

Projektowanie systemów rozproszonych

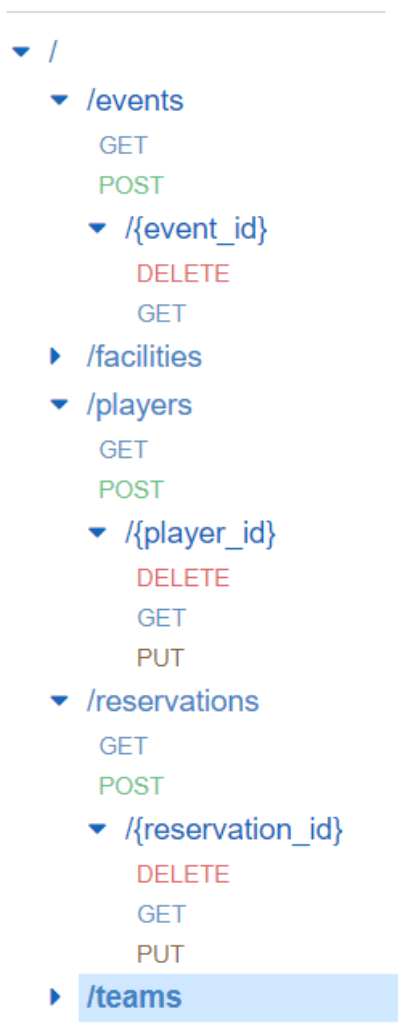
Projekt – aplikacja do zarządzania wydarzeniami sportowymi

1. Opis

Celem projektu było zrobienie aplikacji pozwalającej na planowanie i zarządzanie wydarzeniami sportowymi. Głównymi funkcjami aplikacji jest możliwość dodawania zawodników, grupowanie się w drużyny, możliwość dodawania obiektów sportowych oraz możliwość organizacji wydarzeń z udziałem zawodników (drużyn) na wybranych obiektach (rezerwacje)

2. Struktura

W aplikacji możemy wyróżnić zasoby: events, facilities, players, reservations, teams dodane w usłudze API Gateway. Na zasobach możemy wykonywać operacje CRUD.



Dane przechowywane są w bazie DynamoDB. Utworzone tabele do przechowywania danych dla poszczególnych zasobów.

	Name	Status
<input type="radio"/>	sports-event-manager-Teams577113D3-2	Active
<input type="radio"/>	sports-event-manager-SportFacilitiesF7Ff	Active
<input type="radio"/>	sports-event-manager-Reservation658FF	Active
<input type="radio"/>	sports-event-manager-PlayersCF26CFA0	Active
<input type="radio"/>	sports-event-manager-EventsD32975C2-	Active

Przykładowa struktura tabeli Event:

▼	Item {5}
⊕	id String : c90b04b0-64e6-4651-89b3-67cba44e3090
⊕	name String : GAME#6
⊕	players String : players_list
⊕	▼ reservation Map {6}
⊕	date String : 2020-07-13 20:04:53
⊕	end_date String : 2020-08-16 12:00:00
⊕	id String : 91a6d0b6-9047-4a01-9949-22b45ffc36cb
⊕	▶ player Map {8}
⊕	▶ sport_facility Map {3}
⊕	start_date String : 2020-08-16 10:00:00
⊕	▶ teams List [2]

Obsługa zapytań odbywa się przy pomocy funkcji lambda.

<input type="radio"/>	sports-event-manager-getreservations596CAC18-C2TTZHQUW4VX
<input type="radio"/>	sports-event-manager-deletesportfacilityE3797867-1ND70WSUFS0YI
<input type="radio"/>	sports-event-manager-getreservationDBDC9C7B-1HCQP0B4RZKY8
<input type="radio"/>	sports-event-manager-addevent68766CAE-EB8QK4O04KHC
<input type="radio"/>	sports-event-manager-getsportfacilityCB5E63D6-14UWR917Z5HJ0
<input type="radio"/>	sports-event-manager-addreservation118EF68F-UHEDHUH5VWS7
<input type="radio"/>	sports-event-manager-updatesportfacilityAF1FB46B-1SNRZO7ELCZ1P
<input type="radio"/>	sports-event-manager-addplayerEA660A53-XXJTA59G2P8X
<input type="radio"/>	sports-event-manager-updateteamFE3FF4E9-1QHD1JWMQ7ZZK
<input type="radio"/>	sports-event-manager-getsportfacilities3608DAE1-TXFJAB9XB2NV

3. Wykorzystane usługi

Amazon Reckognition – została wykorzystywana przy dodawaniu zdjęcia profilowego zawodnika. Zdjęcie musi zawierać wyraźną twarz tylko jednej osoby. Pozostałe zdjęcia zostają odrzucone, a do bazy zostaje wpisany komunikat 'BAD PHOTO'.

S3 – do przechowywania dodanych zdjęć profilowych

SES – został wykorzystany do generowania powiadomień o dokonanej rezerwacji. Usługa została skonfigurowana w kodzie, jednak ograniczenie konta AWS nie pozwala na jej przetestowanie.

SQS – kolejka została wykorzystana przy obsłudze dodawania obiektów sportowych.

SNS – usługa została wykorzystana do wysyłania powiadomień email do zawodników o zbliżającym się wydarzeniu do którego zostali dodani wraz z nazwą, datą rozpoczęcia i lokalizacją. Ze względu na konieczności potwierdzania subskrypcji i ewentualnego generowania nowych testowych kont email, funkcjonalność została ograniczona do jednego konta.

4. IaC

Aplikacja została napisana przy pomocy frameworka AWS-CDK w języku python i z wykorzystaniem biblioteki Boto3.

5. CI/CD

Kod aplikacji przechowywany jest w repozytorium git w serwisie GitHub. Do obsługi CD został skonfigurowany Pipeline w usłudze Azure Devops, który na zatwierdzonej zmianę w repozytorium automatycznie przenosi kod na produkcję. Konfiguracja znajduje się w pliku azure-pipelines.yml.

← Jobs in run #20200715.1

SzymonS16.SportsEventManager

Jobs

▼	✔ Job Python37	2m 27s
	✔ Initialize job	2s
	✔ Checkout SzymonS16/S...	3s
	✔ Install Nodejs	<1s
	✔ CmdLine	10s
	✔ Use Python 3.7	<1s
	✔ Install dependencies	20s
	✔ CDKdeploy	1m 49s
	✔ Post-job: Checkout Sz...	<1s
	✔ Finalize Job	<1s

✔ CDKdeploy

View raw log

9 More information on this task can be found in the [task reference](https://docs.aws.amazon.com/vsts/latest/userguide/awsshell.html).

10

11 #####Task Permissions

12 Permissions for this task to call AWS service APIs depend on the activities in the supplied script.

13 -----

14 Configuring credentials for task

15 ...configuring AWS credentials from service endpoint '7a0accde-7836-44a2-8f15-d5a8b425c305'

16 ...endpoint defines standard access/secret key credentials

17 Configuring region for task

18 ...configured to use region us-east-1, defined in task.

19 /bin/bash /home/vsts/work/_temp/awsshellscript_2597.sh

20 sports-event-manager: deploying...

21 [0%] start: Publishing af53ec4f33e7669c006b267708b72e2f08c4678d0f99b81791a7b16a5961d0e9:current

22 [50%] success: Published af53ec4f33e7669c006b267708b72e2f08c4678d0f99b81791a7b16a5961d0e9:current

23 [50%] start: Publishing 4c533099a422584625b8d66c710501450a62411dcf8a53f87e81b19eddf5a04a:current

24 [100%] success: Published 4c533099a422584625b8d66c710501450a62411dcf8a53f87e81b19eddf5a04a:current

25 sports-event-manager: creating CloudFormation changeset...

26

27 ✔ sports-event-manager

28

29 Outputs:

30 sports-event-manager.SportsEventManagerApiEndpointB29A08C6 = https://n2f1pdnt9h.execute-api.us-east-1.amazonaws.com/prod/

31

32 Stack ARN:

33 arn:aws:cloudformation:us-east-1:341220512670:stack/sports-event-manager/8fcffda0-a9bc-11ea-89b2-0aff04b5e65f

34

35 Finishing: CDKdeploy