T. CHANDRAKANTH

51760529

Migrate On-PREM PostgreSQL to Azure Database for PostgrESQL

# **Contents**

[About PostgreSQL 2](#_Toc352773136)

[Installation process 3](#_Toc352773137)

[Setting or Changing TCP Port 8](#_Toc352773138)

[Connect to PostgreSQL from pgAdmin 9](#_Toc352773139)

[Create Azure database for PostgreSQL server 11](#_Toc352773140)

[Connect to Azure database for PostgreSQL from Localhost 13](#_Toc352773141)

[Migrating PostgreSQL database from On Prem to Azure 15](#_Toc352773142)

* [Dump and Restore method: 15](#_Toc352773143)
* [Azure Data Migration Service (DMS): 19](#_Toc352773144)
* [Backup and Restore 27](#_Toc352773145)
* [Challenges 30](#_Toc352773146)

# **What is PostgreSQL?**

PostgreSQL is a general purpose and object-relational database management system, the most advanced open source database system. PostgreSQL was developed based on [POSTGRES 4.2](http://db.cs.berkeley.edu/postgres.html) at [Berkeley Computer Science Department](http://www.cs.berkeley.edu/), University of California.

PostgreSQL was designed to run on UNIX-like platforms. However, PostgreSQL was then also designed to be portable so that it could run on various platforms such as Mac OS X, Solaris, and Windows.

PostgreSQL is free and open source software. Its source code is available under PostgreSQL license, a liberal open source license. You are free to use, modify and distribute PostgreSQL in any form.

PostgreSQL requires very minimum maintained efforts because of its stability.  Therefore, if you develop applications based on PostgreSQL, the total cost of ownership is low in comparison with other database management systems.

## **PostgreSQL features highlights**

PostgreSQL has many advanced features that other enterprise database management systems offer, such as:

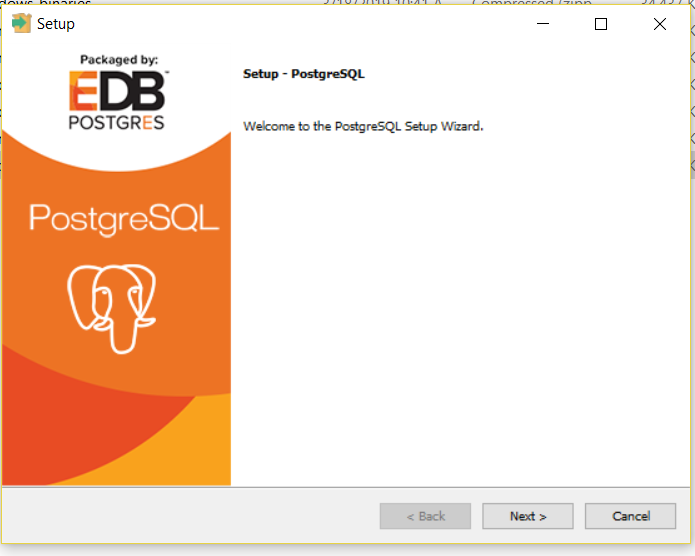
* User-defined types
* Table inheritance
* Sophisticated locking mechanism
* Foreign key referential integrity
* Views, rules, subquery
* Nested transactions (savepoints)
* Multi-version concurrency control (MVCC)
* Asynchronous replication

The recent versions of PostgreSQL support the following features:

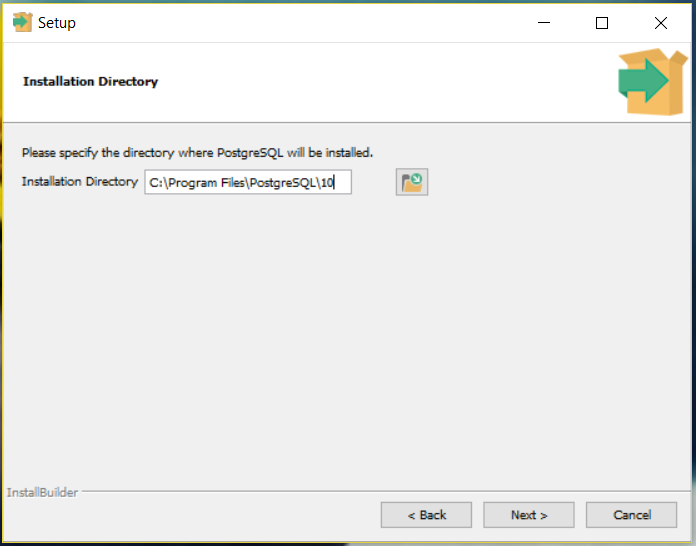
* Native Microsoft Windows Server version
* Tablespaces
* Point-in-time recovery

# **Installation process of PostgreSQL**

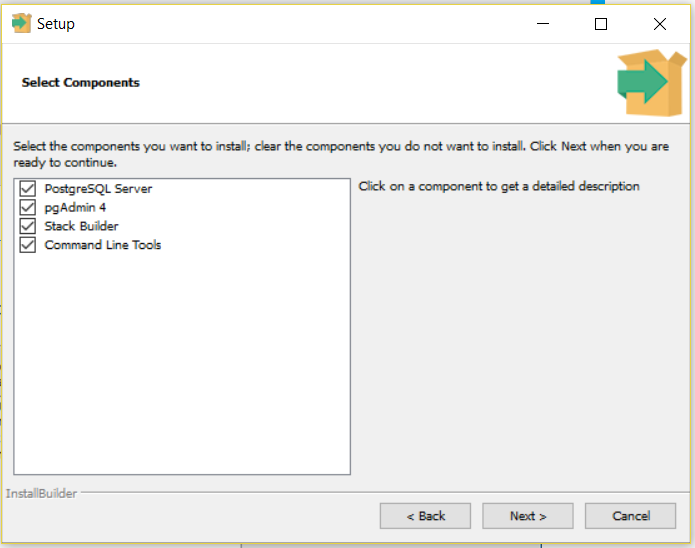
1. Double click on PostgreSQL installation file and click next in the below wizard



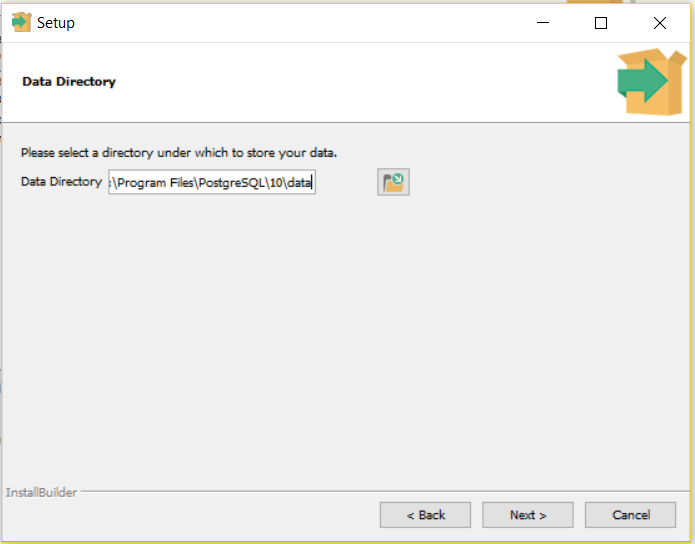
1. In the below path PostgreSQL will be installed. we can change the path



1. Select the components which needs to be installed

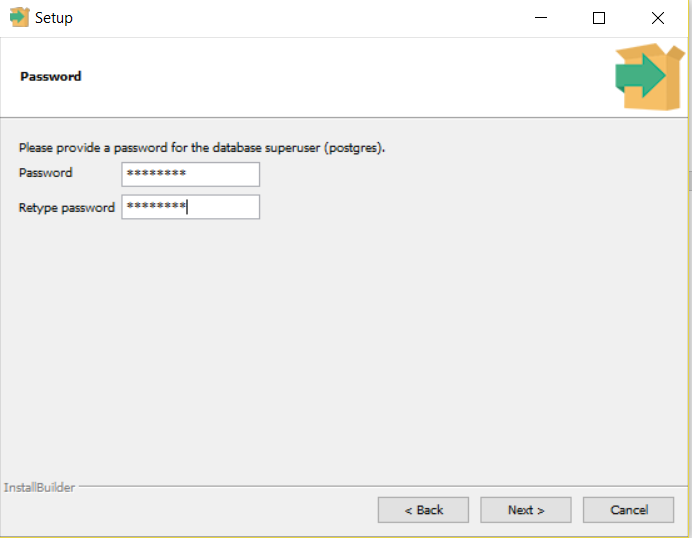


1. In the below path the data will be stored. We can change the path as per our requirement



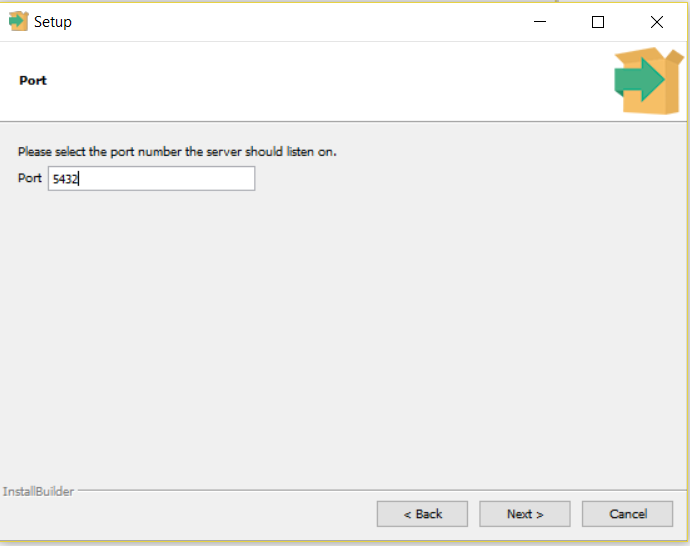
1. Create your own password in the below window to connect to PostgreSQL

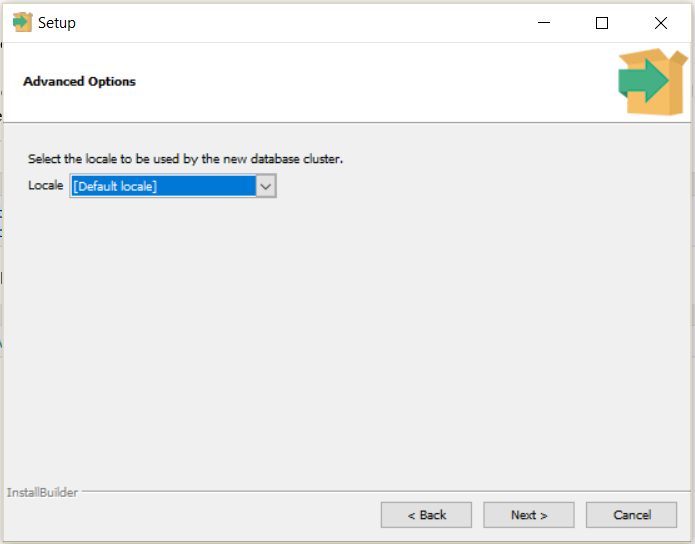
Password: admin123



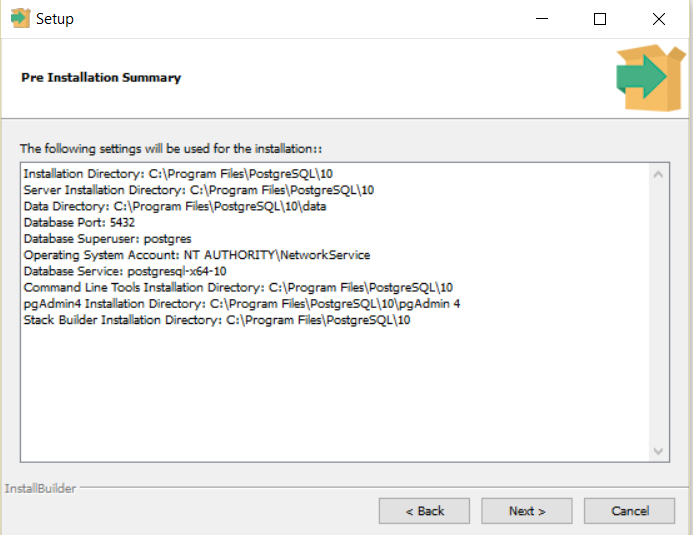
1. Below is the default port Number for PostgreSQL, we can change it if required so that server will listen to

TCP/IP connections

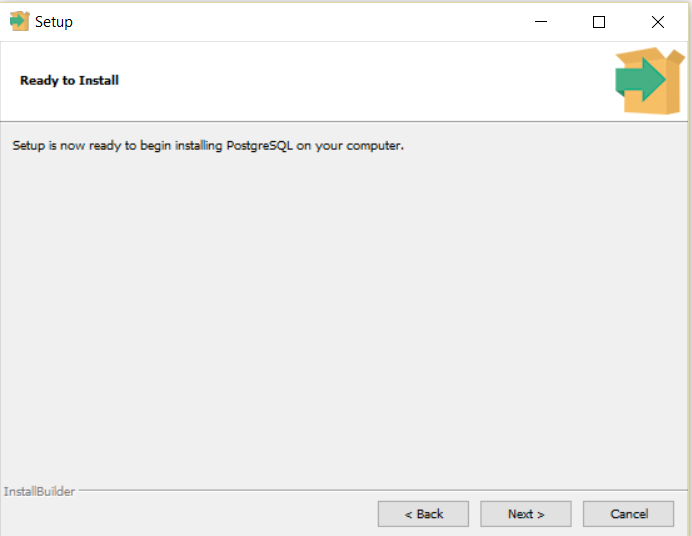




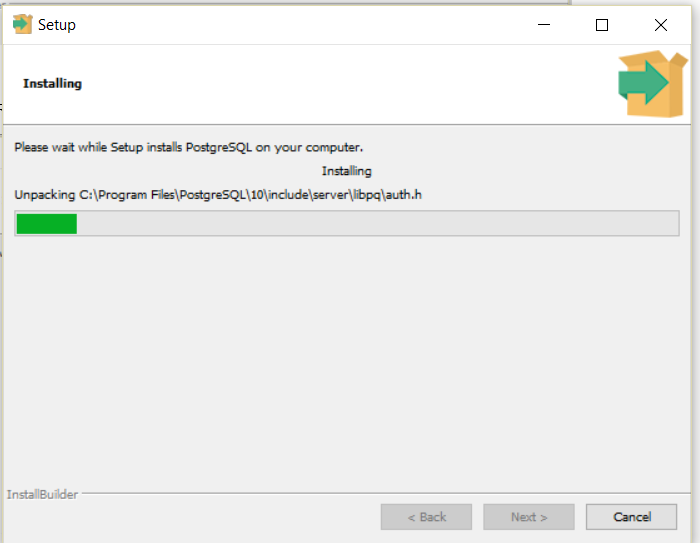
1. In the below wizard it will show the summary of installation paths and directory path where PostgreSQL installs and data stores.



1. Ready to Install



1. Once you click Next in the above wizard, installation will start



1. Click Finish to complete PostgreSQL



# **Setting or changing the TCP port**

The **port** option sets the PostgreSQL server port number that will be used when listening for TCP/ IP connections. The default port number is **5432** but you can change it as required. Use the **port** option with the   
**listen addresses** option to control the interface where the port will be listening. Use **‘\*’** to listen on all interfaces on the host, specify a single host name or IP address to listen on a single interface, or separate several hostname or IP addresses by space (such as ‘10.10.0.1 10.11.0.1’).



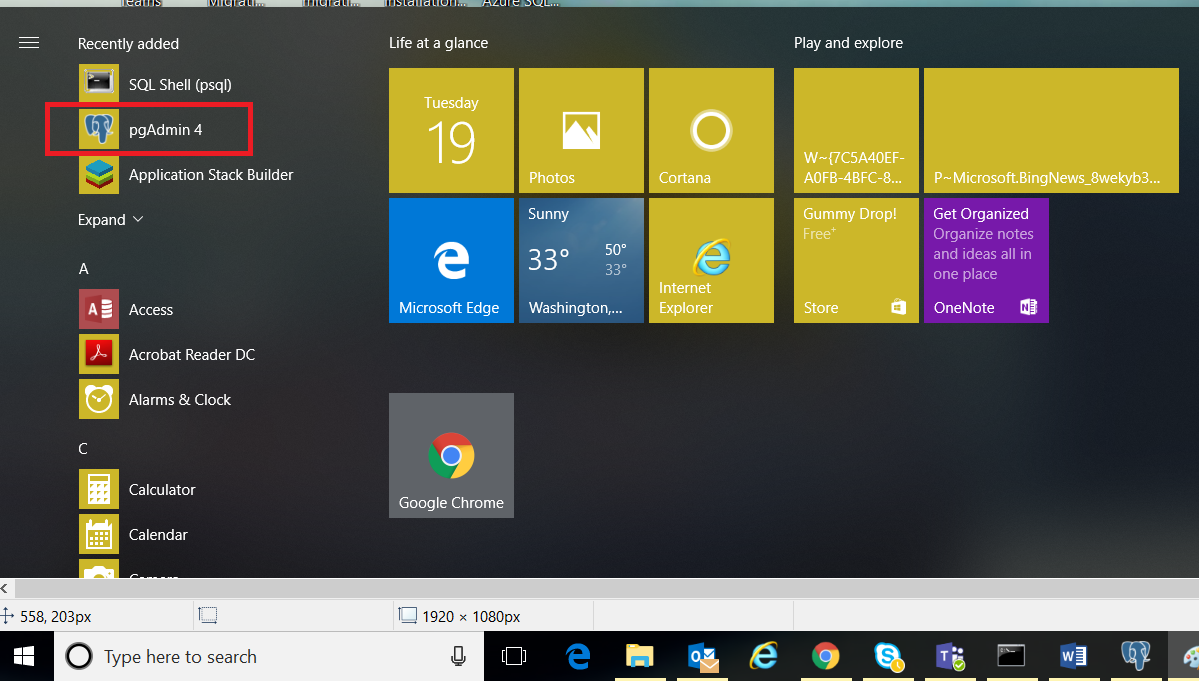
Restart the server for the changes to take effect.



# **How to Connect to Server using pdAdmin4**

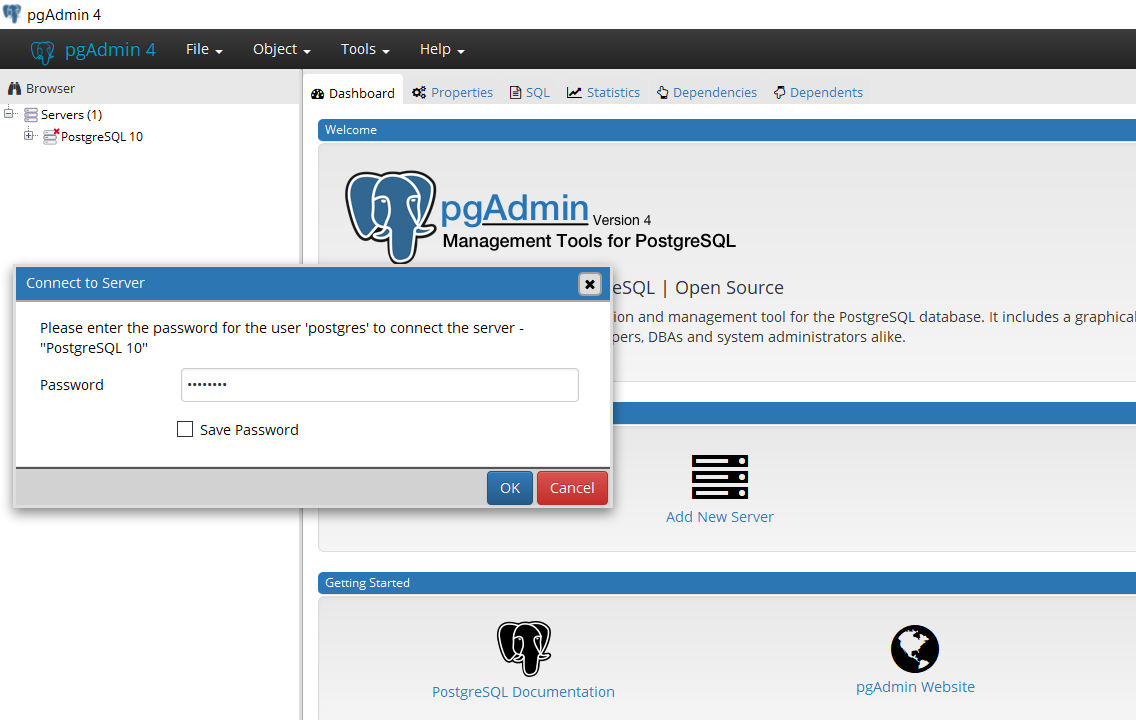
Once PostgreSQL installation completes, by following below steps we can connect to server/instance

1. Click Windows > open pgAdmin 4 > expand servers > expand instance > provide password to connect

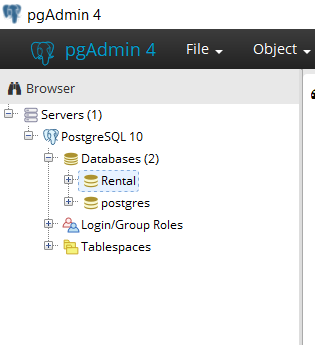


1. Once you click on the pgAdmin 4, in the below wizard expand servers > expand postgreSQL10 > it will ask for password to connect to server
2. Provide the password which we have created at the time of installation.

My Password: admin123



1. After providing the password we can able to see the databases, logins etc.



1. Now the source server is ready, create database, schemas (tables, stored procedures, views, triggers) to migrate it to Azure database for PostgreSQL. Below is the sample database file

Having tables, views, triggers, functions etc

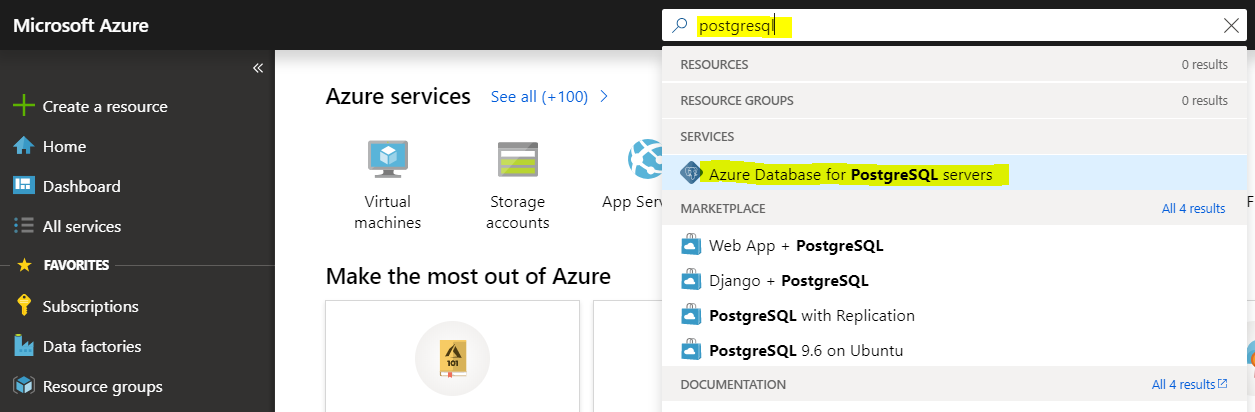


# **Create Azure database for PostgreSQL server (Target server)**

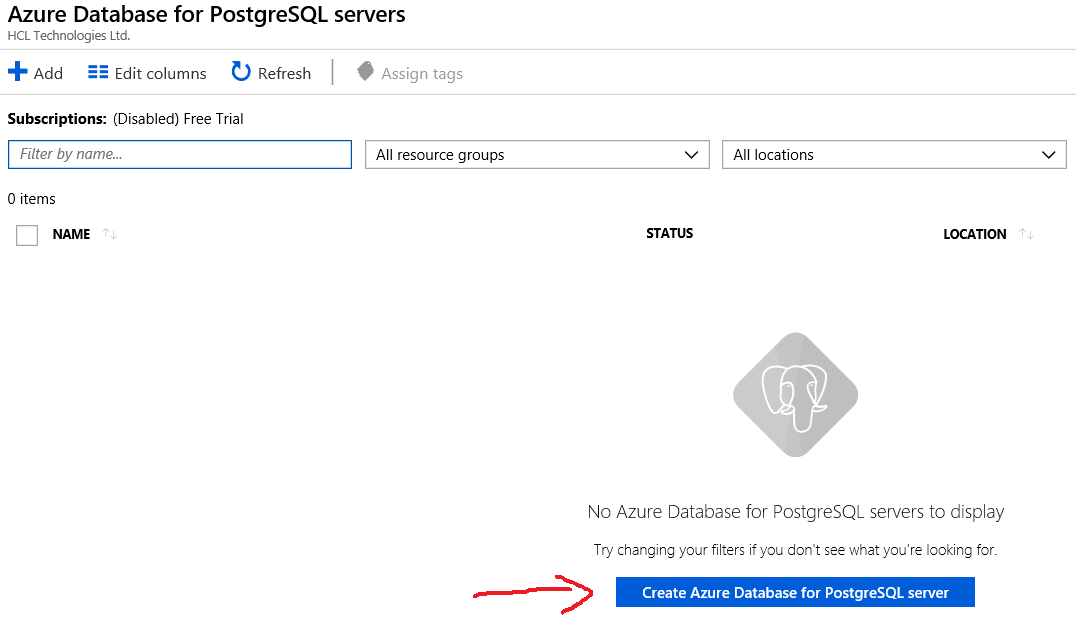
Connect to azure portal using <https://portal.azure.com>

Provide login and password details to connect

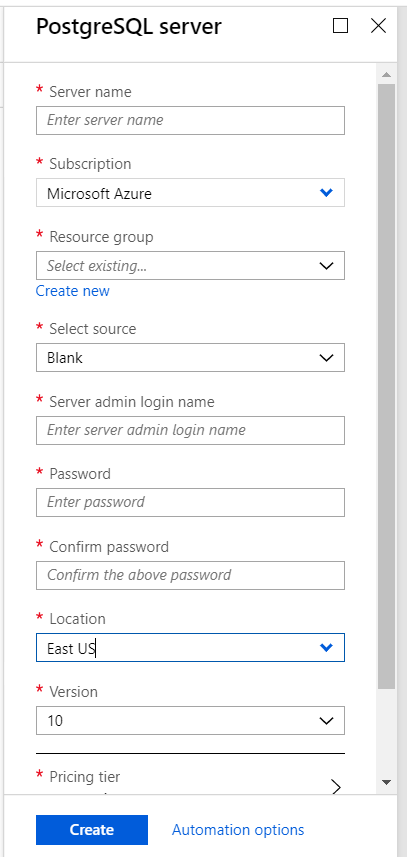
1. In search bar type PostgreSQL, in dropdown you will see **Azure database for PostgreSQL servers**. Click on that



1. Click on Create Azure Database for PostgreSQL server



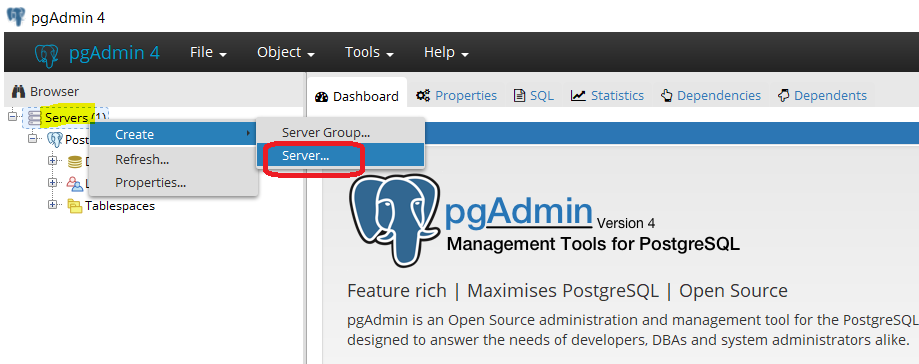
Provide below details for creation



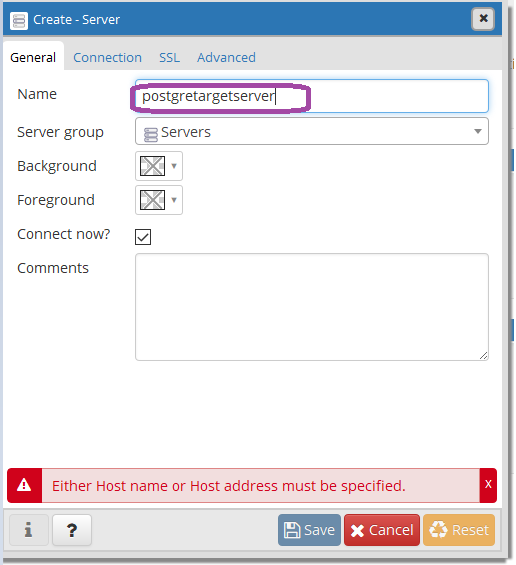
* Enter server name
* Select subscription from dropdown
* Select Resource Group from dropdown if already created or else create new
* Select source as blank as this is new server
* Enter admin login name and password to connect to server from instance
* Select Location, Version and pricing tier from dropdown and click on Create

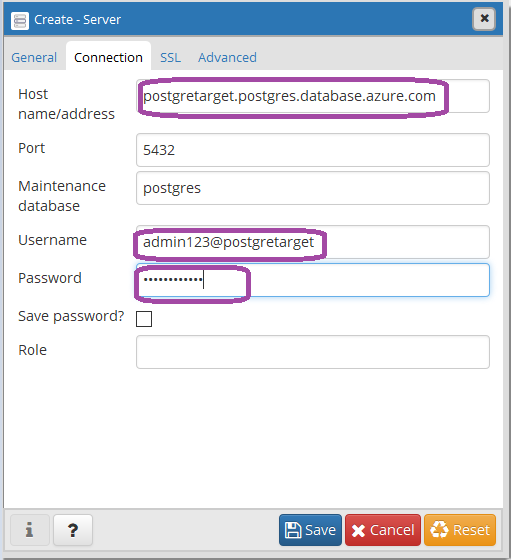
# **Connect to Azure PostgreSQL from local machine using pgAdmin**

1. Open pgAdmin in local machine > Right click Servers > create > click server

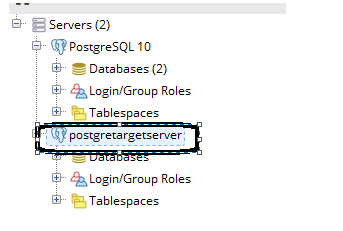


1. Once you click on server the below window will come, provide target server (azure PostgreSQL) details and connect
2. In General tab give Name as you like and in connection tab provide azure server name, admin login and password and then click save





1. Azure PostgreSQL connected successfully from local machine.



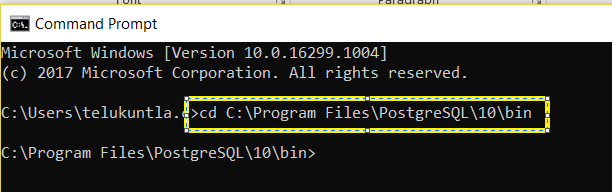
# **Migrating database objects from On-Prem PostgreSQL to Azure**

1. Using DUMP and Restore commands

To take backup/dump of on-prem database

1. Connect to on prem server > open command prompt > navigate to below path, and run pg\_dump command to take backup/dump of database

C:\Program Files\PostgreSQL\10\bin



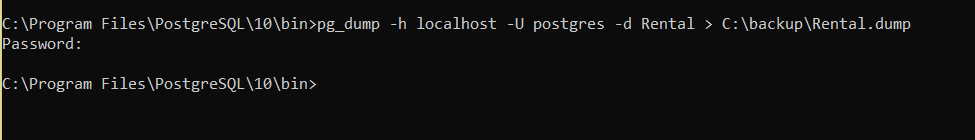
1. Run below command in command prompt to take database backup/dump.
2. Once you run the below command it will ask for server password to connect to server

Server name: localhost

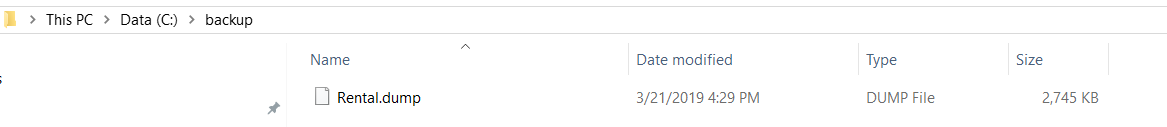
Username: postgres

Database name: Rental

pg\_dump -h localhost -U postgres -d Rental > C:\backup\Rental.dump

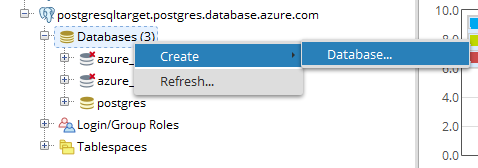


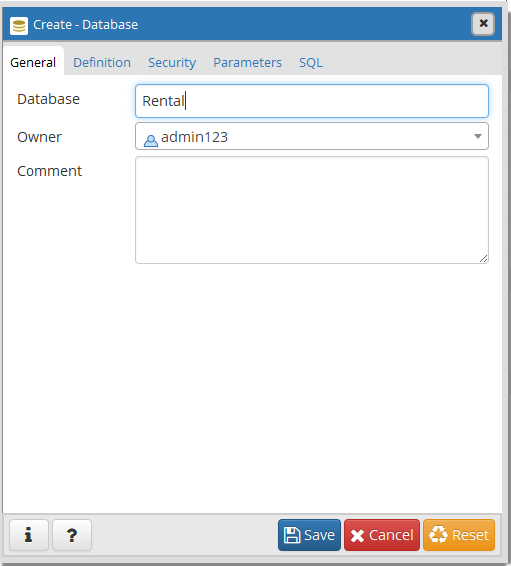
1. Database backup is done in the given path C:\backup\Rental.dump



Restore database on Azure database for PostgreSQL (Target server)

1. Connect to Azure server > Create a blank database by right clicking on databases





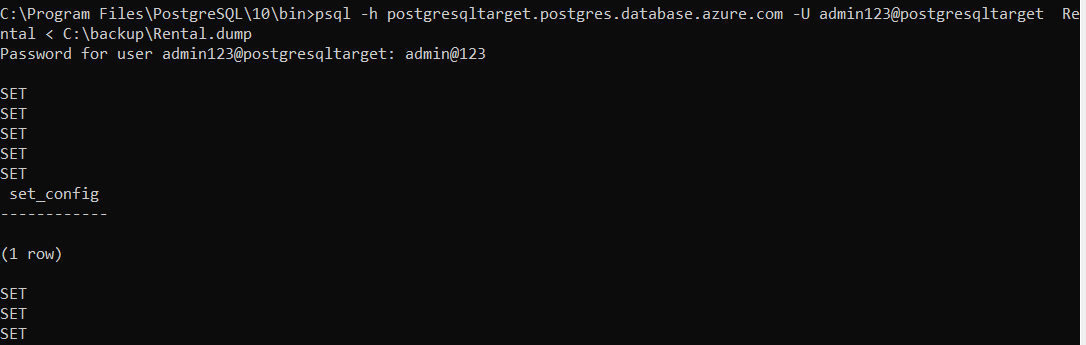
1. After you've created the target database, you can use the psql command to restore the data into the target database from the dump file.

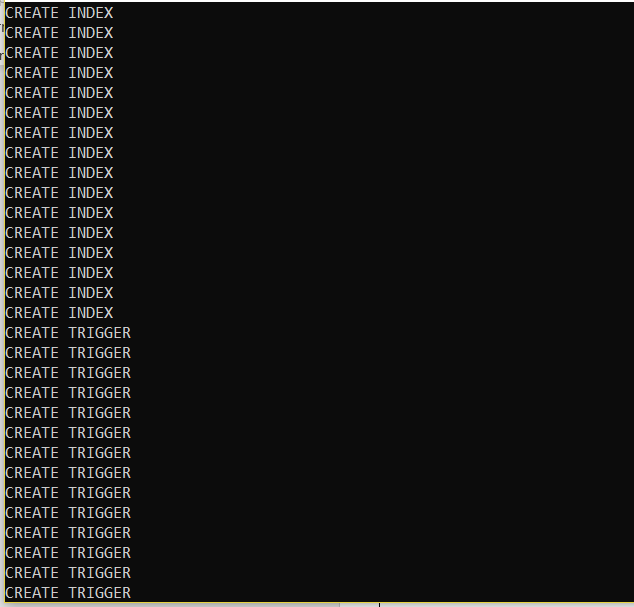
psql -h postgresqltarget.postgres.database.azure.com -U admin123@postgresqltarget Rental < C:\backup\Rental.dump

Provide target server name

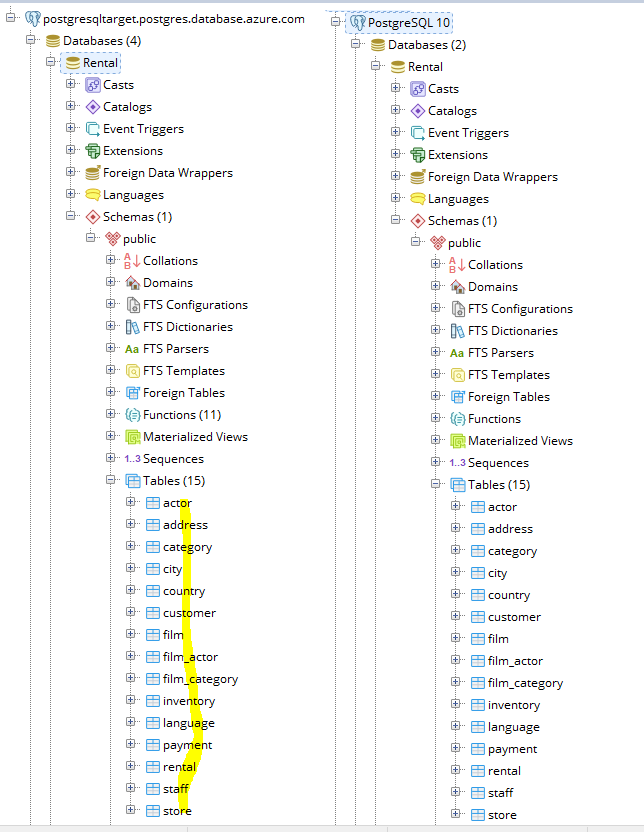
User name

Database name to restore data in it



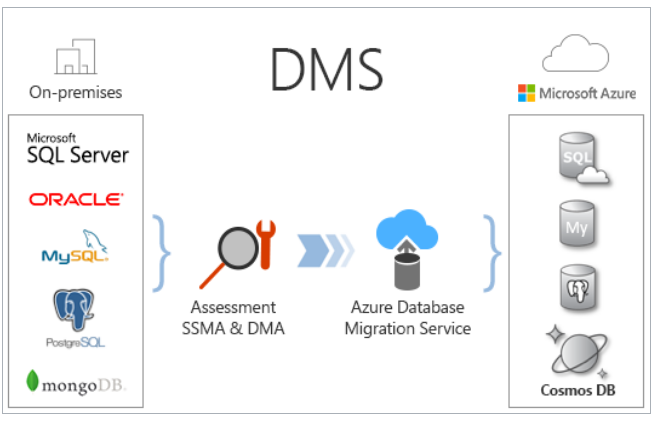


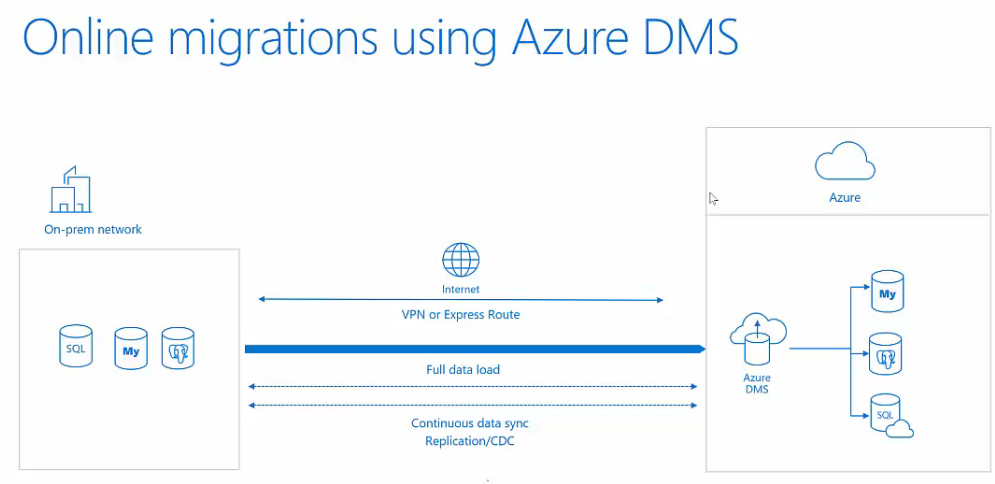
1. Now you can see all the database objects are created in Azure server compared with on-prem

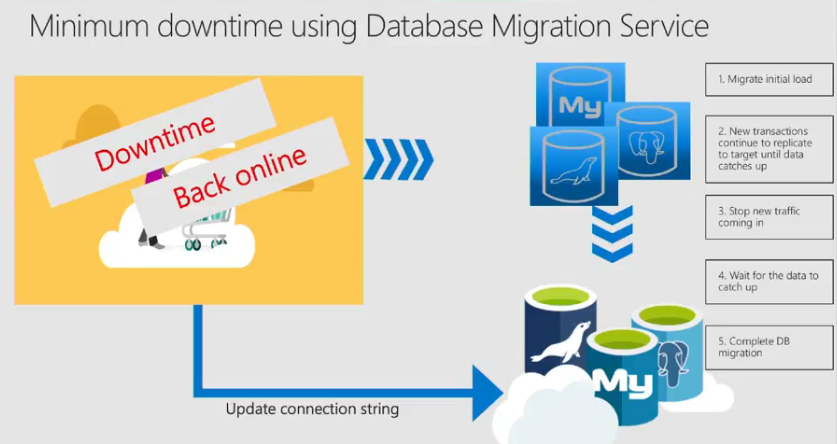


Azure Database Migration Service (DMS)

the [Azure Database Migration Service](https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fazure.microsoft.com%2Fen-us%2Fservices%2Fdatabase-migration%2F&data=02%7C01%7Cjtoland%40microsoft.com%7C435f68aba9b5442b901108d5fe5376db%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636694556804879848&sdata=dd26tanfxT1ic26XdcQbDya7ZbuZ%2BW8lmhYwX0oZkKk%3D&reserved=0) (DMS) provides users the ability to migrate on-premises PostgreSQL databases or PostgreSQL databases running in a virtual machine to Azure Database for PostgreSQL while the source database remains online during migration







Before we start migration, initially we need to synchronize schemas between Source and Destination servers (On-Prem and Azure PostgreSQL)

1. In On-Prem server open command prompt > Navigate to below path

C:\Program Files\PostgreSQL\10\bin

Run below command to create a schema dump file for a database

pg\_dump -h localhost -U postgres -d dvdrental -s > C:\backup\dvdrental.dump

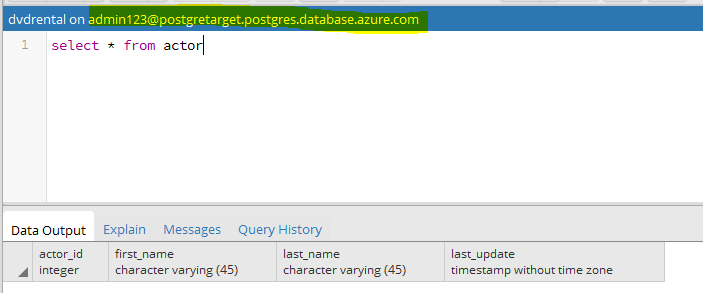
1. Create an empty database in your target environment, which is Azure Database for PostgreSQL.
2. Import the schema into the target database you created by restoring the schema dump file.

psql -h postgretarget.postgres.database.azure.com -U admin123@postgretarget dvdrental < C:\backup\dvdrental.sql

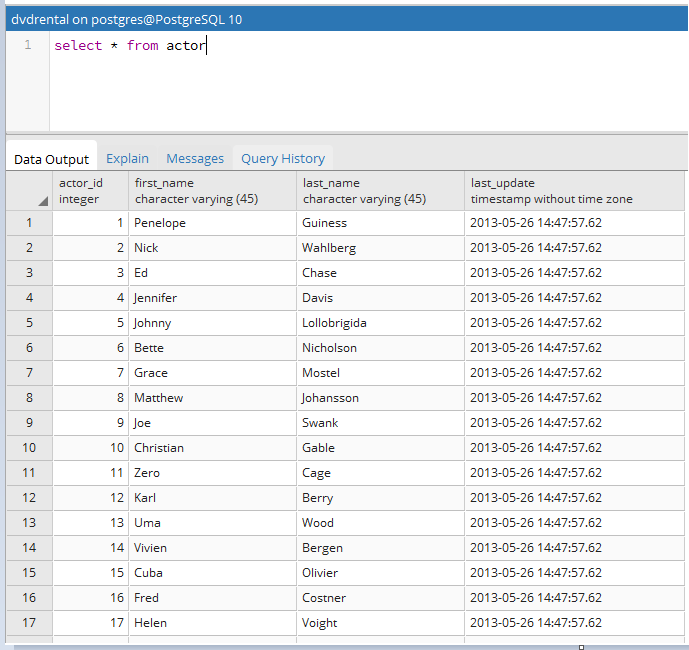
1. Successfully schema created in azure server

Please find the below screenshots of 1 table from both On-Prem and Azure

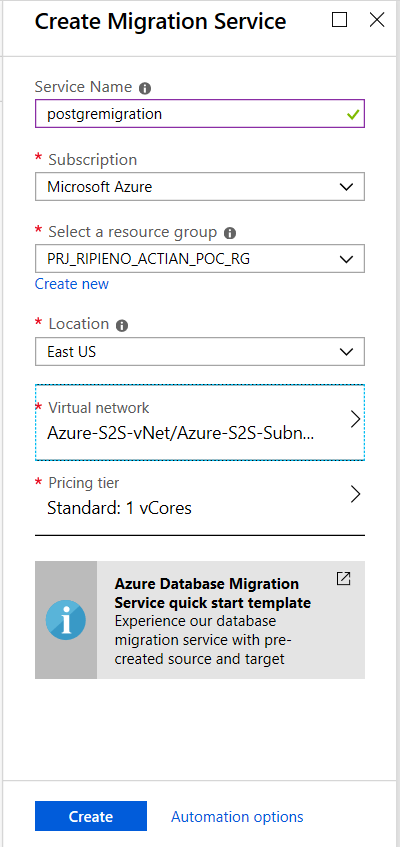
Only schema is created in Azure



In On-Prem we have data



1. To migrate data, go to azure portal > search for Azure data migration services under all services
2. Click on Create Azure data migration services > Provide below required details and click Create

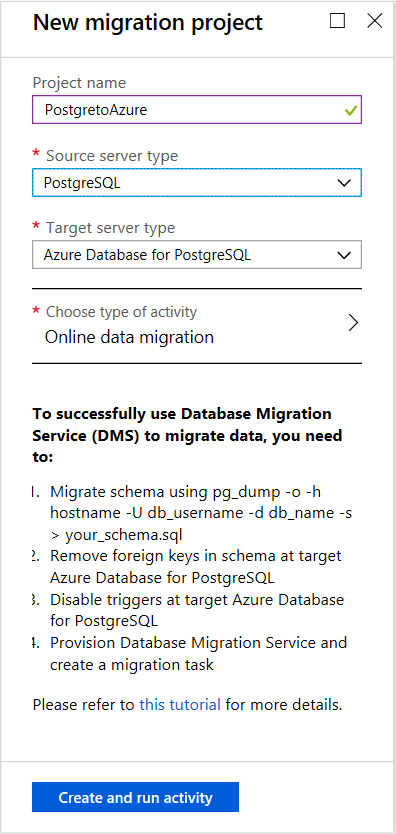


1. After Service created > go to Overview > Click on New Migration Project
2. Provide details in respective fields and click on Create and run activity.

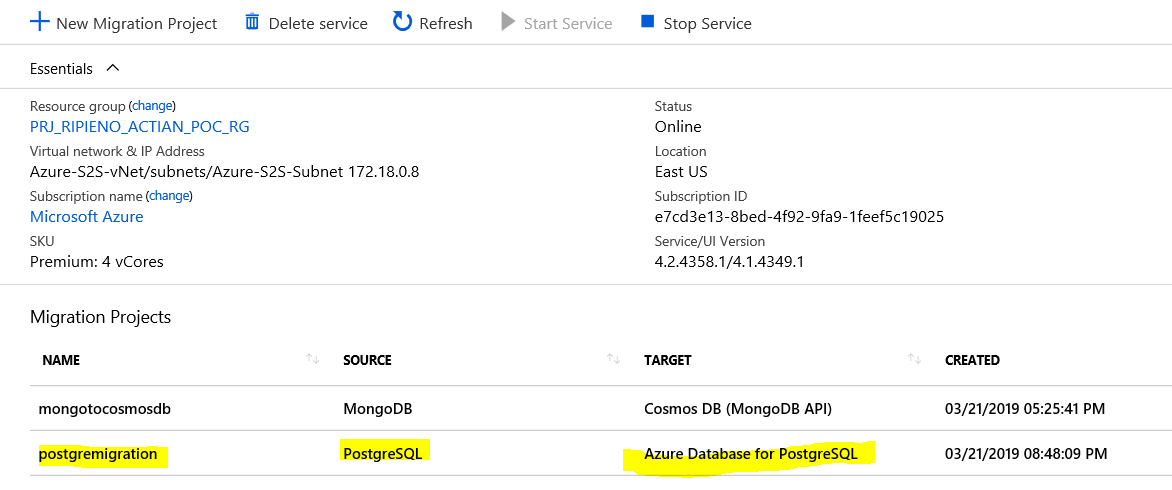
We selected

Source server type: PostgreSQL (On-Prem)

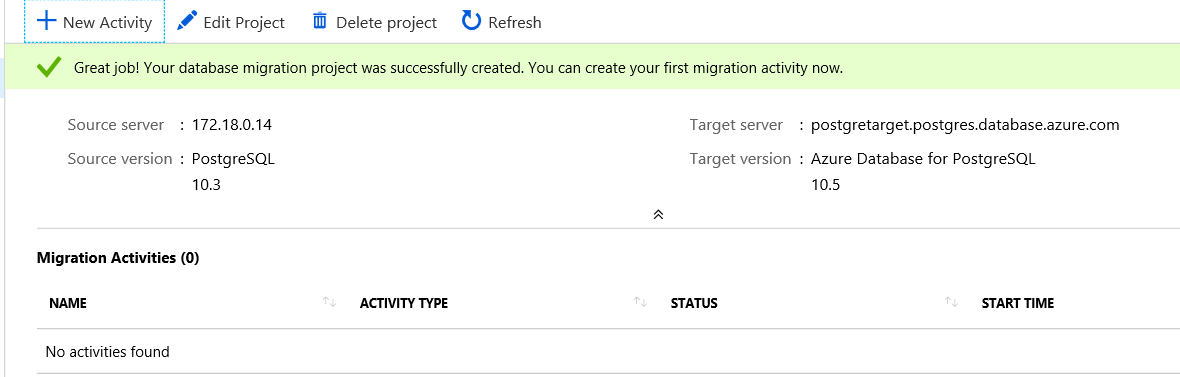
Target server type: Azure database for PostgreSQL



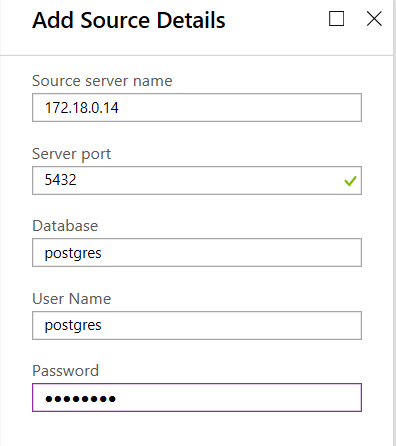
1. Once the Project is created, click on Project Postgremigration Project > Overview > Click on New Activity > Provide Source server details and target server details> Select databases > Click save. Please find below screenshots



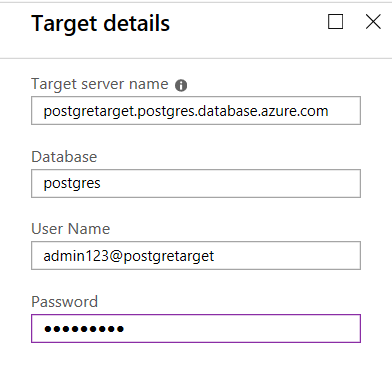
1. Click New activity



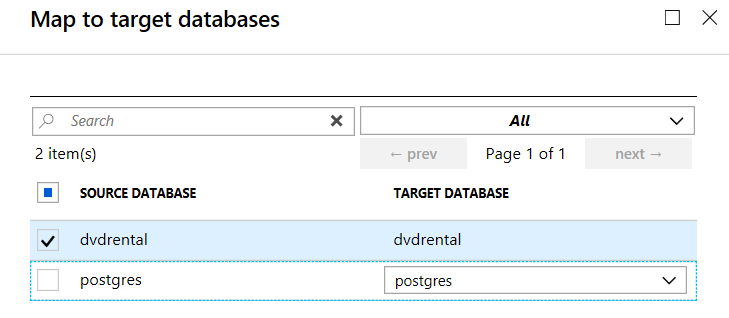
1. Give Source server details and click save



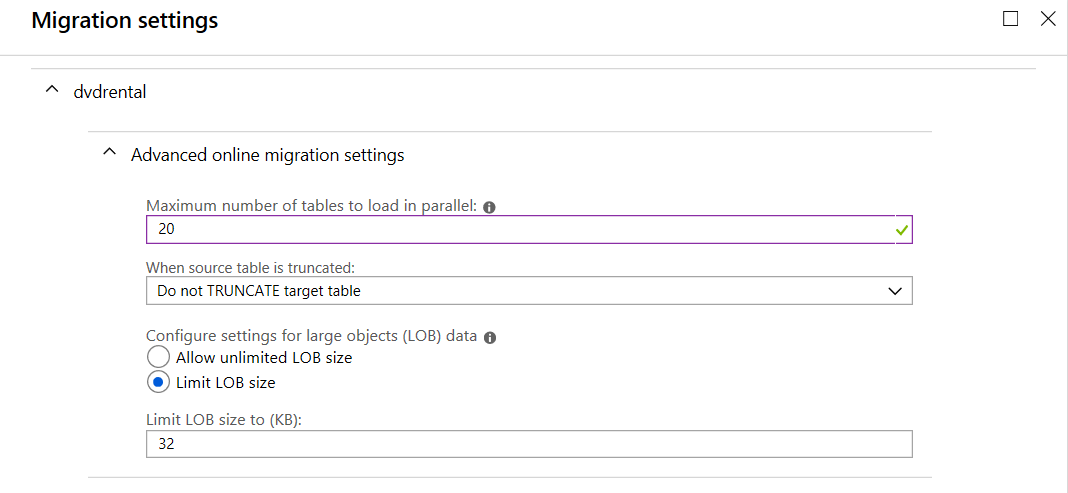
1. Give Target server details and click save. Azure server details



1. Select Source and target database to migrate data

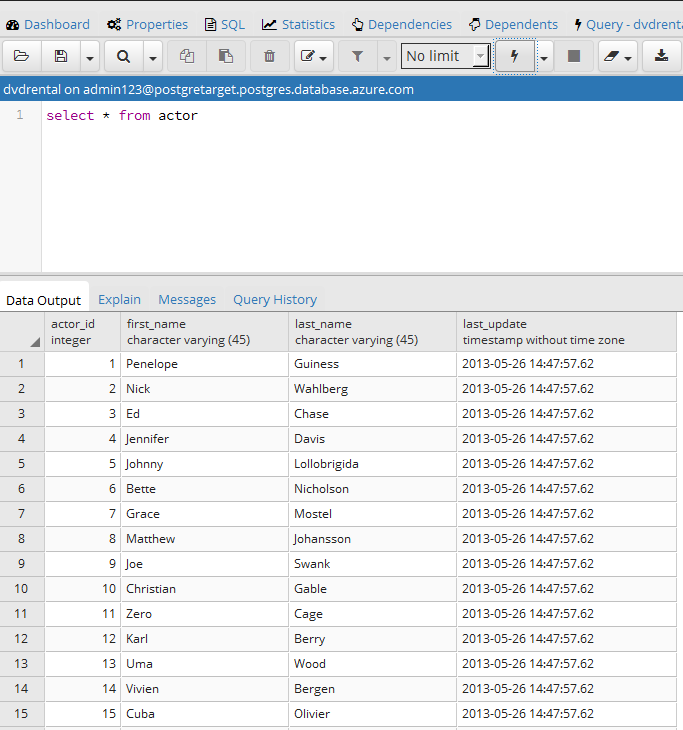


1. Give migration settings as required and click save



1. In Next window, give activity and Click on Run Migration.

Once the migration got completed you can see the data in Target server

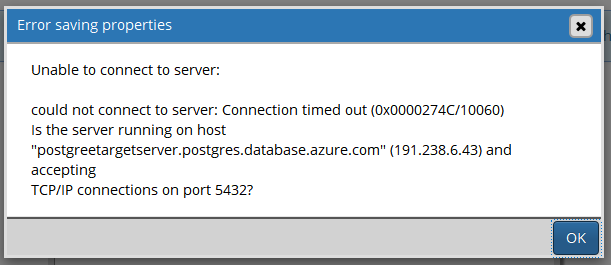


**Common scenarios:**

* SQL Server ⟶ [Azure SQL Database](https://datamigration.microsoft.com/scenario/sql-to-azuresqldb)
* SQL Server ⟶ [Azure SQL Database Managed Instance](https://datamigration.microsoft.com/scenario/sql-to-azuresqldbmi)
* Mongo DB ⟶ [Azure Cosmos DB](https://datamigration.microsoft.com/scenario/mongo-to-cosmos)
* MySQL ⟶ [Azure Database for MySQL](https://datamigration.microsoft.com/scenario/mysql-to-azuremysql)
* PostgreSQL ⟶ [Azure Database for PostgreSQL](https://datamigration.microsoft.com/scenario/postgresql-to-azurepostgresql)
* DB2 ⟶ [Azure SQL Database](https://datamigration.microsoft.com/scenario/db2-to-azuresqldb)
* Oracle ⟶ [Azure SQL Database](https://datamigration.microsoft.com/scenario/oracle-to-azuresqldb)
* Oracle ⟶ [Azure SQL Database Managed Instance](https://datamigration.microsoft.com/scenario/oracle-to-sqlserver)
* Oracle ⟶ [Azure Database for PostgreSQL](https://datamigration.microsoft.com/scenario/oracle-to-azurepostgresql)

https://docs.microsoft.com/en-us/azure/dms/tutorial-postgresql-azure-postgresql-online

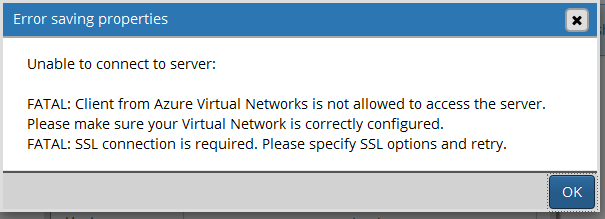
Error 1:



Error 2:

Client connections to Basic tier servers through Virtual Network Service Endpoints are not supported. Virtual Network Service Endpoints are supported for General Purpose and Memory Optimized servers.

Error 3:



Add existing Virtual network connectivity under connection security in Azure PostgreSQL