

Assignment 1 – Event-driven CSV Cleaner (S3 → Lambda → S3)

****Goal:**** When a CSV lands in an S3 “incoming/” prefix, trigger a Lambda that validates & cleans rows and writes a cleaned file to “processed/”. Lambda must use an external library via a Lambda layer.

****Tasks:****

1. Create an S3 bucket with prefixes: `incoming/` and `processed/`.
2. Create an S3 event notification to trigger Lambda on upload.
3. Write a Lambda (Python 3.12) that reads the CSV, removes invalid emails, trims spaces, normalizes city names, and posts a summary using `requests` library (via Lambda Layer).
4. Save cleaned data in `processed/`.

****Dataset:**** customers.csv

****Layer creation:****

```

```
mkdir -p layer/python
pip install requests -t layer/python
cd layer && zip -r9 ..//requests-layer.zip .
```

**\*\*Deliverables:\*\***

- Lambda code, IAM policy, and CloudWatch screenshot.
- Evidence of processed output in S3.

# Assignment 2 – EC2 Log Cruncher with S3 Sync

**\*\*Goal:\*\*** Use Linux on EC2 to automate log parsing from S3, summarize them, and push metrics back to S3.

**\*\*Tasks:\*\***

1. Launch EC2 (Amazon Linux 2023) and attach an IAM role with S3 access.
2. `aws s3 sync s3://logs/ /var/log/web/`
3. Write Python to compute top IPs, 4xx/5xx counts, and most requested path.
4. Upload `summary.json` to S3 every 5 minutes using cron.

**\*\*Dataset:\*\*** web\_access.log

**\*\*Linux work:\*\***

- Setup AWS CLI, cron, and permissions.
- Test automation and view CloudWatch logs.

**\*\*Deliverables:\*\***

- summary.json, S3 screenshots, IAM policy, and crontab output.

## **Assignment 3 – PDF-to-Text Pipeline (S3 → Lambda with pypdf layer)**

**\*\*Goal:\*\*** Extract text from PDFs uploaded to S3 using Lambda and the `pypdf` library from a layer.

**\*\*Tasks:\*\***

1. Upload PDF to `s3://pdfs/` .
2. Lambda reads PDF using `pypdf.PdfReader` and saves extracted text to `pdfs-text/` .
3. Create a layer with pypdf.

**\*\*Layer creation:\*\***

```
```  
mkdir -p layer/python  
pip install pypdf -t layer/python  
cd layer && zip -r9 ./pypdf-layer.zip .  
```
```

**\*\*Dataset:\*\*** sample.pdf

**\*\*Deliverables:\*\***

- Lambda function ARN, layer ARN, CloudWatch log snippet, and output verification.

## **Assignment 4 – JSONL Compactor (EC2 producer → S3 → Lambda)**

**\*\*Goal:\*\*** EC2 script uploads JSONL chunks to S3; Lambda compacts them into a single daily file and writes a `report.json` .

**\*\*Tasks:\*\***

1. Use EC2 to generate and push JSONL batches to S3 `staging/` .
2. Lambda merges all JSONL files into one under `curated/` .
3. Create `report.json` with counts and file info.

**\*\*Dataset:\*\*** orders.jsonl

**\*\*Linux work:\*\***

- Create user `etl` , manage environment variables, automate uploads.

**\*\*Deliverables:\*\***

- EC2 & Lambda code, IAM policy, S3 screenshots, report.json.