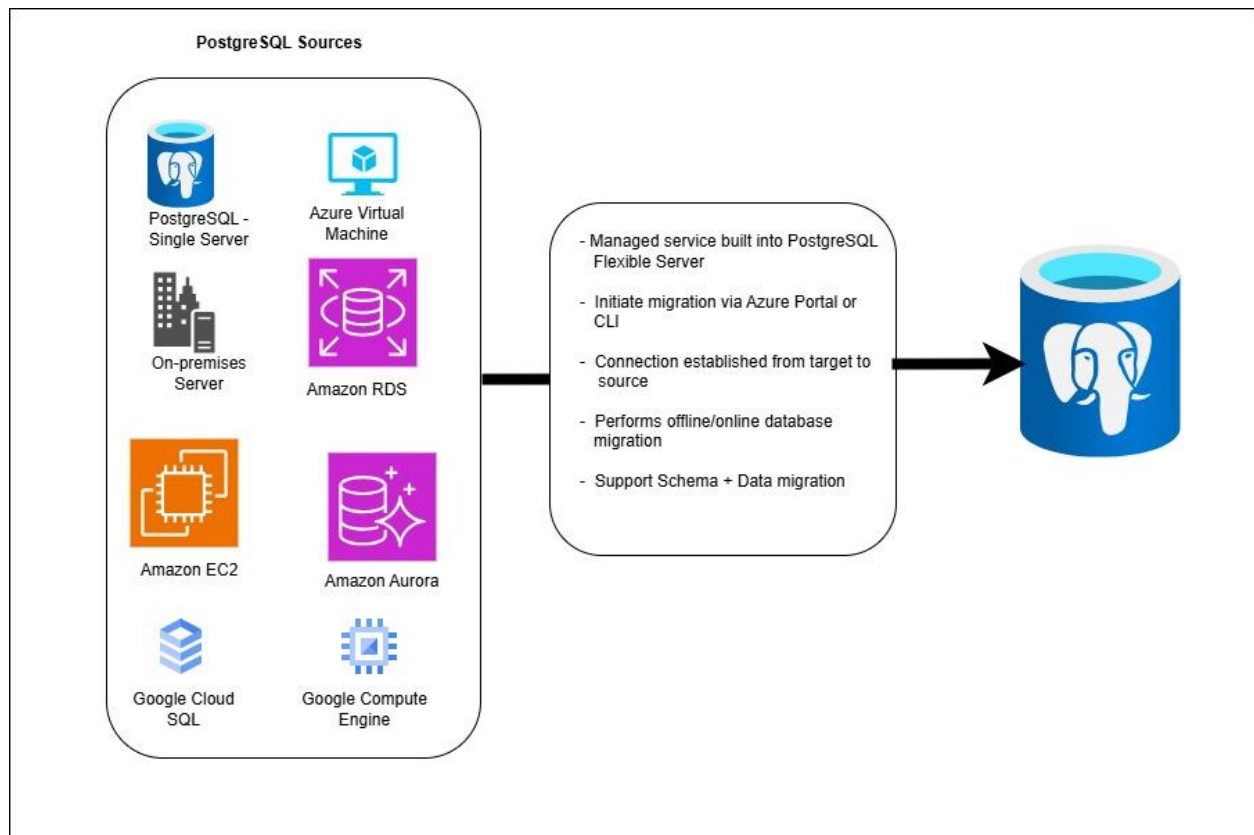


# Azure PostgreSQL Database Migration

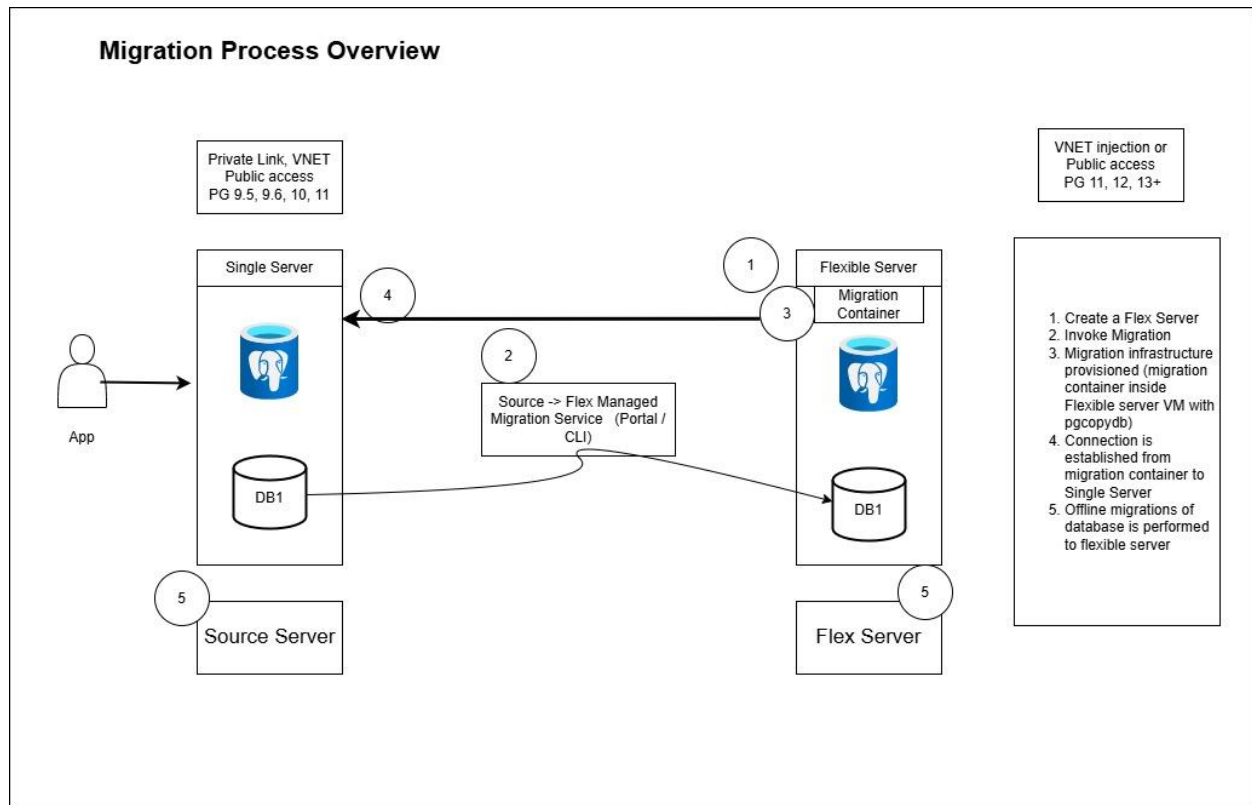
## USE CASE:

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.

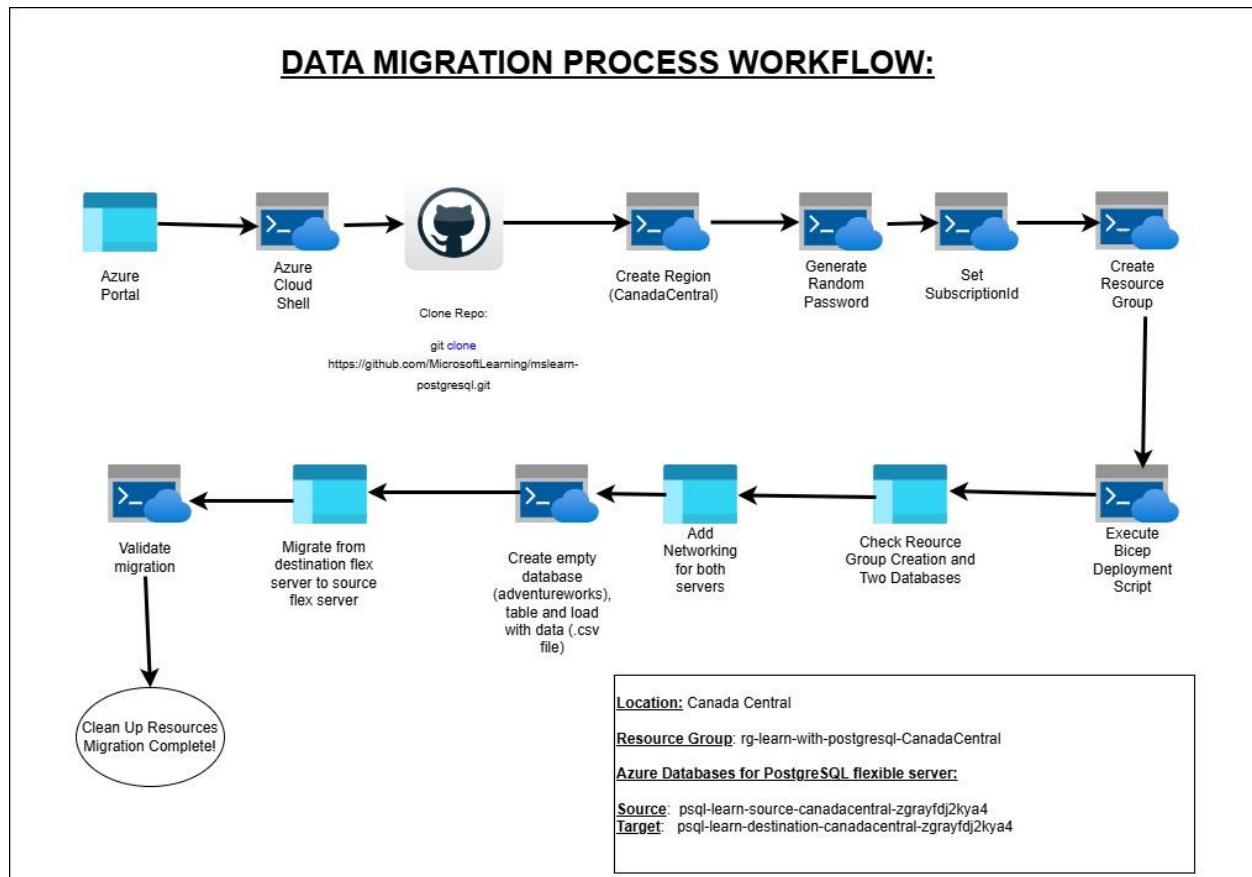
Below are the PostgreSQL sources you can migrate by using the migration service in Azure Database for PostgreSQL. All environments can be transitioned to Azure Database for PostgreSQL:



## 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.

The screenshot displays the Azure portal's main dashboard. At the top, there's a search bar and a 'Copilot' button. Below this, the 'Azure services' section features icons for various services like 'Create a resource', 'App Services', 'Microsoft Entra ID', 'Azure DevOps organizations', 'Load balancers', 'Subscriptions', 'Language understanding', 'Key vaults', 'Cost Management', and 'More services'. The 'Resources' section has tabs for 'Recent' and 'Favorite', showing a table of recent resources:

Name	Type	Last Viewed
Azure subscription 1	Subscription	2 days ago
NetworkWatcherRG	Resource group	6 days ago

Below the resources table is a 'Navigate' section with icons for 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'. At the bottom, the 'Tools' section is visible. A 'Welcome to Azure Cloud Shell' dialog box is open, prompting the user to select a shell (Bash or PowerShell). Below this, a 'Getting started' dialog box is shown, allowing the user to select a subscription and configure storage options.

**Welcome to Azure Cloud Shell**

Select Bash or PowerShell. You can change shells any time via the environment selector in the Cloud Shell toolbar. The most recently used environment will be the default for your next session.

**Getting started**

Select a subscription to get started. You can optionally mount a storage account to persist files between sessions. [Learn more](#)

☒ No storage account required ⓘ  
☐ Mount storage account ⓘ

Subscription \*  
Azure subscription 1

☐ Use an existing private virtual network [Learn more](#)

**Apply** Previous

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
```

```
Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help
Subscription used to launch your CloudShell 9c116612-82db-49b1-8246-f0eddbaf87f1 is not registered to Microsoft.CloudShell Namespace. Please follow these
r. In future, unregistered subscriptions will have restricted access to CloudShell service.

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
marianne [ ~ ]$ git clone https://github.com/MicrosoftLearning/mslearn-postgresql.git
Cloning into 'mslearn-postgresql'...
remote: Enumerating objects: 1655, done.
remote: Counting objects: 100% (292/292), done.
remote: Compressing objects: 100% (168/168), done.
remote: Total 1655 (delta 211), reused 125 (delta 109), pack-reused 1363 (from 1)
Receiving objects: 100% (1655/1655), 5.27 MiB | 19.48 MiB/s, done.
Resolving deltas: 100% (922/922), done.
marianne [ ~ ]$ REGION=eastus
marianne [ ~ ]$ RG_NAME=rg-learn-work-with-postgresql-$REGION
marianne [ ~ ]$ 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 $ADMIN_PASSWORD
Your randomly generated PostgreSQL admin user's password is:
1***
marianne [ ~ ]$
```

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 subscriptionId

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

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

```
marianne [ ~ ]$ az account set --subscription 9c116612-82db-49b1-8246-f0eddbaf87f1
marianne [ ~ ]$ az group create --name $RG_NAME --location $REGION
{
  "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-eastus",
  "location": "eastus",
  "managedBy": null,
  "name": "rg-learn-work-with-postgresql-eastus",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
marianne [ ~ ]$
```

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
The configuration value of bicep.use_binary_from_path has been set to 'false'.
{"status": "failed", "error": {"code": "DeploymentFailed", "target": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-eastus/providers/Microsoft.Resources/deployments/de
ploy-postgresql-server-migration", "message": "At least one resource deployment operation failed. Please list deployment operations for details. Please see https://aka.ms/arm-deployment-operations for usage details.",
"details": [{"code": "ResourceDeploymentFailure", "target": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-eastus/providers/Microsoft.DBforPostgreSQL/flexibleServers/
psql-learn-source-eastus-usuggez7sgbu", "message": "The resource write operation failed to complete successfully, because it reached terminal provisioning state 'failed'.", "details": [{"code": "LocationIsOfferRestrict
ed", "message": "Subscriptions are restricted from provisioning in location 'eastus'. Try again in a different location. For exceptions to this rule, see how to request a quota increase in https://aka.ms/postgres-
request-quota-increase."}]}, {"code": "ResourceDeploymentFailure", "target": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-eastus/providers/Microsoft.DBforPostgreSQL/
flexibleServers/psql-learn-destination-eastus-usuggez7sgbu", "message": "The resource write operation failed to complete successfully, because it reached terminal provisioning state 'failed'.", "details": [{"code": "
LocationIsOfferRestricted", "message": "Subscriptions are restricted from provisioning in location 'eastus'. Try again in a different location. For exceptions to this rule, see how to request a quota increase in htt
ps://aka.ms/postgres-request-quota-increase."}]}}]}
```

```
marianne [ ~ ]$ REGION=CanadaCentral
marianne [ ~ ]$ RG_NAME=rg-learn-work-with-postgresql-$REGION
marianne [ ~ ]$ 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 $ADMIN_PASSWORD
Your randomly generated PostgreSQL admin user's password is:
ZrDp9LP5HofcJuG6Q1
marianne [ ~ ]$ az account set --subscription 9c116612-82db-49b1-8246-f0eddbaf87f1
marianne [ ~ ]$ az group create --name $RG_NAME --location $REGION
{
  "id": "/subscriptions/9c116612-82db-49b1-8246-f0eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral",
  "location": "canadacentral",
  "managedBy": null,
  "name": "rg-learn-work-with-postgresql-CanadaCentral",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```



```

marianne [ ~ ]$ az deployment group create --resource-group $RG_NAME --template-file "mslearn-postgresql/Allfiles/Labs/Shared/deploy-postgresql-server-migration.bicep" --parameters adminLogin-pg/admin adminLoginPa
ssword-$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": "e643ecf-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/destinationAllowAllAzureServicesAndResourcesWithinAzureIps",
        "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral",
        "resourceName": "psql-learn-destination-canadacentral-zgrayfdj2kya4/destinationAllowAllAzureServicesAndResourcesWithinAzureIps",
        "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": "PT0M17.07222115",
    "error": null,
    "mode": "Incremental",
    "onErrorDeployment": null,
    "outputResources": [
  
```



```

    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f8eddbaf87f1/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-f8eddbaf87f1/resourceGroups/rg-learn-work-with-postgresql-CanadaCentral/providers/Microsoft.DBforPostgreSQL/flexibleServers/psql-learn-destination-canadacentra
l-zgrayfdj2kya4/firewallRules/destinationAllowAllAzureServicesAndResourcesWithinAzureIps",
      "resourceGroup": "rg-learn-work-with-postgresql-CanadaCentral"
    },
    {
      "id": "/subscriptions/9c116612-82db-49b1-8246-f8eddbaf87f1/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-f8eddbaf87f1/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-f8eddbaf87f1/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-f8eddbaf87f1/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-f8eddbaf87f1/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
},

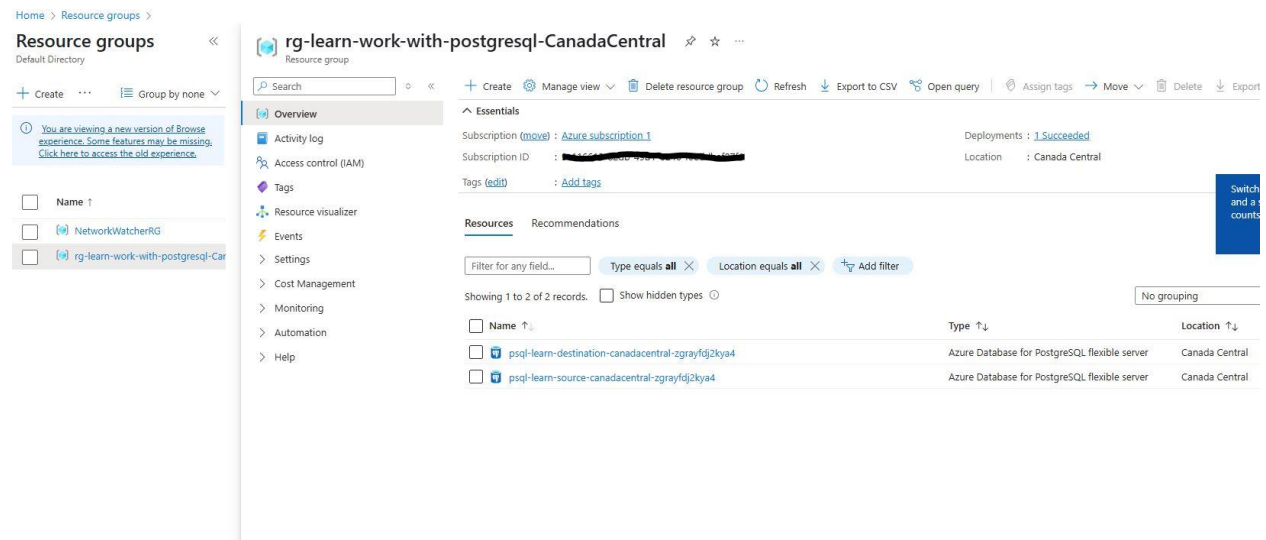
```

```

    "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,

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

Home > Resource groups > rg-learn-work-with-postgresql-CanadaCentral > psql-learn-destination-canadacentral-zgrayfdj2kya4

psql-learn-destination-canadacentral-zgrayfdj2kya4 | Networking

Azure Database for PostgreSQL flexible server

Search

Overview  
Activity log  
Access control (IAM)  
Tags  
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

Encrypted connections

TLS/SSL is enforced on the server by default. You can download the SSL public certificate from the above menu. To disable SSL, please update the require\_secure\_transport server parameter to OFF. You can also set TLS version by setting ssl\_min\_protocol\_version and ssl\_max\_protocol\_version server parameters. [Learn more](#)

**Public access**

☒ Allow public access to this resource through the internet using a public IP address

**Firewall rules**

Inbound connections from the IP addresses specified below will be allowed to port 5432 on this server. [Learn more](#)

Some network environments may not report the actual public-facing IP address needed to access your server. Contact your network administrator if adding your IP address does not allow access to your server.

☒ Allow public access from any Azure service within Azure to this server

+ Add current client IP address (99.65.113.169) + Add 0.0.0.0 - 255.255.255.255

Firewall rule name	Start IP address	End IP address
destinationAllowAll	0.0.0.0	255.255.255.255
ClientIpAddress_2025-6-30-12-49-41	99.65.113.169	99.65.113.169

Private endpoints

Create private endpoints to allow hosts in the selected virtual network to access this server

+ Create private endpoint ✓ Approve ✗ Reject 🗑 Delete ↻ Refresh

Private endpoints	Connection state	Virtual network / subnet	Connection name	Description
No results				

Add or remove endpoints by pressing Ctrl+Shift+P

Home > Resource groups > rg-learn-work-with-postgresql-CanadaCentral > psql-learn-source-canadacentral-zgrayfdj2kya4

psql-learn-source-canadacentral-zgrayfdj2kya4 | Networking

Azure Database for PostgreSQL flexible server

Search

Overview  
Activity log  
Access control (IAM)  
Tags  
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

Azure Database for PostgreSQL flexible server is replacing certificates with the latest TLS certificate from Microsoft RSA Root Certificate Authority 2017. New servers are automatically created with the new TLS certificates while existing servers are upgraded during a gradual roll-out. Clients connecting to PostgreSQL Flexible servers must install the root CA certificates (public key only) below in their trusted root CA stores. No action is required if this client-side update has already been applied.

- Microsoft RSA Root Certificate Authority 2017
- DigiCert Global Root G2
- DigiCert Global Root CA

Additional Recommendations:

- Clients should select connectivity setting 'verify-full' for true TLS protection.
- Certificate pinning should be avoided since it will break connections when regular certificate rotation happens.
- [Click here for more information on TLS](#)

**Encrypted connections**

TLS/SSL is enforced on the server by default. You can download the SSL public certificate from the above menu. To disable SSL, please update the require\_secure\_transport server parameter to OFF. You can also set TLS version by setting ssl\_min\_protocol\_version and ssl\_max\_protocol\_version server parameters. [Learn more](#)

**Public access**

☒ Allow public access to this resource through the internet using a public IP address

**Firewall rules**

Inbound connections from the IP addresses specified below will be allowed to port 5432 on this server. [Learn more](#)

Some network environments may not report the actual public-facing IP address needed to access your server. Contact your network administrator if adding your IP address does not allow access to your server.

☒ Allow public access from any Azure service within Azure to this server

+ Add current client IP address (99.65.113.169) + Add 0.0.0.0 - 255.255.255.255

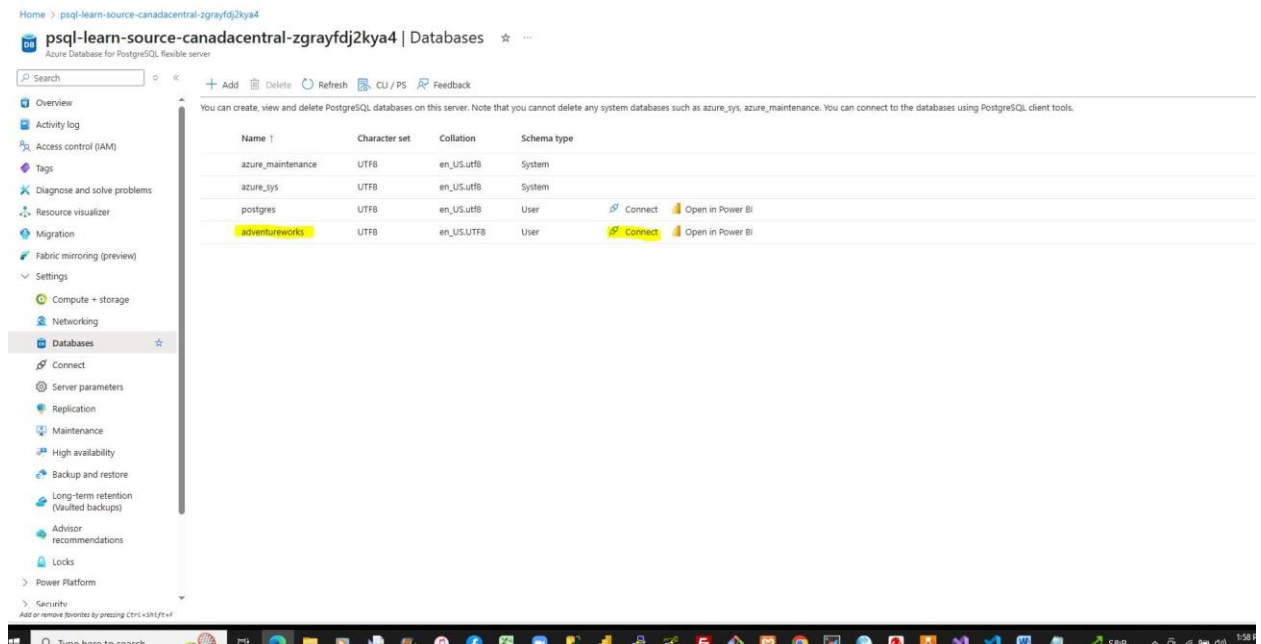
Firewall rule name	Start IP address	End IP address
sourceAllowAll	0.0.0.0	255.255.255.255
ClientIpAddress_2025-6-30-12-53-59	99.65.113.169	99.65.113.169

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**.



- 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**

```

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;

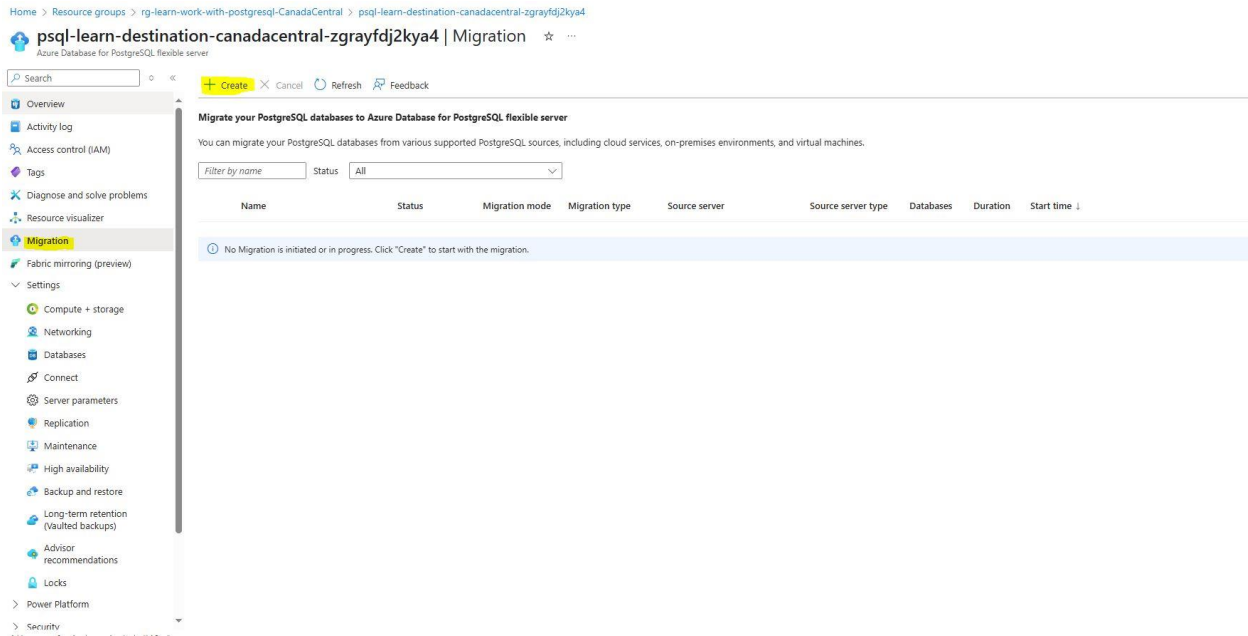
DROP TABLE IF EXISTS production.workorder;
CREATE TABLE production.workorder
(
    workorderid integer NOT NULL,
    productid integer NOT NULL,
    orderqty integer NOT NULL,
    scrappedqty smallint NOT NULL,
    startdate timestamp without time zone NOT NULL,
    enddate timestamp without time zone,
    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:

- Migration name - **Migration-AdventureWorks**.
- 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.
- Migration option - **Validate and Migrate**.
- Migration mode - **Offline**.
- Select **Next: Select Runtime Server >**.
- Select **No** for *Use Runtime Server*.
- Select **Connect to source >**.

## Migrate PostgreSQL to Azure Database for PostgreSQL Flexible Server

Microsoft

1 Setup 2 Select Runtime Server 3 Connect to source 4 Select migration target 5 Select database(s) for migration 6 Summary

Provide unique migration name, source type, migration option and mode.

Migration name *	<input type="text" value="Migration-AdventureWorks"/>
Source server type *	<input type="text" value="On-premise Server"/>
Migration option *	<input type="text" value="Validate and Migrate"/>
Migration mode *	<input type="text" value="Offline"/>

### Migration pre-requisites

Before you start your migration, check the list of pre-requisites below for migrations into Azure Database for PostgreSQL.

- Source PostgreSQL version must be greater than 9.5
  - Ensure the source server is reachable from target server. [Learn more](#)
  - To use the same extensions as in the source, modify the "azure.extensions" parameter on the "Server parameters" page of the target. [How to do?](#)
  - Use a user account that has admin privileges in both source and target PostgreSQL servers.
- For more details about migrations to Azure Database for PostgreSQL, [see the tutorial](#)

Next : Select Runtime Server >

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

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

## Migrate PostgreSQL to Azure Database for PostgreSQL Flexible Server

Microsoft

1 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 >

## Migrate PostgreSQL to Azure Database for PostgreSQL Flexible Server

Microsoft

✓ Setup ✓ Select Runtime Server **3 Connect to source** 4 Select migration target 5 Select database(s) for migration 6 Summary

### Source Database server details

Server name * ⓘ	psql-learn-source-canadacentral-zgrayfdj2kya4.postgres.database.azure.c... ✓
Port * ⓘ	5432 ✓
Administrator login * ⓘ	pgAdmin ✓
Password * ⓘ	..... ✓
SSL mode * ⓘ	Prefer ✓
Test Connection * ⓘ	 <a href="#">Connect to source</a>
	✓ Connection successful. Click 'Next' to continue.

< Previous

Next : Select migration target >

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

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

## Migrate PostgreSQL to Azure Database for PostgreSQL Flexible Server ...

Microsoft

✓ Setup ✓ Select Runtime Server ✓ Connect to source **4 Select migration target** 5 Select database(s) for migration 6 Summary

### Target Database server details

Subscription * ⓘ	Azure subscription 1
Resource group * ⓘ	rg-learn-work-with-postgresql-CanadaCentral
Server name * ⓘ	psql-learn-destination-canadacentral-zgrayfdj2kya4.postgres.database.azure.com
Location ⓘ	canadacentral
PostgreSQL version ⓘ	16
Custom FQDN/IP ⓘ	<input type="text" value="Enter a Custom FQDN or IP Address (Optional)"/>
Administrator login ⓘ	pgAdmin
Password * ⓘ	<input type="password" value="....."/>
Test Connection * ⓘ	<a href="#">Connect to target</a>

✓ Connection successful. Click 'Next' to continue.

< Previous

Next : Select database(s) for migration >

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.



## Migrate PostgreSQL to Azure Database for PostgreSQL Flexible Server ...

Microsoft

✔ Setup   ✔ Select Runtime Server   ✔ Connect to source   ✔ Select migration target   **5 Select database(s) for migration**   6 Summary

### Database(s) for migration

Click [here](#) to refresh the databases

Choose databases to migrate (up to 8) \* ⓘ

Filter by name...

☐ Database name

☒ adventureworks

☐ postgres

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.

## Migrate PostgreSQL to Azure Database for PostgreSQL Flexible Server ...

Microsoft

✔ Setup   ✔ Select Runtime Server   ✔ Connect to source   ✔ Select migration target   ✔ Select database(s) for migration   ✔ Summary

### Setup

Migration name	Migration-AdventureWorks
Migration option	ValidateAndMigrate
Source server type	On-premise Server
Migration mode	Offline

### Source Server details

Server name	psql-learn-source-canadacentral-zgrayfdj2kya4.postgres.database.azure.com
Port	5432
Administrator login	pgAdmin

### Migration target details

Subscription	Azure subscription 1
Resource group	rg-learn-work-with-postgresql-CanadaCentral
Server name	psql-learn-destination-canadacentral-zgrayfdj2kya4.postgres.database.azure.com
Location	canadacentral
PostgreSQL version	16
Custom FQDN/IP	--
Administrator login	pgAdmin

### Database(s) for migration

Selected databases to migrate	adventureworks
-------------------------------	----------------

Start Validation and Migration

< Previous

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.

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

Home > Resource groups > rg-learn-work-with-postgresql-CanadaCentral > psql-learn-destination-canadacentral-zgrayfdj2kya4

## psql-learn-destination-canadacentral-zgrayfdj2kya4 | Migration

Azure Database for PostgreSQL flexible server

Search  + Create Cancel Refresh Feedback

- Overview
- Activity log
- Access control (IAM)
- Tags
- 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

### Migrate your PostgreSQL databases to Azure Database for PostgreSQL flexible server

You can migrate your PostgreSQL databases from various supported PostgreSQL sources, including cloud services, on-premises environments, and virtual machines.

Filter by name  Status  All

Name	Status	Migration mode	Migration type	Source server	Source server type	Databases	Duration	Start time
Migration-AdventureWorks	In progress	Offline	Validate and Migrate	psql-learn-source-canadacen...	On-premise Server	1	--	2025-06-30 18:39:42 UTC

Microsoft Azure Upgrade Search resources, services, and docs (G+/) Copilot

Home > Migration-AdventureWorks

Download validation report Refresh Feedback

### Essentials

Migration option	: Validate and Migrate	Target Server Name	: psql-learn-destination-canadacentral-zgrayfdj2kya4
Source Server Name	: psql-learn-source-canadacentral-zgrayfdj2kya4.postgres.database.azure.com:5432@pgAdmin	Target Server Subscription	: Azure subscription 1
Selected databases to migrate	: adventureworks	Target Server Resource Group	: rg-learn-work-with-postgresql-CanadaCentral
Validation status	: <span style="color: green;">✔</span> Succeeded	Target Server Location	: Canada Central
Validation start time	: 2025-06-30 18:40:05 UTC	Target Server Version	: 16
		Target Server Custom FQDN/IP	: --
		Migration mode	: Offline
		Migration status	: <span style="color: green;">✔</span> Succeeded
		Migration start time	: 2025-06-30 18:40:37 UTC

### Validation details for instance

Validation Name	Validation status	Duration (hh:mm:ss)	Start time (UTC)	End time (UTC)
AuthenticationAndConnectiv...	<span style="color: green;">✔</span> Succeeded	00:00:32	2025-06-30 18:40:05	2025-06-30 18:40:37
SourceVersionValidation	<span style="color: green;">✔</span> Succeeded	00:00:32	2025-06-30 18:40:05	2025-06-30 18:40:37
ServerParametersValidation	<span style="color: green;">✔</span> Succeeded	00:00:32	2025-06-30 18:40:05	2025-06-30 18:40:37

### Validation and migration details for databases

Filter by name  Migration status  All

Database Name	Validation status	Migration status	Migration Details	Duration (hh:mm:ss)	Start time (UTC)	End time (UTC)
adventureworks	<span style="color: green;">✔</span> 4 Succeeded	<span style="color: green;">✔</span> Succeeded	Migration su...	00:00:04	2025-06-30 18:41:07	2025-06-30 18:41:11

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

Diagnose and solve problems	azure_sys	UTF8	en_US.utf8	system		
Resource visualizer	postgres	UTF8	en_US.utf8	User	Connect	Open in Power BI
Migration	adventureworks	UTF8	en_US.utf8	User	Connect	Open in Power BI
Fabric mirroring (preview)						
Settings						
Compute + storage						
Networking						
Databases						
Connect						
Server parameters						
Replication						
Maintenance						
High availability						

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.