

Cattaneo Luca 1079489  
Locatelli Erica 1081101

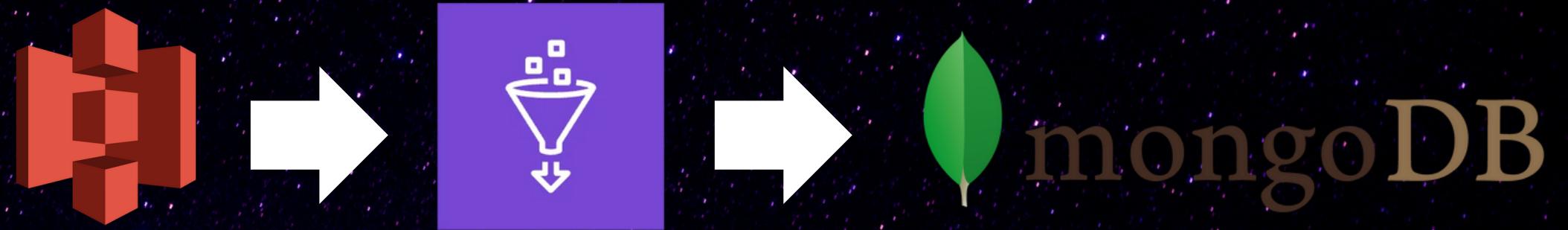
B E Y O N D

T E R R A F O X



HW2

# WATCH\_NEXT



Il nostro primo obiettivo è prendere i dati grezzi da diverse fonti, arricchirli, e creare un Data Warehouse pulito e tematico su MongoDB.

- Il Job legge diversi file CSV contenenti informazioni su talk, dettagli, immagini, tag e video correlati.
- Attraverso numerosi Join, unisce i dataset in uno unico.
- Aggrega i tag e i video suggeriti in liste per ogni talk.
- Applica un filtro tematico conservando solo i talk pertinenti al nostro tema (scienza, spazio, pianeti, Big Bang, ecc.).

# WATCH\_NEXT

1. Preleviamo i file dal bucket per costruire il dataset

```
# READ THE DETAILS
details_dataset_path = "s3://tedx-2025-data-mp-30072025/details.csv"
details_dataset = spark.read.option("header", "true").option("quote", "\"").option("escape", "\\\"").csv(details_dataset_path)
details_dataset = details_dataset.select(col("id").alias("id_ref"), col("description"), col("duration"), col("publishedAt"))

# READ THE IMAGES
images_dataset_path = "s3://tedx-2025-data-mp-30072025/images.csv"
images_dataset = spark.read.option("header", "true").option("quote", "\"").option("escape", "\\\"").csv(images_dataset_path)
images_dataset = images_dataset.select(col("id").alias("id_ref"), col("url"))
```

2. Eseguiamo i join

```
# BUILD THE MAIN DATASET BY JOINING DETAILS AND IMAGES
tedx_dataset_main = tedx_dataset.join(details_dataset, tedx_dataset.id == details_dataset.id_ref, "left").drop("id_ref")
tedx_dataset_main = tedx_dataset_main.join(images_dataset, tedx_dataset_main.id == images_dataset.id_ref, "left").drop("id_ref")
tedx_dataset_main.printSchema()
```

3. Filtriamo i tag più pertinenti

```
# FILTERING FOR BeyondTEDx
filtered_dataset = tedx_dataset_agg.where(size(array_intersect(col("tags"),
array(lit("science"), lit("space"), lit("aliens"), lit("asteroid"), lit("astronomy"),
lit("Big Bang"), lit("dark matter"), lit("exploration"), lit("evolution"),
lit("innovation"), lit("Mars"), lit("Moon"), lit("Mission Blue"), lit("planets"),
lit("NASA"), lit("quantum"), lit("rocket science"), lit("Solar System"), lit("Sun"),
lit("String Theory"), lit("Universe")))) > 0)
```

# WATCH\_NEXT: RISULTATI

```
_id: "563312"
slug : "kris_de_meyer_feeling_stuck_on_climate_change_here_s_what_to_do"
speakers : "Kris De Meyer"
title : "Feeling stuck on climate change? Here's what to do"
url : "https://talkstar-assets.s3.amazonaws.com/production/talks/talk_145018/..."
description : "To spark action on climate change, the conventional wisdom says that a..."
duration : "772"
publishedAt : "2025-02-21T15:48:50Z"
tags : Array (5)
WatchNext_id : Array (6)
  0: "4463"
  1: "137123"
  2: "192"
  3: "142396"
  4: "243"
  5: "115477"
WatchNext_title : Array (6)
  0: "How to transform apocalypse fatigue into action on global warming"
  1: "Listen to your intuition – it can help you navigate the future"
  2: "A critical look at geoengineering against climate change"
  3: "How to make big decisions in challenging circumstances"
  4: "New thinking on the climate crisis"
  5: "Why change is so scary – and how to unlock its potential"
```



MongoDB

# LEARNING PATH- AWS GLUE E LAMBDA FUNCTION

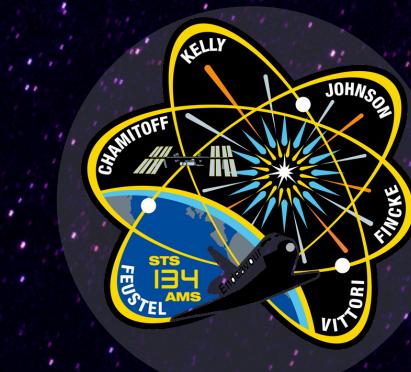


Lo scopo di questo secondo job è di utilizzare i dati preparati dal primo job per costruire dinamicamente dei "Percorsi Formativi". Non gestisce dati grezzi, ma lavora su informazioni già pulite e aggregate. Quando un utente studente completa il percorso sblocca un badge personalizzato che arricchirà la sua collezione.

- Il job legge il dataset di base, già arricchito con i tag, dalla collezione tedx\_data su MongoDB.
- Il job fa un ciclo su una lista predefinita di temi chiave e, per ogni tema, filtra e raggruppa tutti i talk pertinenti.
- Infine, crea i metadati del percorso e una descrizione.

La funzione Lambda BeyondTEDx-LearningPath-generator funge da “ponte” tra lo scheduler e il motore di elaborazione dati.

- Viene attivata automaticamente una volta al giorno dal trigger schedulato su Amazon EventBridge.
- Avvia il Job AWS Glue



# LEARNING PATH- AWS GLUE E LAMBDA FUNCTION



```
# temi principali dei path
main_tags_for_paths = ["Mars", "dark matter", "rocket science", "innovation", "exploration", "NASA", "Universe"]
all_paths_dfs = []

for tag in main_tags_for_paths:
    print(f"--> Elaborazione percorsi per il tag: '{tag}'")

    # Filtra i talk che contengono il tag corrente
    talks_for_path = base_talks_df.filter(array_contains(col("tags"), tag))

    # Crea un percorso solo se ci sono almeno 3 talk sull'argomento
    if talks_for_path.count() >= 3:
        path_agg_df = talks_for_path.sort("publishedAt") \
            .agg(
                collect_list("id").alias("talk_ids"),
                _sum(col("duration").cast("long")).alias("total_duration_sec")
            )
```

LAMBDA



## JOB AWS GLUE

### # 2. GESTORE DELLA LAMBDA

```
def lambda_handler(event, context):
    logger.info(f"## TRIGGER RICEVUTO ##")
    logger.info(f"## Tentativo di avvio del job Glue: '{GLUE_JOB_NAME}' ##")

    try:
        # Avvia l'esecuzione del job specificato.
        response = client.start_job_run(JobName=GLUE_JOB_NAME)

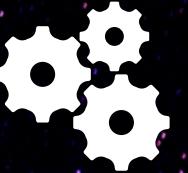
        run_id = response['JobRunId']

        logger.info(f"## SUCCESSO ##")
        logger.info(f"Job '{GLUE_JOB_NAME}' avviato con successo.")
        logger.info(f"Job Run ID: {run_id}")

        # Restituisce una risposta positiva.
        return {
            'statusCode': 200,
            'body': json.dumps(f"Job {GLUE_JOB_NAME} avviato. Run ID: {run_id}")
        }
```

# LEARNING PATH- AWS GLUE E LAMBDA

## FUNCTION: RISULTATI



```
_id: "NASA"
▶ talk_ids : Array (16)
  path_title : "Explore: Nasa"
  talks_count : 16
  total_duration_minutes : 182
  badge_to_unlock : "Expert in Nasa"
  path_description : "A path of 16 talk (182.0 min) to explore the theme of NASA."
  main_tag : "NASA"
```



MONGO DB

```
_id: "Mars"
▶ talk_ids : Array (25)
  path_title : "Explore: Mars"
  talks_count : 25
  total_duration_minutes : 435
  badge_to_unlock : "Expert in Mars"
  path_description : "A path of 25 talk (435.0 min) to explore the theme of Mars."
  main_tag : "Mars"
```

# CRITICITÀ TECNICHE

01

## Mancanza di Tag

Alcuni Ted non hanno nessun tag associato e quindi non potranno mai essere consigliati in watch\_next

02

## Problema di pertinenza

Alcuni Ted possono risultare non completamente pertinenti con i temi scelti per la nostra applicazione

03

## Gestione della Configurazione di rete

Problemi nella configurazione di rete per comunicare con MongoDB, che ha causato il fallimento immediato del job



# POSSIBILI EVOLUZIONI

01

## Quiz finali

Introdurre dei quiz per verificare le competenze acquisite al termine del percorso formativo.

02

## Aumentare il numero di tag quando si filtra il dataset

In modo da generare percorsi più mirati agli interessi degli utenti

03

## Espandere il dataset

Fare scraping regolarmente in modo che il dataset sia sempre completo e aggiornato

04

## Valutazione dei path

Permettere agli utenti di esprimere valutazioni anche sui percorsi formativi oltre che sui singoli ted.

