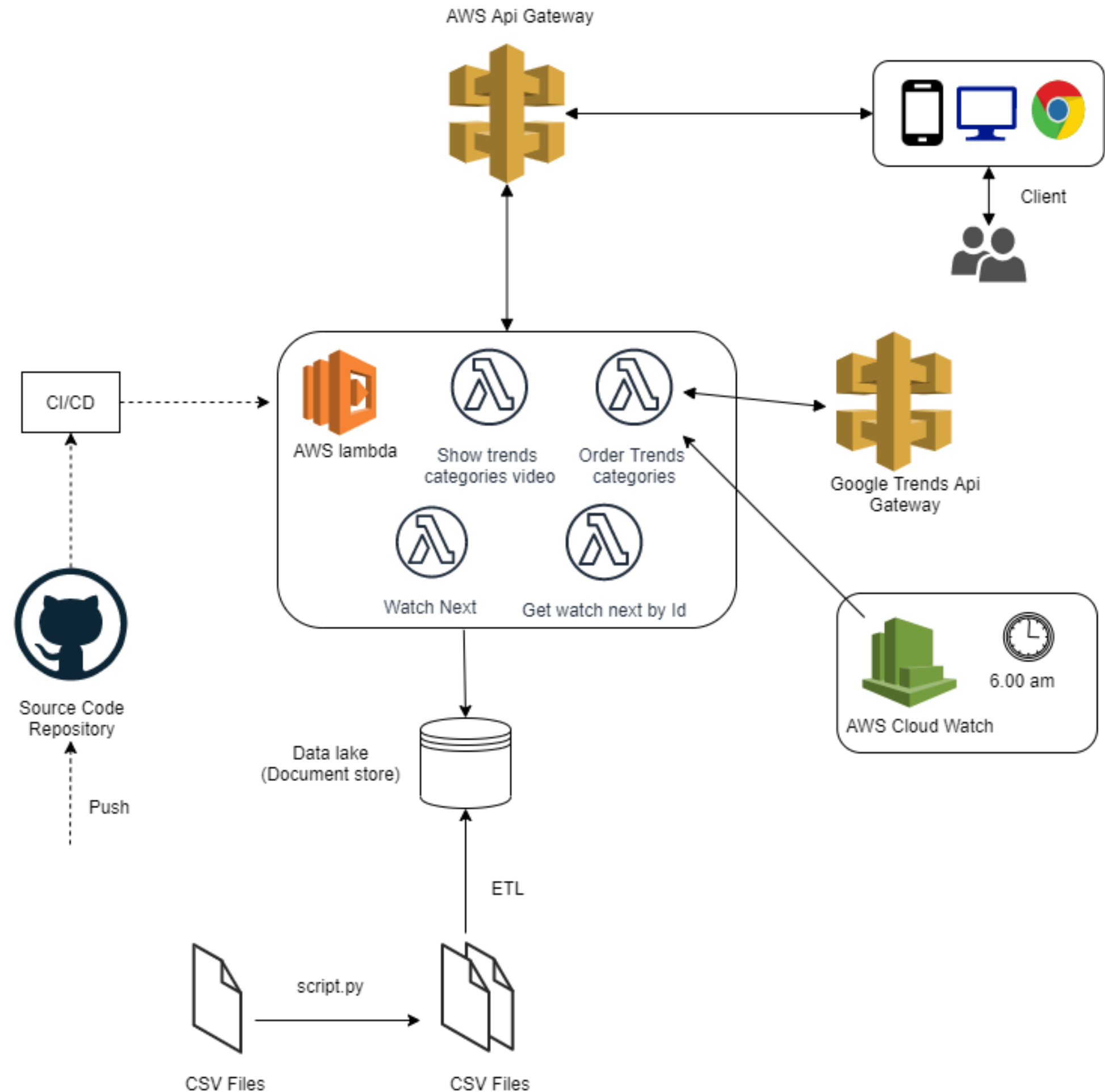




Samuele Ferri, Steven Gotti, Simone Sudati
3 - AWS Lambda Functions

Architettura

Abbiamo aggiornato la nostra architettura:
attraverso **AWS Cloud Watch (Event)**
invochiamo ogni giorno la funzione
Order Trends Categories che aggiorna il
campo *google_trens_score* per ogni
categoria



01

Get_Watch_Next_by_Idx

La prima Lambda function che abbiamo realizzato serve ad esporre un API che restituisce il video successivo a quello attuale.

Tra la lista dei "Watch Next" del video attuale selezioniamo quello con più visualizzazioni e ne ritorniamo:

- Titolo
- Descrizione
- URL

```
connect_to_db().then(() => {
  console.log('=> get_all talks');
  talk.find({_id: body.idx})
    .then(talks => {
      talk.find({_id: talks[0].watch_next_s})
        .sort('num_views')
        .then(t => {
          callback(null, {
            statusCode: 200,
            body: JSON.stringify({id: t[0]._id, title: t[0].title, details: t[0].details, url: t[0].url, num_views: t[0].num_views})
          })
        })
    })
  .catch(err =>
    callback(null, {
      statusCode: err.statusCode || 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the talks.'
    })
  );
});
```

02

Update Google Score

Collezione

- I documenti della collezione rappresentano i tag/categorie dei video a disposizione.
- Ogni categoria, nel campo *google_api_score*, contiene il punteggio con il quale verranno ordinate le categorie da presentare all'utente.

```
_id: "Google"  
> video_idx: Array  
google_api_score: "32"
```

```
_id: "crowdsourcing"  
> video_idx: Array  
google_api_score: "12"
```

```
_id: "Mars"  
> video_idx: Array  
google_api_score: "98"
```

```
_id: "String theory"  
> video_idx: Array  
google_api_score: "73"
```

```
_id: "animation"  
> video_idx: Array  
google_api_score: "87"
```

03

Update Google Score

- Event Trigger: AWS Cloud Watch Event
La lambda function viene azionata giornalmente alle 6 di mattina.
- Function: Interrogazione a Google Trends API per ottenere lo score giornaliero delle categorie e aggiornare il campo *google_score_api*.
- NPM: google-trends-api
(<https://www.npmjs.com/package/google-trends-api>)

The screenshot displays the AWS CloudWatch Event console configuration page. It is divided into two main sections: 'Name and description' and 'Define pattern'.

Name and description:

- Name:** A text input field containing 'TriggerLambdaUpdateGoogleTrendsScore'. Below the field, a note states: 'Maximum of 64 characters consisting of lower/upper case letters, -, _, .'.
- Description - optional:** A text input field with the placeholder text 'Enter description'.

Define pattern:

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Two options are available for defining the pattern:

- Event pattern:** Indicated by an unselected radio button. Subtext: 'Build a pattern to match events'. An 'Info' link is present.
- Schedule:** Indicated by a selected radio button. Subtext: 'Invoke your targets on a schedule'. An 'Info' link is present.

Under the 'Schedule' option, there are two sub-options:

- Fixed rate every:** Includes a numeric input field and a dropdown menu currently set to 'Hours'.
- Cron expression:** Indicated by a selected radio button. Subtext: 'CRON expression have six required fields, which are separated by white space. [Learn more about CRON expression.](#) Enter CRON expression below to see the next 10 trigger date(s)'. Below this is a text input field containing the expression 'cron(6 00 * * ? *)'.

At the bottom, there is a section labeled 'Sample event(s)' with a right-pointing triangle icon.

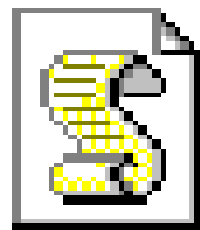
04

Layer google-trends-api

Per poter utilizzare il package di *google-trends-api* è necessario aggiugnere il pacchetto all'ambiente virtuale e caricare il file .zip come layer su AWS da selezionare poi nella lambda function.



node_modules



index.js



package.json



package-lock.js
n

05

Update Google Score

```
connect_to_db().then(() => {  
  
  // Boolean, if `google-trens-api` is working (true) or not (false)  
  var isGoogleTrendsAPIWorking = false // Deprecated package  
  
  async function f() {  
    for await (const c of cat.find()) {  
      console.log(c)  
  
      var key = c._id  
  
      if (isGoogleTrendsAPIWorking) {  
        // Get the categories' score from 0 to 100  
        googleTrends.interestOverTime({keyword: key, startTime: new Date(Date.now() - 1)}).then(function(results){  
          console.log(results)  
  
          // Score from Google Trends API  
          var score = results  
        })  
        .catch(function(err){  
          console.error(err)  
        });  
      } else {  
        // Score random  
        var score = Math.floor((Math.random() * 100) + 1);  
      }  
  
      await cat.updateOne({_id: key}, {google_api_score: score});  
    }  
  }  
  
  f()  
})
```

06

Dato che il package è deprecato, i valori sono settati momentaneamente in maniera randomica

Esperienza utente

L'utente, ogni mattina, avrà sulla propria homepage le categorie di video più attuali nelle ultime ore.

Per ogni categoria viene consigliata la visione dei migliori video basandosi sul numero di visualizzazioni.

07

Criticità tecniche

- Selezionare i video in base ad un tag in tendenza può produrre risultati non del tutto pertinenti, in quanto i video hanno anche altri tag e potrebbero trattare un argomento leggermente diverso.
- Abbiamo riscontrato che alcuni video presenti in alcune liste dei Watch Next non hanno un corrispettivo ID nel database aggiornato al 2021.
- A causa del pacchetto deprecato, non siamo riusciti a recuperare tutti i valori da Google Trends API.
- A causa delle restrizioni dell'account Student non si poteva usare [AWS Cloud Watch Event](#)

08

Possibili evoluzioni

- Trovare un modo smart per associare ai video meno tag ma più pertinenti, oppure dei "super-tag".
- Il video successivo ad ora proposto è quello maggiormente visto tra i Watch Next. Una possibile evoluzione sarebbe di selezionare solamente i video con tag simili o in base ad un indice calcolato ad-hoc.
- Trovare un altro modo per prendere i dati dalle API di Google visto che il pacchetto in questione è deprecato.

Links

GitHub Repository: <https://github.com/samuelexferri/unibg-cloudmobile>

Trello Board: <https://trello.com/b/iEHLw5as/tcm-1>

10