

## **Raspberry Pi as an Intermediary Between Printer and End Devices**

The Raspberry Pi can serve as an effective intermediary to facilitate printing from devices that are not directly compatible with a given printer. This setup enables any end device to print using the printer via the Raspberry Pi. It also provides the feedback system to the user that displays the currently active printers on the Rpi.

### ***Hardware:***

- **Printer**
- **Raspberry Pi 4 Model B**
- **End Devices**

### ***Software:***

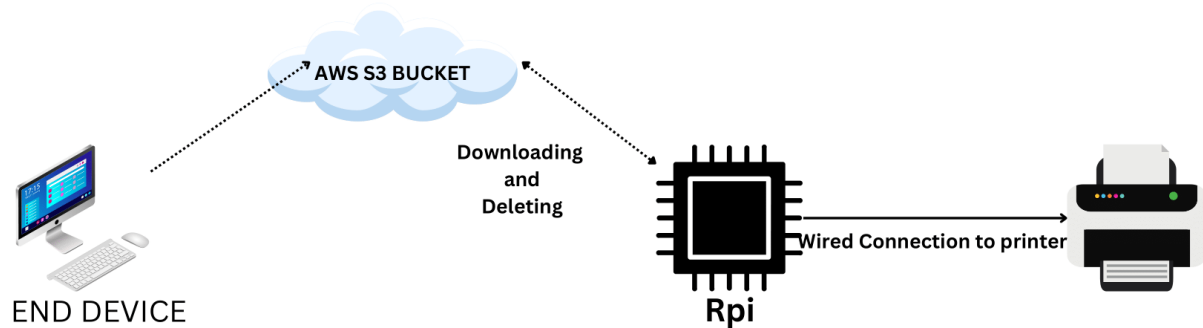
- **AWS:**
  - Used for communication between the end device (which sends the print command) and the Raspberry Pi.
  - Utilized to create the API.
- **CUPS (Common Unix Print System):**
  - Connects the Raspberry Pi to the printer and manages print jobs.
- **Python:**
  - Handles downloading the file to be printed from the server and sending the print command to the printer.
- **Postman:**
  - To test the API

### ***Workflow: (for printing)***

- 1. File Upload:**
  - The end device uses the API created on AWS to upload the file to an AWS S3 bucket.
- 2. File Retrieval and Printing:**
  - A Python script running on the Raspberry Pi continuously monitors the AWS S3 bucket for new files.
  - When a new file is detected, the Raspberry Pi downloads and stores it temporarily.
  - The Python script then sends a print command to the printer.
- 3. Post-Printing Cleanup:**

- After the printing is completed, the downloaded file on the Raspberry Pi is deleted.
- The file in the S3 bucket is also deleted to maintain cleanliness and free up storage.

This process ensures seamless printing from any device, regardless of its direct compatibility with the printer, leveraging the Raspberry Pi as a versatile and efficient bridge.



### ***Workflow: (for feedback system):***

1. Detecting the active printers:
  - The python script will detect the currently active printers that are connected to Rpi using CUPS.
  - It will post the list of active printers to AWS IoT Core.
2. Accessing the List of active printers:
  - The API will be created to access the recently posted message on AWS IoT Core.
  - Calling that Get API will give user the list of all currently active printers.

## **Step by Step process to create the System**

1. **Creating connection form End device to AWS S3 bucket:**

[API Creation.docx](#) (This contains the step-by-step process to create an API to put file in the S3 Bucket)

To use many printers at a time on Rpi **different APIs** will be created that contains the put method to upload in **different** BUCKET.

## **2. Connecting the Printer to Raspberry pi using CUPS:**

[Add printer to RPI](#) (This contains the step-by-step process to connect printers to Rpi via USB using cups)

## **3. Creating the AWS IoT core For Feedback system:**

[Configure AWS IoT core](#) (This contains the step-by-step process to Configure AWS IoT core)

## **4. Setting Rpi to do print jobs and provide feedback using python**

[Setting Rpi](#) (This contains the step-by-step process to setup Rpi to preform necessary function and the required python script)

## **5. Testing**

The above system can be tested using the Postman application:  
currently the system contains two printer that are connected to Rpi and have the following APIs:

### **Printer 1:**

Curl:

```
curl --location --request PUT 'https://cxqeevq251.execute-api.us-east-1.amazonaws.com/dev/printfile-storage/label.jpg' \
--header 'Content-Type: application/octet-stream' \
--header 'file-name: print.pdf' \
--data '@/X:/Desktop/Printer bridge/pp.jpg'
```

API:

<https://cxqeevq251.execute-api.us-east-1.amazonaws.com/dev/printfile-storage/label.jpg>

### **Printer 2:**

```
curl --location --request PUT 'https://x6eekbvni9.execute-api.us-east-1.amazonaws.com/dev/file-storage-2/print.jpg' \
--header 'Content-Type: image/jpeg' \
--data '@/X:/Desktop/Printer bridge/pp.jpg'
```

API:

## **Setting up Windows machine for acting as Intermediate print server.**

- AWS configuration does not change.
- Setup windows machine with AWSCLI environment.
  - Run `aws configure` command in your terminal or command prompt.
  - Enter your AWS Access Key ID, Secret Access Key, region (e.g., `us-east-1`), and output format (e.g., `json`).
- Install the following libraries:
  - `pip install boto3`
  - `pip install pywin32`
  - `pip install AWSIoTPythonSDK`
  - `pip install schedule`
- Run the following code to get the desired output:  
[Python Code](#)
- The same above API can be used to send file to printer via AWS and Windows machine