**Assessment 26– ELK**

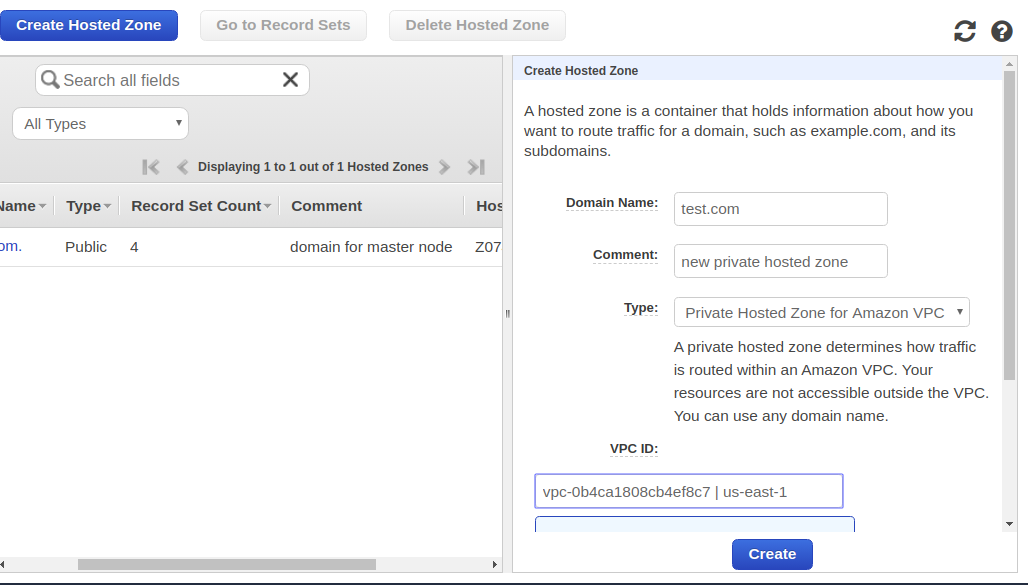
**Trainee Name : Gargi Sharma**

**Mentor Name : Mr. Akansh Gupta**

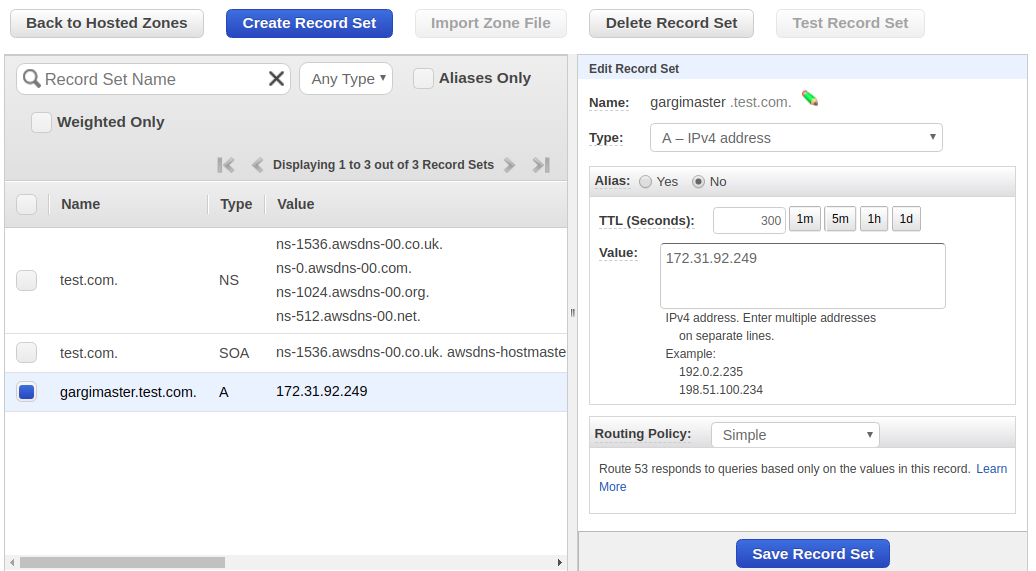
**College Name : UPES**

1. **Elasticsearch 2 node cluster(1st master+data, 2nd data+kibana node) setup through userdata.**

Configure a private hosted zone in AWS route 53 which will help in getting the ip for the elasticsearch setup.

****

Create A record set( Put the IP of a previous instance). We are creating A record because it directly maps with the IP.



Once done, note down the A name record for the master node so as to use it in the user data.

**Userdata to install elastic search on master node:**

*#!/bin/bash*

*sudo apt update*

*sudo apt install default-jdk -y*

*wget -P /tmp https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.0-amd64.deb*

*sudo dpkg -i /tmp/elasticsearch-7.6.0-amd64.deb*

*echo "path.data: /var/lib/elasticsearch*

*path.logs: /var/log/elasticsearch*

*node.master: true*

*node.data: true*

*node.ingest: true*

*network.host: [\"localhost\", \"gargimaster.test.com\"]*

*http.port: 9200*

*discovery.seed\_hosts: [\"gargimaster.test.com\"]*

*node.name: garging-1*

*cluster.name: g-cluster*

*cluster.initial\_master\_nodes: ["garging-1"]" > elasticsearch.yml*

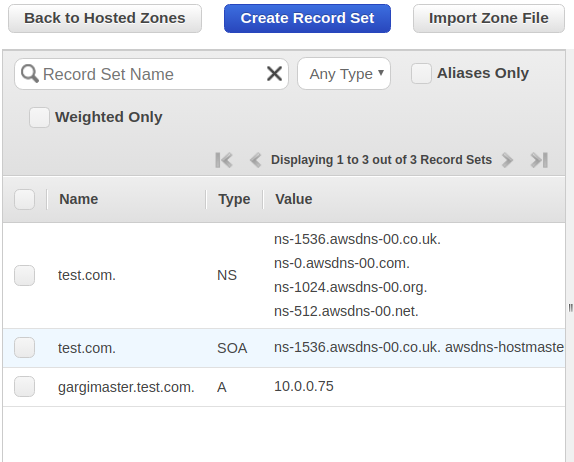
*sudo mv elasticsearch.yml /etc/elasticsearch/elasticsearch.yml*

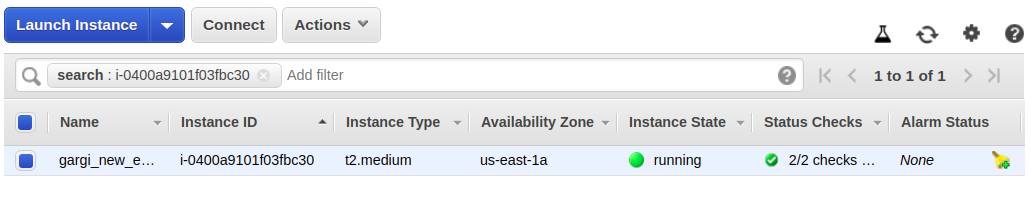
*sudo systemctl enable elasticsearch*

*sudo systemctl daemon-reload*

*sudo systemctl start elasticsearch*

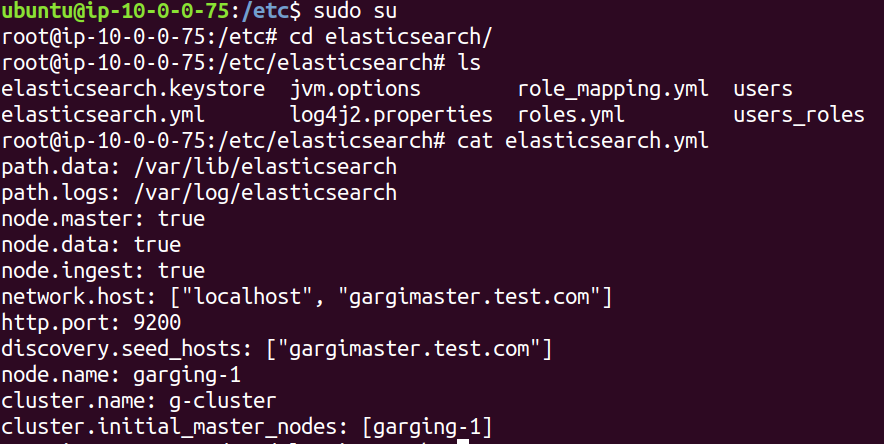
Now launch an ec2 instance with this userdata. Since elasticsearch works on port 9200, open this port in the security group of the instance. Also edit the A name record with private IP of the instance.



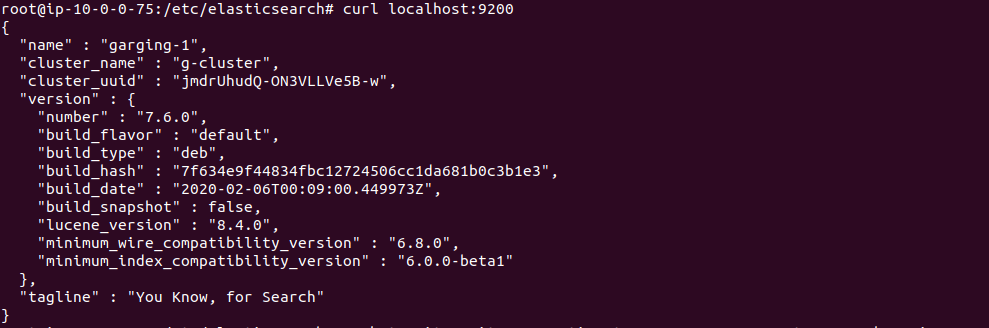


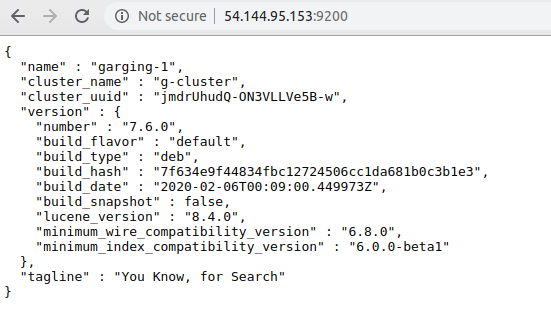
Now ssh into the instance.

Cat /etc/elasticsearch/elasticsearch.yml:



Curl localhost:9200

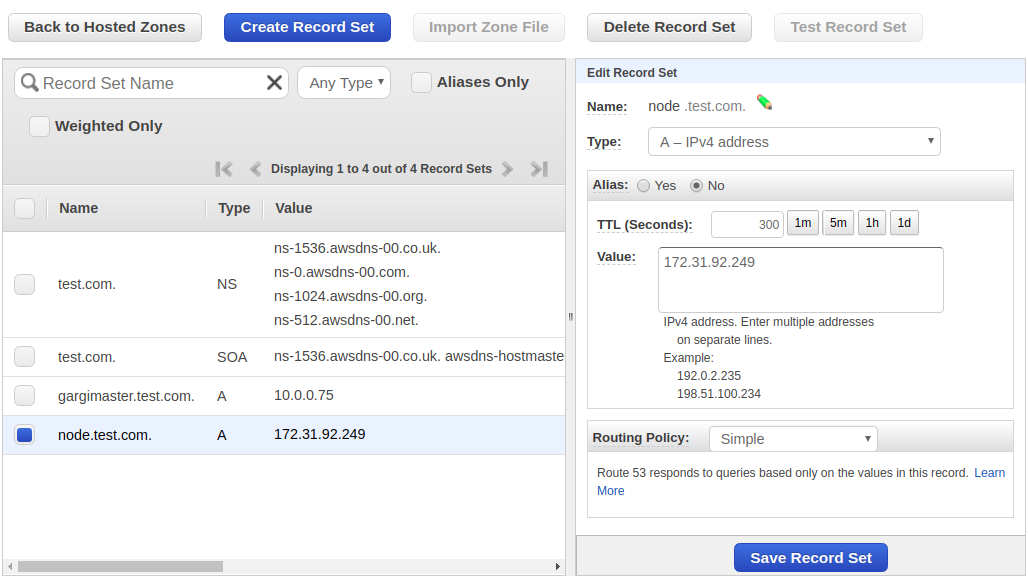




Elasticsearch is up and running in master node.

Now we will setup the data node where both elasticsearch and kibana have to be installed.

First create an A name for the data node in the same private hosted zone.



**Userdata for data node:**

*#!/bin/bash*

*sudo apt update*

*sudo apt install default-jdk -y*

*wget -P /tmp https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.0-amd64.deb*

*sudo dpkg -i /tmp/elasticsearch-7.6.0-amd64.deb*

*echo "path.data: /var/lib/elasticsearch*

*path.logs: /var/log/elasticsearch*

*node.master: true*

*node.data: true*

*node.ingest: true*

*network.host: [\"localhost\", \"node.test.com\"]*

*http.port: 9200*

*discovery.seed\_hosts: [\"gargimaster.test.com\"]*

*node.name: nodegargi-1*

*cluster.name: g-cluster*

*cluster.initial\_master\_nodes: ["nodegargi-1"]" > elasticsearch.yml*

*sudo mv elasticsearch.yml /etc/elasticsearch/elasticsearch.yml*

*sudo systemctl enable elasticsearch*

*sudo systemctl daemon-reload*

*sudo systemctl start elasticsearch*

*wget https://artifacts.elastic.co/downloads/kibana/kibana-7.6.0-amd64.deb*

*dpkg -i kibana-7.6.0-amd64.deb*

*echo 'server.port: 5601*

*server.host: "0.0.0.0"*

*elasticsearch.hosts: ["http://localhost:9200"]' > kibana.yml*

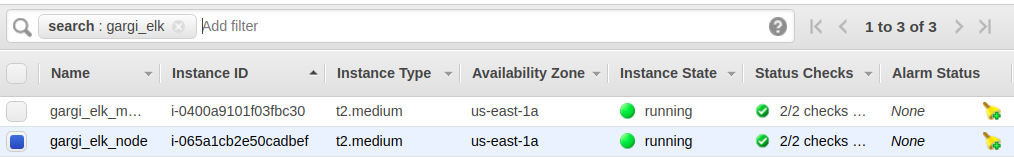
*sudo mv kibana.yml /etc/kibana/kibana.yml*

*sudo systemctl enable kibana*

*sudo systemctl daemon-reload*

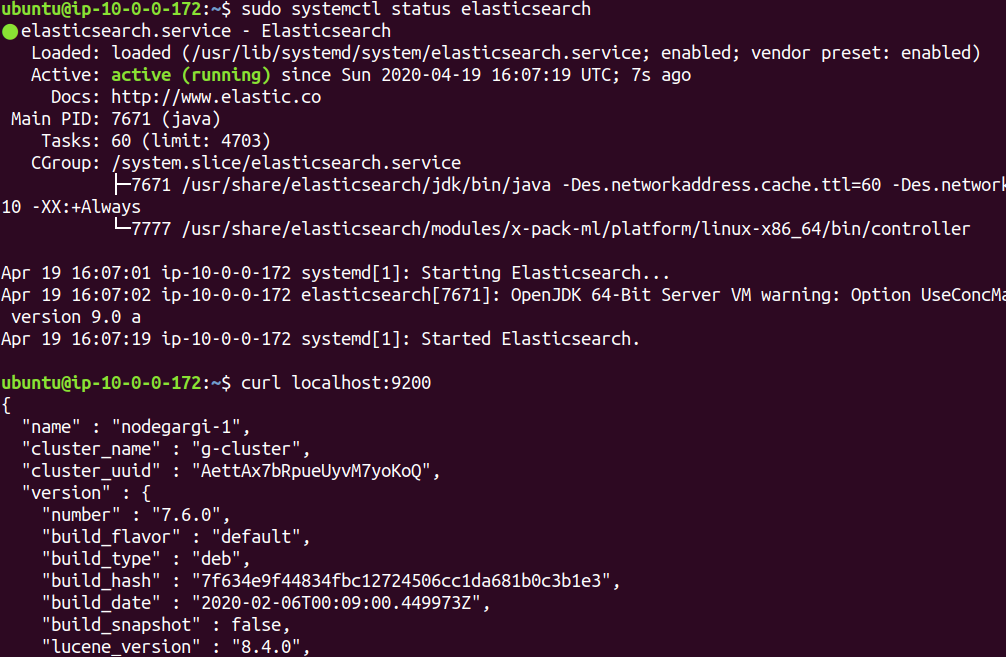
*sudo systemctl start kibana*

Now launch a t2.medium instance with this user data and make sure to allow 5601 and 9200 ports for kibana and elasticsearch respectively.

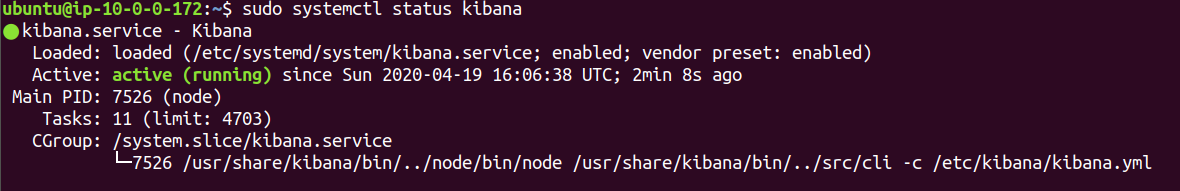


Once done, ssh into the instance and curl localhost:9200 for elasticsearch and localhost:5601 for kibana.

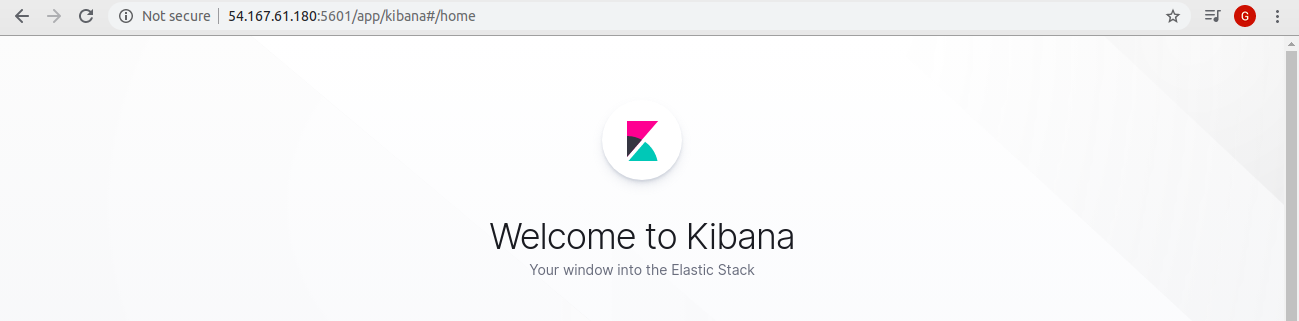
Elasticsearch:



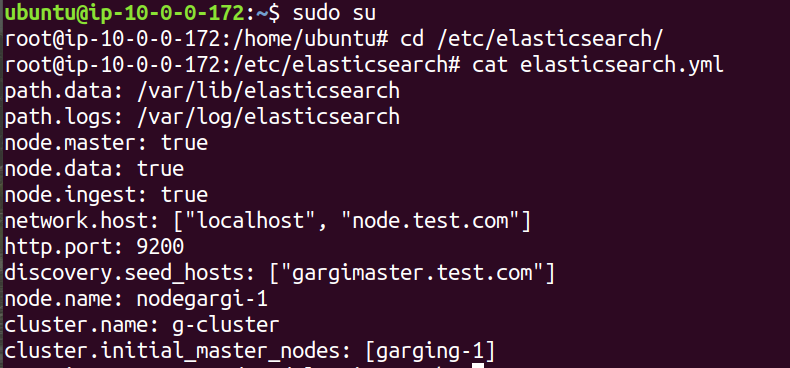
Kibana:



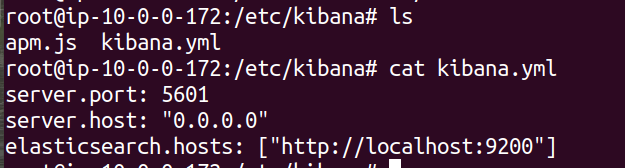
Kibana Dashboard:



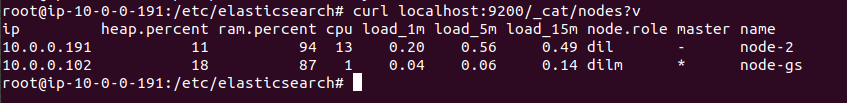
Cat /etc/elasticsearch/elasticsearch.yml



Cat /etc/kibana/kibana.yml:



*Both master and data nodes are up and running. Check if both the nodes are connected.*

**

1. **Write regex for Apache and Nginx logs**

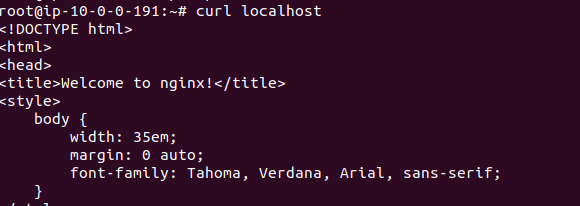
**Install Apache and Nginx and enable error and access logs for both**

**Make separate conf for apache and nginx logs and include them in td-agent.conf**

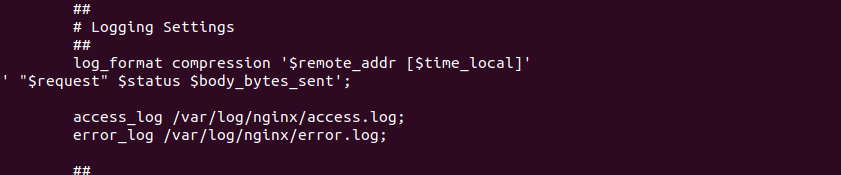
**Parse the logs with proper access and error logs format (check on web for different fields on logs)**

**Make a separate index for nginx and apache logs on kibana also.**

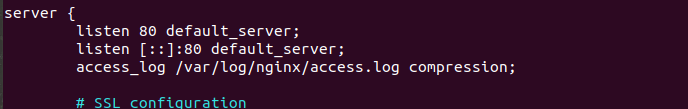
Install nginx:



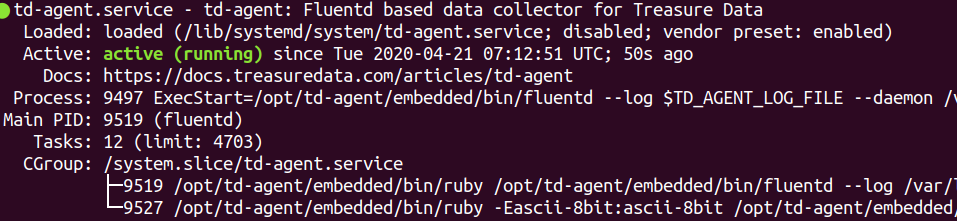
Edit /etc/nginx/nginx.conf and edit log format:



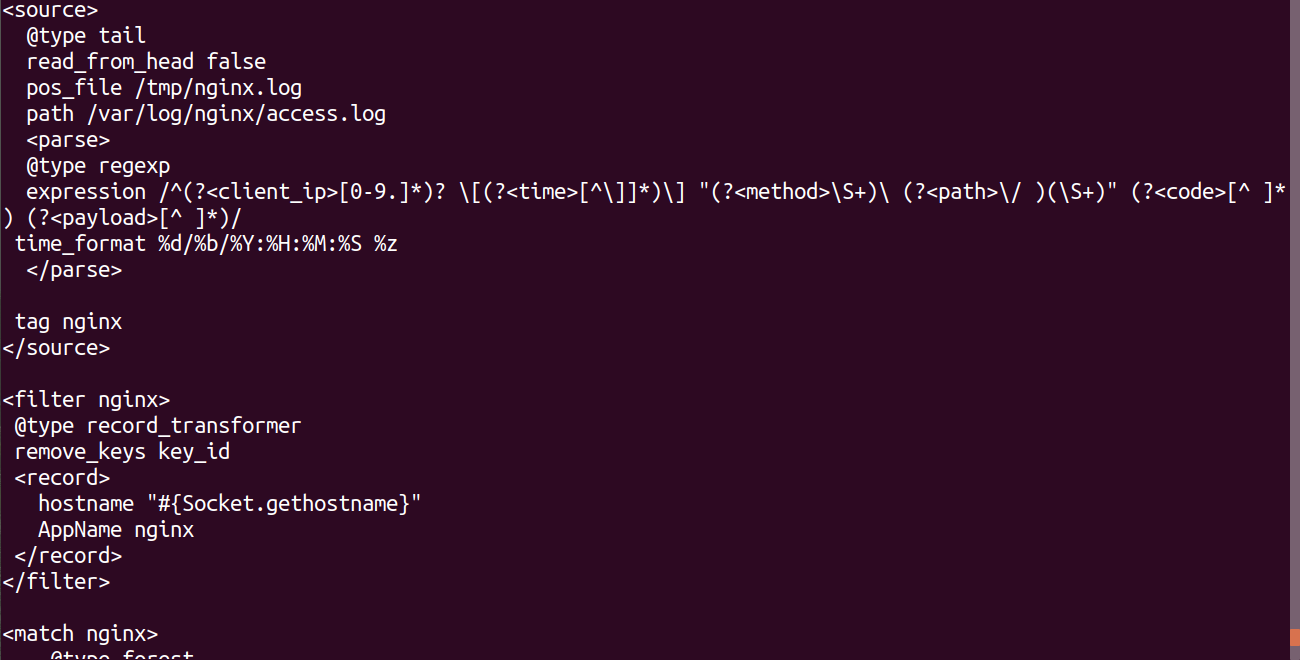
Now edit /etc/nginx/sites-available/default and add access logs:

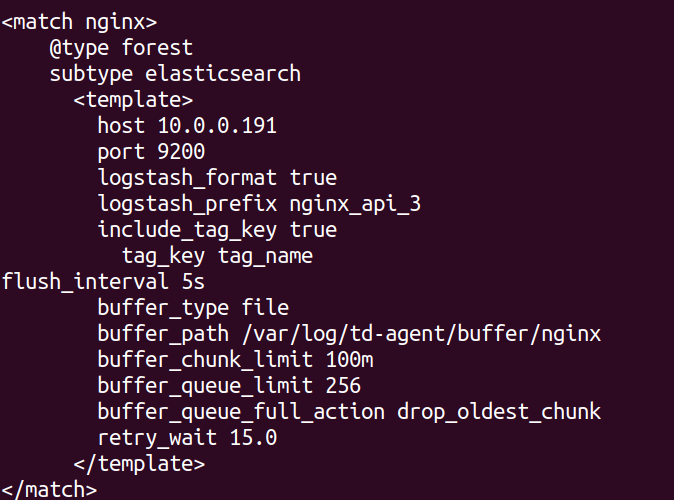


Install tg-agent(distribution of fluent-d)



Now edit/etc/tg-agent/td-agent.conf file:





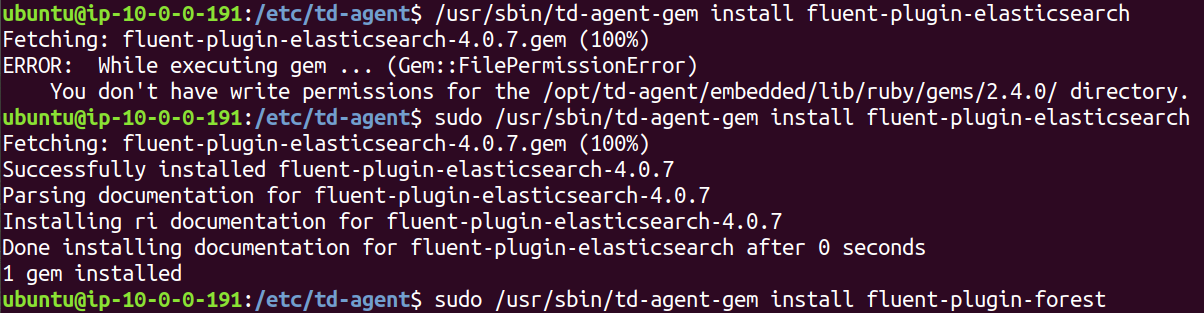
Then install 2 plugins:

sudo apt-get install make libcurl4-gnutls-dev --yes

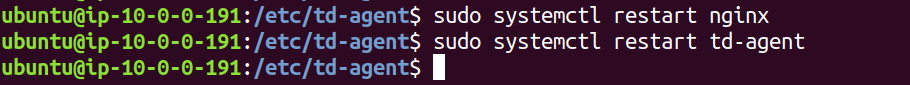
sudo apt-get install build-essential

/usr/sbin/td-agent-gem install fluent-plugin-elasticsearch

/usr/sbin/td-agent-gem install fluent-plugin-forest



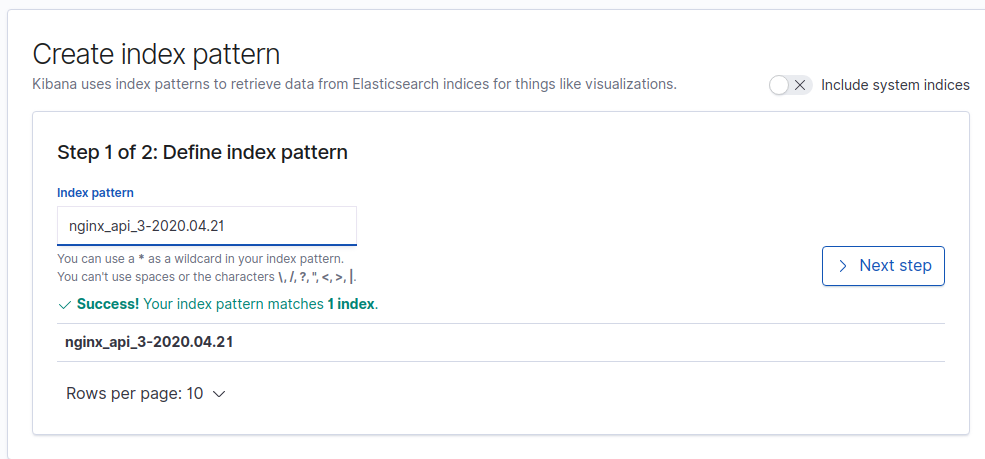
Now restart nginx and td-agent.

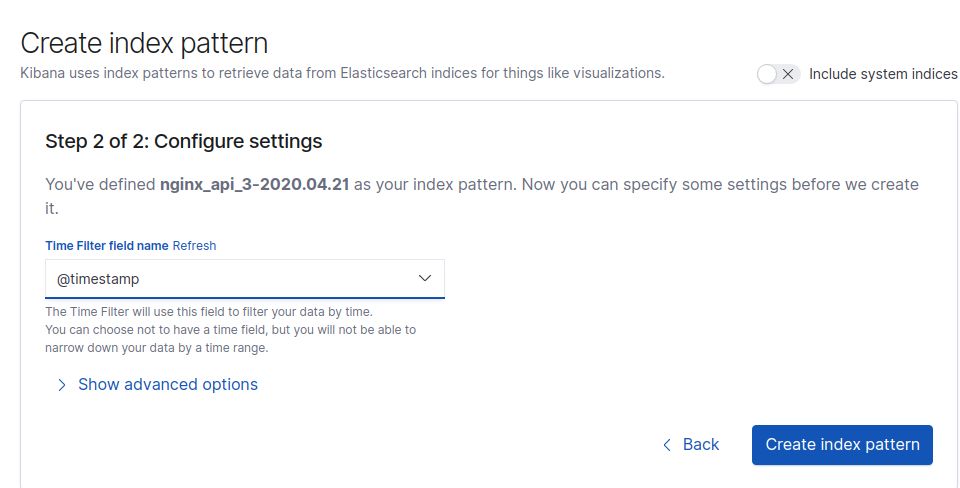


After this, run chmod 777 /var/log/nginx/access.log .

Restart nginx and td-agent.

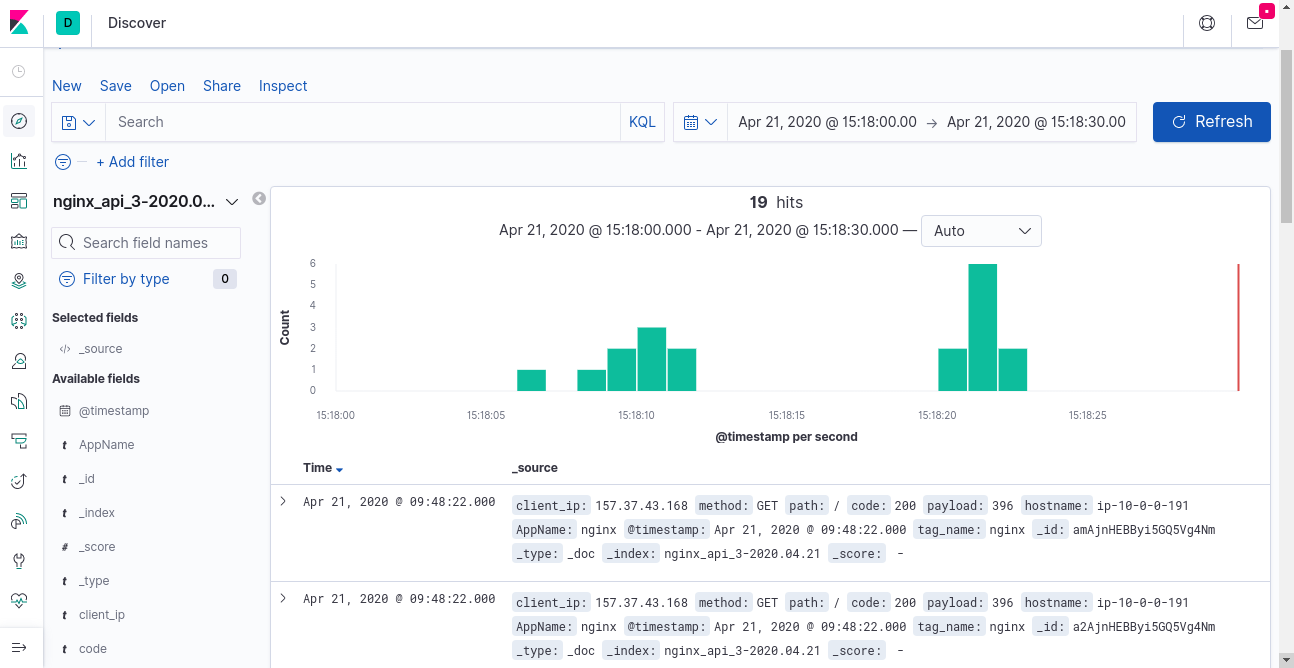
Now move to Kibana UI and create an index pattern and create an index pattern.





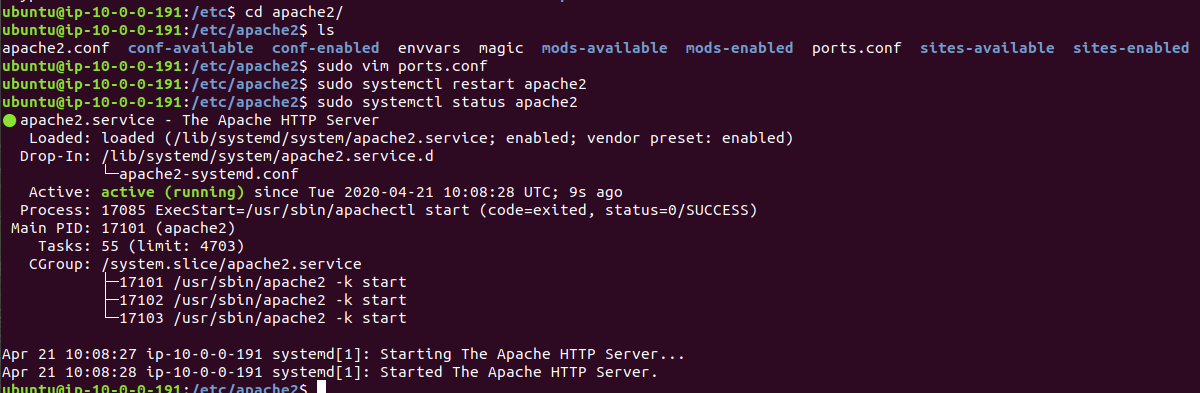
After creating the index pattern, go to discover .

Now hit nginx multiple times and check the kibana dashboard.

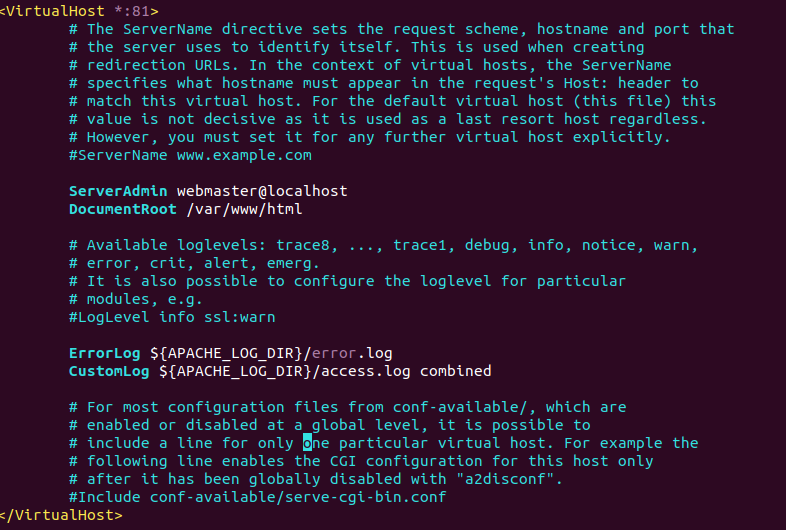


Apache Logs:

Install apache and change the port of apache in /etc/apache2/port.conf:

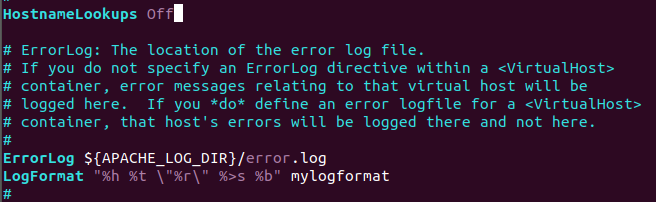


Now edit the port in /etc/apache2/default.conf and change the port here also:

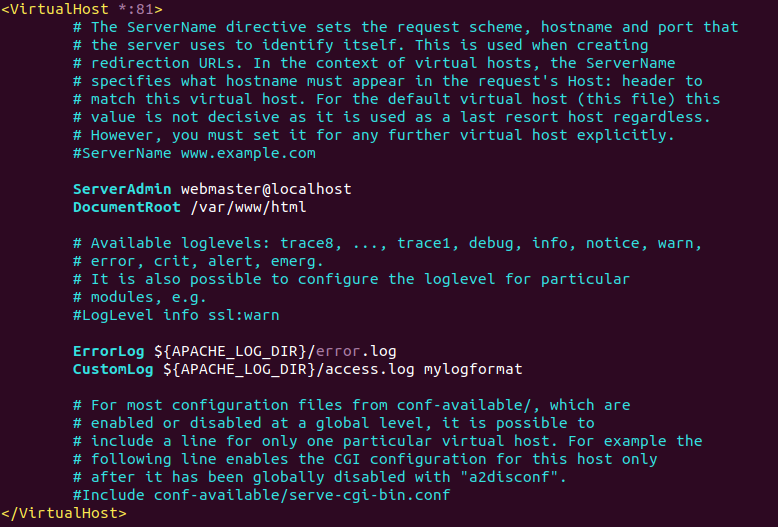


Now add log format to apache:

In apache2.conf:



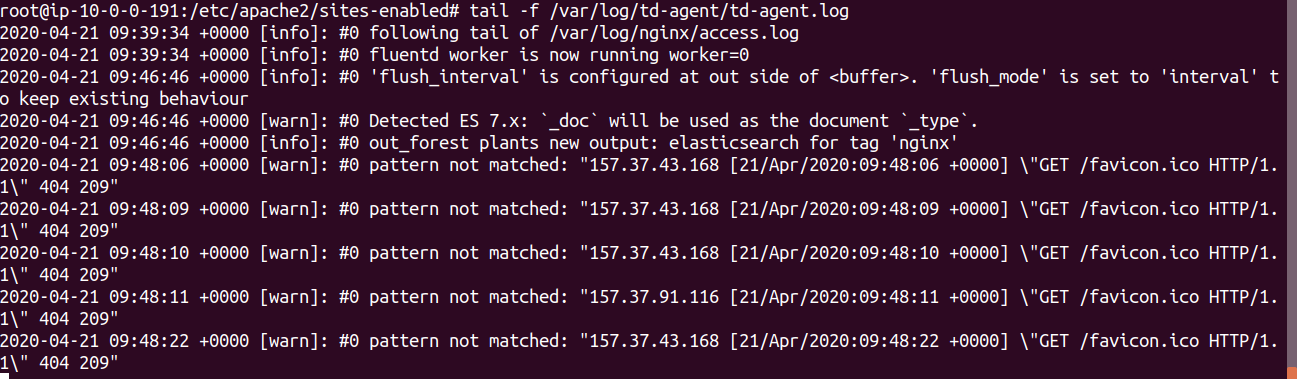
In sites-enabled/default.conf:



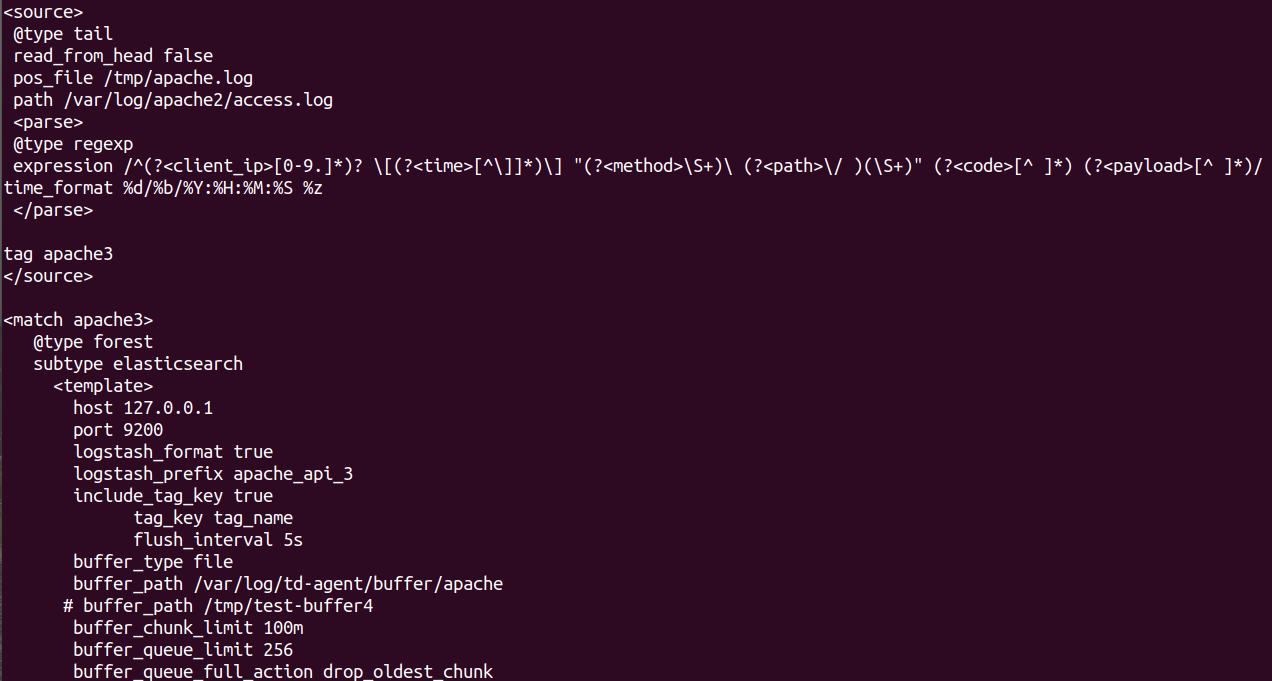
Now change the permissions:

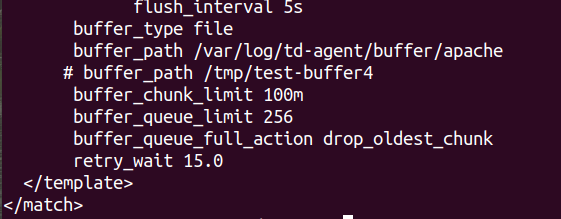


Now check logs:

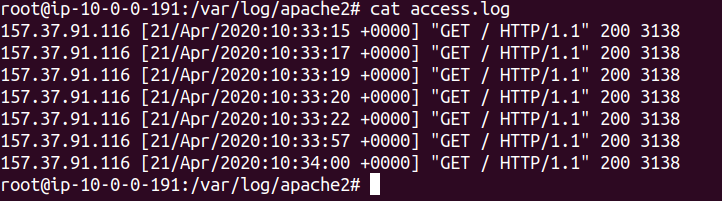


Add apache block to the td-agent.conf:

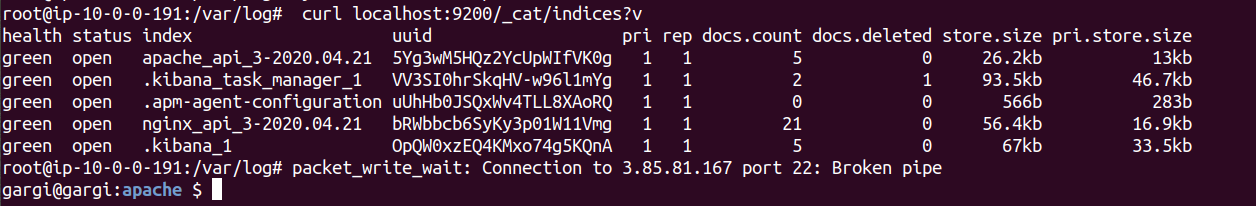




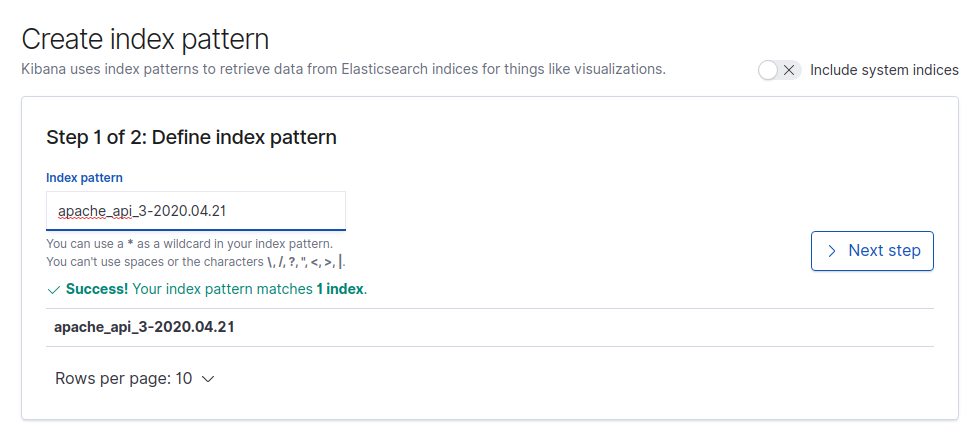
Now restart apache2 and td-agent and check logs in /var/log/apache2/access.log:

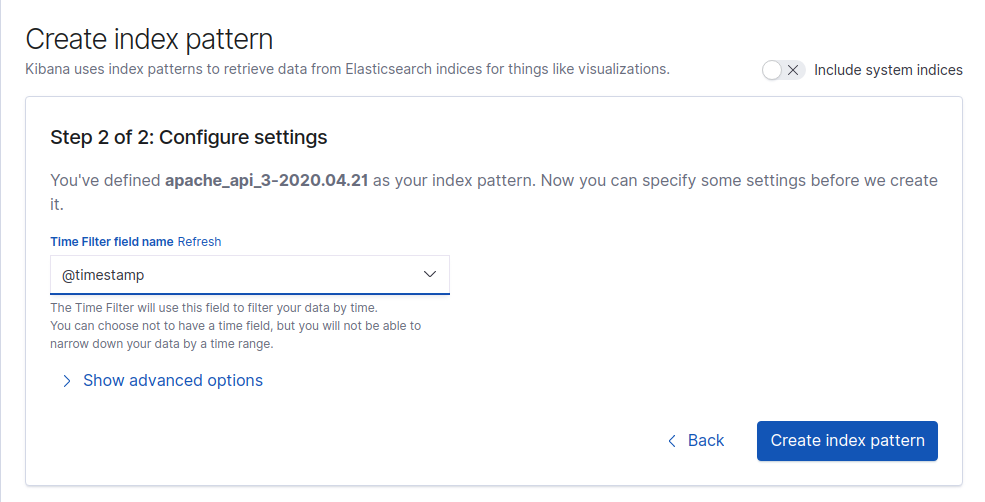


Now check for indices:

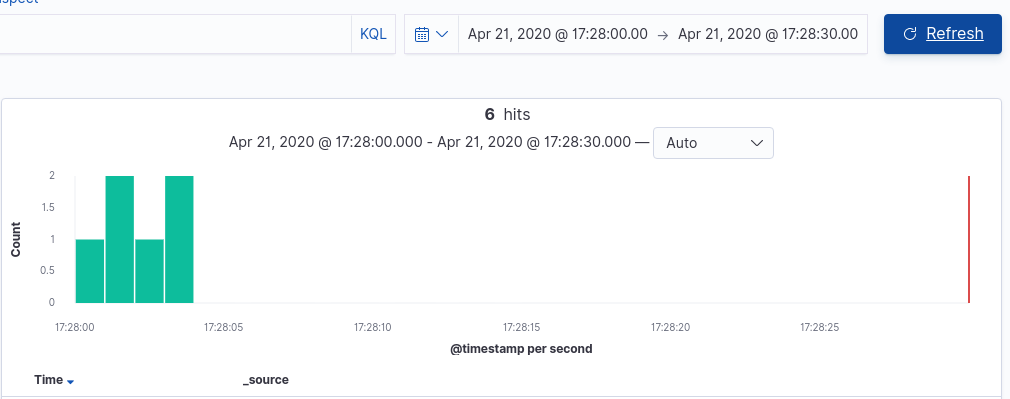


Now go to kibana dashboard and create index:





Now go to discover and check for grapes. Hit apache website and then check the dashboard.

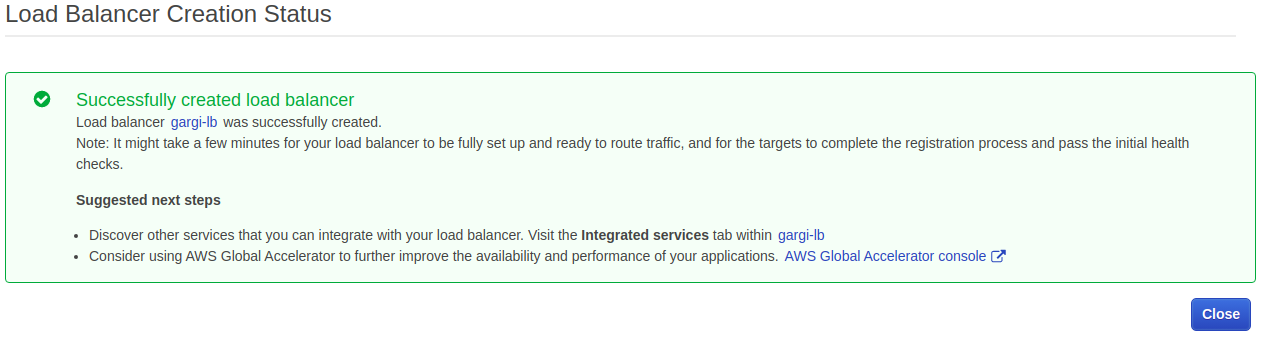


1. **Alb logs to s3 then to elk**

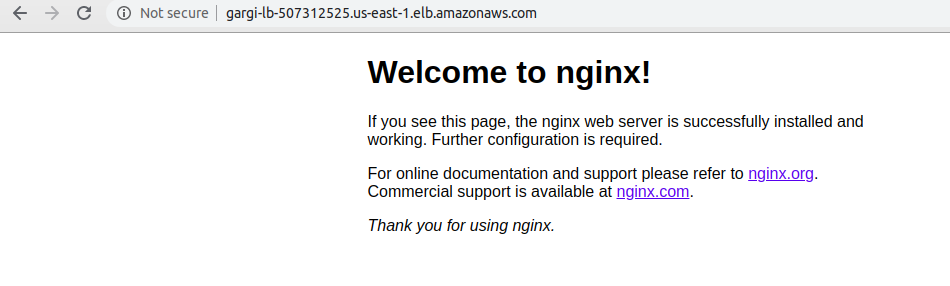
**Enable alb logs on s3**

**Using td-agent send the logs on s3 to elk (search for this plugin elb-access-log plugin)**

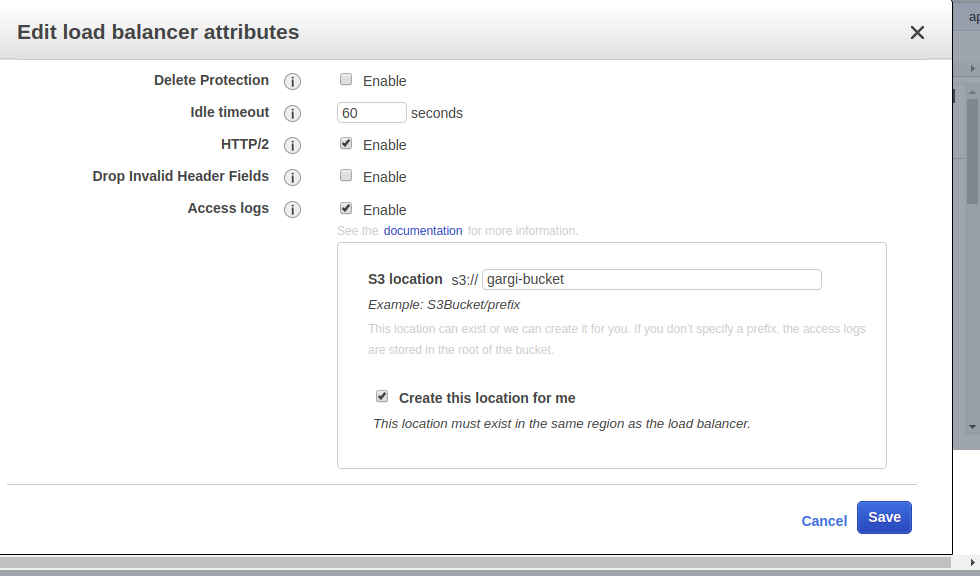
Create an application load balancer:



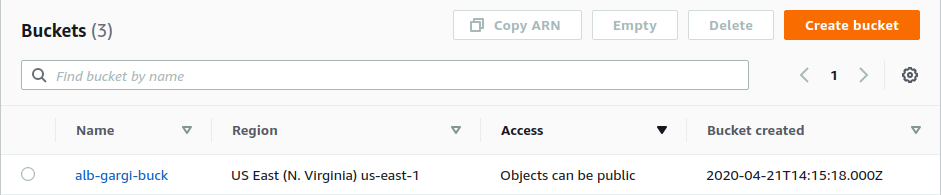
Hit the DNS of the the ALB



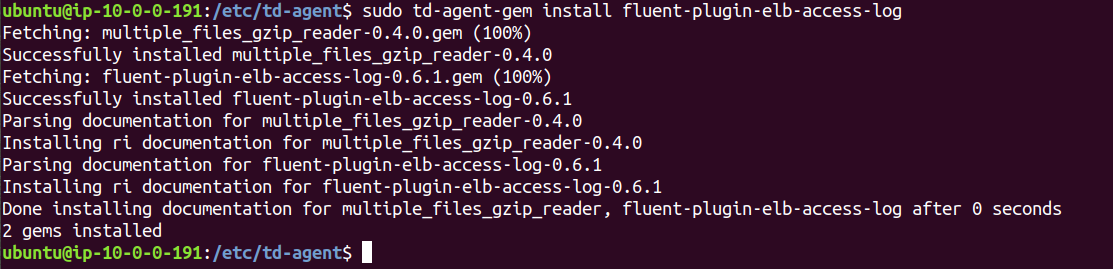
Edit the ALB and create a s3 bucket through it.



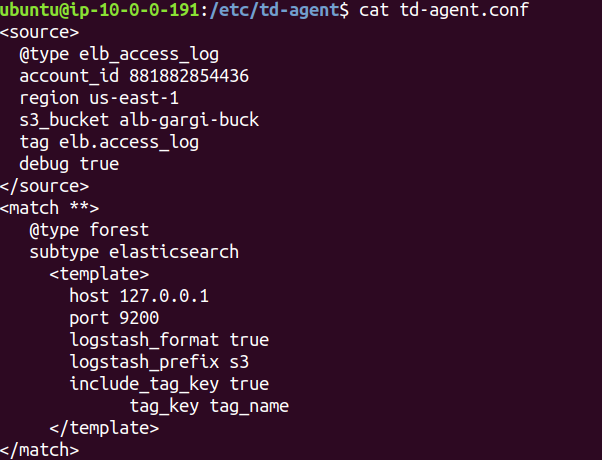
S3 bucket:



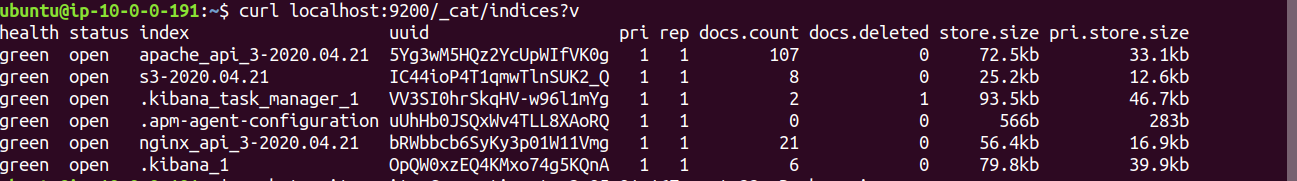
Now install fluent-plugin-elb-access-log



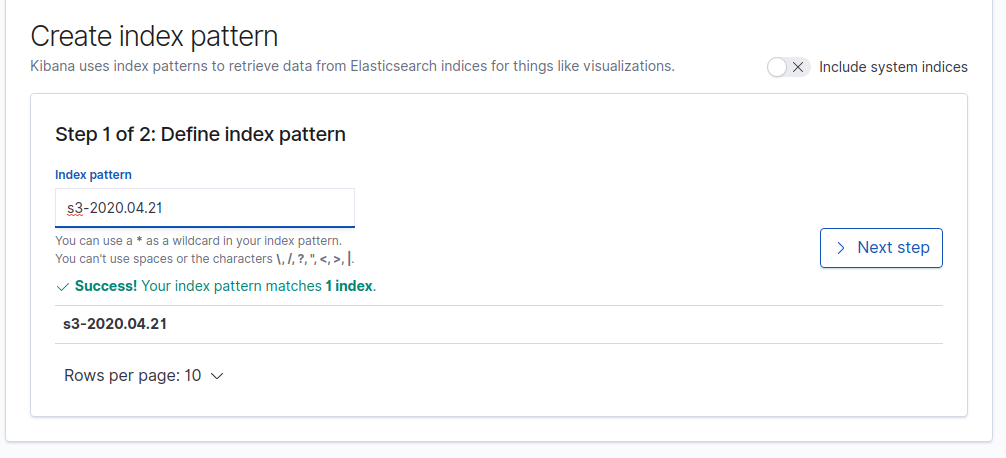
Now edit td-agent.conf:

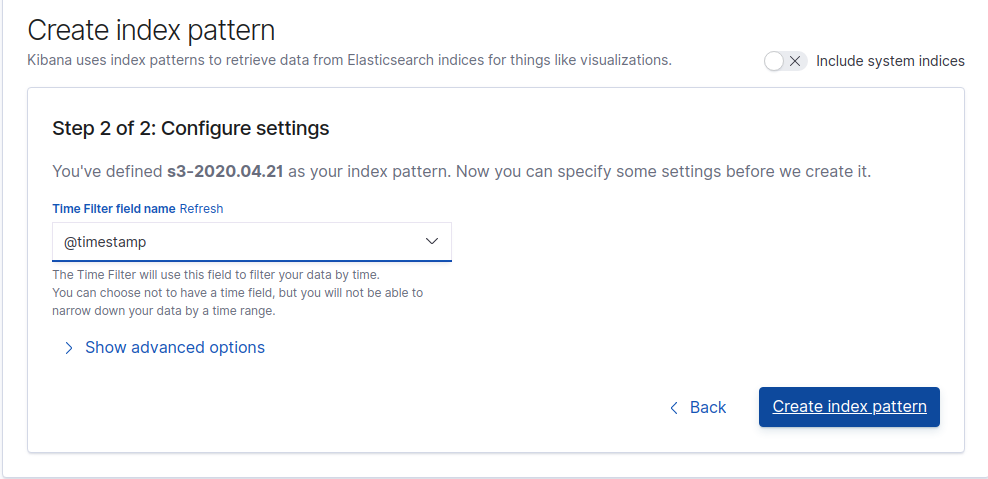


Now restart nginx, td-agent and elasticsearch and run the following command. It should show the index of s3.



Now go to kibana dashboard and create index for s3 bucket:





Now go to discover and select the newly created index. Then hit the ALB DNS multiple times to see the log output on kibana dashboard:

