

TASK - 3

Name : Jenin Krishna K P
Location : Chennai

File processing system :

The screenshot shows the Amazon S3 console interface. On the left, there's a sidebar with options like General purpose buckets, Storage Lens, and IAM Access Analyzer for S3. The main area is titled 'file-processing-bucket-jenin' and shows 'Objects (1)'. A table lists the single object: 'sample.txt' (txt type, last modified Nov 2, 2025, 18:09:18 UTC+05:30, 74.0 B, Standard storage class). At the bottom, there are links for CloudShell and Feedback, and a footer with copyright information and a timestamp of 4:16 PM on 11/6/2025.

The screenshot shows the Amazon DynamoDB console. On the left, there's a sidebar with options like Dashboard, Tables, DAX, and Events. The main area is titled 'FileProcessingResults' and shows the 'Settings' tab. A table lists three items: 'FileProcessingResults', 'Orders', and 'UserSubmissions'. The 'FileProcessingResults' item is selected. The 'General information' section provides details about the table, including its partition key ('fileName (String)'), capacity mode ('On-demand'), and item count (1). It also mentions that point-in-time recovery (PITR) is off. At the bottom, there are links for CloudShell and Feedback, and a footer with copyright information and a timestamp of 4:18 PM on 11/6/2025.

The screenshot shows the AWS Lambda console interface. At the top, there are tabs for 'sample.txt - Object in S3 bucket', 'View table | Amazon DynamoDB', and 'imageProcessorFunction | Function'. The main area is titled 'imageProcessorFunction' and shows a 'Function overview' section. It includes a 'Diagram' button, a 'Template' button, and a visual representation of the function's architecture. The diagram shows an 'imageProcessorFunction' box connected to an 'S3' box with a line labeled '+ Add destination'. Below the diagram are buttons for '+ Add trigger' and '+ Add destination'. To the right of the diagram, there is a 'Description' field (empty), a 'Last modified' field (3 days ago), and a 'Function ARN' field (arn:aws:lambda:ap-south-1:297186792789:function:imageProcessorFunction). A 'Function URL' field is also present. On the far right, there are 'Info' and 'Tutorials' tabs, with 'Tutorials' currently selected. A sidebar on the right provides a tutorial for creating a simple web app, with a 'Start tutorial' button. The bottom of the screen shows a toolbar with icons for CloudShell, Feedback, and various system status indicators.

This screenshot shows the AWS Lambda code editor for the 'imageProcessorFunction'. The left sidebar has an 'EXPLORER' section with a file tree showing 'lambda_function.py' under 'IMAGEPROCESSORFUNCTION'. Below it are sections for 'DEPLOY [UNDEPLOYED CHANGES]' (with a 'Deploy' button) and 'TEST EVENTS [NONE SELECTED]' (with a 'Create new test event' button). The main area is a code editor with the following Python code:

```
lambda_function.py
1 import json
2 import boto3
3 import os
4 s3 = boto3.client('s3')
5 def lambda_handler(event, context):
6     try:
7         record = event['Records'][0]
8         bucket_name = record['s3']['bucket']['name']
9         object_key = record['s3']['object']['key']
10        print(f"New image uploaded: {object_key} in bucket: {bucket_name}")
11        download_path = f"/tmp/{os.path.basename(object_key)}"
12        s3.download_file(bucket_name, object_key, download_path)
13        processed_key = object_key.replace("uploads/", "thumbnails/")
14        s3.upload_file(download_path, bucket_name, processed_key)
15        print(f"Processed image uploaded to: {processed_key}")
16        return {
17            'statusCode': 200,
18            'body': json.dumps(f"Processed image stored at {processed_")
19        }
20    except Exception as e:
21        print(f"Error: {str(e)}")
22        return {
```

The right side of the screen features a 'Tutorials' sidebar with a 'Create a simple web app' section, similar to the one in the first screenshot. The bottom of the screen shows a toolbar with icons for CloudShell, Feedback, and various system status indicators.

TextFileProcessor-role-8b4wn0eg

Summary

Creation date
November 02, 2025, 17:42 (UTC+05:30)

Last activity
5 days ago

ARN
arn:aws:iam::297186792789:role/service-role/TextFileProcessor-role-8b4wn0eg

Maximum session duration
1 hour

Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (2)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AWSLambdaBasicExecutionRole-aec07ea6-bd5d-4...	Customer managed	1
TextFileProcessorInlinePolicy	Customer inline	0

Permissions boundary (not set)

TextFileProcessor

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

RDS databases

Monitoring and operations tools

Concurrency and

Triggers (1)

S3: file-processing-bucket-jenin
arn:aws:s3:::file-processing-bucket-jenin

Tutorials

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial

The screenshot shows the AWS DynamoDB console with the 'FileProcessingResults' table selected. The left sidebar includes links for Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Under 'Explore items', there is a 'UserSubmissions' table. A message bar at the top right indicates 'Completed - Items returned: 1 - Items scanned: 1 - Efficiency: 100% - RCU consumed: 2'. The main area displays the table schema and a single item: sample.txt with charCount 68, lineCount 3, and a preview of the file content.

The screenshot shows the AWS CloudWatch Logs console for the 'TextFileProcessor' Lambda function. The left sidebar lists AI Operations, Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), Metrics, and Application Signals (APM). The 'Log groups' section is selected, showing the 'TextFileProcessor' group. A message bar at the top right indicates 'Completed - Items returned: 1 - Items scanned: 1 - Efficiency: 100% - RCU consumed: 2'. The main area displays log events for a specific execution, including:

```
END RequestId: cf7dc43a-4c1e-4ada-9b21-d1db7ecab23
REPORT RequestId: cf7dc43a-4c1e-4ada-9b21-d1db7ecab23 Duration: 132.50 ms Billed Duration: 133 ms Memory Size: 512 MB Max Memory Used: 167 MB
START RequestId: 06ca7246-8447-4c89-a73d-a3900d1b1837 Version: $LATEST
Triggered by s3://file-processing-bucket-jenin/sample.txt
Saved result to DynamoDB table: FileProcessingResults
END RequestId: 06ca7246-8447-4c89-a73d-a3900d1b1837
REPORT RequestId: 06ca7246-8447-4c89-a73d-a3900d1b1837 Duration: 921.39 ms Billed Duration: 922 ms Memory Size: 512 MB Max Memory Used: 171 MB
```

eclipse-workspace - text-file-processor/src/main/java/com/example/text_file_processor/TextFileProcessorHandler.java - Eclipse IDE

```
1 package com.example.text_file_processor;
2
3 import com.amazonaws.services.lambda.runtime.Context;
4
5 public class TextFileProcessorHandler implements RequestHandler<S3Event, String> {
6
7     private static final String TABLE_NAME = System.getenv("TABLE_NAME") != null
8         ? System.getenv("TABLE_NAME")
9         : "FileProcessingResults";
10
11     private final S3Client s3 = S3Client.create();
12     private final DynamoDbClientddb = DynamoDbClient.create();
13
14     @Override
15     public String handleRequest(S3Event event, Context context) {
16         LambdaLogger logger = context.getLogger();
17
18         var record = event.getRecords().get(0);
19         String bucket = record.getS3().getBucket().getName();
20         String key = record.getObject().getUnencodedKey();
21
22         logger.log("Triggered by S3 object: s3://" + bucket + "/" + key);
23
24         String content = readObjectAsString(bucket, key);
25         String normalized = content.replaceAll("\\s+", " ").trim();
26
27         int lineCount = countLines(content);
28         int wordCount = normalized.isEmpty() ? 0 : normalized.split("\\s+").length;
29         int charCount = content.replace("\r", "").replace("\n", "").length(); // chars excluding newlines
30
31         String preview = buildPreview(normalized, 100);
32
33         Map<String,AttributeValue> item = new HashMap<>();
34         item.put("fileName",AttributeValue.builder().s(key).build());
35         item.put("lineCount",AttributeValue.builder().n(Integer.toString(lineCount)).build());
36         item.put("wordCount",AttributeValue.builder().n(Integer.toString(wordCount)).build());
37         item.put("charCount",AttributeValue.builder().n(Integer.toString(charCount)).build());
38
39     }
40
41     private int countLines(String content) {
42         return content.lines().count();
43     }
44
45     private String readObjectAsString(String bucket, String key) {
46         String content = "";
47         try {
48             GetObjectRequest request =GetObjectRequest.builder().bucket(bucket).key(key).build();
49             ObjectInputStream objectInputStream = new ObjectInputStream(new ByteArrayInputStream(s3.getObject(request).getInputStream().readAllBytes()));
50             objectInputStream.readObject();
51             content = objectInputStream.readObject().toString();
52         } catch (IOException | ClassNotFoundException e) {
53             e.printStackTrace();
54         }
55         return content;
56     }
57
58     private String buildPreview(String content, int previewLength) {
59         return content.substring(0, previewLength);
60     }
61
62     private void putItemInDynamoDB(Map<String,AttributeValue> item) {
63         PutItemRequest request = PutItemRequest.builder().tableName(TABLE_NAME).item(item).build();
64        ddb.putItem(request);
65     }
66
67     private void updateTableStatus(String tableName) {
68         UpdateTableRequest request = UpdateTableRequest.builder().tableName(tableName).attributeDefinitions(AttributeDefinition.builder().attributeName("FileProcessingResults").attributeType("S").build()).build();
69         dynamoDbClient.updateTable(request);
70     }
71
72     private void updateTableStatus() {
73         updateTableStatus(TABLE_NAME);
74     }
75
76     private void updateTableStatus(String tableName) {
77         updateTableStatus(tableName);
78     }
79
80     private void updateTableStatus() {
81         updateTableStatus();
82     }
83
84     private void updateTableStatus() {
85         updateTableStatus();
86     }
87
88     private void updateTableStatus() {
89         updateTableStatus();
90     }
91
92     private void updateTableStatus() {
93         updateTableStatus();
94     }
95
96     private void updateTableStatus() {
97         updateTableStatus();
98     }
99
100    private void updateTableStatus() {
101        updateTableStatus();
102    }
103
104    private void updateTableStatus() {
105        updateTableStatus();
106    }
107
108    private void updateTableStatus() {
109        updateTableStatus();
110    }
111
112    private void updateTableStatus() {
113        updateTableStatus();
114    }
115
116    private void updateTableStatus() {
117        updateTableStatus();
118    }
119
120    private void updateTableStatus() {
121        updateTableStatus();
122    }
123
124    private void updateTableStatus() {
125        updateTableStatus();
126    }
127
128    private void updateTableStatus() {
129        updateTableStatus();
130    }
131
132    private void updateTableStatus() {
133        updateTableStatus();
134    }
135
136    private void updateTableStatus() {
137        updateTableStatus();
138    }
139
140    private void updateTableStatus() {
141        updateTableStatus();
142    }
143
144    private void updateTableStatus() {
145        updateTableStatus();
146    }
147
148    private void updateTableStatus() {
149        updateTableStatus();
150    }
151
152    private void updateTableStatus() {
153        updateTableStatus();
154    }
155
156    private void updateTableStatus() {
157        updateTableStatus();
158    }
159
160    private void updateTableStatus() {
161        updateTableStatus();
162    }
163
164    private void updateTableStatus() {
165        updateTableStatus();
166    }
167
168    private void updateTableStatus() {
169        updateTableStatus();
170    }
171
172    private void updateTableStatus() {
173        updateTableStatus();
174    }
175
176    private void updateTableStatus() {
177        updateTableStatus();
178    }
179
180    private void updateTableStatus() {
181        updateTableStatus();
182    }
183
184    private void updateTableStatus() {
185        updateTableStatus();
186    }
187
188    private void updateTableStatus() {
189        updateTableStatus();
190    }
191
192    private void updateTableStatus() {
193        updateTableStatus();
194    }
195
196    private void updateTableStatus() {
197        updateTableStatus();
198    }
199
200    private void updateTableStatus() {
201        updateTableStatus();
202    }
203
204    private void updateTableStatus() {
205        updateTableStatus();
206    }
207
208    private void updateTableStatus() {
209        updateTableStatus();
210    }
211
212    private void updateTableStatus() {
213        updateTableStatus();
214    }
215
216    private void updateTableStatus() {
217        updateTableStatus();
218    }
219
220    private void updateTableStatus() {
221        updateTableStatus();
222    }
223
224    private void updateTableStatus() {
225        updateTableStatus();
226    }
227
228    private void updateTableStatus() {
229        updateTableStatus();
230    }
231
232    private void updateTableStatus() {
233        updateTableStatus();
234    }
235
236    private void updateTableStatus() {
237        updateTableStatus();
238    }
239
240    private void updateTableStatus() {
241        updateTableStatus();
242    }
243
244    private void updateTableStatus() {
245        updateTableStatus();
246    }
247
248    private void updateTableStatus() {
249        updateTableStatus();
250    }
251
252    private void updateTableStatus() {
253        updateTableStatus();
254    }
255
256    private void updateTableStatus() {
257        updateTableStatus();
258    }
259
260    private void updateTableStatus() {
261        updateTableStatus();
262    }
263
264    private void updateTableStatus() {
265        updateTableStatus();
266    }
267
268    private void updateTableStatus() {
269        updateTableStatus();
270    }
271
272    private void updateTableStatus() {
273        updateTableStatus();
274    }
275
276    private void updateTableStatus() {
277        updateTableStatus();
278    }
279
280    private void updateTableStatus() {
281        updateTableStatus();
282    }
283
284    private void updateTableStatus() {
285        updateTableStatus();
286    }
287
288    private void updateTableStatus() {
289        updateTableStatus();
290    }
291
292    private void updateTableStatus() {
293        updateTableStatus();
294    }
295
296    private void updateTableStatus() {
297        updateTableStatus();
298    }
299
300    private void updateTableStatus() {
301        updateTableStatus();
302    }
303
304    private void updateTableStatus() {
305        updateTableStatus();
306    }
307
308    private void updateTableStatus() {
309        updateTableStatus();
310    }
311
312    private void updateTableStatus() {
313        updateTableStatus();
314    }
315
316    private void updateTableStatus() {
317        updateTableStatus();
318    }
319
320    private void updateTableStatus() {
321        updateTableStatus();
322    }
323
324    private void updateTableStatus() {
325        updateTableStatus();
326    }
327
328    private void updateTableStatus() {
329        updateTableStatus();
330    }
331
332    private void updateTableStatus() {
333        updateTableStatus();
334    }
335
336    private void updateTableStatus() {
337        updateTableStatus();
338    }
339
340    private void updateTableStatus() {
341        updateTableStatus();
342    }
343
344    private void updateTableStatus() {
345        updateTableStatus();
346    }
347
348    private void updateTableStatus() {
349        updateTableStatus();
350    }
351
352    private void updateTableStatus() {
353        updateTableStatus();
354    }
355
356    private void updateTableStatus() {
357        updateTableStatus();
358    }
359
360    private void updateTableStatus() {
361        updateTableStatus();
362    }
363
364    private void updateTableStatus() {
365        updateTableStatus();
366    }
367
368    private void updateTableStatus() {
369        updateTableStatus();
370    }
371
372    private void updateTableStatus() {
373        updateTableStatus();
374    }
375
376    private void updateTableStatus() {
377        updateTableStatus();
378    }
379
380    private void updateTableStatus() {
381        updateTableStatus();
382    }
383
384    private void updateTableStatus() {
385        updateTableStatus();
386    }
387
388    private void updateTableStatus() {
389        updateTableStatus();
390    }
391
392    private void updateTableStatus() {
393        updateTableStatus();
394    }
395
396    private void updateTableStatus() {
397        updateTableStatus();
398    }
399
400    private void updateTableStatus() {
401        updateTableStatus();
402    }
403
404    private void updateTableStatus() {
405        updateTableStatus();
406    }
407
408    private void updateTableStatus() {
409        updateTableStatus();
410    }
411
412    private void updateTableStatus() {
413        updateTableStatus();
414    }
415
416    private void updateTableStatus() {
417        updateTableStatus();
418    }
419
420    private void updateTableStatus() {
421        updateTableStatus();
422    }
423
424    private void updateTableStatus() {
425        updateTableStatus();
426    }
427
428    private void updateTableStatus() {
429        updateTableStatus();
430    }
431
432    private void updateTableStatus() {
433        updateTableStatus();
434    }
435
436    private void updateTableStatus() {
437        updateTableStatus();
438    }
439
440    private void updateTableStatus() {
441        updateTableStatus();
442    }
443
444    private void updateTableStatus() {
445        updateTableStatus();
446    }
447
448    private void updateTableStatus() {
449        updateTableStatus();
450    }
451
452    private void updateTableStatus() {
453        updateTableStatus();
454    }
455
456    private void updateTableStatus() {
457        updateTableStatus();
458    }
459
460    private void updateTableStatus() {
461        updateTableStatus();
462    }
463
464    private void updateTableStatus() {
465        updateTableStatus();
466    }
467
468    private void updateTableStatus() {
469        updateTableStatus();
470    }
471
472    private void updateTableStatus() {
473        updateTableStatus();
474    }
475
476    private void updateTableStatus() {
477        updateTableStatus();
478    }
479
480    private void updateTableStatus() {
481        updateTableStatus();
482    }
483
484    private void updateTableStatus() {
485        updateTableStatus();
486    }
487
488    private void updateTableStatus() {
489        updateTableStatus();
490    }
491
492    private void updateTableStatus() {
493        updateTableStatus();
494    }
495
496    private void updateTableStatus() {
497        updateTableStatus();
498    }
499
500    private void updateTableStatus() {
501        updateTableStatus();
502    }
503
504    private void updateTableStatus() {
505        updateTableStatus();
506    }
507
508    private void updateTableStatus() {
509        updateTableStatus();
510    }
511
512    private void updateTableStatus() {
513        updateTableStatus();
514    }
515
516    private void updateTableStatus() {
517        updateTableStatus();
518    }
519
520    private void updateTableStatus() {
521        updateTableStatus();
522    }
523
524    private void updateTableStatus() {
525        updateTableStatus();
526    }
527
528    private void updateTableStatus() {
529        updateTableStatus();
530    }
531
532    private void updateTableStatus() {
533        updateTableStatus();
534    }
535
536    private void updateTableStatus() {
537        updateTableStatus();
538    }
539
540    private void updateTableStatus() {
541        updateTableStatus();
542    }
543
544    private void updateTableStatus() {
545        updateTableStatus();
546    }
547
548    private void updateTableStatus() {
549        updateTableStatus();
550    }
551
552    private void updateTableStatus() {
553        updateTableStatus();
554    }
555
556    private void updateTableStatus() {
557        updateTableStatus();
558    }
559
560    private void updateTableStatus() {
561        updateTableStatus();
562    }
563
564    private void updateTableStatus() {
565        updateTableStatus();
566    }
567
568    private void updateTableStatus() {
569        updateTableStatus();
570    }
571
572    private void updateTableStatus() {
573        updateTableStatus();
574    }
575
576    private void updateTableStatus() {
577        updateTableStatus();
578    }
579
580    private void updateTableStatus() {
581        updateTableStatus();
582    }
583
584    private void updateTableStatus() {
585        updateTableStatus();
586    }
587
588    private void updateTableStatus() {
589        updateTableStatus();
590    }
591
592    private void updateTableStatus() {
593        updateTableStatus();
594    }
595
596    private void updateTableStatus() {
597        updateTableStatus();
598    }
599
599 private void updateTableStatus() {
600     updateTableStatus();
601 }
```

eclipse-workspace - text-file-processor/pom.xml - Eclipse IDE

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/xsd/maven-4.0.0.xsd>
<modelVersion>4.0.0</modelVersion>
<groupId>com.example</groupId>
<artifactId>text-file-processor</artifactId>
<version>1.0.0</version>
<properties>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <aws.sdk.version>2.26.19</aws.sdk.version>
</properties>
<dependencies>
    <!-- AWS SDK v2 -->
    <dependency>
        <groupId>software.amazon.awssdk</groupId>
        <artifactId>s3</artifactId>
        <version>${aws.sdk.version}</version>
    </dependency>
    <dependency>
        <groupId>software.amazon.awssdk</groupId>
        <artifactId>dynamodb</artifactId>
        <version>${aws.sdk.version}</version>
    </dependency>
    <!-- Lambda core & events -->
    <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-lambda-java-core</artifactId>
    </dependency>
</dependencies>
<dependencyManagement>
    <!-- AWS SDK v2 -->
    <dependencyManagement>
        <dependency>
            <groupId>software.amazon.awssdk</groupId>
            <artifactId>aws-lambda-java-core</artifactId>
            <version>${aws.sdk.version}</version>
        </dependency>
    </dependencyManagement>
</dependencyManagement>
<build>
    <plugins>
        <plugin>
            <groupId>com.amazonaws</groupId>
            <artifactId>aws-lambda-maven-plugin</artifactId>
            <version>1.4.0</version>
            <configuration>
                <functionName>TextFileProcessor</functionName>
                <roleArn>arn:aws:iam::123456789012:lambda-role</roleArn>
                <handler>com.example.TextFileProcessorHandler::handleRequest</handler>
                <environmentVariables>
                    <entry>TABLE_NAME,${TABLE_NAME}</entry>
                </environmentVariables>
            </configuration>
        </plugin>
    </plugins>
</build>

```