**שלום לכולם.**
**המפגש הבא שלנו יהיה ביום חמישי בעוד שבוע וחצי, אחרי החג.**
**כמו שאמרנו אתמול, יש לנו משימה לחופשה.**

**המשימה מייצגת את התחלת הפרוייקט שילווה אותנו עד סוף הקורס. אפשר, מי שמעוניינ\ת, לעבוד בזוגות על הפרוייקט, מעכשיו ועד הסוף. אפשר גם לעבוד ביחידים.**

**המשימה מצורפת בתור מסמך PDF.**
**המשימה מערבת שימוש בכל הכלים שנגענו והשתמשנו בהם עד עכשיו – חלקם בצורה מעמיקה יותר וחלקם פחות.**

**הרעיון הוא לקחת רעיון לאפליקציה קטנה כלשהי, לחלק אותו לשלוש (או יותר) חלקים, ולהריץ אותו בסביבת הענן שלנו, תוך שימוש בכל השירותים שלמדנו עליהם.**

**המשימה אמורה לאתגר. לא הכל למדנו לפרטים, וחלק מהקורס שלנו הוא לפתח יכולות לימוד עצמי בנושאי DevOps. השימוש ב AI מומלץ ואף נדרש.**

**אופן ההגשה של המשימה מוגדר במסמך. המשימה היא ללא ציון, אך חובה על מנת להתחיל לבנות את פרוייקט הסיום של הקורס.**

**אני תמיד זמין לשאלות או הערות, ואם מעוניינים, אפנה גם זמן כדי לתמוך בכם בביצוע.**

**משימת בית**

**מטרה**

מטרת המשימה היא לבדוק יכולת תכנון, הקמה ותפעול של סביבת AWS בסיסית עבור אפליקציה פשוטה במבנה של כמה שירותים נפרדים.

**תיאור המשימה**

יש להקים אפליקציה פשוטה המורכבת מ־3 שירותים נפרדים, כאשר כל שירות רץ על מכונת **EC2** נפרדת.

המערכת צריכה לכלול גם שימוש בשירותי AWS נוספים לצורך אחסון, חשיפה חיצונית, והתראות.

**דרישות**

1. **אפליקציה עם 3 שירותים**

יש לפתח או להרים אפליקציה פשוטה המורכבת מ־3 שירותים נפרדים, לדוגמה:

- שירות Frontend – שמבוסס על nginx
- שירות Backend
- שירות Worker / Service נוסף לפי בחירתכם

דרישות:

- כל שירות ירוץ על **EC2 instance נפרד**
- יש להגדיר תקשורת **תקינה** בין השירותים
- יש לתעד את תפקידו של כל שירות

2. **שימוש ב־RDS**

יש להקים מסד נתונים מסוג **RDS** ולשמור בו את ה־ dataשל האפליקציה.

דרישות:

- בחירת מנוע DB לפי העדפה MySQL / PostgreSQL :
- יצירת טבלה/ות בסיסיות עבור נתוני האפליקציה
- האפליקציה צריכה לבצע לפחות פעולת כתיבה אחת ופעולת קריאה אחת מול ה־ DB

3. **שמירת קובץ ב־S3**

יש להוסיף יכולת לשמור קובץ ל **S3 Bucket**

דרישות:

- העלאת קובץ דרך אחד השירותים ל S3
- תיעוד מבנה ה־ Bucket והרשאות הגישה שהוגדרו

4. **התקנת nginx וחשיפה חיצונית**

יש להתקין **nginx** על אחד השרתים ולחשוף דרכו שירות אחד כלפי חוץ.

דרישות:

- השירות יהיה נגיש חיצונית דרך HTTP

5. **שימוש ב־SNS**

יש להשתמש ב־ **SNS** כדי לשלוח הודעת מייל על שינויים באפליקציה.

דוגמאות אפשריות:

- העלאת קובץ חדש ל־ S3
- יצירת רשומה חדשה ב־ DB
- שינוי סטטוס באפליקציה

דרישות:

- יצירת SNS Topic
- Subscription למייל
- שליחת הודעה בפועל בעקבות אירוע באפליקציה

---

**תוצרים נדרשים**

1. **קוד מקור**

יש להגיש את קוד המקור של האפליקציה, כולל:

- 3 השירותים ומה הם עושים

2. **תיעוד**

יש לצרף קובץ README מסודר הכולל:

- הסבר קצר על הארכיטקטורה של האפליקציה
- פירוט על כל רכיב במערכת
- הוראות הקמה והרצה
- הסבר על החיבורים בין השירותים
- פירוט על השימוש ב־ RDS, S3, nginx־ו SNS כחלק מהאפקליקציה

3. **תרשים ארכיטקטורה**

יש לצרף תרשים שמציג:

- מבנה הרשת בענן
- שלושת שרתי ה־ EC2 ובאילו רשתות הם רצים
- RDS
- S3
- SNS
- nginx
- כיוון זרימת התקשורת בין הרכיבים

4. **צילומי מסך** של השירותים השונים ב AWS, בזמן הריצה של האפליקציה.

---

**דגשים טכניים**

- ניתן להשתמש בכל שפת תכנות מועדפת
- דגש על פשטות, סדר, ותיעוד ברור
- חשוב להקפיד על Security Groups והרשאות IAM מינימליות
  - אם נאלצתם לפתוח גישה או הרשאות מעבר למינימום, הסבירו מדוע.

---

**על מה לשים דגש ?**

- נכונות טכנית
- ארכיטקטורה ברורה

- עבודה תקינה של כל הרכיבים
- תיעוד מסודר
- שימוש נכון בשירותי AWS
- הבנה בסיסית של networking, permissions ו־ troubleshooting

---

**אופן ההגשה:**

- שליחה במייל מסודר עם כל התוצרים ל Aviad.diamant@gmail.com
- הצגה אופציונאלית בשיעור

**בונוס**

אפשר להוסיף אחד או יותר מהבאים:

- שימוש ב־ CloudWatch logs
- הוספת HTTPS ע"י שימוש ב certificate
- הוספת health check בין השירותים

---

**לוח זמנים**

זמן מומלץ לביצוע **7 – 5 שעות.**