HK01 Company Limited
Offsite Test for Data Engineer

- Please contain the solutions to this problem set in one public GitHub repo
- In the GitHub repo, we expect the following:
    - Codes / test cases,
    - README.MD in describing the setup as well as the execution instructions, and
    - Preferably, for Question 3, the URL to a running system.
- The codes submitted is required to be working as instructed in your README.MD
- Estimated time for this paper: 48 hours

Q1a. Access Log analytics

You are given with a public trace containing two month's log of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida (http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html)
Download path: ftp://ita.ee.lbl.gov/traces/NASA_access_log_Aug95.gz

Part of the access logfile

```
hsccs_gatorbox07.unm.edu - - [01/Aug/1995:01:44:06 -0400] "GET
/pub/winvn/release.txt HTTP/1.0" 404 -
204.199.188.113 - - [01/Aug/1995:01:49:06 -0400] "GET /facilities/vab.html
HTTP/1.0" 200 4045
ts01-ind-21.iquest.net - - [01/Aug/1995:02:53:22 -0400] "GET
/pub/winvn/readme.txt HTTP/1.0" 404 -
mpngate1.ny.us.ibm.net - - [01/Aug/1995:03:02:37 -0400] "GET /ksc.htm1 HTTP/1.0"
404 -
slip37-78.il.us.ibm.net - - [01/Aug/1995:03:03:45 -0400] "GET /images/launch.gif
HTTP/1.0" 200 240458
```

Please prepare a script (any script language executable on Linux in AWS environment) to

- Count the total number of HTTP requests recorded by this access logfile
- Find the top-10 (host) hosts makes most requests from 18th Aug to 20th Aug
- Find out the country with most requests originating from (according the source host / IP)

There are no hard execution time limit on this question.

Q2. RDBMS

Our data warehouse stores clickstream from client side and here is part of data scheme (in form of 2 SQL database table):

- Clickstream (aka. event) on user behaviour collected from our mobile app, with database scheme:
    - Table name: clickstream
    - Table fields:
        - userId: string
            - Unique ID of user
        - time: datetime
            - The time when we receive the event
        - action: string ENUM { FIRST_INSTALL, LIKE_ARTICLE, ... }
            - Type of event, eg. FIRST_INSTALL is sent when user first install the app; LIKE_ARTICLE is sent when user click like on our article, etc
        - objectId: string
            - For action is LIKE_ARTICLE, objectId means the ID of the article user like/read
    - Example of a record

time	userId	action	objectId
2017-04-05 00:00:00	F75DA5D1	LIKE_ARTICLE	23465

- Article records, with database scheme:
    - Table name: articles
    - Table fields:
        - id: string
        - title: string
        - created_by: datetime
        - updated_by: datetime
    - Example of a record

id	title	created_by	update_by
23465	Hello world	2017-04-05 00:00:00	2017-04-05 00:00:00

Please prepare SQL queries statement (works on either MySQL/ProgressSQL/Redshift) to

- Find the top-10 articles (title, ID and like received) with most LIKE received from user on 2017-04-01
- Find the count of users who install the app (i.e. with FIRST_INSTALL event) on 2017-04-01 and use our app at least once (i.e. with any event) between 2017-04-02 and 2017-04-08

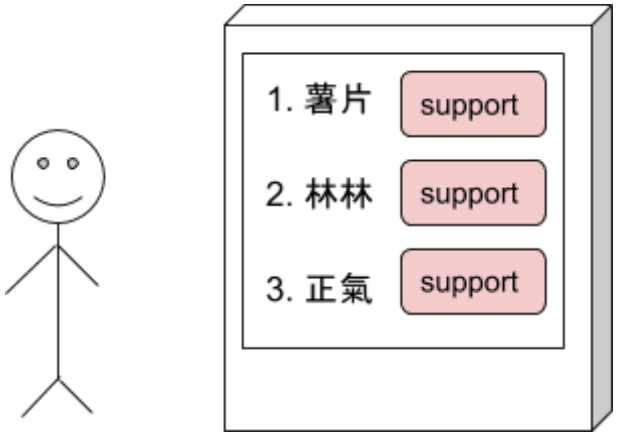
Q3. Simple (but hard) counter

Programming language requirements: Python, JavaScript, HTML

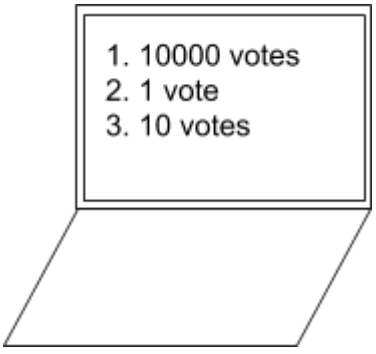
Consider we are going to set up devices on the street and ask people to express their opinions on three candidates (eg. 薯片, 林林, 正氣).

Please implement a system with the following features:

- A web interface for people to express their opinions on candidates (which will be installed onsite via physical hardware - tens/hundreds of kiosks with touch screen)
    - No need to consider any abusive usages,
    - All casted“vote” are considered to be valid,
    - The system needs to handle traffic spikes, since people can physically press on the touchscreen of the kiosk repeatedly in a very short time.
    - You may assume the connection from device and server is secure (HTTP is ok) and the server is in an isolated network (no need to handle DDoS or external attack, or CORS/XSS/etc)



- A web interface for displaying the voting result aggregated/collected from individual kiosks
    - the current total vote count for each candidate, and
    - visualization (eg. line-chart, histogram) showing (i) the total vote count for each candidate and (ii) how the last-10-minute vote distributed to the candidates.



You can choose any technology / database / cache / AWS services, as long as it is executable in AWS environment. Please also document clearly on how to set up the environment (eg. database / language runtime / etc). Automated installation, e.g., shell script, is prefered when suitable.

Last but not least, there are no hard performance goal to meet for this question, but the architecture design/scalability will be taken into consideration.