

Your Own Serverless Computing Application: COVID-19 Evolution Charts

Sistemi Distribuiti e Cloud Computing - A.A. 2019/2020

Flavio Scaccia
Ingegneria informatica
Università Tor Vergata
Roma, Italia
flavio.scaccia@gmail.com

1. INTRODUZIONE

COVID-19, il virus che ha scatenato l'epidemia partita dalla Cina e che ha costretto le autorità a mettere in quarantena l'intera città di Wuhan, punto da cui il virus si sarebbe propagato, è diventato l'argomento centrale degli ultimi mesi. L'emergenza è diventata in poco tempo globale producendo danni in ambito sanitario ed economico, arrivando a toccare i settori commerciali, del turismo e del lavoro. Le varie nazioni del mondo sono state costrette a prendere misure adeguate per fare in modo che il virus potesse essere contenuto il più possibile, stabilendo quarantene e provocando il blocco totale del sistema. Lavarsi le mani e usare la mascherina sono ottimi rimedi per contrastare l'epidemia ma assolutamente non sufficienti a contenere il virus e allo stesso tempo permettere ai cittadini di poter tornare a lavorare e vivere la vita normalmente, anche se con qualche limitazione. È qui che si capisce l'importanza di questo progetto: nasce infatti l'esigenza di poter monitorare giorno dopo giorno l'andamento del virus per poterlo tenere sotto controllo ed implementare conseguenti misure di contenimento.

2. DESCRIZIONE ARCHITETTURA PROGETTUALE

L'obiettivo del progetto è quello di permettere ad un utente di poter visualizzare l'andamento del virus nel tempo attraverso grafici interattivi che mostrano la

situazione giorno per giorno. L'interfacciamento a tali grafici avverrà per mezzo di pagine web dove l'utente potrà navigare dal proprio browser. I grafici generati consentiranno di rappresentare l'andamento nazionale e quello regionale della situazione in Italia. Inoltre, l'utente avrà la possibilità di poter scaricare in locale i dati nazionali relativi a tutte le variabili tenute sotto controllo: *ricoverati con sintomi, terapia intensiva, totale ospedalizzati, isolamento domiciliare, totale positivi, variazione totale positivi, nuovi positivi, dimessi guariti, deceduti, casi da sospetto diagnostico, casi da screening, totale casi, tamponi, casi testati*.

Per la costruzione di un'applicazione così, si farà uso dell'infrastruttura proposta da Amazon: AWS. AWS offre uno svariato numero di servizi rendendolo uno tra i migliori provider cloud del mercato. Esso permette di utilizzare servizi relativi al calcolo e all'archiviazione delle risorse, servizi utilizzati in particolar modo per la costruzione di questa applicazione web, progettandola secondo un'architettura di tipo *Serverless*.

2.1. SERVERLESS

Serverless è l'architettura nativa del cloud che consente di trasferire le responsabilità operative ad AWS. Essa consente di creare ed eseguire applicazioni e servizi senza dover gestire in prima persona alcun server. Elimina le attività di gestione delle infrastrutture come il provisioning del server, la scelta del sistema operativo e il provisioning della capacità. Tutte le operazioni necessarie per

l'esecuzione e la scalabilità dell'applicazione saranno gestite in automatico. Il serverless consente di creare applicazioni a un basso costo, infatti i costi operativi risultano ridotti perché viene addebitato solo il tempo di calcolo effettivamente impiegato nel cloud, e non quello di esecuzione dei server. Inoltre viene automaticamente gestita anche la tolleranza ai guasti.

2.2. SERVIZI

Per questa applicazione serverless vengono utilizzati tre differenti servizi:

- **Lambda**

Lambda è un servizio di elaborazione che consente di eseguire il codice senza gestire i server o effettuare l'approvvigionamento. Il codice viene eseguito solo quando viene richiamato da un determinato evento (trigger), che potrebbe essere ad esempio una richiesta di un utente intermediata dal servizio *API Gateway*; in questo modo potrà essere utilizzato per effettuare un'elaborazione di dati e memorizzazione in S3. Con Lambda è possibile quindi costruire la parte di back-end della propria applicazione. Per il suo utilizzo è necessario utilizzare uno dei linguaggi supportati, in questo caso Python.

- **API Gateway**

Amazon API Gateway è un servizio AWS per la creazione, la pubblicazione, la gestione, il monitoraggio e la protezione di API REST, HTTP e WebSocket. Gli utenti si interfacceranno quindi a tale servizio, che farà in modo di richiamare la corretta funzionalità da eseguire, ad esempio richiamare le funzioni Lambda.

- **S3**

Amazon S3 è uno storage di oggetti creato per memorizzare e recuperare dati da qualsiasi origine. È un servizio altamente disponibile e scalabile a costi ridotti.

2.3. IAM ROLE

Un ruolo IAM è un'identità che bisogna creare nell'account per disporre di autorizzazioni specifiche. Invece di essere associato in modo univoco a una persona, un ruolo è destinato a essere assunto da chiunque e non ha credenziali associate.

Tale ruolo deve essere necessariamente configurato per poter delegare l'accesso agli utenti alle applicazioni o ai servizi utilizzati.

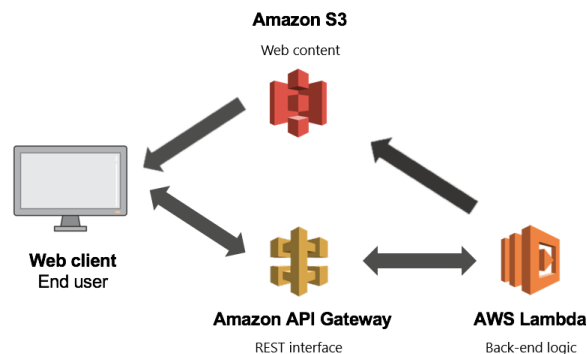


Figura 1. Architettura Serverless usata

3. SCELTE PROGETTUALI

Attraverso l'account AWS Educate è possibile utilizzare tutti i servizi citati sopra. Il funzionamento dell'applicazione web sul Covid consisterà dunque in un sito web memorizzato all'interno del servizio S3. Dalla pagina principale è possibile navigare nel sito attraverso determinati link o bottoni. Le funzionalità che possono essere eseguite sono 3: visualizzazione dei grafici relativi all'andamento nazionale, visualizzazione dei grafici relativi al confronto tra regioni e, come ultima funzionalità, il download dei dati nazionali.

Le prime due saranno visualizzate in un'apposita pagina web, mentre l'ultima avvierà il download direttamente dalla pagina principale.

Ogni punto di accesso a tale funzionalità viene gestito da apposite API Gateway che fungeranno da intermediatori tra la richiesta dell'utente e l'esecuzione della funzione Lambda. Le API Gateway in questione operano con richieste REST di tipo *GET*. Nella loro configurazione viene abilitato il CORS (Cross-Origin resource sharing) in modo da poter consentire richieste effettuate da domini diversi da quello di Amazon.

Ogni funzione Lambda svolgerà il lavoro back-end di tali funzionalità. Essa conterrà il codice scritto in linguaggio Python3, facendo uso di varie librerie. Una

volta richiamate, svolgeranno l'esecuzione del codice e restituiranno un determinato valore in caso di successo o insuccesso della funzione.

Le prime due funzioni Lambda (successivamente spiegate nel dettaglio), interagiranno con il servizio S3. Infatti dopo la costruzione della dovuta pagina html con i grafici aggiornati, passerà all'eliminazione da S3 di eventuali pagine vecchie e alla memorizzazione delle pagine appena create. Una volta terminata la memorizzazione, la funzione lambda restituirà il risultato di successo/insuccesso alla API Gateway chiamante che a sua volta permetterà all'utente di passare alla pagina html per la visualizzazione dei grafici.

I grafici saranno creati in modo da poter permettere un'interazione dell'utente con essi, per poter visualizzare più dettagliatamente eventuali informazioni.

3.1. FUNZIONALITÀ

L'utente sarà in grado di poter effettuare le tali funzionalità:

- **Visualizzazione grafici nazionali**

Si passerà ad una pagina web dove sarà possibile visualizzare diversi grafici. Questi sono stati creati nel momento in cui l'utente ha espresso la volontà di volerli visualizzare nella pagina principale e sono stati generati sul momento dall'apposita funzione Lambda che si sarà occupata del reperimento dei dati, la generazione dei grafici, la generazione della nuova pagina html con i grafici aggiornati e finale memorizzazione di questi su S3. In particolare, i grafici nazionali avranno per ascissa le date e per ordinata il valore delle variabili scelte per la visualizzazione del grafico.

Tutti i grafici sono interattivi, c'è quindi la possibilità di zoomare, muoversi sul grafico o porre il mouse su un punto della curva per visualizzare i valori in una determinata data.

Si è deciso di costruire tre grafici nazionali. Il primo è stato generato per la visualizzazione di quattro curve con modalità "stack", in modo che nessuna delle variabili si sovrapponga al grafico di un'altra ma tutti i grafici siano impilati uno sopra l'altro. Le variabili mostrate e quindi le curve mostrate saranno riferite a

"nuovi positivi", "isolamento domiciliare", "ricoverati con sintomi", "terapia intensiva". Il secondo grafico è di tipo giornaliero, quindi è possibile vedere l'andamento giorno per giorno di "nuovi positivi", "dimessi guariti", "deceduti". Il terzo grafico invece mostra l'andamento dei tamponi sempre giornalmente.

- **Confronto grafici regionali**

Allo stesso modo della visualizzazione dei grafici nazionali, anche per questa funzionalità esisterà un'apposita funzione Lambda che effettuerà gli stessi passaggi, dal reperimento dei dati regionali fino alla memorizzazione delle pagine aggiornate sul servizio S3.

La differenza sta nel tipo di grafici generati: infatti questa funzionalità permette di visualizzare dei grafici che mettono a confronto le curve relative all'andamento regione per regione.

Il primo grafico si riferirà al totale dei casi per ogni regione, mostrando come alcune di queste regioni hanno un maggiore impatto a livello nazionale.

Il secondo invece mostra il numero totale dei tamponi che vengono effettuati nelle varie regioni.

- **Download dati**

Questa terza funzionalità richiamerà sempre una funzione Lambda che questa volta avrà il compito di reperire i dati dalla fonte, creare un file in binario in modo che questo possa essere scaricato e trasmesso all'utente. L'utente sceglierà se leggere i dati o scaricarli in locale.

3.2. STRUMENTAZIONE

- **Linguaggio front-end**

HTML, CSS, JS

- **Linguaggio back-end**

Python3

- **Tool**

Browser per la visualizzazione del sito, Sublime-Text (editor testuale) per la scrittura del codice.

- **Librerie**

Bokeh: Libreria per la visualizzazione interattiva sui moderni web browser. Consente la costruzione dei

grafici in modo semplice e conciso, offrendo svariati modi per la loro costruzione. Inoltre è in grado di generarli in maniera efficiente e veloce anche a fronte di datasets grandi.

Pandas: Libreria che consente la manipolazione di dati di svariati formati, tra cui CSV o JSON. Consente di prelevare tali dati e creare un oggetto Python fatto di righe e colonne che potrà essere gestito in maniera efficiente e veloce.

Datetime: Modulo che offre la possibilità di formattare in modo appropriato le date e il tempo. Viene utilizzato per formattare gli output delle date o per accedere all'esatta data attuale.

Requests: Libreria che consente la creazione di richieste http in Python, che possono essere di diversi tipi (GET, POST, etc...).

Boto3: Libreria che permette di interagire con i servizi di Amazon Web Services (AWS) direttamente da Python. Abilita gli sviluppatori a creare, configurare e gestire servizi come S3 o EC2.

4. IMPLEMENTAZIONE

• Costruzione infrastruttura su AWS

Innanzitutto risulta necessaria la creazione di un IAM Role, in questo caso chiamato "lambda-s3-role", che è stato configurato con la policy "AWSExecuteLambda" che permette l'esecuzione delle funzioni Lambda una volta richiamate e la policy relativa al permesso di interagire con il Bucket S3 per l'inserimento ed eliminazione degli oggetti da esso. La creazione avviene dalla console di AWS sotto il nome IAM > Create Role e successivamente si effettua una "Attach Policy" per l'aggiunta delle politiche.

Una volta creato il Role, questo dovrà essere utilizzato per qualunque funzione Lambda si voglia creare; solo in questo modo si avrà la possibilità di poterle eseguire.

È stato poi creato un Bucket S3 con il nome "www.plotcovid.it" e lo si è reso pubblico attraverso le Policy per fare in modo che gli oggetti al suo interno possano essere reperibili dagli eventuali utenti

che navigano sul sito. Infatti all'interno del Bucket S3 verranno immagazzinate tutte le pagine html, fungendo da repository del sito. Il Bucket è formato da una pagina di index, una di 404, in caso di pagina non trovata, e due directory. Una directory "risorse" conterrà due pagine html: la prima che mostrerà i grafici nazionali e la seconda i grafici a livello regionale. La seconda directory "grafici" conterrà uno ad uno i singoli grafici, visualizzabili singolarmente a schermo intero dal sito. La creazione di queste pagine avviene dinamicamente nel momento dell'esecuzione delle funzioni Lambda. Nelle pagine generali, oltre ai grafici, verranno mostrate anche la data e l'ora dell'ultimo aggiornamento in modo da verificare che i dati siano recenti.

Sempre dalla console sarà poi necessario creare una funzione Lambda per ogni funzionalità che si vuole offrire all'utente. In questa applicazione ci sono 3 funzioni Lambda scritte in codice Python e descritte dettagliatamente nel paragrafo successivo. Durante la creazione delle funzioni Lambda sarà necessario non dimenticare di attaccare il ruolo precedentemente creato. Inoltre, per la corretta esecuzione del codice, avrà vitale importanza la creazione di un Layer. Il Layer consente di inserire tutte le dipendenze richieste dal codice interno alla funzione Lambda.

Le tre funzioni Lambda in questione sono "CreateNationalCharts", "CreateRegionalCharts", "GetData". Quest'ultime verranno richiamate dall'utente attraverso l'utilizzo di tre API Gateway. Anche queste create dalla console sono state create in modo da rispondere a richieste GET. Durante la creazione è necessario abilitare il CORS per fare in modo che possano essere inviate richieste da utenti che non fanno parte dello stesso dominio di Amazon. Dopo il deploy delle API si avrà a disposizione un link che permette l'esecuzione della funzione Lambda.

• Codice

Il punto di accesso alle varie funzionalità è rappresentato dalla pagina iniziale del sito, "index.html". Questa è una semplice pagina html, con miglioramenti grafici in CSS e gestita da piccoli script Javascript. Tali script vengono usati per la costruzione

di richieste AJAX, quindi richieste http asincrone, che consentono di richiamare le API Gateway per l'elaborazione delle pagine html in tempo reale con i grafici aggiornati. Oltre a queste, viene inserita una loading-bar per mostrare all'utente che la richiesta di visualizzazione o download dei dati è in caricamento.

La pagina di index permette quindi il passaggio nel sito attraverso dei link e un bottone per il download.

Una volta che la richiesta è stata elaborata dalla funzione Lambda, si viene redirezionati al link della pagina generale di visualizzazione di tutti i grafici, quindi alla pagina "italia_completa.html" se la richiesta era di visualizzazione nazionale, o alla pagina "regionali.html" nel secondo caso. Tali pagine sono state create all'interno di codice Python in modo dinamico e conterranno dei riferimenti alle vere pagine html generate e contenenti i singoli grafici. Tale riferimento avviene attraverso il tag <iframe>.

Il codice Python è ciò che verrà eseguito dalla funzione Lambda quando richiamata e quindi rappresenta il cuore di questa applicazione web.

Si prende in considerazione il codice riferito alla funzionalità di visualizzazione nazionale e leggibile all'interno del repository¹ github indicato, con nome "createNationalCharts.py".

Vengono effettuati tre step: reperimento dati dalla fonte, costruzione grafici/costruzione pagine html, interazione con S3 per la memorizzazione delle pagine aggiornate.

Nel reperimento dei dati viene utilizzato un metodo della libreria *Pandas* che consente di scaricare dall'URL indicato e conoscendo a priori il tipo, in questo caso CSV, per poi creare un oggetto di tipo *DataFrame* che può banalmente essere interpretato come una tabella.

Nel secondo step, si passa a costruire i vari file html salvati inizialmente su una cartella temporanea del sistema in cui viene eseguito il codice e che poi una volta sovrascritti con i relativi grafici, verranno salvati

su S3 nell'ultimo step. I dati che si è deciso di estrapolare dalla fonte sono relativi alle regioni, ma questo non risulta essere un ostacolo in quanto è possibile utilizzare un ulteriore metodo "*pivot_table*" della libreria *Panda* che permette di effettuare la somma dei valori con stessa data sommando quindi i valori delle varie regioni e ottenendo il valore nazionale.

Prima della costruzione dei grafici si passano tali valori all'interno di un altro oggetto di tipo *DataSource* che permette di essere usato dalla libreria di creazione grafici, *Bokeh*.

Una volta che tale struttura dati risulta essere pronta, si passa alla costruzione del grafico effettivo. Si parte dunque dalla costruzione del layout indicando quali sono i valori sull'ascisse (le date) e sulle ordinate (variabile scelte dai dati e inserite all'interno di una lista) per poi passare all'effettiva costruzione delle curve. C'è la possibilità di costruire il grafico in vari modi, infatti si è deciso di utilizzare per un grafico la modalità a stack, per il grafico giornaliero quello delle barre e per i grafici regionali una semplice curva che mette in mostra il confronto con le altre regioni. Per il calcolo dei grafici giornalieri vengono sottratti i valori del giorno precedente.

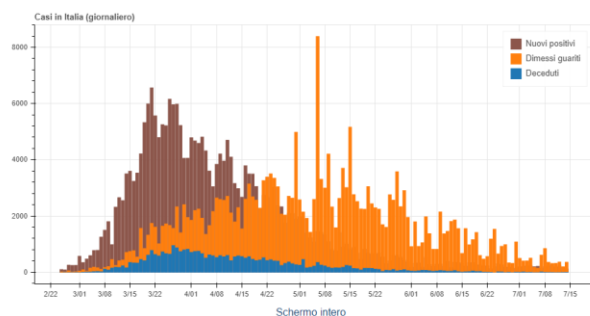


Figura 2. Grafico giornaliero a barre

Quando tutti i grafici sono stati costruiti correttamente e salvati in un file temporaneo html, si passa all'ultimo step, quello di memorizzazione su S3. Attraverso la libreria *Boto3* si ha la possibilità di interagire con il proprio Bucket. Da questo si eliminano inizialmente eventuali vecchie versioni

¹ Repository: <https://github.com/flaxalf/covid-19-evolution>

delle pagine, dopodiché si memorizzano le nuove pagine nelle varie directory.

L'implementazione della costruzione dei grafici regionali presenta un'implementazione simile, con la differenza che i grafici sono creati attraverso dei metodi di bokeh chiamati "line".

Tutti i file Python si riferiranno ad un altro file chiamato "variables.py". In questo vengono inserite tutte le costanti, in particolare conterrà i valori dei vari URL da cui si prelevano i dati, conterrà gli header e i footer html in comune con le varie pagine ed aggiunge infine il valore dell'ultimo aggiornamento dei grafici.

5. LIMITAZIONI E FUNZIONALITÀ FUTURE

Inizialmente, si voleva memorizzare i dati all'interno del servizio S3 e, per farlo, volevo realizzare una funzione Lambda con aggiornamento giornaliero che prelevasse i dati dopo le 18 immagazzinando i dati aggiornati sul sistema di memorizzazione S3. Ciò non è stato possibile in quanto, nel momento della costruzione, ho riscontrato che non sarebbe stato possibile a causa delle limitazioni dell'account AWS Educate relativa al servizio CloudWatch. In particolare la limitazione si è riscontrata nel momento di creazione di un "Cron Expressions" che mi avrebbe permesso di schedulare l'esecuzione della funzione Lambda. Anche se la memorizzazione si sarebbe potuta implementare in altri modi, ho preferito effettuare la generazione dei grafici sul momento riscaricando i dati.

Un'ulteriore limitazione è rappresentata dalla mancanza di configurabilità delle variabili dei grafici da parte dell'utente. Questa rappresenta in ogni caso una potenziale funzionalità futura insieme alla memorizzazione dei dati.

6. CONCLUSIONI

Il risultato di questo progetto è visualizzabile al link: <https://s3.amazonaws.com/www.plotcovid.it/index.html>.

È possibile infine effettuare la lettura dei grafici per trarre qualche conclusione.

Dopo un'analisi, si può notare come si ha un picco dei nuovi positivi il 21 marzo con un valore che risulta essere pari a 6557. Negli ultimi mesi, invece, il contagio si è nettamente abbassato; lo si nota dalla diminuzione dei nuovi positivi. Allo stesso modo, anche il numero di cittadini in isolamento domiciliare è in decrescita.

Si ha un picco dei dimessi guariti nel giorno 6 maggio: 8014 guariti. Dopo varie ricerche, il dato rappresenta però la somma dei guariti di più giorni, una questione di contabilizzazione, dunque.

Si noti poi che il numero di tamponi è cresciuto sensibilmente; inoltre si nota che ogni lunedì il numero di tamponi è nettamente inferiore all'andamento del periodo.

A fronte di un alto numero di tamponi, il numero di positivi è relativamente basso, fatto che tende a sottolineare come ci troviamo in una fase calante del virus.

RIFERIMENTI

- [1] AWS Lambda, 2020. <https://aws.amazon.com/lambda/>.
- [2] AWS API Gateway. <https://aws.amazon.com/it/api-gateway/>.
- [3] AWS S3. <https://aws.amazon.com/it/s3/>.
- [4] Python Bokeh. <https://docs.bokeh.org/en/latest/index.html>.