

חלק שני – תכנון פריסת המערכת בענן AWS

במהלך תכנון המערכת, חילקתי אותה לשלושה חלקים עיקריים:
Frontend, שירות ראשון לשליפת קטגוריות מ־SQL, ושירות שני לשמירת נתונים ב־MongoDB.
בחרתי לעבוד עם AWS, ולהיעזר בשירותים מנוהלים ככל האפשר לצורך פשטות ואבטחה.

ארכיטקטורה כללית:

- **Frontend:** אפליקציית React + Redux מתארכת ב־S3, מוגשת דרך CloudFront.
- **API Gateway:** משמש כנקודת כניסה מרכזית בין ה־Frontend לשירותי ה־Backend.
- **Backend 1 (.NET + SQL):** רץ ב־ECS Fargate, ומתחבר למסד SQL Server ב־RDS.
- **Backend 2 (Node + Mongo):** רץ גם הוא ב־Fargate, ושומר נתונים ב־MongoDB (Mongo Atlas).

הערה חשובה:

למרות שהדרישה ציינה העדפה ל־Elasticsearch החלטתי להשתמש ב־MongoDB מכיוון שהיה לי ניסיון מוכח איתו בפרויקטים קודמים, ויכולתי להקים ולחבר אותו במהירות במסגרת הזמן שניתנה לי. בנוסף, מאחר ולא נדרש חיפוש טקסטואלי מורכב או דירוג תוצאות MongoDB – ענה במדויק על צרכי הפונקציונליות והפשטות של המסך השני.

הסבר על הרכיבים:

Frontend

- מאוחסן ב־S3 ומוגש דרך CloudFront לצורך CDN מהיר ואבטחת HTTPS.
- את הקוד ניתן לדחוף ל־S3 דרך CodePipeline או GitHub Actions.

API Gateway

- משמש כנקודת כניסה מרכזית לבקשות מה־Frontend אל השירותים השונים (Backend 1 ו־Backend 2).
- מאפשר ניהול גישה, תיעוד, ואבטחה של כל קריאה נכנסת.

Backend 1

- שירות .NET 8 רץ על קונטיינר ECS (Fargate) כדי לא לנהל שרתים בעצמי.
- השירות רץ כמופע יחיד אך תשתית ECS מאפשרת בקלות הוספת מופעים נוספים במידת הצורך.
- מסד הנתונים הוא SQL Server שמתארח ב־Amazon RDS.
- כל הגישה למסד נעשית דרך Entity Framework.

Backend 2

- שירות Node.js עם Express ששומר את פרטי ההזמנה והפריטים הנבחרים.
- השירות ארוז בקונטיינר ורץ על ECS Fargate, עם אפשרות להתרחבות (scaling) בהתאם לעומסים.
- הנתונים נשמרים בפורמט JSON במסד MongoDB בענן (MongoDB Atlas).

- כל השירותים נפרסים תחת VPC סגור עם Security Groups לפי צורך.
- משתמשת ב־IAM Roles בין השירותים.
- ניתוח לוגים וביצועים נעשה דרך CloudWatch.
- ניתן להוסיף התראות ו־Alarms לפי CPU או שגיאות לוגיות.

אוטומציה

- אני מתכננת להשתמש ב־GitHub Actions או CodePipeline כדי לבצע build ו־deploy אוטומטי לקוד.
- אם היה צורך להרחיב את המערכת, – הייתי מוסיפה גם קבצי Terraform לפריסה מלאה של הסביבה.

לסיכום:

המערכת בנויה על עקרונות של scalability, maintainability ואבטחה. אם היה לי זמן או תקציב נוסף, הייתי מוסיפה CD מלא עם בדיקות אוטומטיות בפריסה ו־Infrastructure as Code (Terraform).

שרטוט

השרטוט הבא מציג את מבנה המערכת כפי שנבנתה בענן AWS – החל מהמשתמש, דרך CloudFront, ועד למסדי הנתונים.

כל רכיב בתרשים נבחר מתוך שיקולי יעילות, אבטחה וניהול פשוט, והוא מתבסס על תכנון ידני שלי בהתאם לארכיטקטורה שהגדרתי. כל השירותים הרציפים מתוכננים לרוץ על ECS Fargate, עם יכולת התרבות (scale out) לפי הצורך.

