

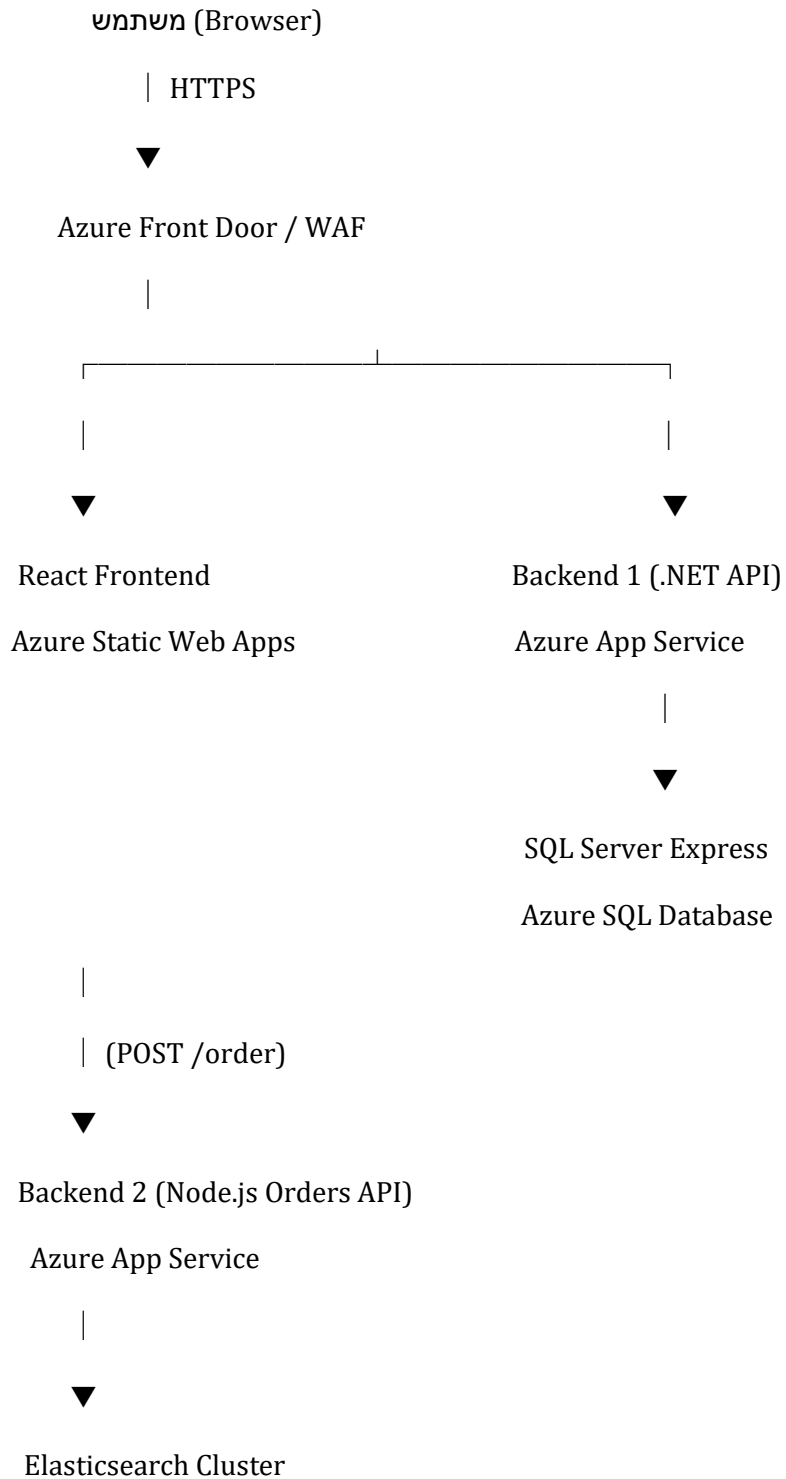
ארכיטקטורת מערכת ודבאופס בענן

בצד לקוח הפרויקט פותח באמצעות React + Redux
לצורך שליפת קטגוריות ומוצרים מול SQL Server מול .NET Web API.
לטיפול בהזמנות ושמירתן בNode/Express ושירות Elasticsearch
. Azure היישום נפרס לסביבת ענן . CI/CD באמצעות תהליכי

ארכיטקטורה ורכיבים מרכזיים

1. Front-End – React + Redux: אחראי להצגת הממשק, הוספה לסל ותקשורת מול השרתים.
2. Shopping API – .NET + SQL Server: שירותי קטגוריות ומוצרים. נתונים מנוהלים ב SSMS.
3. Orders API – Node/Express: שירותי הזמנות ושמירה ל Elasticsearch.
4. Database – SQL Server: בענן.
5. Elasticsearch – שמירת הזמנות וחיפוש מהיר.
6. DevOps CI/CD – GitHub Actions/ Azure DevOps: אוטומטי Build & Deploy לביצוע.
7. Key Vault + Monitor + Insights – שמירת סודות, ניטור ובקרת מערכת.

שרטוט ארכיטקטורה



הסבר על השרטוט

Clients & Routing

המשתמש נכנס דרך הדפדפן ל־URL ציבורי. התעבורה עוברת דרך **Front Door / Application Gateway + WAF**, שמטפל בתעבורה מאובטחת ב־HTTPS, בצמצום מתקפות) כמו / SQL Injection XSS) ובהפניית הבקשות לשירות המתאים.

Front-End – React + Redux

ה־Front-End מתארח כ־Static Web App בכל שינוי קוד מתבצע build אוטומטי דרך pipeline, והתוצאה נפרסת לענן. משם נשלחות קריאות ל־Shopping API ו־Orders API.

Shopping API + SQL Server (SSMS)

שירות ה־NET App Service מטפל בבקשות של "רשימת קטגוריות ומוצרים". הנתונים נשמרים ב־Azure SQL Database (SQL Server) בענן, (שמנוהל ע"י SSMS מבחינת טבלאות, מיגרציות ו־Queries. הגישה ל־SQL מוגבלת לרשת הפנימית, (VNet) וה־Connection String נשמר ב־Key Vault.

Orders API + Elasticsearch

שירות Node/Express מקבל את ההזמנה (פרטי לקוח + סל מוצרים) ושומר אותה ב־Elasticsearch. הוא מאפשר בעתיד יכולות חיפוש ודוחות. גם כאן, הגישה ל־Elasticsearch נעשית מהרשת הפנימית בלבד, ופרטי ההתחברות מנוהלים דרך Key Vault.

אוטומציה (CI/CD)

כל אחד מהשירותים (Front-End, Shopping API, Orders API) מחובר ל־Azure / GitHub Actions DevOps Pipeline על כל שינוי בקוד:

- מתבצע build + בדיקות.
- אם הכול מצליח Deploy – אוטומטי לשירות הרלוונטי. זה מבטיח תהליך מסודר, עקבי וחוזר על עצמו.

ניטור, אבטחה וניהול תצורה

- **Application Insights** אוסף Telemetry זמני תגובה, שגיאות trace, ים.
- **Azure Monitor / Alerts** מאפשרים להגדיר התראות (למשל, הרבה שגיאות 500).
- **Key Vault** מנהל סיסמאות connection strings, ו־API keys ללא קידוד בתוך הקוד.
- קונפיגורציה לפי סביבה (Dev/Prod) מנוהלת כ־App Settings ולא בתוך קובצי source.