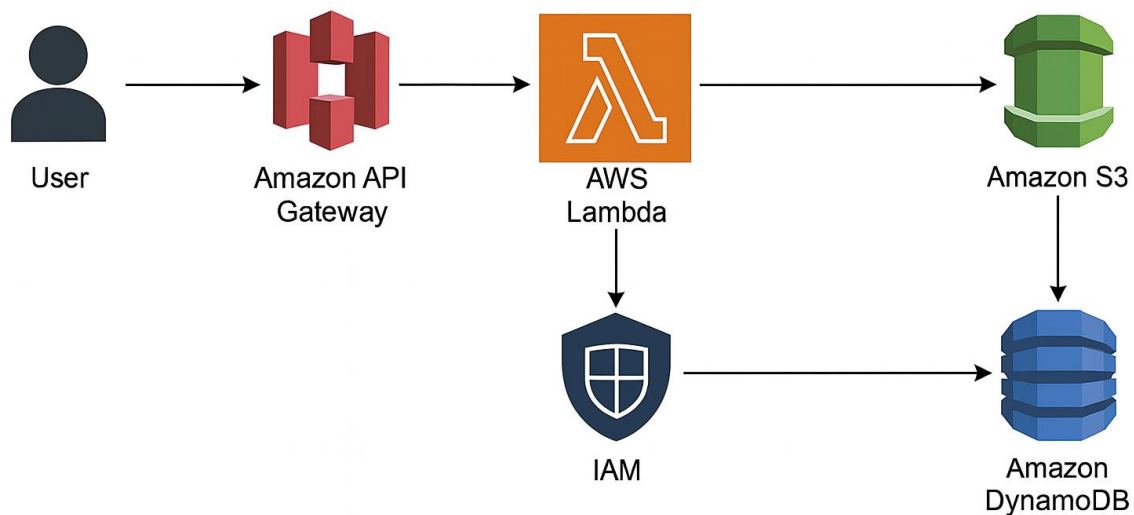


## תיאור ארכיטקטורה – מערכת חלוקת קבוצות בענן AWS



### User

- המשתמש מבצע פעולה כלשהי באפליקציה – לדוגמה: הזנת שחקנים, יצירת משחק, שליחת תמונה, או בקשה לחלוקת קבוצות.

### Amazon API Gateway

- משמש כשכבת תיווך בין המשתמש לבין שרתי הבקאנד.
- מאפשר שליחה מאובטחת של בקשות (HTTP כגון GET, POST וכו').
- מנתב את הבקשה אל הפונקציה הרלוונטית ב-Lambda.

## AWS Lambda

- אחראית על הלוגיקה העסקית של המערכת.
- לדוגמה: שמירת נתוני שחקן, הפעלת אלגוריתם חלוקת קבוצות, ניתוח תמונה.
- מתבצע בצורה **שרת ללא תשתית** – (serverless) הפונקציה מופעלת רק כשצריך ומשלמים לפי שימוש.

## IAM (Identity and Access Management)

- שומר על הרשאות ובקרת גישה בין השירותים.
- מוודא של- Lambda יש גישה רק לטבלאות או דליים שאליהם הוגדרה לה גישה.
- מונע גישה לא מורשית לשירותים רגישים כמו DynamoDB או S3.

## Amazon S3

- משמש לאחסון קבצים סטטיים) כגון HTML/JS/CSS – frontend או תמונות שמנותחות.
- לדוגמה: תמונות פרופיל של שחקנים, צילומי תוצאות או תיעוד משחק.

## Amazon DynamoDB

- מסד נתונים NoSQL מהיר.
- מאחסן מידע מובנה כמו:
  - – games משחקים
  - – players\_new שחקנים
  - – teams קבוצות
- הנתונים נכתבים אליו מתוך Lambda ונשלפים ממנו לפי הצורך.

## סיכום

מערכת זו משתמשת באדריכלות serverless הכוללת את Lambda ו-API Gateway, מאחסן נתונים ב־ DynamoDB ואחסון קבצים ב־ S3, עם הרשאות מאובטחות דרך – IAM פתרון גמיש, מודרני, יעיל וחסכוני בענן של AWS.