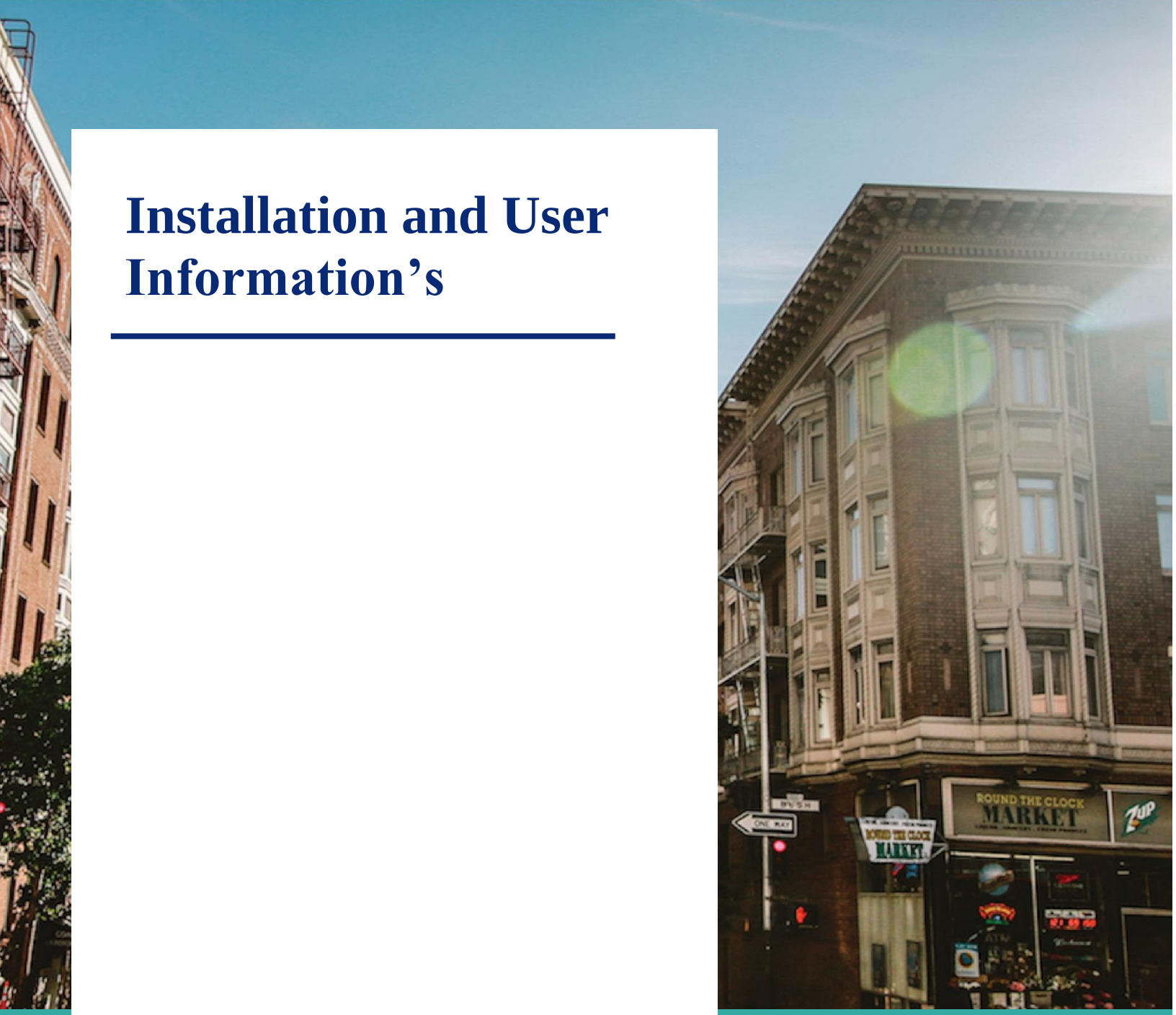
Installation and User Information's



Distributed Storage System

***Supervisor**: Mrs. Laura Thiele, Prof Mr. Sascha Bosse*
***Authored by**: Abhijith Remesh(221424), Baizil MD(221544)*

Overview

To monitor and display the current inventory levels of two warehouse components via GUI. The inventory data which is fetched from the scaling device needs to be displayed in real time. Furthermore, it should be able to store the log history of inventory data in a database and should be able to scale up flexibly with the increase in the number of warehouse components. The analysis of inventory data is also a major concern which can result in future order and stock replenishment forecasting.

Goals

- Interfacing the scaling device with the Raspberry pi.
- Establishment of real time communication via MQTT protocol.
- Fetching the real time inventory levels of two types of warehouse components.
- GUI Dashboard development to display the current inventory levels.
- Storing the inventory data to a database on time interval basis or demand basis.
- Leveraging the inventory data for exploratory data analysis.

In the Box

The whole distributed system contains the following items,

- PCE-BHS-10000N Weight Scale
- Raspberry Pi 3B
- USB Cable

Requirements

Minimum Pc Requirements

- Microsoft Windows 7 or Later
- Internet Connectivity

Minimum Software Requirements

- Node-RED
- MySQL
- MQTT

Installation Manual

Setting up Node-RED on windows

- Navigate to this https://nodejs.org/en/ for installing node.js
- Download and install the MSI file.
- After installation, open a new command prompt and type
  - 'node --version && npm --version'
    - To ensure the correct installation which should respond with version numbers ideally.
- To install Node-Red, execute the following at the cmd
  - npm install -g --unsafe-perm node-red.
- To run Node-RED, open the cmd and type 'node-red'.
- Node-Red starts listening on http://127.0.0.1:1880/.

Adds the following node instances into the Node-RED by running these commands in the command prompts.

- Dashboard node: - npm install node-red-dashboard
- Mysql node: - npm install node-red-node-mysql

- Python shell node: - npm install node-red-contrib-pythonshell
- Persist node: - npm install node-red-contrib-persist

Setting up Node-RED on raspberry Pi

- To install Node-RED on raspberry pi, type the following command in raspberry pi,
  - bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
- To run Node-RED in raspberry pi, type the below command
  - node-red-start
- To run Node-RED as a service on pi for auto start at system boot, type the following command:
  - sudo systemctl enable nodered.service
- Node-R starts listening on http://raspberrypiipaddress:1880/.

Adds the following node instances into the Node-RED by running these commands in the putty consoles.

- Serial-in node: - npm install node-red-node-serialport
- Dashboard node: - npm install node-red-dashboard

Install and Setting up MySQL and MySQL workbench on Windows

Navigate through https://www.mysql.com/downloads/ for installing MySQL. After successful installation, MySQL server starts listening on port 3306 and the database can be created using the MySQL workbench tool by executing a SQL script.

In order to authenticate the MySQL node in the Node-RED, Run the SQL script given below,

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '<yourpassword>'

Create a database distributedstorage,

A sample model of the database is shown below:

Id	Article number	Article name	Unit Weight
1	731308	Waschenklammer	4g
2	708763	Tintenpatrone	1.4g
3	700555	Screws	3g

The SQL script which creates the corresponding database and inserts the corresponding values is shown the table:

```
CREATE DATABASE `distributedstorage`;
CREATE TABLE `distributedstorage`.`inventorydb` (
`id` INT NOT NULL AUTO_INCREMENT,
`article_number` INT NOT NULL,
`article_name` VARCHAR(45) NOT NULL,
`unit_weight` FLOAT NOT NULL,
PRIMARY KEY (`id`),
UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,
UNIQUE INDEX `article_number_UNIQUE` (`article_number` ASC) VISIBLE,
UNIQUE INDEX `article_name_UNIQUE` (`article_name` ASC) VISIBLE);
```

```
INSERT INTO `distributedstorage`.`inventorydb` (`id`,`article_number`,`article_name`,`unit_weight`)
VALUES ('1','731308','waschеklammer', 4);
INSERT INTO `distributedstorage`.`inventorydb` (`id`,`article_number`,`article_name`,`unit_weight`)
VALUES ('2','708763','tintenpatrone', 1.4);
INSERT INTO `distributedstorage`.`inventorydb`
(`id`,`article_number`,`article_name`,`unit_weight`)
VALUES ('3','700555','screw', 3);
```

Create another database inventorylog,

time	scale_id	article_number	unit_weight	obtained_weight	count

```
CREATE DATABASE `inventorylog`;
CREATE TABLE `inventorylog`. `datalog` (
`time` varchar(45) NOT NULL,
`scale_id` varchar(45) NOT NULL,
`article_number` int(11) NOT NULL,
`unit_weight` float NOT NULL,
`obtained_weight` float DEFAULT NULL,
`count` int(11) DEFAULT NULL) ;
```

Setting up the mosquitto client on windows

- Navigate to this https://mosquitto.org/download/.
- For windows 64.
    - Download mosquitto-1.6.7-install-windows-x64.exe (~1.4 MB) (64-bit build, Windows Vista and up, built with Visual Studio Community 2017).
- Install the executable.
- Open a command prompt and go to the mosquitto install directory and type mosquitto.
- Type ‘mosquitto -h’ for help.
- Type ‘mosquitto -v’ for
- verbose, more detailed understanding.
- To run mosquitto broker as a service to run automatically at reboot,
    - Open the power shell command prompt as admin and navigate to the directory where mosquitto is installed and type ‘./mosquitto install’.

- Open a normal cmd as admin and type the below commands:
  - sc query mosquitto
  - sc start mosquitto
  - sc query mosquitto
- Reboot the machine.
- Open a normal cmd as admin and type ‘sc query mosquitto’ and you should see the service in running state.
- mosquitto broker listens on port 1883.
- Run ‘netstat -a’ at the cmd and check if ‘0.0.0.0:1883’ is in listening state.

o The mosquitto client at the local machine end acts as the subscriber and subscribes the inventory information based on the topic.

Setting up the mosquitto broker on raspberry Pi

o Type the command
  - sudo apt install -y mosquitto mosquitto-clients

o To run mosquitto as service, type as follows
  - sudo systemctl enable mosquitto.service.

o Test for the installation as follows which displays the version installed
  - mosquitto -v

o The mosquitto broker starts listening on http://raspberrypiipaddress:1883/.

o The mosquitto broker at the pi end acts as the publisher and publishes the inventory information on a topic.

Install python and relevant libraries for applying machine learning algorithms

Install python using anaconda distribution-**https://www.anaconda.com/distribution/.**

This will install all the necessary packages/libraries in the system, in case if it's not installed then follow the below steps,

Install the following libraries/packages,

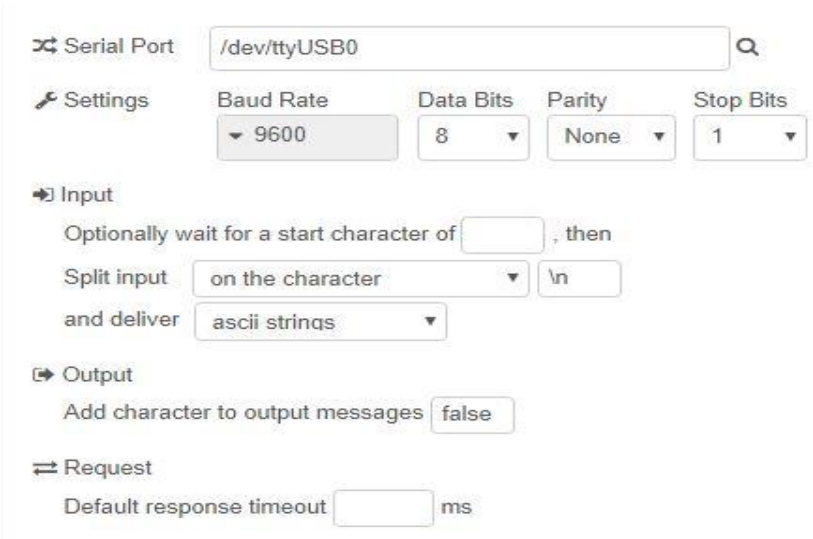
- mysql.connector.python
  - pip install mysql-connector-python
- Numpy
  - pip install numpy
- Pandas
  - pip install pandas
- Scikit-learn
  - pip install -U scikit-learn

Operational Manual

In order to setup the system working, some prerequisite scripts must be made available on your directory. Those can be found in the folder 'Distributed Storage System' in the student branch of the GitLab repository -**https://gitmrcc.iti.cs.ovgu.de/lthiele/distributed-storage-system/-/tree/Student** .Create a folder with name 'Distributed Storage System' in the 'C' drive of your system as this(C:\Distributed Storage System) would be used in the Node-RED flow. Download all the prerequisite folders in to the newly created folder. Then follow the below instructions carefully,

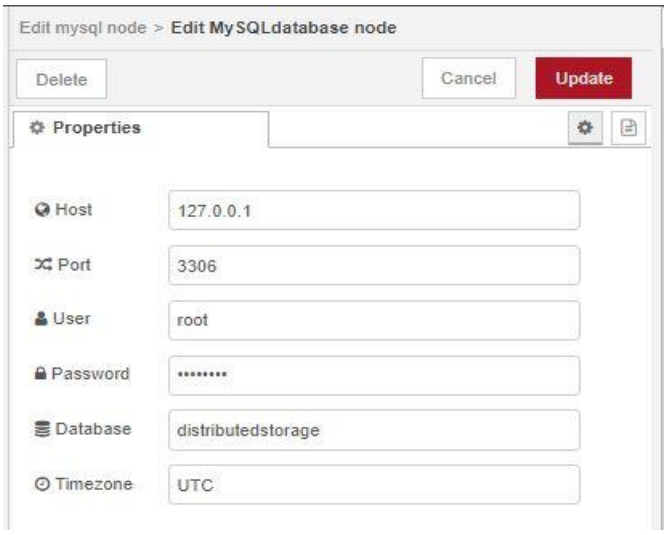
- Power up the Raspberry Pi and weight scale, connect them using a USB cable.

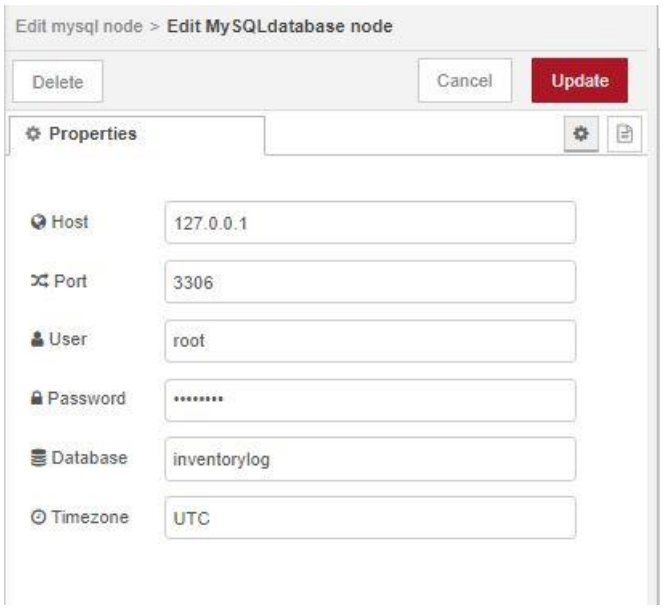
- Identify the ip address of the Raspberry Pi and navigate to it Node-RED portal (ip address:1880) as the node red is running as service in the Raspberry Pi.
- Ensure the files devicenum.sh and piserial.sh (can be found in the prerequisite folder) should be made available in the Raspberry Pi (/home/pi) as the Node-RED flow makes use of it.
- Import the Node-RED flow (Raspberrypi flow.json) at the Raspberry Pi which is available in the prerequisite folder.
- Run the command, *ls -l /dev/tty** or Run the command, *dmesg | grep "tty"* which tells the serial port where the USB cable is connected.
- Specify this serial port while configuring the serial.in node (scale) in the Node-RED flow at Raspberry Pi. Also specify the Baud Rate, Data Bits, Parity and Stop Bits as shown below,



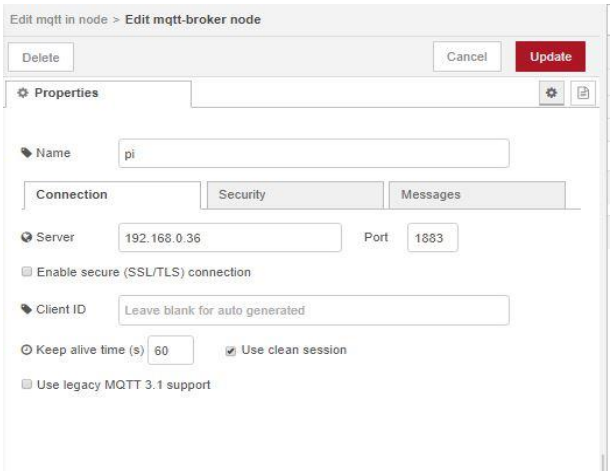
- To start the Node-RED instance in Windows, Open the command prompt and type 'node-red' and navigate to the its Node-RED portal (localhost:1880).

- Import the Node-RED flow (Localmachine flow.json) at the local machine which is available in the prerequisite folder.
- Ensure that all the prerequisite scripts are available in a specific file path as specified in the Node-RED flow (C:\Distributed Storage System)
- Specify the username and password credentials used in the SQL workbench in the mysql node's user and password fields and mention the same database as created in the MySQL workbench on the Node-RED flow at local machine.





- Specify the ip address of the Raspberry Pi at all MQTT nodes in the Node-RED flows of both Raspberry Pi and local machine,



- First Deploy the Node-Red flow at the Raspberry Pi end and then deploy the flow at the local machine.