PostgreSQL Tablespace Backup & Restore using pg_dump

Understanding

- pg_dump is a logical backup tool — it does NOT copy the physical tablespace files.
- It records the tablespace name for each object in the dump.
- When restoring, you must create the same tablespaces (or remap them) before loading the dump.

1. Identify Objects in a Specific Tablespace

Run the following query to find which objects are in a specific tablespace:

```
SELECT relname, schemaname, tablespace
FROM pg_tables
WHERE tablespace = 'my_tablespace';
```

2. Take Backup of Objects in a Tablespace

Example: Dump a single table in a specific tablespace:

```
pg_dump -h <host> -p <port> -U <user> -F c \
  -t my_schema.my_table \
  -f /backup/my_tablespace_data.backup \
  mydb
```

Full database dump (preserves tablespace assignments):

```
pg_dump -h <host> -p <port> -U <user> -F c \
  -f /backup/mydb_with_tablespaces.backup \
  mydb
```

3. Restoring with Tablespaces

1. Create the required tablespaces before restoring:

```
CREATE TABLESPACE my_tablespace LOCATION
'/new/location/my_tablespace';
```

2. Restore the backup:

```
pg_restore -h <host> -p <port> -U <user> \
  -d mydb \
  /backup/mydb_with_tablespaces.backup
```

4. Restoring to a Different Location

If you want to restore tablespaces to a new location:
1. Dump the database in plain text format:

```
pg_dump -h <host> -p <port> -U <user> -F p mydb > mydb.sql
```

2. Edit the `CREATE TABLESPACE` lines in `mydb.sql` to point to the new location.

3. Restore using psql:

```
psql -h <host> -p <port> -U <user> -d mydb -f mydb.sql
```

Key Points

- pg_dump never copies physical tablespace directories — only logical assignments.
- You must pre-create required tablespaces before restore.
- If location changes, either edit the dump file or create the tablespace in the new location before restore.