**IoT – Create a Job with RaspberryPi**

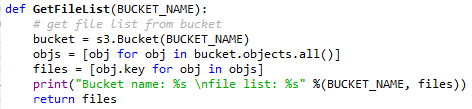
**0. Contents**  
- Creating and testing a job with Raspberry Pi  
- ‘.txt’ file is used instead of ‘.img’ file  
- Downloading txt file from S3, if the current version is greater than the new version

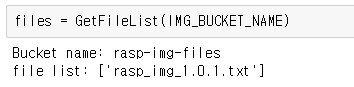
1. **Implementation**

**1-1. ‘*GetFileList’***

Gets the file list in the S3 bucket.

|  |  |
| --- | --- |
| Input | BUCKET\_NAME: bucket name in your AWS |
| Output | All file names in a list formation |

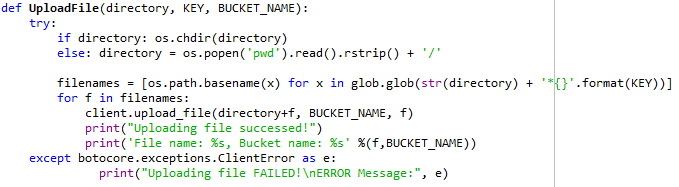
**Code:**

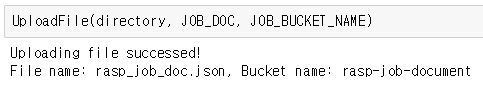
**Example:**

**1-2. ‘*UploadFile’***

Upload the file on the S3 bucket.

|  |  |
| --- | --- |
| Input | directory: file path in your device KEY: file name to upload BUCKET\_NAME: bucket name in your AWS |
| Output | “Successed” if succeed “Failed” if failed |

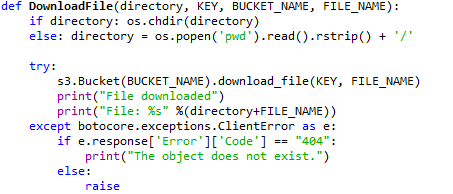
**Code:**

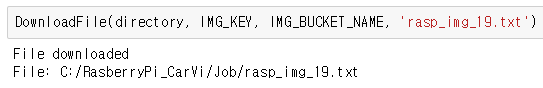
**Example:**

**1-3. ‘*DownloadFile***

Download the file on the S3 bucket.

|  |  |
| --- | --- |
| Input | directory: file path in your device KEY: file name to upload BUCKET\_NAME: bucket name in your AWS  FILE\_NAME: new file name |
| Output | “Downloaded” if succeed “Does not exist” if failed |

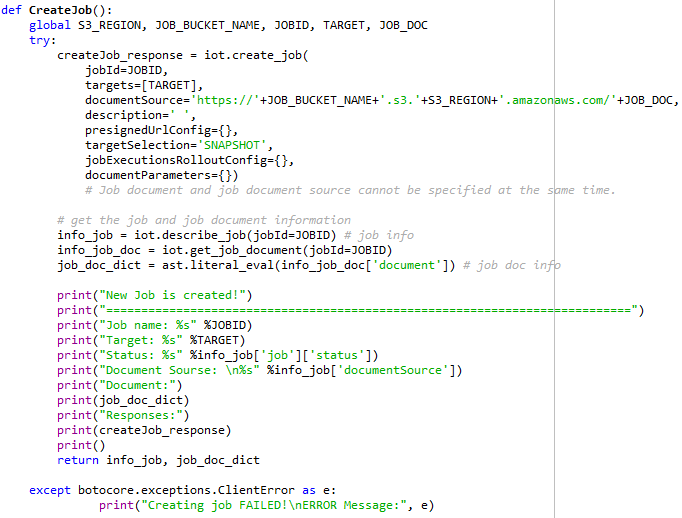
**Code:**

**Example:**

**1-4. ‘*CreateJob’***

Gets the file list in the S3 bucket.

|  |  |
| --- | --- |
| Input | - |
| Output | description of job and the job document |

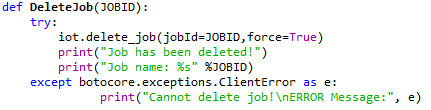
**Code:**

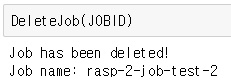
**Example:**

**1-5. ‘*DeleteJob’***

Gets the file list in the S3 bucket.

|  |  |
| --- | --- |
| Input | JOBID(job name) |
| Output | - |

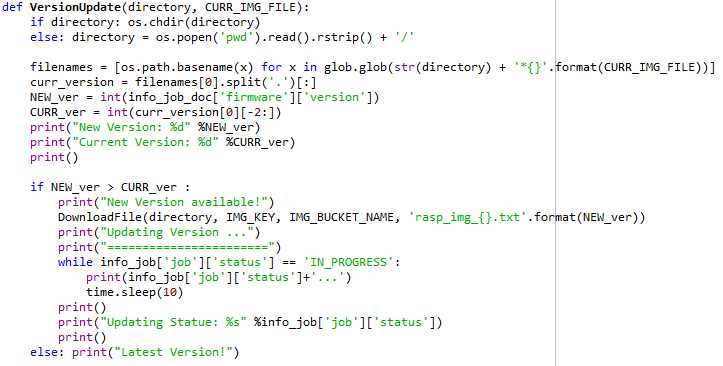
**Code:**

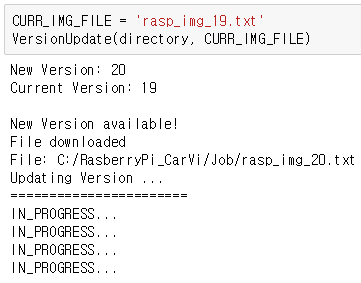
**Example:**

**1-6. ‘*VersionUpdate’***

Gets the file list in the S3 bucket.

|  |  |
| --- | --- |
| Input | directory(file location), CURR\_IMG\_FILE(IMG file with the version) |
| Output | - |

**Code:**

**Example:**

1. **Basic Info**

**2-1. S3 buckets**

Two S3 buckets are created. One for IMG files and one for job documents.

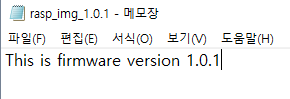
|  |  |
| --- | --- |
| Bucket name | Description |
| *rasp-img-files* | contains IMG files. |
| *rasp-job-document* | contains job documents |

  
[Fig 1] S3 buckets

**2-2. IMG files**

The text file is used for testing.

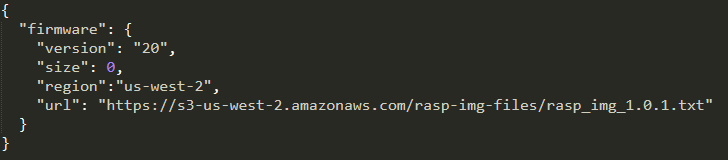
|  |  |
| --- | --- |
| File name | Description |
| *rasp\_img\_1.0.1.txt* | alternate the IMG file |

  
[Fig 2] ‘rasp\_img\_1.0.1.txt’ file contents

**2-3. Job documents**

Contains the job definition in a json format.

|  |  |
| --- | --- |
| File name | Description |
| *rasp\_job\_doc.json* | *version* – version of IMG file  *size* – size of IMG file  *region* – S3 bucket region  *url* – link to the IMG file |

  
[Fig 3] ‘rasp\_job\_doc.json’ file contents

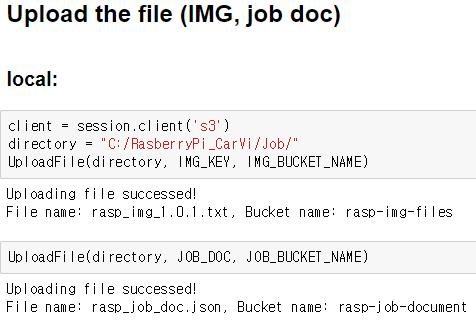
1. **Steps**

**3-1. Upload the new IMG file on S3**

Firstly, open ‘S3’ session.

Upload the new IMG file on ‘***rasp-img-files***’ S3 bucket.  
Here, file is named ***‘rasp\_img\_1.0.1.txt’***.

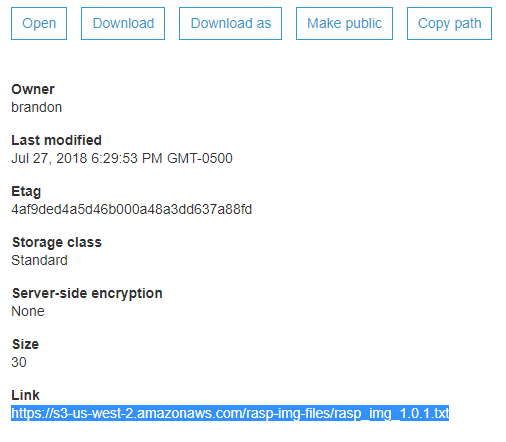
**‘*UploadFile’*** function is used.

  
[Fig 4] Example code of uploading the IMG file. (IMG file name = ‘*rasp\_img\_1.0.1.txt*’)

  
[Fig 5] ‘*rasp-img-files*’ bucket

**3-2. Copy the link of the new IMG file**

Click the new IMG file and copy the link of the file. Need it to edit the job document later.

  
[Fig 6] ‘*rasp\_img\_1.0.1.txt*’ file link

**3-3. Create a new job document**

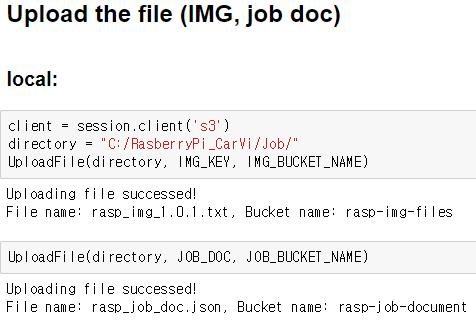
Create a new job document in the json format and add the IMG file link copied before.  
New version = ‘20’.  
**Make sure updating ‘*version’* and ‘*link’*.**

  
[Fig 7] ‘*rasp\_job\_doc.json*’ file

**3-4. Upload the new job document on S3**

Upload the job document file on ‘*rasp-job-document*’ S3 bucket.  
Here, file is named ‘*rasp\_job\_doc.json*’.

**‘*UploadFile’*** function is used.

  
[Fig 8] Example code of uploading the job document. (file name = ‘*rasp\_job\_doc.json*’)

  
[Fig 9] ‘*rasp-job-document*’ bucket

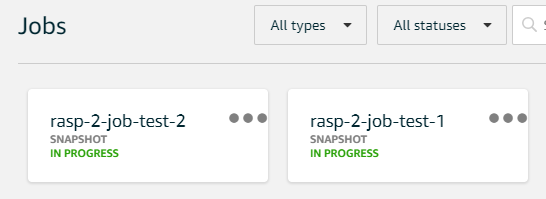
**3-5. Create a new job**

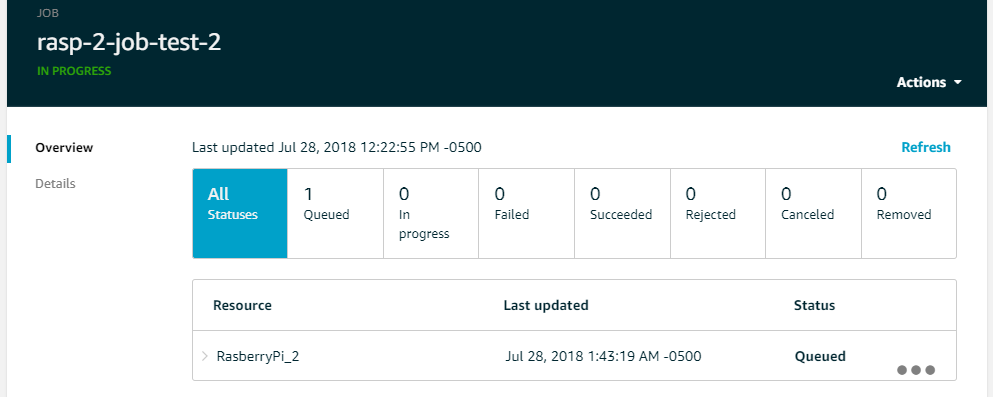
Create a new job with the job document. Define the region name to open the iot session. The region can be found from the endpoint. Set the ‘jobid’ and ‘target’ to create the job.

‘jobid’ must ne unique for AWS account.   
‘target’ can be a list of things and thing groups.

‘***CreateJob’*** function is used. This function takes no parameters and returns the description of the new job and the job document that has been used.

  
[Fig 9] Example output from ‘*CreateJob’* function

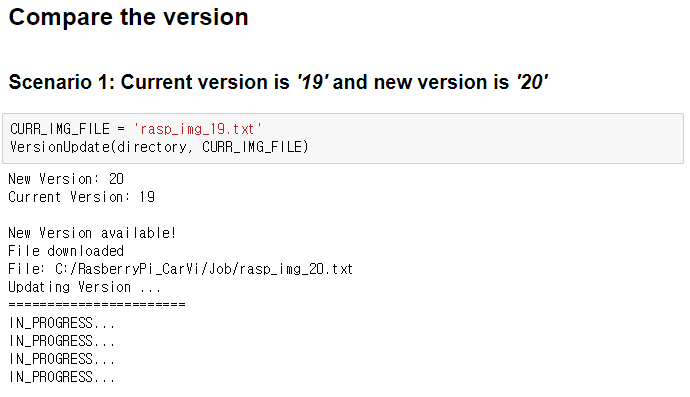
  
[Fig 10] job page on AWS

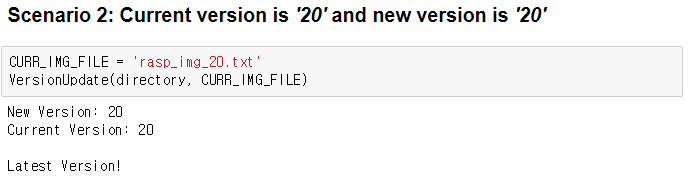
  
[Fig 11] The new job’s status on AWS

**3-6. Compare the version**

Compare the **‘version’** between **the job document** and **the current version (S3)**.  
If the job document has greater version, **download the new IMG file** from the url referred in the job document.

Process will be represented like below.

  
[Fig 12] Example code and output if the update is in progress

  
[Fig 13] Example code and output if the version is the same

1. **Raspberry Pi**

**4-1. Python files**

**4-1-1. ‘jobsSample.py’**

This python file accesses on AWS IoT.

Completes the jobs that are connected to the things or thing groups.

The thing name or thing group name is required on the command line.

**Main task:**

- Comparing versions in the job document (current version vs new version)  
- Getting the job information and the job document  
- Checking the job status (such as IN\_PROGRESS, CANCELED, SUCCEEDED)  
- Downloading files from the S3 bucket (‘*.img*’ or ‘*.txt*’)  
- Completing updating the jobs

**4-1-2. ‘runJob.py’**

This python file is running the command line with the required parameters.

Parameters: endpoint, rootCA, certificate, thing name

**Example:**

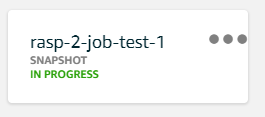
*python jobsSample2.py -e azozmhv0nd77.iot.us-west-2.amazonaws.com -r root\_CA.pem -c 3ddd7abd2c-certificate.pem.crt -k 3ddd7abd2c-private.pem.key -n RasberryPi\_2*

**4-2. Steps**

**4-2-1. Create a job**

Create a job by running ‘***CreateJob’*** function from the Jupyter.





**4-2-2. Run ‘runJob.py’**

Execute the python file.

Output examples are shown below.

**Example 1: Current version is '19' and new version is '20'**

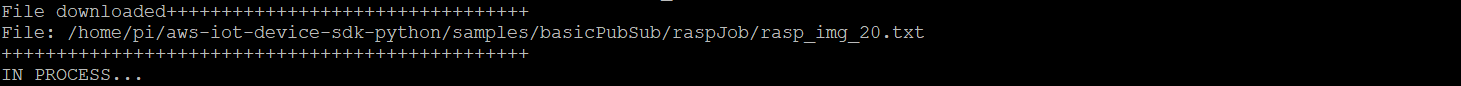
Assume that the device has version 19.



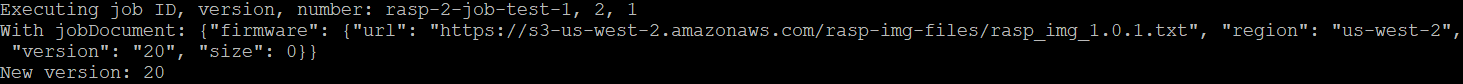
Check the version

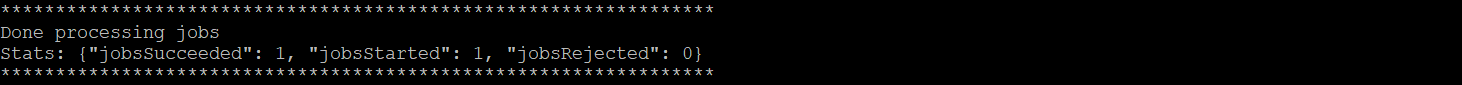


Download the file (.img or .txt)

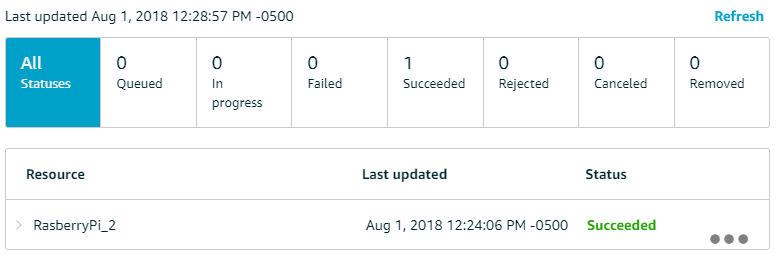


Complete the job





The job status has been changed to ‘SUCCEEDED’.



The device has the version 20.

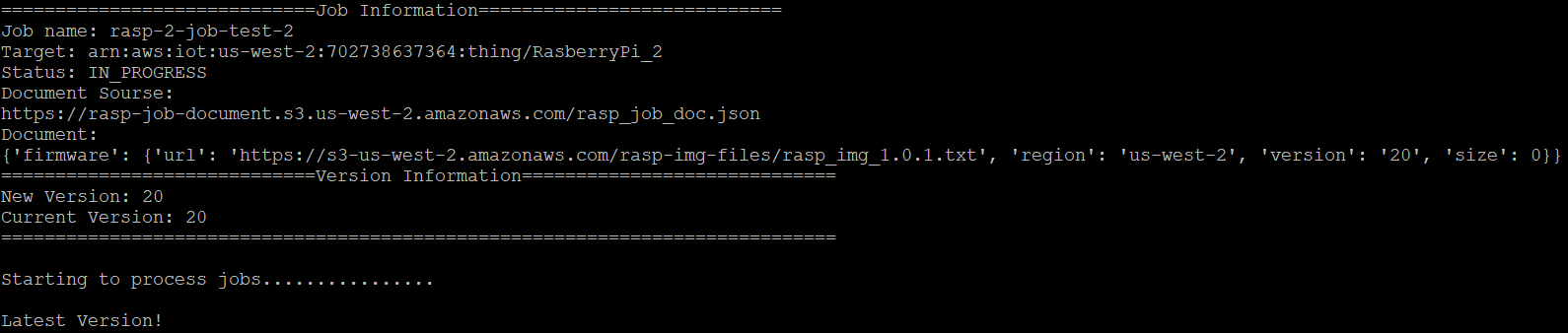


**Example 2: Current version is '20' and new version is '20'**

Assume that the device has version 20.

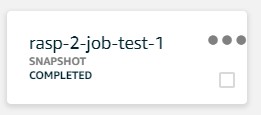


Check the version and show that the device has the latest version.



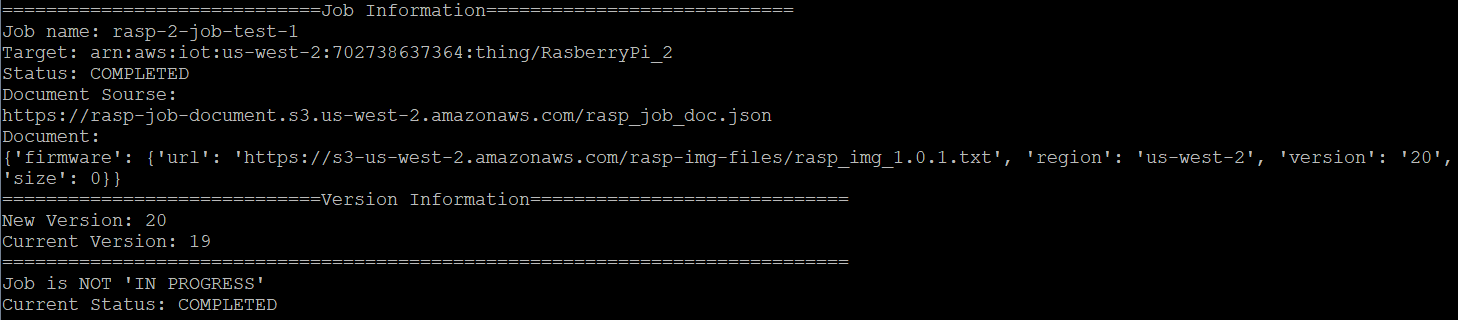
**Example 3: Job status is not ‘IN PROGRESS’**

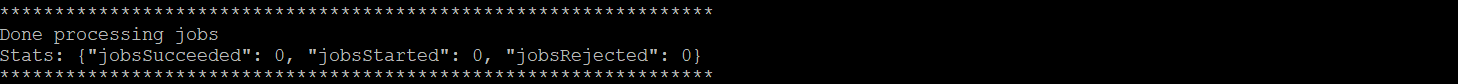
The job has been completed already.



Check the version but the job status is not ‘IN PROGRESS’.

The file is not downloaded.





**4-2-1. Create a job**

Create

**4-2-1. Create a job**

Create

**4-2-1. Create a job**

Create