How can we post that log on DNIF console

Upto now we understood

1. How to create AWS instance
2. How to access that AWS instance.
3. How to host the site on that instance.
4. How to create access log in apache or JSON format.
5. How we write web service to get that access log.

Now we are understanding the concept of how we get JSON format access log, how we parse it and how we write post service to post data on DNIF console.

We have written the code in python where we required “request” API which is used to get data using “http/https” protocol. This python script is present in process1 directory of this repository.

We have to run this python script on the machine or On the instance **where we installed the our DNIF**. What are the things we did in it are as follows.

1. First of all we written a one “get” request to get the access log data from the server. Url for it is ***“http://54.190.62.177:3011/getAccessLog"***

2. Now we we get the data in JSON format as given below.

```
[
  {
    "res": {
      "statusCode": 200
    },
    "req": {
      "url": "/login",
      "headers": {
        "accept": "application/json",
        "content-type": "application/json",
        "host": "54.190.62.177:3011",
        "content-length": "69",
        "connection": "close"
      },
      "method": "POST",
      "httpVersion": "1.1",
      "originalUrl": "/login",
      "query": {

      },
      "body": {
        "userType": "admin",
        "userName": "admin",
        "password": "admin",
        "email": ""
      }
    },
    "responseTime": 1,
    "level": "info",
    "message": "POST /login 200 1ms",
    "timestamp": "2018-06-05T13:05:50.388Z"
  }
]
```

3. Here “req”, “res” are two object we get and on that two object other keys are present.

4. If we directly post that data as we get into the DNIF console.Only two columns will generate one is for “req” and one is for “res”. So it will get hard for us for further analysis.

5. In this code we parse this JSON and separate out the other data from “req” and “res” object. So after parsing we will get the data as below.

```
[
  {
    "body": {
      "userName": "admin",
      "userType": "admin",
      "password": "admin",
      "email": ""
    },
    "content-length": "69",
    "originalUrl": "/login",
    "level": "info",
    "url": "/login",
    "timestamp": "2018-06-05T13:05:50.388Z",
    "connection": "close",
    "accept": "application/json",
    "host": "54.190.62.177:3011",
    "responseTime": 1,
    "message": "POST /login 200 1ms",
    "httpVersion": "1.1",
    "content-type": "application/json",
    "method": "POST",
    "statusCode": 200
  }
]
```

6. Now all keys get separate out, now this newly created JSON we attached to our post call. The url used for that post call is given below.
http://192.168.0.29:9234/json/receive

7. Here IP address 192.168.0.29 is the IP address given in **“docker-compose.yml”.** This IP address we used for connection in DNIF console.

8. If call is executed successfully then we will get the response as

```
{
        “discarded” : 0,
        “added” : 10,

}
```

here 10 is size of our json array.

9. Now you can see your data in DNIF console for further analysis.