Pg_upgarde error face:-

Error1:-

```
 postgres@cdbtestdcserver1:/data/pgsql_16$ pg_controldata /data/pgsql_16 | grep "Data page checksum version"
Data page checksum version:          0
postgres@cdbtestdcserver1:/data/pgsql_16$ /usr/lib/pgsql-16/bin/pg_upgrade -b /usr/lib/pgsql-15.6/bin/ -B /usr/lib/pgsql-16/bin/ -D /data/pgsql_16/ -d /data/patroni/ --check
Performing Consistency Checks
-----------------------------
Checking cluster versions                                   ok

old cluster uses data checksums but the new one does not
Failure, exiting
```

Sol:-

1. To check datachecksum:-

```
pg_controldata /data/pgsql_16 | grep "Data page checksum version"
```

Expected output:-

Data page checksum version: 1 → Checksums are enabled

If Data page checksum version: 0 → Checksums are disabled.

Then reintailze data directory with checksum:-

```
/usr/lib/pgsql-16/bin/initdb   --data-checksums -D /data/pgsql_16
```

Note:-

What is Data Checksum in PostgreSQL?

Data checksum is a feature in PostgreSQL that helps detect **data corruption** by verifying the integrity of data pages. It ensures that data written to disk is not altered due to hardware failures, disk corruption, or other issues.

How Does Data Checksum Work?

- When **checksums are enabled**, PostgreSQL calculates a **checksum** (a small numerical hash) for each **8 KB page** before writing it to disk.
- When reading the page back, PostgreSQL recalculates the checksum and **compares it** with the stored value.
- If the checksum does not match, PostgreSQL **reports a corruption error**, helping administrators detect disk or memory corruption early.

⍰ Potential Negative Impacts	
Impact	**Details**
⍰ Slight Performance Overhead (~1-2%)	PostgreSQL performs an extra checksum validation on every page read from disk.
⍰ Increased CPU Usage	Extra CPU cycles are needed to compute and verify checksums.
⍰ Slightly Larger Disk Writes	Each data page stores a checksum, slightly increasing the size of stored data.
⍰ Cannot Enable or Disable Later	Must be set at initdb time; requires a full reinitialization to change.

⍰ When Data Checksums Are Worth It		
Use Case	**Should You Enable Checksums?**	**Why?**
Production Databases	✅Yes	Protects against silent data corruption.
Banking, Finance, Healthcare, etc.	✅Yes	High reliability and data integrity are critical.
High-Performance Systems (OLTP, Analytics)	⍰ Maybe	Test the impact on CPU and disk performance first.
Test/Development Environments	❌No	Not necessary unless testing for corruption detection.

==

Error2:-

When we upgrade postgresql if password set we faced below error:-

```
[postgres@MCVD41S01023 ~]$ /usr/lib/pgsql-16.7/bin/pg_upgrade -d  /data/postgres_14 -D
/data/pgsql_16  -b /usr/pgsql-14/bin -B /usr/lib/pgsql-16.7/bin  -c
Performing Consistency Checks
-----------------------------
Checking cluster versions                                ok

connection to server on socket "/var/lib/pgsql/.s.PGSQL.50432" failed: fe_sendauth: no password
supplied


could not connect to source postmaster started with the command:
"/usr/pgsql-14/bin/pg_ctl" -w -l
"/data/pgsql_16/pg_upgrade_output.d/20250313T232957.377/log/pg_upgrade_server.log" -D
"/data/postgres_14" -o "-p 50432 -b  -c listen_addresses='' -c unix_socket_permissions=0700 -c
unix_socket_directories='/var/lib/pgsql'" start
Failure, exiting
```

In that case used below sloution:-

Solution 1: Use PGPASSWORD Environment Variable

Run pg_upgrade with the PGPASSWORD environment variable:-

```
export PGPASSWORD='your_postgres_password'
/usr/lib/pgsql-16.7/bin/pg_upgrade \
  -d /data/postgres_14 \
  -D /data/pgsql_16 \
  -b /usr/pgsql-14/bin \
  -B /usr/lib/pgsql-16.7/bin \
  -c
```

Solution 2: Use ~/.pgpass **File** (Recommended for Security)

Create a .pgpass file in the **home directory**:

```
echo "localhost:50432:*:postgres:your_postgres_password" > ~/.pgpass
chmod 600 ~/.pgpass
```

Then run pg_upgrade as usual:

```
/usr/lib/pgsql-16.7/bin/pg_upgrade \
  -d /data/postgres_14 \
  -D /data/pgsql_16 \
  -b /usr/pgsql-14/bin \
  -B /usr/lib/pgsql-16.7/bin \
  -c
```

Error3:-

```
postgres@cdbtestdcserver1:~$ /usr/lib/pgsql-16/bin/pg_upgrade -b /usr/lib/pgsql-15.6/bin/ -B /usr/lib/pgsql-16/bin/ -D /data/pgsql_16/ -d /data/patroni/ --username=postgres --check
Performing Consistency Checks
-----------------------------
Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for system-defined composite types in user tables  ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Checking for incompatible "aclitem" data type in user tables ok
Checking for presence of required libraries                 fatal

Your installation references loadable libraries that are missing from the
new installation.  You can add these libraries to the new installation,
or remove the functions using them from the old installation.  A list of
problem libraries is in the file:

/data/pgsql_16/pg_upgrade_output.d/20250318T185054.477/loadable_libraries.txt
Failure, exiting
postgres@cdbtestdcserver1:~$ cat
/data/pgsql_16/pg_upgrade_output.d/20250318T185054.477/loadable_libraries.txt
could not load library "$libdir/pg_cron": ERROR:  could not access file
"$libdir/pg_cron": No such file or directory
```

In database: test

Sol:-

Issue: Missing Loadable Libraries During pg_upgrade

Your PostgreSQL upgrade failed because the **pg_cron** extension is missing in PostgreSQL 16.

Install pg_cron extension

✓**If you need** pg_cron, install it in PostgreSQL 16 before upgrading.
✗**If you don't need it**, drop it from the database before running pg_upgrade

---

Error4:-

```
postgres@bestprddb1-new:/data$ /usr/lib/pgsql-16.4/bin/pg_upgrade -d /data/postgresql-14/data -
D /data/postgresql-16/data -b /usr/lib/pgsql-14.7/bin -B /usr/lib/pgsql-16.4/bin -c
Performing Consistency Checks
-----------------------------
Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for system-defined composite types in user tables  ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Checking for incompatible "aclitem" data type in user tables ok

New cluster database "postgres" is not empty: found relation "cron.job"
Failure, exiting
```

Sol:-

Step-by-Step Fix:-

1. pg_ctl -D /data/postgresql-16/data stop

2. Remove Existing PostgreSQL 16 Data Directory

rm -rf /data/postgresql-16/data

3. Re-initialize With Checksums:-

/usr/lib/pgsql-16.4/bin/initdb -D /data/postgresql-16/data --data-checksums

4. Run pg_upgrade **Again:-**

```
/usr/lib/pgsql-16.4/bin/pg_upgrade \
  -d /data/postgresql-14/data \
  -D /data/postgresql-16/data \
  -b /usr/lib/pgsql-14.7/bin \
  -B /usr/lib/pgsql-16.4/bin \
  -c
```