Microservice with ELK Stack Server

Proof of Concept (POC)

The Diagram flow:





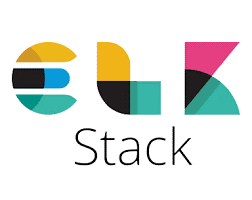
**Microservice server from runcloud**



**ELK Stack Server**



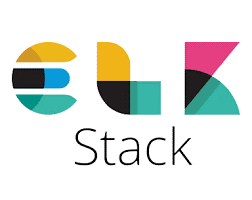
**Log file from microservice server transfer to ELK stack docker container using filebeat**



**Log file from microservice server transfer to GCP Bucket by using cronjob daily**

**Log file from GCP bucket sent to elk stack using gcsbeat**

The POC setup details:



**Microservice server**

**IP: 172.42.42.20**

**Services install: filebeat, filebeat apache modules**



**ELK docker container**

**IP: 172.42.42.200**

**Services install: Docker, Docker-compose, elasticsearch, Kibana**

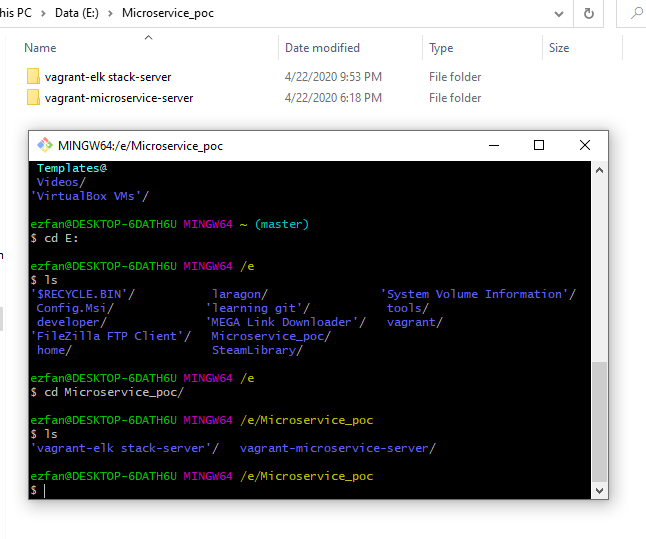


The Tools for this POC:-

1. Vagrant: - I use vagrant to save my vm so that it can use to another local user
2. Virtual box: - please install virtual box to store the vm
3. Docker: - docker for storing elasticsearch and kibana in the container.
4. Docker-compose: - use for combine the elasticsearch and kibana configuration in one single YAML file
5. Elasticsearch: - Raw data flows into Elasticsearch from a variety of sources, including logs, system metrics, and web applications. Data ingestion is the process by which this raw data is parsed, normalized, and enriched before it is indexed in Elasticsearch. Once indexed in Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data. From Kibana, users can create powerful visualizations of their data, share dashboards, and manage the Elastic Stack.
6. Kibana: - Searching, viewing, and visualizing data indexed in Elasticsearch and analyzing the data through the creation of bar charts, pie charts, tables, histograms, and maps. A dashboard view combines these visual elements to then be shared via browser to provide real-time analytical views into large data volumes
7. Google Cloud SDK: - google cloud sdk are used for send data to GCP bucket

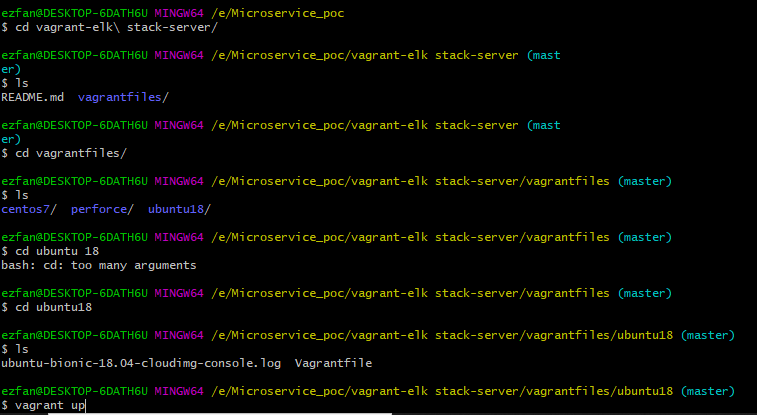
MANUAL: - **(\* NOTE make sure vagrant and virtual box are installed in your local device)**

1. First please enter the directory Microservice\_poc, as you can see there are two directory vagrant-elk-stack-server (ELK docker Server) & vagrant-microservice-server (Microservice server)

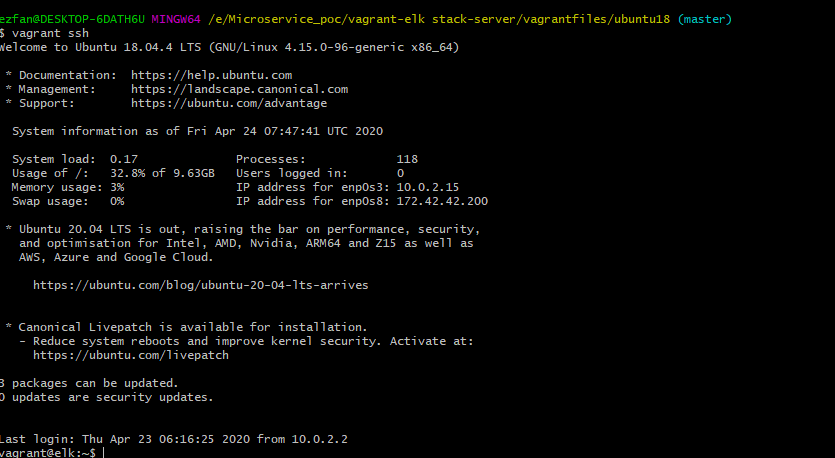


1. Enter vagrant-elk-stack-server directory -> vagrantfiles -> ubuntu 18 -> then vagrant up

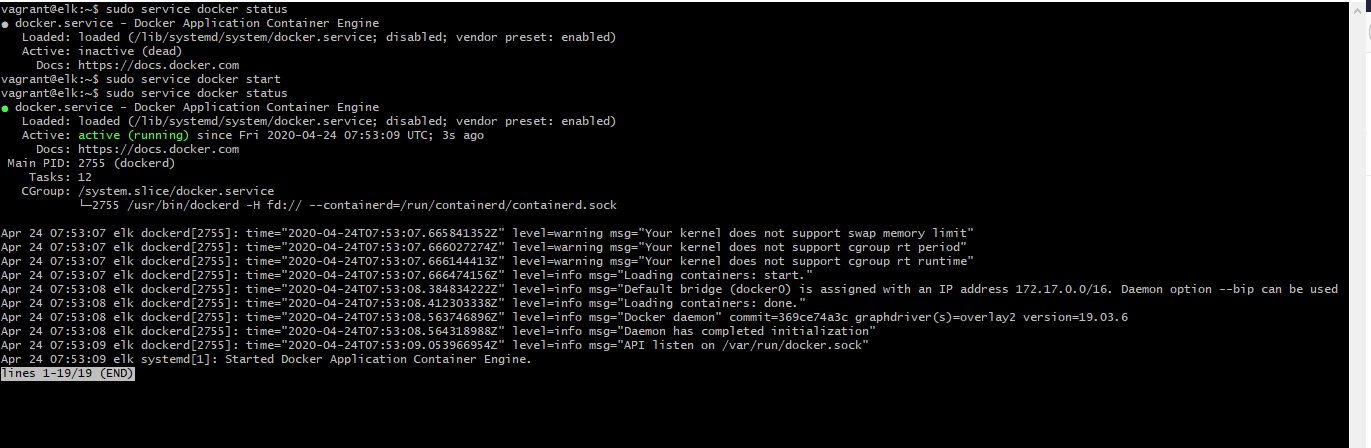
( to register the VM into virtualbox)



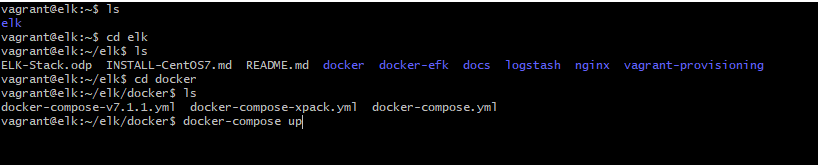
1. Then type vagrant ssh to enter into the elk stack vm



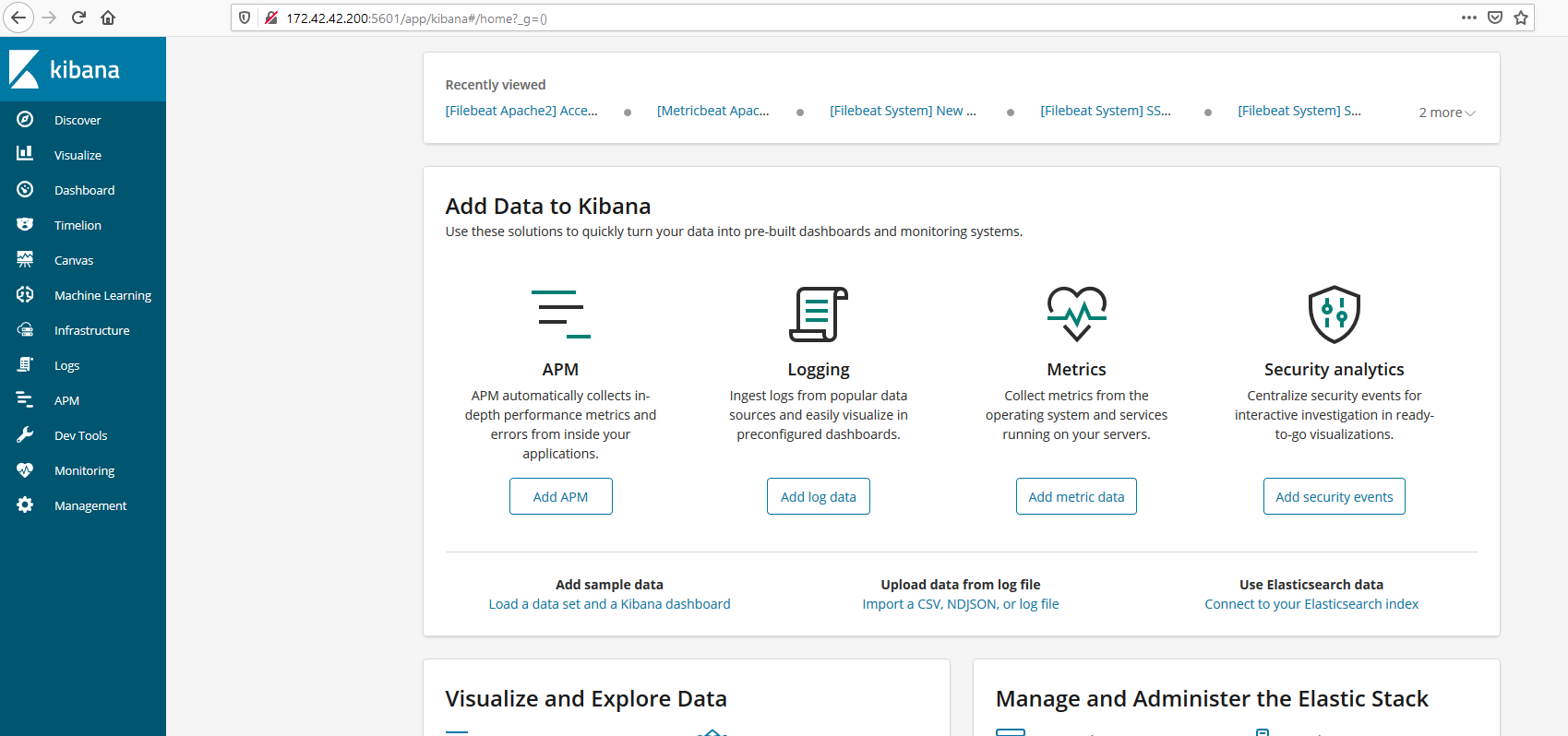
1. Make sure the docker service is up



1. Then enter directory elk -> docker then type docker-compose up to activate elasticsearch and kibana container



1. Make sure your local device can ping to 172.42.42.200, then go to browser and enter this url 172.42.42.200:5601, as you can see the kibana dashboard are successfully activate



1. **Now it time to activate the microservice server for filebeat configuration,** Enter vagrant-microservice-server directory then vagrant up

( to register the VM into virtualbox)



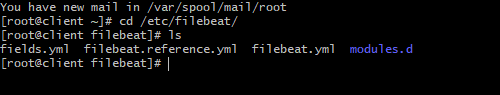
1. Now enter the microservice server type ssh [root@172.42.42.20](mailto:root@172.42.42.20) and the password is admin

Username: root

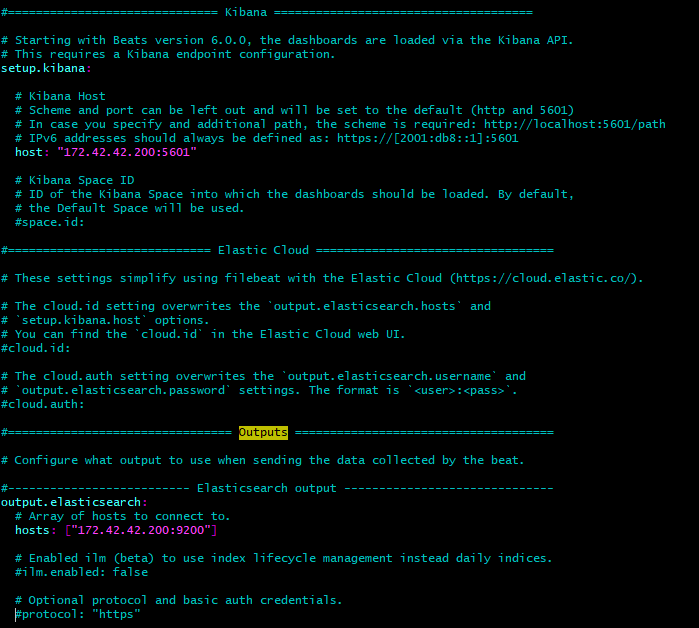
Password: admin



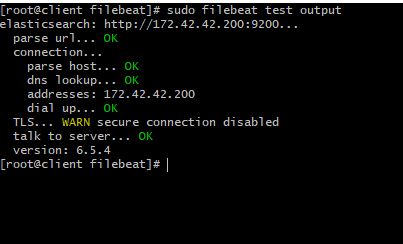
1. The filebeat config file are store in YML file at /etc/filebeat/filebeat.yml



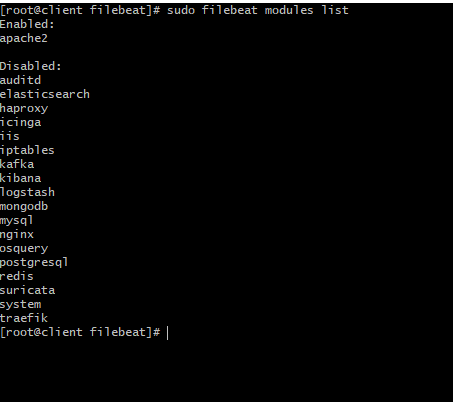
1. Now enter the yml file by using vim or other text editor as you can see I just config the file that filebeat need to sent the data to my destination ELK stack server 172.42.42.200:5601 (Kibana) & 172.42.42.200:9200 (Elasticserach), you need to change the ip here if the elk stack server ip change has change.



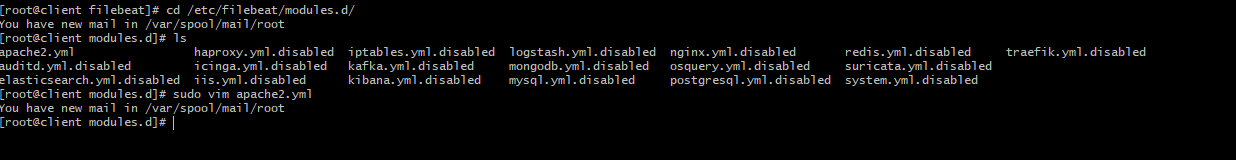
1. Now type sudo filebeat test output to make sure filebeat are connect to ELK stack server



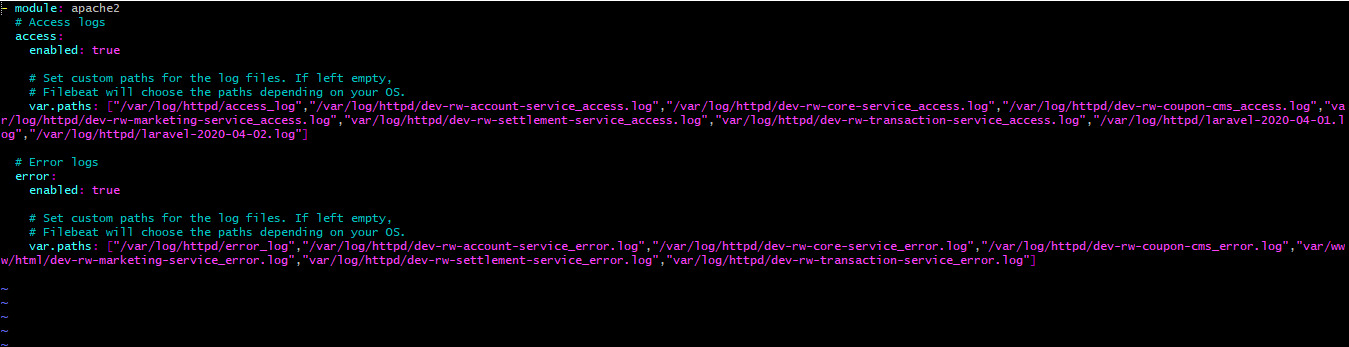
1. Now type sudo filebeat modules list to check the filebeat modules list that are enable. For this POC I activate the apache modules, you may try to activate other modules such as mysql, nginx, etc... type sudo filebeat enable mysql or else. For activate the modules



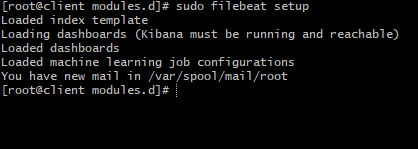
1. Next go to /etc/filebeat/modules.d/ to see the YML file for each modules in filebeat. If you not enable the modules the .yml file will be disable as shown in the picture below



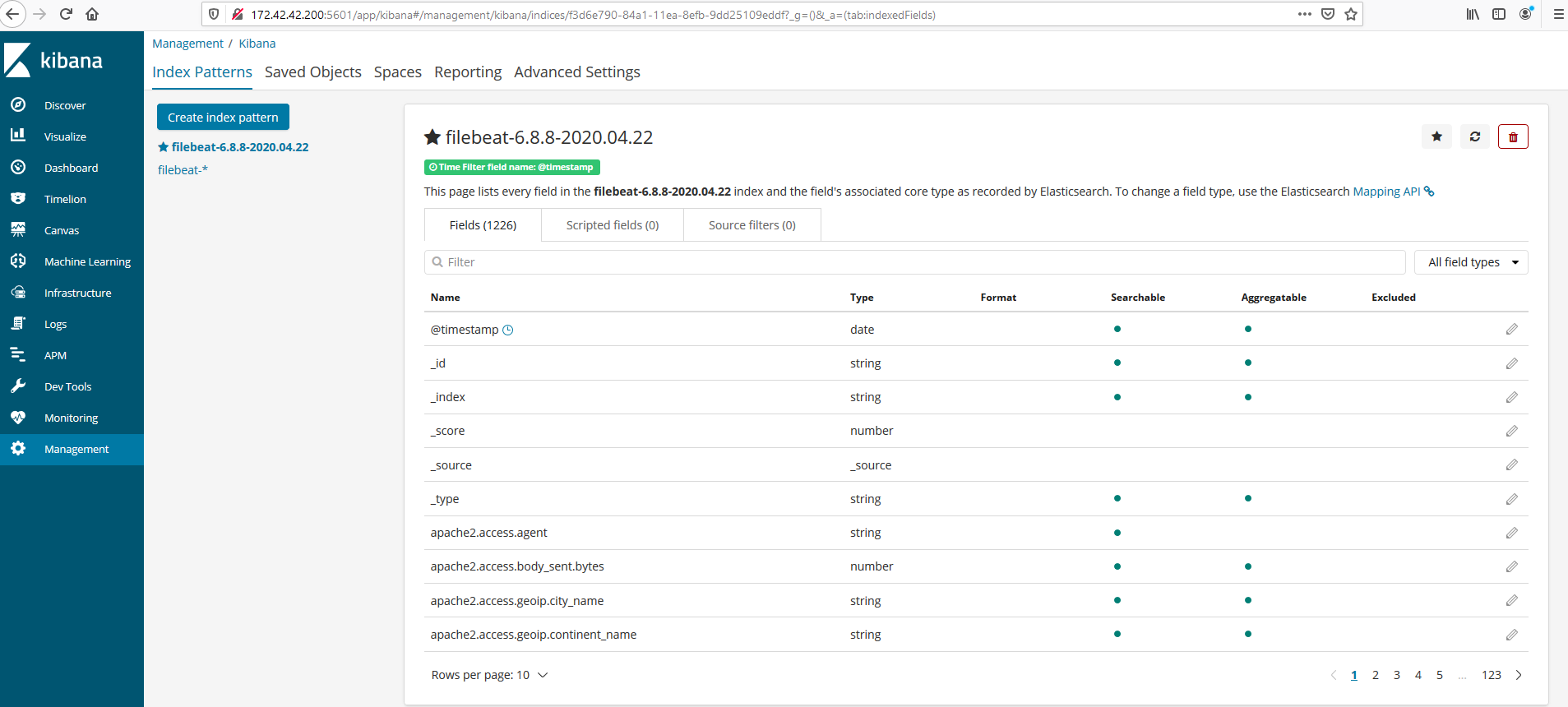
1. Now enter the apache2.yml file to config apache2 filebeat module, as you can see you need to include the log file path, for example the picture below shows the apache log file store at var/log/httpd/, you need to include the file location and put in here. They are access and error log for this module.



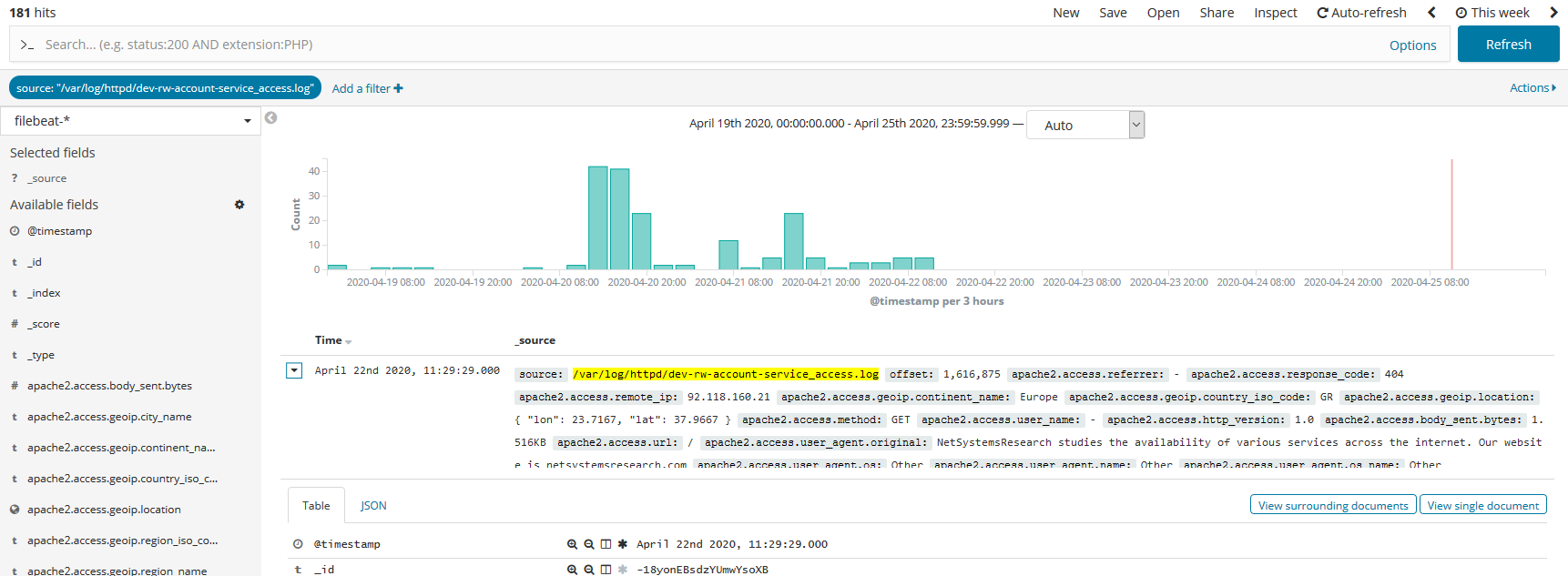
1. After that exit the file then type sudo filebeat setup, to install predefined indexes sent to the ELK stack server it may take a few second for the process loaded, wait until the loaded dashboard output



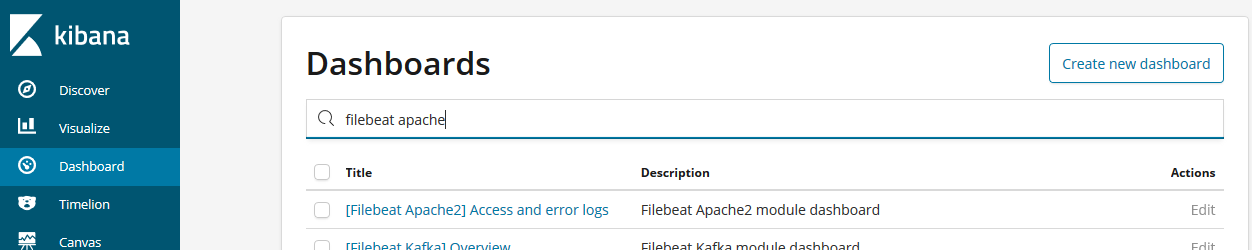
1. Now go back to kibana and click management at navbar, make sure the index pattern filebeat is insert in the kibana, you can create new index and delete the index



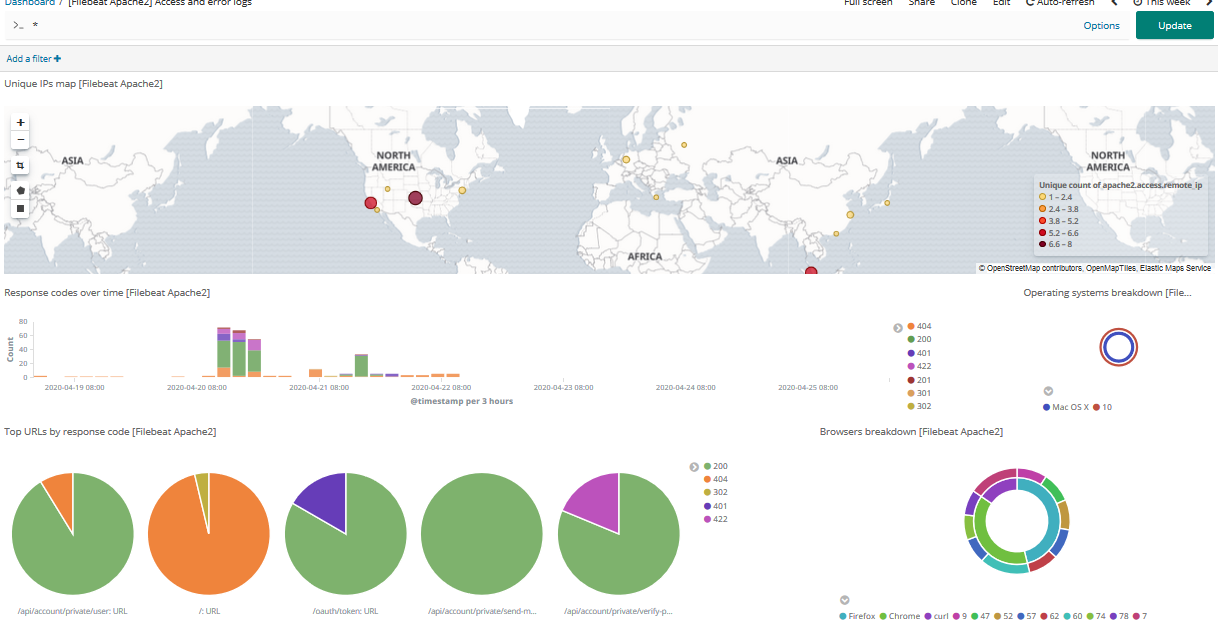
1. Now go to discover to see the log gather from filebeat, as you can see it can filter the log by date, data filtering and custom field you may choose for your own customization.



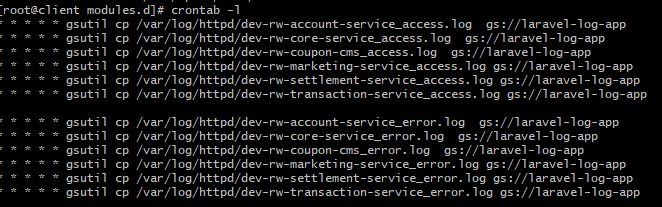
1. Next I also installed a dashboard template for apache access and error log, simply click dashboard and search for filebeat apache and click the Access and error logs

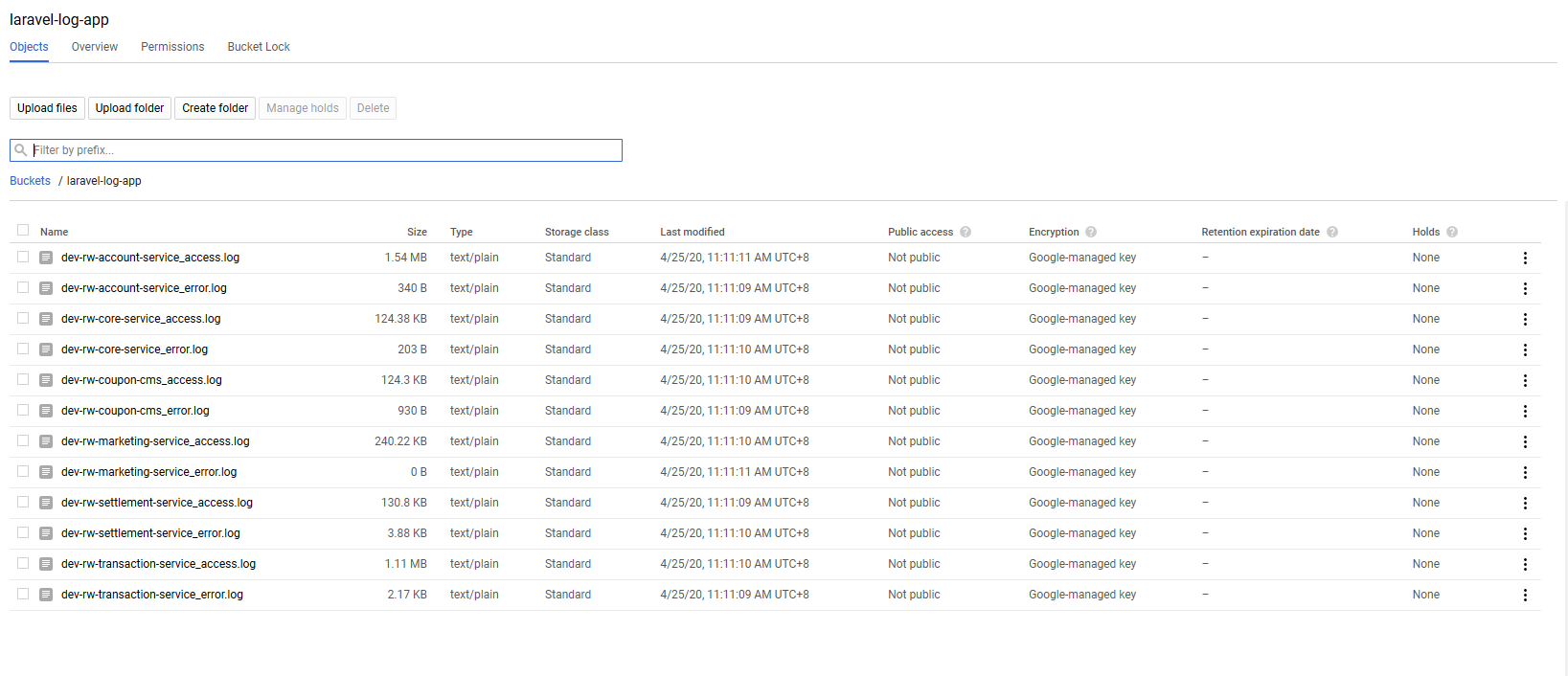


1. Here you can see the dashboard unique public ip map, response code over time and many more you may discover more for your own research



1. Now for log upload to gcp bucket im using google cloud sdk service and using gsutil command to upload the data. For scheduling im using cronjob to include my gsutil script in crontab. Please go to the microservice server and type crontab -l.

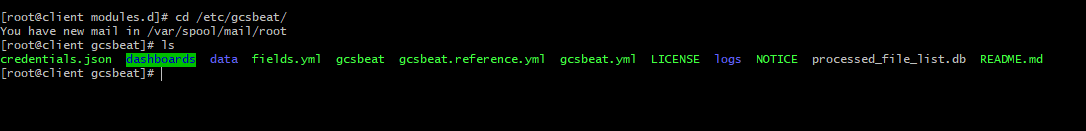


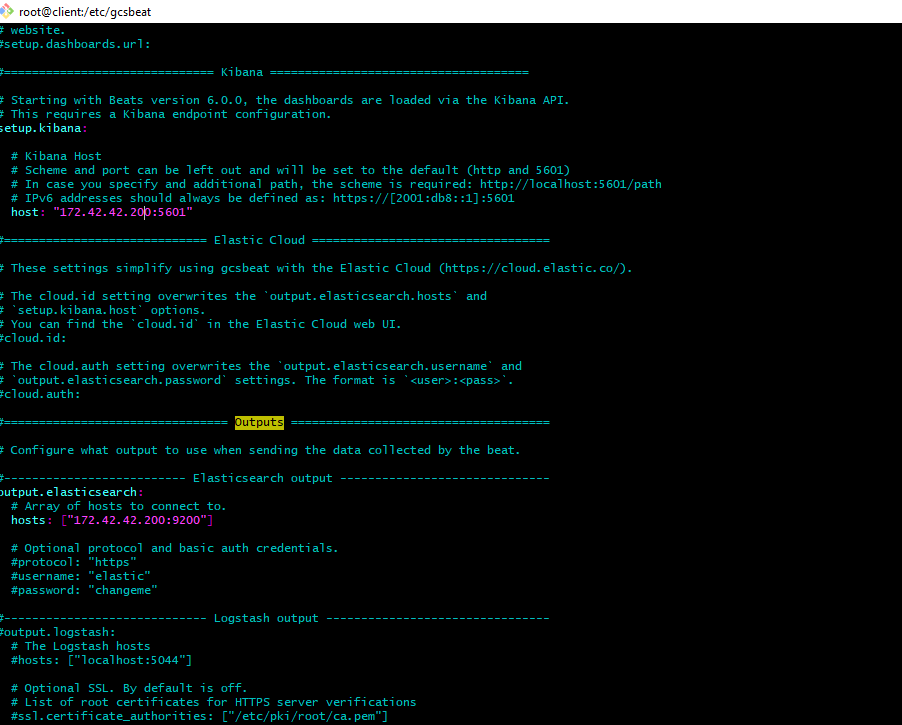


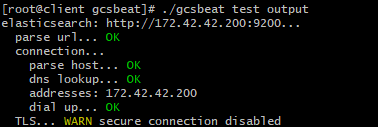
1. For the GCP bucket log to ELK stack server you may refer to

[**https://github.com/GoogleCloudPlatform/gcsbeat**](https://github.com/GoogleCloudPlatform/gcsbeat)**,** the process are the same as filebeat but you need to include credentials.json from your gcp into the microservice server for the authentication. I just installed the package it located at /etc/gcsbeat/

and the gcsbeat.yml file for the configuration







1. Run./gcsbeat -c gcsbeat.yml -e -v to start the gcsbeat processing.

