בס"ד

Splunk

הסביבות יהיו יומיים

גרסת splunk עבור ההדרכה: 9.0

SPL - Search Procesing Language - פורמט פקודה:

חלק ראשון חייב להיות חיפוש, אחריו המידע עובר ב-pipeline. המידע שעובר ב-pieline לא מכיר את ההיסטוריה, אז למשל אם שונה שם עמודה – פקודות לאחר מכן לא יכירו את שם העמודה הקודם.

סוגי פקודות ספלאנק:

* Streaming – פקודה הפועלת על **כל** event, ורץ בנפרד על כל פקודה
* Transforming – פקודה המבוצעת על כלל ה-Data (כמו sort/deduplicate), ומחכה לככל הלוגים על מנת להתבצע.

מכיוון שכך, נעדיף לבצע את כלל הפקודות על ה-streaming, כי ברגע שמבצעים פקודת transforming – כל שאר הפקודות ישתמשו ב-search head ולא ב-indexer-ים השונים. מכיוון לשרוב יש מספר indexer-ים - מהירות העיבוד גדולה בהרבה דרכם.

# המלצות שימוש כללי ב-splunk:

* לא להשתמש ב-\*, תמיד להשתמש באינדקס שידייק מראש את החיפוש, ואף לחסום את האפשרות למצוא ב-\*.
* לא להמשתמש ב-all time בחיפוש, תמיד לצמצם זמן ( לאלא אם כן אין ברירה, כמו לוגים שלא מופיעים וכד')
* לקבוע מראש איזה שדות אנחנו רוצים – חוסך בתעבורה אם קיימים ב-AWS למשל.
* הוספת דוקומנטציה – פתיחה וסגירה של מקטע דוקומנטציה ע"י ```.
* סוגי חיפוש: יש 3 סוגים חיפוש:
  + מהיר – מציג רק שדות ספציפיים
  + Verbose – מציג את כל המידע
  + חכם – בוחר עבורך את סוג החיפוש בהתאם לשאילתה
* Job – splunk מאפשר ללראות את רכיבי השאילתה, ע"י שימוש ב-inspect job, מראה את כל רכיבי החיפוש, כמה פריטים עובדו וכמה זמן זה לקח. מעולה עבור debug. שם ניתן להיכנס ללוג המלא (search log), ולחפש את הנקודה בה אנחנו נתקלים, ע"י חיפוש השורה שמכילה את המילה Lispy.

# Brakers ופרסור המידע

על מנת לבצע עיבוד מידע מהיר, ה-splunk מבצע פרסור מסוג masur (כדוגמת רווח, שורה...) ו-minor (כדוגמת נקודה, שטרודל...). טיפ לזיהוי minor/masur braker: כשהעכבר נמצא על טקסט ה-splunk מדגיש את הטקסט, והוא יתחיל מה-masur brakers ועד ל-minor brakers.   
הדרך שבה splunk עובד היא ע"י שמירת המידע הגולמי כ-raw data מחולק לשורות לפי mazur brakers, ובנוסף לקיחת לקסיקון של כלל הביטויים המופיעים ב-data מחולקים לפי minor brakers, עם מצביעים לאיזו שורה מה-raw event המידע הזה נלקח.

# ביצוע חיפושים יעילים

Term - ישנם חיפושים שעוברים פרסור ע"י מנוע החיפוש, ולכן אם אני מחפש IP (לדוג' 127.0.0.1) יהפוך לרשימות חיפוש לפי minor braker (0 0 1 127 (. Term מתעלם מ-minor braker בתנאי שאחריו יש masur braker במידע שכתוב בלוגים עצמם.

Tstats – בדומה ל-stats הוא נותן סיכום לנתונים, אבל יותר יעילה (עד כמעט פי 20) מכיוון שהיא רצה על הלקסיקון (רץ ב- streaming?), לעומת stats שרצה על ה-buckets (המידע הגולמי עצמו). מכיוון שכך, ה-tstats עובד על שדות index\_fields, שהם כברירת מחדל host, source, source\_type (ניתן להוסיף עוד שדות לשם, אבל צריך לשים לב שזה שווה את זה). על מנת לעקוף מגבלה זאת, ניתן להשתמש ב-prefix.

Prefix – כל הנתונים חייבים להיות ב-lowerCase, מה שיקרה הוא שהמידע שיועבר הוא החוצה (למשל עבור פונ' ה-sum) זה מה שיגיע מיד אחרי ה-prefix.

Eval – יצירת משתנים תוך כדי השאילתה. הפורמט: eval x=condition(yyyy). התנאי יכול להיות: If - אם הלוג מכיל A – שים B, אחרת – שים C. Case - תנאי מורכב, בו התנאי הראשון שמתרחש יקבע את הערך (ניתן להוסיף תנאי אחרון בסגנון true()). פונקציה מתמטית, עיבוד השדה ועוד.