MOCKSERVER DEVELOPERS GUIDE

I. DESCRIPTION

Mockserver is a Node.js program that allow user to refresh their AWS credential file based on a given configuration file supplied by the user. The program will automatically write new access key, secret key, and token onto the credentials file 1 minute before those keys and token expired. Both the credentials and the configuration file can be found at under users .aws folder, which is located within user home folder.

II. PREREQUISITES

Requirement:

- Node v.8.11 or higher
- curl: yum install curl (for Amazon Linux)

Installing the following node module packages:

```
npm install moment
npm install request
npm install tunnel
npm install aws-sdk
```

Before running the program, please set the proxy, for example, on Linux:

```
export http_proxy=http://name-of-your-proxy.com:8080
export https_proxy=http://name-of-your-proxy.com:8080
```

A configuration file name “config”, a credential file “credentials” (can be blank), ***TempCredScript.js***, and its node modules must be located at your home directory under .aws folder. To check, change directory to that folder and list the content (for example, on Mac OS):

```
$ cd $HOME/.aws && ls
> config    credentials   TempCredScript.js    node_modules
```

To run the program simply located the folder and type:

```
$ node mock.js
```

III. MAIN RESOURCES

A. **The folder structure:**

mockserver/

- mock.js
- readconfig.js
- getkey.js
- writecred.js
- node_modules/

.aws/

- config
- credentials
- TempCredScript.js
- node_modules/

B. **Mock.js:** this is the main file. Main.js requires other three files to run (readconfig.js, getkey.js and writecred.js should locate in the same folder). List of functions:

   a. **callconfig():** this function will call the readconfig.js file to read in the configuration file. A configuration file name *config* needed to be located in users home folder.

      i. **Parameters:** none

      ii. **Return:** *datacsv* - an array of profile objects which include target name, credential process (or source profile and instance metadata).

   b. **Processor(args):** this function require datacsv as a parameter, and the target to profile for settimeout. Processor will call itself ***2 minute*** before the expiration date.

      i. **Parameters:** *args* is an array that contain the *profile* to be called, and *datacsv*, a list of all the profile

      ii. **Return:** a promise that make a call to get the JSON that contain access key, secret key and optional token

   c. **execute(args):** this will call the getkey.js module to obtain access key, secret key, and secret token for the target

      i. **Parameters:** *args* is an array that contain the *profile* to be called, and *datacsv*, a list of all the profile

      ii. **Return:** return a promise to write a new credential

C. **Getkey.js:** this file obtain the access key, secret key and session token (optional) of a given profile. List of functions:

a. **getJSON(args):** this function will check what way to get the JSON (if the instance metadata is EC2, if it's a source profile or if it's a credential process). If it's a credential process, getJSON will execute the command automatically.
    i. **Parameters:** *args* is an array that contain the *profile* to be called, and *datacsv*, a list of all the profile
    ii. **Return:** a promise that contain the JSON, or error if no JSON can be obtained, or JSON without keys
b. **credentialSource(args):** if function is a credential source, then make a GET request using curl to get the info. Return JSON object with key info.
    i. **Parameters:** *args* is an array that contain the *profile* to be called, and *datacsv*, a list of all the profile
    ii. **Return:** a promise that contain the JSON obtain by making GET request to the instance metadata
c. **checkFileRole(p, json):** if function has an assume role, this will get the access key, secret key, and credential source to call assume role and return a new access key and secret key.
    i. **Parameters:** p is the profile with assume role, and json is the object that contain the access key, secret key or optional token to make the assume role called later
    ii. **Return:** a promise that contain the JSON obtain by making a call to the assume role

D. **Readconfig.js:** this is the configuration file. It will be called only once initially at the beginning, and will be called again if there is change to the file. It returns an array of all the profiles within the configuration file.