Software engineer (Challenge)

Implementation (about 70% of the expected delivery)

We would like to create a bot detection system that will collect HTTP traffic from our customers servers and analyze them as fast as possible.

Step 1

- Read the log from http://www.almhuette-raith.at/apache-log/access.log (a short summary of its content : https://stackoverflow.com/questions/9234699/understanding-apaches-access-log).
- In real life, we have a continuous stream of events coming from the web-servers. In order to emulate this, you can have your program loop continuously over the log file.
- Send the stream to a "detection system" that will analyze every request log.
- Detect "some" bots in this traffic (it's not important if the detection is not good here).

Step 2

- Document the capacities and limitations of the system, and how you assessed them.
- Feel free to document your ideas for improvement.

Step 3:

- Push the limitations of this system as far as you can (given the time you allocated on the challenge) to consume the logs as fast as possible.

Delivery

The expected (and preferred) delivery for step 1 and 3 is **Java or Scala code** + automated tests (using any Open Source tool) on critical parts + a README file explaining how to build and run it. If it is not possible, a partial implementation and a text explanation of how you would finish it is accepted. Any open-source component can be used.

The expected delivery for step 2 is text.

Your delivery will be assessed on the following criteria:

- Is it doing the job?

- Does it follow best practices on maintainability?
- How are your tests, and are they covering the most important areas?

Scalability & Architecture (about 30% of the expected delivery)

Now, you need to make such a system real-time and ready for production, to support:

- 200 requests per second in one month,
- 2000 in three months,
- 20000 in 6 months

Questions

- Do you think it is possible? Why?
- How would you architecture it (name its components)? Based on which technologies?
- What would you need before pushing to a production environment?
- How will you attest its health in production?
- How do you make sure it's actually doing its job?
- How would you store the results (max retention would be 10 days)?
- Is is efficient and easy to monitor?

Delivery

The expected delivery is text and diagrams.