

KidsRead.appspot.com



מדריך למתחזק

Deployment

הקדמה: האתר פועל באופן מלא תחת שירותי הענן של Google Cloud.

Project name: KidsRead

Project ID: kidsread

Project number: 35788023348

Google Cloud App Engine

- ביצוע Deployment של האתר ל – App Engine
 - יש להתקין Google Cloud SDK
 - נכנסים לאפליקציה שהותקנה ומנווטים לתיקיית BookReading
 - מעניקים הרשאות גישה בעזרת הפקודה:
`gcloud auth login`
- עבור החשבון kidsread.appspot@gmail.com (סיסמא מופיעה בקובץ app.standard.yaml תחת המשתנה GMAILPW)
 - מריצים את הפקודה `gcloud app deploy app.standard.yaml`
 - בסיום התהליך השינויים יעלו לענן
 - ניתן להכנס לאתר בכתובת: <https://kidsread.appspot.com>
 - מידע נוסף ניתן למצוא ב <https://cloud.google.com/appengine/docs/standard/nodejs/building-app/deploying-web-service>
- ניתן לראות logs של האתר בעזרת הפקודה `gcloud app logs tail -s default` באפליקציית Google Cloud SDK

Google Cloud SQL

קיים instance של PostgreSQL בענן:

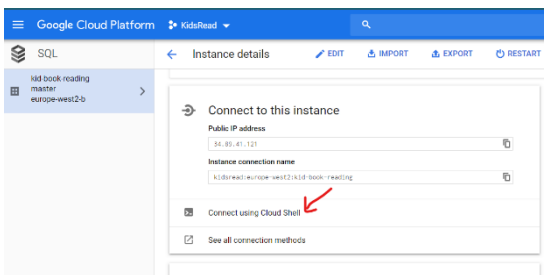
Instance id: kid-book-reading-db

Location: europe-west2-b

Public IP address: 34.89.41.121

Instance connection name: kidsread:europe-west2:kid-book-reading

- כיצד להתחבר ל DB באמצעות Cloud Shell
- נכנסים Google Cloud SQL ולוחצים :



- מתחברים לשרת באמצעות הפקודה :

```
gcloud sql connect kid-book-reading --user=postgres --quiet
```

מכניסים סיסמא עבור המשתמש postgres (מופיע בקובץ app.standard.yaml תחת המשתנה SQL_PASSWORD)
מתחברים לבסיס הנתונים באמצעות הפקודה:

```
\c kidBookReadingDB postgres
```

מכניסים סיסמא עבור המשתמש postgres (מופיע בקובץ app.standard.yaml תחת המשתנה SQL_PASSWORD)
כעת נוצר חיבור לבסיס הנתונים וניתן להריץ פקודות של PostgreSQL.

- מבנה ה DDL שאיתו נבנה בסיס הנתונים נמצא בתיקיית docs : kidBookReadingDDL.sql
על מנת לבנות בסיס נתונים חדש ניתן להשתמש בכל הפקודות שבקובץ זה. (יש לשים לב שאם היה מידע אמיתי של משתמשי האתר אזי מידע זה ימחק)

- אתחול ראשוני שהוכנס לבסיס הנתונים נמצא בתיקיית docs : initializeDB.sql
ניתן להשתמש בקובץ זה כדי להכניס מידע ראשוני לבסיס הנתונים. (יש לשים לב שאם היה מידע אמיתי של משתמשי האתר אזי מידע זה ימחק)

- על מנת לגבות את הבסיס הנתונים שקיים בענן ניתן להעזר במדריך זה :
<https://cloud.google.com/sql/docs/postgres/import-export/exporting>

Google Cloud Storage

קיים bucket בשם kidsread.

קבצים שמועלים באופן דינמי באתר מועלים ל bucket זה בעזרת הפונקציה: `uploadToCloud` אשר נמצאת ב: `src\queries.js`.

מידע נוסף על אופן העלאת הקבצים ניתן למצוא כאן:

<https://cloud.google.com/appengine/docs/standard/nodejs/using-cloud-storage>

פיתוח ותחזוקה

סביבת הפיתוח בצד השרת היא Node.js. שרת ה web שנבחר הינו express.

ה-View engine של הדפים הדינמיים הוא hbs, וכל הדפים הדינמיים בעלי סיומת זו (.hbs) נמצאים בתיקיית templates.

על מנת לשמור על סדר ביצענו חלוקה לקבצים נפרדים עבור כל קבצי js וקבצי ה css לרוב הדפים הדינמיים.

כך למשל עבור דף הפרופיל של הילד

`\templates\views\kid\profile.hbs`

כל הסקריפטים נמצאים ב

`\public\js\kid\kid-profile-scripts.js`

וכל עיצובי ה css נמצאים ב

`\public\css\kid\profile.css`

פירוט חבילות npm תחת dependencies

- **google-cloud/storage@** – חבילה עזר לשימוש ב Google Cloud Storage.
- **bcrypt** – חבילה להצפנת סיסמאות על מנת לשמור סיסמאות בבסיס המידע בצורה מאובטחת.
- **body-parser** – חבילה שמאפשרת ל express לקבל ולפענח פרמטרים בפורמט JSON שנשלחים ב Ajax מהדפדפן, באמצעות XMLHttpRequest . ניתן לגשת לפרמטרים תחת req.body
- **connect-flash** – מאפשר שליחת הודעות פלאש זמניות למשתמש בתוך Session, אשר נעלמות לאחר הצגתן.
- **connect-pg-simple** – חבילה שנועדה לניהול אחסון PostgreSQL session.
- **cookie-parser** – חבילה לניהול cookies תחת האובייקט req.cookies
- **dotenv** – חבילה שמשמשת לטעינת משתני סביבה מתוך קובץ .env. לתוך האובייקט process.env. מאפשר להגן על סיסמאות ופרטים סודיים אחרים וגם מאפשר שימוש במשתנים שייחודיים לכל משתמש.
- **Express** – תשתית המעניקה כלים לפיתוח אפליקציות Web. מאפשרת להגדיר middlewares עבור בקשות HTTP. מגדירה טבלת ניתוב לטיפול בנתיבים שונים. מאפשרת לרנדר דפי HTML דינמיים על בסיס של העברת ארגומנטים לדפי התבנית.
- בעזרת חבילה זו ניתן להגדיר שרת Web שמאזין לבקשות HTTP.
- **express-promise-router** – חבילה שמאפשרת לארגן נתיבים של Express בצורה נוחה. מאפשרת להגדיר נתיבים בקובץ נפרד.
- **express-session** – חבילה שמאפשרת לטפל בסשנים ובעוגיות.
- **fs-extra** – חבילה שמאפשרת טיפול בקבצים מקומיים, כגון העברה ממקום למקום.
- **handlebars-helpers** – חבילה שמכילה פונקציות עזר רבות עבור handlebars.
- **Hbs** – חבילה שמהווה view engine של handlebars עבור Express. מאפשרת לרנדר דפים דינמיים.
- **Multer** – חבילה שמאפשרת העלאת קבצים מתוך טפסי HTML.
- **multi-lang** – חבילה שמאפשרת תרגום לשפות שונות. משמשת תרגום בצד השרת בפרוייקט שלנו. הקובץ שמכיל את התרגומים השונים בצד השרת src\language-server.json

- **node-gyp** – חבילה שמאפשרת ניהול גירסאות של חבילות אחרות. הכרחית עבור חבילה אחרת מהרשימה.
- **node-sass** – חבילה שמאפשרת לקמפל קבצי scss.
- **Nodemailer** – חבילה שמאפשרת שליחת אימיילים.
- **Passport** – חבילה שמשמשת עבור אימות משתמשים שמחוברים לאתר.
- **passport-local** – חבילה שמאפשרת אימות משתמשים באמצעות שם משתמש וסיסמה.
- **Path** – חבילה שמעניקה כלים לעבודה עם קבצים ותיקיות.
- **Pg** – לקוח של PostgreSQL.
- **Validator** – חבילת עזר לטיפול במחרוזות.

פירוט חבילות npm תחת devDependencies

- **Nodemon** – חבילה שמאפשרת פיתוח נוח ע"י אתחול מחדש של node בכל פעם שקובץ שנמצא במעקב מתעדכן.
- **npm-check** – מאבחן dependencies של חבילות ישנות או שלא תקינות.

הפרדה בין production לבין development

על מנת להפריד בין קוד שרץ במצב פיתוח (development) לקוד שרץ באתר (production)

אנחנו עושים שימוש בקבוע שמוגדר בקובץ .env. תחת השם .NODE_ENV.

ערכו של משתנה זה במצב פיתוח הוא NODE_ENV=DEVELOPMENT, לעומת בענן מוגדר אוטומטית המשתנה להיות NODE_ENV=PRODUCTION.

באופן זה אנו מפרידים בין שני המצבים.

פיתוח מקומי יחד עם גישה לבסיס נתונים שבענן

על מנת להקל על הפיתוח והתחזוקה ניתן לגשת ישירות למידע שנמצא בבסיס הנתונים שבענן באופן הבא :

- משנים את פרטי הבסיס נתונים בקובץ .env. כך שיתאימו לפרטי בסיס הנתונים שבענן ולא בסיס נתונים מקומי:

יש לשנות את המשתנים: SQL_USER, SQL_PASSWORD, SQL_DATABASE לערכים זהים לערכים שבקובץ app.standard.yaml

את המשתנה SQL_PORT יש לשנות ל 5433 (בשונה מהפורט הרגיל 5432)

- פותחים Windows Powershell

- מנווטים אל תיקיית docs שבה נמצא הקובץ cloud_sql_proxy.exe

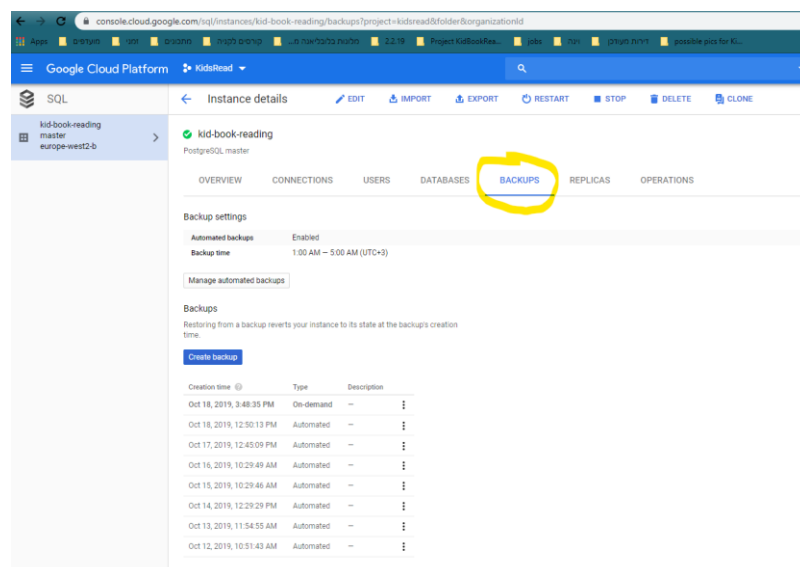
- מריצים את הפקודה:

```
./cloud_sql_proxy.exe -instances=kidsread:eu-west-2:kid-book-reading=tcp:5433
```

- כעת ניתן להריץ את הפיתוח המקומי בעזרת הפקודה npm run dev, כך מתקבלת גישה ישירה לבסיס הנתונים שבענן בשילוב עם פיתוח מקומי.

גיבוי בסיס הנתונים ב- Google Cloud SQL

לבסיס המידע מתבצע גיבוי אוטומטי באופן יומי בין השעות 1:00 – 5:00.



The screenshot shows the Google Cloud Platform console for a SQL instance named 'kid-book-reading'. The 'BACKUPS' tab is selected, showing backup settings and a list of backups.

Backup settings

- Automated backups: Enabled
- Backup time: 1:00 AM – 5:00 AM (UTC+3)

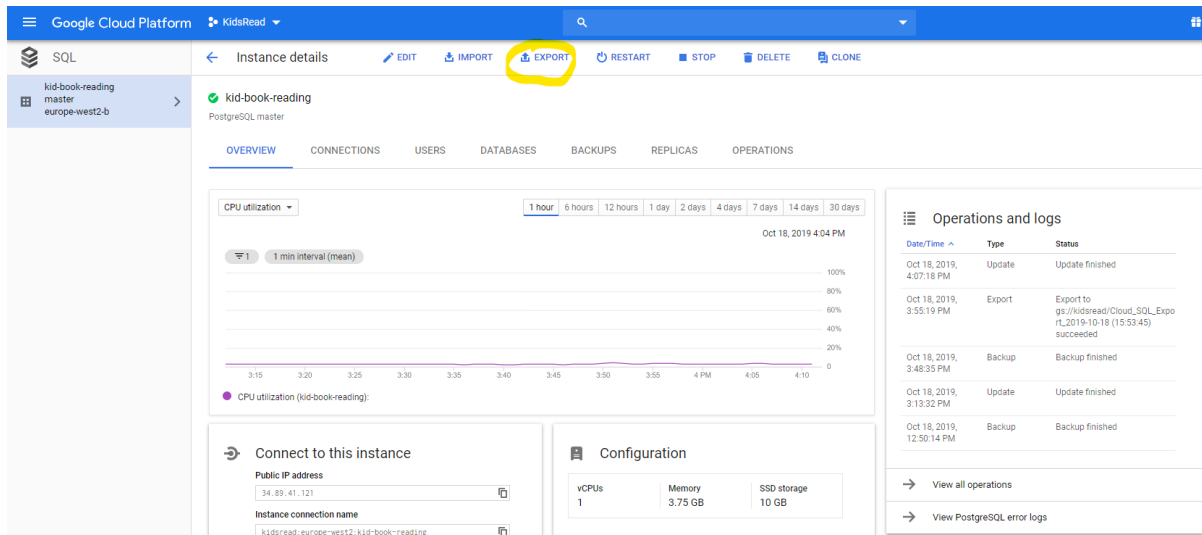
Backups

Restoring from a backup reverts your instance to its state at the backup's creation time.

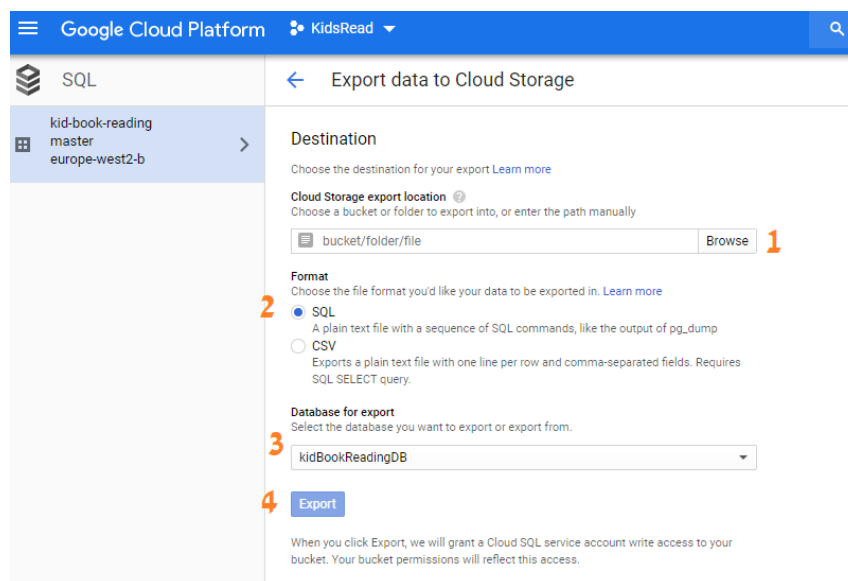
Creation time (⌚)	Type	Description
Oct 18, 2019, 3:48:35 PM	On-demand	–
Oct 18, 2019, 12:50:13 PM	Automated	–
Oct 17, 2019, 12:45:09 PM	Automated	–
Oct 16, 2019, 10:29:49 AM	Automated	–
Oct 15, 2019, 10:29:46 AM	Automated	–
Oct 14, 2019, 12:29:29 PM	Automated	–
Oct 13, 2019, 11:54:55 AM	Automated	–
Oct 12, 2019, 10:51:43 AM	Automated	–

ניתן לייצא קובץ גיבוי של בסיס הנתונים ע"י Export באופן הבא :

- לוחצים על Export :



1. בוחרים לאיזה bucket הקובץ יעלה.
2. בוחרים פורמט. SQL מייצא רצף פקודות מלא עבור הבסיס נתונים.
3. בוחרים את בסיס הנתונים.
4. לוחצים Export.



- לאחר מכן ניתן להוריד את הקובץ שנוצר מהמיקום שנבחר.
- למידע נוסף על גיבוי:

<https://cloud.google.com/sql/docs/postgres/backup-recovery/backups>



יצירת קשר

בכל בעיה ניתן ליצור קשר עם המפתחים באימייל :

יבגני קצ'ין - ykechin@campus.haifa.ac.il

מראם חטיב - Khatibmaram11@gmail.com