# Check Point Cloud – Senior/TL Home Assignment

## What do you need to do?

1. Implement a web UI application on port **3001** written in React using the Redux library.
   The application should be single-paged with 2 **separated** menu sections:
   a. **Home**: this section contains a simple HTML page with "Hello Guest" text on top and a drop-down menu with 3 options: black (default), red & blue. Whenever choosing an option the "Hello Guest" text color should change accordingly.

   b. **Apps**: this section contains a simple HTML table for applications with 4 columns: *ID, name, key* & *creation time*, for example:

| ID | Name | Key | Creation Time |
|---|---|---|---|
| 76318339-f66f-4494-a494-bcd2fe8e277b | App1 | Key1 | 2019-06-26 12:31:29 |
| 81d65ee6-1d6b-4afc-9a66-9ddcdce89f25 | App2 | Key1 | 2019-06-29 21:11:29 |
| 1b3439aa-4a95-4e07-a55d-ee72a6ef3a5b | App3 | Key1 | 2019-07-02 15:31:51 |

   In addition below the table there will be an HTML form for adding new application with name & key.
   The data for this table should be **saved & retrieved from a backend application** (#2)

2. Implement a RESTful backend application on port **8123** written in Go.
   The application should have 2 APIs which will be used by the web UI application from #1:
   a. **Add new application**: received application *name* & *key* and create an application with a generated UUID. The application key is sensitive and must be stored secured (you can choose how to do that)
   b. **Get all applications**: returns all applications data sorted by name

   In addition, all the applications data should be kept on PostgreSQL DB (and only there). The Postgres connection string will be provided via Environment Variable called "PG_URL" (example: `PG_URL= "host=localhost port=5432 user=admin password=password dbname=checkpoint sslmode=disable"`)

3. Authentication: the backend application should only accept HTTP requests with special **Cookie header** named "CHECKPOINTID" with static value "let-me-pass",
   e.g. *`curl 'http://localhost:8123/…' --header 'Cookie: CHECKPOINTID=let-me-pass;'…`*

4. Bonus: add unit-tests to your backend application with coverage >= 75%

## How should you deliver your code?

- Please send all source files in an encrypted zip file (with a complex password)
  - If can’t send in mail, please provide a link to a google drive containing the zip file
    - Sharing **private** GitHub repository is also a valid method for delivery.
  - Your code must compile without any warnings
  - Proper code documentation is required

- In addition, both web & backend applications (**including** PostgreSQL DB) should be run as Docker containers:
  - Include docker-compose.yml file to your zip root folder for running the applications via docker-compose tool.
  - In case you didn’t manage to work with docker-compose, provide **clear** instructions of how to run your web & backend applications (test those instructions locally before submitting)

## How are we going to test your applications?

So, we will perform few tests:

a. Start your web & backend applications via docker-compose tool, e.g.:

```
docker-compose up -d --build
```

b. Call the add **new application API** and create 3 applications

c. Open a browser on http://localhost:3001 and view your web application **Home** & **Apps** sections with the new created data

d. In case you completed the bonus part, we will also run unit-tests on your backend and verify the coverage %

## How much time do you have to complete the task?

48 hours...

If you need more time (or have any question), please contact me by email (naore@checkpoint.com) or via cell phone (050-6410099).

**This task is confidential and proprietary,**
**shall not be shared and should be deleted upon completion**.

# Good Luck!