Presentazione primo compitino

Giudici Mattia 1065452
Trapletti Andrea 1066781
Verdi Michele 1067606

FASI PER LO SCHELETRO PRINCIPALE

- Aggiornamento policy IAM
- Creazione Bucket
- Creazione Dynamo Database
- Creazione Funzione lambda per gestione metodi POST e GET
- Creazione API Gateway

FASI PER LIMITARE L'ACCESSO

- Creazione Autorizzazione
- Creazione Funzione Lambda per verifica Token Autorizzazione

BONUS

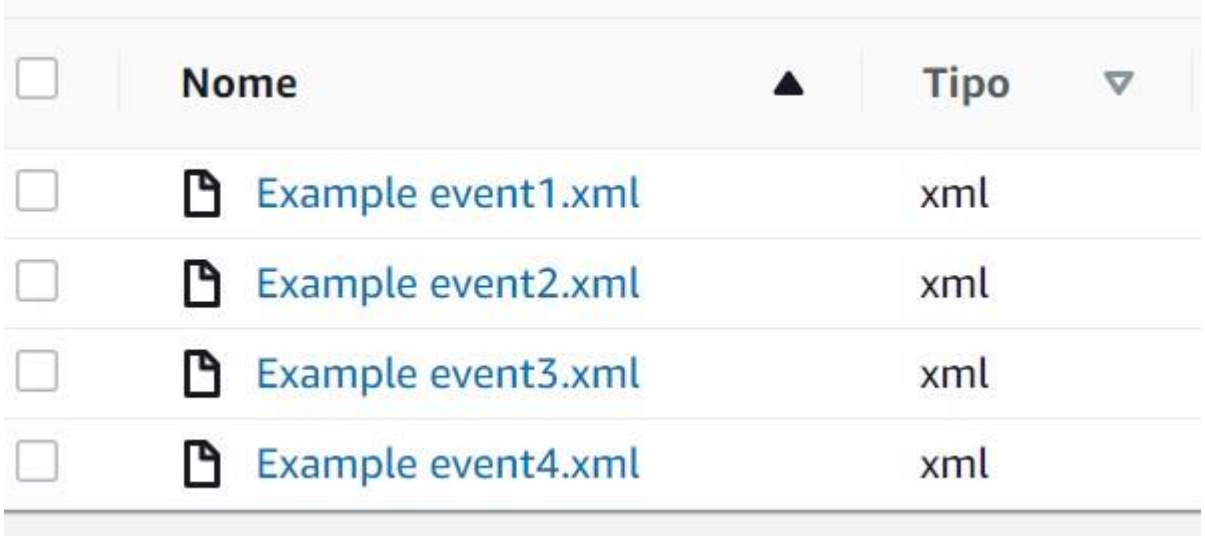
Aggiornamento policy IAM per il ruolo “LabRole”

Non avendo i permessi per creare un nuovo ruolo ne abbiamo utilizzato uno tra i presenti, in questo caso “LabRole”. Abbiamo quindi aggiornato le policy di questo ruolo per avere accesso a tutte le funzioni di Bucket S3, Lambda e API gateway, il risultato finale è il seguente:

Nome della policy	Tipo	Descrizione
c52342a828652l1932278t1w940134865129-VocLabPolicy1-QY…	Gestite …	
c52342a828652l1932278t1w940134865129-VocLabPolicy2-UP…	Gestite …	
c52342a828652l1932278t1w940134865129-VocLabPolicy3-110…	Gestite …	
AmazonEKSClusterPolicy	Gestite …	This policy provides Kubernetes the permissions it requires to manage reso…
AmazonEKSWorkerNodePolicy	Gestite …	This policy allows Amazon EKS worker nodes to connect to Amazon EKS Cl…
AmazonS3FullAccess	Gestite …	Provides full access to all buckets via the AWS Management Console.
AmazonAPIGatewayInvokeFullAccess	Gestite …	Provides full access to invoke APIs in Amazon API Gateway.
AmazonSSMManagedInstanceCore	Gestite …	The policy for Amazon EC2 Role to enable AWS Systems Manager service …
AWSResourceAccessManagerResourceShareParticipantAcc…	Gestite …	Provides access to AWS Resource Access Manager APIs needed by a reso…
AWSLambda_FullAccess	Gestite …	Grants full access to AWS Lambda service, AWS Lambda console features, …

Creazione Bucket in S3

Abbiamo Creato un Bucket all’interno del quale abbiamo inserito i file arrivati da una richiesta di POST, divisi per “ClassResult” (Categoria).



Creazione Dynamo Database

Abbiamo implementato un database realizzato grazie al servizio “DynamoDB”. questo utilizza come unica chiave di partizione “idGara”, come chiave di ordinamento “data” (data del giorno in cui è stato mandato un file tramite una richiesta POST alla funzione lambda). Ogni volta che la funzione lambda inserisce un file gara all’interno del bucket s3 provvede anche ad inserire una voce all’interno di questo database.

idGara	data	Categoria	nome_file
Example ev...	2022-05-03	Men Elite	Example event4.xml
Example ev...	2022-05-03	Open	Example event3.xml
Example ev...	2022-05-03	Open	Example event1.xml
Example ev...	2022-05-03	Men Elite	Example event2.xml

Creazione Funzione Lambda

Abbiamo scritto una funzione utilizzando Lambda in linguaggio python. Un’unica funzione lambda per gestire le richieste di GET e di POST; inoltre gestisce le richieste di POST per le quali mandiamo direttamente un intero file xml, oppure per le quali mandiamo direttamente il testo del file xml. La funzione è in grado di gestire più gare e dà un nome univoco per ogni gara. Se in un file xml ci sono più <ClassResult>, ognuno di questi ClassResult viene gestito come una singola gara. La funzione provvede anche a mandare i file sia nel bucket che nel database. Per poter controllare velocemente il funzionamento della nostra funzione abbiamo utilizzato dei casi di test di cui dispone la funzione Lambda.

Creazione API Gateway

Per far comunicare con l'esterno il nostro cloud abbiamo bisogno di un API, abbiamo implementato l'API tramite il servizio API Gateway, selezionando REST Api.
Abbiamo quindi creato le risorse (collegando ogni risorsa alla nostra lambda, tutte le risorse create sono risorse proxy) alle quali è possibile accedere, ed eventuali metodi, con i quali abbiamo gestito richieste GET su qualsiasi risorsa e il caricamente di un file, tramite POST, solo chiamando la funzione lambda con la giusta autorizzazione. Una volta ultimati questi passaggi abbiamo distribuito la nostra API creando una fase, che ci è servita per ogni aggiornamento alla nostra API.

Creazione API Gateway, le richieste:

POST urlApiDistribuita/funzione_lambda

GET urlApiDistribuita/funzione_lambda/nome_bucket

La richiesta http torna quindi i valori di tutti i file presenti nel bucket. Per leggere uno di questi file:

GET urlApiDistribuita/funzione_lambda/nome_bucket/nome_file

o GET urlApiDistribuita/funzione_lambda/nome_file

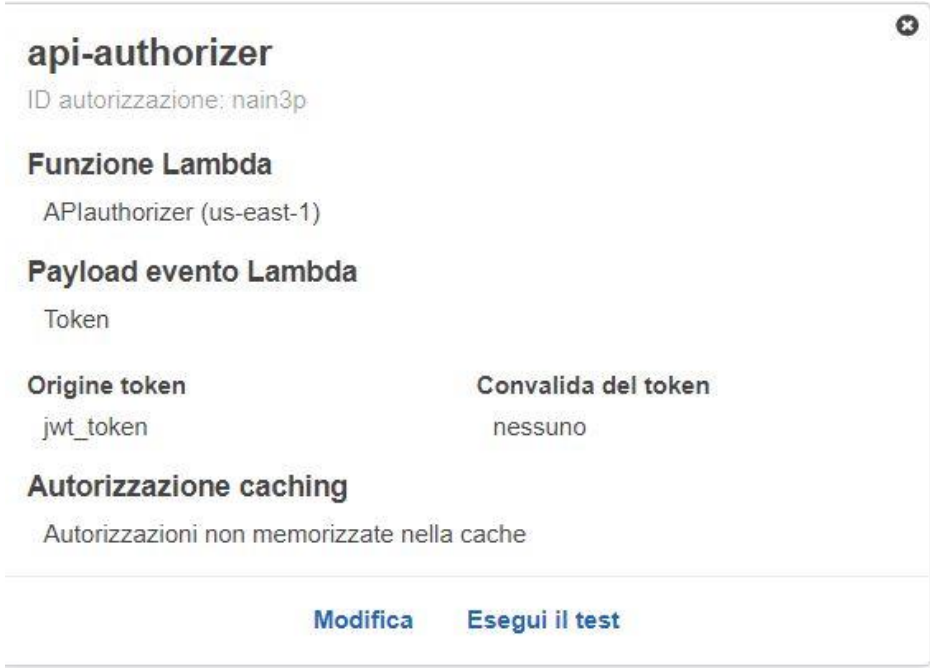
Per leggere un file dentro la cartella fileinteri:

GET urlApiDistribuita/funzione_lambda/nome_bucket/fileinteri/nome_file

Creazione Autorizzazione

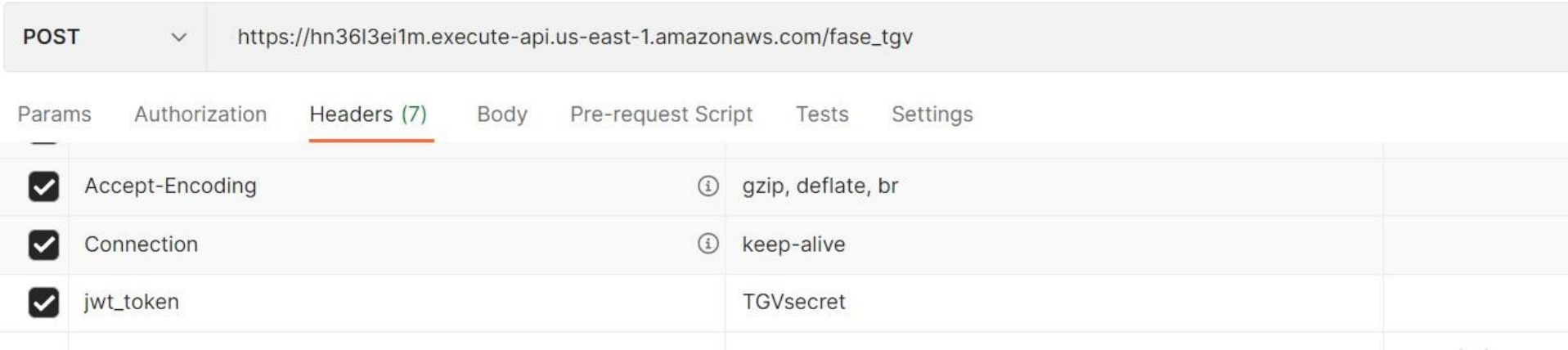
Per poter gestire la possibilità di effettuare una chiamata POST abbiamo utilizzato la funzione autorizzazioni presente all'interno delle impostazioni della nostra API.

Abbiamo quindi creato una nuova autorizzazione di tipo token, e l'abbiamo collegata con una funzione lambda creata appositamente. Per effettuare una richiesta di tipo POST bisogna inserire negli header di essa il corretto nome del token e la chiave esatta.



Creazione Funzione Lambda per gestione autorizzazione

Per verificare che il token inserito negli header sia corretto utilizziamo una funzione lambda, questa verificherà la correttezza della chiave inserita e provvederà quindi alla creazione di una policy da restituire all'API cosi da permettere l'accesso, oppure provvederà con il rifiutare la richiesta.



CODICI FUNZIONI LAMBDA

Il codice della nostra funzione può essere visionato nella seguente repository github alla quale dovrebbe aver ricevuto l'invito:

https://github.com/Mattia-Giudici/TCM-TraplettiVerdiGiudici

Nella stessa repository, inoltre, nella cartella Simulatore, si può trovare il nostro piccolo software per la gestione del punto BONUS del progetto (realizzato con l'utilizzo di Visual Studio Code, linguaggio: python).

BONUS

Abbiamo scritto un piccolo software in python utilizzando Visual Studio Code.

Cliccando sul file Simulatore.py,

viene mandato in esecuzione il nostro

codice, il quale ci permette di accedere

al File System del nostro PC e di

inserire un file XML di una gare, che,

tramite una richiesta POST viene

messo nel nostro Bucket e nel nostro

DB. Se il POST ha successo nella console

troviamo la scritta <Response [200]>.