

6/6/72

עיצוב ופיתוח מערכות נתוני עתק ומחשוב ענן  
פרויקט סיום תשפ"ב – סמסטר ב'

כללי:

בתקופה הנוכחית, בנוסף לנתונים המופקים ממערכות תפעוליות, מאותרים ונאספים נתונים רבים ממגוון מקורות כבסיס לניתוח, גיבוש תובנות, אימוץ אסטרטגיות וקבלת החלטות מבוססת ראיות.

מערכות Big Data Analytics דורשות משאבים ארגוניים רבים ולכן שימוש בספקי שירותי ענן לאחסון ועיבוד הם פרקטיקה נפוצה. בתכנון מערכת מבוצרת מסוג זה אין מניעה לטוות מגוון שירותים ממספר ספקים, בשילוב שירותים שמקורם בארגון עצמו, לכלל פתרון שלם.

בפרויקט הנוכחי ניצור פתרון לעיבוד ותצוגת נתונים בהשראת תבנית למדה, המנטרת את התעבורה האוירית לישראל דרך נתב"ג ומאפשרת למידת מכונה וגיבוש מודל חיזוי מהנתונים הנצברים - הפתרון ישלב שירותי ענן ושירותי Web לכלל מערכת שלמה.

מהות הפתרון:

מערכת בנויה משלוש תת מערכות שיחדיו מאפשרות ניטור Near Real Time למצב נחיתות והמראות באמצעות Dashboard וכן יצירת מודל חיזוי, עבור כל טיסה, האם תעמוד בזמנים או עיכוב בל"ז.

דרישות פונקציונליות:

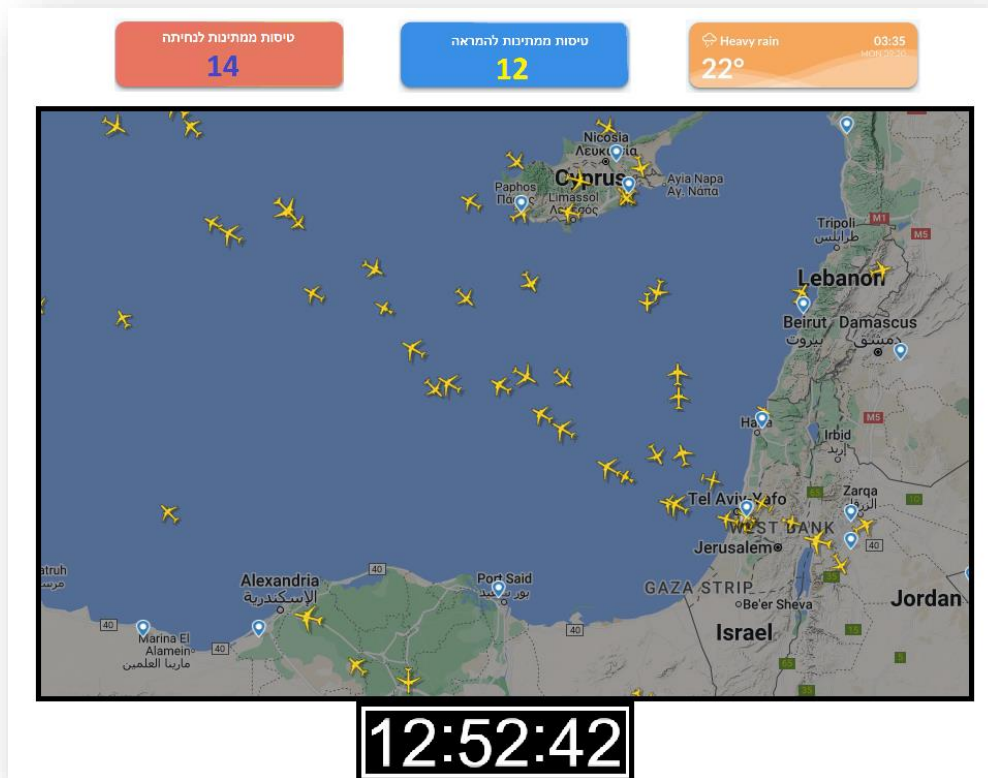
1. המערכת תציג ב- Real Time באמצעות דשבורד מפה ובה מציני מיקום מטוסים בדרכם מישראל או אל ישראל (רק נתב"ג).
2. בדשבורד יופיעו כרטיסיות עם מספר מטוסים האמורים להמריא ולנחות ברבע שעה הקרובה, וכן כרטיסיה המציגה את מזג האויר בשדה.
3. לחיצה על כרטיסיות המטוסים תציג רשימה של המטוסים הרלוונטים וכן **חיזוי** האם טיסה זו תעמוד בזמנים או תאחר.
4. המערכת תאפשר ללמוד מנתוני עבר אודות עמידה בזמנים מתוך הנתונים הבאים:
  - סוג תקופה (חגים, חופשות קיץ, ימים רגילים)
  - חודש בשנה
  - יום בשבוע
  - חברה
  - ארץ מוצא
  - ארץ יעד
  - סוג הטיסה:
    - עד 1500 ק"מ : קצרה
    - עד 3500 ק"מ: בינונית
    - מעל 3500 ק"מ: ארוכה
  - מזג האויר בארץ מוצא
  - מזג האויר בארץ היעד
  - אופי הגעה:
    - מתחת ל-15 דק' : תקין
    - עד שעה : איחור
    - מעל שעה : איחור כבד

דרישות לא פונקציונליות:

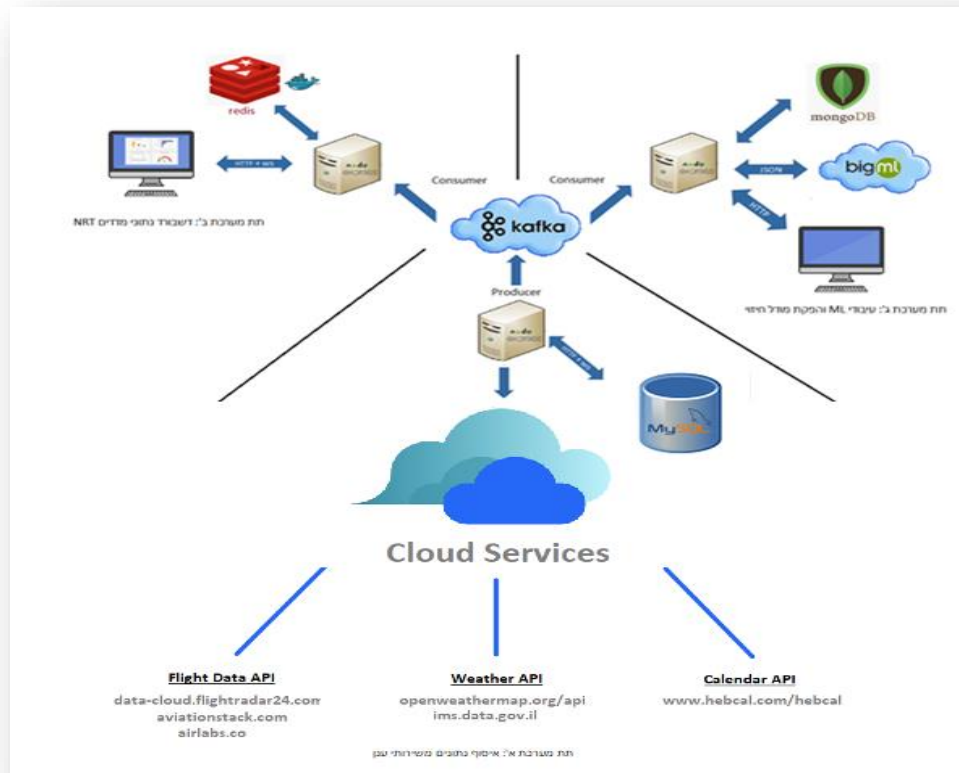
1. המערכת תפותח בתפיסה מונחית שירותים בגישת Micro Services.
2. המערכת תתבסס על שלוש תת מערכות ושירותות Kafka שימש כ- Message Broker לאינטגרציה ביניהם.
3. שלוש תת מערכות הן:
  - א': תת מערכת איסוף, קליטת נתונים וארגון.
  - ב': תת מערכת Dashboards ונתוני זמן אמת.
  - ג': תת מערכת אחסון נתוני טיסות היסטוריים ויצירת מודל חיזוי באמצעות למידת מכונה.
4. תת מערכת א' (איסוף נתונים) תשתמש בחבילת Axiom לאיסוף הנתונים בצורה מחזורית משירותי ענן, תתעד במסד נתונים מסוג MySQL (לוקאלי/Docker) את הגישה לסוג שירות הרבות יום ושעת הגישה.
5. ניתן להשתמש במקורות הבאים לקבלת נתונים:

כתובת	סוג נתונים
<a href="https://www.flightradar24.com/">https://www.flightradar24.com/</a> ALL : <a href="https://data-cloud.flightradar24.com/zones/fcgi/feed.js?faa=1&amp;bounds=41.449%2C21.623%2C16.457%2C53.063&amp;satellite=1&amp;mlat=1&amp;flarm=1&amp;adsb=1&amp;gnd=1&amp;air=1&amp;vehicles=1&amp;estimated=1&amp;maxage=14400&amp;gliders=1&amp;stats=1">https://data-cloud.flightradar24.com/zones/fcgi/feed.js?faa=1&amp;bounds=41.449%2C21.623%2C16.457%2C53.063&amp;satellite=1&amp;mlat=1&amp;flarm=1&amp;adsb=1&amp;gnd=1&amp;air=1&amp;vehicles=1&amp;estimated=1&amp;maxage=14400&amp;gliders=1&amp;stats=1</a> Specific: <a href="https://data-live.flightradar24.com/clickhandler/?version=1.5&amp;flight=2c1c8792">https://data-live.flightradar24.com/clickhandler/?version=1.5&amp;flight=2c1c8792</a>	טיסות RT
<a href="https://aviationstack.com/">https://aviationstack.com/</a> <a href="https://www.npmjs.com/package/flight-data/v/2.0.3">https://www.npmjs.com/package/flight-data/v/2.0.3</a> (supporting package)	טיסות RT
<a href="https://airlabs.co/docs/flights">https://airlabs.co/docs/flights</a>	טיסות RT
<a href="https://openweathermap.org/">https://openweathermap.org/</a>	נתוני מזג אויר
<a href="http://lms.data.gov.il">http://lms.data.gov.il</a>	נתוני מזג אויר
<a href="https://www.hebcal.com/home/219/hebrew-date-converter-rest-api">https://www.hebcal.com/home/219/hebrew-date-converter-rest-api</a>	מועדים וחגי ישראל

6. Kafka מומלץ להריץ בענן ולהשתמש בספק : <https://www.cloudkarafka.com/>
7. Redis מומלץ להריץ במסגרת Docker עם ה-image [https://hub.docker.com/\\_/redis](https://hub.docker.com/_/redis)
8. MongoDB ניתן להריץ לוקאלי, ע"ג Docker Image או משירות ענן כגון : <https://www.mongodb.com/>
9. תת מערכת ב' תשתמש בתשתית Bing Maps להצגת מפות (ראו דוגמא במודל)
10. תת מערכת ג' אוגרת נתוני טיסות היסטוריים ע"ג MongoDB ומשמשת מקור נתונים ליצירת מודל חיזוי.
11. בין השרת לדשבורד (דפדפן) בתת מערכת ב' יעודכנו נתונים באמצעות פרוטוקול Socket.io.
12. ארגון השירותים בסביבת Node.js תתבסס על תבנית MVC (השתמשו ב-express כמרכיב View המחולל תצוגה).
13. תת מערכת ג' תאפשר לנתח את הנתונים ההיסטוריים באמצעות אלגוריתם למידת מכונה וליצור מודל חיזוי **עמידה בזמנים**, על ידי שימוש בשירותי BigML.com וחבילת Node המתאימה. יש להכין תצוגה המאפשרת להגדיר טווח תאריכים ללמידה ויצירת מודל. במידה וקיים קושי ליצור מודל מול שירות BigML ניתן להשתמש בחבילת Node ל ניתוח לוקאלי.



תמונה 1: דשבורד הצגת נתוני טיסות בנתב"ג



דיאגרמה 1: שלושת תת המערכות בפתרון

#### הערות:

1. ההגשה בשלישיות אך ניתן גם בזוגות.
2. נא להרשם להגשה בגיליון הרישום בלינק הזה.
3. מימוש למידת מכונה באמצעות BigML הינה אופציה למגישים בזוגות אך חובה לשלישיה. מי שאינו משתמש ב-BigML יאתר חבילת ניתוחים לוקאלית לבניית מודל חיזוי.
4. מוזמנים להשתמש בדוגמאות הקוד במודל אך יש להתאימם לפתרון שלכם.
5. את הקוד ותאור הפרויקט יש לתעד ב-GitHub.
6. תאריך ההגשה: 4.9 באמצעות זום (יתכן שינוי נא לעקוב בגיליון הרישום).
7. בונוס: פריסת תת מערכת א' ע"ג ספק שירותי אירוח כגון <http://heroku.com> כמתואר כאן: <https://www.geeksforgeeks.org/deploying-node-applications/?ref=rp>

בהצלחה.