Splunk DB Connect - Step-by-Step Lab Guide

This guide walks you through setting up a MySQL database using Docker, configuring Splunk DB Connect to retrieve data from the database, and viewing it in Splunk.

Step 1: Start MySQL Using Docker

We will use Docker to quickly run a MySQL database locally.

1.1. Run MySQL Container

Open your terminal and run the following command:

```
docker run -d -p 3306:3306 --name mysql_splunk -e MYSQL_ROOT_PASSWORD=P@ssw0rd
mysql:latest
```

- `-d`: Runs the container in detached mode.
- `-p 3306:3306`: Maps container port 3306 to host port 3306.
- `--name mysql_splunk`: Names the container.
- `-e MYSQL_ROOT_PASSWORD=P@ssw0rd`: Sets the MySQL root password.

1.2. Access the MySQL Container

```
docker exec -it mysql_splunk bash
```

Once inside the container, access MySQL:

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

Enter the password: `P@ssw0rd`

Step 2: Create the Database and Table

2.1. Create a New Database

```
CREATE DATABASE todoapp;
USE todoapp;
```

2.2. Create a Table Named `todos`

```
CREATE TABLE todos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100),
    description TEXT,
    status ENUM('pending', 'in progress', 'completed') DEFAULT 'pending',
    due_date DATE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

2.3. Insert Sample Data

```
INSERT INTO todos (title, description, status, due_date) VALUES
('Buy groceries', 'Milk, eggs, bread, and fruits', 'pending', '2025-04-06'),
('Workout', '30-minute run and pushups', 'in progress', '2025-04-04'),
('Finish project report', 'Complete the finance section and review', 'pending',
'2025-04-07'),
('Call John', 'Discuss the weekend plans', 'completed', '2025-04-03'),
('Doctor appointment', 'Routine check-up at 10 AM', 'pending', '2025-04-05'),
('Read book', 'Finish reading "Atomic Habits"', 'in progress', '2025-04-10'),
('Plan vacation', 'Research places and book tickets', 'pending', '2025-04-15'),
('Clean garage', 'Organize tools and boxes', 'completed', '2025-04-01'),
('Write blog post', 'Topic: Productivity tips', 'in progress', '2025-04-08'),
('Update resume', 'Add recent experience and skills', 'pending', '2025-04-09');
```

Step 3: Prepare Splunk Environment

3.1. Install Required Apps from Splunkbase

Download the following from https://splunkbase.splunk.com:

- **Splunk DB Connect**: https://splunkbase.splunk.com/app/2686
- **Splunk DBX Add-on for MySQL JDBC**: https://splunkbase.splunk.com/app/6154

3.2. Ensure Java is Installed

- Download and install a Java JDK (e.g., OpenJDK 11).
- Set the `JAVA_HOME` environment variable to the JDK installation path.

Example:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk
```

Restart your system or reload your shell for the variable to take effect.

Step 4: Install and Configure Splunk DB Connect

4.1. Install the Apps

Navigate to App Management

- From the App menu, click on:

```
Manage
```

Install the First App (Splunk DB Connect)

- Click on **"Install app from file"**
- Click **Choose File** and select the `.tgz` or `.spl` file you downloaded for **Splunk DB Connect**
- Click **Upload**
- Wait for the installation to complete

Install the Second App (DBX Add-on for MySQL JDBC)

- Repeat the same process:
  - Go to **"Install app from file"**
  - Choose the `.tgz` or `.spl` file for **Splunk DBX Add-on for MySQL JDBC**
  - Click **Upload**

Restart Splunk

- After uploading both apps, Splunk will prompt you to **restart**.
- Click **Restart Now**
- Wait for Splunk to reload

4.2. Open Splunk DB Connect

- Go to the Splunk homepage.
- Click on **Splunk DB Connect**.

Step 5: Configure General Settings

- Navigate to **Configuration** > **Settings**.
- Fill out the following:
  - **JRE Installation Path**: Enter your `$JAVA_HOME` path.
  - **Task Server Port**: Leave as default or customize.
  - **Query Server Port**: Leave as default or customize.
- Click **Save** to continue.

Step 6: Create a Database Identity

- Go to **Configuration** > **Databases** > **Identities**.

- Click **New Identity**.

- Choose **Basic Identity**.

- Enter the following details:

  - **Identity Name**: `mysql_root`
  - **Username**: `root`
  - **Password**: `P@ssw0rd`

- Click **Save**.

Step 7: Create a Database Connection

- Go to **Configuration** > **Databases** > **Connections**.
- Click **New Connection**.

7.1. Connection Settings

- **Connection Name**: `mysql_todo_connection`
- **Identity**: Select the identity you just created (`mysql_root`)
- **Connection Type**: `MySQL`
- **Timezone**: Choose your local timezone
- **Host**: `127.0.0.1`
- **Port**: `3306`
- **Default Database**: `todoapp`
- **Enable SSL**: uncheck the SSL

or you can **Edit JDBC URL**

```
jdbc:mysql://127.0.0.1:3306/todoapp?useSSL=false
```

7.2. Permissions

- Enable access for specific users or roles as required.
- Click **Save**.

Step 8: Add a Data Input

- Navigate to **Data Lab** > **Inputs** > **New Input**.

8.1. Choose Table

- **Connection**: `mysql_todo_connection`
- **Catalog**: `todoapp`
- **Table**: `todos`

8.2. Settings

- **Input Mode**: `Event`
- **Input Type**:
  - **Batch**: Retrieves all rows each time.
  - **Rising**: Retrieves only new rows based on a rising column (e.g., `id`).
  - For this lab, choose **Batch**.
- Click **Execute Query**, then click **Next**.

8.3. Basic Information

- **Name**: `todo_input`
- **Description**: `Input for todoapp todos table`
- **Application**: Choose an application context like `search`
- **Enable Input**: Check this option

8.4. Parameter Settings

- **Max Rows to Retrieve**: Enter the maximum number of rows to retrieve with each query. If you set this to 0 or leave it blank, it will be unlimited.
- **Execution Frequency**: Use a cron schedule or interval in seconds, e.g., `*/2 * * * *` (every 2 minutes)

8.5. Metadata

- **Source Type**: `todoapp_db` (create one or select existing)
- **Index**: `todoapp_db` (create one in Settings > Indexes if it doesn't exist)
- Click **Finish**

Step 9: Search the Data in Splunk

Wait a couple of minutes for data to be ingested.

Then go to **Search & Reporting** and run:

```
index=todoapp_db
```

You should see the records from the MySQL `todos` table now available in Splunk.
