1. **AirFlow** - תהליך לילי שמוריד את הקבצי CSV למכונה ועושה להם UnZip:
   1. מוחק את התיקיות מאתמול ויוצר חדשות.
   2. מוריד את הקבצים.
   3. עושה unzip .
   4. מסדר את הקבצים הרלוונטים בתתיקייה אחת ואת השאר מוחק.
   5. **בPandas** הופך את כל הקבצי הCSV ל DataFrame.
   6. מסנן את כל הנתונים לגוש דן בלבד לפי קובץ Zone.
   7. יוצר DataFrame אחד שמאחד את כל הנתונים על כל Trip.
   8. שומר את הנתונים **לMongo** כJson.
      1. ניתן גם לבחון DynamoDB ו AWS DocumentDB
2. **Generator** - מתחבר ל API של משרד התחבורה ומביאי נתונים בזמן אמת:
   1. שולף מהMongo את המספ קווים הקיימים ואת הTrip שלהם.
   2. כל 15 שניות:
      1. מתחבר לAPI ומחפש מידע על המיקומי אוטובוסים בקווים הנבחרים.
      2. אם האוטובוס כרגע בתחנה:מוסיפים מידע רנדומאלי כמות נוסעים שעלתה/ירדה בכל תחנה.
      3. שולח את הנתונים לkinesis.
3. **Kinesis** – צינור מעבר מידע:
   1. שומרים את כל הנתונים שמתקבלים בArchive ב S3 כקבצי parquetע"י **kinesis firehose**
      1. שומר בpartitions לפי שעה.
   2. **kinesis data analytics** – סינון המידע בStream ביצירת Topics חדשים
      1. topic חדש שבודק עם האוטובוס מגיע ל90% תפוסה ומתריע להוסיף עוד אוטובוס

ושמירת המידע מתי האוטובוס נגיע לכל התחנה מול השעה המתוכננת + מספר הנוסעים עליו כל בתחנה.

שומר בpartition לפי מספר קו ב S3 כקבצי parquetע"י **kinesis firehose**

* + - 1. שומרים את הנתונים **HBase** את הנתונים בזמן אמת.
    1. Topic חדש שבודק עם האוטובוס כבר מספר 4 תחנות עומד על 0 נוסעים כדי לבטל אוטובוס נוסף או שלחת מיניבוס בקו הבא – שליחת התראה.

1. **aws glue** - מבצע crawler כל שעה בשביל לדעכן את הpartition החדשים שנוצרים ץ
2. **Athena** - מפיקה דוחות קבצים בS3 לבדוק זמן הגעה בפועל עבור כל קו וסטטיסטיקות עבור כל מפעיל.
3. **Tableau** – מציג מידע:
   1. מידע של הדוחות.
   2. מפה של כל המסלולים ואיפה נמצא כל אוטובוס בזמן אמת מתוך הHBase.
4. להוסיף:
   1. Spark
   2. ELK