



Software Documentation

BetterData Synthetic Data engine

Version 1.0.0

BetterData PTE Limited
June 6, 2022

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 2 |
| 1.1 | About BetterData | 2 |
| 1.2 | About Betterdata Synthetic Data Engine | 2 |
| 1.3 | About Metrics | 2 |
| 2 | Software Installations | 2 |
| 2.1 | Setup Guide | 2 |
| 3 | Modules | 5 |
| 3.1 | Overview | 5 |
| 3.2 | Authenticators | 6 |
| 3.2.1 | Module: Create New Users | 6 |
| 3.2.2 | Module: Login to Software | 7 |
| 3.2.3 | Module: Delete an User | 8 |

List of Code Listings

1 Introduction

1.1 About BetterData

BetterData is a data privacy company that deals in transforming production data into privacy-preserving and highly realistic synthetic data. Share, access and build with data safely across teams, businesses, and international borders.

1.2 About Betterdata Synthetic Data Engine

Real data is flawed as it is limited in its scope, biased in nature and restricted to use because of increasing data protection regulations worldwide. Our technology solves this by providing a unified and one-stop data platform that combines state-of-the-art AI models with cutting-edge privacy-preserving techniques to convert limited real data into limitless synthetic data that has 99+ % utility of the real data but 0%privacy risks. The synthetic data can be used to simulate 'what-if' situations for analysis, remove biases from datasets and enable open access to data with 100% compliance.

1.3 About Metrics

2 Software Installations

2.1 Setup Guide

Pre-installation Requirements:

- **Operating System:** Linux as e.g. a specific Linux Distribution (ubuntu 20.04).
- Active Internet Connectivity.
- VCPU: minimum 8 core.
- RAM: minimum 16gb.
- Storage: minimum 256gb.

Installation instructions and steps: Unzip "Client-pack" and open it. Open a terminal in "Client-pack" folder:

1. Lets install MLPack.
type:

```
1 cd ./ML_Pack/  
2 sudo bash setup.sh
```

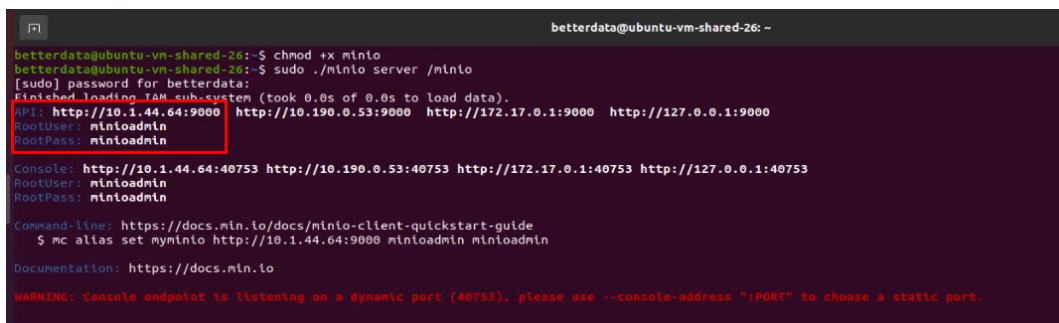
Don't close this terminal, open another terminal window by clicking "+" sign on top left in current terminal.

2. Lets install Minio.
type:

```
1 sudo bash install_minio.sh
```

Now note down the first ip and port, access key, secret key, this will be used later.
for example:

Note: This is only an example. Your minio_ip will be different on your terminal but you may see something similar like this.



```
betterdata@ubuntu-vm-shared-26: ~  
betterdata@ubuntu-vm-shared-26:~$ chmod +x minio  
betterdata@ubuntu-vm-shared-26:~$ sudo ./minio server /minio  
[sudo] password for betterdata:  
Finished loading IAM subsystem (took 0.0s of 0.0s to load data).  
API: http://10.1.44.64:9000 http://10.190.0.53:9000 http://172.17.0.1:9000 http://127.0.0.1:9000  
RootUser: minioadmin  
RootPass: minioadmin  
Console: http://10.1.44.64:40753 http://10.190.0.53:40753 http://172.17.0.1:40753 http://127.0.0.1:40753  
RootUser: minioadmin  
RootPass: minioadmin  
Command-line: https://docs.min.io/docs/minio-client-quickstart-guide  
$ mc alias set myminio http://10.1.44.64:9000 minioadmin minioadmin  
Documentation: https://docs.min.io  
WARNING: Console endpoint is listening on a dynamic port (40753). please use --console-address ":PORT" to choose a static port.
```

Figure 1: Snapshot of minio server on running on terminal. Here, minio_ip: 10.1.44.64. This will be required on "info.sh" file later where says minio_ip, provide this ip. RootUser is the default MINIO_ACCESS_KEY and RootPass is the default MINIO_SECRET_KEY.

Don't close this terminal, just minimize.
Now Lets deploy software backend.

3. Go to **Client-pack/webapp-backend/src** folder and open and modify the **install.sh** in a text-editor if you want to setup your own details for

mysql. Otherwise system will use our default :

DATABASE: betterdata, USER: USER, PASSWORD:PASSWORD.

```
1 CREATE DATABASE '<database>'
2 CREATE USER '<user>'@'%' IDENTIFIED WITH authentication_plugin BY '<password>'
3 CREATE USER '<user>'@'%' IDENTIFIED BY '<password>'
```

4. Now open **info.sh** in a text-editor and fill the <fields> in info.sh with the configurations you made in **install.sh** file mentioned:

```
1 export DATABASE_URL='mysql://<username>:<password>@<ip>:3306/<database name>';
2 export MINIO_URL='<minio_ip>:9000';
3 export MINIO_ACCESS_KEY='<minio RootUser>'
4 export MINIO_SECRET_KEY='<minio RootPass>';
5 export BUCKET='client-demo';
6 export MINIO_CREDS='<minio_ip>:9000,<minio RootUser>,<minio RootPass>';
7 export ML_ENDPOINT='http://<minio_ip>:8000/'
8 export MYSQL_CREDS='mysql+pymysql://<username>:<password>@
9                               <minio_ip>:3306/<database>';
```

5. Save it using **CTRL + S** or save button from UI. Now Open a new terminal in "Client-pack" folder:
type:

```
1 cd ./webapp-backend/src/
2 cp -r info.sh /root
3 sudo -i
4 exit
5 #again open a new terminal in "client-pack" folder
6 cd ./webapp-backend/src/
7 sudo bash setup.sh
```

Now if everything is setup successfully, open a browser tab and type:

localhost:3000/docs

You should see betterdata synthetic engine UI screen. Perform your operations here. See the software documentation for details.

3 Modules

3.1 Overview

If user goes to a web browser and types in the address bar: “localhost:3000/docs”, the below UI will appear from where the user can operate betterdata synthetic data engine software.

| | | |
|------|---------------------------------------|-----------------------------|
| POST | /user | Create User |
| POST | /login | Login |
| POST | /password | Change Password |
| POST | /delete_user | Delete User |
| POST | /list_current_user_details | List Current User Details |
| POST | /train | Create Model |
| POST | /download_synthetic_data_sample | Download Synthetic Data |
| POST | /download_synthetic_report_sample | Download Synthetic Report |
| POST | /download_synthetic_data_zip_sample | Download Synthetic Data Zip |
| POST | /list_models | List Models |
| POST | /train_cancel | Train Cancel |
| POST | /deploy | Generate Data |
| POST | /metrics | Metrics Data |
| POST | /generate_cancel | Train Cancel |
| POST | /list_all_synthetic_data | List Synthetic Data |
| POST | /list_model_synthetic_data | List Model Synthetic Data |
| POST | /list_all_processes | List All Processes |
| POST | /download_synthetic_data_original | Download Synthetic Data |
| POST | /download_synthetic_report_original | Download Synthetic Report |
| POST | /download_synthetic_data_zip_original | Download Synthetic Data Zip |

Figure 2: Snapshot of betterdata synthetic engine software UI.

3.2 Authenticators

Authentication modules are responsible for managing users on betterdata synthetic data engine software. Right now, we support two type of users- admin and non-admin/ normal. An admin user will have full control over the software like creating and managing users, delete user accounts, modify databases, ports, rules etc.

On the other hand, non-admin/ normal users will have permission to login to system, upload daatsets, train models, tune hyper parameters, generate and download synthetic data, metrics reports, check statuses etc.

3.2.1 Module: Create New Users

Table 1: Module: Create New Users

| API Name | EXAMPLE HTTP REQUEST | DESCRIPTION |
|----------------|----------------------|--|
| Create User | post /user | Creates a new user by providing unique username and password |
| Parameter Name | Value/Type | |
| username | string | <any unique name> |
| password | string | <any unique passord> |

The screenshot shows a web interface for creating a new user. At the top, it indicates a POST request to the endpoint /user. Below this, there is a section for parameters, which is currently empty. A 'Request body' section is visible, showing a JSON object with 'username' and 'password' fields. The interface includes buttons for 'Cancel', 'Reset', 'Execute', and 'Clear'.

Figure 3: Snapshot of Create User module.

3.2.2 Module: Login to Software

Table 2: Module: Login to Software

| API Name | EXAMPLE HTTP REQUEST | DESCRIPTION |
|----------------|----------------------|---|
| Login | post /login | User need to login with username and password to use the application. |
| | | |
| Parameter Name | Value/Type | Description |
| username | string | <already registered user name> |
| password | string | <already registered password> |

POST /login Login

Parameters

No parameters

Request body required

application/json

```
{  "username": "string",  "password": "string"}
```

Execute

Figure 4: Snapshot of Login module.

3.2.3 Module: Delete an User

Table 3: Module: Delete an User

| API Name | EXAMPLE HTTP REQUEST | DESCRIPTION |
|----------------|----------------------|--|
| Delete User | post /delete_user | Delete User is only done by admin who inserts the user to be deleted |
| Parameter Name | Value/Type | Description |
| username | string | <already registered user name> |
| password | string | <already registered password> |

The screenshot shows a REST client interface for the endpoint `POST /delete_user` (labeled 'Delete User'). The interface is divided into two main sections: 'Parameters' and 'Request body'. In the 'Parameters' section, there is a single query parameter named `targetusername` of type `string`, which is marked as required. The 'Request body' section is set to `application/json` and contains a JSON object: `{ "username": "string", "password": "string" }`. A blue 'Execute' button is located at the bottom of the interface.

Figure 5: Snapshot of delete an user module.