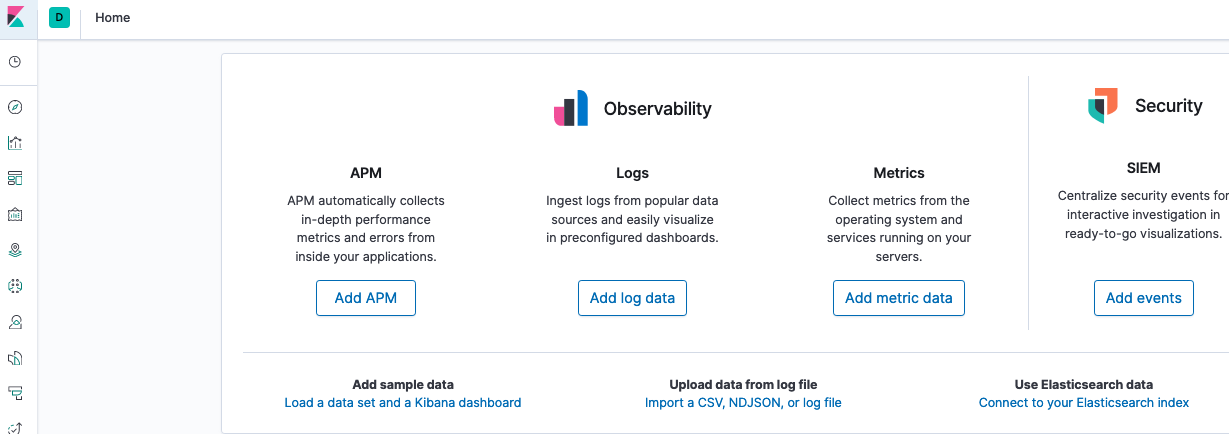
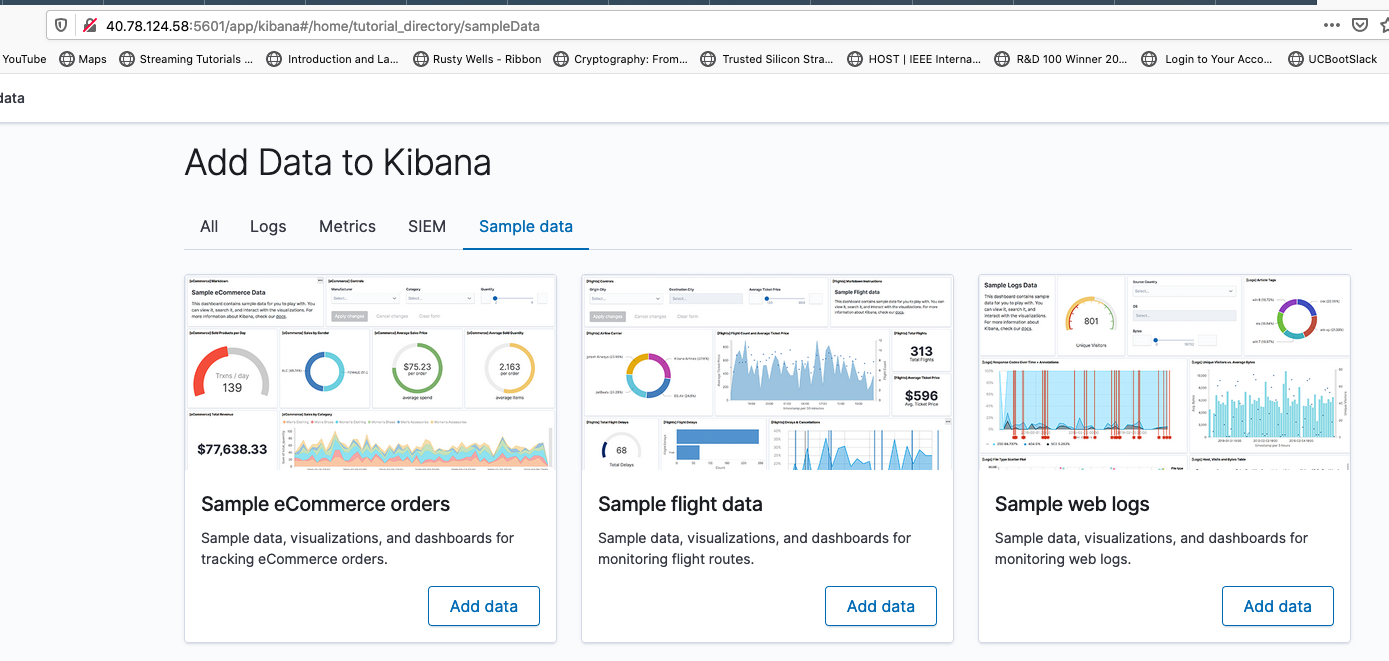
Kibana Analysis

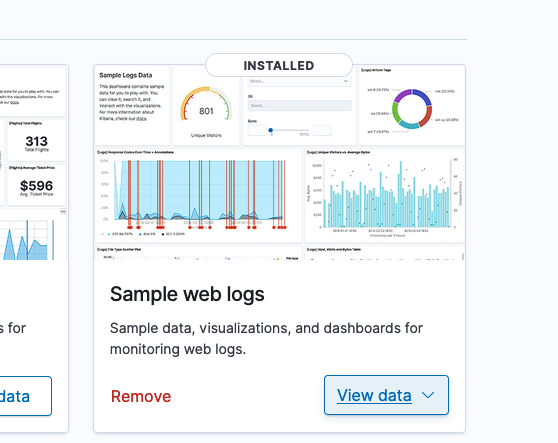
Details of the results of the Kibana-related activities

Activity File: Exploring Kibana

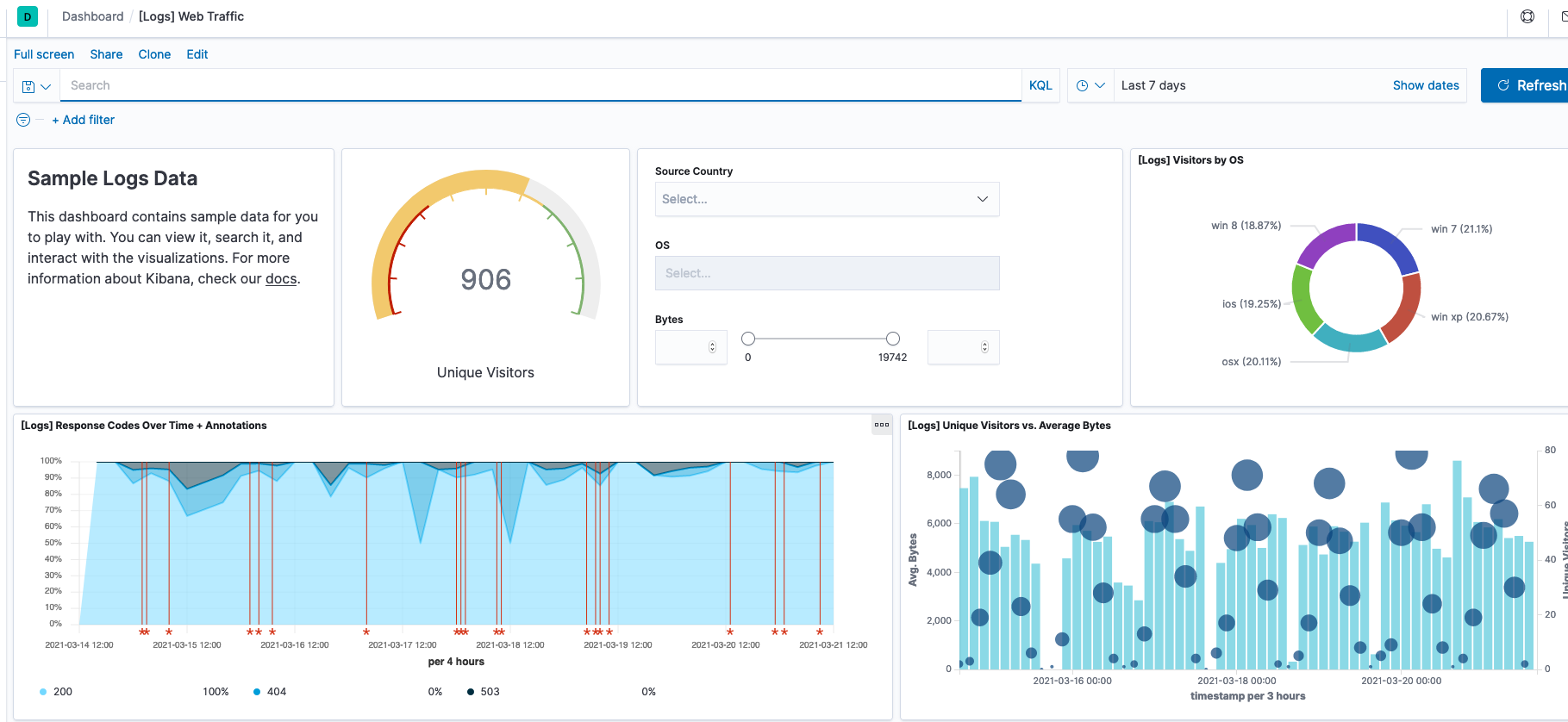
1. Add sample log data to Kibana: click on Load a data set link





Then click Add data in Sample web logs panel above which produces the following:

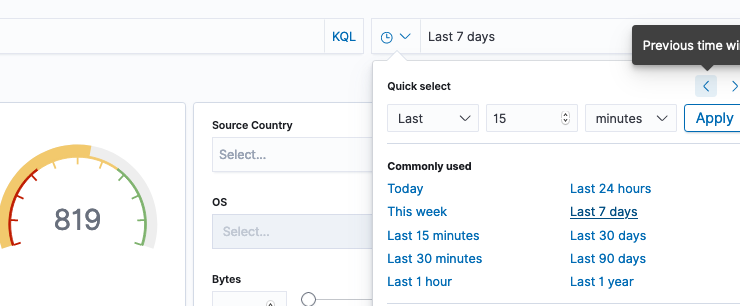
To see the sample data click on View data-> Dashboard above to show following dashboard:



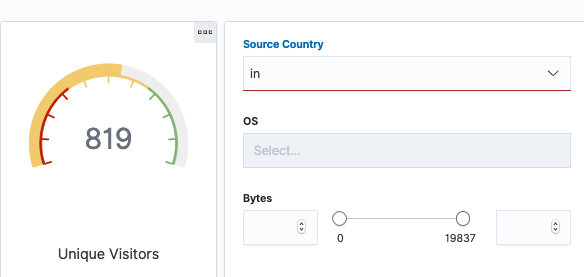
2. Questions:

* In the last 7 days how many unique visitors from India?

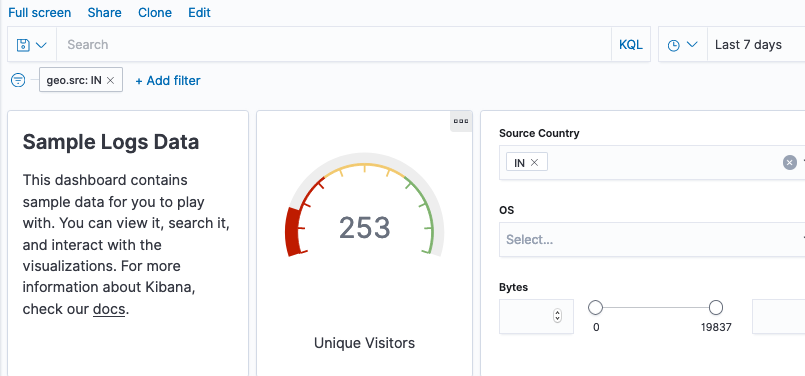
Set time frame to last 7 days



Set filter to IN (India) by click down arrow and typing IN into filter box



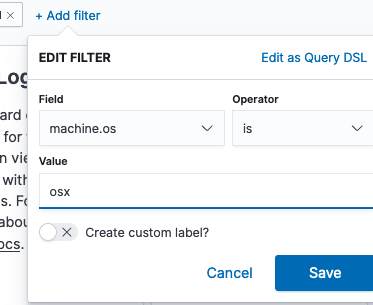
Dashboard updates to those new settings and produces below which shows 253 unique visitors from India over last 7 days.

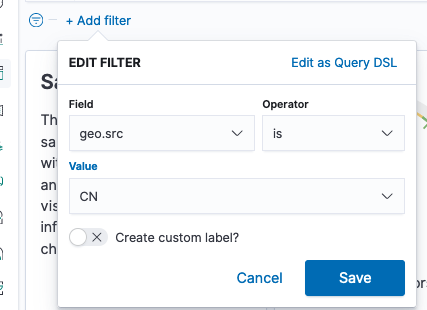


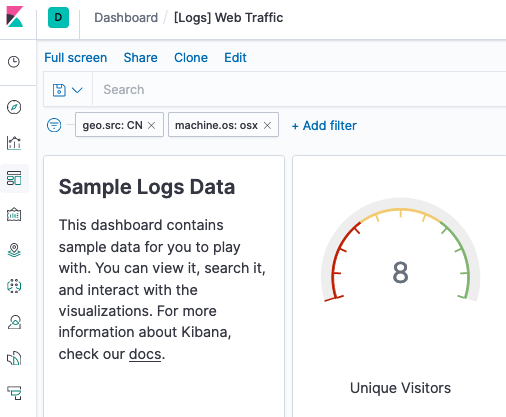
* + In the last 24 hours, of visitors from China, how many were using Max OSX?

Change time frame to last 24 hours.

Set 2 filters to geo.src is CN and machine.os is osx





This produces the answer of 8

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

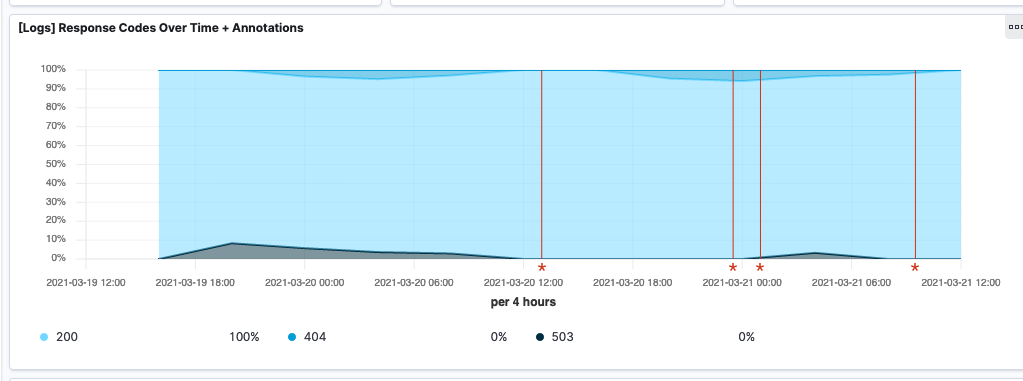
Set time frame to last 2 days

Looking at the [Logs] Response Codes Over Time chart below reveals 100% of visitors received 404 (page not found) and 0% received 503 (http service not available) responses. Seems there is a general problem with a particular missing page that affects all visitors.

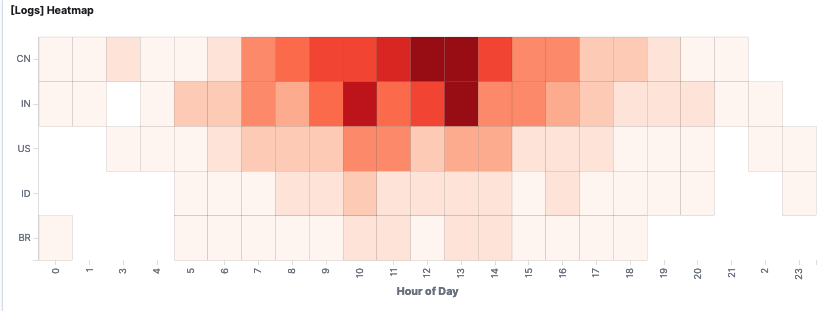
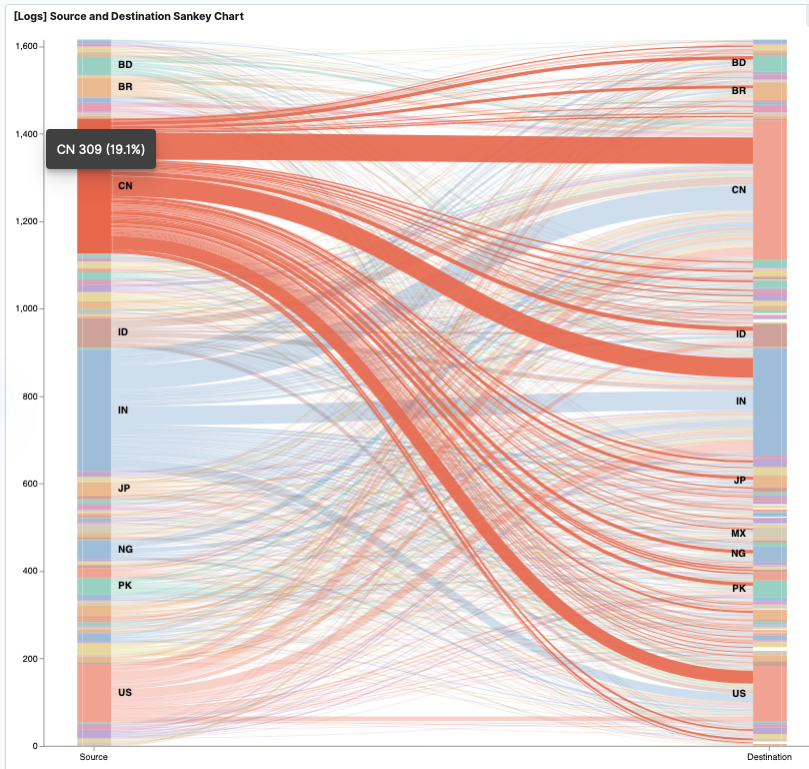
* In the last 7 days, what country produced the majority of the traffic on the website?

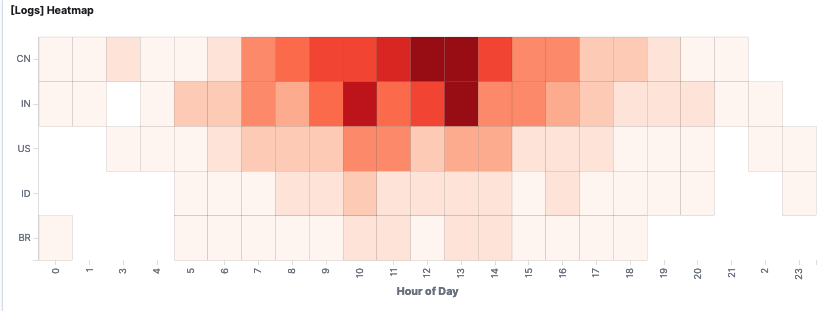
Set time frame back to 7 days

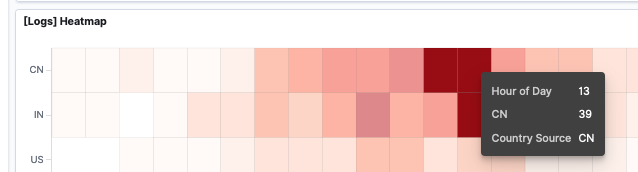
Look at the [Logs] Source and Destination Sankey Chart and determine which vertical bar on the left is tallest. That corresponds to the most source traffic. Hover over the tallest vertical area which happens to belong to China to see that China contributed 19.1% of total traffic



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

Top row of Heat Map is China. Both 12:00 and 13:00 have the same color and are the darkest which means the highest value. Both show value of 13





* 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

The [Logs] Host, Visits and Bytes Table shows download types

gz represents files compressed with gzip

css are cascading style sheets which are involved in web page presentation

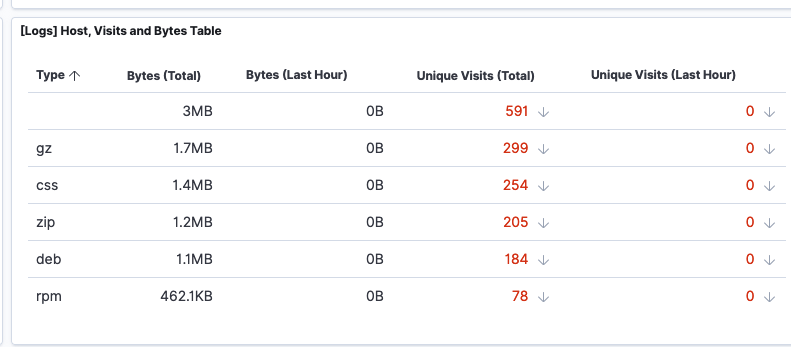
zip represents compressed file archives generated with zip

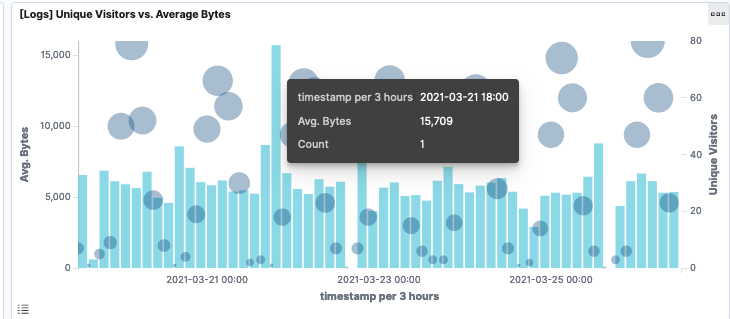
beb are software package files for Debian installs

rpm files are package files from Red Hat

3. Deeper look at the data.

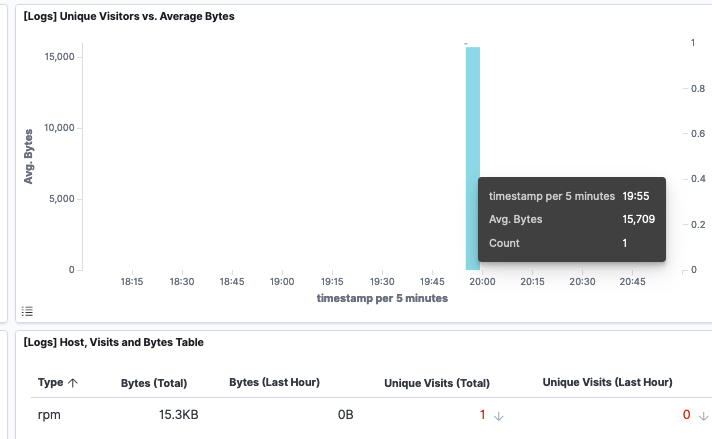
Unique visitors vs avg. bytes last 7 days: One unique visitor om 3/21 at 18:00 shows about double the typical Avg Bytes as below.



In and of itself that is noteworthy since it’s different but not orders of magnitude different so not automatically a large cause for concern. It could just be that user is doing something that’s just double the typical size but legit. But something that merits investigation.

Filtered the data by this event by double-clicking the bar:

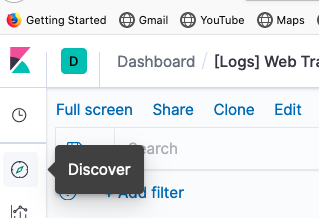
That shows actual timestamp is 19:55, file type downloaded was a rpm file and the request came from India. The response was 200 meaning it succeeded



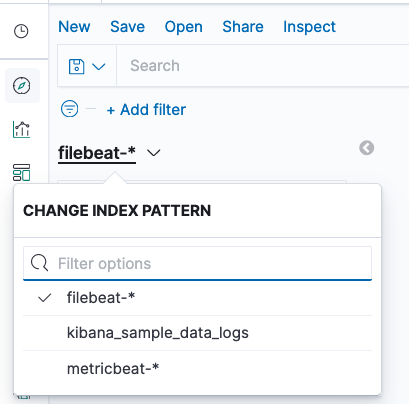




Change to Discover by clicking Discover icon:



Change on down arrow to change to kibana-sample-data-logs



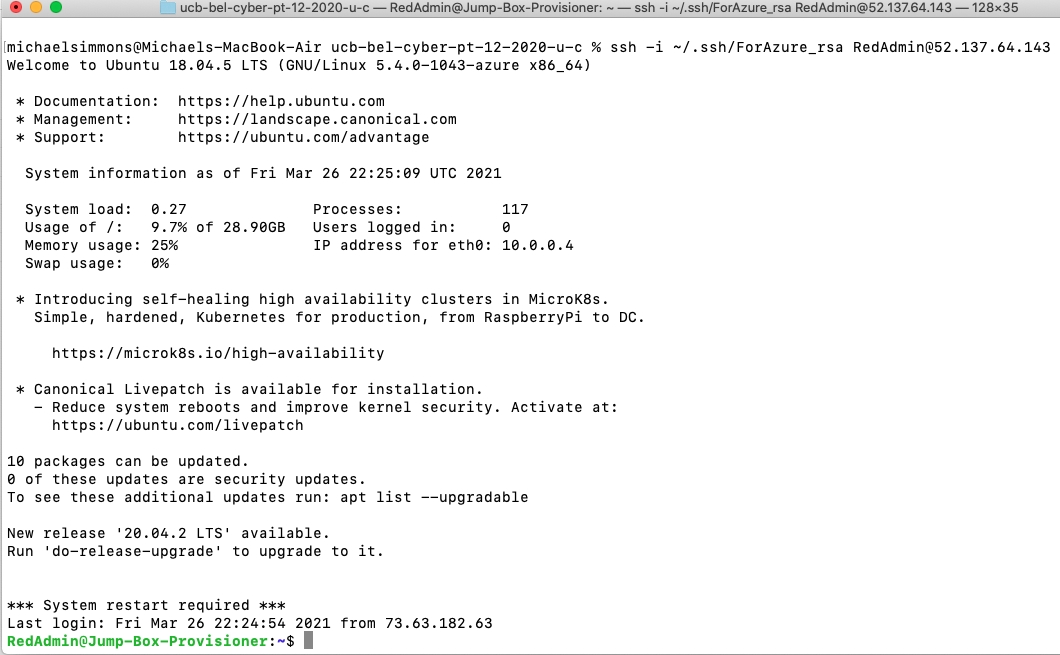
* What is the source IP address of this activity?: 35.143.166.159
* What are the geo coordinates of this activity?: geo.coordinates:{ "lat": 43.34121, "lon": -73.6103075 }
* What OS was the source machine running?: Windows 8
* What is the full URL that was accessed?:https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.3.2-i686.rpm
* From what website did the visitor's traffic originate?: 

6. Nothing malicious here: The user was simply trying to install filebeats. If this was a personal pc install no issue. If a company machine they may not be authorized to install this sort of thing and the user should be contacted and advised not to do this again if not authorized. However it could have been someone in IT with a legitimate reason.

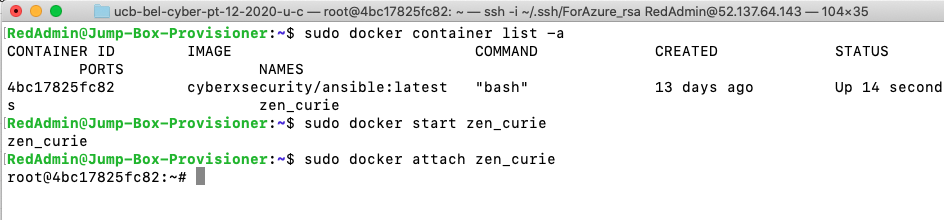
Activity File: Kibana Continued

SSH Barrage

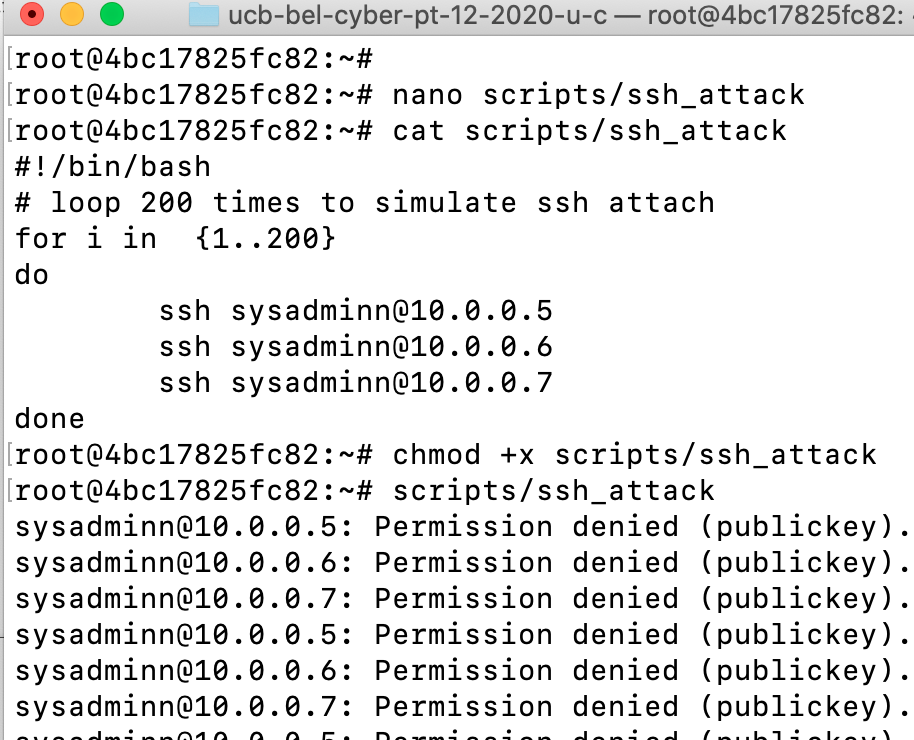
Connected to Jump-Box:

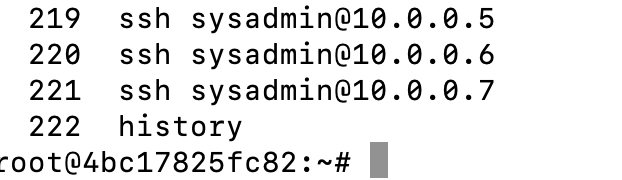


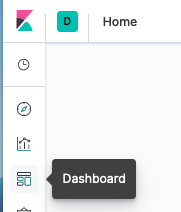
Check zen\_curie container still there, start it and attach to it

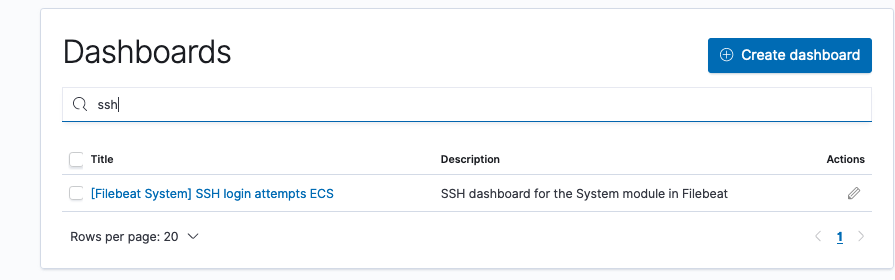


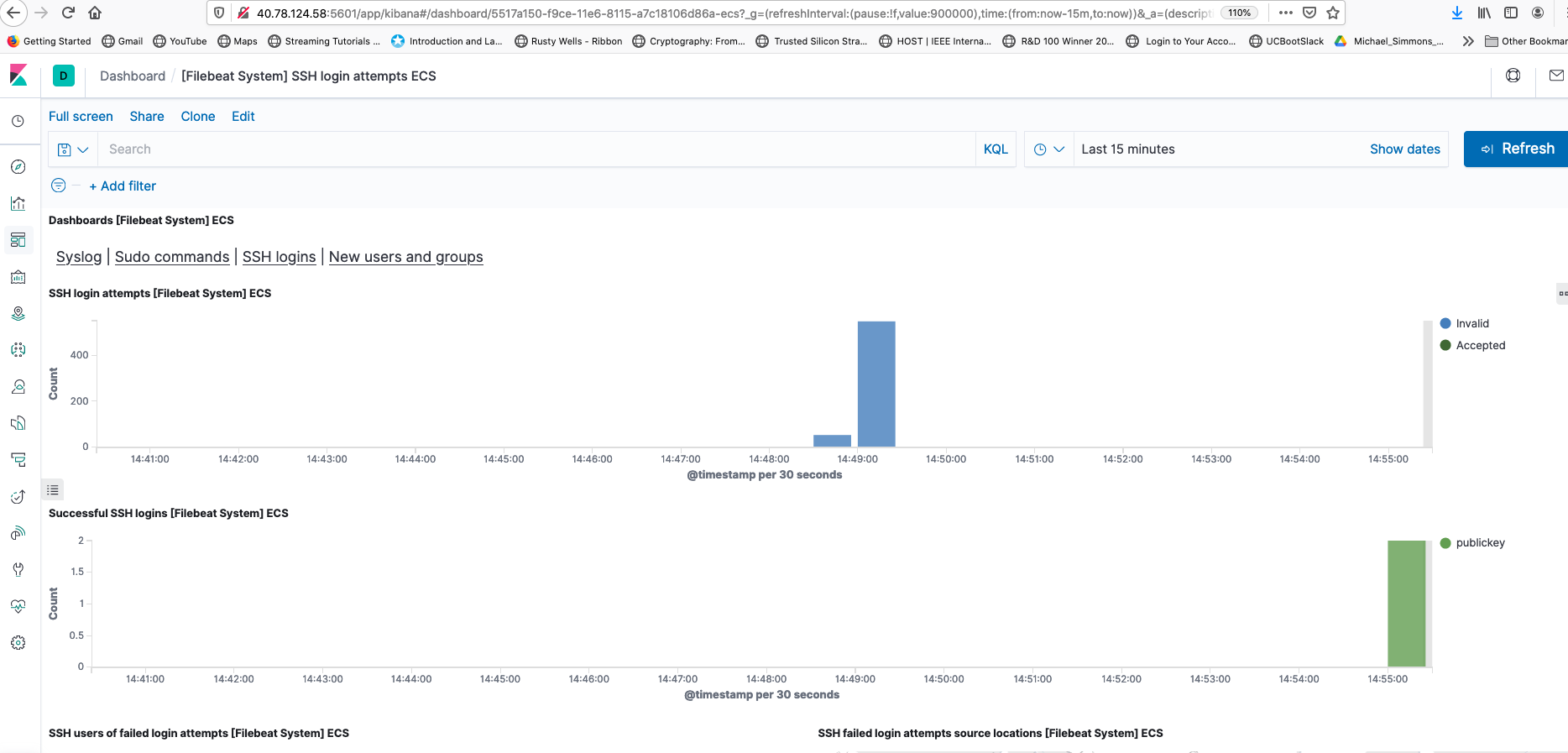
Wrote the following script that generated many failed ssh logins and ran it. It loops 200 times through 3 IPs so should generate 600 failed logins



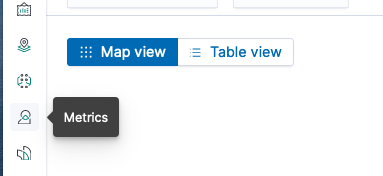
Also run 3 successful ssh commands:

Open Kibana and click on Dashboard

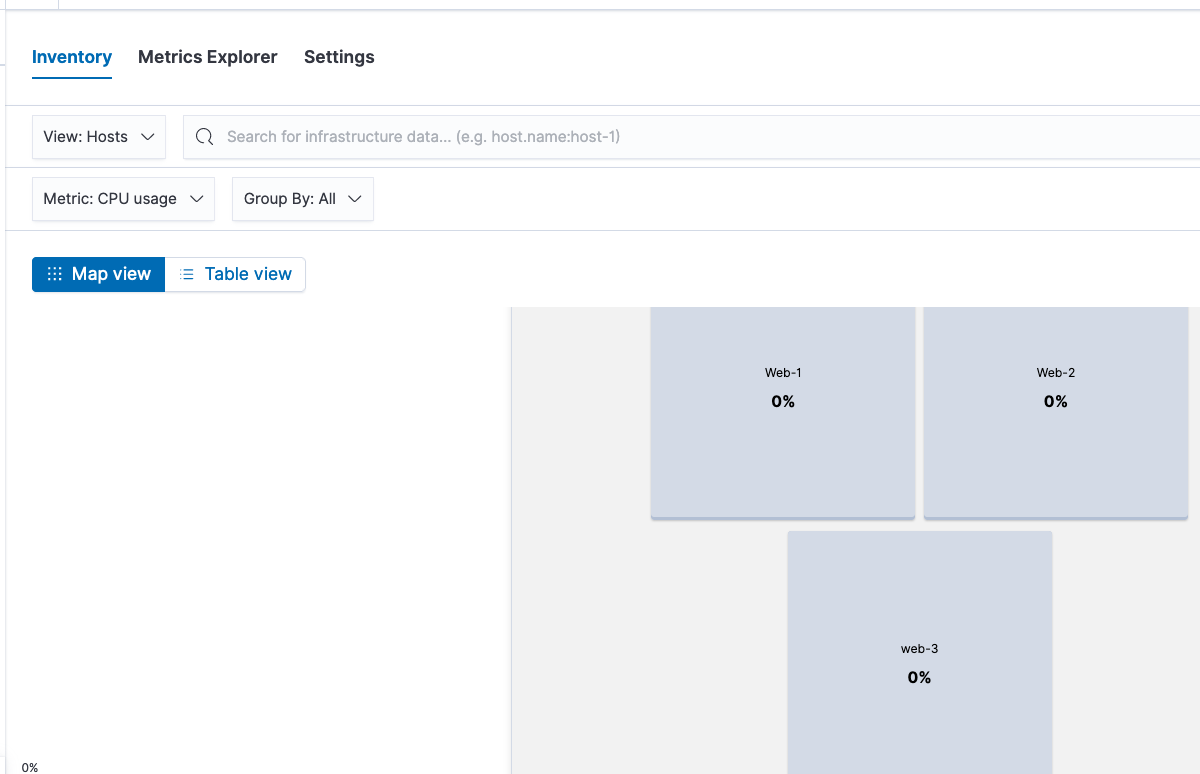
Search for ssh and click on resulting link

600 fails and 3 successes are seen. One bar is 547. The other is 53 which add to 600. They are in different bars due to it splitting in 30 sec time intervals

Linux Stress

Kibana before stress command shows little CPU usage.

Click on Metrics icon to bring this up.



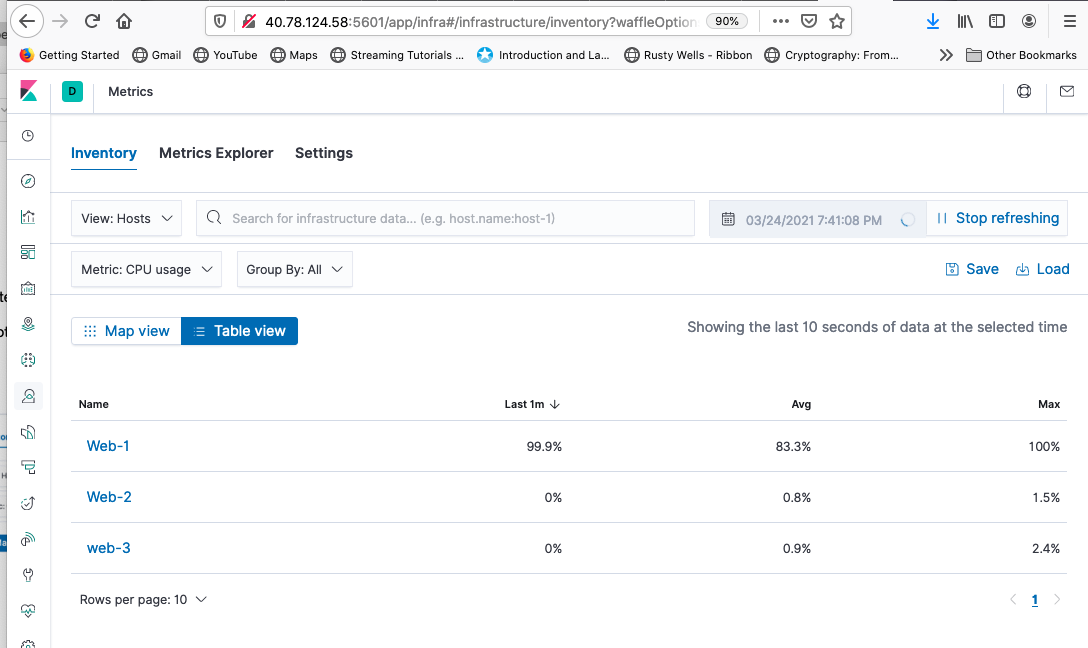
Connect to first web server and Install and run stress command one first web server:

ssh [sysadmin@10.0.0.5](mailto:sysadmin@10.0.0.5)

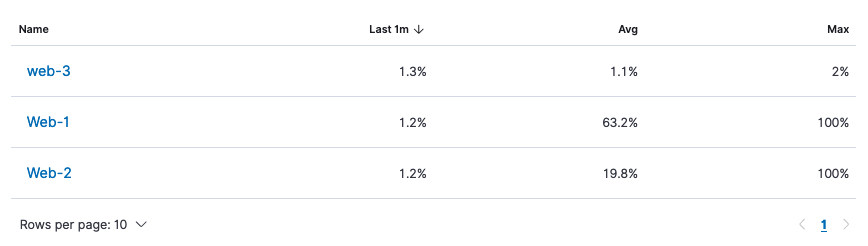
sysadmin@Web-1:~$ sudo apt update

sysadmin@Web-1:~$ sudo apt install stress

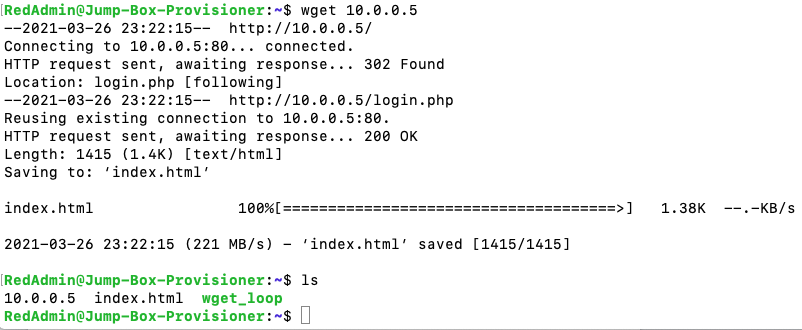
sysadmin@Web-1:~$ stress --cpu 1

After a few minutes look at Kibana again. Changed from Map to Table view: It shows high usage on Web-1, nearly 100%

Repeat for other two web servers:

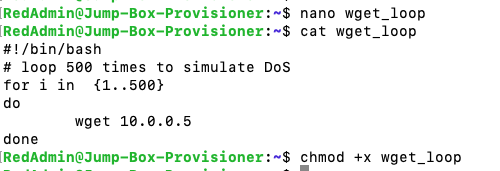
Stop the the stress command on all three. Now CPU usage back to baseline.

Wget DoS

From Jump-Box run wget once on command line:

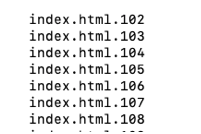
It saves index.html to current dir.

Create a script to run it 500 times:



Then run it: ./wget\_loop

See the same output as running on the command line but multiple times.

This generates many copies of the index file:

Change Kibana to View Network Traffic. Note two spikes from since ran the loop twice.

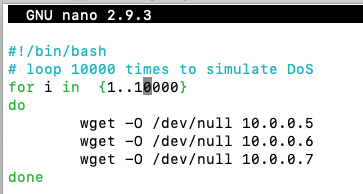


Improving wget

wget to a location: wget -O wget\_file 10.0.0.5

will only generate 1 file no matter how many times it’s run.

Changing the file location to /dev/null will not generate any files.

Modified script (also for fun changed it to run 10000 times:

After running the script:

