

PostgreSQL Artifact Extraction Prerequisites

Contents

- 1. Introduction3
- 2. Artifact Extraction.....3
 - 2.1 DDL Scripts3
 - 2.2 Query Execution Logs3
 - 2.2.1 Config File Location3
 - 2.2.2 Data Directory Path.....3
 - 2.2.3 Logging4
 - 2.2.4 Restarting Services.....4
 - 2.3 Other Database Objects4
 - 2.4 Other Scripts and Artifacts5
- 3. Getting Help.....6

1. Introduction

LeapLogic's Assessment profiles existing inventory, identifies complexity, performs dependency analysis, and provides recommendations for migration to modern data platform.

2. Artifact Extraction

LeapLogic requires certain artifacts to perform an assessment. As a prerequisite, to fetch the required data you must have admin privileges. You can retrieve these artifacts from your GIT instance or the PostgreSQL repository where all the artifacts such as DDL scripts, stored procedures, functions, query execution logs, DML scripts, and other database objects are stored. LeapLogic needs all these artifacts in the form of .sql files.

2.1 DDL Scripts

You can use the `pg_dump` command in PostgreSQL to export the DDLs from the database.

Note: It is recommended to execute the `pg_dump` command for all databases, particularly the ones you intend to migrate.

Use the following command to export the DDL scripts to the machine where PostgreSQL is installed. Modify the command details highlighted in **yellow** based on your requirements. This command will export all the DDLs of the database objects present in the PostgreSQL database.

```
pg_dump -h host -U username -d database -p 5432 --schema-only -f /path/pg_dump_database_name.sql
```

2.2 Query Execution Logs

To export the query execution logs, make sure that the error reporting and logging are enabled in PostgreSQL. The following are steps to enable logging.

2.2.1 Config File Location

Run the command below in PostgreSQL to find the location of the configuration file.

```
postgres=# SHOW config_file;
```

2.2.2 Data Directory Path

The data directory is the storage location for the query execution log. Run the below command in PostgreSQL to find the data directory path.

```
postgres=# SHOW data_directory;
```

2.2.3 Logging

The following are properties that are used to enable logging in PostgreSQL. Modify the 'postgresql.conf' file to activate these properties.

```
log_destination = 'csvlog'

logging_collector = on

log_directory = 'pg_log'

log_filename = 'postgresql-query-execution-%Y-%m-%d_%H%M%S.log'
```

2.2.4 Restarting Services

After the properties are modified, you should restart the PostgreSQL service to initiate logging.

```
$ service postgresql restart
```

Copy the log file from the data directory location and share it with the LeapLogic team.

2.3 Other Database Objects

To enhance the assessment results for your environment and workloads, it is recommended to export additional database objects. Please refer to the below script to export the data as separate delimited files.

```
--- 2.3.1 Database Objects
SELECT table_name,
table_catalog AS databasename,
table_schema AS SchemaName,
table_type
FROM information_schema.tables
WHERE table_type = 'BASE TABLE'
AND table_schema NOT IN
('pg_catalog', 'information_schema')
UNION
SELECT table_name,
table_catalog AS databasename,
table_schema AS SchemaName,
'VIEW' AS table_type
FROM information_schema.views
WHERE table_schema NOT IN ('pg_catalog', 'information_schema')
AND table_name !~ '^pg_';

--- 2.3.2 Database Objects Count.
SELECT databasename,
table_type,
count(table_name) AS table_count
FROM (SELECT table_name,
table_catalog AS databasename,
table_schema AS SchemaName,
table_type
FROM information_schema.tables
```

```

WHERE table_type = 'BASE TABLE'
AND table_schema NOT IN
    ('pg_catalog', 'information_schema')
UNION
SELECT table_name,
table_catalog AS databasename,
table_schema AS SchemaName,
'VIEW' AS table_type
    FROM information_schema.views
    WHERE table_schema NOT IN ('pg_catalog', 'information_schema')
        AND table_name !~ '^pg_'
GROUP BY 1, 2;

--- 2.3.3 Database Volume
select t1.datname AS db_name,
        pg_size_pretty(pg_database_size(t1.datname)) as db_size
from pg_database t1
order by pg_database_size(t1.datname) desc;

--- 2.3.4 Table volumn
SELECT
table_catalog AS databaseName,
table_name,
pg_size_pretty(pg_total_relation_size(quote_ident(table_name))) AS total_size
from information_schema.tables
where table_schema = 'public'
order by 3 desc;

--- 2.3.5 User List
SELECT username
    FROM pg_user;

```

2.4 Other Scripts and Artifacts

Copy any other scripts such as DML scripts etc. from your GIT instance and share them with the LeapLogic team to produce more extensive insights.

3. Getting Help

Contact LeapLogic technical support at info@leaplogic.io