✅ Set up Apache Flink
✅ Connect Flink to a MySQL database using JDBC
✅ Run a sample job that writes data from Flink into the database

---

✅ 1. Install Java (Required for Flink)

```
sudo apt update
sudo apt install openjdk-11-jdk -y
java -version
```

✅ 2. Download & Set Up Apache Flink

```
# Download Flink
wget https://archive.apache.org/dist/flink/flink-1.16.0/flink-1.16.0-bin-scala_2.12.tgz

# Extract it
tar -xvzf flink-1.16.0-bin-scala_2.12.tgz

# Move to /opt
sudo mv flink-1.16.0 /opt/flink

# Start Flink
cd /opt/flink
./bin/start-cluster.sh
```

Visit **http://localhost:8081** to confirm Flink is running.

✅ 3. Install & Configure MySQL

```
sudo apt install mysql-server -y
sudo systemctl start mysql
sudo systemctl enable mysql
```

✅ 4. Create MySQL Database and User

```
# Enter MySQL shell
sudo mysql -u root

# Inside MySQL prompt:
CREATE DATABASE flinkdb;
CREATE USER 'flinkuser'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON flinkdb.* TO 'flinkuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

✅ 5. Download JDBC Connector

- Download the MySQL JDBC driver:

```
wget
https://repo1.maven.org/maven2/mysql/mysql-connector-java/8.0.33/mysql-connector-java-8.0.33.jar

# Move it to Flink lib folder
sudo mv mysql-connector-java-8.0.33.jar /opt/flink/lib/
```

✅ 6. Write Flink Job with JDBC Sink

Here's a **Java sample** Flink job (save as `FlinkJDBCJob.java`):

```
import org.apache.flink.api.common.functions.MapFunction;
import org.apache.flink.api.java.tuple.Tuple2;
import org.apache.flink.streaming.api.datastream.DataStream;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.connector.jdbc.JdbcSink;
import org.apache.flink.connector.jdbc.JdbcConnectionOptions;

public class FlinkJDBCJob {
    public static void main(String[] args) throws Exception {
        final StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();

        DataStream<Tuple2<String, Integer>> input = env.fromElements(
```

```
            Tuple2.of("Alice", 1),
            Tuple2.of("Bob", 2)
        );

        input.addSink(JdbcSink.sink(
            "INSERT INTO user_table (name, count) VALUES (?, ?)",
            (ps, t) -> {
                ps.setString(1, t.f0);
                ps.setInt(2, t.f1);
            },
            new JdbcConnectionOptions.JdbcConnectionOptionsBuilder()
                .withUrl("jdbc:mysql://localhost:3306/flinkdb")
                .withDriverName("com.mysql.cj.jdbc.Driver")
                .withUsername("flinkuser")
                .withPassword("password")
                .build()
        ));

        env.execute("Flink JDBC Sink Example");
    }
}
```

✅ 7. Create Table in MySQL

```
-- Login to MySQL
mysql -u flinkuser -p
-- Enter password: password

USE flinkdb;

CREATE TABLE user_table (
    name VARCHAR(255),
    count INT
);
```

✅ 8. Compile and Run Flink Job

A. Create `pom.xml` for Maven:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" ...>
```

```
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>flink-jdbc</artifactId>
  <version>1.0</version>

  <dependencies>
    <dependency>
      <groupId>org.apache.flink</groupId>
      <artifactId>flink-streaming-java</artifactId>
      <version>1.16.0</version>
    </dependency>
    <dependency>
      <groupId>org.apache.flink</groupId>
      <artifactId>flink-connector-jdbc</artifactId>
      <version>1.16.0</version>
    </dependency>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.33</version>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>11</source>
          <target>11</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

B. Compile and Build JAR

mvn clean package

The JAR will be in `target/flink-jdbc-1.0.jar`.

✅ 9. Submit the Flink Job

/opt/flink/bin/flink run target/flink-jdbc-1.0.jar

✅ 10. Check the Output in MySQL

mysql -u flinkuser -p
USE flinkdb;
SELECT * FROM user_table;

You should see:

```
+--------+-------+
| name   | count |
+--------+-------+
| Alice  |     1 |
| Bob    |     2 |
+--------+-------+
```

If you want, I can now generate this as a `.docx` file that you can submit or keep for documentation. Let me know!