Building And Automating Serverless Auto-Scaling Data Pipelines In AWS
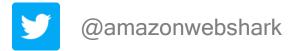
Damien Jones (he/him)

AWS Consultant @ Steamhaus

2024-10-08 AWS User Group Leeds

MrDamienJones

MrDamienJones

amazonwebshark.com

damien@amazonwebshark.com

@amazonwebshark

@amazonwebshark

https://www.flaticon.com/

Agenda

Problem Definition

Solution Architecture

Demos

Summary & Questions

github.com/MrDamienJones
/Community-Sessions

Damien Jones

♂ He/Him 🌍 Manchester UK 🦈 Fin Fan

Consultant @ Steamhaus

Using AWS since 2019

Creator @ amazonwebshark.com

Runner; Keen Gardener; Dog Dad

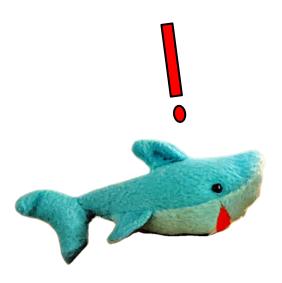# The 4 Vs Of Big Data

Characteristics of Big Data…

…and events…

…and API requests…

…metrics …traces …logs …

Variety

"The state of being diverse or varied."

Variety

"The state of being diverse or varied."

Structure
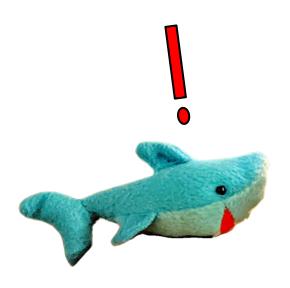
Intent

Sensitivity

Velocity

"The speed at which something is moving in a given direction."

Velocity

"The speed at which something is moving in a given direction."

Streaming or Batch
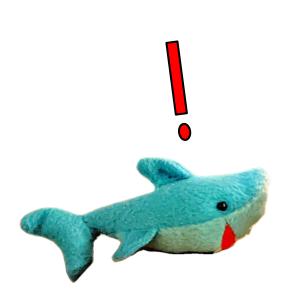
Synchronous or Asynchronous

Scheduling

Veracity

"The quality of being true or the habit of telling the truth."

Veracity

"The quality of being true or the habit of telling the truth."

External Security
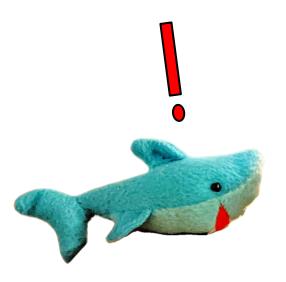
Validation & Health
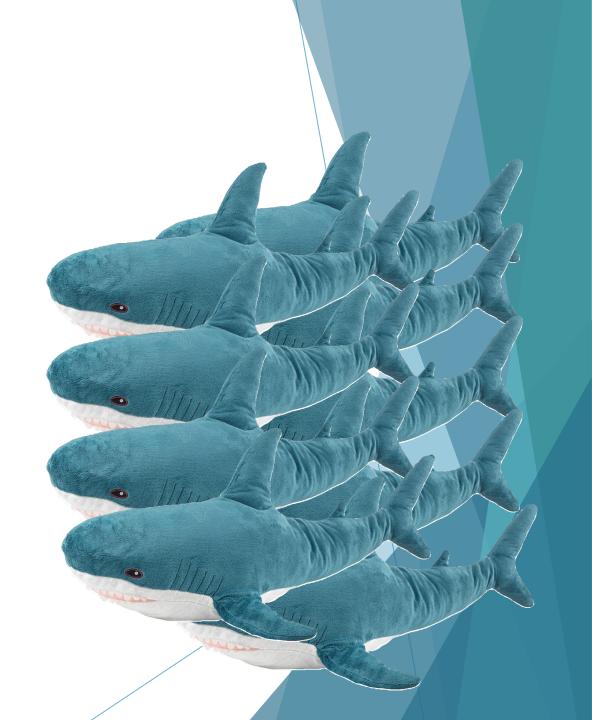
Internal Security

Volume

"The amount of space occupied."

Volume

"The amount of space occupied."

Size or Amount

Access Patterns

Backups

Building And Automating Serverless Auto-Scaling Data Pipelines In AWS

Building And Automating Serverless Auto-Scaling Data Pipelines In AWS

AWS Lambda

Serverless compute service

Supports multiple languages

Auto-scales on demand

Up to 1000 concurrent executions

Amazon S3

Serverless object storage

Store anything for any reason

1000s of requests per second

Object protection & integrity checks

Building And Automating Serverless Auto-Scaling Data Pipelines In AWS

Building And Automating Serverless Auto-Scaling Data Pipelines In AWS

Amazon Athena

AWS Glue

Amazon Athena

Serverless interactive query service

Query & access controls

Create derived tables

Read-Only & Open Table support

AWS Glue

Fully managed serverless ETL service

Crawlers discover data automatically

Up to 2000 concurrent ETL job runs

Data Catalog indexes data assets

Building And ==Automating== Serverless Auto-Scaling Data Pipelines In AWS

Building And Automating Serverless Auto-Scaling Data Pipelines In AWS

AWS Step Functions

Amazon EventBridge Scheduler

AWS Step Functions

Serverless task orchestration

Invoke over 220 AWS services / 10k API calls

Design workflows visually and as code

Standard & Express workflows

Design {} Code Config

Exit Actions ▼ Execute ⧉ Save ▼

↩ Undo ↪ Redo Zoom in Zoom out Center Duplicate Delete Feedback

Q Search ‹

Actions Flow Patterns Info

MOST POPULAR

AWS Lambda
Invoke

Amazon SNS
Publish

Amazon ECS
RunTask

AWS Step Functions
StartExecution

AWS Glue
StartJobRun

THIRD-PARTY API

HTTP Endpoint
Call third-party API

Start

Lambda: Invoke
data_wordpressapi_raw

Lambda: Invoke
data_wordpressapi_bronze

Glue: StartCrawler
Start Bronze Crawler

Catch #1

Glue: StartJobRun
Glue Start Silver Python Shell

Catch #1

Parallel state
Parallel

RulesetEvaluationRun
ics_pages

Glue: StartDataQualityRulesetEvaluationRun
DQ Run: silver-posts

Glue: StartCrawler
Start Silver Crawler

Catch #1

Catch #1

SNS: Publish
PublishFailure

End

Workflow

○ Definition ›

The top level Amazon States Language properties for this workflow. Learn more ⧉

Start at

The state that is the starting point of the workflow.

data_wordpressapi_raw ▼

Comment - *optional*

A human-readable description of the state machine.

Runs Lambda functions to ingest WordPress API data and transform to Parquet.

TimeoutSeconds - *optional*

The maximum number of seconds an execution of the state machine can run. If it runs longer than the specified time, the execution fails with a States.Timeout.

600

WordPress_Raw_To_Bronze Standard

↺ Undo   ↻ Redo   ⊕ Zoom in   ⊖ Zoom out

Q Search   ‹

Actions   **Flow**   **Patterns**   Info

MOST POPULAR

AWS Lambda
Invoke

Amazon SNS
Publish

Amazon ECS
RunTask

AWS Step Functions
StartExecution

AWS Glue
StartJobRun

THIRD-PARTY API

HTTP Endpoint
Call third-party API

Zoom out   Center   Duplicate   Delete

Start

Lambda: Invoke
data_wordpressapi_raw

Lambda: Invoke
data_wordpressapi_bronze

Glue: StartCrawler
Start Bronze Crawler

Catch #1

Glue: StartJobRun
Glue Start Silver Python Shell

Catch #1

Parallel state
Parallel

...lesetEvaluationRun
...ics_pages

Glue: StartDataQualityRulesetEvaluationRun
DQ Run: silver-posts

Catch #1

Glue: StartCrawler
Start Silver Crawler

Catch #1

SNS: Publish
PublishFailure

End

Feedback

Workflow   ⬤ Definition ＞

The top level Amazon States Language properties for this workflow. Learn more ⧉

Start at
The state that is the starting point of the workflow.

data_wordpressapi_raw ▼

Comment - *optional*
A human-readable description of the state machine.

Runs Lambda functions to ingest WordPress API data and transform to Parquet.

TimeoutSeconds - *optional*
The maximum number of seconds an execution of the state machine can run. If it runs longer than the specified time, the execution fails with a States.Timeout.
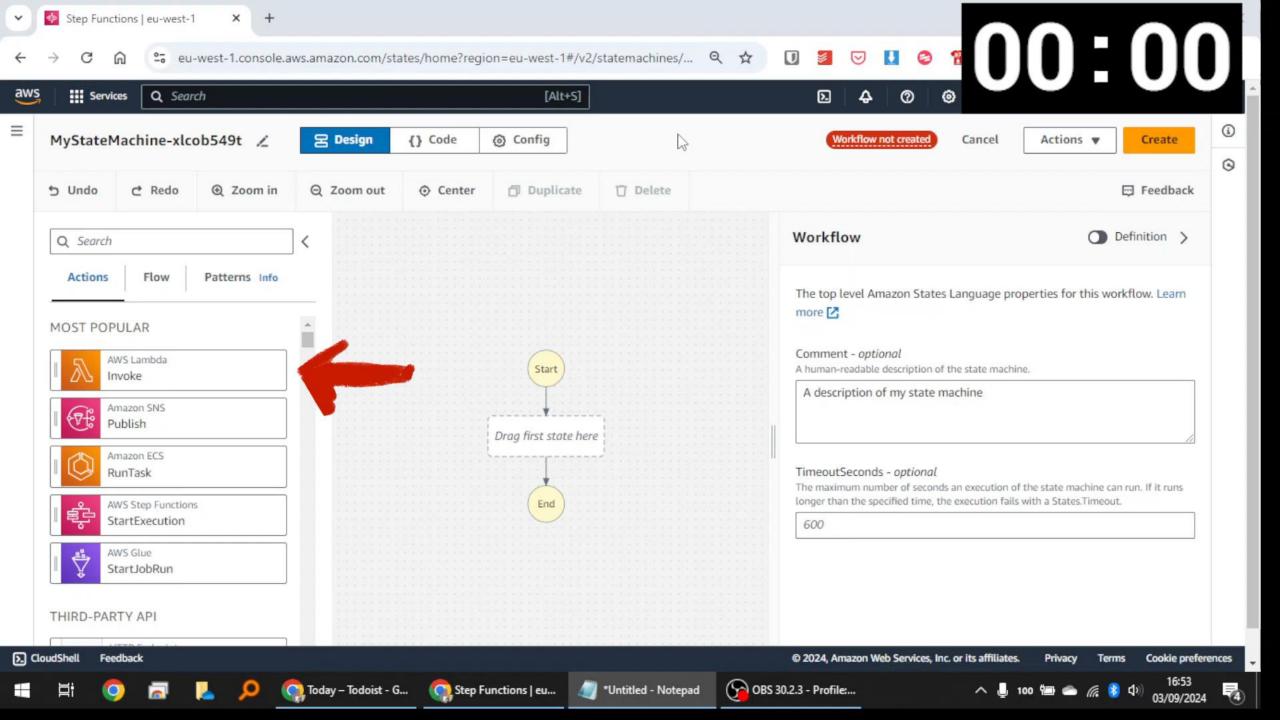
600

AWS Step Functions Demo

Lambda Function: API Call

Glue Job: ETL

Athena Query: MSCK REPAIR TABLE

Amazon EventBridge Scheduler

Automate recurring & one-off tasks

Invoke over 220 AWS services
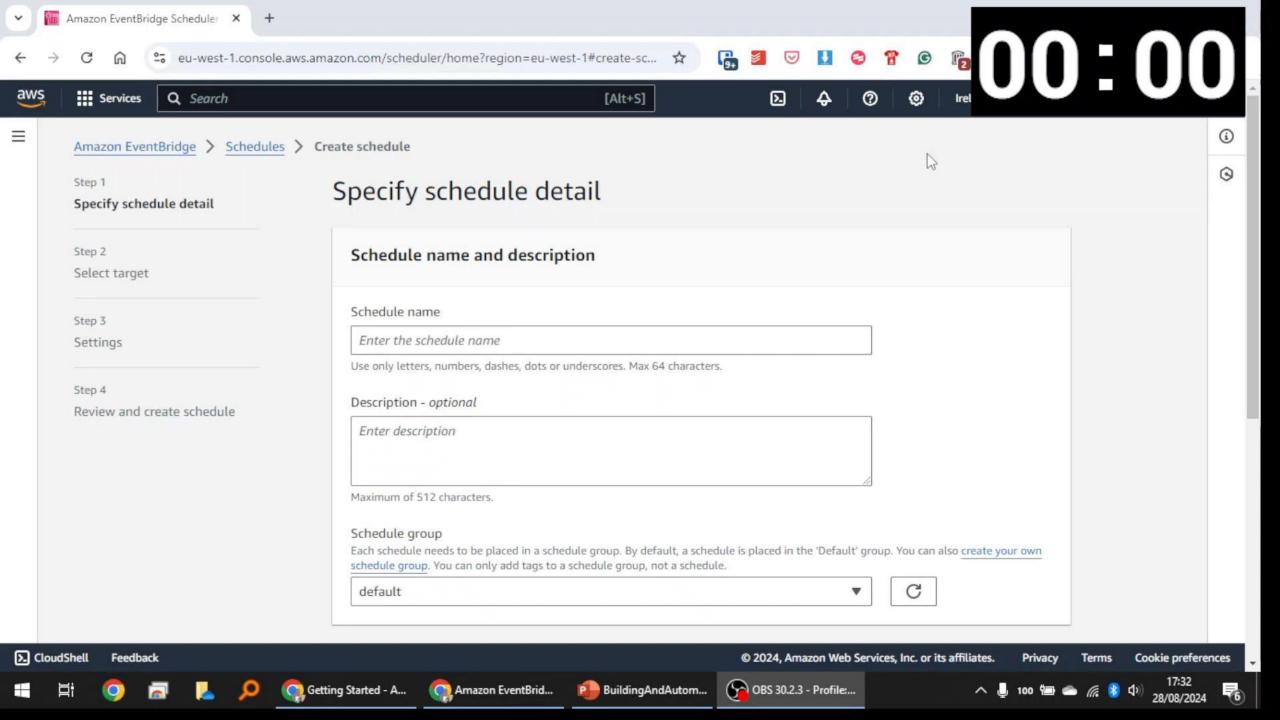
Set times or fixed-rate schedules

Checks target response

Amazon EventBridge Scheduler Demo

Set schedule

Link Step Function workflow

Set configuration

Specify schedule detail

Schedule name and description

Schedule name

Enter the schedule name

Use only letters, numbers, dashes, dots or underscores. Max 64 characters.

Description - optional

Enter description

Maximum of 512 characters.

Schedule group

Each schedule needs to be placed in a schedule group. By default, a schedule is placed in the 'Default' group. You can also create your own schedule group. You can only add tags to a schedule group, not a schedule.

default

Step 1
Specify schedule detail

Step 2
Select target

Step 3
Settings

Step 4
Review and create schedule

Summary

Problem Definition

Solution Architecture

Demos

Summary & Questions

github.com/MrDamienJones
/Community-Sessions

Thanks!

in MrDamienJones

github MrDamienJones

W amazonwebshark.com

✉ damien@amazonwebshark.com

🐦 @amazonwebshark

▶ @amazonwebshark

github.com/MrDamienJones
/Community-Sessions