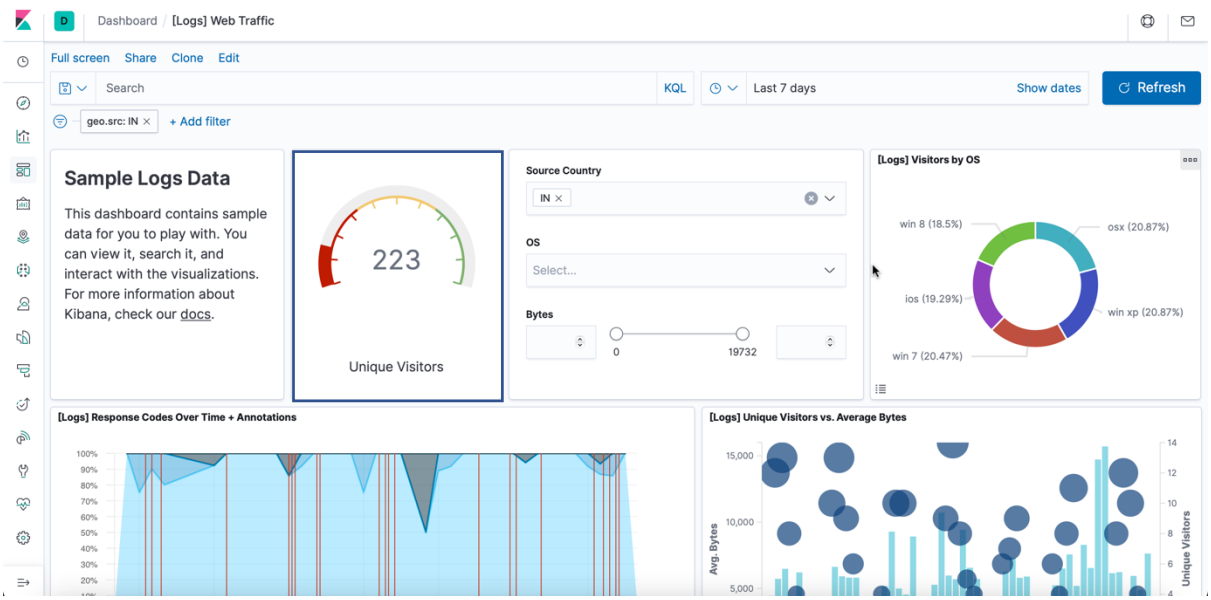
Exploring Kibana

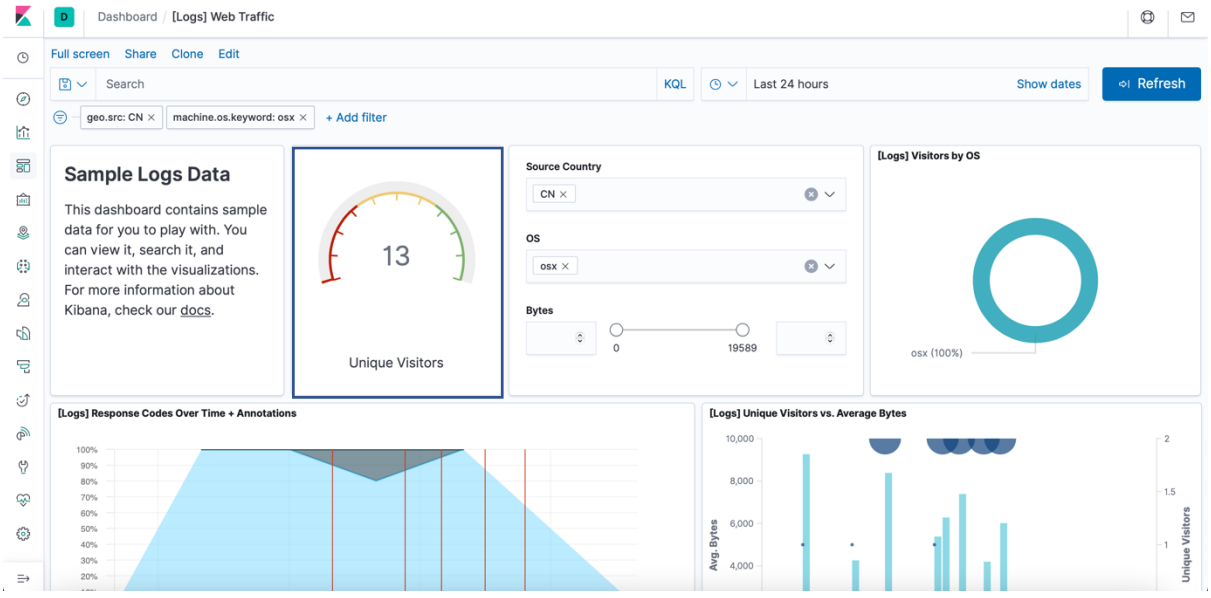
1. In the last 7 days, how many unique visitors were located in India?

   Answer : 223



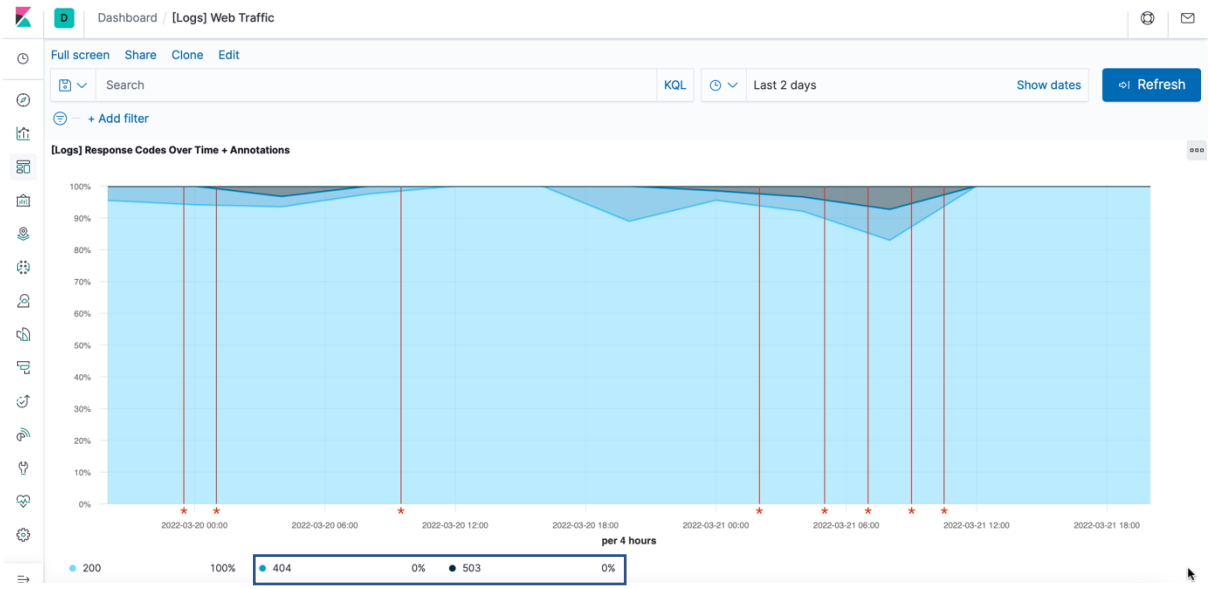
2. In the last 24 hours of the visitors from China, how many were using Mac OSX?

   Answer : 13



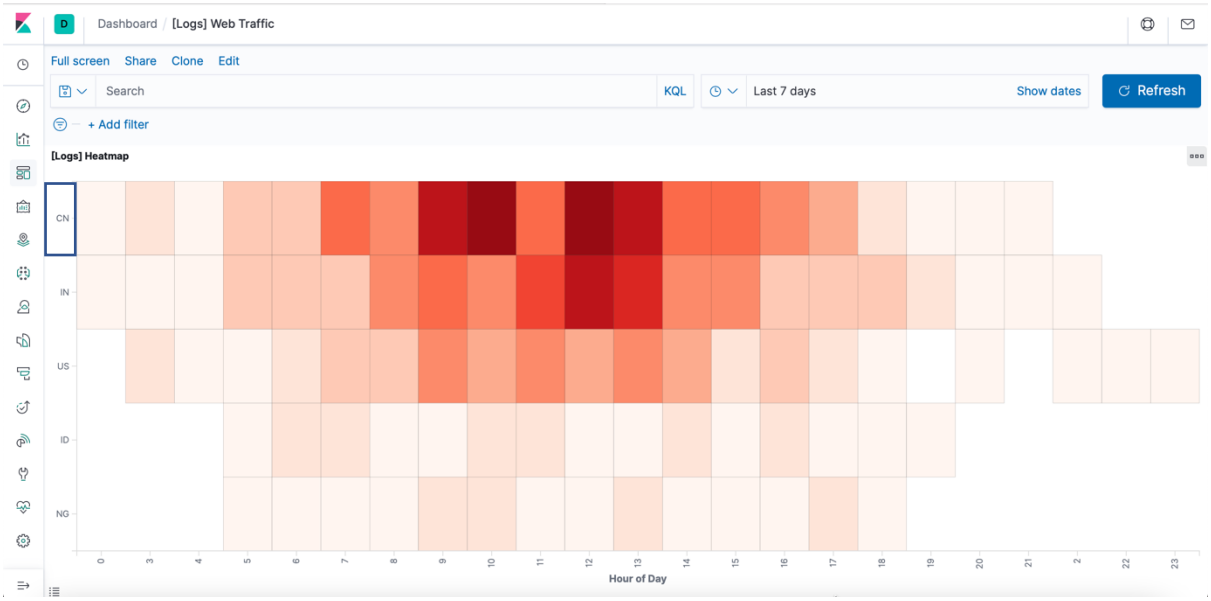
3. In the last 2 days, what percentage of visitors received 404 errors? How about 503 errors?

   Answer : 404 – 0% ; 503 – 0%



4. In the last 7 days, what country produced the majority of the traffic on the website?

   Answer : China



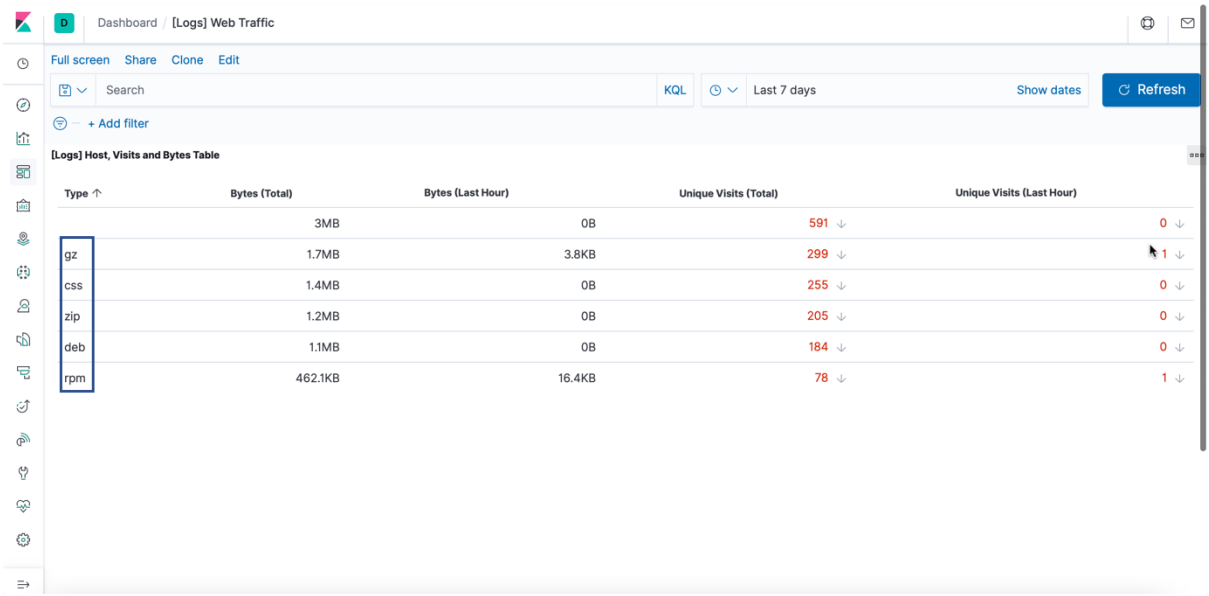
5. Of the traffic that's coming from that country, what time of day had the highest amount of activity?

   Answer : 10 am and 12 pm

6. List all the types of downloaded files that have been identified for the last 7 days, along with a short description of each file type.

    - **gz:** .gz files are archived files compressed by the standard GNU zip (gzip) compression algorithm. It stands for Gnu Zipped Archive.
    - **css:** .css files are used to format the contents of a webpage like indentation, font, size, color, line spacing, border and location of HTML information on a webpage. It stands for Cascading Style Sheet.
    - **zip:** .zip files are archive file that contains one or more compressed files or directories. It supports lossless data compression. It stands for zipped file.
    - **deb:** A .deb file is a Debian Software Package file used by Debian Linux Distribution and its variants. Each DEB file is a standard Unix archive that contains two .tar archives: one for installer control information and another for installable data.

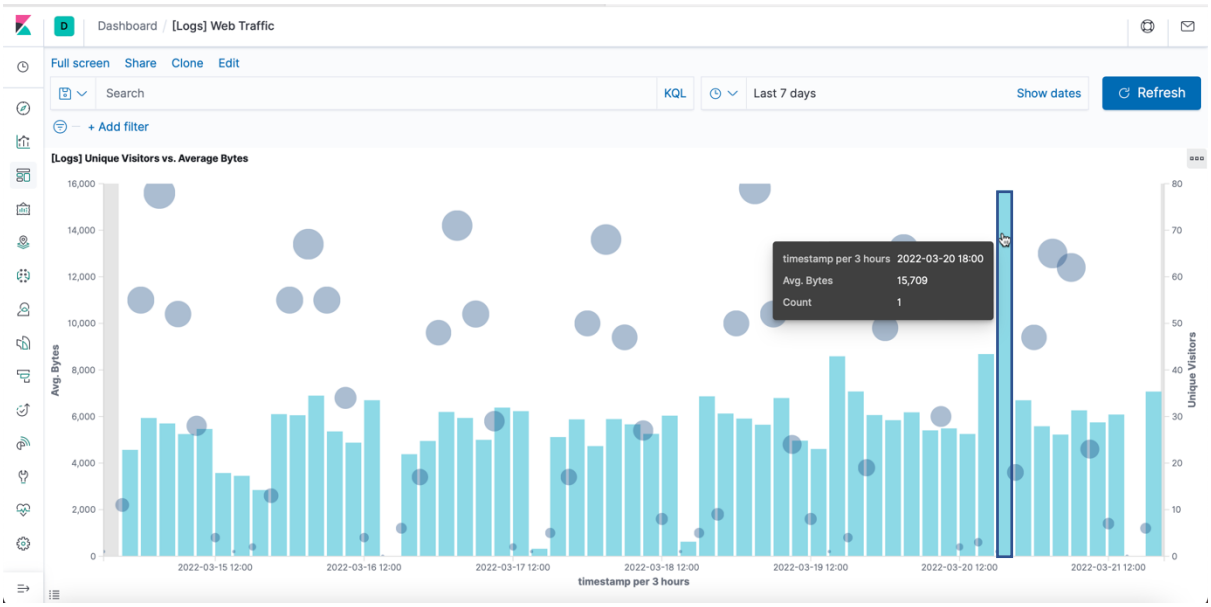
- **rpm:** .rpm file is an installation package originally developed for the Red Hat Linux operating system. It stands for Red Hat Package Manager File.



From Unique Visitors Vs. Average Bytes chart,

7. Locate the time frame in the last 7 days with the most amount of bytes (activity).

   Answer: 6 pm on 20th March 2022



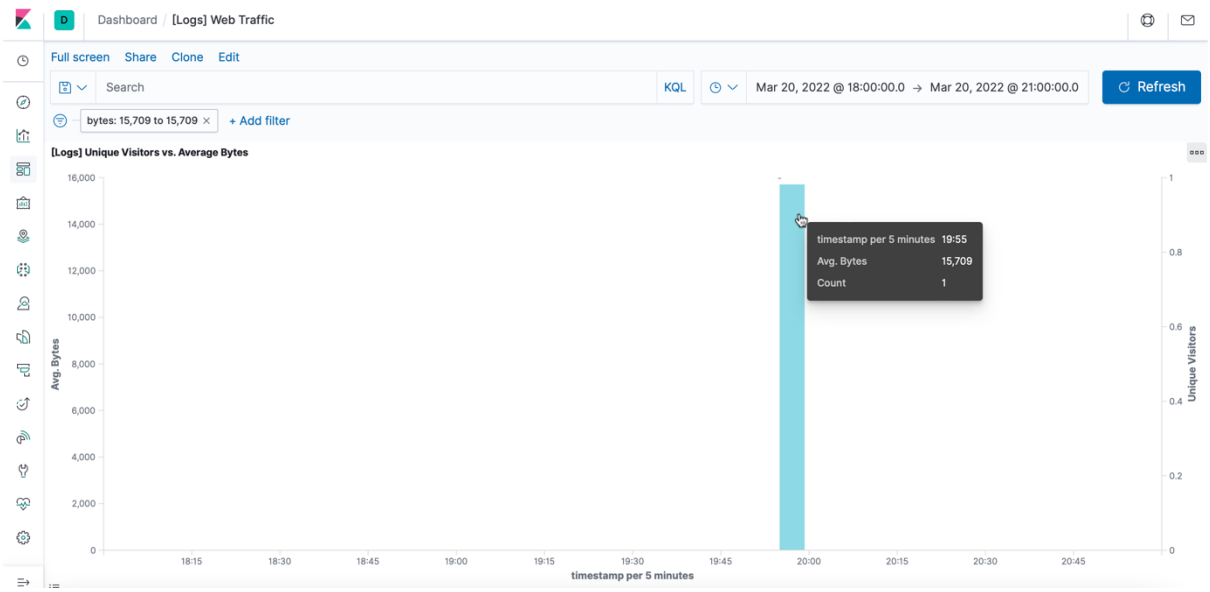
8. In your own words, is there anything that seems potentially strange about this activity?

   Answer: It is strange that a single visitor is using a much higher number of bytes (15709) than other visitors.

On filtering the data by this event,

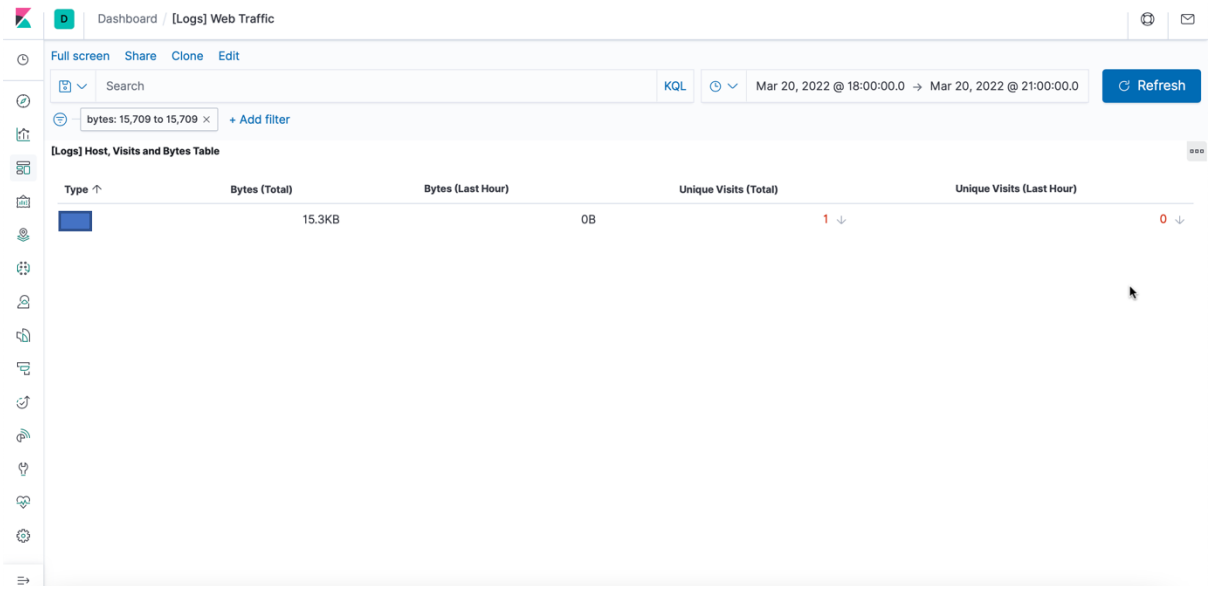
9. What is the timestamp for this event?

   Answer: The time stamp is 19:55 for the filter Mar 20, 2022 @ 18:00:0 Mar 20, 2022 @ 21:00:0.



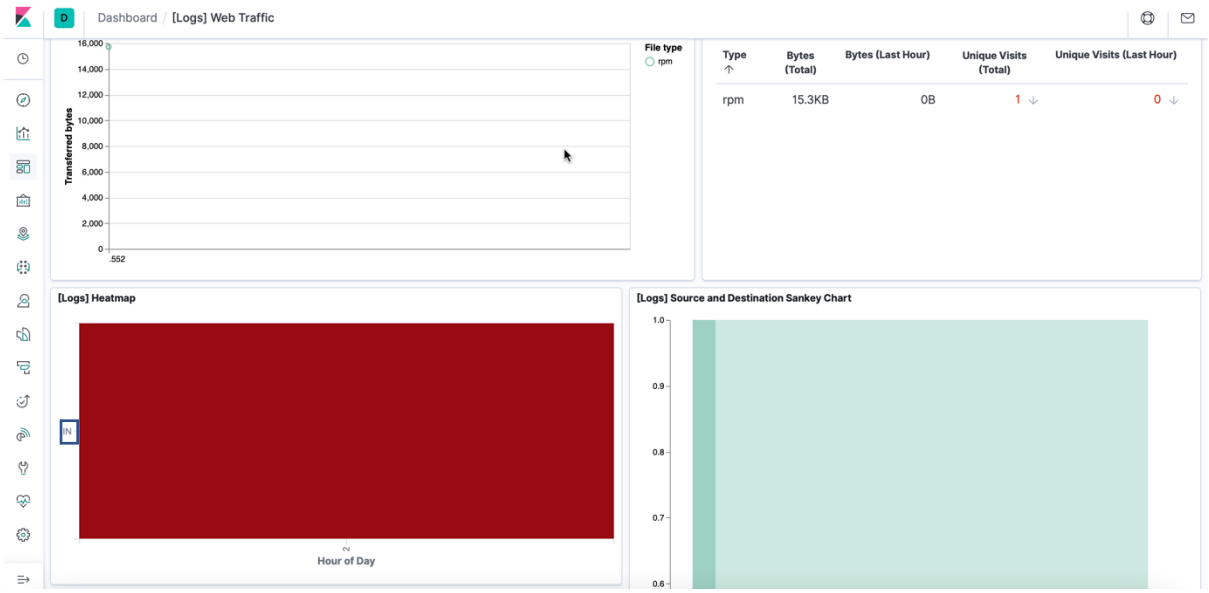
10. What kind of file was downloaded?

    Answer: An rpm file



11. From what country did this activity originate?

    Answer: India



12. What HTTP response codes were encountered by this visitor?

    Answer: 200

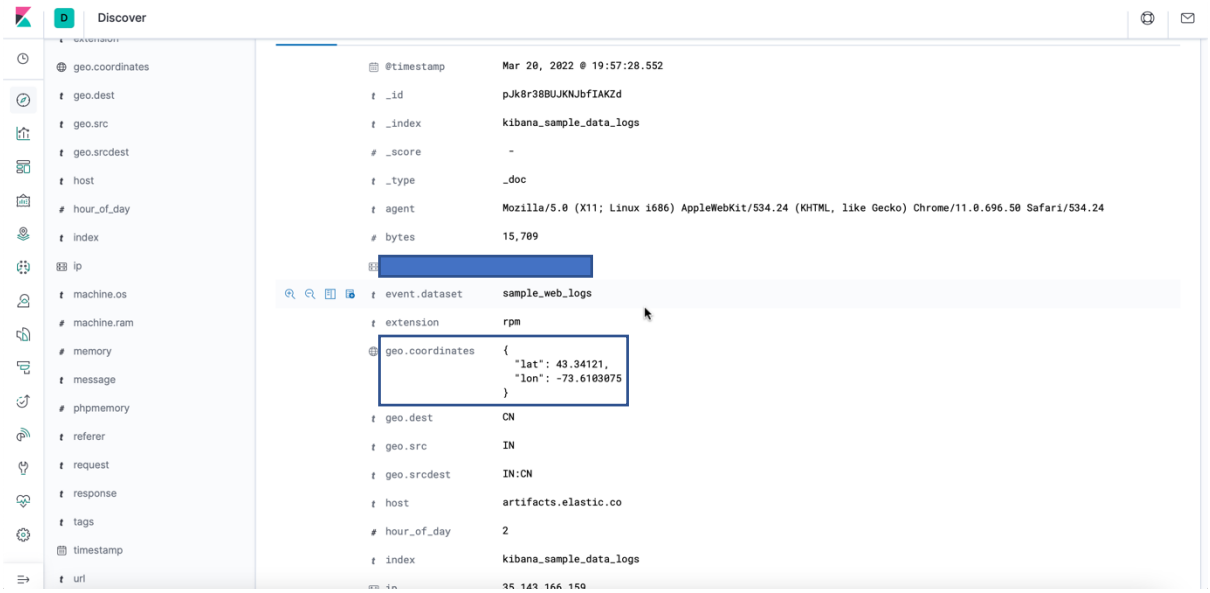
Switch over to the Kibana Discover page,

13. What is the source IP address of this activity?

    Answer: 35.143.166.159

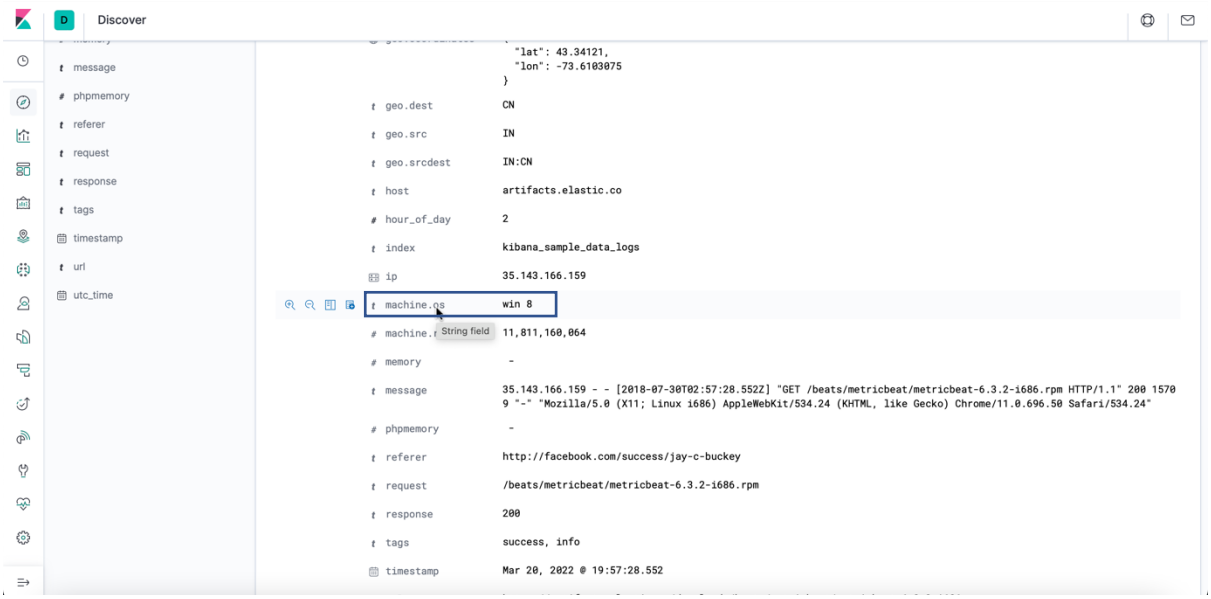
14. What are the geo coordinates of this activity?

    Answer: { "lat": 43.34121, "lon": -73.6103075 }



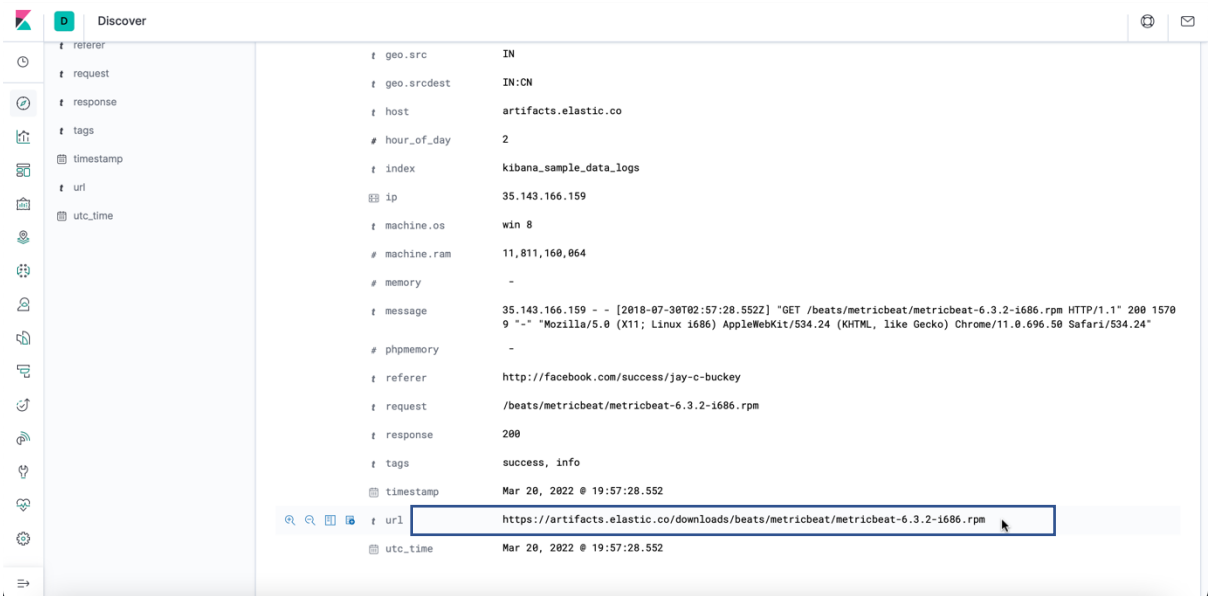
15. What OS was the source machine running?

    Answer: Windows 8



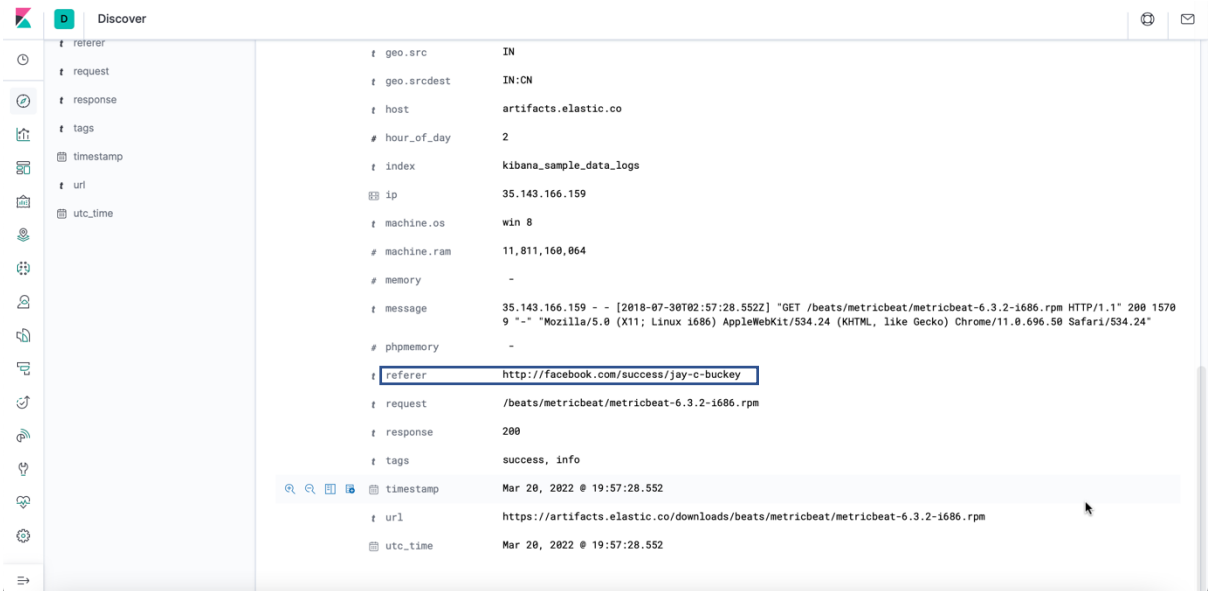
16. What is the full URL that was accessed?

    Answer: https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.3.2-i686.rpm



17. From what website did the visitor's traffic originate?

    Answer: Facebook



18. What do you think the user was doing?

    Answer: I think the user was trying to download an installation package (Metricbeat) for Linux from the website.

19. Was the file they downloaded malicious? If not, what is the file used for?

    Answer: The installation does not seem malicious but could be. The file is usually used to download or update Metricbeat.

20. Is there anything that seems suspicious about this activity?

Answer: Yes, the referrer for the download website was Facebook.

21. Is any of the traffic you inspected potentially outside of compliance guidelines?

    Answer: Since the download link was posted on Facebook, it might be outside of compliance guidelines. Ideally speaking, it is not expected to have a download/update link posted to social media.

In order to verify the ELK Server is functioning properly and Filebeat and Metricbeat are collecting data correctly, the following tasks are performed:

1. Generate a high amount of failed SSH login attempts and verify that Kibana is picking up this activity.
2. Generate a high amount of CPU usage on the pen-testing machines and verify that Kibana picks up this data.
3. Generate a high amount of web requests to your pen-testing servers and make sure that Kibana is picking them up.

Generate a high amount of failed SSH login attempts

1. Instead of accessing the Web-1 through the Ansible container, we connect from the Jumpbox. This would record the failed login attempts because the Ansible container contains our SSH keys.

   ssh RedAdmin@10.0.0.5

2. Ran the above command in a loop to generate failed login log entries.

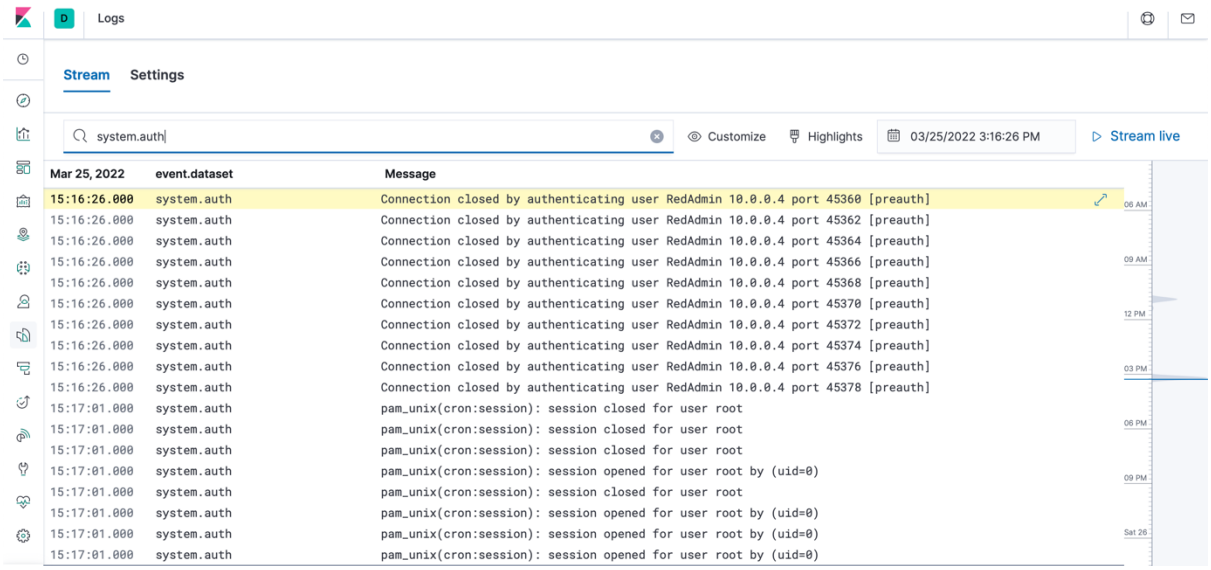
   for i in {1..10}; do ssh RedAdmin@10.0.0.5; done

   Syntax breakdown:

   - **for** begins the for loop.
   - **i ∫** creates a variable named **i** that will hold each number **in** our list.
   - **{1..10}** creates a list of 10 numbers, each of which will be given to our **i** variable.
   - **;** separates the portions of **for** loop when written on one line.
   - **do** indicates the action taken by each loop.
   - **ssh RedAdmin@10.0.0.5** is the command do runs.
   - **;** separates the portions of **for** loop when written on one line.
   - **done** closes the for loop.

```
nandhinisreekumar — RedAdmin@Jump-Box-Provisioner: ~ — ssh -i .ssh/id_rsa RedAdmin@13.83.47.196 — 116×24
RedAdmin@Jump-Box-Provisioner:~$ ssh RedAdmin@10.0.0.5
The authenticity of host '10.0.0.5 (10.0.0.5)' can't be established.
ECDSA key fingerprint is SHA256:8Grb8VoDAu9vDHLez8pRk6cujn+6oRCS7OqG8peipAg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.5' (ECDSA) to the list of known hosts.
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@Jump-Box-Provisioner:~$ for i in {1..10}; do ssh RedAdmin@10.0.0.5; done
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@10.0.0.5: Permission denied (publickey).
RedAdmin@Jump-Box-Provisioner:~$
```

3. Checked Kibana logs if the login attempts were recorded.



Bonus: Created a nested loop that generates SSH login attempts across all webservers.

while true; do for i in {5..7}; do ssh RedAdmin@10.0.0.$i; done; done

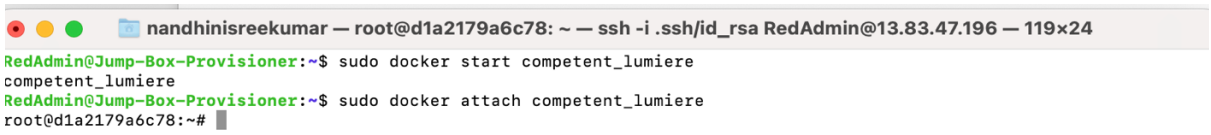
Syntax Breakdown:

- **i** in creates a variable named **i** that will hold each number **in** our list.
- **{5..7}** creates a list of numbers (5, 6 and 7), each of which will be given to **i** variable to represent the IP addresses of the webservers.
- **;** separates the portions of **for** loop when it is written on one line.
- **do** indicates the action taken each loop.
- **ssh RedAdmin@10.0.0.$i** is the command do runs . It is passing in the $i variable so the **ssh** command will be run on each webserver.
- **;** separates the portions of **for** loop when it is written on one line.
- **done** closes the for loop.

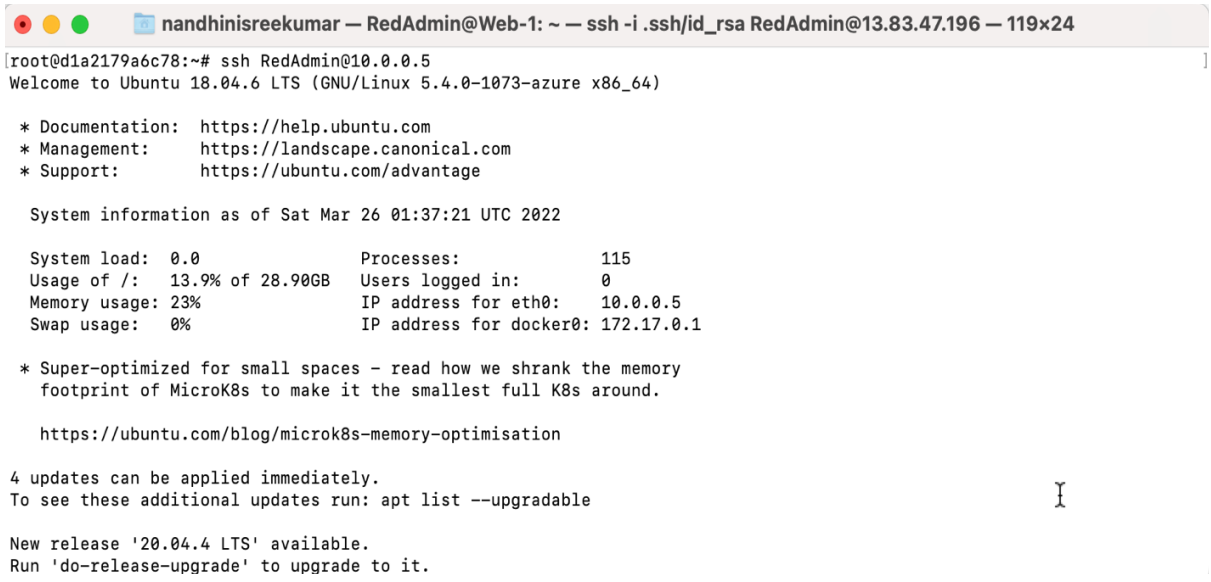
Generate a high amount of CPU usage on the pen-testing machines

1. Started and attached to the Ansible container from the Jumpbox

   sudo docker start competent_lumiere
   sudo docker attach competent_lumiere



2. Connected by SSH from the Ansible container to Web-1.

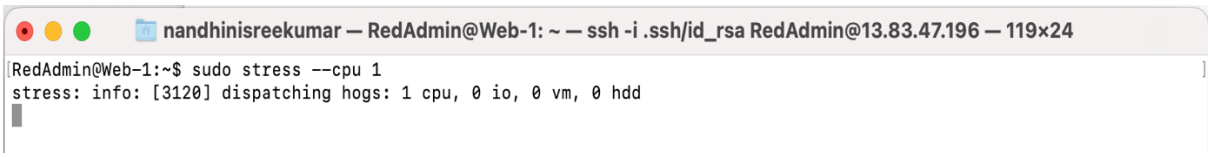


3. Intstalled the stress program.

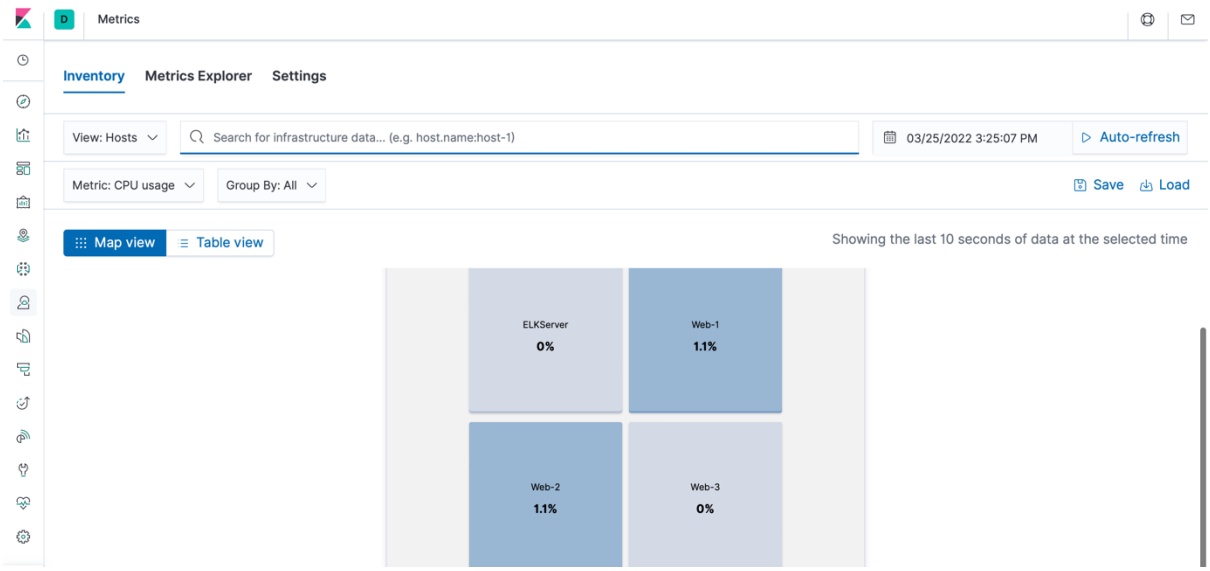
   sudo apt install stress

4. Let stress run for a few minutes.

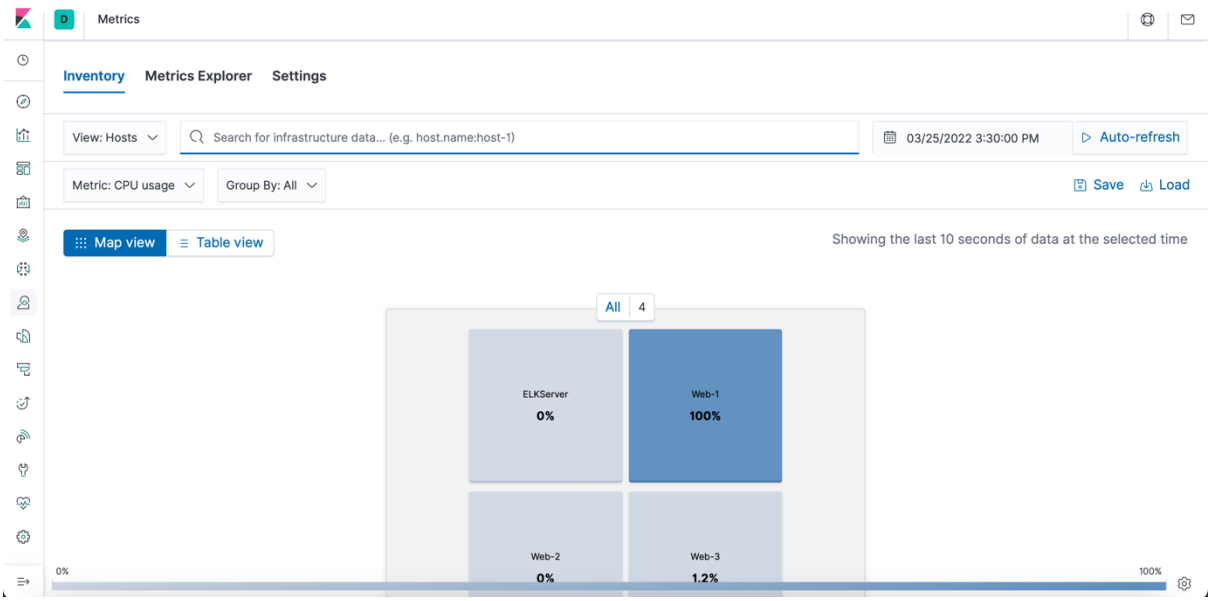
   sudo stress --cpu 1

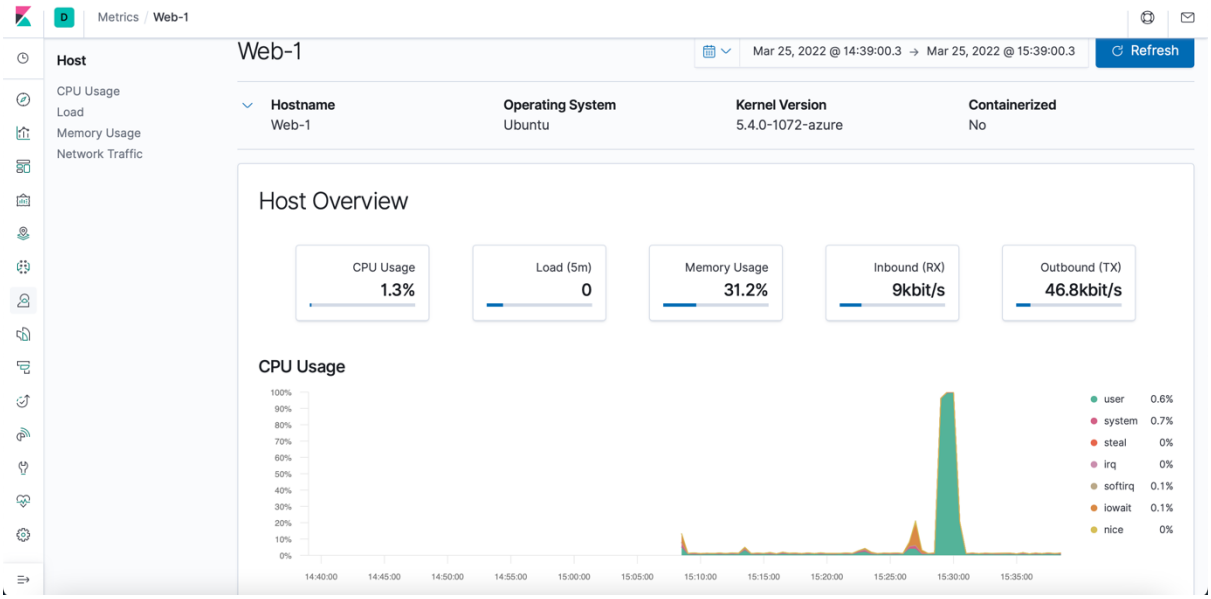


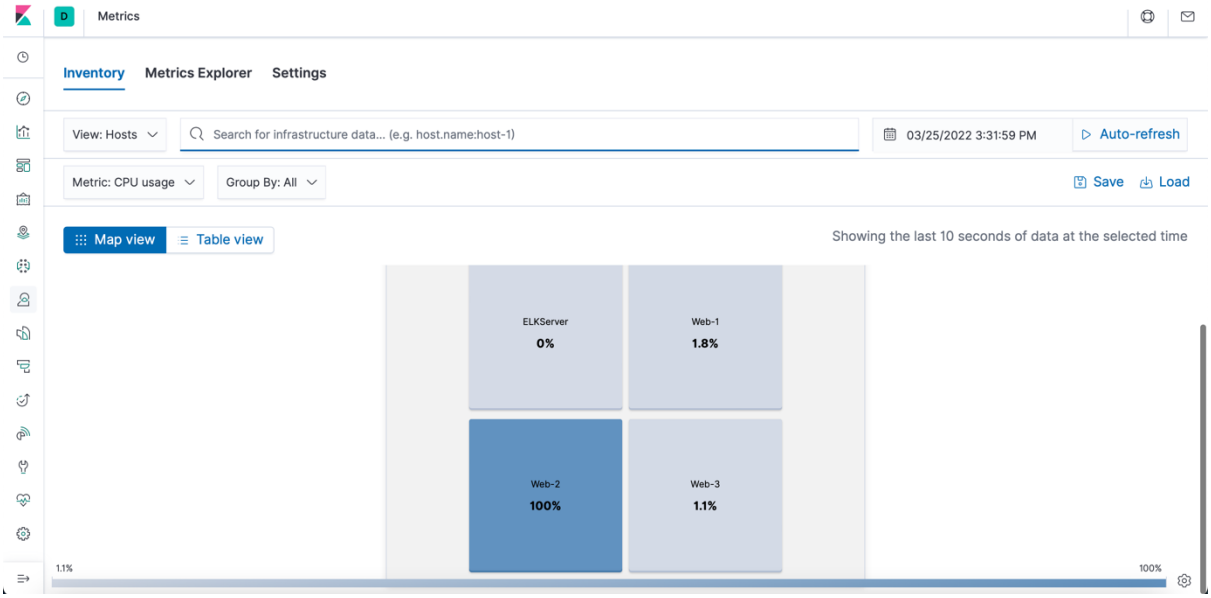
5. Checked Kibana for the change in the system metrics.

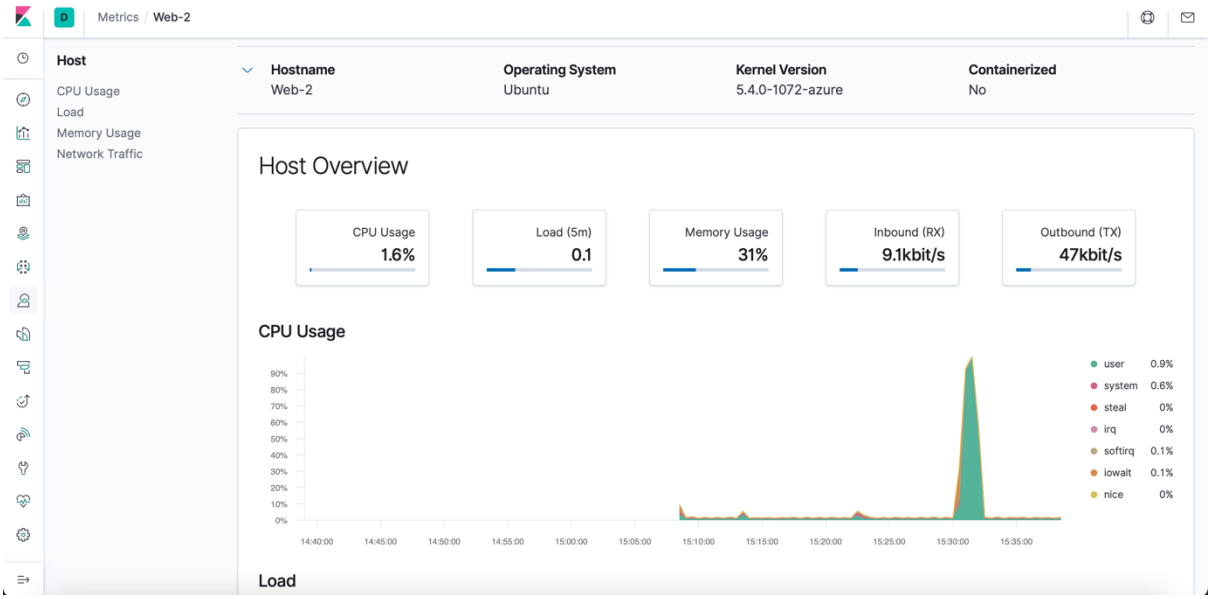


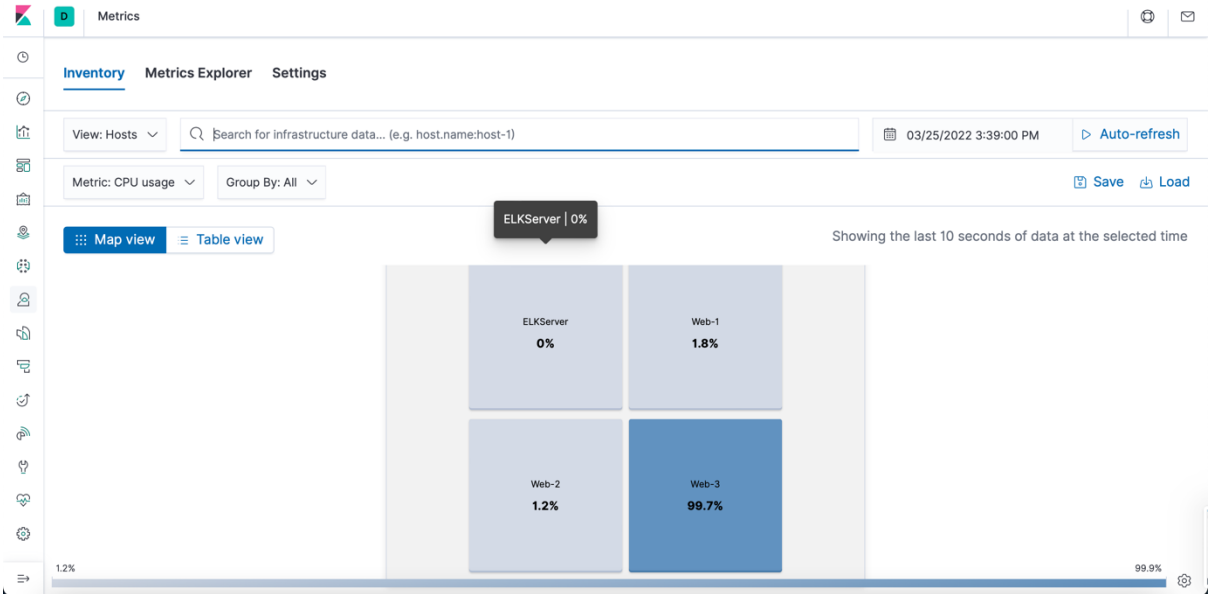
6. Ran the stress program on all three of the VMs and checked the Metric page on Kibana.

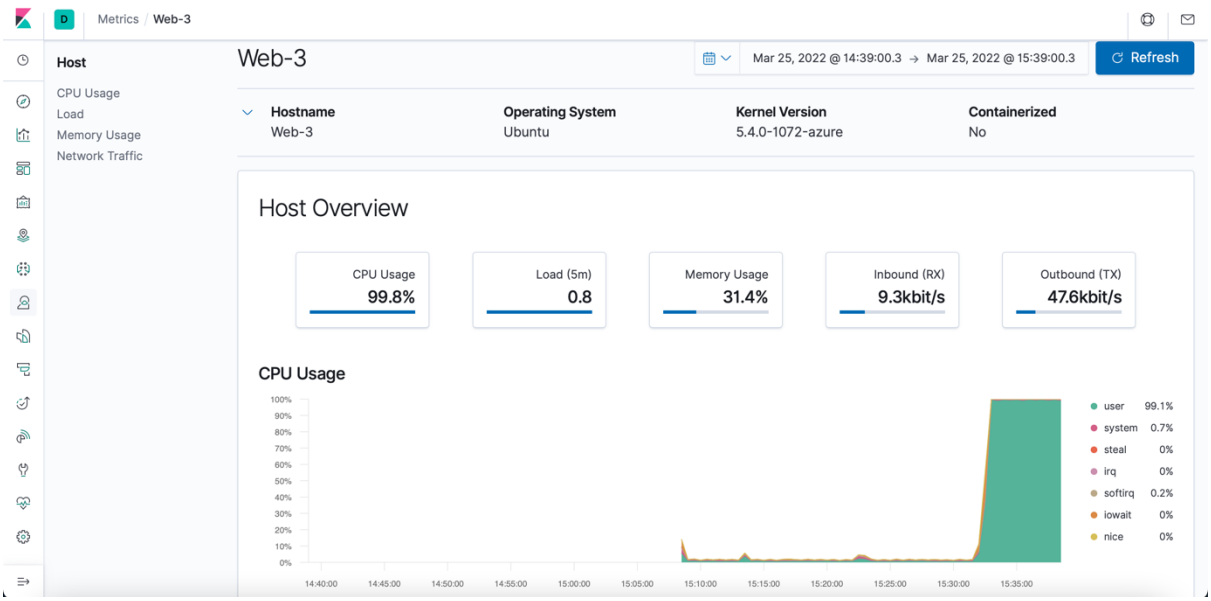



Metrics / Web-1
Host
CPU Usage
Load
Memory Usage
Network Traffic
Web-1
Mar 25, 2022 @ 14:39:00.3 → Mar 25, 2022 @ 15:39:00.3
Refresh
Hostname
Web-1
Operating System
Ubuntu
Kernel Version
5.4.0-1072-azure
Containerized
No
Host Overview
CPU Usage
1.3%
Load (5m)
0
Memory Usage
31.2%
Inbound (RX)
9kbit/s
Outbound (TX)
46.8kbit/s
CPU Usage
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
14:40:00
14:45:00
14:50:00
14:55:00
15:00:00
15:05:00
15:10:00
15:15:00
15:20:00
15:25:00
15:30:00
15:35:00
user 0.6%
system 0.7%
steal 0%
irq 0%
softirq 0.1%
iowait 0.1%
nice 0%


Metrics
Inventory
Metrics Explorer
Settings
View: Hosts
Search for infrastructure data... (e.g. host.name:host-1)
03/25/2022 3:31:59 PM
Auto-refresh
Metric: CPU usage
Group By: All
Save
Load
Map view
Table view
Showing the last 10 seconds of data at the selected time
ELKServer
0%
Web-1
1.8%
Web-2
100%
Web-3
1.1%
1.1%
100%


Metrics / Web-2
Host
CPU Usage
Load
Memory Usage
Network Traffic
Hostname
Web-2
Operating System
Ubuntu
Kernel Version
5.4.0-1072-azure
Containerized
No
Host Overview
CPU Usage
1.6%
Load (5m)
0.1
Memory Usage
31%
Inbound (RX)
9.1kbit/s
Outbound (TX)
47kbit/s
CPU Usage
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
14:40:00
14:45:00
14:50:00
14:55:00
15:00:00
15:05:00
15:10:00
15:15:00
15:20:00
15:25:00
15:30:00
15:35:00
user 0.9%
system 0.6%
steal 0%
irq 0%
softirq 0.1%
iowait 0.1%
nice 0%
Load


Metrics
Inventory
Metrics Explorer
Settings
View: Hosts
Search for infrastructure data... (e.g. host.name:host-1)
03/25/2022 3:39:00 PM
Auto-refresh
Metric: CPU usage
Group By: All
Save
Load
Map view
Table view
ELKServer | 0%
Showing the last 10 seconds of data at the selected time
ELKServer
0%
Web-1
1.8%
Web-2
1.2%
Web-3
99.7%
1.2%
99.9%



Generate a high amount of web requests to your pen-testing servers

1. Logged into the Jumpbox.

   ssh RedAdmin@13.83.47.196

2. Ran wget command to download index.html file.

   wget 10.0.0.5

3. Listed the contents to view the fie downloaded.

   ls

4. Ran a loop to generate a lot of web requests using wget.

   for i in {1..10}; do wget 10.0.0.5; done

   Syntax breakdown:

   - **for** begins the for loop.
   - **i in** creates a variable named **i** that will hold each number **in** our list.
   - **{1..10}** creates a list of 10 numbers, each of which will be given to our **i** variable.
   - **;** separates the portions of **for** loop when written on one line.
   - **do** indicates the action taken by each loop.
   - **wget 10.0.0.5** is the command do runs.
   - **;** separates the portions of **for** loop when written on one line.
   - **done** closes the for loop

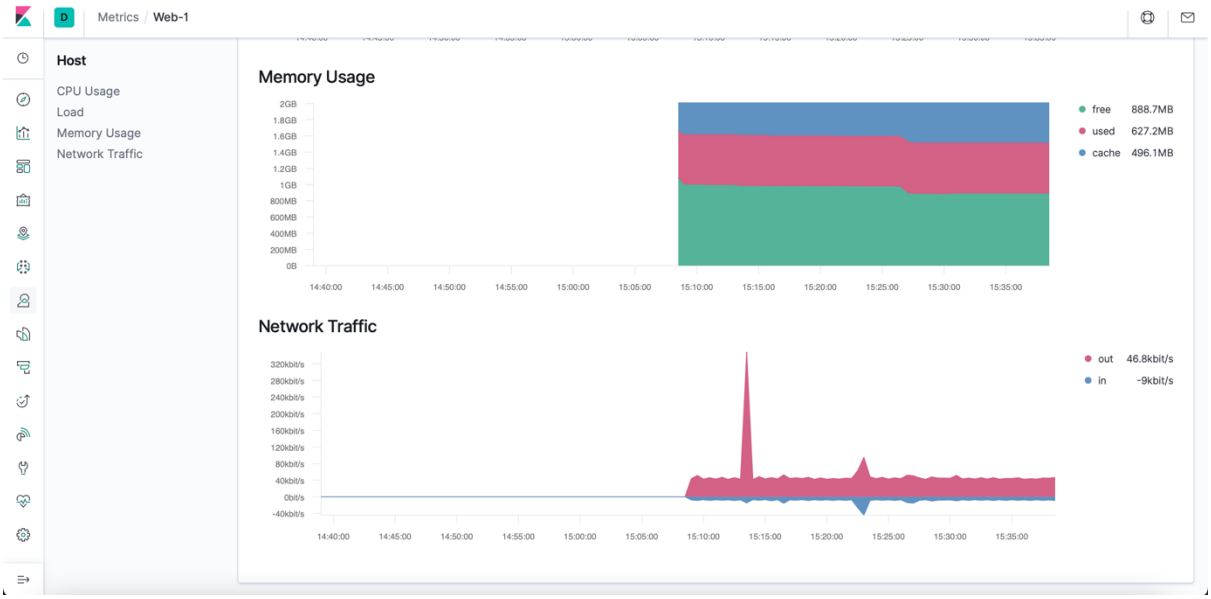
```
nandhinisreekumar — RedAdmin@Jump-Box-Provisioner: ~ — ssh -i .ssh/id_rsa RedAdmin@13.83.47.196 — 121×24
[RedAdmin@Jump-Box-Provisioner:~$ wget 10.0.0.5
--2022-03-25 23:38:21--  http://10.0.0.5/
Connecting to 10.0.0.5:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: login.php [following]
--2022-03-25 23:38:21--  http://10.0.0.5/login.php
Reusing existing connection to 10.0.0.5:80.
HTTP request sent, awaiting response... 200 OK
Length: 1415 (1.4K) [text/html]
Saving to: 'index.html'

index.html                    100%[================================================>]   1.38K  --.-KB/s    in 0s

2022-03-25 23:38:21 (245 MB/s) - 'index.html' saved [1415/1415]

[RedAdmin@Jump-Box-Provisioner:~$ ls
index.html
[RedAdmin@Jump-Box-Provisioner:~$ for i in {1..10}; do wget 10.0.0.5; done
--2022-03-25 23:38:50--  http://10.0.0.5/
Connecting to 10.0.0.5:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: login.php [following]
--2022-03-25 23:38:50--  http://10.0.0.5/login.php
Reusing existing connection to 10.0.0.5:80.
```

On checking the Metrics page for Web-1 on Kibana, the following was noted.



Bonus: Since **wget** creates a lot of duplicate files, we use **rm** command to delete all files. We use the following command not to save any files.

while true; do wget 10.0.0.5 -O /dev/null; done

```
nandhinisreekumar — RedAdmin@Jump-Box-Provisioner: ~ — ssh -i .ssh/id_rsa RedAdmin@13.83.47.196 — 116×24
RedAdmin@Jump-Box-Provisioner:~$ rm *
RedAdmin@Jump-Box-Provisioner:~$ while true; do wget 10.0.0.5 -O /dev/null; done
--2022-03-25 23:09:34--  http://10.0.0.5/
Connecting to 10.0.0.5:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: login.php [following]
--2022-03-25 23:09:34--  http://10.0.0.5/login.php
Reusing existing connection to 10.0.0.5:80.
HTTP request sent, awaiting response... 200 OK
Length: 1415 (1.4K) [text/html]
Saving to: '/dev/null'

/dev/null                   100%[=========================================>]   1.38K  --.-KB/s    in 0s

2022-03-25 23:09:34 (169 MB/s) - '/dev/null' saved [1415/1415]

--2022-03-25 23:09:34--  http://10.0.0.5/
Connecting to 10.0.0.5:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: login.php [following]
--2022-03-25 23:09:34--  http://10.0.0.5/login.php
Reusing existing connection to 10.0.0.5:80.
HTTP request sent, awaiting response... 200 OK
Length: 1415 (1.4K) [text/html]
```

Bonus: Write a nested loop that sends your wget command to all VMs over and over.

while true; do for i in {5..7}; do wget -O /dev/null 10.0.0.$i; done; done

Syntax Breakdown:

- **i** in creates a variable named **i** that will hold each number **in** our list.
- **{5..7}** creates a list of numbers (5, 6 and 7), each of which will be given to **i** variable to represent the IP addresses of the webservers.
- **;** separates the portions of **for** loop when it is written on ine line.

- **do** indicates the action taken each loop.
- **wget 10.0.0.$i** is the command do runs . It is passing in the $i variable so the **ssh** command will be run on each webserver.
- **done** closes the for loop.