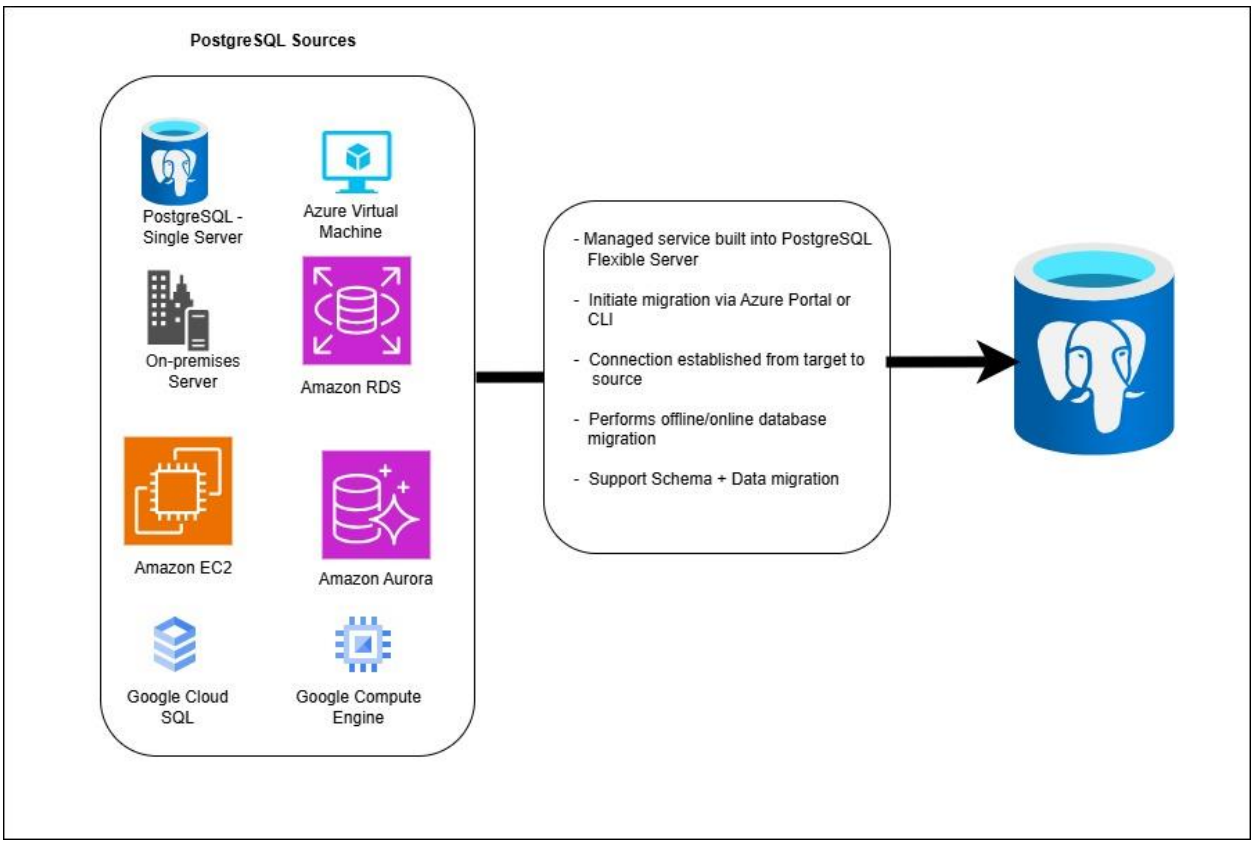
Azure PostgreSQL Database Migration

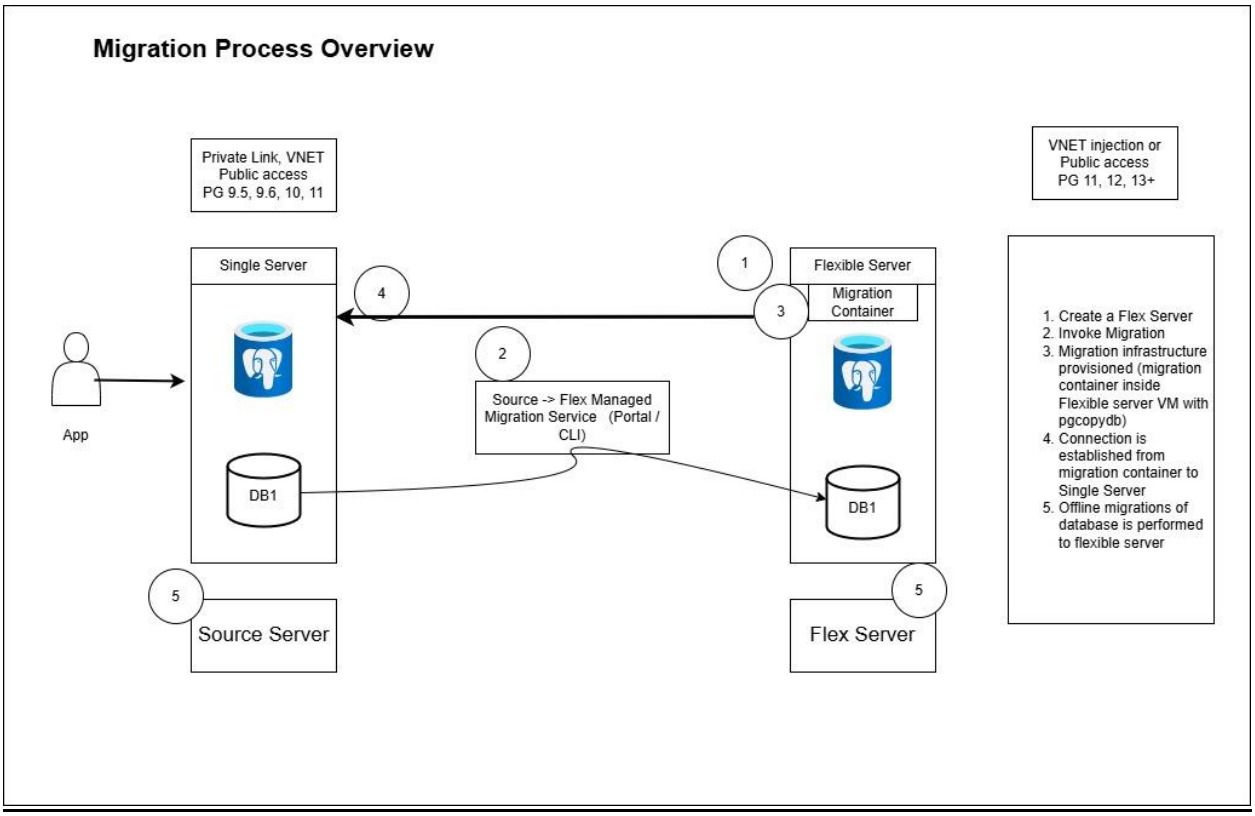
USE CASE:

In this project, I have created an Azure Database for PostgreSQL flexible server and perform an offline database migration from an Azure Database for PostgreSQL server using the Migration feature within the Azure Database for PostgreSQL Flexible Server.

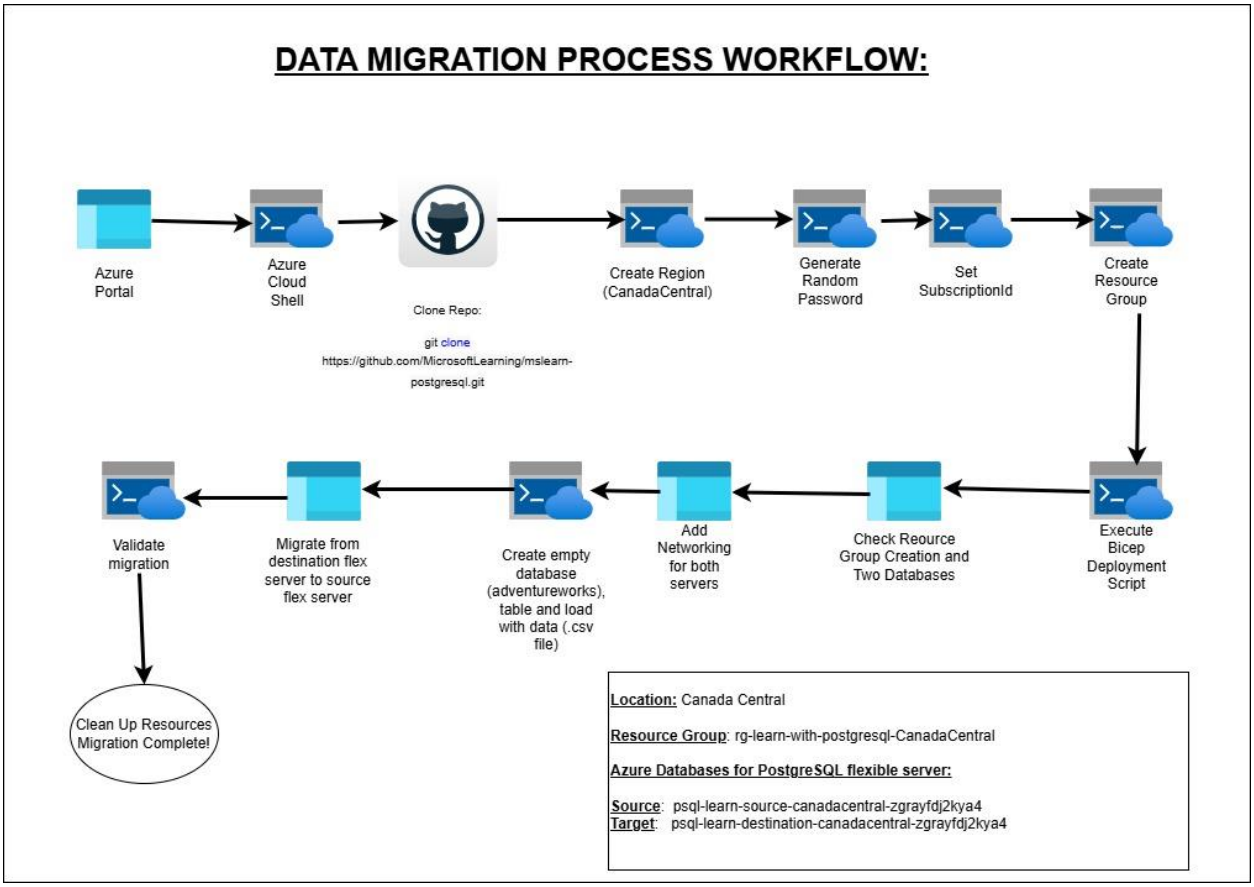
Below are the PostgreSQL sources you can migrate by using the migration service in Azure Database for PostgreSQL. All environments can be transitioned to Azure Database for PostgreSQL:



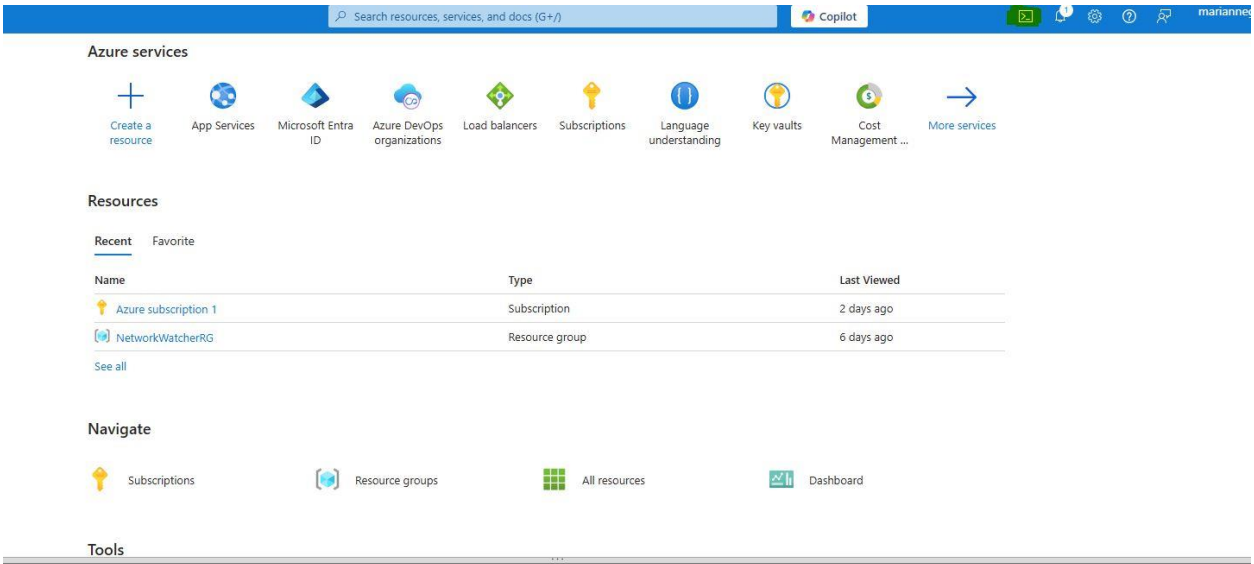
MIGRATION PROCESS OVERVIEW:

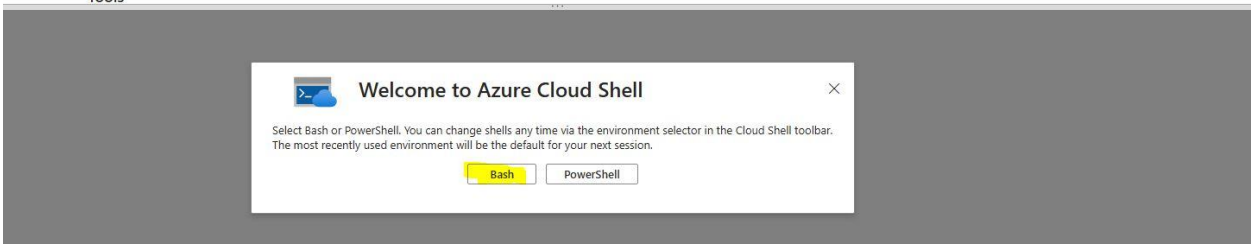


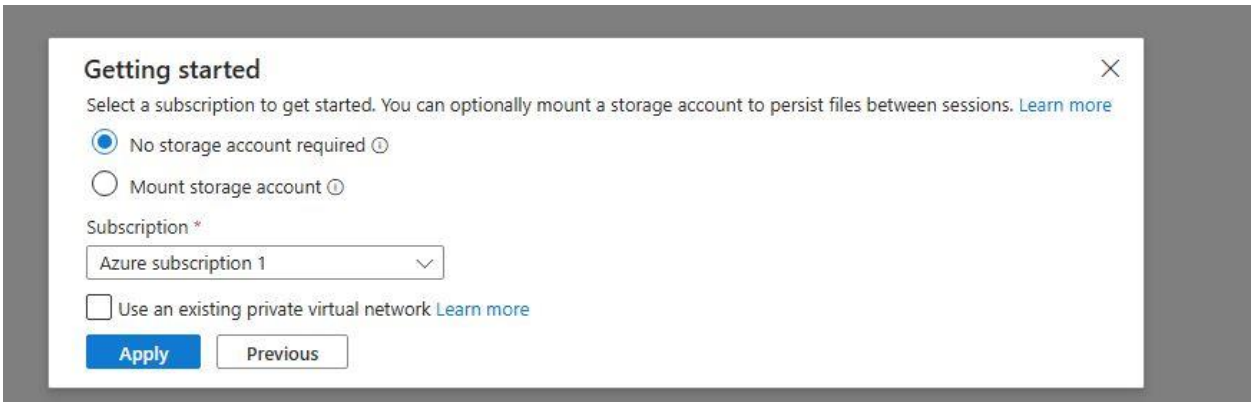
DATA MIGRATION PROCESS WORKFLOW SPECIFIC TO THIS PROJECT (USE CASE):



Step 1. Open Azure portal and select Cloud Shell in the upper right hand corner. When prompted select Bash shell.







Step 2. At the cloud shell prompt, enter the following to clone the GitHub repo containing the project resources.

git clone https://github.com/MicrosoftLearning/mslearn-postgresql.git

Step 3. Next, you run three commands to define variables to reduce redundant typing when using Azure CLI commands to create Azure resources. The variables represent the name to assign to your resource group (`RG_NAME`), the Azure region (`REGION`) into which resources will be deployed, and a randomly generated password for the PostgreSQL administrator login (`ADMIN_PASSWORD`).

`REGION=eastus`

I changed to:

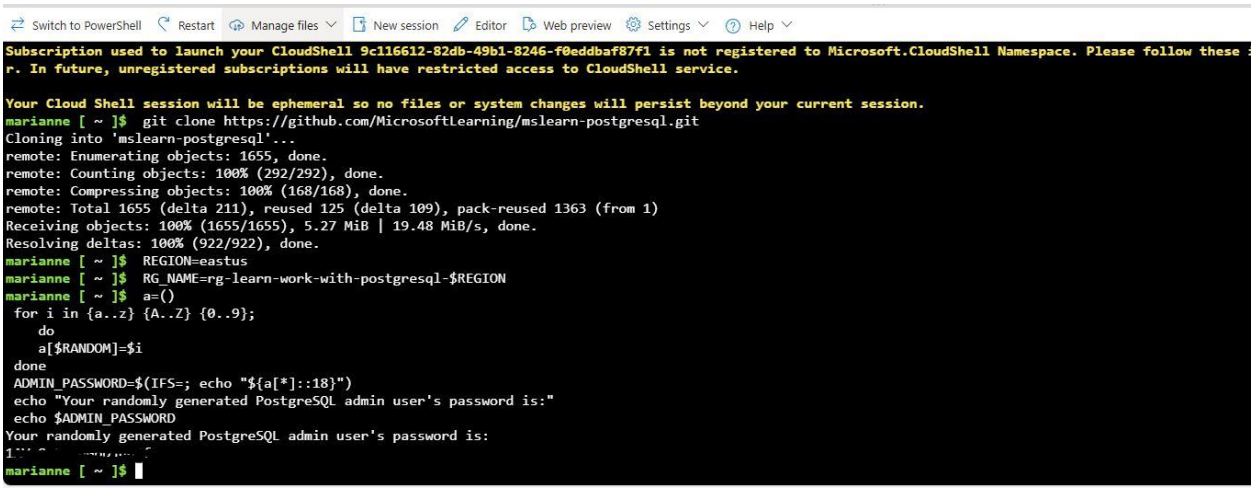
`REGION=CanadaCentral`

The following command assigns the name to be used for the resource group that will house all the resources used in this exercise. The resource group name assigned to the corresponding variable is `rg-learn-work-with-postgresql-$REGION`, where `$REGION` is the location you specified above. However, you can change it to any other resource group name that suits your preference.

`RG_NAME=rg-learn-work-with-postgresql-$REGION`

The final command randomly generates a password for the PostgreSQL admin login. Make sure you copy it to a safe place to use later to connect to your PostgreSQL flexible server.

```
a=()
 for i in {a..z} {A..Z} {0..9};
   do
   a[$RANDOM]=$i
 done
 ADMIN_PASSWORD=$(IFS=; echo "${a[*]::18}")
 echo "Your randomly generated PostgreSQL admin user's password is:"
 echo $ADMIN_PASSWORD
```



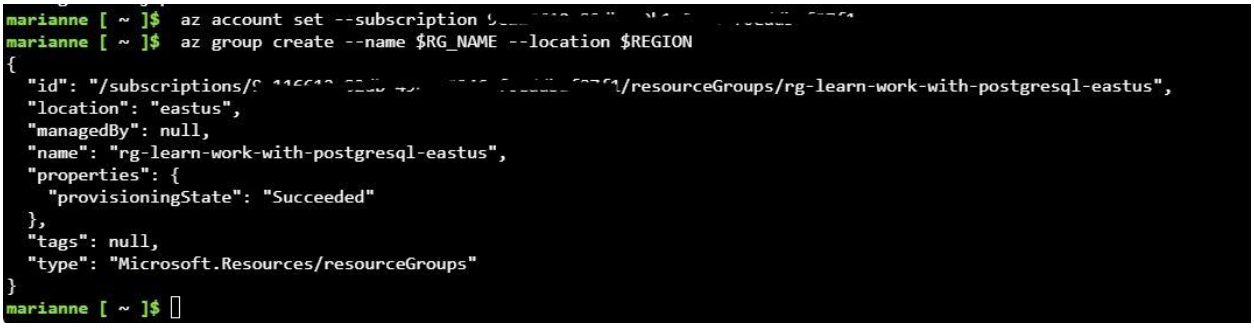
Step 4. If you have access to more than one Azure subscription, and your default subscription is not the one in which you want to create the resource group and other resources for this exercise, run this command to set the appropriate subscription, replacing the <subscriptionName|subscriptionId> token with either the name or ID of the subscription you want to use:

az account set --subscription <subscriptionName|subscriptionId>

I chose to use subscribtionId

Step 5. Run the following Azure CLI command to create your resource group:

az group create --name $RG_NAME --location $REGION



Step 6. Finally, use the Azure CLI to execute a Bicep deployment script to provision Azure resources in your resource group:

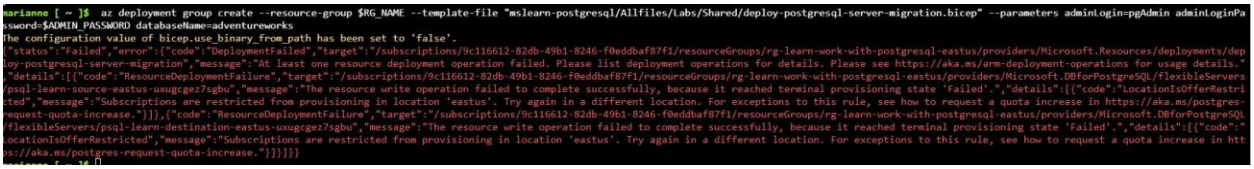
az deployment group create --resource-group $RG_NAME --template-file "mslearn-postgresql/Allfiles/Labs/Shared/deploy-postgresql-server-migration.bicep" --parameters adminLogin=pgAdmin adminLoginPassword=$ADMIN_PASSWORD databaseName=adventureworks

The Bicep deployment script provisions the Azure services required to complete this exercise into your resource group. The resources deployed are two Azure Database for PostgreSQL - Flexible Servers. A source and a destination server for the migration.

The deployment typically takes several minutes to complete (5-10+ minutes). You can monitor it from the Cloud Shell or navigate to the **Deployments** page for the resource group you created above and observe the deployment progress there.

ERROR: is Location of EastUS. I chose another location of Central Canada and reran the commands. See the below screen shots.

ERROR MESSAGE MEANING: If the selected region is restricted from provisioning specific resources, you must set the `REGION` variable to a different location and rerun the commands to create the resource group and run the Bicep deployment script.



```
marianne [ ~ ]$  az deployment group create --resource-group $RG_NAME --template-file "mslearn-postgresql/Allfiles/Labs/Shared/deploy-postgresql-server-migration.bicep" --parameters adminLogin=pgAdmin adminLoginPa
ssword=$ADMIN_PASSWORD databaseName=adventureworks
{
  "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.Resources/deployments/deploy-postgresql-server-migration",
  "location": null,
  "name": "deploy-postgresql-server-migration",
  "properties": {
    "correlationId": "e643eccf-90be-46b0-a72a-16d2b2440a3c",
    "debugSetting": null,
    "dependencies": [
      {
        "dependsOn": [
          {
            "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral
-zgrayfdj2kya4",
            "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
            "resourceName": "psql-learn-source-canadacentral-zgrayfdj2kya4",
            "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers"
          }
        ],
        "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral-zgr
ayfdj2kya4/firewallRules/sourceAllowAllAzureServicesAndResourcesWithinAzureIps",
        "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
        "resourceName": "psql-learn-source-canadacentral-zgrayfdj2kya4/sourceAllowAllAzureServicesAndResourcesWithinAzureIps",
        "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers/firewallRules"
      },
      {
        "dependsOn": [
          {
            "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral
-zgrayfdj2kya4",
            "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
            "resourceName": "psql-learn-source-canadacentral-zgrayfdj2kya4",
            "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers"
          }
        ],
```

```
        ],
        "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral-zgr
ayfdj2kya4/firewallRules/sourceAllowAll",
        "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
        "resourceName": "psql-learn-source-canadacentral-zgrayfdj2kya4/sourceAllowAll",
        "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers/firewallRules"
      },
      {
        "dependsOn": [
          {
            "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral
-zgrayfdj2kya4",
            "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
            "resourceName": "psql-learn-source-canadacentral-zgrayfdj2kya4",
            "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers"
          }
        ],
        "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral-zgr
ayfdj2kya4/databases/adventureworks",
        "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
        "resourceName": "psql-learn-source-canadacentral-zgrayfdj2kya4/adventureworks",
        "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers/databases"
      },
      {
        "dependsOn": [
          {
            "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadace
ntral-zgrayfdj2kya4",
            "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
            "resourceName": "psql-learn-destination-canadacentral-zgrayfdj2kya4",
            "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers"
          }
        ],
```

```
        ],
        "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadacentra
l-zgrayfdj2kya4/firewallRules/destinatinAllowAllAzureServicesAndResourcesWithinAzureIps",
        "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
        "resourceName": "psql-learn-destination-canadacentral-zgrayfdj2kya4/destinatinAllowAllAzureServicesAndResourcesWithinAzureIps",
        "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers/firewallRules"
      },
      {
        "dependsOn": [
          {
            "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadace
ntral-zgrayfdj2kya4",
            "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
            "resourceName": "psql-learn-destination-canadacentral-zgrayfdj2kya4",
            "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers"
          }
        ],
        "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadacentra
l-zgrayfdj2kya4/firewallRules/destinationAllowAll",
        "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
        "resourceName": "psql-learn-destination-canadacentral-zgrayfdj2kya4/destinationAllowAll",
        "resourceType": "Microsoft.DBforPostgreSQL/flexibleServers/firewallRules"
      }
    ],
    "diagnostics": null,
    "duration": "PT6M17.0722211S",
    "error": null,
    "mode": "Incremental",
    "onErrorDeployment": null,
    "outputResources": [
      {
```

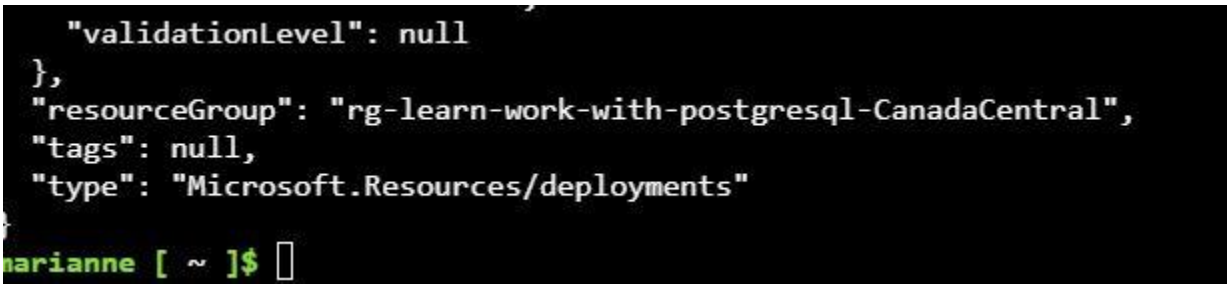
```
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadacentra
l-zgrayfdj2kya4",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    },
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadacentra
l-zgrayfdj2kya4/firewallRules/destinatinAllowAllAzureServicesAndResourcesWithinAzureIps",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    },
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadacentra
l-zgrayfdj2kya4/firewallRules/destinationAllowAll",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    },
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral-zgr
ayfdj2kya4",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    },
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral-zgr
ayfdj2kya4/databases/adventureworks",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    },
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral-zgr
ayfdj2kya4/firewallRules/sourceAllowAll",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    },
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-source-canadacentral-zgr
ayfdj2kya4/firewallRules/sourceAllowAllAzureServicesAndResourcesWithinAzureIps",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    }
  ],
```



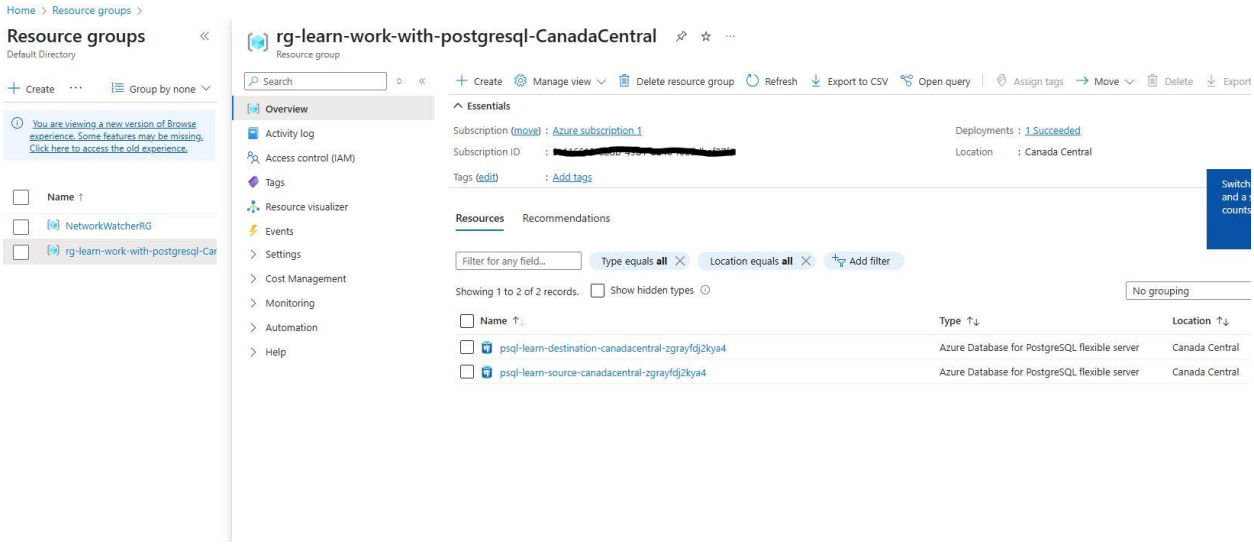
```
  "outputs": {
    "databaseName": {
      "type": "String",
      "value": "adventureworks"
    },
    "destServerFqdn": {
      "type": "String",
      "value": "psql-learn-destination-canadacentral-zgrayfdj2kya4.postgres.database.azure.com"
    },
    "destServerName": {
      "type": "String",
      "value": "psql-learn-destination-canadacentral-zgrayfdj2kya4"
    },
    "sourceServerFqdn": {
      "type": "String",
      "value": "psql-learn-source-canadacentral-zgrayfdj2kya4.postgres.database.azure.com"
    },
    "sourceServerName": {
      "type": "String",
      "value": "psql-learn-source-canadacentral-zgrayfdj2kya4"
    }
  },
  "parameters": {
    "adminLogin": {
      "type": "String",
      "value": "pgAdmin"
    },
    "adminLoginPassword": {
      "type": "SecureString"
    },
    "databaseName": {
      "type": "String",
      "value": "adventureworks"
    },
    "destinationServerName": {
      "type": "String",
      "value": "psql-learn-destination-canadacentral-zgrayfdj2kya4"
    },
```

```
    "value": "psql-learn-destination-canadacentral-zgrayfdj2kya4"
  },
  "location": {
    "type": "String",
    "value": "canadacentral"
  },
  "postgresVersion": {
    "type": "String",
    "value": "16"
  },
  "sourceServerName": {
    "type": "String",
    "value": "psql-learn-source-canadacentral-zgrayfdj2kya4"
  }
},
"parametersLink": null,
"providers": [
  {
    "id": null,
    "namespace": "Microsoft.DBforPostgreSQL",
    "providerAuthorizationConsentState": null,
    "registrationPolicy": null,
    "registrationState": null,
    "resourceTypes": [
      {
        "aliases": null,
        "apiProfiles": null,
        "apiVersions": null,
        "capabilities": null,
        "defaultApiVersion": null,
        "locationMappings": null,
        "locations": [
          "canadacentral"
        ],
        "properties": null,
        "resourceType": "flexibleServers",
        "zoneMappings": null
      },
```


},
{
"aliases": null,
"apiProfiles": null,
"apiVersions": null,
"capabilities": null,
"defaultApiVersion": null,
"locationMappings": null,
"locations": [
null
],
"properties": null,
"resourceType": "flexibleServers/firewallRules",
"zoneMappings": null
},
{
"aliases": null,
"apiProfiles": null,
"apiVersions": null,
"capabilities": null,
"defaultApiVersion": null,
"locationMappings": null,
"locations": [
null
],
"properties": null,
"resourceType": "flexibleServers/databases",
"zoneMappings": null
}
]
}
],
"provisioningState": "Succeeded",
"templateHash": "6092500531197253688",
"templateLink": null,
"timestamp": "2025-06-30T17:30:03.258744+00:00",
"validatedResources": null,
"validationLevel": null
},


"validationLevel": null
},
"resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
"tags": null,
"type": "Microsoft.Resources/deployments"
marianne [~]$

Resource Group and two Databases were created as expected.

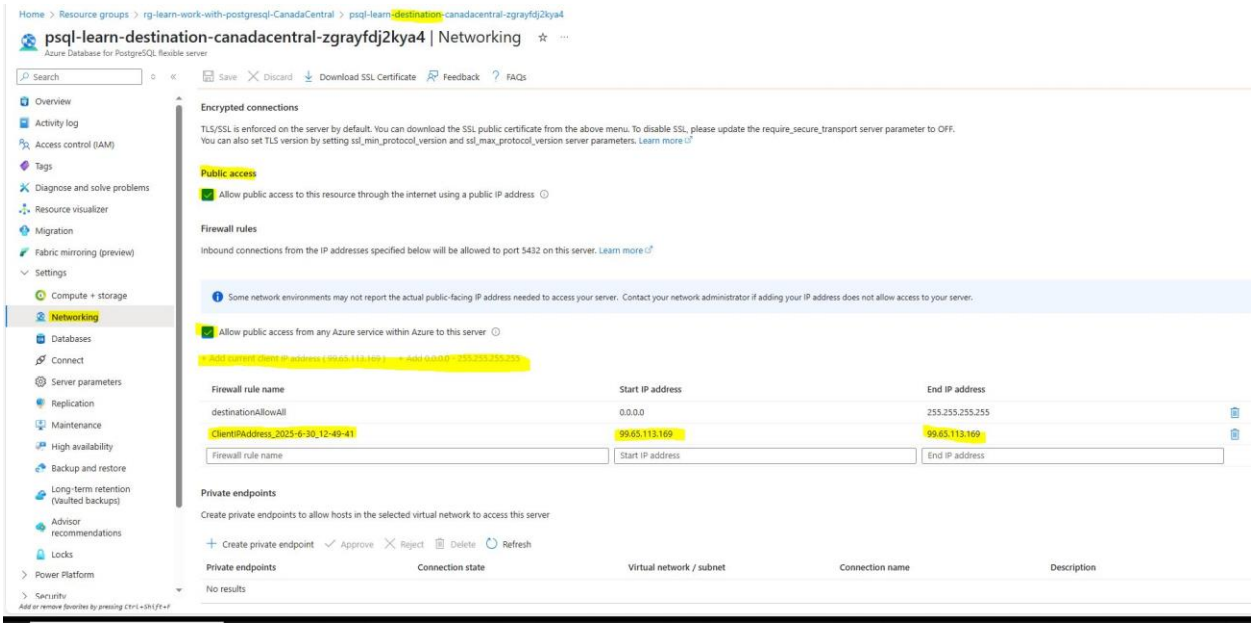


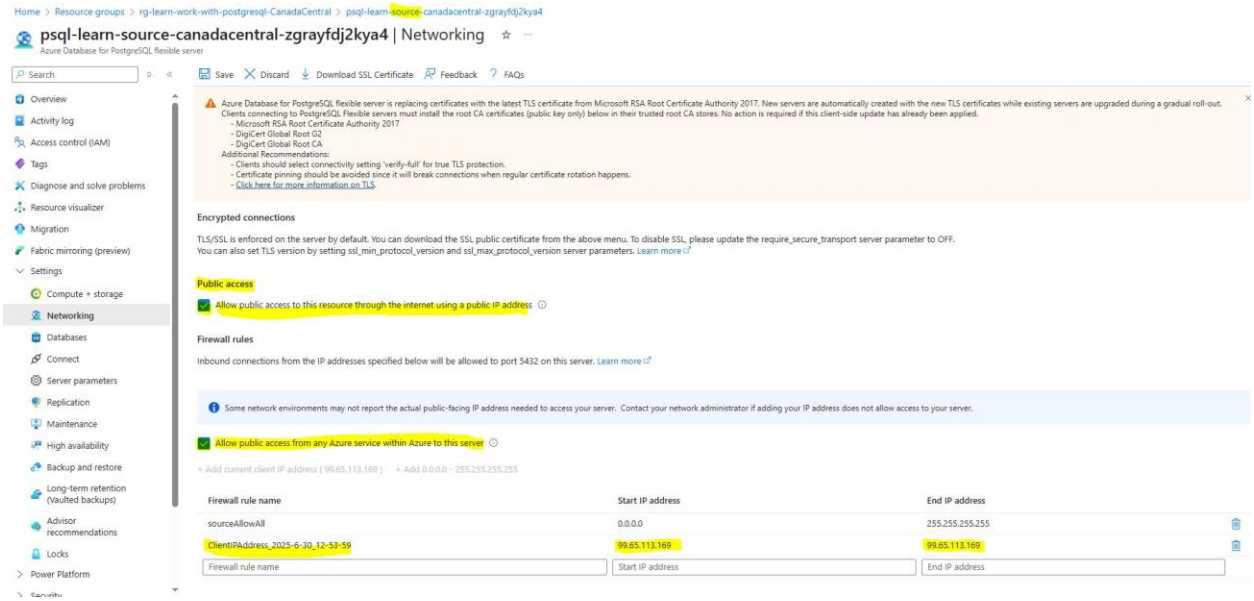
Step 7. Close the Cloud Shell pane once your resource deployment is complete.

Step 8. On the Azure portal, review the names of the two new Azure Database for PostgreSQL servers. Notice that when you list the databases of the source server it includes the **adventureworks** database but the destination one doesn't.

Step 9. Under the **Networking** section of *both* servers,

a. Select **+ Add current IP address (xxx.xxx.xxx)** and **Save**.
b. Select the **Allow public access from any Azure service within Azure to this server** checkbox.
c. Select the **Allow public access to this resource through the internet using a public IP address** checkbox.



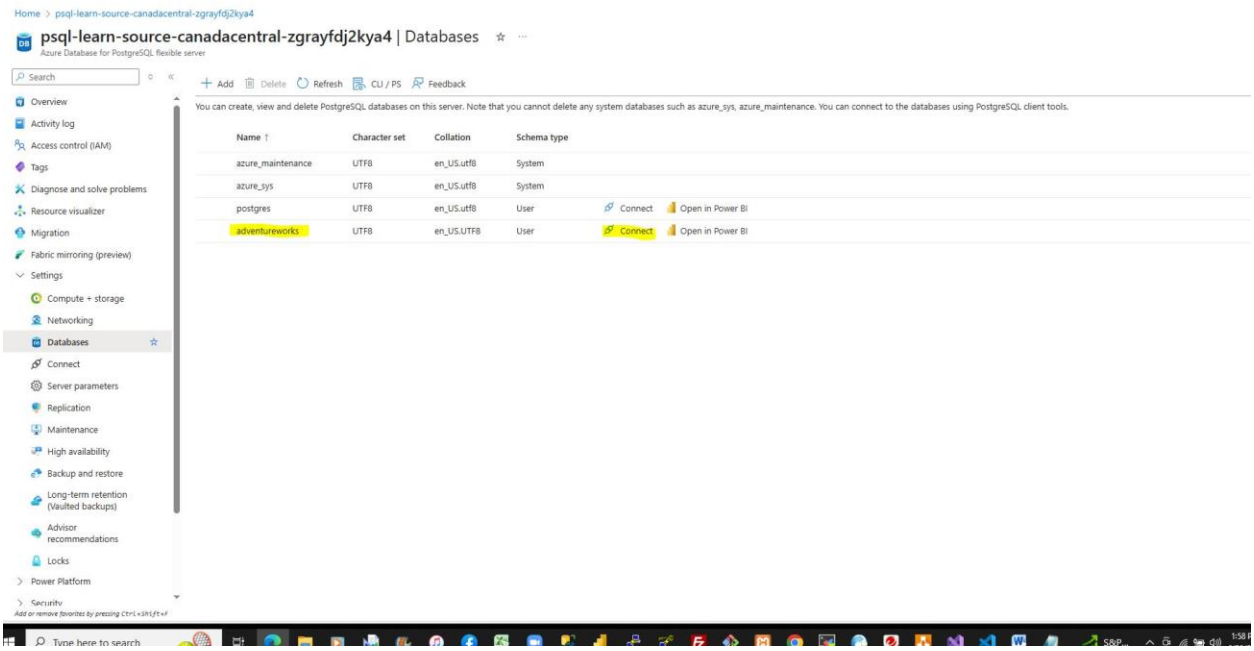


Create a database on the Azure Database for PostgreSQL server

Now, we need to set up the database, which you will migrate to the Azure Database for PostgreSQL Flexible Server. This step needs to be completed on your source PostgreSQL Server instance, which will need to be accessible to the Azure Database for PostgreSQL Flexible Server in order to complete this lab.

First, we need to create an empty database, which we will create a table and then load it with data.

1. In the Azure portal, navigate to the newly created source Azure Database for PostgreSQL server (***psql-learn-source***-location-uniquevalue).
2. In the resource menu, under **Settings**, select **Databases** select **Connect** for the adventureworks database. Connect in the left pane under Connect for cmd line.



3. At the "Password for user pgAdmin" prompt in the Cloud Shell, enter the randomly generated password for the **pgAdmin** login.

   Once logged in, the psql prompt for the adventureworks database is displayed.

4. Run the following command to create the production.workorder table for loading in data:

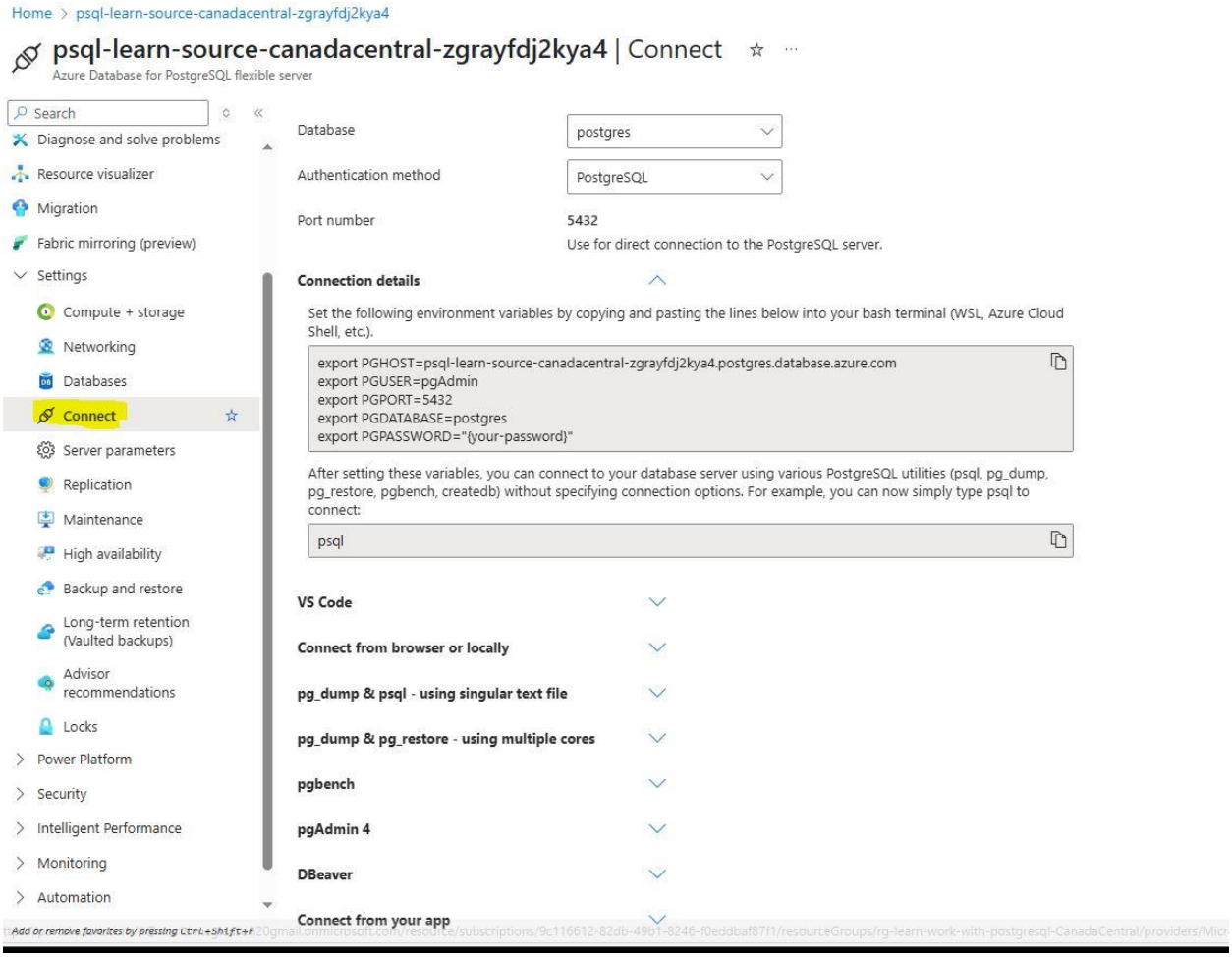
```
DROP SCHEMA IF EXISTS production CASCADE;
CREATE SCHEMA production;

DROP TABLE IF EXISTS production.workorder;
CREATE TABLE production.workorder
(
    workorderid integer NOT NULL,
    productid integer NOT NULL,
    orderqty integer NOT NULL,
```

```
    scrappedqty smallint NOT NULL,
    startdate timestamp without time zone NOT NULL,
    enddate timestamp without time zone,
    duedate timestamp without time zone NOT NULL,
    scrapreasonid smallint,
    modifieddate timestamp without time zone NOT NULL DEFAULT now()
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

```
\COPY production.workorder FROM 'mslearn-postgresql/Allfiles/Labs/10/Lab10_workorder.csv' CSV HEADER
```

The command output should be `COPY 72101`, indicating that 72101 rows were written into the table from the CSV file.

Close the **Cloud Shell**.

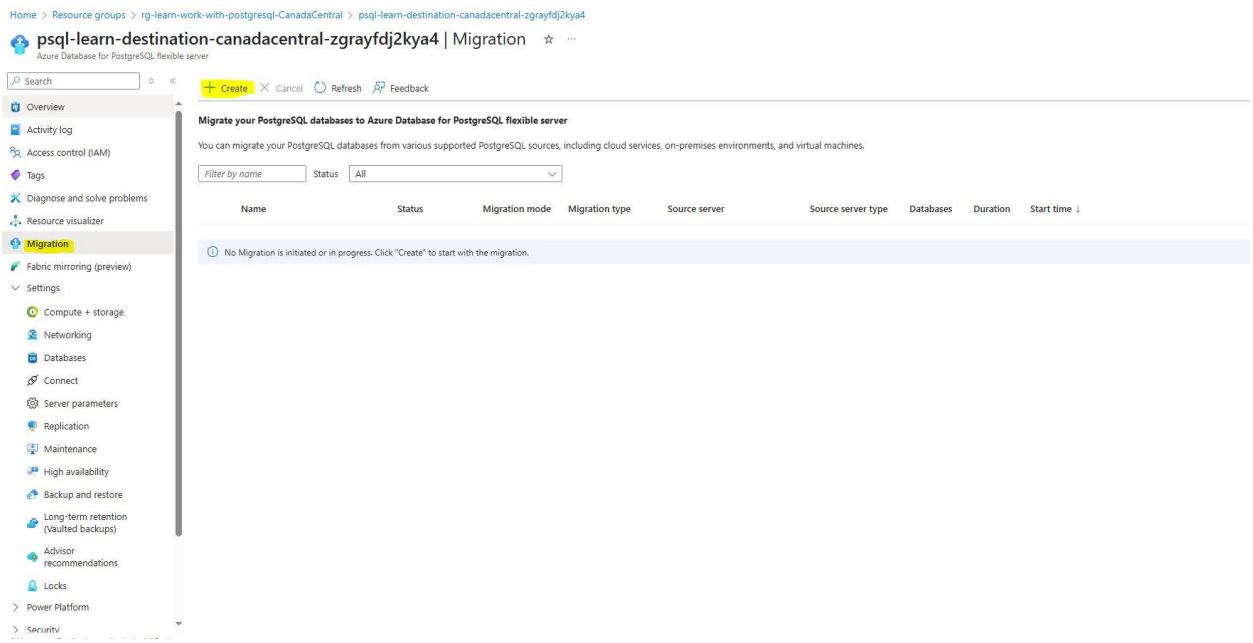

Home > psql-learn-source-canadacentral-zgrayfdj2kya4
psql-learn-source-canadacentral-zgrayfdj2kya4 | Connect
Azure Database for PostgreSQL flexible server
Search
Diagnose and solve problems
Resource visualizer
Migration
Fabric mirroring (preview)
Settings
Compute + storage
Networking
Databases
Connect
Server parameters
Replication
Maintenance
High availability
Backup and restore
Long-term retention (Vaulted backups)
Advisor recommendations
Locks
Power Platform
Security
Intelligent Performance
Monitoring
Automation
Database
postgres
Authentication method
PostgreSQL
Port number
5432
Use for direct connection to the PostgreSQL server.
Connection details
Set the following environment variables by copying and pasting the lines below into your bash terminal (WSL, Azure Cloud Shell, etc.).
export PGHOST=psql-learn-source-canadacentral-zgrayfdj2kya4.postgres.database.azure.com
export PGUSER=pgAdmin
export PGPORT=5432
export PGDATABASE=postgres
export PGPASSWORD="{your-password}"
After setting these variables, you can connect to your database server using various PostgreSQL utilities (psql, pg_dump, pg_restore, pgbench, createdb) without specifying connection options. For example, you can now simply type psql to connect:
psql
VS Code
Connect from browser or locally
pg_dump & psql - using singular text file
pg_dump & pg_restore - using multiple cores
pgbench
pgAdmin 4
DBeaver
Connect from your app
Add or remove favorites by pressing Ctrl+Shift+F

```
marianne [ ~ ]$ export PGHOST=psql-learn-source-canadacentral-zgrayfdj2kya4.postgres.database.azure.com
export PGUSER=pgAdmin
export PGPORT=5432
export PGDATABASE=adventureworks
export PGPASSWORD=
marianne [ ~ ]$ psql
psql (16.7, server 16.9)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

adventureworks=> DROP SCHEMA IF EXISTS production CASCADE;
    CREATE SCHEMA production;

    DROP TABLE IF EXISTS production.workorder;
    CREATE TABLE production.workorder
    (
        workorderid integer NOT NULL,
        productid integer NOT NULL,
        orderqty integer NOT NULL,
        scrappedqty smallint NOT NULL,
        startdate timestamp without time zone NOT NULL,
        enddate timestamp without time zone,
        duedate timestamp without time zone NOT NULL,
        scrapreasonid smallint,
        modifieddate timestamp without time zone NOT NULL DEFAULT now()
    )
    WITH (
        OIDS = FALSE
    )
    TABLESPACE pg_default;
NOTICE:  schema "production" does not exist, skipping
DROP SCHEMA
CREATE SCHEMA
NOTICE:  table "workorder" does not exist, skipping
DROP TABLE
CREATE TABLE
adventureworks=> \COPY production.workorder FROM 'mslearn-postgresql/Allfiles/Labs/10/Lab10_workorder.csv' CSV HEADER
COPY 72101
adventureworks=>
```

Create Database Migration Project in Azure Database for PostgreSQL Flexible Server

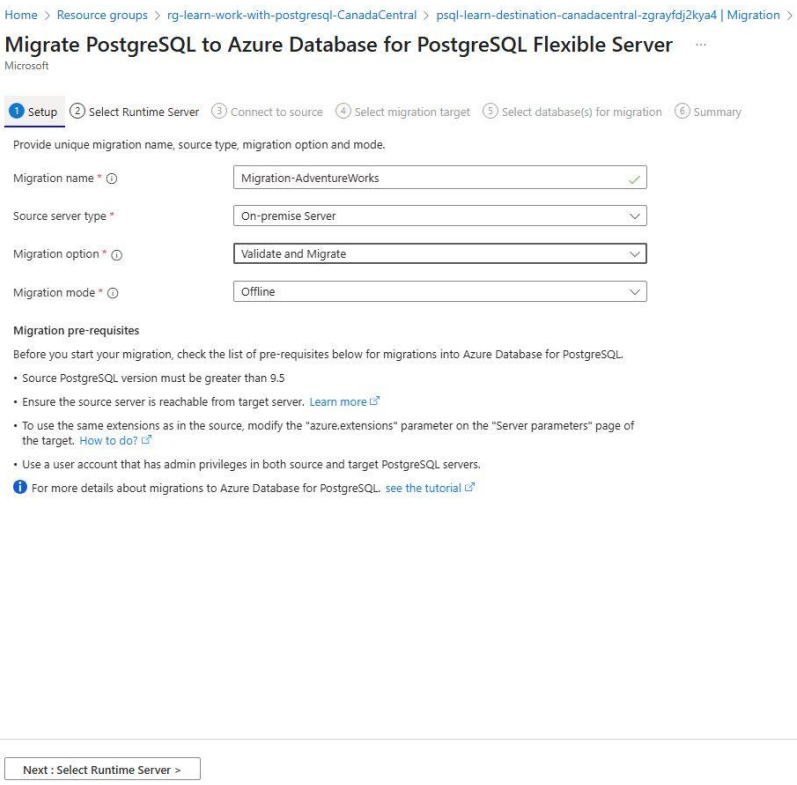
Step 1. On the destination server, select **Migration** from the menu on the left of the flexible server blade.

Click on the **+ Create** option at the top of the **Migration** blade



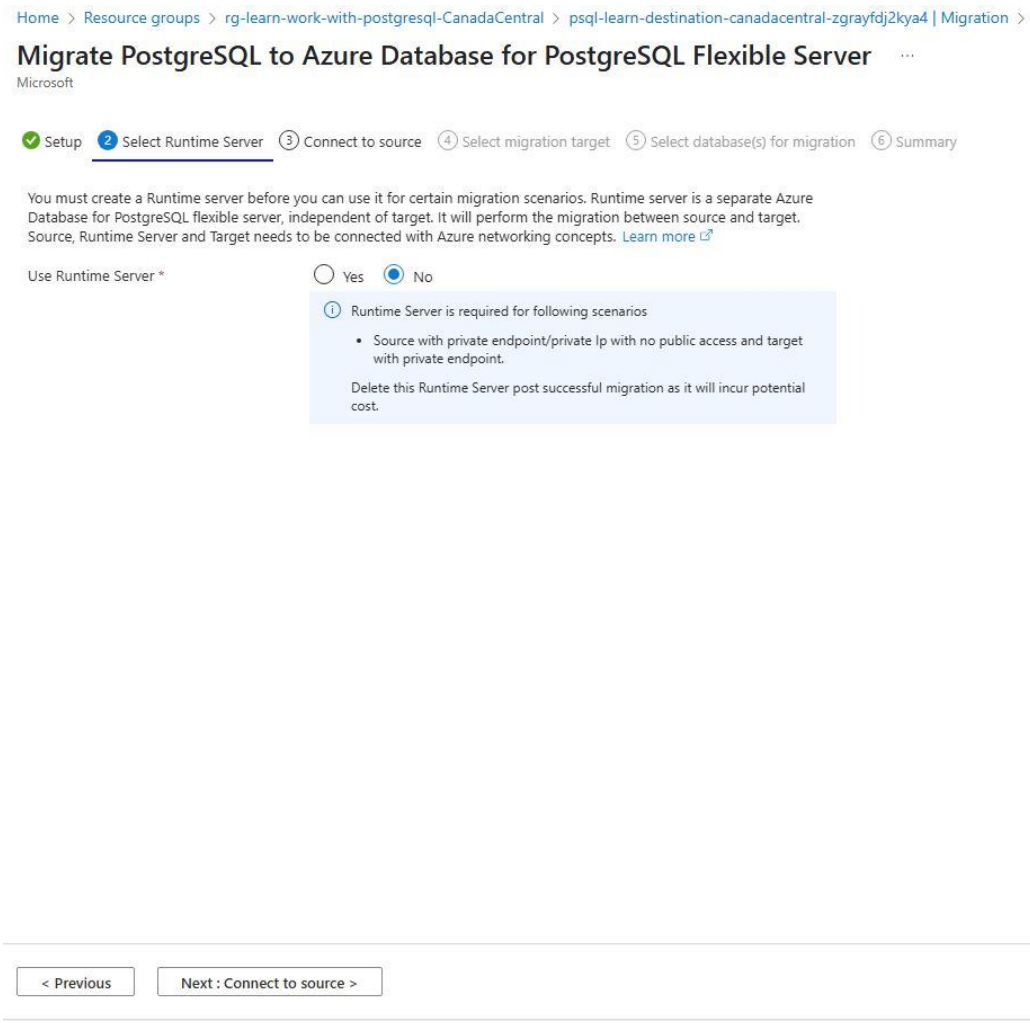
Step 3. On the **Setup** tab, enter each field as follows:

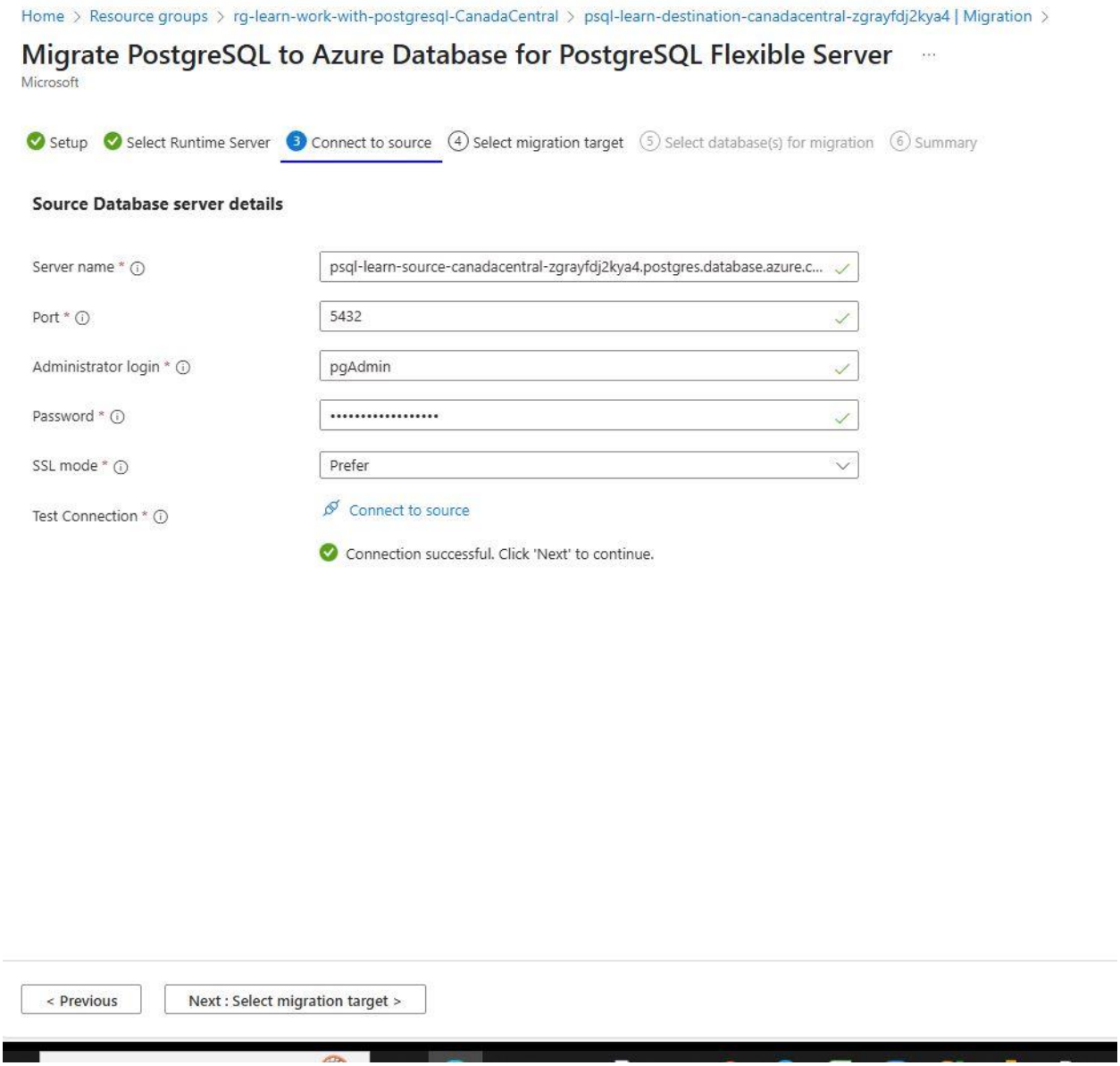
a. Migration name - **Migration-AdventureWorks**.
b. Source server type - For this lab, no matter if you are doing a migration from on-premise or from an Azure Database for PostgreSQL, select **On-premise Server**. On a production environment, pick the correct source server type.
c. Migration option - **Validate and Migrate**.
d. Migration mode - **Offline**.
e. Select **Next: Select Runtime Server >**.
f. Select **No** for *Use Runtime Server*.
g. Select **Connect to source >**.



Step 4. For migration from an Azure Database for PostgreSQL- On the **Connect to source** tab, enter each field as follows:

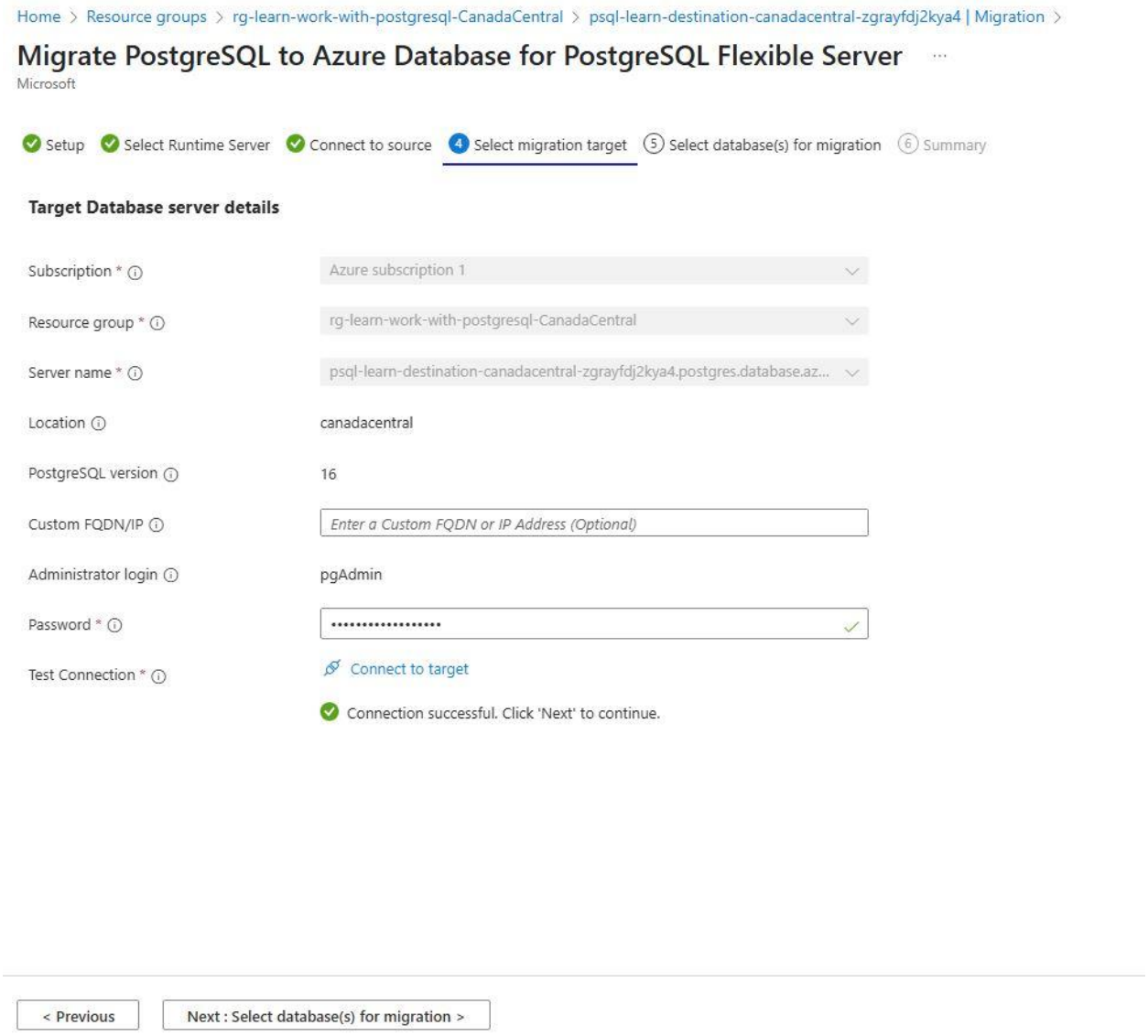
a. Server name - The address of your server that you are using as the source.
b. Port - The port your instance of PostgreSQL uses on your source server (default of 5432).
c. Server admin login name - The name of an admin user for your PostgreSQL instance (default pgAdmin).
d. Password - The password for the PostgreSQL admin user you specified in the previous step.
e. SSL mode - Prefer.
f. Click on the **Connect to source** option to validate the connectivity details provided.
g. Click on the **Next: Select migration target** button to progress.


Home > Resource groups > rg-learn-work-with-postgresql-CanadaCentral > psql-learn-destination-canadacentral-zgrayfdj2kya4 | Migration >
Migrate PostgreSQL to Azure Database for PostgreSQL Flexible Server
Microsoft
Setup
2 Select Runtime Server
3 Connect to source
4 Select migration target
5 Select database(s) for migration
6 Summary
You must create a Runtime server before you can use it for certain migration scenarios. Runtime server is a separate Azure Database for PostgreSQL flexible server, independent of target. It will perform the migration between source and target. Source, Runtime Server and Target needs to be connected with Azure networking concepts. Learn more
Use Runtime Server *
Yes
No
Runtime Server is required for following scenarios
Source with private endpoint/private Ip with no public access and target with private endpoint.
Delete this Runtime Server post successful migration as it will incur potential cost.
< Previous
Next : Connect to source >



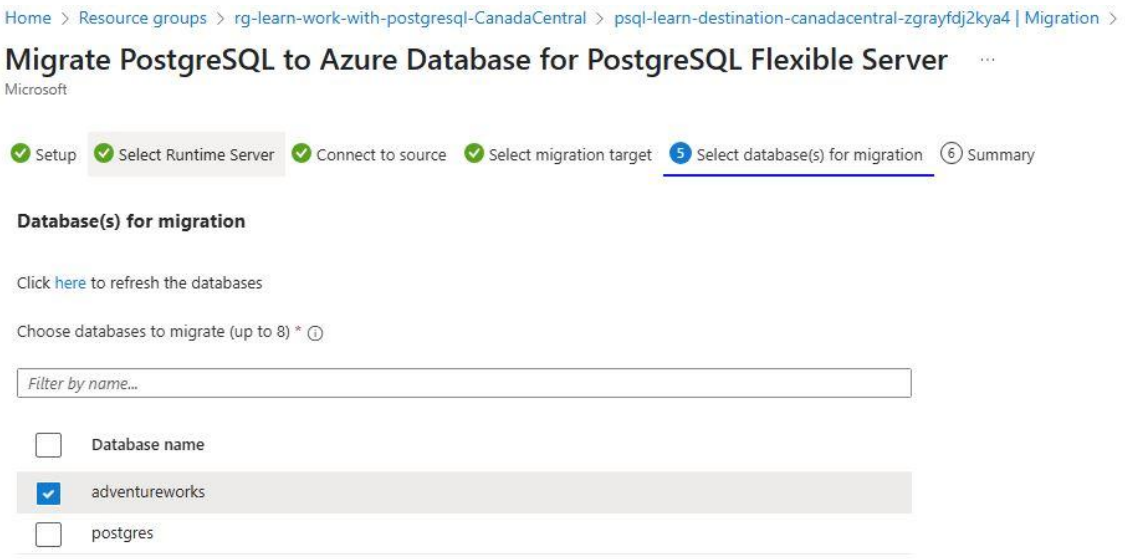
Step 5. The connectivity details should be automatically completed for the target server we are migrating to.

a. In the password field - enter the randomly generated password for the **pgAdmin** login you created with the bicep script.
b. Click on the **Connect to target** option to validate the connectivity details provided.
c. Click on the **Next : Select database(s) for migration >** button to progress.

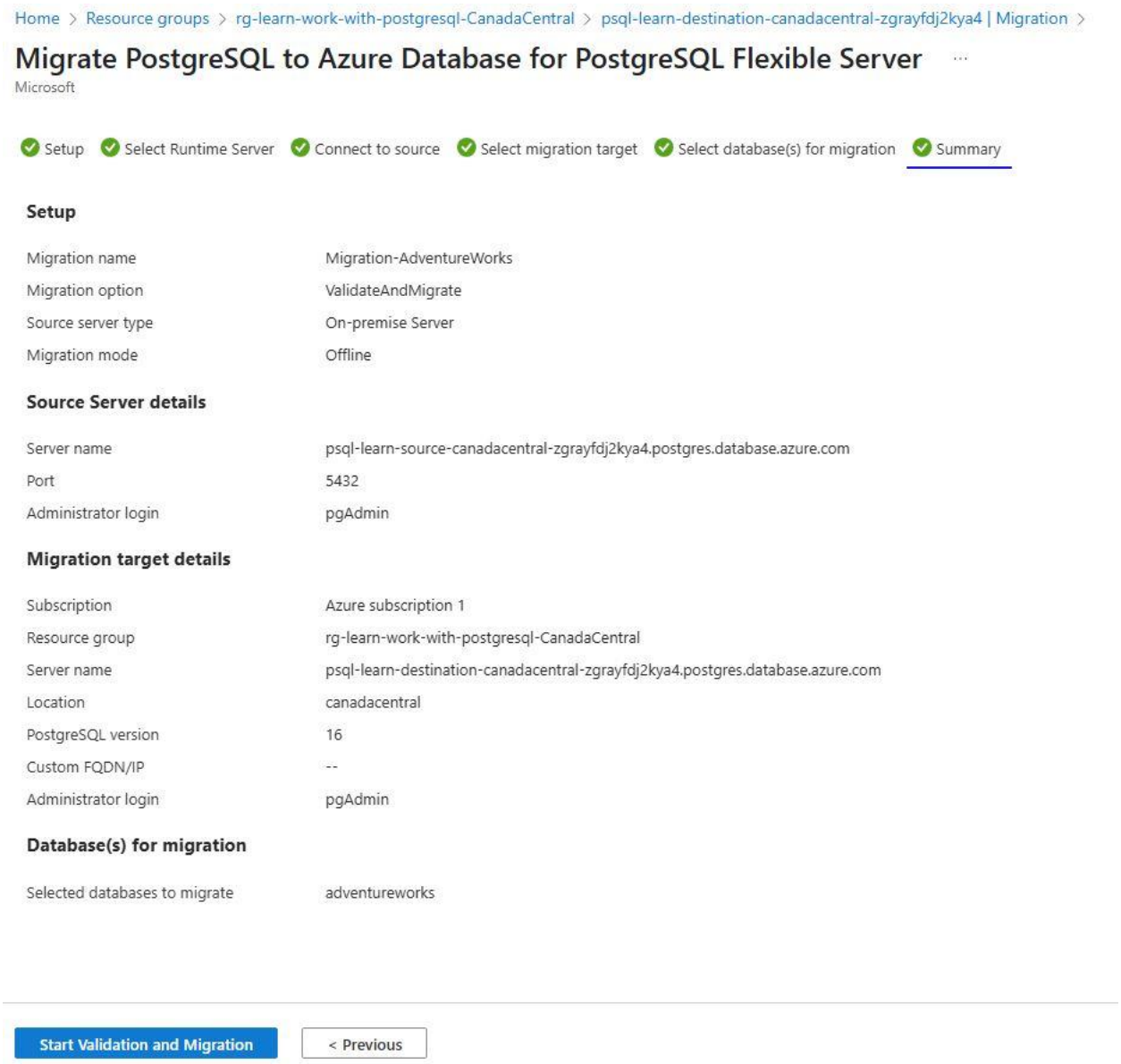


Step 6. On the **Select database(s) for migration** tab, select the **adventureworks** from the source server you want to migrate to the flexible server.

Step 7. Click on the **Next : Summary >** button to progress and review the data provided.

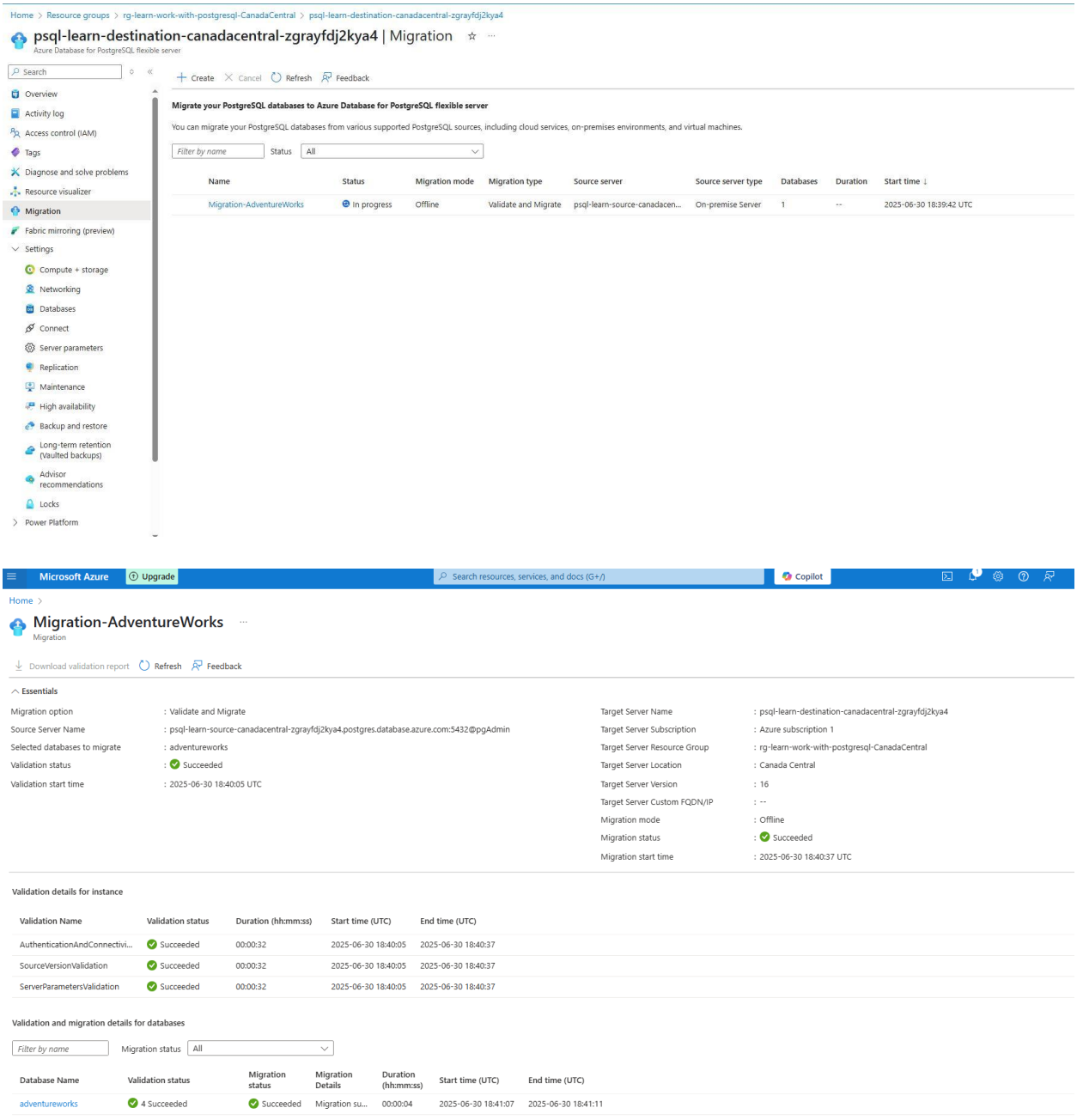


Step 8. On the **Summary** tab, review the information and then click the **Start Validation and Migration** button to start the migration to the flexible server.

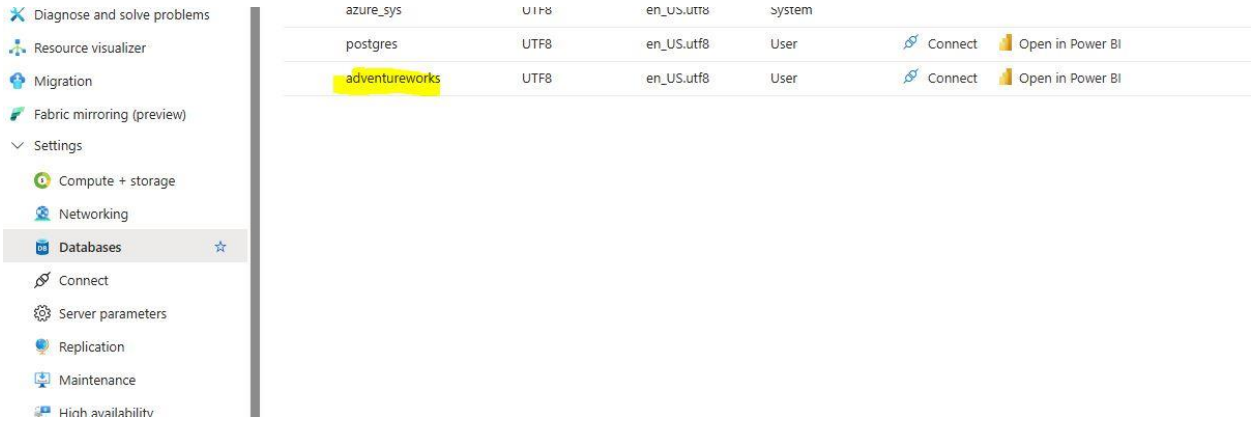


Step 9. On the **Migration** tab, you can monitor the migration progress by using the **Refresh** button in the top menu to view the progress through the validation and migration process.

a. By clicking on the **Migration-AdventureWorks** activity, you can view detailed information about the migration activity's progress.



Step 10. Once the migration is complete, check the destination server, you should now find the **adventureworks** database also listed under that server.



Once the migration process is complete, we can perform post-migration tasks such as data validation in the new database and configuring high availability before pointing the application at the database and turning it on again.

Validation:

```
adventureworks-> \dt production.*
            List of relations
   Schema   |    Name    | Type  |  Owner
------------+------------+-------+---------
 production | workorder | table | pgAdmin
```

72101 rows is the correct rows that were imported into the table. The migration worked successfully!!

```
adventureworks=> SELECT COUNT(*) FROM production.workorder;
 72101
```

Clean-up:

Delete the **rg-learn-work-with-postgresql-canadacentral** resource group to remove all resources we deployed as part of this use case.