Peaceland Poc

The goal is to write a poc demonstrating a working architecture of peaceland.

The typical architecture will have 4 components (probably 4 Main) :
1) a program simulating the drone and sending drone like data to your solution (see subject for details on a message). Your system will store message in a distributed stream making it available to the component 2 and 3. (this part should not be done with spark)
2) handle riot alert message from stream
3) store message formatted as drone message in a distributed storage (ex: HDFS/S3)
4) analyse stored data with a distributed processing component (like spark). As a proof of your system capacity to analyse the store data answer 4 questions of your choice. (ex: is there more riot during the week or during week-end?).

All components may run independently, they must be scalable and used in a scalable way.
For component 3) you may use kafka connect or its equivalent (kinesis firehose).

Coding instruction
Any code must be written in functional scala (compile to jvm on javascript doesn't matter). Unless I accept it as an exception the keywords «for, while, return, var, throw, null» are forbidden as well as importing anything mutable. The method «.get» is forbidden as well.
Foreach as a collection or rdd method is accepted.
One exception for now : if you want to display a number of received/stored… message or alert you may use the keyword «var».
Some student may choose as an option to code the five component in another functional language (F#, Haskell…).
If you want to use nodejs for one of the five component you may use it through scala-js example (https://github.com/scalajs-io/nodejs)

Submission (2 points)

For the project you should use a git repo, work of different members of the group should be visible in different commits.
For submission you should send me an email with your git repo and the last commit hash.

If your repository is private you should grant me access.
Late submission email are accepted, minus 2 point per late day(s).

Once those 4 parts done you can work on the personnal part :
The optional part is quite open, the goal is for every group to work on something there are curious about or they find interesting for there CV. They can be done in the language of your choice unless you re using spark.

Here are some suggestion :
1) project deployed on the cloud (azure, aws, gcp,…) using IaC like terraform.

2) website using its dedicated db/queue to display every received riot alert (instead of a basic email/log/console print)
3) using docker and docker compose for the 4 components of the project (and kubernetes as ressource manager for spark if spark is used).
4) once component 3 is done, using spark-notebook/zeppelin to generate charts
5) using some dataviz or custom website to present the result of the spark analysis (within an end to end pipeline)
6) using an ml model and adding information in the message to achieve predictive maintenance of the drone.
7) Whole project code in haskell/F#…
8) any idea you find relevant for the project if I validate it

FAQ

- Except when doing the cloud option everything can be written and deploy locally on your own computer (not distributed). The recommendation regarding not using distributed kafka/spark was only for the distributed option.
- What matter is to create a working and scalable poc to demonstrate the architecture Analysis pertinence doesn't matter.)
- Scoring of the citizen is already handle in the drone you don't to write any code to adjust it apart from your drone simulator.
- Most student should focus on the basic components. A group can achieve a good mark (up to 16/20) without the personal part. The personal part is for the curious group which want to do more.
- If you want the cloud option to be done 100% correct all the components apart from drone simulator should be running on the cloud.
- presentation should be done with slides for the context and the architecture, small demo. Given time for presentation + demo is 10/12 minutes (without question