Step-by-Step Migration Process (Postgresql-xx to Postgresql-xx).

STEP 1: Verify Versions

```
psql -p 5432 -c "SELECT version();"
psql -p 5433 -c "SELECT version();"
```

Ensure they show PostgreSQL 12.x and 14.x respectively.

STEP 2: Configure Source (PostgreSQL 12)

2.1. Modify `postgresql.conf`

- Enable logical replication:

```
wal_level = logical
max_replication_slots = 10
max_wal_senders = 10
```

- Reload configuration:

```
SELECT pg_reload_conf();
```

2.2. Update `pg_hba.conf`

- Allow the target DB to connect for replication:

```
host    replication     repuser    <target_ip>/32    md5
host    all             repuser    <target_ip>/32    md5
```

- Reload again:

```
SELECT pg_reload_conf();
```

2.3. Create Replication Role

```
CREATE ROLE repuser WITH REPLICATION LOGIN PASSWORD 'StrongPassword';
```

STEP 3: Prepare the Target (PostgreSQL 14)

- Ensure wal_level is at least replica (default):

```
wal_level = replica
```

Also ensure connectivity from the source to the target works:

```
psql -h <target_ip> -U repuser -d postgres
```

STEP 4: Create Database and Schema on Target

- Create the same database name on the target

```
CREATE DATABASE mydb;
```

- Then dump the **schema only** (no data) from source and restore it into the target

```
pg_dump -h source_host -p 5432 -U postgres -s mydb > schema.sql
psql -h target_host -p 5433 -U postgres -d mydb -f schema.sql
```

STEP 5: Create a Publication on the Source

- Connect to your source database

```
psql -h source_host -p 5432 -U postgres -d mydb
```

- Create the publication

```
CREATE PUBLICATION mypub FOR ALL TABLES;
```

- You can also specify only certain tables

```
CREATE PUBLICATION mypub FOR TABLE public.table1, public.table2;
```

- Check

```
\dRp+
```

STEP 6: Create a Subscription on the Target

- Connect to your target PostgreSQL 14 database

```
psql -h target_host -p 5433 -U postgres -d mydb
```

- Create the subscription

```
CREATE SUBSCRIPTION mysub CONNECTION 'host=<source_host> port=5432
user=repuser password=StrongPassword dbname=mydb'

PUBLICATION mypub;
```

STEP 7: Verify Replication

- Check subscription status

```
SELECT * FROM pg_stat_subscription;
```

- You should see something like

```
relid | received_lsn | last_msg_send_time | last_msg_receipt_time | ...
-------------------------------------------------------------------------
...   | ...          | ...                | ...                   | ...
```

- Check replication lag:

```
SELECT
  application_name,
  state,
  sent_lsn,
  write_lsn,
  flush_lsn,
```

```
  replay_lsn
FROM pg_stat_replication;
```

STEP 8: Monitor Data Sync

- You can periodically check the count of rows on both databases:

```
SELECT count(*) FROM table_name;
```

- Once counts match and replication lag is zero, you are synced.

STEP 9: Application Switchover (Cutover)

When you're ready to migrate:

- Stop writes on source DB (put app in maintenance mode).
- Wait for replication to catch up.
- Confirm zero replication lag.
- Drop the subscription on target (This step makes target independent).

```
DROP SUBSCRIPTION mysub;
```

- Point application connections to PostgreSQL 14.

STEP 10: Post-Migration Cleanup

- Remove the publication from source and Remove replication role if not needed

```
DROP PUBLICATION mypub;
DROP ROLE repuser;
```

- Validate application functionality.
- Run vacuum and analyze:

```
VACUUM ANALYZE;
```

STEP 11: Check Schema Compatibility - Make sure your schema is compatible (some deprecated syntax may break in 14).

STEP 12: Sequence Synchronization:

- Logical replication doesn't replicate sequence values. Manually sync them.

```
SELECT setval('table_id_seq', (SELECT MAX(id) FROM table));
```

STEP 13: Foreign Keys & Triggers

- These may cause issues on target; disable during sync if needed and re-enable after full sync.

```
ALTER TABLE table_name DISABLE TRIGGER ALL;
ALTER TABLE table_name ENABLE TRIGGER ALL;
```