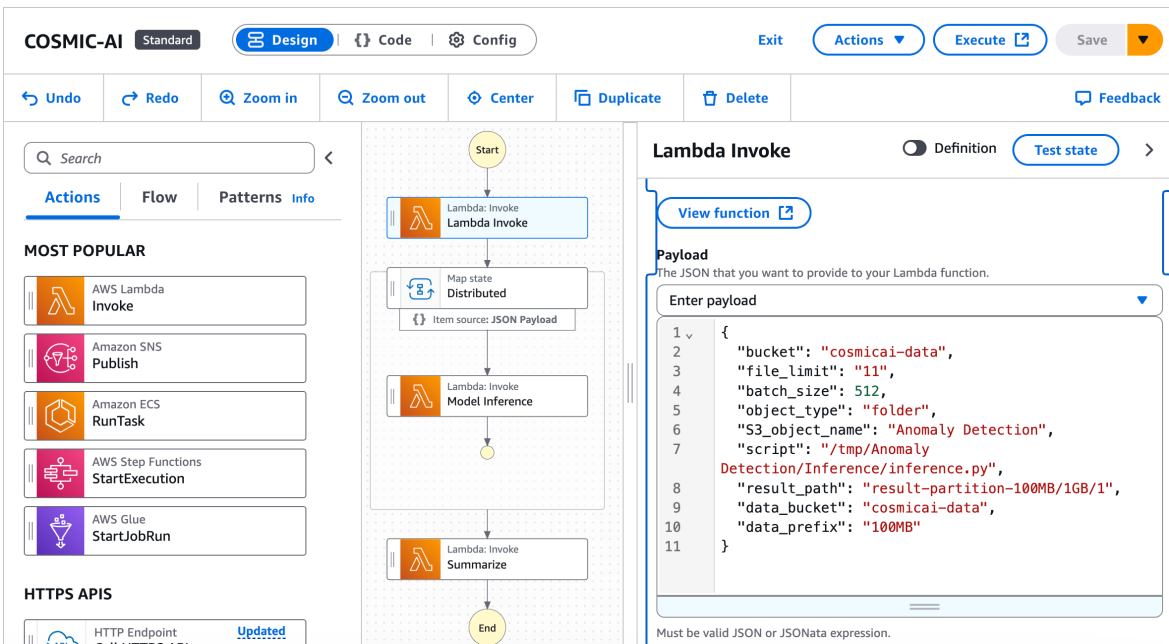


Project Step 1 Assignment: AWS Lambda Step Function

Sam Knisely, Darreion Bailey

1. Screenshots of your designed state machine in AWS Step Functions



2. Copy of IAM role configurations used for S3 access

team4-summer Info Delete

Allows Lambda functions to call AWS services on your behalf.

Summary Edit

Creation date
July 07, 2025, 19:48 (UTC-04:00)

Last activity
17 hours ago

ARN
arn:aws:iam::21112577852:role/team4-summer

Maximum session duration
1 hour

Permissions Trust relationships Tags Last Accessed Revoke sessions

Permissions policies (4) Info

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonS3FullAccess	AWS managed	14
AWSLambda_FullAccess	AWS managed	5
AWSLambdaBasicExecutionRole	AWS managed	3
CloudWatchActionsEC2Access	AWS managed	2

team4-summer
[Info](#)

Delete

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date
July 07, 2025, 19:48 (UTC-04:00)

Last activity
17 hours ago

ARN
am:aws:iam::211125778552:role/team4-summer

Maximum session duration
1 hour

Edit

Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions

Trusted entities

Edit trust policy

Entities that can assume this role under specified conditions.

```

1 - {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": [
8           "lambda.amazonaws.com",
9           "states.amazonaws.com"
10        ]
11      },
12      "Action": "sts:AssumeRole"
13    }
14  ]
15 }

```

3. Sample JSON payloads used for parameter passing

We tested 3 different JSON payloads with varying world sizes (11, 50, and 100).

```
{
  "bucket": "cosmicai-data",
  "file_limit": "11",
  "batch_size": 512,
  "object_type": "folder",
  "S3_object_name": "Anomaly Detection",
  "script": "/tmp/Anomaly Detection/Inference/inference.py",
  "result_path": "result-partition-100MB/1GB/1",
  "data_bucket": "cosmicai-data",
  "data_prefix": "100MB"
}
```

```
{
  "bucket": "cosmicai-data",
  "file_limit": "50",
  "batch_size": 512,
  "object_type": "folder",
  "S3_object_name": "Anomaly Detection",
  "script": "/tmp/Anomaly Detection/Inference/inference.py",
  "result_path": "result-partition-100MB/1GB/2",
  "data_bucket": "cosmicai-data",
}
```

```
"data_prefix": "100MB"}
```

```
{"bucket": "cosmicai-data",  
 "file_limit": "100",  
 "batch_size": 512,  
 "object_type": "folder",  
 "S3_object_name": "Anomaly Detection",  
 "script": "/tmp/Anomaly Detection/Inference/inference.py",  
 "result_path": "result-partition-100MB/1GB/3",  
 "data_bucket": "cosmicai-data",  
 "data_prefix": "100MB"}
```

4. Screenshots of execution results showing successful workflow completion

World size = 11

> COSMIC-AI > Execution: 78f1e470-562b-49ff-aa17-96b7b8b67469

Execution status
Succeeded

Execution type
Standard

Execution ARN
[arn:aws:states:us-east-1:211125778552:execution:cosmic-ai:78f1e470-562b-49ff-aa17-96b7b8b67469](#)

IAM role ARN
[arn:aws:iam::211125778552:role/team4-summer](#)

State transitions [Learn more](#)
16

Start time
Jul 8, 2025, 01:52:30.026 (UTC-04:00)

End time
Jul 8, 2025, 01:52:40.319 (UTC-04:00)

Duration
00:00:10.293

Alias
-

Version
-

Graph view | Table view

Graph view

Start

AWS Lambda: Invoke
Lambda Invoke

Iteration #10
Map state
Distributed
Item source: JSON Payload

AWS Lambda: Invoke
Model Inference

AWS Lambda: Invoke
Summarize

End

Lambda Invoke

Logs | Lambda | Log group

```
127 "DATA_PREFIX": "100MB",  
128 "DATA_PATH": "100MB/107.pt",  
129 "RESULT_PATH": "result-partition-100MB/1GB/1",  
130 "BATCH_SIZE": 512  
131 },  
132 {  
133   "S3_BUCKET": "cosmicai-data",  
134   "S3_OBJECT_NAME": "Anomaly Detection",  
135   "SCRIPT": "/tmp/Anomaly Detection/Inference/inference.py",  
136   "S3_OBJECT_TYPE": "folder",  
137   "WORLD_SIZE": "11",  
138   "RANK": "10",  
139   "DATA_BUCKET": "cosmicai-data",  
140   "DATA_PREFIX": "100MB",  
141   "DATA_PATH": "100MB/108.pt",  
142   "RESULT_PATH": "result-partition-100MB/1GB/1",  
143   "BATCH_SIZE": 512  
144 }  
145 ]
```

World size = 50

COSMIC-AI > Execution: d9e788ca-7208-47e2-a68c-a3cc17b96aac

Execution status
Succeeded

Execution type
Standard

Execution ARN
arn:aws:states:us-east-1:211125778552:execution:COSMIC-AI:d9e788ca-7208-47e2-a68c-a3cc17b96aac

IAM role ARN
arn:aws:iam::211125778552:role/team4-summer

State transitions
55

Start time
Jul 8, 2025, 19:51:56.374 (UTC-04:00)

End time
Jul 8, 2025, 19:52:00.668 (UTC-04:00)

Duration
00:00:04.294

Alias
-

Version
-

Graph view | Table view

Graph view

Start

AWS Lambda Invoke
Lambda Invoke

Iteration #49
Map state
Distributed
Item source: JSON Payload

AWS Lambda Invoke
Model Inference

AWS Lambda Invoke
Summarize

End

Actions

Lambda Invoke

Logs | Lambda | Log group

Test state

634

"DATA_PREFIX": "100MB",

635

"DATA_PATH": "100MB/25.pt",

636

"RESULT_PATH": "result-partition-100MB/1GB/2",

637

"BATCH_SIZE": 512

638

},

639

{

640

"S3_BUCKET": "cosmicai-data",

641

"S3_OBJECT_NAME": "Anomaly Detection",

642

"SCRIPT": "/tmp/Anomaly Detection/Inference/inference.py",

643

"S3_OBJECT_TYPE": "folder",

644

"WORLD_SIZE": "50",

645

"RANK": "49",

646

"DATA_BUCKET": "cosmicai-data",

647

"DATA_PREFIX": "100MB",

648

"DATA_PATH": "100MB/26.pt",

649

"RESULT_PATH": "result-partition-100MB/1GB/2",

650

"BATCH_SIZE": 512

651

}

652

}

World size = 100

COSMIC-AI
> Execution: f2cfc470-2116-44a9-b3e4-c8c4d4e748a2

Details
Execution input and output
Definition

Execution status
Succeeded

Execution type
Standard

Execution ARN
arn:aws:states:us-east-1:211125778552:execution:COSMIC-AI:f2cfc470-2116-44a9-b3e4-c8c4d4e748a2

IAM role ARN
arn:aws:iam::211125778552:role/team4-summer

State transitions [Learn more](#)
105

Start time
Jul 8, 2025, 19:50:33.853 (UTC-04:00)

End time
Jul 8, 2025, 19:50:43.444 (UTC-04:00)

Duration
00:00:09.591

Alias
-

Version
-

Graph view
Table view

Graph view

Start

AWS Lambda: Invoke
Lambdas Invoke

Iteration #99
Map state
Distributed

Item source: JSON Payload

AWS Lambda: Invoke
Model Inference

AWS Lambda: Invoke
Summarize

Lambdas Invoke

Logs | Lambdas | Log group

```

1285 "URI_PATH": "100MB/0.pt",
1286 "RESULT_PATH": "result-partition-100MB/1GB/2",
1287 "BATCH_SIZE": 512
1288 },
1289 {
1290   "S3_BUCKET": "cosmicai-data",
1291   "S3_OBJECT_NAME": "Anomaly Detection",
1292   "SCRIPT": "/tmp/Anomaly
Detection/Inference/inference.py",
1293   "S3_OBJECT_TYPE": "folder",
1294   "WORLD_SIZE": "100",
1295   "RANK": "99",
1296   "DATA_BUCKET": "cosmicai-data",
1297   "DATA_PREFIX": "100MB",
1298   "DATA_PATH": "100MB/71.pt",
1299   "RESULT_PATH": "result-partition-100MB/1GB/2",
1300   "BATCH_SIZE": 512
1301 }

```

5. Performance measurement table showing memory usage, duration, and cost metrics

World Size	Partition	Requests	Duration (s)	Memory	Cost (\$)
100	100MB	10018	2.64	1.4	0.62
50	100MB	5008	2.86	1.8	0.43
11	100MB	3327	1.87	1.9	0.2

6. Brief explanation (1-2 paragraphs) of your implementation approach and any challenges encountered

We implemented an AWS step function and tested the various JSON payloads mentioned above. This involved testing different world sizes (11, 50, and 100) of the 100 MB partition sized data. The state machine starts with an initialization using a lambda function, then a distributed map is created using a lambda container and a script is run to fetch the environment variables, JSON payload, and run inference and save JSON outputs for each rank. Lastly, a final lambda function is used to combine the results from the previous inference JSON and save a final output.

Our team created our own state machine before the class walkthrough on July 8th. We followed the steps on the GitHub and created our lambda functions using the python codes provided there. We also created our own IAM role with the necessary permissions. We called out state machine “COSMIC-AI”, our IAM role “team4-summer”, and our lambda functions “data-parallel-init2”, “inference”, and “summarize”. We utilized the comsmicai-data S3 bucket that was already created. We ran into some challenges with the IAM role creation because we gave it the necessary permissions but also had to go in and update the trusted relationships. Overall, it took us some time but we were able to figure out how to set up the state machine ourselves.