Serverless Summit.
by globaldatanet

AWS Step Functions: Steep Curve; Maximum Power

Matt Morgan
AWS Community Builder
https://mattmorgan.cloud

globaldatanet ALICE & BOB. COMPANY aws Serverless Guru sedai kreuzwerker

Serverless Summit.

Serverless Devotee

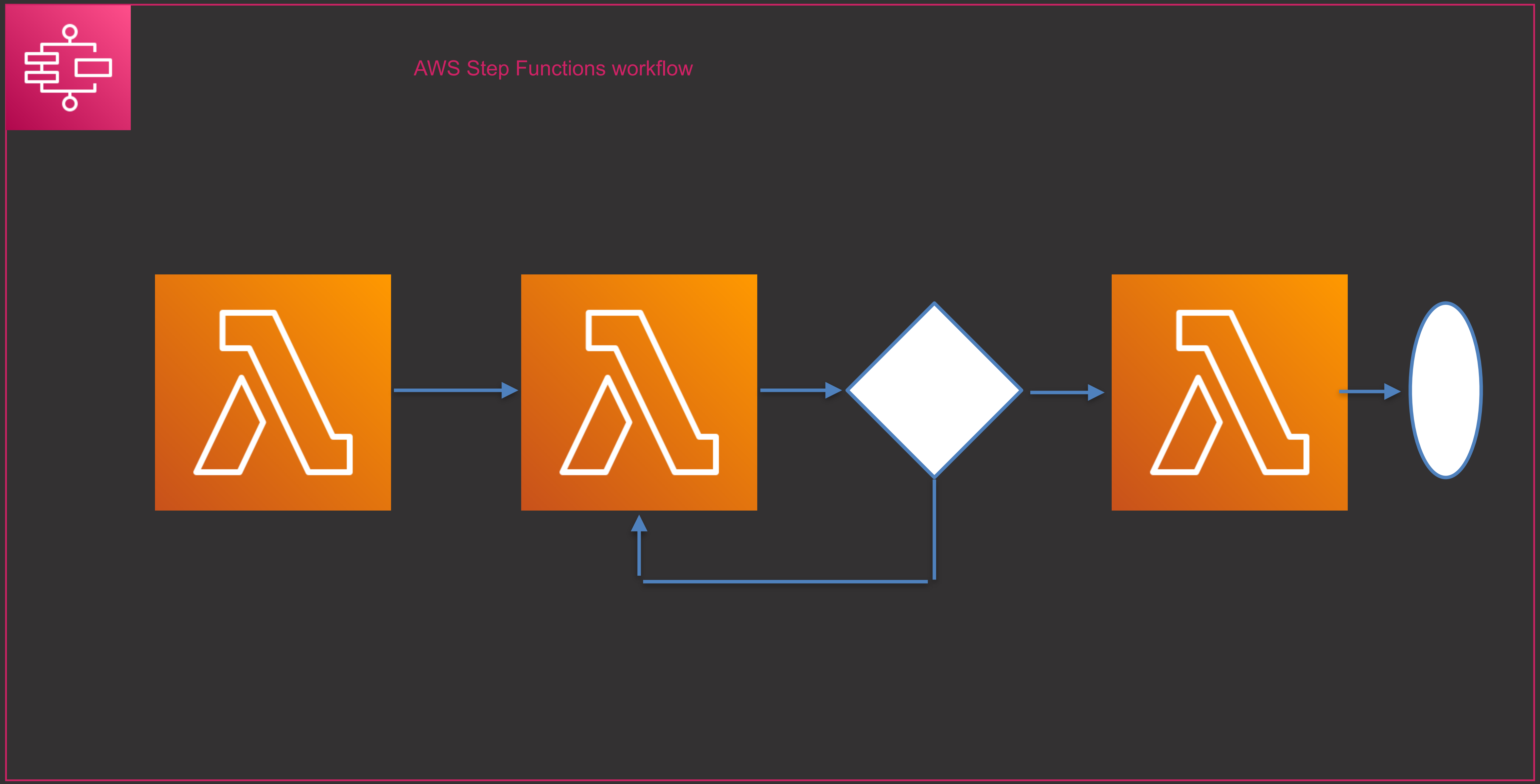
Serverless Summit.

Serverless Devotee

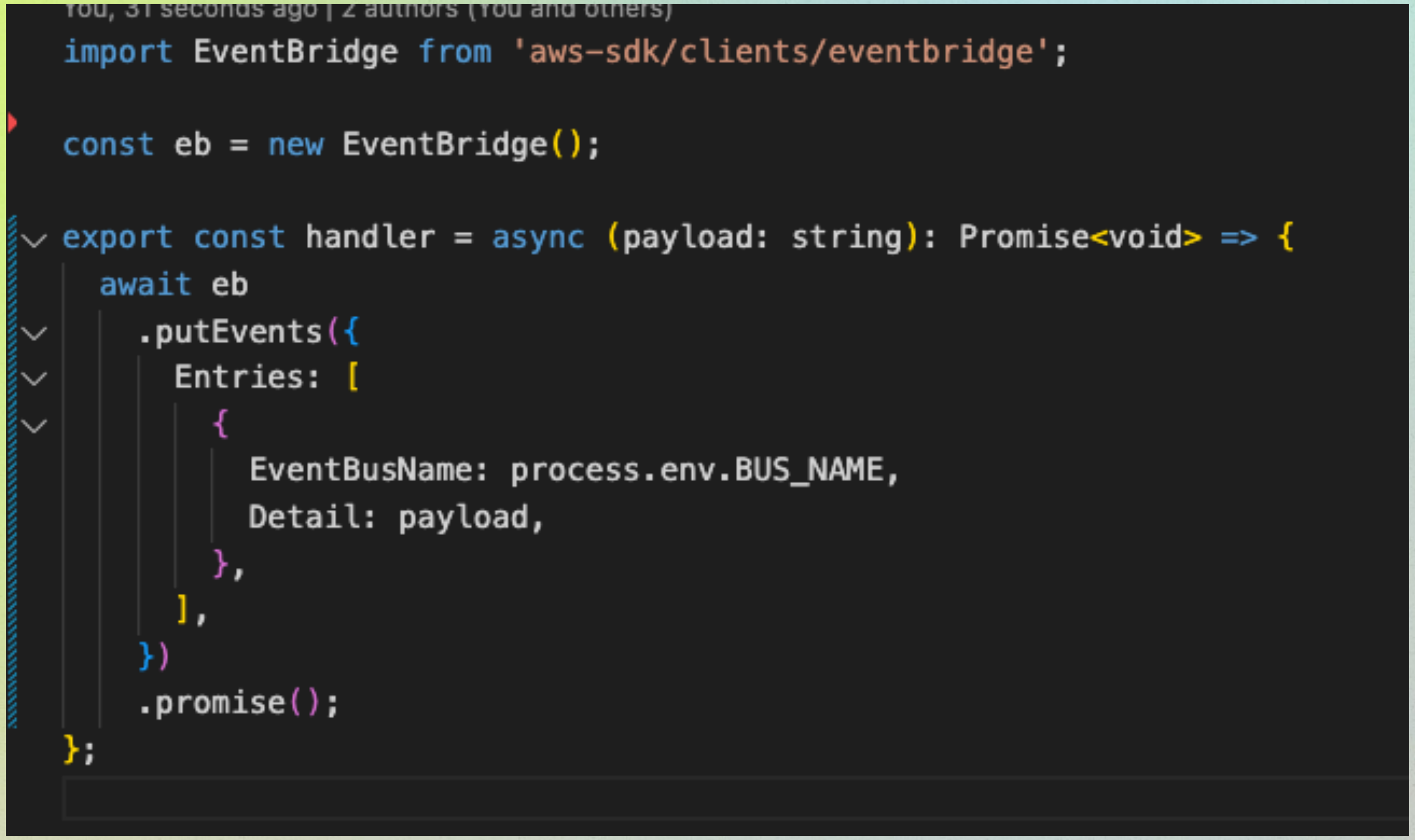
Why Step Functions?

- Asynchronous background processing
- Own a complex workflow end-to-end
- Spin up instantly and consume no resources while idle
- Pause/start/stop workflows
- For me, this is use case #1 for serverless!

Step Functions as a Lambda Orchestrator



Lambda as SDK Proxy?



```
You, 31 seconds ago | 2 authors (You and others)
import EventBridge from 'aws-sdk/clients/eventbridge';

const eb = new EventBridge();

export const handler = async (payload: string): Promise<void> => {
  await eb
    .putEvents({
      Entries: [
        {
          EventBusName: process.env.BUS_NAME,
          Detail: payload,
        },
      ],
    })
    .promise();
};
```

- Cold Starts
- Code Ownership
- Dependencies
- Runtime Deprecation

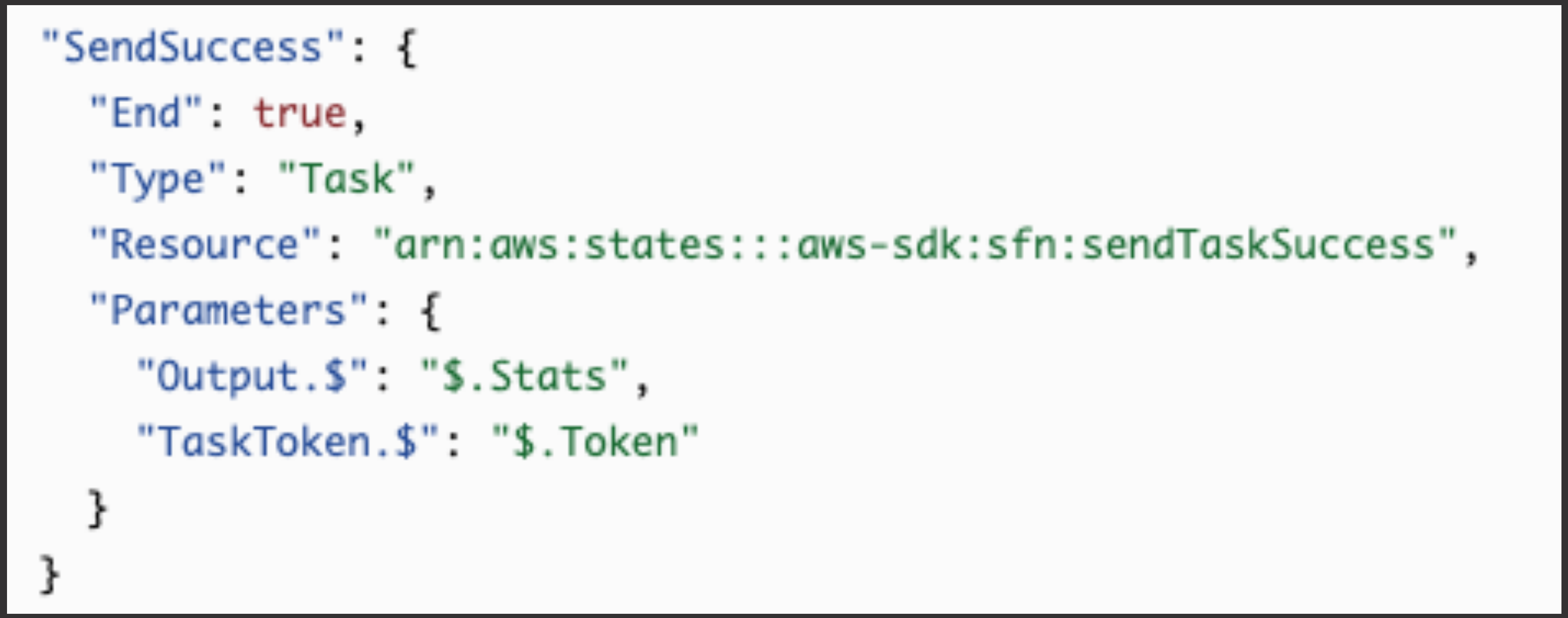
Service Integrations and Intrinsics are Powerful!

- Direct SDK integrations
- Array manipulation
- String operations
- Math
- Hash/random strings
- Encoding/Decoding
- JSON

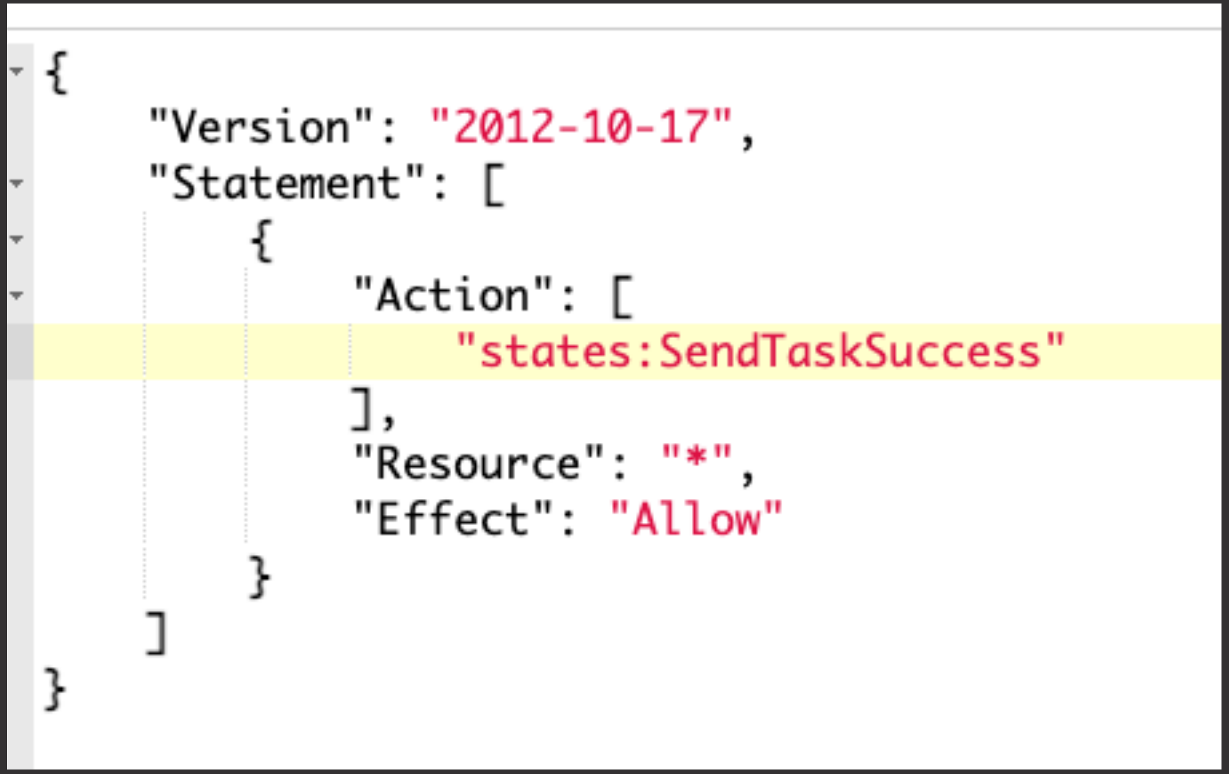
Service Integrations and Intrinsics are Powerful!

- Fast
- Cheap
- No Dependencies
- Managed Compute (no OOM)
- Observable

Service Integrations and Intrinsics are Hard!

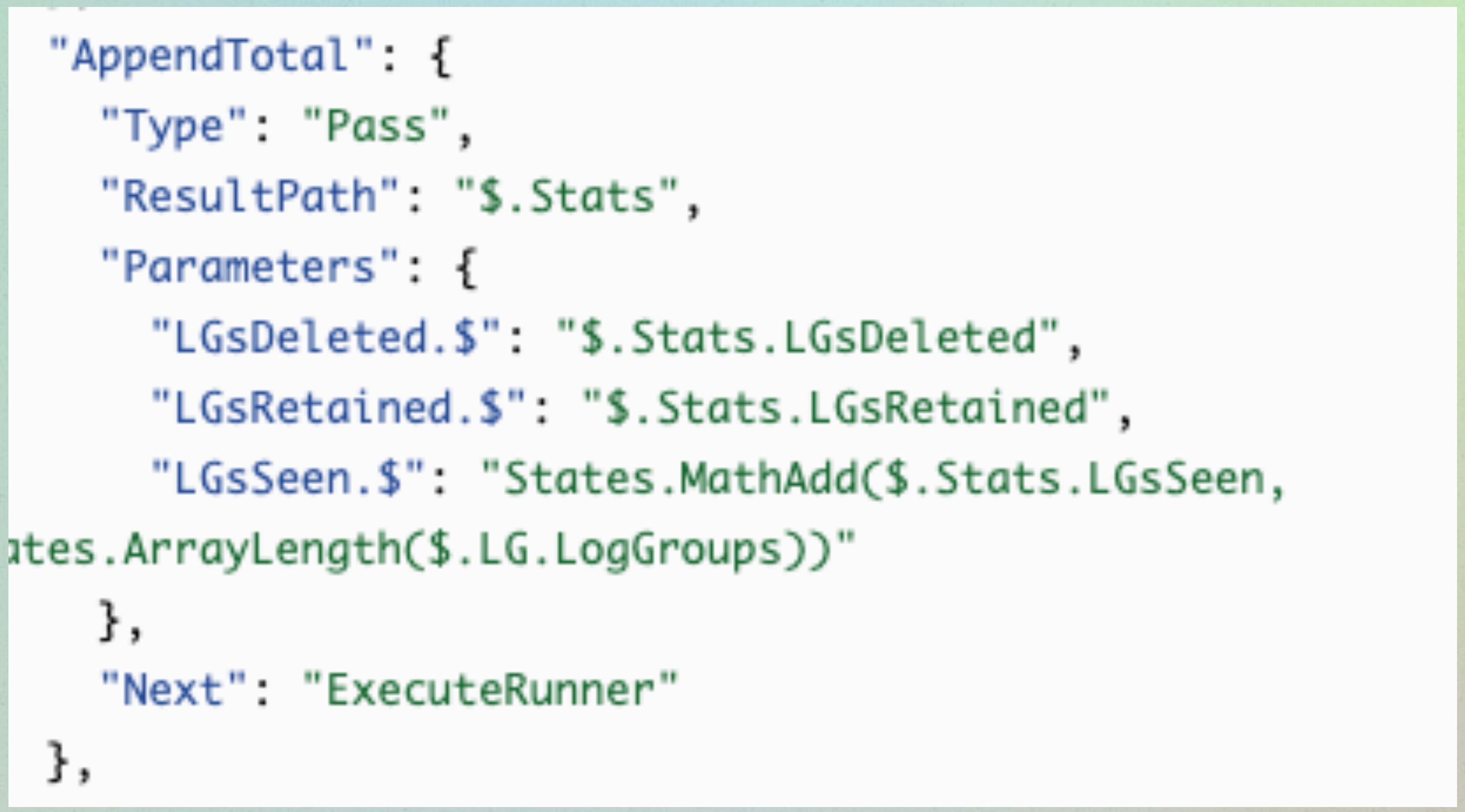


```
"SendSuccess": {
  "End": true,
  "Type": "Task",
  "Resource": "arn:aws:states:::aws-sdk:sfn:sendTaskSuccess",
  "Parameters": {
    "Output.$": "$.Stats",
    "TaskToken.$": "$.Token"
  }
}
```

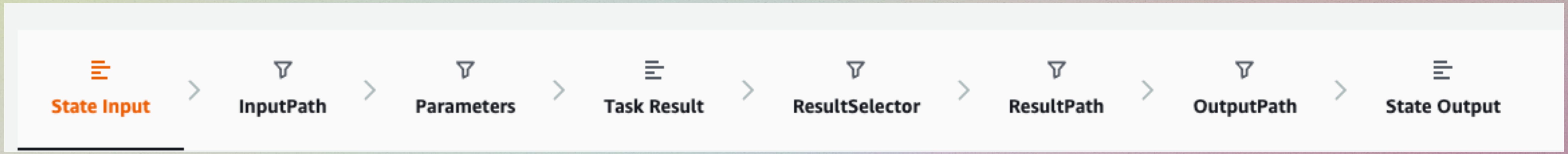


```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "states:SendTaskSuccess"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Service Integrations and Intrinsics are Hard!



```
 "AppendTotal": {
   "Type": "Pass",
   "ResultPath": "$.Stats",
   "Parameters": {
     "LGsDeleted.$": "$.Stats.LGsDeleted",
     "LGsRetained.$": "$.Stats.LGsRetained",
     "LGsSeen.$": "States.MathAdd($.Stats.LGsSeen,
ates.ArrayLength($.LG.LogGroups))"
   },
   "Next": "ExecuteRunner"
 },
```

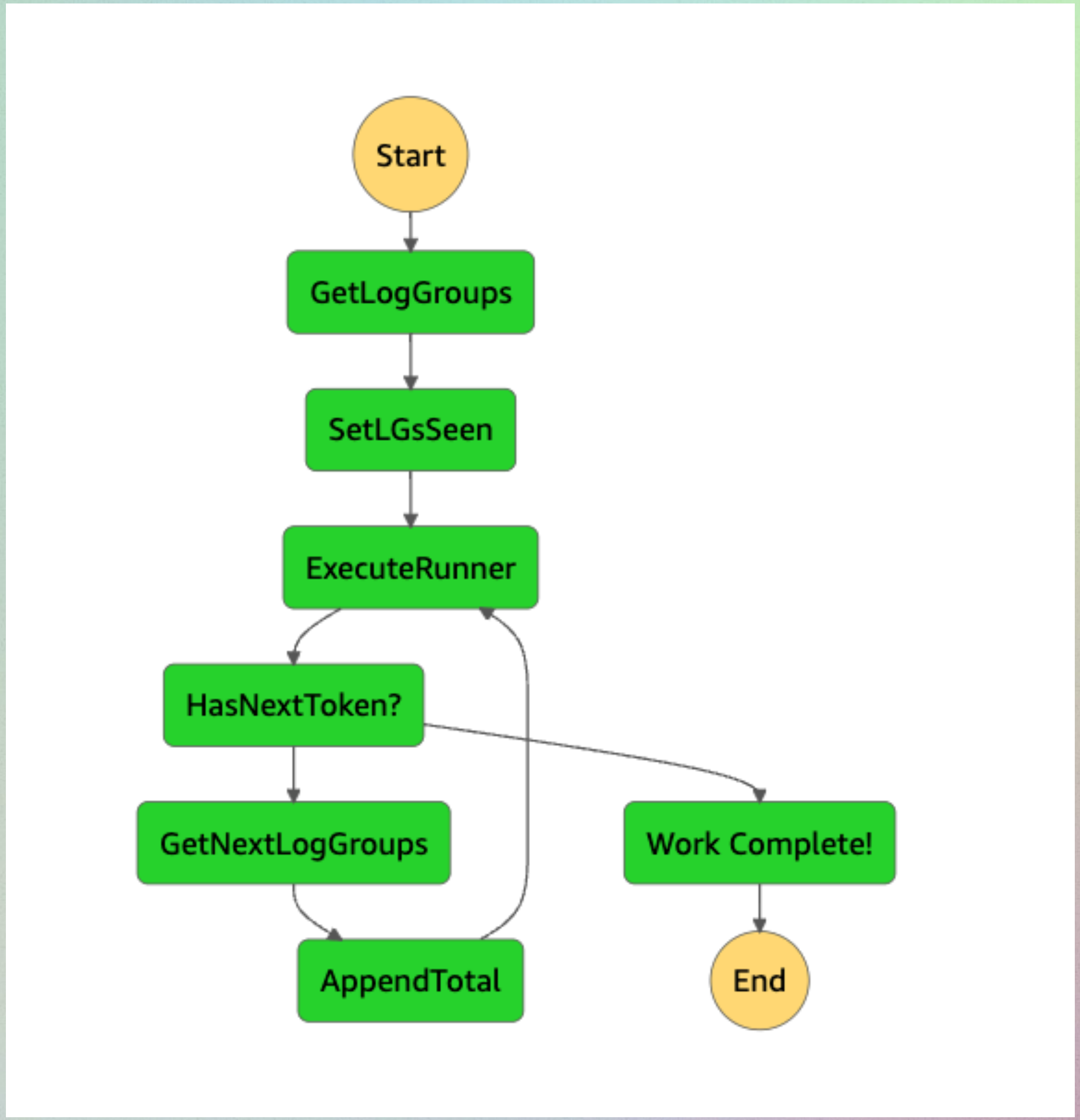


Data Flow Simulator

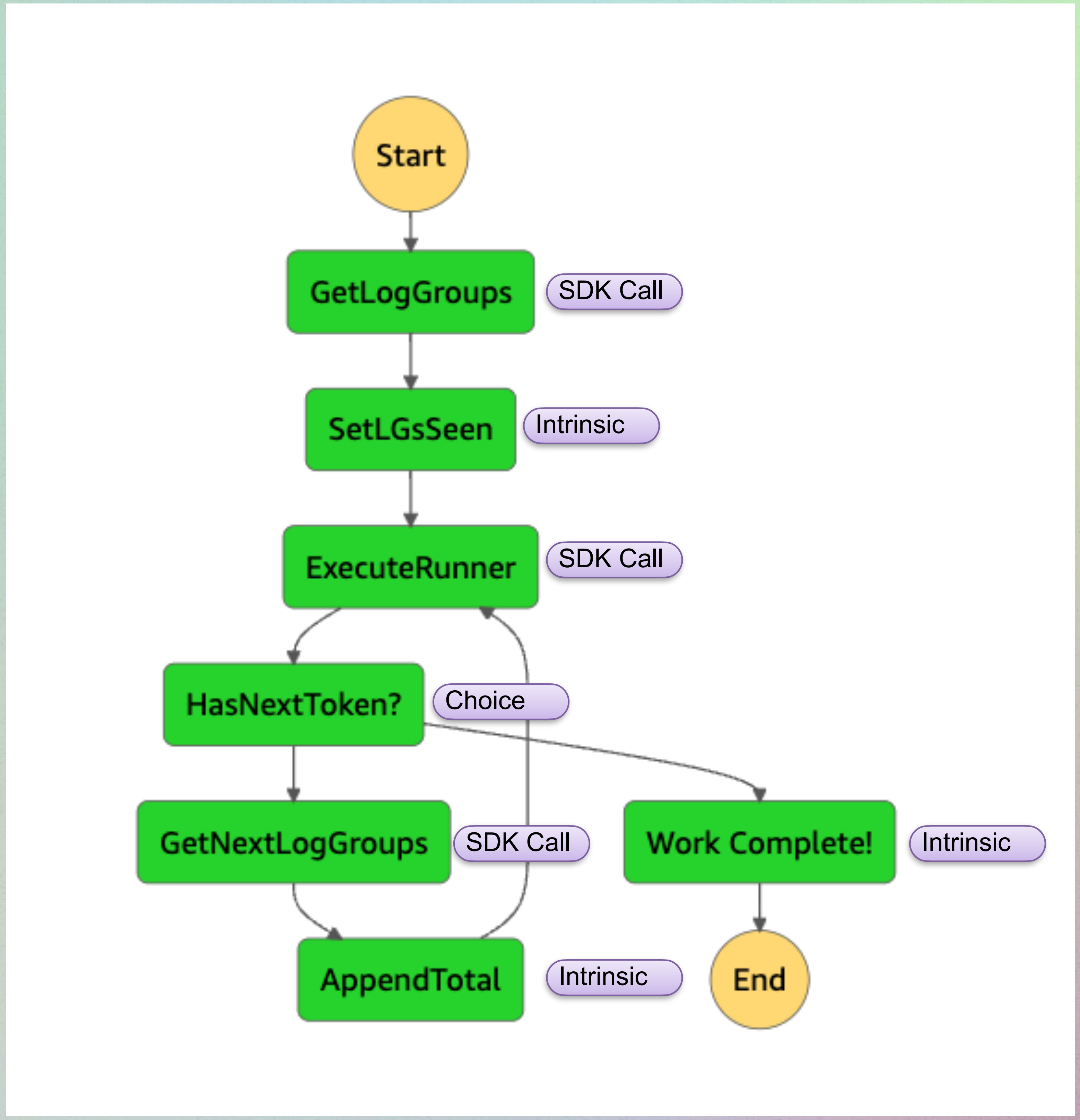
AWS Logs Comptroller

- Set LogGroup Retention if unset
- Prune “orphaned” Lambda LogGroups
- Can be scheduled
- Any scale
- Log results

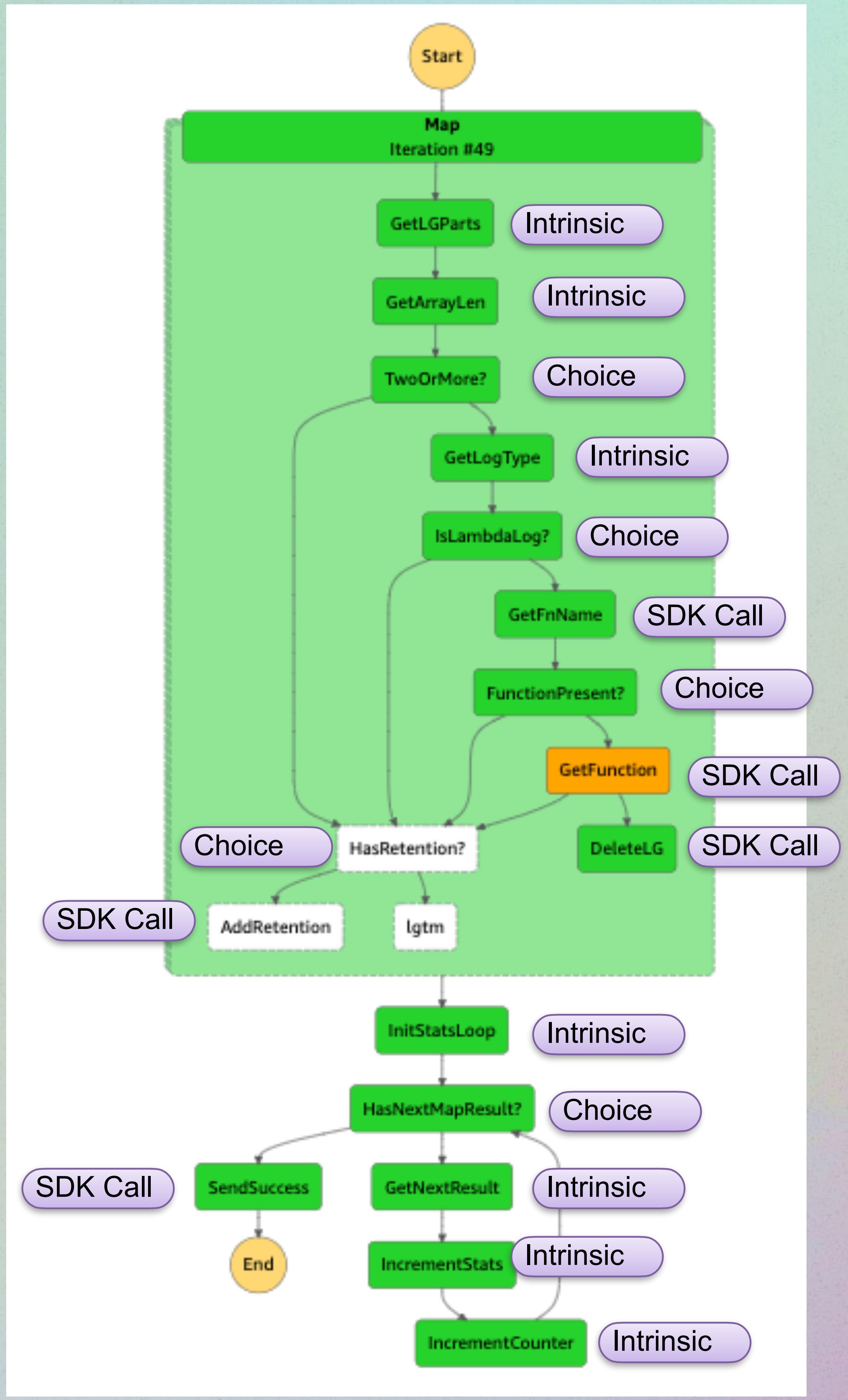
AWS Logs Comptroller



AWS Logs Comptroller



AWS Logs Comptroller



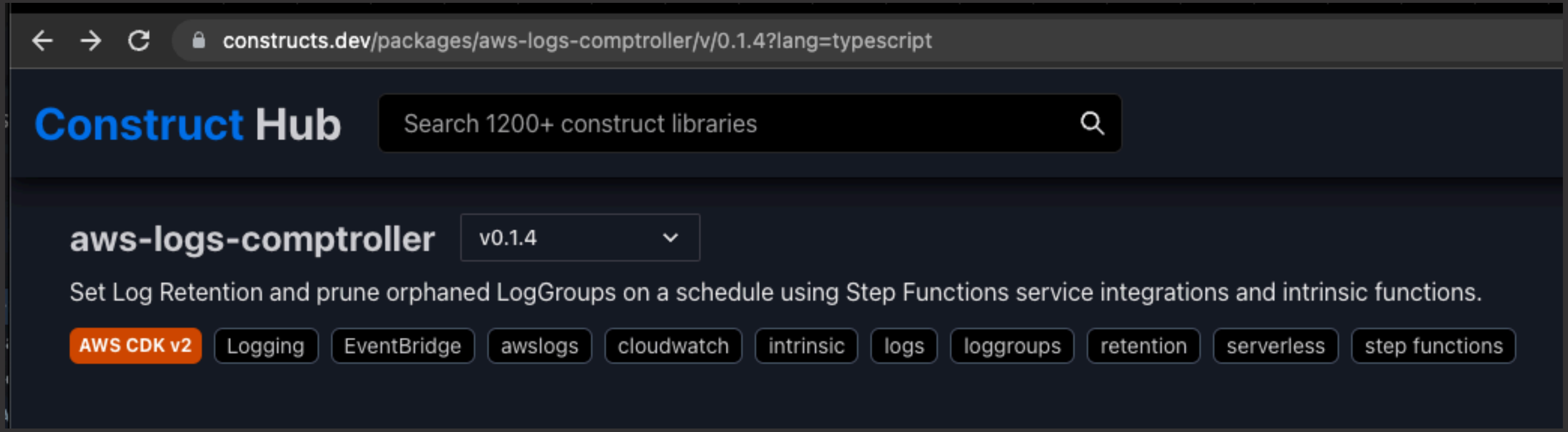
The Sweet Spot

- Highly optimized and scaled-out workflows
- Distributed constructs
- Curb any spaghetti tendencies

Tools

- Workflow Studio
- cdk watch / sam sync
- functionless.org

Try My Construct!



constructs.dev/packages/aws-logs-comptroller/v/0.1.4?lang=typescript

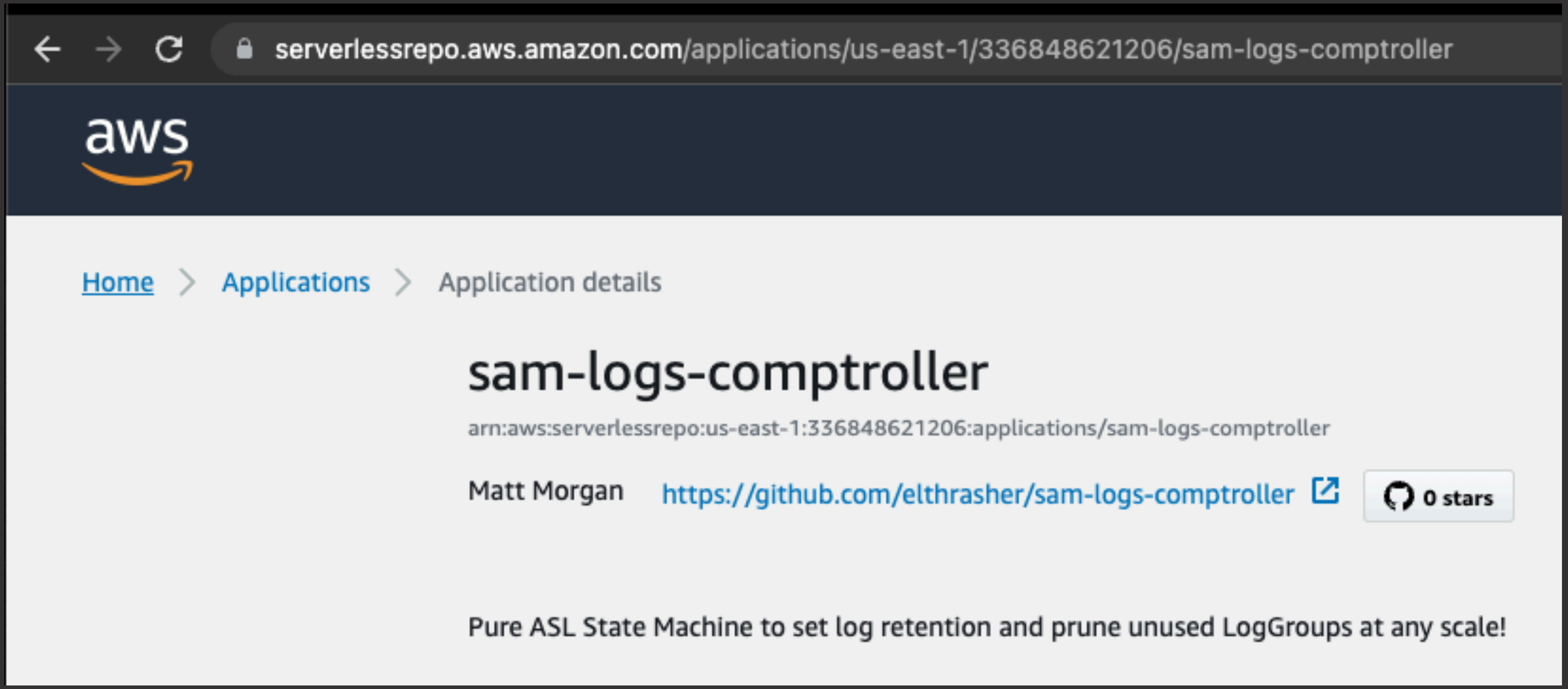
Construct Hub

Search 1200+ construct libraries

aws-logs-comptroller v0.1.4

Set Log Retention and prune orphaned LogGroups on a schedule using Step Functions service integrations and intrinsic functions.

AWS CDK v2 | Logging | EventBridge | awslogs | cloudwatch | intrinsic | logs | loggroups | retention | serverless | step functions



serverlessrepo.aws.amazon.com/applications/us-east-1/336848621206/sam-logs-comptroller

Home > Applications > Application details

sam-logs-comptroller

arn:aws:serverlessrepo:us-east-1:336848621206:applications/sam-logs-comptroller

Matt Morgan https://github.com/elthrasher/sam-logs-comptroller 0 stars

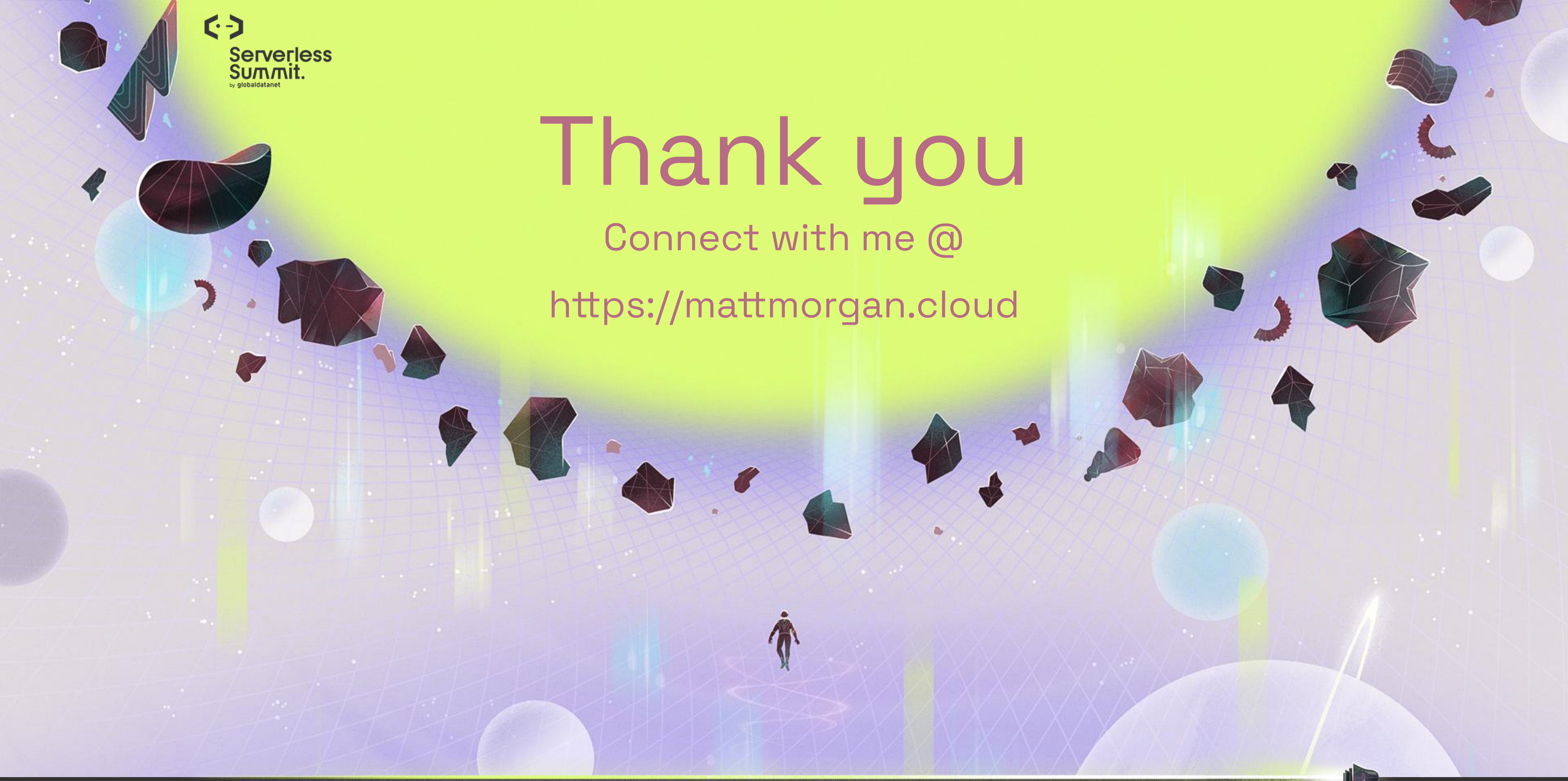
Pure ASL State Machine to set log retention and prune unused LogGroups at any scale!

Serverless Summit.
by globaldatanet

Thank you

Connect with me @

https://mattmorgan.cloud



globaldatanet ALICE & BOB. COMPANY aws Serverless Guru sedai kreuzwerker