**DELOY TCP SERVER WITH UI**

**Note:**

* The Python version must be **3.10**, and **aiohttp** must be **3.11.13**.
* After completing the configuration, please log out of the server using the button in the sidebar.

## **I. Setup Server**

* **Step 1: Create a Project Folder**

First, create a folder to store your projects. In this guide, we will create a folder named "TCPServerUI" and navigate into it:

|  |
| --- |
| mkdir TCPServerUI  cd TCPServerUI |

* **Step 2: Check if Python 3.10 is installed**

Verify whether Python is installed on your system by running:

|  |
| --- |
| python3 --version |

If Python 3.10 is not installed, use the following commands to install it:

|  |
| --- |
| sudo apt update && sudo apt install python3.10 -y |

* **Step 3: Create a Virtual Environment**

Set up a virtual environment inside the project folder

|  |
| --- |
| python3 -m venv .venv |

* **Step 4: Activate the Virtual Environment**

Activate the virtual environment using the command below

|  |
| --- |
| source .venv/bin/activate |

* **Step 5: Clone the Project Repository**

Since this repository is private, you need to set up an SSH key to authenticate and clone it. Follow these steps carefully.

* + **Open a New Terminal and Generate an SSH Key.**

|  |
| --- |
| ssh-keygen -t rsa -b 4096 -C "your\_email@example.com" |

Replace "your\_email@example.com" with your GitHub email.

When prompted:

* + - Press Enter to accept the default location (~/.ssh/id\_rsa).
    - Optionally, set a passphrase for added security or press Enter to skip
  + **Copy Your Public Key.**

|  |
| --- |
| cat ~/.ssh/id\_rsa.pub |

Copy the output (it starts with ssh-rsa).

* + **Add It to GitHub.**
    - Click the **"add a new public key"** link from your screenshot.
    - In the **Title** field, enter something like "My Laptop SSH Key".
    - In the **Key** field, paste the copied SSH key.
    - Click **Add SSH Key.**

A screenshot of a computer

AI-generated content may be incorrect.

* + Clone the repository.

|  |
| --- |
| git clone [git@github.com](mailto:git@github.com):CloudBurst-Australia/iot-server-dreamsedge.git  cd iot-server-dreamsedge |

* + Checkout to branch **dev\_v1.0.**1 to test.

|  |
| --- |
| git checkout dev\_v1.0.1 |

* **Step 6: Install Required Libraries**

Install the necessary dependencies listed in the **requirements.txt** file:

|  |
| --- |
| pip install --no-cache-dir -r requirements.txt |

* **Step 7:** **Run the Application**

After pulling the code from GitHub, run the command below to start the TCP server.

|  |
| --- |
| ./run-app.sh |

* When the server starts, it will display the IP address for accessing the UI configuration in the terminal.

A screen shot of a computer

AI-generated content may be incorrect.

* Access the server using the IP address: 192.168.0.125:8000. The login page will be displayed on the screen.

A screenshot of a computer

AI-generated content may be incorrect.

## **II. Configuration server with UI**

* If you want to configuration server, you need to login with the account.

|  |
| --- |
| Username: admin  Password: 123 |

A screenshot of a computer

AI-generated content may be incorrect.

* When logged in successfully, it will navigate to the **server\_status** page, which displays the state of the TCP server and AWS connection.
* When the server starts, it will initially be running.
* AWS connection means that if a device connects to the TCP server and successfully pushes data to the DreamsEdge server, the state will change to Connected..

A screenshot of a computer

AI-generated content may be incorrect.

* To configure the TCP server, go to Configuration TCP Server in the left sidebar.
* At this point, you will see the **TCP Server Configuration**, which includes TCP Server Control, Enter Username and Password, Facility Lists, and Select Facility ID.
  + **TCP Server Control**: Start/Stop TCP Server.
  + **Username and Password**: Used to obtain an authentication token.
  + **Facility Lists**: Displays the current facility list from the DreamsEdge platform.
  + **Select Facility ID**: Select an ID to register the TCP server.
* Scrolling down, you will see the **AWS Connection Configuration**.
  + **Certificate Files**: Used to upload certificate files from AWS Server.
  + **Update AWS IoT Endpoint**: Used to update the AWS IoT endpoint and check the connection with it.

A screenshot of a computer

AI-generated content may be incorrect.

* To start configuring TCP Server, you must stop server. After stop, all button below will enable to submit.

A screenshot of a login form

AI-generated content may be incorrect.

* + **Username and Password**: If you enter the correct username and password, a success message will be displayed below the button. If the credentials are incorrect, it will display "Wrong username or password”.

A screenshot of a login page

AI-generated content may be incorrect.

* + After logging in successfully, the facility list will be displayed below, allowing you to choose a facility ID to register the TCP server. After pressing **Submit**, the information will be updated on the server.

A screenshot of a computer

AI-generated content may be incorrect.

* To configuration AWS:
  + At first, it will show the current certificate files that included in tcp server and current aws IoT endpoint.

A close up of a certificate

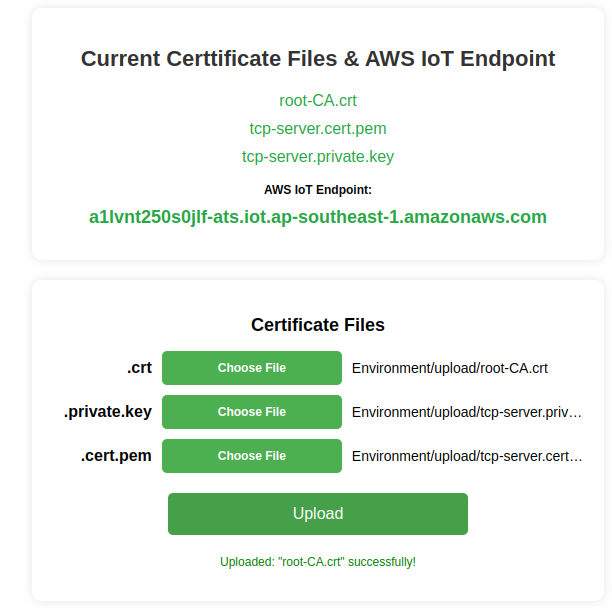
AI-generated content may be incorrect.

* + To upload the new certificate files, follow the tail of file to upload, it just accept the file that have tail: .crt, .private.key, .cert.pem. If you upload successfully, it will update the name of file into the “Current Certtificate Files & AWS IoT Endpoint”.

A screenshot of a certificate

AI-generated content may be incorrect.

* + If the uploaded file is valid, it will be successfully uploaded to the server.



* + If the file is not valid, it will not be uploaded.

A screenshot of a computer

AI-generated content may be incorrect.

* + **Update AWS IoT Endpoint**: Enter your AWS endpoint and press **"Update"**. After that, your endpoint will be displayed in **"Current Certificate Files & AWS IoT Endpoint"**
  + **Check Connection**: To check the connection to your endpoint, press **"Check Connection"**. If successful, it means your endpoint is correct.

A screenshot of a computer

AI-generated content may be incorrect.

* + If your endpoint is correct, it will display **“Connection successful”.**

A screenshot of a computer

AI-generated content may be incorrect.

* + If it is incorrect, it will display **“Connection failed”.**
* After completing the configuration, start the TCP server to connect and receive data from devices. I tested the connection and data reception from one device. After that, I pushed the data to the DreamsEdge Web to verify that the connection between the TCP server and DreamsEdge was successful. The result is shown below.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## **III. Configuration device**

* Power on the AeroSense Wavve device, and run the AeroSenseWavveTool on your computer. Click “BlueTooth”.

A screenshot of a computer

AI-generated content may be incorrect.

* Click Scan to scan for nearby Bluetooth devices. Select the Bluetooth with the name " AeroSense Wavve ". If the Bluetooth name is not found, press and hold the reset button on the back of the device for 3 seconds to reset the device, and then click Scan to search again. Click Connect to connect to the Wavve. If the connection is successful, the Bluetooth State will be Connected.

A screenshot of a computer

AI-generated content may be incorrect.

* + Enter the name and password of your WiFi network, and click Set.
  + Find the IP of the **Ubuntu Server laptop**. **The TCP port is 8899.**
  + Enter the local IP address of **Ubuntu Server laptop** and Port and click Set. To create a Server with Local IP by AerosenseWavve-SDK.

A screenshot of a computer

AI-generated content may be incorrect.

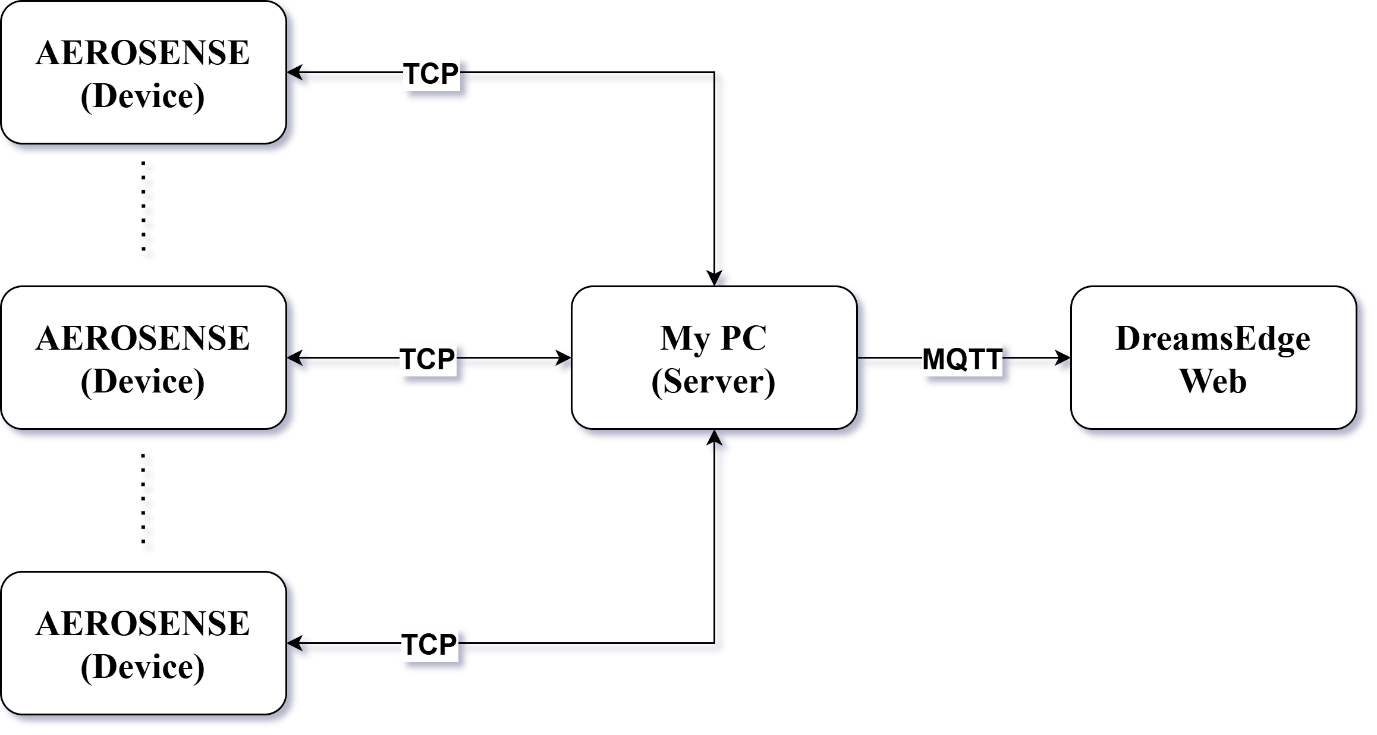
* Reset the power of the device.
* To check the IP address of your **Ubuntu Server laptop**. Follow the instructions below.

|  |
| --- |
| sudo apt update  sudo apt install net-tools  ifconfig |

A computer screen shot of a computer program

AI-generated content may be incorrect.

## **IV. Test Model and Current Results**



* + We have optimized the TCP server code and tested data updates from devices to the DreamsEdge Web to check for packet loss or whether the server can handle multiple devices.
  + The result shows that no packets were lost across the **3 devices** during the 30-minute testing period.

A screenshot of a computer

AI-generated content may be incorrect.

## **V. TCP Server Stress Testing**

* Instead of testing with only three devices, we plan to create multiple connections from devices to the TCP server to evaluate whether the server can handle the entire data load. Additionally, we will measure the packet loss rate. Our goal is to test this model with more than 100 devices.