

## תרגיל 1

מגישים: אורי הלף 315529545, עידו סגל 209371178.

- הרצת הפרויקט: על מנת להריץ את הפרויקט נדרש לעדכן את credentials במערכת ההפעלה, בשביל להתחבר למעבדה. לאחר מכן יש להיכנס לתיקיית הפרויקט, ולהריץ את שורת ההרצה כפי שמופיעה בקובץ התרגיל.
- איך הפרויקט עובד: לפרויקט יש שלושה חלקים.
  - חלק ראשון, החלק הלוקאלי רץ במחשב של הלקוח, תפקידו לקחת את קובץ הקלט, לעלות אותו ל AWS S3, ולאחר מכן שולח הודעה ב AWS SQS עם הכתובת בו נמצא הקובץ. לאחר מכן מאזין SQS על הודעה של סיום העבודה, בשלב זה מוריד קובץ html הגמור ושומר אותו לוקאלי. תפקידו לייצר Bucket עבור הפרויקט, ערוץ SQS פרטי בו מאזין לתשובה מה manager, ואם manager לא קיים, ייצור גם manager וערוץ SQS המשמש את כלל הלקוחות בשליחת הודעות ל manager.
  - חלק שני, manager. תפקידו להאזין להודעות מלקוחות, להקים workers במידת הצורך. כאשר מתקבלת הודעה, manager יפרק את קובץ הקלט כפי שמופיע ב S3 לתתי משימות ויעביר הודעות לתור SQS אל Workers.
  - בנוסף יאזין manager להודעות חזרה מה Workers בהם מופיע הפענוח. ה manager יעקוב אחר ההודעות, כאשר כל ההודעות פוענחו, ייצור את קובץ html ויעביר חזרה ל S3 וישלח הודעה ל local.
  - החלק האחרון, Workers. חלק זה אחראי לקבל הודעה ב SQS, ההודעה מכילה קישור לתמונה, Workers מפענח את התמונה ומשיב חזרה ל Manager את ההודעה המפוענחת.
- השתמשנו בשרתי t2.micro, אליהם טענו Custom AMI אשר ביססנו על Amazon Linux 2. אמינ שיצרנו מכילים את קבצי ההרצה. זמן הריצה בריצה הראשונית, כולל הקמת הסביבה, הוא כ-3 דקות עם n=3. בריצה שבה הסביבה כבר רצה, זמן עד סיום התכנית הוא כ-50 שניות.
- על צרכי אבטחה ענינו בכך שחיברנו את השרתים בעזרת פרופיל IAM ולכן הסיסמאות לא עוברות בטווח.
- חשבנו רבות על סקאלביליות, בעולם האמיתי לא היינו מכניסים חסם על מספר ה Workers והיינו מקצים שרת חזק יותר ל Manager שיוכל לעמוד במספר רב של בקשות.
- במקרה ש Worker נופל, ישנו Thread ב Manager שדוגם את מספר ה Workers הפעילים ודואג להקים חדשים אם המספר נמוך מהמצופה. בנוסף, במקרה של נפילת תהליך java בצורה פנימית ב Worker, ישנו CronJob שמעלה את התהליך מחדש בכל דקה עגולה.
- לכל Worker יש 30 שניות להשלים את הפענוח, במידה ו Worker קפא, Worker אחר ייקח את המשימה לעצמו. 30 שניות הוא מספר שרירותי ובמידה ונראה כי חלק מהפענוחים לוקחים יותר זמן, כלומר Worker לא קפא, אלא עדיין חישב, נגדיל את הזמן שניתן לכל Worker.
- ה Manager הוא multithreaded, בחרנו לעשות זאת במטרה שאם נכנס פעולה כבדה, למשל קובץ שמכיל הרבה שורות URL אז תהליך הפירוק להודעות לא יפגע בהאזנה ל Workers, לא יפגע בתהליך של שליחת קובץ html גמור ולא יפגע בתהליך ניטור על מספר ה Workers הפעילים.
- הפרויקט עובד גם עם מספר רב של לקוחות, והתשובות התאימו למצופה.
- בסיום, כל לקוח מוחק את המשאבים שייחודיים אליו, SQS Bucket ששייך לו, בעת קבלת termination ולאחר סיום כל הפרויקטים הקיימים, ה manager מוחק את שאר התורים, מכבה את ה Workers ולאחר מכן מכבה את עצמו.
- אנחנו יוצרים שרתים בהתאם למגבלה שאותה ניתן ליצור, ואילו לא הייתה מגבלה, לא היינו מגבילים כך.
- בארכיטקטורה שלנו אין Workers לא פעילים, כל ה Workers מאזינים לאותו תור, וברגע שיש הודעה אחד ה Workers לוקח לעצמו את הפענוח. שאר ה Workers ממשיכים להאזין להודעות נוספות.