YUGABYTE VOYAGER PREREQUISITES

Prerequisites
The following sections describe the prerequisites for installing YugabyteDB Voyager.

Operating system
You can install YugabyteDB Voyager on the following:

- RHEL 8, 9
- CentOS 8
- Ubuntu 18.04, 20.04, 22.04
- macOS (for MySQL/Oracle source databases on macOS, install yb-voyager using the Docker option.)

Hardware requirements

- Disk space of at least 1.5 times the estimated size of the source database
- 2 cores minimum (recommended)

Software requirement

- Java 17. Any higher versions of Java might lead to errors during installation or migration.

Prepare the host

- The node where you'll run the yb-voyager command should:
- connect to both the source and the target database.
- have sudo access.

Important links

- Install yb-voyager | YugabyteDB Docs
- YugabyteDB Voyager introduction | YugabyteDB Docs

***Steps To Migration from The Postgres to Yugabyte Using the Voyager**

1. Prepare the Source Database

```
Create a database user with READ access for migration.
User to be created: **ybvoyager** - Command:
```

```
CREATE USER ybvoyager PASSWORD ‘password’
```

Grant permissions for migration using a specific script

```
Script location: `/opt/yb-voyager/guardrails-scripts/` Command to execute
the script:

psql -h <host> \
-d <database> \
-U <username> \ # A superuser or a privileged user with enough permissions
to grant privileges
-v voyager_user='ybvoyager' \
-v schema_list='<comma_separated_schema_list>' \
-v is_live_migration=0 \
-v is_live_migration_fall_back=0 \
-f <path_to_the_script>
```

2.Prepare the target database

*Create the target YugabyteDB database in your YugabyteDB cluster. The database name can be the same or different from the source database name.
If the target YugabyteDB database name is not provided, yb-voyager assumes the target YugabyteDB database name to be yugabyte.

```
      CREATE DATABASE target_db_name;
```

*For a local YugabyteDB cluster or YugabyteDB Anywhere, create a user and role with the superuser privileges using the following command:

```
    CREATE USER ybvoyager SUPERUSER PASSWORD 'password';
```

3.**Create an export directory**

- yb-voyager keeps all of its migration state, including exported schema and data, in a local directory called the export directory.
- Before starting migration, you should create the export directory on a file system that has enough space to keep the entire source database. Next,
- We should provide the path of the export directory as a mandatory argument (--export-dir) to each invocation of the yb-voyager command in an environment variable.

```
mkdir $HOME/export-dir
export EXPORT_DIR=$HOME/export-dir
```

4.Assess migration

- This step is optional and only applicable to PostgreSQL and Oracle database migrations.
- Assess migration analyzes the source database, captures essential metadata, and generates a report with recommended migration strategies and cluster configurations.

```
yb-voyager assess-migration
--source-db-type postgresql \
--source-db-host hostname
--source-db-user ybvoyager \
--source-db-password password
--source-db-name dbname \
--source-db-schema schema1,schema2
--export-dir /path/to/export/dir
```

5.Export and analyze schema

To begin, export the schema from the source database. Once exported, analyze the schema and apply any necessary manual changes.

Export schema
The yb-voyager export schema command extracts the schema from the source database, converts it into PostgreSQL format

```
yb-voyager export schema --export-dir <EXPORT_DIR> \
        --source-db-type <SOURCE_DB_TYPE> \
        --source-db-host <SOURCE_DB_HOST> \
        --source-db-user <SOURCE_DB_USER> \
        --source-db-password <SOURCE_DB_PASSWORD> \ # Enclose the password
in single quotes if it contains special characters.
        --source-db-name <SOURCE_DB_NAME> \
        --source-db-schema <SOURCE_DB_SCHEMA> # Not applicable for MySQL
```

6.Analyze schema

The yb-voyager analyze-schema command analyses the PostgreSQL schema dumped in the export schema step, and prepares a report that lists the DDL statements which need manual changes.

```
  yb-voyager analyze-schema --export-dir <EXPORT_DIR> --output-format
<FORMAT>
```

7.Import schema

```
yb-voyager import schema --export-dir <EXPORT_DIR> \
        --target-db-host <TARGET_DB_HOST> \
        --target-db-user <TARGET_DB_USER> \
        --target-db-password <TARGET_DB_PASSWORD> \ # Enclose the password
in single quotes if it contains special characters.
        --target-db-name <TARGET_DB_NAME> \
        --target-db-schema <TARGET_DB_SCHEMA> # MySQL and Oracle only
```

8.Export data

Dump the source data into the EXPORT_DIR/data directory using the yb-voyager export data command as follows:

```
yb-voyager export data --export-dir <EXPORT_DIR> \
        --source-db-type <SOURCE_DB_TYPE> \
        --source-db-host <SOURCE_DB_HOST> \
        --source-db-user <SOURCE_DB_USER> \
        --source-db-password <SOURCE_DB_PASSWORD> \ # Enclose the password
in single quotes if it contains special characters.
        --source-db-name <SOURCE_DB_NAME> \
        --source-db-schema <SOURCE_DB_SCHEMA> # Not applicable for MySQL
```

9.Import data

After you have successfully exported the source data and imported the schema in the target YugabyteDB database,
you can import the data using the yb-voyager import data command with required arguments as follows:

```
yb-voyager import data --export-dir <EXPORT_DIR> \
        --target-db-host <TARGET_DB_HOST> \
        --target-db-user <TARGET_DB_USER> \
        --target-db-password <TARGET_DB_PASSWORD> \ # Enclose the password
in single quotes if it contains special characters.
        --target-db-name <TARGET_DB_NAME> \
        --target-db-schema <TARGET_DB_SCHEMA> \ # MySQL and Oracle only.
        --parallel-jobs <NUMBER_OF_JOBS>
```

10.Finalize schema post data import

```
yb-voyager finalize-schema-post-data-import --export-dir <EXPORT_DIR> \
       --target-db-host <TARGET_DB_HOST> \
       --target-db-user <TARGET_DB_USER> \
       --target-db-password <TARGET_DB_PASSWORD> \ # Enclose the password in
single quotes if it contains special characters.
       --target-db-name <TARGET_DB_NAME> \
       --target-db-schema <TARGET_DB_SCHEMA> \ # MySQL and Oracle only
```

11.End migration

```
yb-voyager end migration --export-dir <EXPORT_DIR> \
        --backup-log-files <true, false, yes, no, 1, 0> \
        --backup-data-files <true, false, yes, no, 1, 0> \
        --backup-schema-files <true, false, yes, no, 1, 0> \
        --save-migration-reports <true, false, yes, no, 1, 0> \
        # Set optional argument to store a back up of any of the above
arguments.
        --backup-dir <BACKUP_DIR>
```

12.Delete the ybvoyager user (Optional)

After migration, all the migrated objects (tables, views, and so on) are owned by the ybvoyager user.
Transfer the ownership of the objects to some other user (for example, yugabyte) and then delete the ybvoyager user.

```
  REASSIGN OWNED BY ybvoyager TO yugabyte;
  DROP OWNED BY ybvoyager;
  DROP USER ybvoyager;
```