SOUTENANCE INDE2

Emon Barberis, Paul Hartmann, Nicolas Schmitt

SUMMARY

Python — Log alerts →

Kafka Connect

Kafka - "alerts"

Simulated with Scala

Harmonywatchers

Kafka - "reports"

Filter alerts

Read everything → Spark — Write → S3 — Read every 6h → Spark

Write stats

Grafana — Read stats → Postgres

ARCHITECTURE

MESSAGES

Generate reports :

-Random alcohol level between 0 and 3 g/L
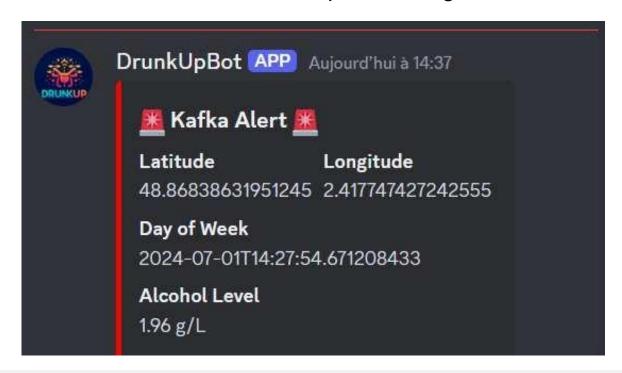- Random location in Paris
- One data every 1000ms

```scala
You, 2 seconds ago | 1 author (You)
case class Report(latitude: Double,
  longitude: Double,
  date: LocalDateTime,
  alcoholLevel: Float){

  override def toString: String = {
    s"latitude: $latitude, longitude:" +
      s" $longitude," +
      s" dayOfWeek: $date," +
      s" alcoholLevel: $alcoholLevel"
  }
}
```

ALERTS

Score < 1.5 == ALERT

Kafka Connect Job : read from the reports to topics and put it in alerts topics instantly

Discord Job: Read from alert topics and log to discord

STORAGE

- Datalake: AWS S3

- Read in Kafka "report" topic every **6 hours** –> 4 files / day

- Write using json format

- Partioned by day:

Amazon S3 〉 Buckets 〉 harmonylandrff 〉 Reports/ 〉 year=2023/ 〉 month=6/ 〉 day=20/

For testing/demo purposes, read Kafka every minute

STATISTICS

Spark Job:

• Run every day
• Write stats in Postgres database
• Grafana plugged on the database

DEMO