Setting Up MySQL and ClickHouse Servers and Clients

1. Installing MySQL Server on Ubuntu

MySQL is a popular relational database management system used for storing and managing structured data.

```
bash
sudo apt update
sudo apt install mysql-server
```

- `sudo apt update`: This command updates the package lists for upgrades and new package installation.
- `sudo apt install mysql-server`: This command installs the MySQL server package on your Ubuntu system.

2. Installing ClickHouse Server and Client on Ubuntu

ClickHouse is an open-source column-oriented database management system used for online analytical processing (OLAP).

Install ClickHouse Server

```
bash
curl https://clickhouse.com/ | sh
```

- `curl https://clickhouse.com/ | sh`: This command installs the ClickHouse server and client packages on your Ubuntu system.

3. Changing Root User Password for MySQL Service

By default, MySQL comes with a root user without a password. It is recommended to set a password for the root user for security purposes.

```
bash
sudo mysql -u root
```

```
sql
ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
FLUSH PRIVILEGES;
```

- `sudo mysql -u root`: This command allows you to log in to the MySQL server as the root user.
- `ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';`: This SQL command changes the password for the root user.

- `FLUSH PRIVILEGES;`: This command reloads the privileges from the grant tables in the MySQL database.

4. Opening Terminals

Open three terminal windows for MySQL, ClickHouse Client, and general commands.

5. Creating and Filling Test Database in MySQL

We'll create a sample database and table in MySQL and populate it with some data.

```
bash
mysql -u root -p
```

```
CREATE DATABASE test_db;
USE test_db;

CREATE TABLE example_table (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50)
);

INSERT INTO example_table (name) VALUES ('John'), ('Jane'), ('Doe');
```

- `mysql -u root -p`: This command allows you to log in to the MySQL server using the root user and prompts for the password.
- `CREATE DATABASE test_db;`: This SQL command creates a new database named `test_db`.
- `USE test_db;`: This SQL command selects the `test_db` database for use.
- `CREATE TABLE example_table (...)`: This SQL command creates a table named `example_table` with an `id` column as the primary key and a `name` column.
- `INSERT INTO example_table (...) VALUES (...)`: This SQL command inserts sample data into the `example_table`.

6. Exporting MySQL Database

Navigate to the desired directory where you want to store the SQL dump file.

```
bash
cd user_files
```

Export the MySQL database to a SQL file.

```
bash
mysqldump -u root -p test_db > test_db.sql
```

- `cd user_files`: This command changes the current directory to `user_files` where we want to store the SQL dump file.
- `mysqldump -u root -p test_db > test_db.sql`: This command dumps the `test_db` database to a SQL file named `test_db.sql`. You will be prompted to enter the MySQL root password.

7. Testing Database for Inconsistencies in ClickHouse Client

In the ClickHouse Client, we'll test the imported database for inconsistencies.

```
bash
clickhouse-client

SELECT * FROM File('test_db.sql', MySQLDump) LIMIT 5;
```

- `clickhouse-client`: This command starts the ClickHouse Client.
- `SELECT * FROM File('test_db.sql', MySQLDump) LIMIT 5;`: This SQL command selects and displays the first 5 rows from the SQL file `test_db.sql` using ClickHouse's `MySQLDump` function.

8. Integrating the Database into a ClickHouse Table

```
sql
CREATE TABLE "Table_name" Engine = MergeTree ORDER BY tuple() AS SELECT * FROM
file('Filename.sql', MySQLDump);
```

- `CREATE TABLE "Table_name" Engine = MergeTree ORDER BY tuple() AS SELECT * FROM file('Filename.sql', MySQLDump);`: This SQL command creates a ClickHouse table named `"Table_name"` using the `MergeTree` storage engine and orders it by the tuple. It then populates the table with data from the SQL file `Filename.sql` using the `MySQLDump` function. Replace `"Table_name"` with your desired table name and `"Filename.sql"` with the actual name of your SQL dump file.