



Prepare for migration

The information below assumes you have [completed the provisioning of your PostgreSQL Flexible server](#) and have a working, available PostgreSQL database on Azure.

As mentioned in [Migration tools](#), the **AWS Schema Conversion Tool** (SCT), and the **AWS Database Migration Service** (DMS) are the tools we have identified and tested to convert and migrate your database and its' data to *either* Cloud used by BNZ (Azure, or AWS).

In the examples below we use these tools to migrate a test DB2 LUW database from an on-premises Windows Server to an Azure PostgreSQL database. You can adapt this procedure to migrate your particular database to an AWS or Azure database.

! INFO

Note: you need to customise any settings shown in the examples to suit your database, Landing Zone, and credentials. Our example settings will not work for your database.

Firstly, set up your AWS environment and accesses

Item	information
To use DMS, you need an AWS logon and Account	Request AWS Account from AWS Cloud Team
Identify your Source database	The example uses our on-premises Windows Server (SWVTCDBSTG01)
Identify your Target (destination) database on the Cloud of your choice (AWS or Azure)	The example uses our test Azure PostgreSQL Flexible Server, pg-bnz-sy0-npdis1-vwuj-dev-1

Item	information
Firewall access to allow SCT and DMS to access the source and target databases	On Prem Service Request AWS Firewall Service Request Azure Firewall Service Request
Create a local account in the source database that can be used by SCT and DMS	This will require someone with database administrator privileges. In our example the DB2 DBAs do this.
Create a local account in the target database that can be used by SCT and DMS	This ID will need admin. access to create, update and delete database components and data

- Our example Azure *Target* database is on our test Azure PostgreSQL Flexible Server, pg-bnz-sy0-npdis1-vwuj-dev-1

The screenshot shows the Azure portal interface for an Azure PostgreSQL Flexible Server. The browser address bar shows the URL: `portal.azure.com/?feature.msalljs=true#@bnz.onmicrosoft.com/resource/subscriptions/993817dc-61e1-4957-94b3-496e0c395840/resour...`. The page title is "pg-bnz-sy0-npdis1-vwuj-dev-1" and the subtitle is "Azure Database for PostgreSQL flexible server".

On the left, there is a navigation pane with the following items: Overview (selected), Activity log, Access control (IAM), Tags, and Diagnose and solve problems.

The main content area shows the "Essentials" section with the following details:

- Subscription (move): [BNZ-NPDIS1-CPS](#)
- Subscription ID: 993817dc-61e1-4957-94b3-496e0c395840
- Resource group (move): [RG-BNZ-SY0-NPDIS1-DB2POC](#)
- Status: Available
- Location: Australia East

On the right, there is a table of server details:

Server name	: pg-bnz-sy0-npdis1-vwuj-dev-1
Server admin login name	: postgres
Configuration	: General Purpose, D2ds v5, 2 vCores
PostgreSQL version	: 15.3
Availability zone	: 2

At the top of the Essentials section, there is a search bar and a row of action buttons: Delete, Reset password, Restore, Restart, Upgrade, Stop, Refresh, CLI / PS, and Feedback.

- Our example AWS *Target* database is using our test AWS PostgreSQL database, inf-testdb

The screenshot shows the AWS Management Console for an Amazon RDS instance. The left sidebar contains the 'Amazon RDS' menu with options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, and Zero-ETL integrations. The main content area is titled 'Connectivity & security' and is divided into three columns:

- Endpoint & port:**
 - Endpoint: inf-testdb.cmwfhjgk7eo1.ap-southeast-2.rds.amazonaws.com
 - Port: 5432
- Networking:**
 - Availability Zone: ap-southeast-2b
 - VPC: Platform VPC (vpc-08a861b38bf6c891b)
 - Subnet group: inf-testdb-db-subnetgroup
 - Subnets: subnet-0072fd3bbeacfb653, subnet-03ca72f5bf8320efe
 - Network type: IPv4
- Security:**
 - VPC security groups: inf-testdb-db-server-sg (sg-03cc848d81666e9d9) - Active
 - Publicly accessible: No
 - Certificate authority: rds-ca-2019
 - Certificate authority date: August 23, 2024, 05:08 (UTC+12:00)
 - DB instance certificate expiration date: August 23, 2024, 05:08 (UTC+12:00)

Next steps



Our examples are using DB2 LUW as our *Source* database, and PostgreSQL as the *Target* database. So we will use SCT to convert our database objects to the format used by PostgreSQL, such as lowercase Table names.



If we were using a database supported by DMS for object conversion, such as Microsoft SQL Server, or Oracle, we would be able to use DMS to migrate both our database **and** its data

 [Edit this page](#)