

# Set-up Instructions

## Using Sensors to Study Human Behavior

### OVERVIEW

This document provides instructions and advice on how to set-up and configure the systems that run the software of the project.

### DEFINITIONS

For the purposes of this document:

1. “Data hub” refers to the web application used to view the collected data.
2. “Data ingestion layer” refers to the application that receives and stores incoming data streams.
3. “Data collection layer” refers to the application that runs on a machine for the purpose of collecting sensor data.
4. “The server” refers to the machine that runs the data hub and the data ingestion layer.
5. “The repository” refers to the single repository which stores all the applications that are part of the infrastructure of this project. This includes the data hub, data ingestion layer, and the software that runs on the Raspberry Pi computers.

### SYSTEM ARCHITECTURE - Data Hub & Data Ingestion Layer

The data hub and data ingestion layer run in parallel on an Ubuntu Linux 18.04 server. The set-up instructions that follow assume a fresh installation of Ubuntu 18.04 as the server operating system. This ships with Python3.6 by default, which is the language used in all server logic.

### INSTALLING Data Hub & Data Ingestion Layer

This section contains sequential instructions on installing the repository on the server and configuring the Python virtual environment with the required packages.

**Step 1:** Install Python package manager, Python development tools, and Python3 virtual environment packages.

```
sudo apt-get install python3-pip python3-dev python3-venv
```

**Step 2:** Clone/install the repository on the server. It is recommended to use Git's cloning feature to accomplish this.

```
git clone https://your.repository.url.git
```

**Step 3:** Change directories to the root directory of the repository that was just cloned:

```
cd repository-name
```

**Step 4:** Create the Python virtual environment:

```
python3 -m venv venv
```

**Step 5:** Enter the virtual environment, so the required packages can be installed:

```
source venv/bin/activate
```

**Step 6:** Inside the virtual environment, install the Python packages that are the dependencies of the data hub and data ingestion layer using the provided Bash script:

```
sh install.sh
```

The website can now be run using the provided Bash script at the root of the repository:

```
sh run.sh
```

## INSTALLING MySQL & MySQL Server

This section contains sequential instructions for setting up the MySQL installation on the server, for use by the data hub and data ingestion layer.

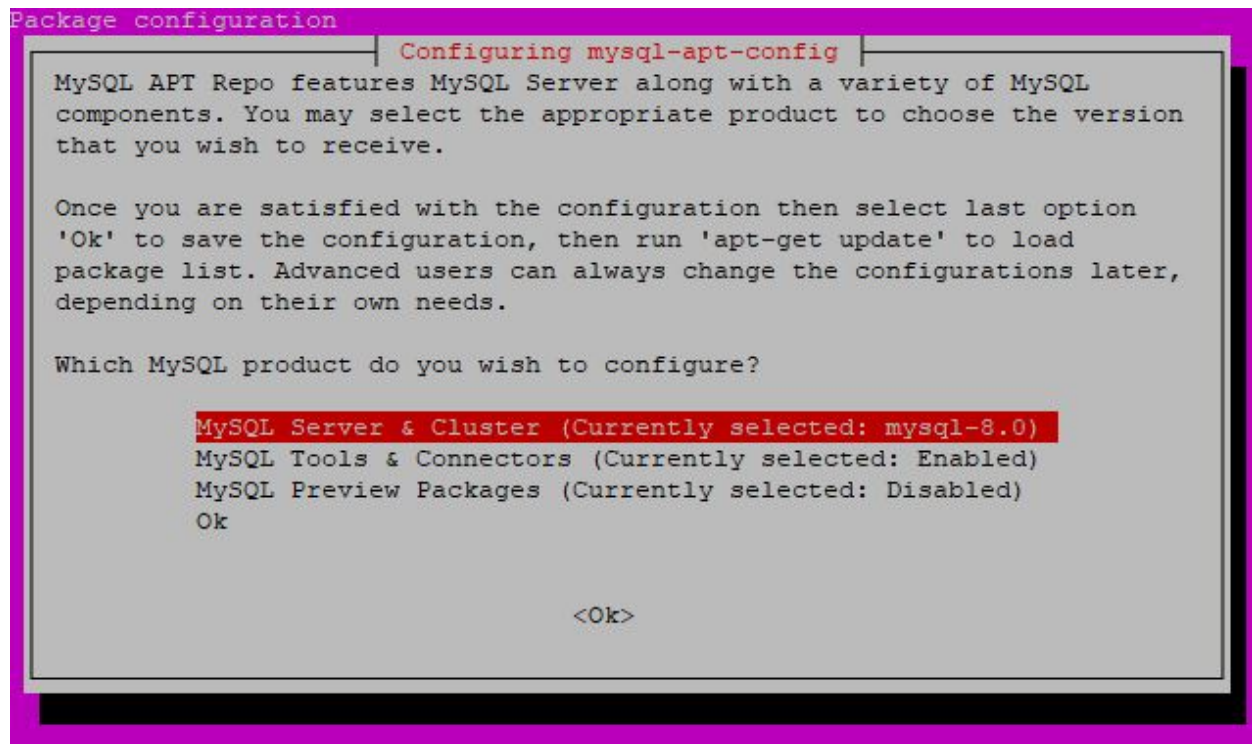
**Step 1:** The server requires at least MySQL 5.7.8. To fetch and install the recommended MySQL 8.13 with MySQL Server:

```
sudo apt-get install default-libmysqlclient-dev

wget -c https://repo.mysql.com/mysql-apt-config\_0.8.13-1\_all.deb
sudo dpkg -i mysql-apt-config_0.8.13-1_all.deb
rm mysql-apt-config_0.8.13-1_all.deb

sudo apt-get install mysql-server
```

**Step 2:** When prompted to configure the MySQL Installation, select the “Ok” option and ensure the following settings:



**Step 3:** Enter the MySQL Server shell:

```
mysql
```

**Step 4:** In the MySQL Server shell, create the MySQL user and database that the data hub and data ingestion layer will use:

```
CREATE USER 'CapstoneMySQLUser'@"localhost' IDENTIFIED BY 'CapstoneMySQLUserDbPw';
GRANT ALL PRIVILEGES ON *.* TO 'CapstoneMySQLUser'@"localhost';
CREATE DATABASE CapstoneData;
```

The MySQL user information (name, host, password) as well as the database name can be configured in the global configuration file. This file is named *config.json* and is located in the root directory of the repository. The above commands must then be run with the altered credentials and database name in the MySQL Server shell.

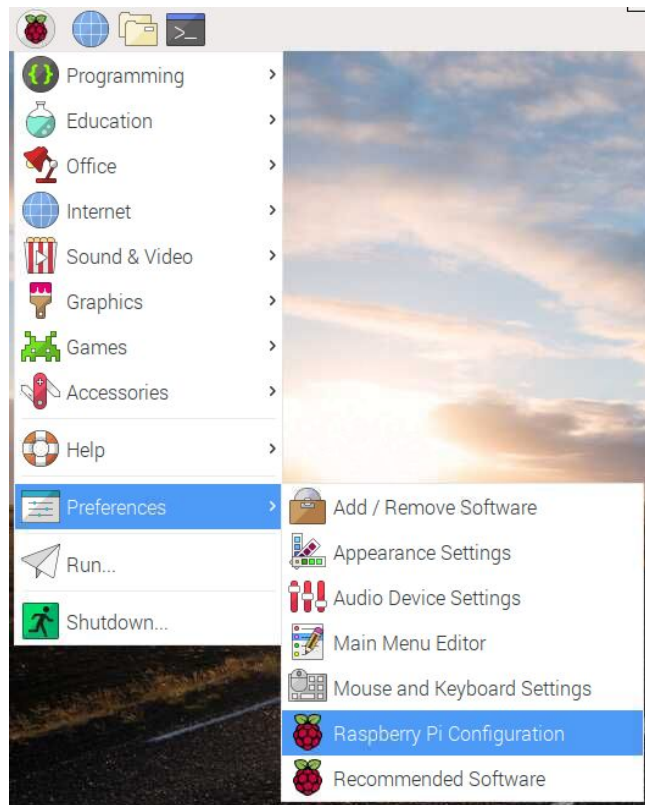
## INSTALLING DATA COLLECTION LAYER

This section contains sequential instructions on installing the repository on a machine for the purposes of collecting sensor data from a sensor attached to the machine.

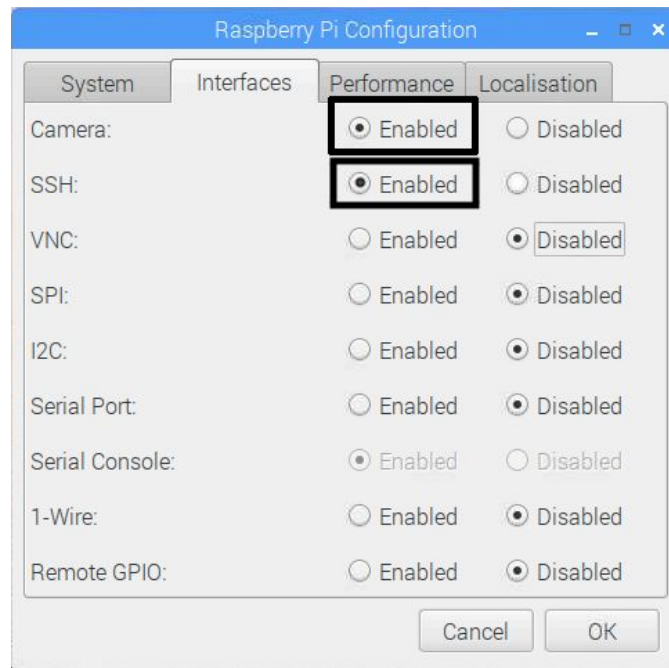
The repository is currently supported only on the Raspbian operating system. Other operating systems are compatible with the repository's sensor software, but would interfere with the collection of certain sensor types that the repository currently supports. The following Raspberry Pi set-up instructions in this section assume a fresh installation of Raspbian 10 (Buster) on a Raspberry Pi Model 4, as well as a properly connected Pi Video Camera and Sense HAT.

**Step 1:** Enable the usage of the Raspberry Pi's camera and connection via SSH

- 1) Go to the main menu and select Raspberry Pi Configuration:



- 2) Select the **Interfaces** tab and ensure that the camera and SSH are both enabled:



- 3) Reboot the Raspberry Pi for the changes to take effect.

## Step 2: Install Python audio interface (PyAudio)

- 1) Clone the PyAudio repository:

```
sudo git clone http://people.csail.mit.edu/hubert/git/pyaudio.git
```

- 2) Install the Python3 development package:

```
sudo apt-get install python-dev
```

- 3) Install PyAudio dependencies:

```
sudo apt-get install libportaudio0 libportaudio2 libportaudiocpp0 portaudio19-dev
```

- 4) Change into the root of the cloned PyAudio repository:

```
cd pyaudio
```

- 5) Run the PyAudio package setup file:

```
sudo python setup.py install
```

## Step 3: Build FFmpeg

- 1) Clone the FFmpeg repository:

```
git clone git://source.ffmpeg.org/ffmpeg --depth=1
```

2) Change directory to the repository that was just cloned:

```
cd ffmpeg
```

3) Configure the FFmpeg installation with MP4 and H264 support:

```
./configure --arch=armel --target-os=linux --enable-lbmp3lame --enable-gpl --enable-libx264 --enable-nonfree
```

4) Make the FFmpeg installation (takes about 10 minutes):

```
make -j4
```

5) Run the FFmpeg installation:

```
sudo make install
```

**Step 4:** Install Sense Hat software:

```
sudo apt-get install sense-hat
```

Reboot the Raspberry Pi after Step 4 for the installation to take effect.

**Step 5:** Clone/install the repository on the Raspberry Pi. It is recommended to use Git's cloning feature to accomplish this.

```
git clone https://your.repository.url.git
```

**Step 6:** Change directories to the data-collection directory of the repository that was just cloned:

```
cd repository-name/data-collection/
```

The Raspberry Pi Video server and Sense HAT stream can now be run by the main.py file at the data-collection level of the repository:

```
python3 main.py
```