL delivers:

- Built-in high availability.
- Data protection using automatic backups and point-in-time-restore for up to 35 days.
- Automated maintenance for underlying hardware, operating system and database engine to keep the service secure and up to date.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Elastic scaling within seconds.
- Enterprise grade security and industry-leading compliance to protect sensitive data at-rest and in-motion.
- Monitoring and automation to simplify management and monitoring for large-scale deployments, Industry-leading support experience.

Deployment models Azure Database for PostgreSQL powered by the PostgreSQL community edition is available in three deployment modes:

- Single Server
- Flexible Server (Preview)
- Hyperscale (Citus)

Database Details

Project Name	postgree10-09-21
Database Name	postgres
DB Version	PostgreSQL 12.2

Azure Infrastructure Cost

Azure Database for PostgreSQL	USD 256.4928
PostgreSQL running on VM	USD 368.93

DATABASE OBJECT LISTING

ObjectCount	
Function	485
Procedure	8
Table	53
View	23
Sequence	0
Index	0

Total	569

Recommendations

Datapoint Name

Unsupported Database Version

Reason of Change

The source database version is not compatible for migration to Azure PostgreSQL.

Recommendation

The source database version is not compatible for migration to the Azure Postgresql. Please update the version to 11 or above. The steps to upgrading the PostgreSQL database are: Steps 1: Backup the Database <ul> open pgAdmin 4 from the Windows Start menu. Log in with the PostgreSQL superuser to check the current name of the DB being used. Users may take an individual database backup by using the help of pg_dump, and we can take a backup of all databases at once using the pg_dumpall command. To do this, open a command prompt and traverse through the appropriate directory. Then run the pg_dump command, as shown here: CD C:\Program Files\PostgreSQL\9.0\bin   pg_dump -U Postgres -W -F t dvdrental > "V:\ PostgreSQL Backup\dvdrental.tar". Steps 2: Uninstalling the Old Version <ul> Open Programs and Features. Select the PostgreSQL version that is installed, and uninstall it. When prompted by the uninstaller - select Entire application: Click Next. Once the uninstall has finished, you will be prompted to reboot the server. Once the server has rebooted, delete the folder PostgreSQL leaves behind, e.g. C:\Program Files\PostgreSQL. Steps 3: Installing the New Version <ul> We can install the new version. You will need to recreate the database superuser password as it was for the previous version. Once the installation has finished, open pgAdmin 4. Steps 4: Recreating the Database   We can now recreate the database that we will restore our backup into. In pgAdmin, connect to the new database by right-clicking on Database and selecting Create > Database. Enter the name of the old database. Click the Save button at the bottom of the window. Steps 5: Restoring the Database Now that we have a new database for restoring the backed-up database, we can switch back to the admin command window we were using in Backing Up Database and run the restore command. Run the restore command and If you gave the backup a different filename, be sure to enter that filename. When prompted, enter the PostgreSQL superuser password. Once the restore is complete, run the Windows Services tool and restart the Postgre Database.

Impact

Mandatory

Estimated Hours

4 Hours task for one developer

Help URL

https://stackoverflow.com/questions/47029055/how-do-i-upgrade-my-postgresql-9-5-to-postgresql-10-on-ubuntu-16-04

Datapoint Name

Missing Primary Keys

Reason of Change

Primary keys are missing from some tables. For migration, all tables in your existing database need a primary key to ensure that the changes can be synced to the Azure PostgreSQL database.

Recommendation

 Primary keys must exist on all tables to be replicated. Migration of multiple tables with the same name, but a different case, might cause unpredictable behavior   There are two methods to add a Primary Key. <OL> Define the primary key when creating the table. <code>CREATE TABLE TABLE (
 column_1 data_type PRIMARY KEY,
 column_2 data_type,
 …);</code> In case the primary key consists of two or more columns, you define the primary key constraint as follows: <code> CREATE TABLE TABLE (
 column_1 data_type,
 column_2 data_type,
 …
 PRIMARY KEY (column_1, column_2)); </code>

 Define primary key when changing the existing table structure. <code> ALTER TABLE table_name
 ADD PRIMARY KEY (column_1, column_2); </code> </OL>

Impact

Mandatory

Estimated Hours

4 Hours task for one developer

Impacted Objects

a_table

bulk_collect_test

catalog1

catalog2

customer

customers

datatypemapping

date_tab

dimension_tab

emp

emp_1

emp_audit

emp_details

emp_log

emp45

emp65

employee

employeess

group_bo_ibo_assoc22

invoices

manager

numtab

optab

orders

orders_1

persons

print_media

student

student_table

student123

suppliers

t

t_row

t_urow

t_urow1

teacher

team

temp

test_data_sovlp

test123

tr

utl_file_dir

trf_unit_price

uc_emp

nulltest

student

employee

t

teacher

customers

orders

orders_1

emp

Help URL

https://www.techonthenet.com/postgresql/primary_keys.php

Datapoint Name

Unsupported Extension

Reason of Change

The source database extensions are not compatible for migration to the Azure PostgreSQL.

Recommendation

Use another way, If you need a specific extension that is not available in Azure PostgreSQL, you can consider using a different extension that supported in Azure PostgreSQL

Impact

Mandatory

Estimated Hours

4 Hours task for one developer

Help URL