This document describes the possible automation approach that can be applied.

It has been split into two parts, however, both of them could be placed under the crontab scheduler. For more on crontab please refer to (https://www.geeksforgeeks.org/crontab-in-linux-with-examples/)

Scenario 1 (in the Orange VM)

To upload a file that is present in the bucket of Minio to the Aidbox, consider the script https://github.com/CODA-19/csv-to-fhir/blob/master/upload_minio_aidbox.py, to explore more consider the approaches tried in jupyter https://github.com/CODA-19/csv-to-fhir/blob/master/upload_minio_orange.ipynb

To set authorization (this allows the Rest API to interact with the Aidbox), consider the description here https://www.youtube.com/watch?v=xWtNNi_Q-dU&t=3s&ab_channel=HealthSamurai

Set the Client id and Client secret values (in the example script here, the first is set as python_load_script, and the later as chum123)

Consider the script (https://github.com/CODA-19/csv-to-fhir/blob/master/upload_minio_aidbox.py)

Assuming there is a bucket (container) by the name chumtestbucket which contains a file culture_data.json.ndjson.gz. In order to create the corresponding URL that would point to the specific file, call the method create_presigned_url. Once the URL is generated create the payload using the line payload={"source":url}. Subsequently, access the Aidbox and perform the upload using the lines `requests.get()` and `requests.post()`

This completes the upload section using a python script

Scenario 2 (the different file generations)

To achieve the task of file generation of different types, for instance csv, then encryption, and then json in a sequential manner, the help of crontab and shell scripts is considered. A typical crontab command scheduled for execution on a specific day/time (https://www.geeksforgeeks.org/crontab-in-linux-with-examples/) is given below (based on the way it is installed at our end)

Where rxyz is the user name, generate_csv.sh, generate_encrypted.sh, and generate_json.sh are the shell scripts to generate csv, perform encryption, and the json files respectively.

15 19 * * SAT /usr/sbin/runuser -l rxyz -c '/usr/local/sbin/generate_csv.sh' && '/usr/local/sbin/generate_encrypted.sh' && '/usr/local/sbin/generate_json.sh'

Each shell script would contain the line that executes the corresponding python code.