**Cloud Computing assignment #2**

**Itay Cohen & Noam Heller**

בעולם האמיתי, אלו הם מצבי הכשל אשר המערכת יכולה להיתקל בהם:

* **עומס יתר:** בעקבות פניות רבות, עלול להיווצר עומס שחורג מהיכולת של המערכת, מה שיוביל לירידה בביצועים או חוסר זמינות של השירות.

כדי להתמודד עם זה, אפשר לנטר את העומס ולעקוב אחרי ניצול המשאבים כדי לזהות עליות בעומס או דפוסים חריגים. אפשר להטמיע מנגנוני מניעת עומס כדי להגביל את מספר הבקשות במקביל, ולאזן על-ידי פיזור עומס על פני מספר מופעים והרחבת משאבים באופן דינמי. בנוסף, להגדיר התראות אם העומס מגיע לסף.

* **פיצול רשת:** אם הרשת מפוצלת, התור לא יוכל לתקשר עם ה- workers. כלומר, תקשורת הרשת בין EC2 instances עלולה להתפצל, וכתוצאה מכך המופעים לא יוכלו לתקשר זה עם זה.

כדי למנוע זאת, אפשר להשתמש בשירותים קיימים כדי ליצור שירות DNS זמין במיוחד. זה יבטיח שהתור תמיד יוכל לתקשר עם ה-workers, גם אם הרשת מפוצלת. כדי להתאושש ממצב כזה, צריך להמתין לשחזור הרשת. לאחר שחזור הרשת, התור יתחיל שוב לתקשר עם ה- workers באופן אוטומטי.

* **מהירות:** במקרה הזה, כאשר מגיעה בקשה חדשה שבעקבותיה יש צורך בהקמת instance חדש, המערכת ממתינה שה-instance שיעלה ורק אז ממשיכה לטפל בבקשות נוספות. דבר זה לוקח זמן ובעולם האמיתי ב- scale גדול יותר הוא לא יעיל.

אפשר להתמודד עם זה ע״י יצירת מופעים ברקע ובמקביל להמשיך לטפל בבקשות חדשות, במקום לחכות לסיום יצירת כל מופע.

* **פרצות אבטחה:** המערכת עלולה להיות פגיעה לאיומי אבטחה, כולל גישה לא מורשית, פרצות נתונים או התקפות זדוניות.

דרכי התמודדות:

* הטמעת מנגנוני אימות כדי להבטיח שרק משתמשים מורשים יוכלו לגשת למערכת.
* הטמעת מנגנוני הצפנה כדי להגן על נתונים רגישים.
* ביקורות אבטחה קבועות והערכות פגיעות כדי לזהות ולהפחית סיכוני אבטחה פוטנציאליים.
* הטמעת מערכות זיהוי ומניעת פריצות לניטור ותגובה לאירועי אבטחה.
* **אובדן נתונים:** נתונים שנשלחו או שנוצרו במהלך העיבוד עלולים ללכת לאיבוד עקב כשלי חומרה, באגים בתוכנה או בעיות אחרות.

על מנת להתמודד עם אובדן נתונים, צריך לגבות את המידע באופן שוטף כדי שבמקרה של תקלה, יהיה מאיפה לשחזר את המידע. בנוסף, יש לבדוק באופן קבוע תהליכי שחזור נתונים ולוודא שהגיבויים מאוחסנים בצורה מאובטחת.

* **כשל ב- dependency:** המערכת מסתמכת על תלות חיצונית כמו שרתי מסד נתונים, ממשקי API של צד שלישי או שירותי ענן, שעלולים לחוות כשלים או שיבושים, מה שישפיע כמובן על התנהגות המערכת.

כדי למנוע זאת, יש לבצע בדיקות תקינות ומעקב אחר תלות חיצונית כדי לזהות כשלים ולהגיב אליהם באופן מיידי. בנוסף, צריך להטמיע מנגנוני יתירות או כשל עבור dependency קריטי כדי להבטיח את זמינות מערכת.

* **כשל ב- instance:** מופע EC2 אחד או יותר עלול להיכשל עקב בעיות חומרה, שגיאות תוכנה, עומס או נסיבות בלתי צפויות אחרות. אם מופע נכשל, העבודה שהייתה פועלת על אותה מכונה תיאבד.

ישנן מספר דרכי התמודדות, ביניהן יישום בדיקות תקינות של מופע כדי לזהות כשלים במופע. ניתן להשתמש במנגנון שחזור אוטומטי כדי להחליף אוטומטית מופעים שנכשלו. בנוסף, אפשר להטמיע מערכת ניטור כדי לקבל הודעות על תקינות המופעים. לבסוף, כדאי לגבות נתונים חשובים כדי שבמקרה שבכל זאת יש כשל במופע, יהיה ניתן לשחזר אותם ולמזער נזקים.