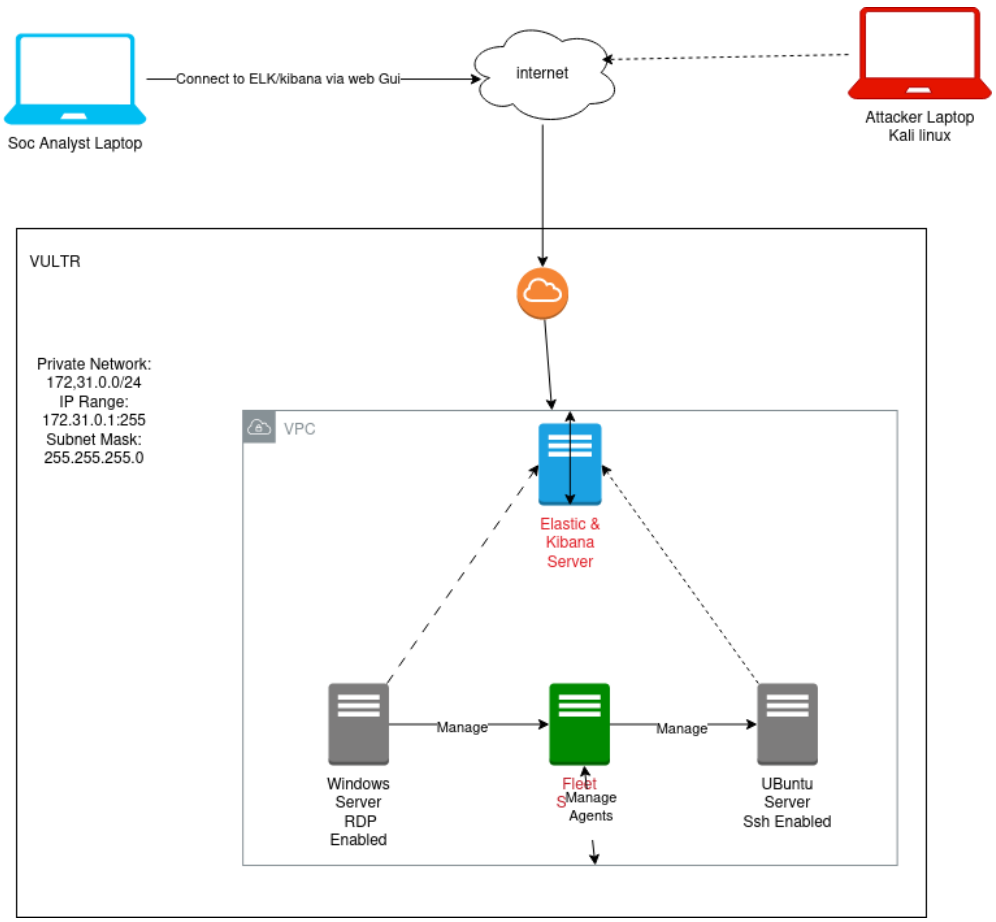
HASSAN MAZHAR_243293_PROJECT_REPORT

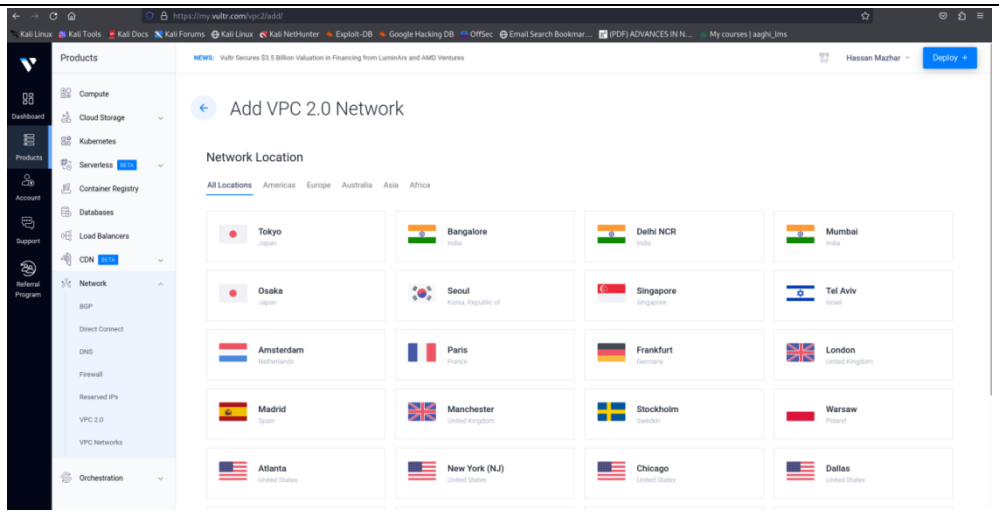
✧ LOGICAL DIAGRAM OF ROJECT IS:



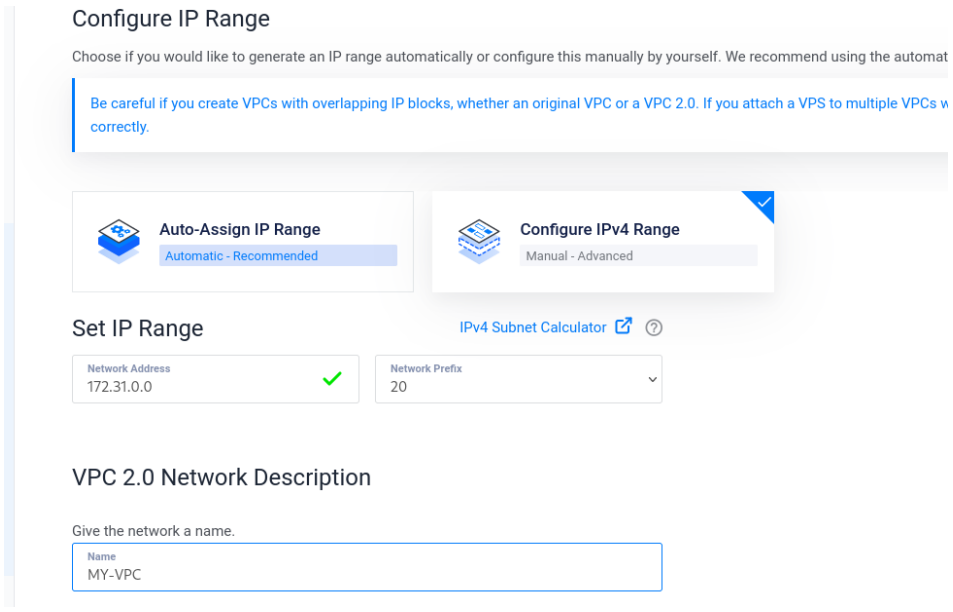
✧

✧ I FOLLOW THE ABOVE IMAGE IN THE BELOW STEPS

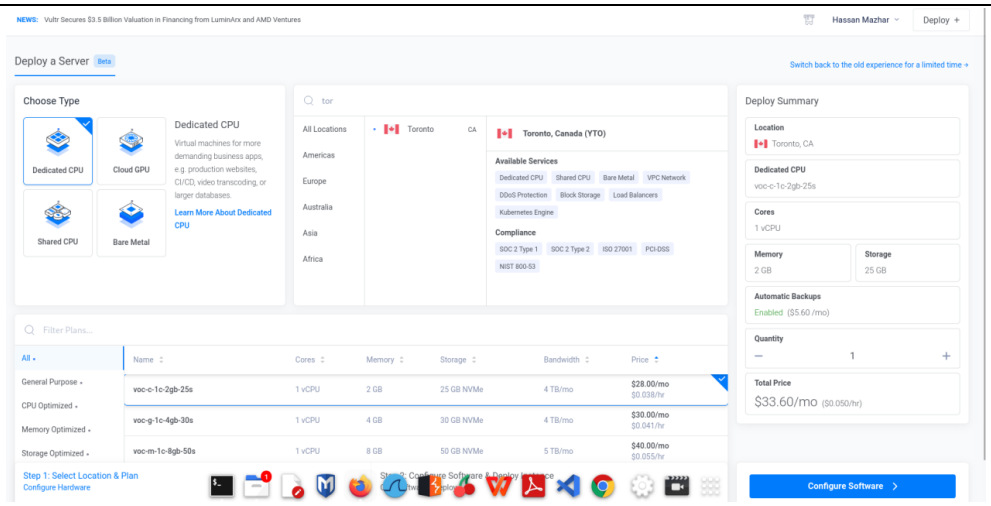
first of all setup the vultr account by giving any credit card number which are not in use like in my case I am using Sadapay.
after that the dashboard looks like in the below image, and then create an VPC(virtual private network first)



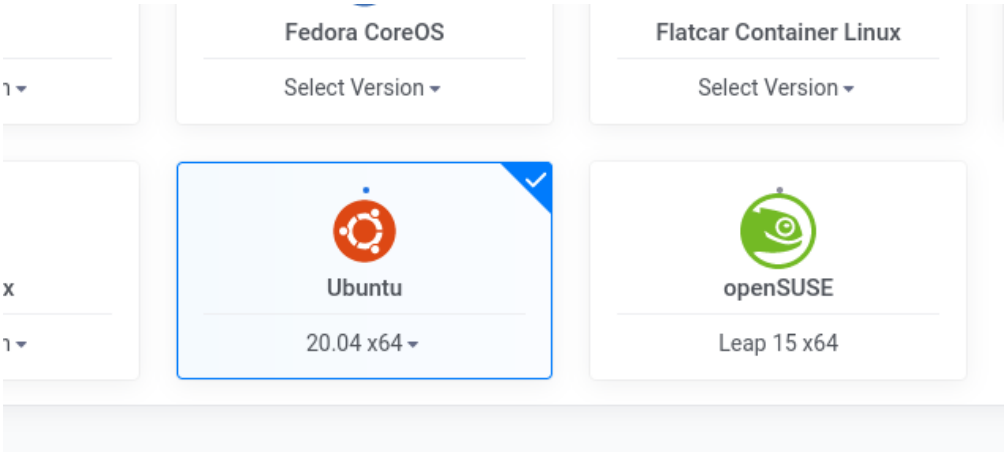
Now Configure an ip address manually by assigning an ipaddress in the below image:



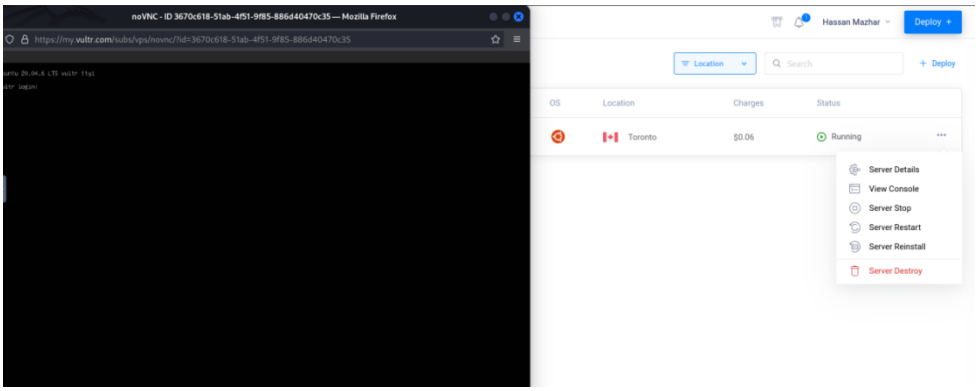
- **configure a server:**
  - **choosing the location like i choose tornoto**
  - **assigning the memory**



- **choose machine:**
  - **i choose ubuntu where i install my elastic search**

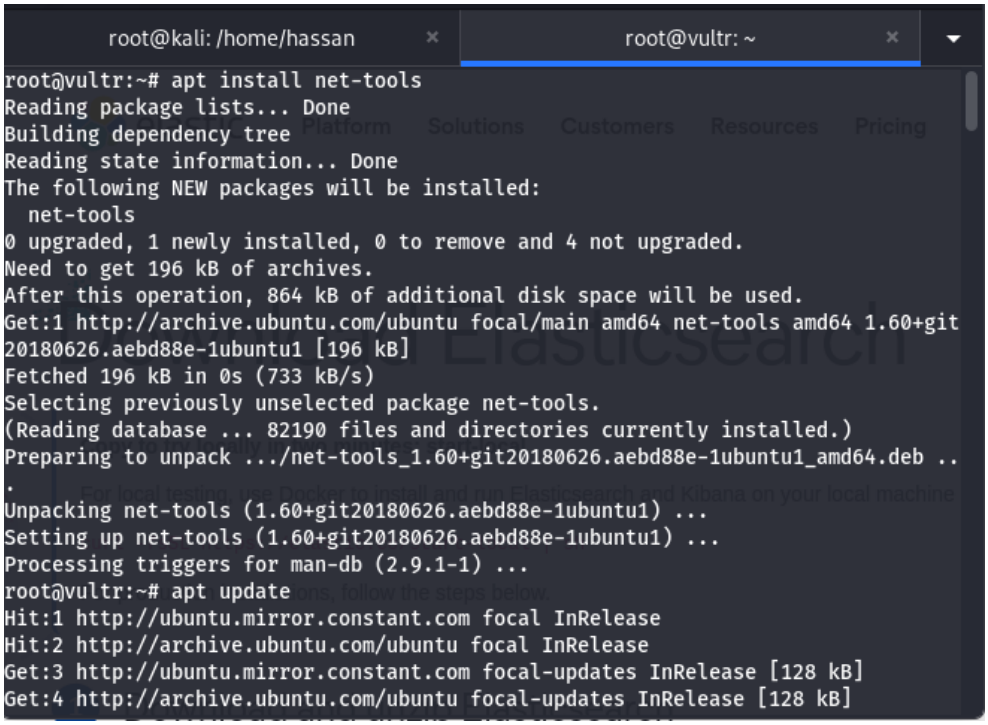


now you can see that the our server is running
and you can change its setting.
you can view its in the console.

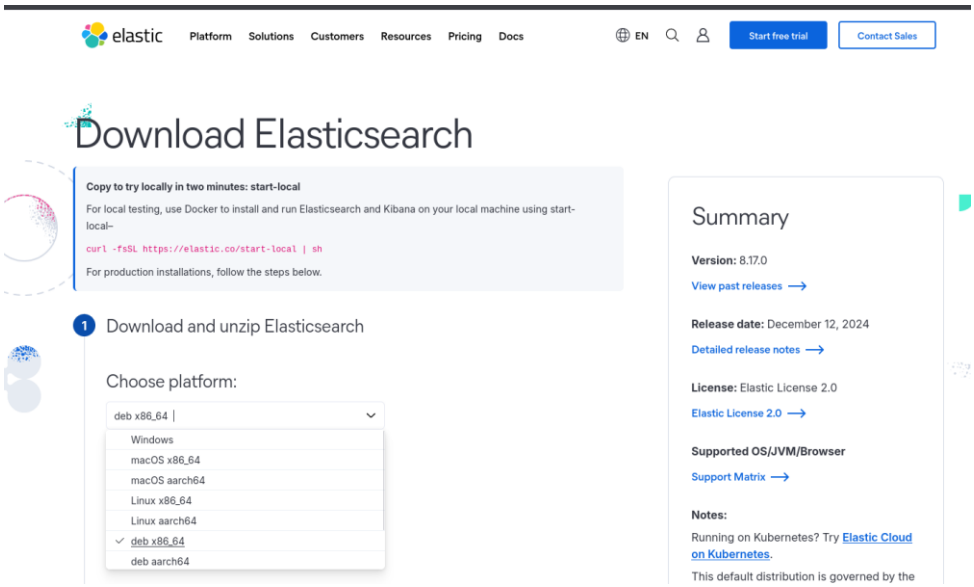


Or you can run in the locally by using ssh:

- **now do ssh root@ipaddress and enter password in the localmachine to use this, so now install some common tools and update and upgrade it:**



- **now install elastic search in it:**



- **installation in the ubuntu server of elastic server:**

```
┌──(root㉿kali)-[/home/hassan]
└─# ssh root@155.138.142.160
root@155.138.142.160's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-200-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat 28 Dec 2024 08:33:43 AM UTC

  System load:  0.0               Processes:               141
  Usage of /:   28.4% of 22.88GB  Users logged in:         1
  Memory usage: 16%               IPv4 address for enp1s0: 155.138.142.160
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Sat Dec 28 07:55:28 2024 from 103.137.24.149
root@vultr:~# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.17.0-amd64.deb
--2024-12-28 08:33:55--  https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.17.0-amd64.deb
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130, 2600:1901:0:1d7::
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 636179540 (607M) [application/vnd.debian.binary-package]
Saving to: ‘elasticsearch-8.17.0-amd64.deb’

elasticsearch-8.17.0-amd64.deb                                4%[=====>
elasticsearch-8.17.0-amd64.deb                               8%[==========>
```

✧

```
root@vultr:~# ls
elasticsearch-8.17.0-amd64.deb  snap
root@vultr:~#
```

✧

- **now run command to install and configure this:**

```
root@vultr:~# dpkg -i elasticsearch-8.17.0-amd64.deb
Selecting previously unselected package elasticsearch.
(Reading database ... 118841 files and directories currently installed.)
Preparing to unpack elasticsearch-8.17.0-amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Unpacking elasticsearch (8.17.0) ...
Setting up elasticsearch (8.17.0) ...
--------------------------- Security autoconfiguration information ------------------------------

Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : TZGlJhVMLSbn*rvw3Gop

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
after creating an enrollment token on your existing cluster.

You can complete the following actions at any time:

Reset the password of the elastic built-in superuser with
'/usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic'.

Generate an enrollment token for Kibana instances with
 '/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana'.

Generate an enrollment token for Elasticsearch nodes with
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s node'.

-------------------------------------------------------------------------------------------------
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
 sudo systemctl daemon-reload
 sudo systemctl enable elasticsearch.service
### You can start elasticsearch service by executing
 sudo systemctl start elasticsearch.service
root@vultr:~#
```

- **this contains an important file where the username and the password is there so stored this file in .txt as well:**

```
root@vultr:~# cat elastic-configure.txt
Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : TZGlJhVMLSbn*rvw3Gop

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
after creating an enrollment token on your existing cluster.

You can complete the following actions at any time:

Reset the password of the elastic built-in superuser with
'/usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic'.

Generate an enrollment token for Kibana instances with
 '/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana'.

Generate an enrollment token for Elasticsearch nodes with
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s node'.

-------------------------------------------------------------------------------------------------
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd
 sudo systemctl daemon-reload
 sudo systemctl enable elasticsearch.service
### You can start elasticsearch service by executing
 sudo systemctl start elasticsearch.service
root@vultr:~#
```

- **now run this commands to update the status:**

```
root@vultr:~# sudo systemctl daemon-reload
root@vultr:~# sudo systemctl enable elasticsearch.service
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.
root@vultr:~#
root@vultr:~#  sudo systemctl start elasticsearch.service
root@vultr:~# sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-12-28 08:37:59 UTC; 1min 6s ago
       Docs: https://www.elastic.co
   Main PID: 24814 (java)
      Tasks: 83 (limit: 2248)
     Memory: 1.4G
     CGroup: /system.slice/elasticsearch.service
             ├─24814 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -Dcli.script=/usr/share/elasticsearch/bi
             ├─24887 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreT
             └─24906 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Dec 28 08:37:29 vultr systemd[1]: Starting Elasticsearch...
Dec 28 08:37:32 vultr systemd-entrypoint[24887]: CompileCommand: dontinline java/lang/invoke/MethodHandle.setAsTypeCache bool dontinline = true
Dec 28 08:37:32 vultr systemd-entrypoint[24887]: CompileCommand: dontinline java/lang/invoke/MethodHandle.asTypeUncached bool dontinline = true
Dec 28 08:37:59 vultr systemd[1]: Started Elasticsearch.
lines 1-16/16 (END)
```

- **so the elastic service is runing, but there is an directory /etc/elastic where all the important files are there, and most important is elasticsearch.yml**
- **where we need to configrue:**

```
root@vultr:/etc/elasticsearch# ls
certs  elasticsearch.keystore  elasticsearch-plugins.example.yml  elasticsearch.yml
root@vultr:/etc/elasticsearch#
```

- **this is the bydefault ipaddress and the portno is :**

```
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
#network.host: 192.168.0.1
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
```

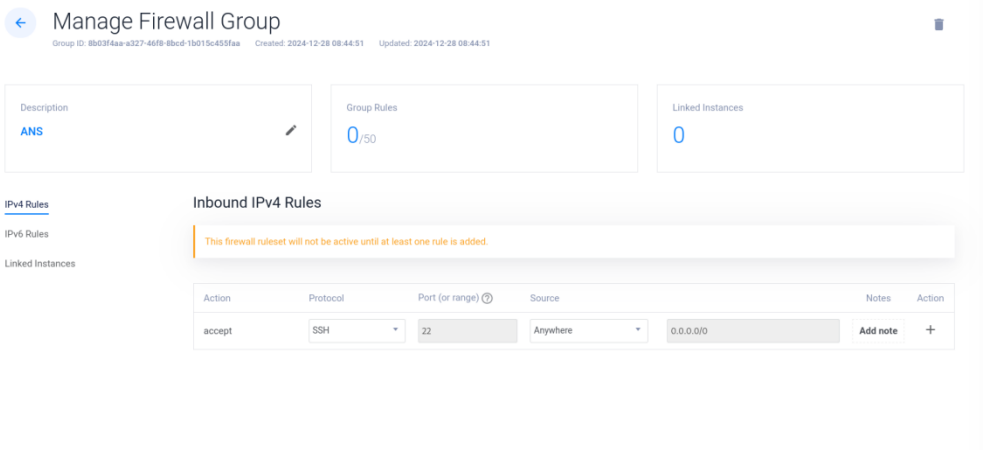
- **use this ipaddress to paste in the above:**

```
root@vultr:/etc/elasticsearch# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 155.138.142.160  netmask 255.255.254.0  broadcast 155.138.143.255
        inet6 fe80::5400:5ff:fe39:883d  prefixlen 64  scopeid 0x20<link>
        ether 56:00:05:39:88:3d  txqueuelen 1000  (Ethernet)
        RX packets 78805  bytes 849419038 (849.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 51490  bytes 3683397 (3.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- **like this:**

```
#
# By default Elasticsearch is only accessible on localhost. Se
# address here to expose this node on the network:
#
network.host: 155.138.142.160
#
# By default Elasticsearch listens for HTTP traffic on the fir
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
```

- **NOW MANAGE FIREWALL GROUPS SO TO OPEN AND ACESS THE ELASTIC SEARCH FROM ANYWHERR:**

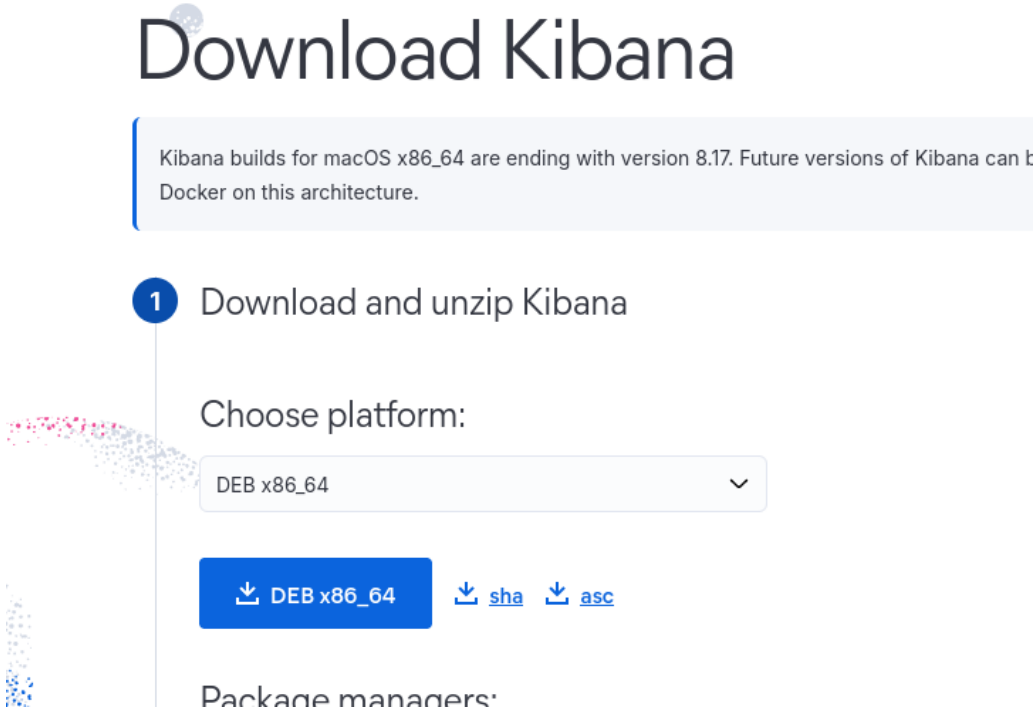


- **after the above changings restart the elasticsearch.service:**

```
root@vultr:/etc/elasticsearch# sudo systemctl restart elasticsearch.service
root@vultr:/etc/elasticsearch# sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-12-28 08:47:00 UTC; 13s ago
       Docs: https://www.elastic.co
   Main PID: 25013 (java)
      Tasks: 76 (limit: 2248)
     Memory: 1.4G
     CGroup: /system.slice/elasticsearch.service
             ├─25013 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=serv
             ├─25088 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkad
             └─25107 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Dec 28 08:46:31 vultr systemd[1]: Stopped Elasticsearch.
Dec 28 08:46:31 vultr systemd[1]: Starting Elasticsearch...
Dec 28 08:46:34 vultr systemd-entrypoint[25088]: CompileCommand: dontinline java/lang/invoke/MethodHandle
Dec 28 08:46:34 vultr systemd-entrypoint[25088]: CompileCommand: dontinline java/lang/invoke/MethodHandle
Dec 28 08:47:00 vultr systemd[1]: Started Elasticsearch.
lines 1-17/17 (END)
```

➢ SETUP THE KIBANA



✧ **installed kibana:**

```
root@vultr:~# wget https://artifacts.elastic.co/downloads/kibana/kibana-8.17.0-amd64.deb
--2024-12-28 08:50:14--  https://artifacts.elastic.co/downloads/kibana/kibana-8.17.0-amd64.deb
Resolving artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130, 2600:1901:0:1d7::
Connecting to artifacts.elastic.co (artifacts.elastic.co)|34.120.127.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 345323288 (329M) [application/vnd.debian.binary-package]
Saving to: 'kibana-8.17.0-amd64.deb'

kibana-8.17.0-amd64.deb                                 100%[=============================================================

2024-12-28 08:50:36 (15.4 MB/s) - 'kibana-8.17.0-amd64.deb' saved [345323288/345323288]

root@vultr:~# ls
elastic-configure.txt  elasticsearch-8.17.0-amd64.deb  kibana-8.17.0-amd64.deb  snap
```

✧

```
root@vultr:~# dpkg -i kibana-8.17.0-amd64.deb
Selecting previously unselected package kibana.
(Reading database ... 120318 files and directories currently install
Preparing to unpack kibana-8.17.0-amd64.deb ...
Unpacking kibana (8.17.0) ...
Setting up kibana (8.17.0) ...
Creating kibana group... OK
Creating kibana user... OK
Kibana is currently running with legacy OpenSSL providers enabled! F
Created Kibana keystore in /etc/kibana/kibana.keystore
root@vultr:~#
```

✧ **now acess this file:**

```
root@vultr:/etc/kibana#
root@vultr:/etc/kibana# ls
kibana.keystore  kibana.yml  node.options
root@vultr:/etc/kibana# nano kibana.yml
```

- **configure like this:**

```
# Kibana is served by a back end server. Thi
server.port: 5601

# Specifies the address to which the Kibana
# The default is 'localhost', which usually
# To allow connections from remote users, se
server.host: 155.138.142.160
# Enables you to specify a path to mount Kib
# Use the `server.rewriteBasePath` setting t
```

- **now save this and start the services for kibana:**

```
root@vultr:~# systemctl daemon-reload
root@vultr:~# systemctl enalbe kibana.service
Unknown operation enalbe.
root@vultr:~# systemctl enable kibana.service
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /lib/systemd/system/kibana.service.
root@vultr:~# systemctl start kibana.service
root@vultr:~# systemctl status kibana.service
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-12-28 08:56:20 UTC; 7s ago
       Docs: https://www.elastic.co
   Main PID: 25380 (node)
      Tasks: 11 (limit: 2248)
     Memory: 321.5M
     CGroup: /system.slice/kibana.service
             └─25380 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/kibana/bin/../src/cli/dist

Dec 28 08:56:20 vultr systemd[1]: Started Kibana.
Dec 28 08:56:21 vultr kibana[25380]: Kibana is currently running with legacy OpenSSL providers enabled! For details and ir
Dec 28 08:56:21 vultr kibana[25380]: {"log.level":"info","@timestamp":"2024-12-28T08:56:21.657Z","log.logger":"elastic-apm
Dec 28 08:56:21 vultr kibana[25380]: Native global console methods have been overridden in production environment.
Dec 28 08:56:22 vultr kibana[25380]: [2024-12-28T08:56:22.944+00:00][INFO ][root] Kibana is starting
Dec 28 08:56:22 vultr kibana[25380]: [2024-12-28T08:56:22.998+00:00][INFO ][node] Kibana process configured with roles: [b
lines 1-16/16 (END)
```

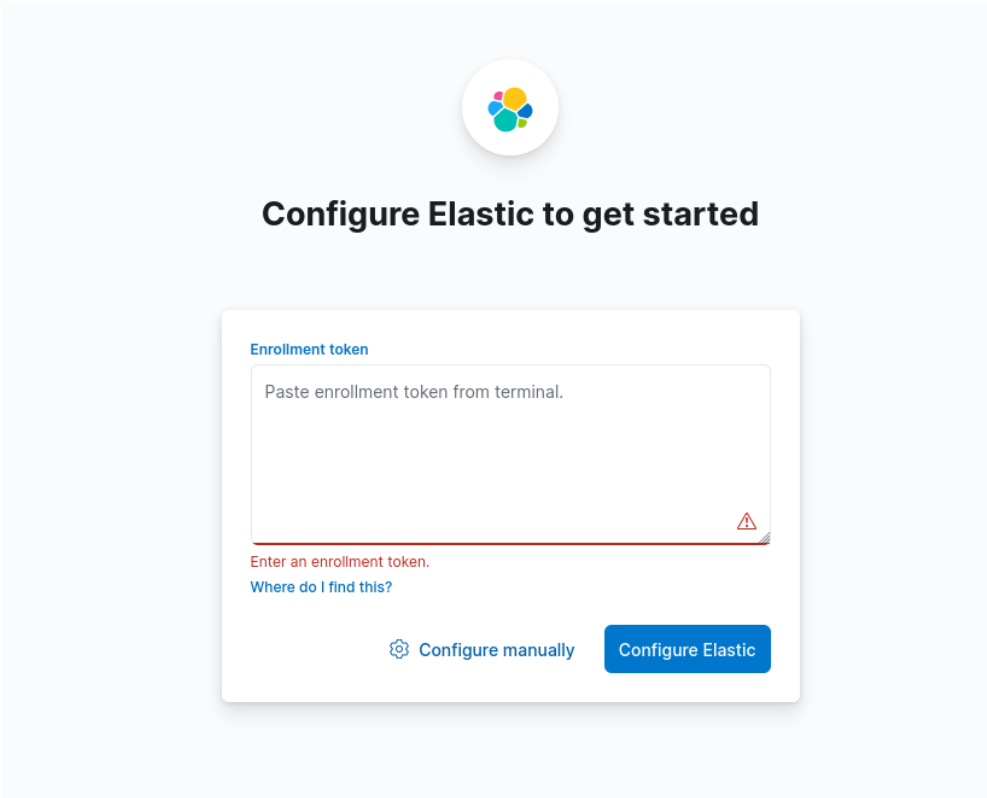
- **before running its gui, generate an token:**

```
root@kali: /home/hassan
root@vultr:/usr/share/elasticsearch/bin# ls
elasticsearch            elasticsearch-cli                 elasticsearch-env            elasticsearch-keystore  elasticsearch-re
elasticsearch-certgen    elasticsearch-create-enrollment-token  elasticsearch-env-from-file  elasticsearch-node      elasticsearch-re
elasticsearch-certutil   elasticsearch-croneval            elasticsearch-geoip          elasticsearch-plugin    elasticsearch-sa
root@vultr:/usr/share/elasticsearch/bin# ./elasticsearch-create-enrollment-token --scope kibana
eyJ2ZXIiOiI4LjE0LjAiLCJhZHIiOlsiMTU1LjEzOC4xNDIuMTYwOjkyMDAiXSwiZmdyIjoiNTQ0OWNkZWJmZTljZDBkMjRhNmUwMGZhZGI4YmZlYzlkYTcwNTE2ZGNjZTY2Y
root@vultr:/usr/share/elasticsearch/bin#
```

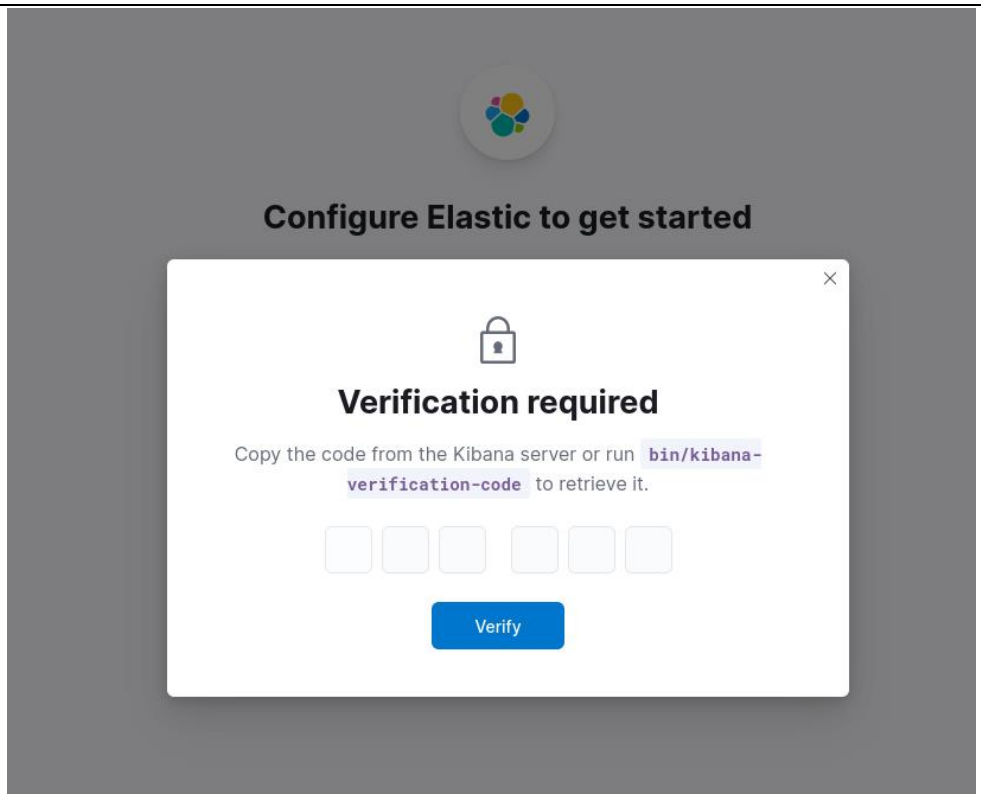
- **now save this enrollment token in an file:**

```
root@vultr:~# nano elastic-enrollment-token
root@vultr:~# cat elastic-enrollment-token
eyJ2ZXIiOiI4LjE0LjAiLCJhZHIiOlsiMTU1LjEzOC4xNDIuMTYwOjkyMDAiXSwiZmdyIjoiNTQ0OWN
root@vultr:~#
```

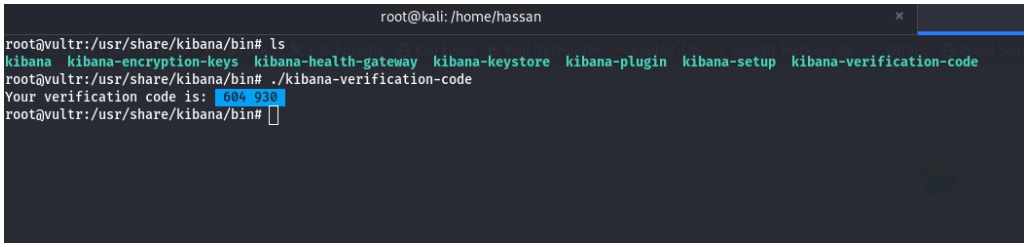
- **now acess the elastic search through web gui:**



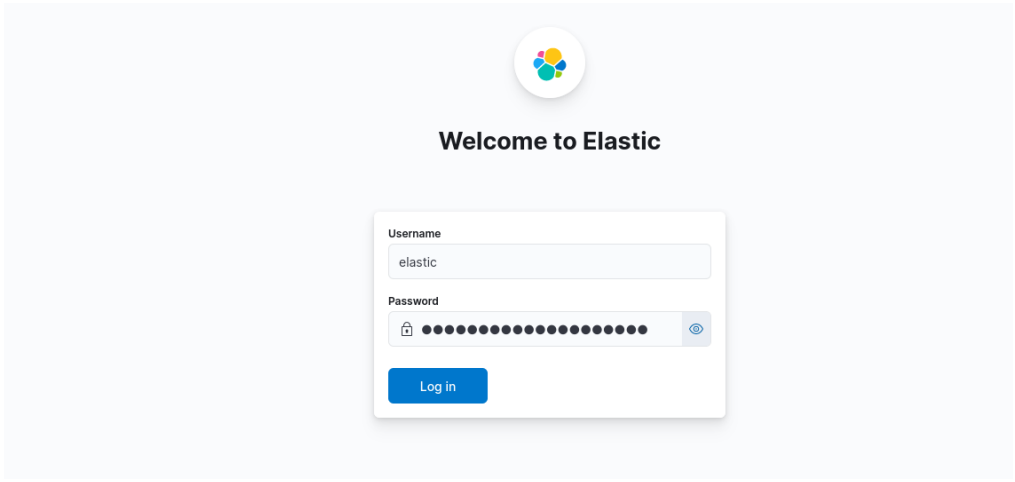
- **now add that token**



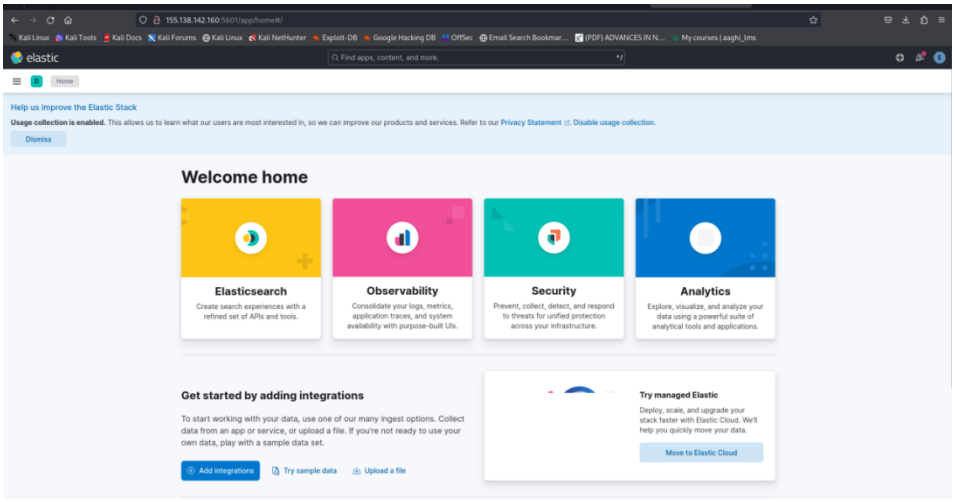
- **now verification the code by runnin this command and paste here:**



- **now enter the otp and now the dashboard look like this:**



✧ **now enter:**



✧

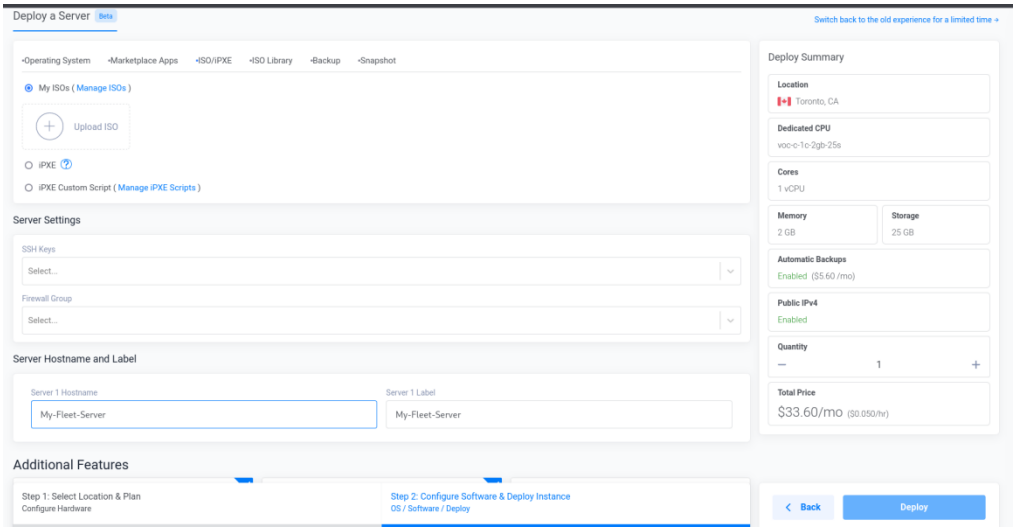
✧ **so the dashboard looks like this**

✧

✧ http://155.138.142.160:5601/app

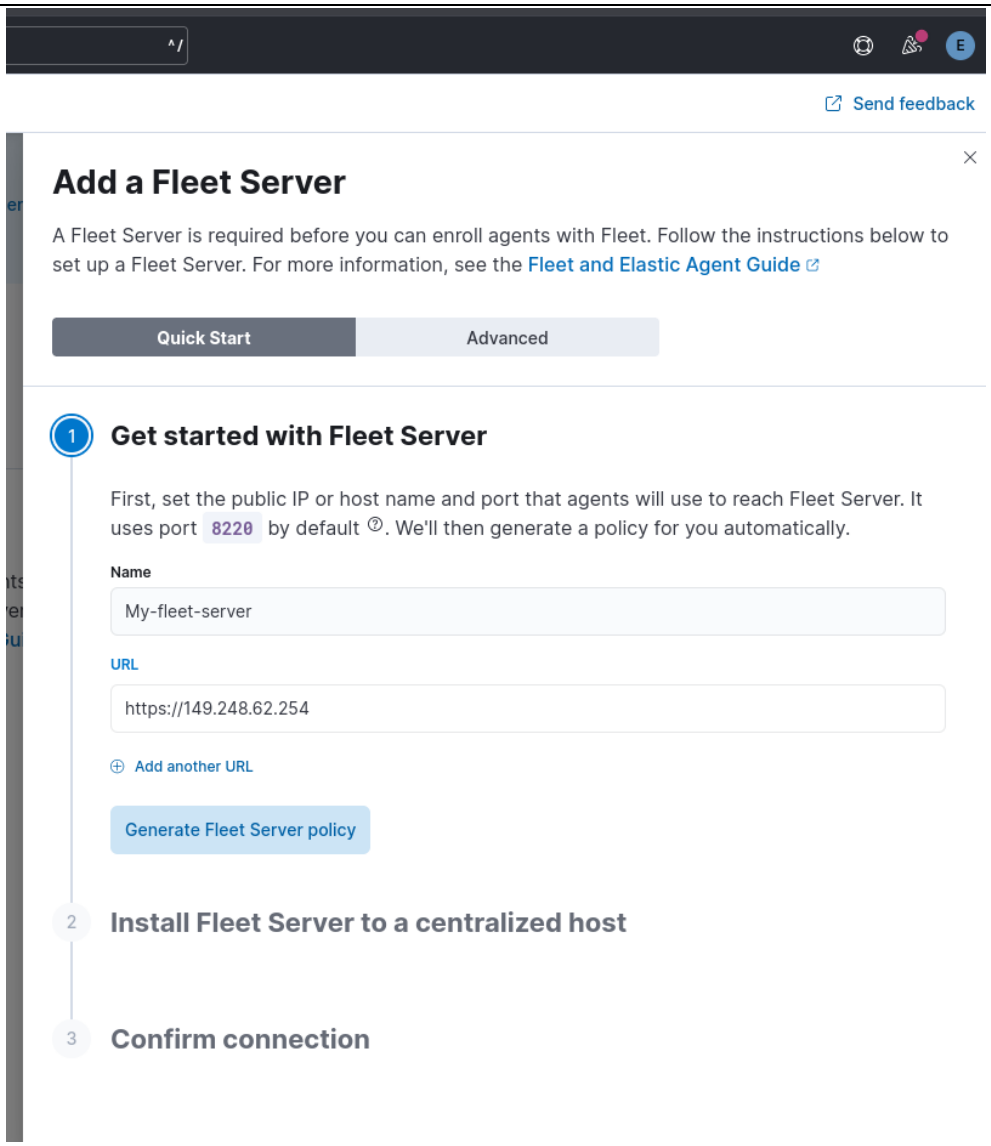
➢ SETUP FLEET SERVER AND ELSTIC AGENT

✧ **now setup and fleet server and elastic agent:**



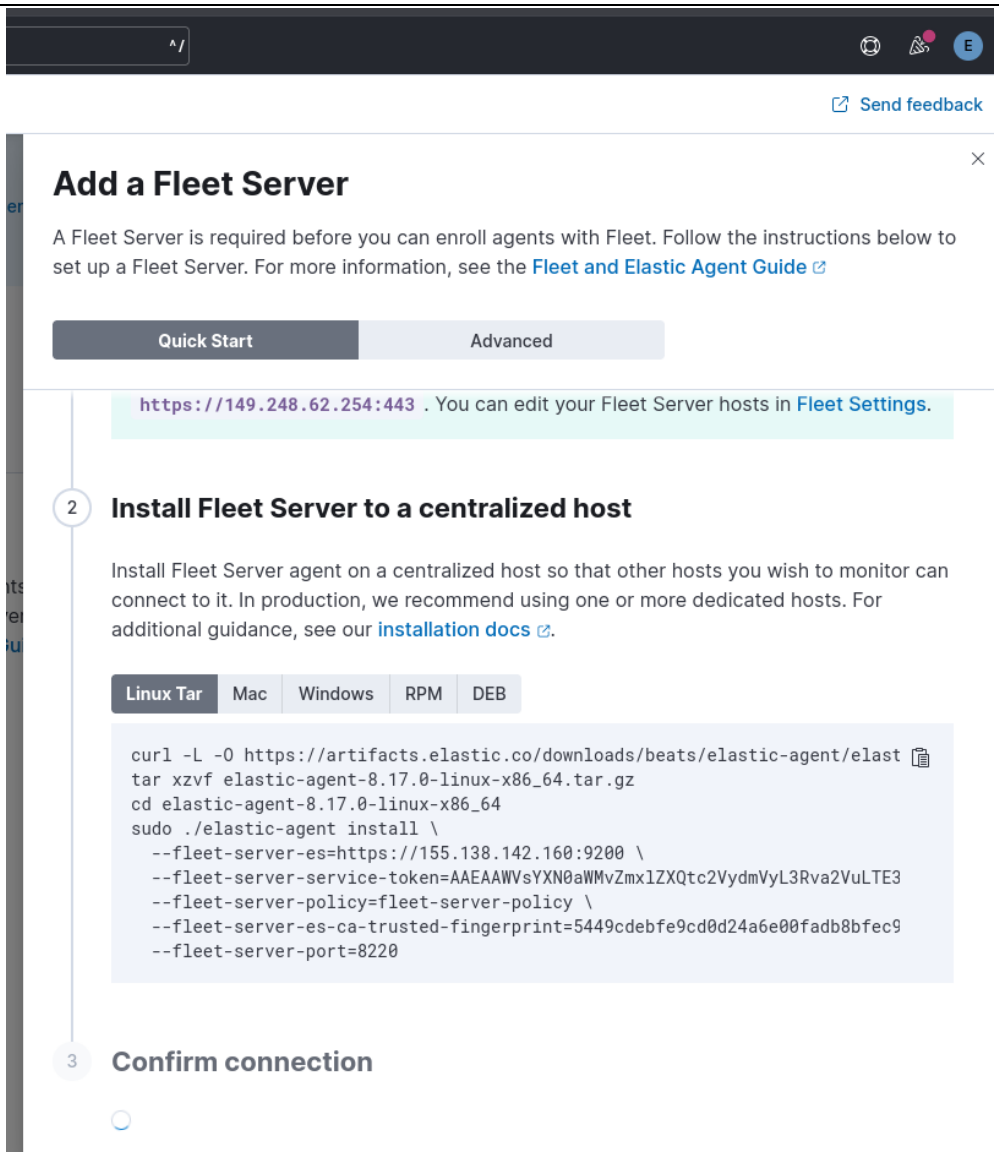
✧

✧ **now add this**

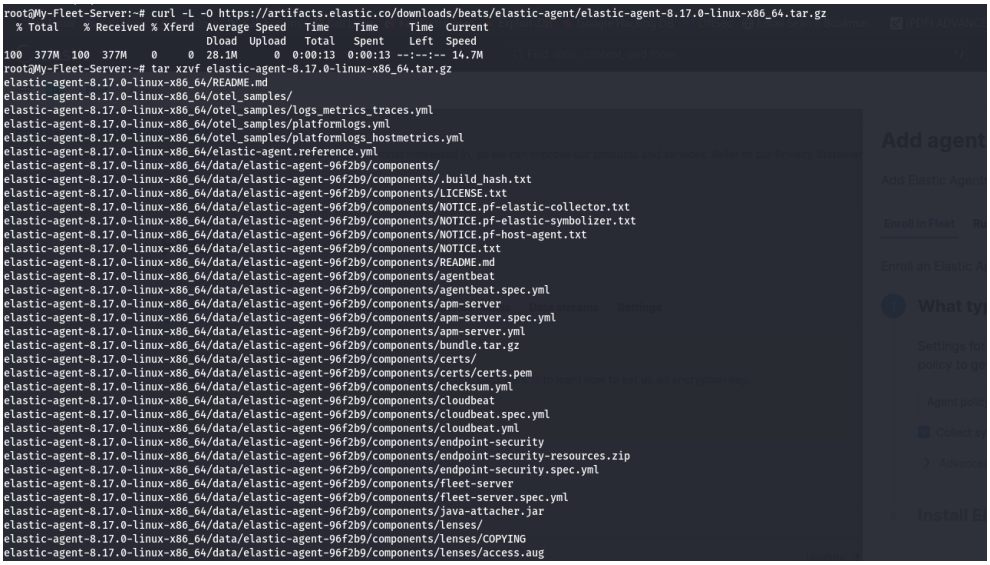


✧

✧ **paste this in the fleet server:**



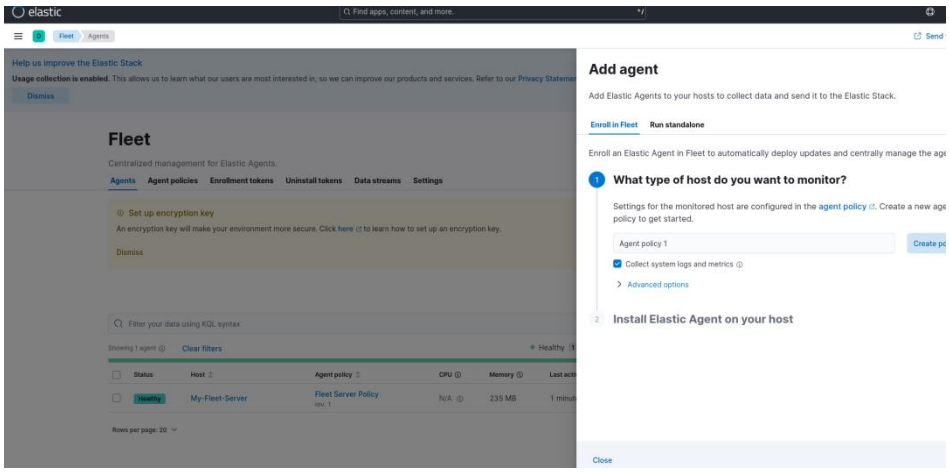
✧ **enter this command:**



- **and run this command first:**

```
root@My-Fleet-Server:~# ufw allow 9200
Rule added
Rule added (v6)
root@My-Fleet-Server:~# ufw allow 8220
Rule added
Rule added (v6)
```

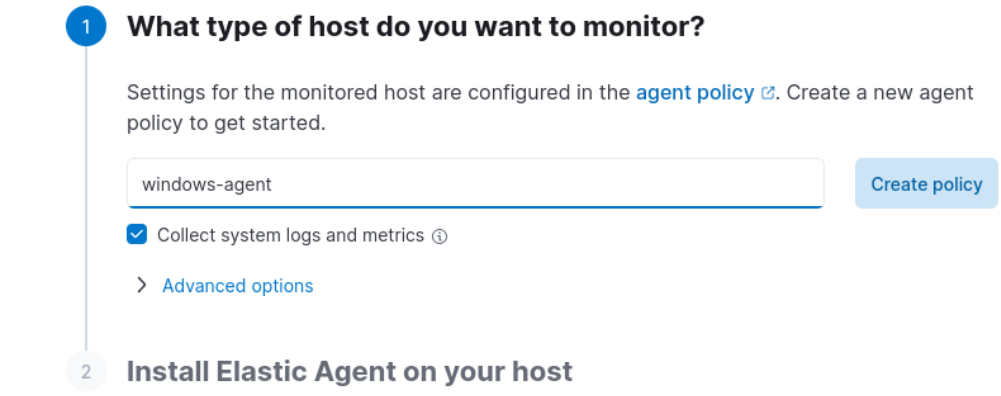
- **now you can see we have an fleet server sucessfully connected:**



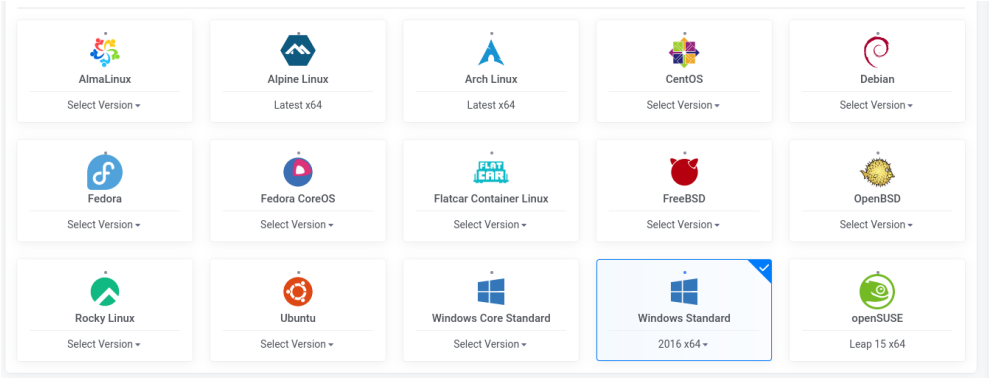
- **see in the terminal as well:**

```
>   --fleet-server-port=8220
Elastic Agent will be installed at /opt/Elastic/Agent and will run as a service. Do you want to continue? [Y/n]:y
[   =] Service Started  [2s] Elastic Agent successfully installed, starting enrollment.
[==  ] Waiting For Enroll...  [4s] {"log.level":"info","@timestamp":"2024-12-28T14:12:08.934Z","log.origin":{"func
,"file.line":437},"message":"Generating self-signed certificate for Fleet Server","ecs.version":"1.6.0"}
[ ===] Waiting For Enroll...  [5s] {"log.level":"info","@timestamp":"2024-12-28T14:12:09.465Z","log.origin":{"func
_cmd.go","file.line":483},"message":"Restarting agent daemon, attempt 0","ecs.version":"1.6.0"}
[=   ] Waiting For Enroll...  [7s] {"log.level":"info","@timestamp":"2024-12-28T14:12:11.470Z","log.origin":{"func
le.line":826},"message":"Fleet Server - Starting","ecs.version":"1.6.0"}
[=   ] Waiting For Enroll...  [11s] {"log.level":"info","@timestamp":"2024-12-28T14:12:15.474Z","log.origin":{"fun
ile.line":807},"message":"Fleet Server - Running on policy with Fleet Server integration: fleet-server-policy; mis
[=   ] Waiting For Enroll...  [11s] {"log.level":"info","@timestamp":"2024-12-28T14:12:16.213Z","log.origin":{"fun
go","file.line":520},"message":"Starting enrollment to URL: https://My-Fleet-Server:8220/","ecs.version":"1.6.0"}
[  ==] Waiting For Enroll...  [13s] {"log.level":"info","@timestamp":"2024-12-28T14:12:17.321Z","log.origin":{"fun
l_cmd.go","file.line":483},"message":"Restarting agent daemon, attempt 0","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2024-12-28T14:12:17.323Z","log.origin":{"function":"github.com/elastic/elastic-a
rriggered restart on running Elastic Agent.","ecs.version":"1.6.0"}
Successfully enrolled the Elastic Agent.
[  ==] Done  [13s]
Elastic Agent has been successfully installed.
root@My-Fleet-Server:~/elastic-agent-8.17.0-linux-x86_64#
```

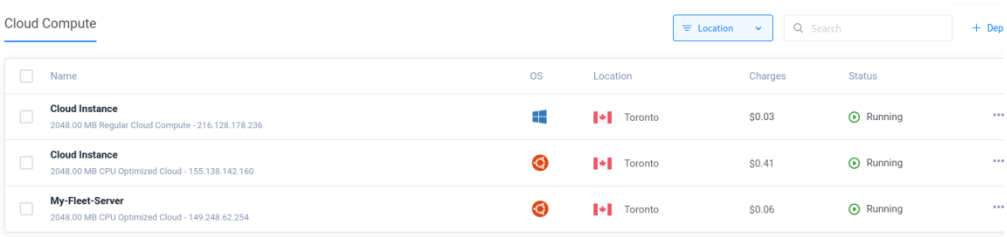
- **now add an windows server as elastic agent, which we will need to montor.**



- **and use the windows server int the vultr:**

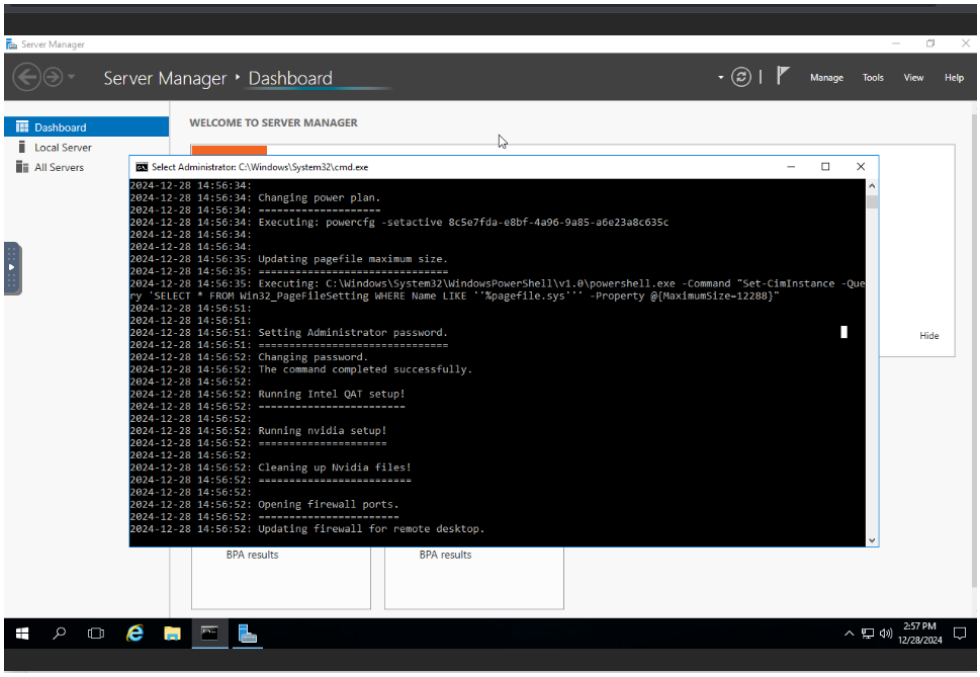


✧ 3 servers:



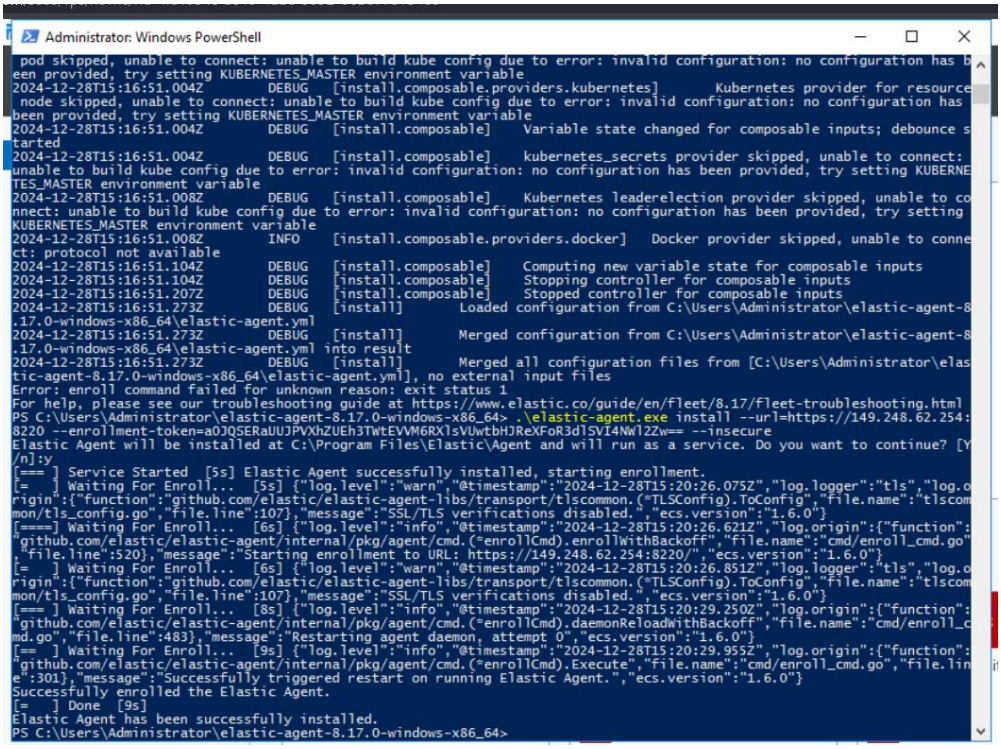
✧

✧ **this is windows:**

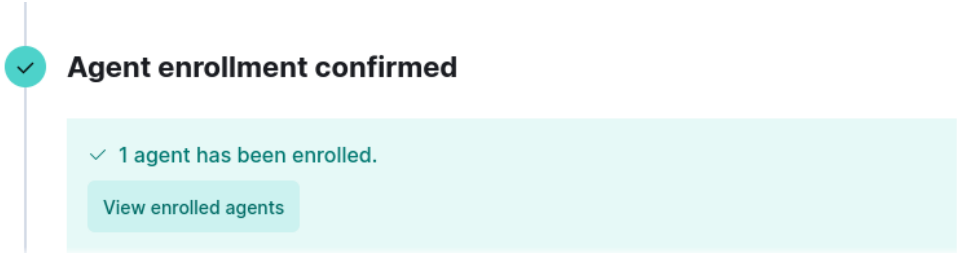


✧

✧ **so the agent is succesfully installed:**

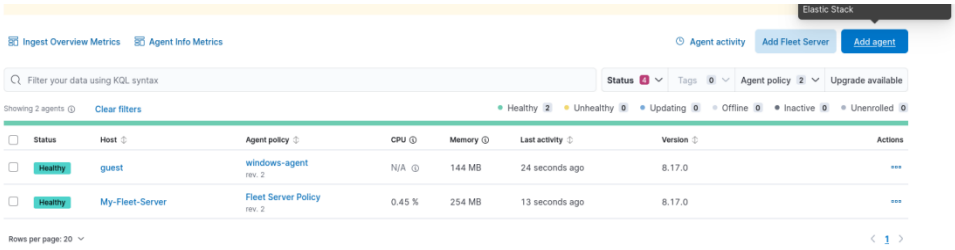


✧



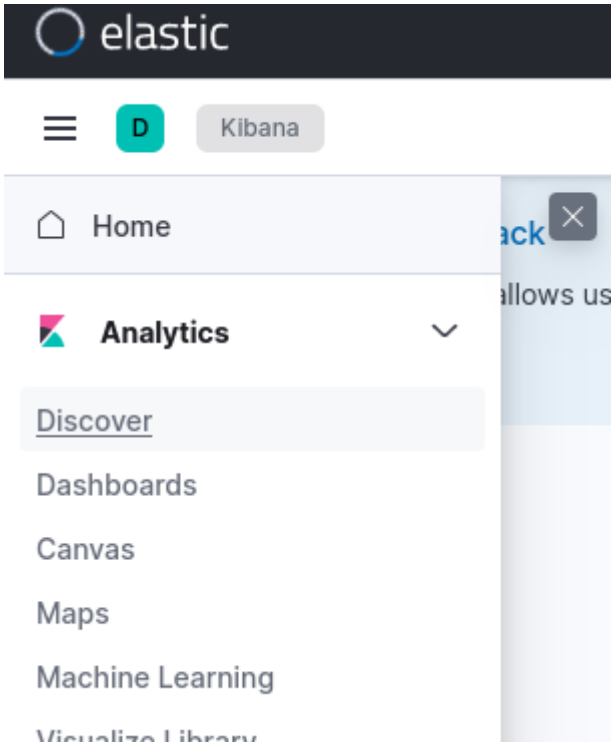
✧

✧ **so the windows agent and the fleet server is successfully installed:**



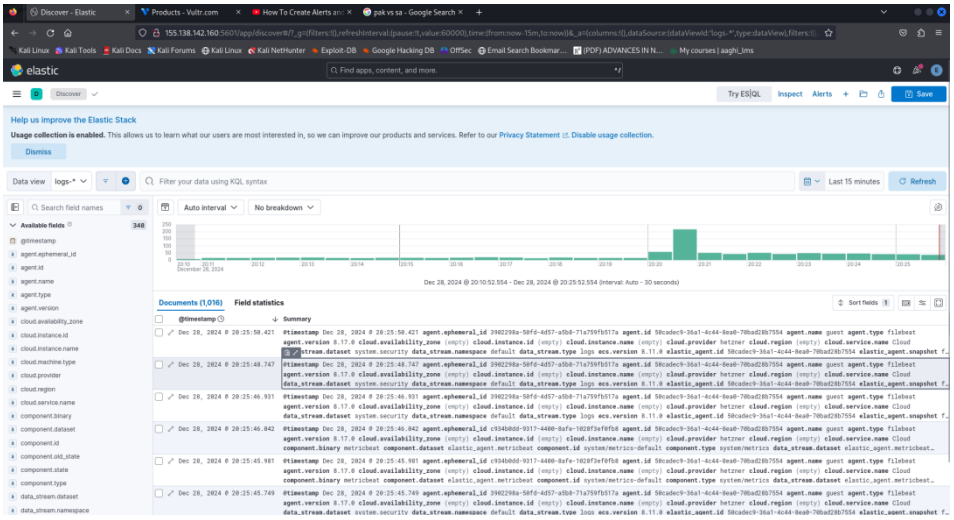
✧

✧ **now click on discover:**



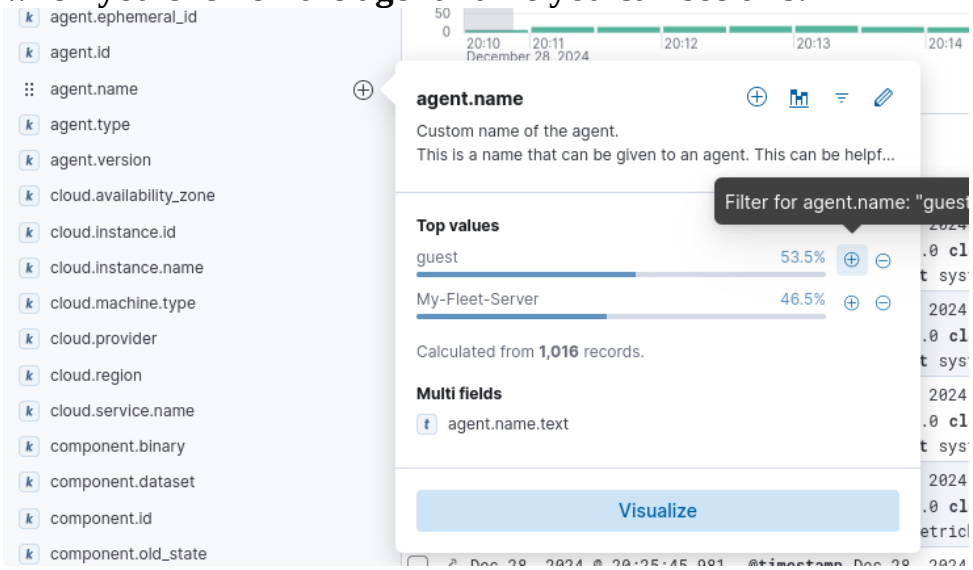
✧

✧ **these are the logs we can see:**



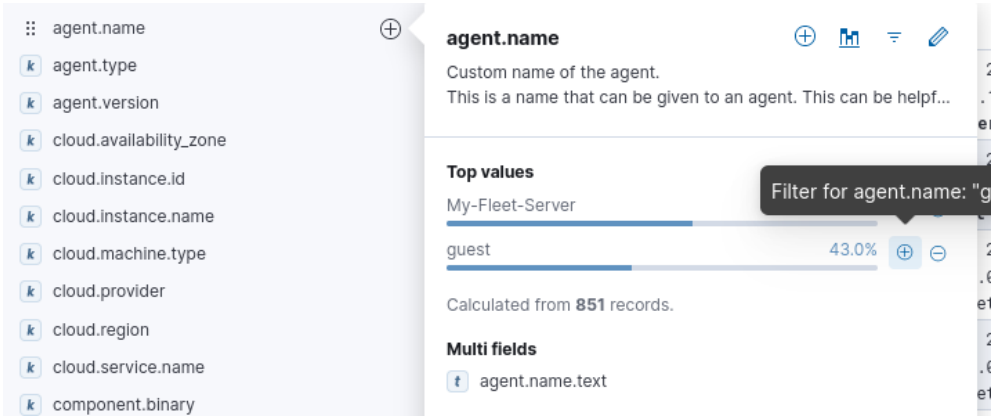
✧

- **when you click on the agent name you can see this:**



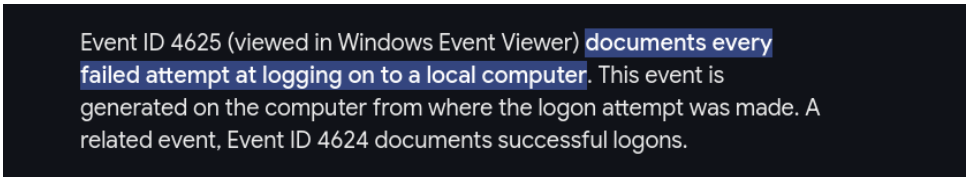
- **so you can monitor this as well**

- **now perform create an alerts in the elasticsearch ELK and perform attack on the windows server..**

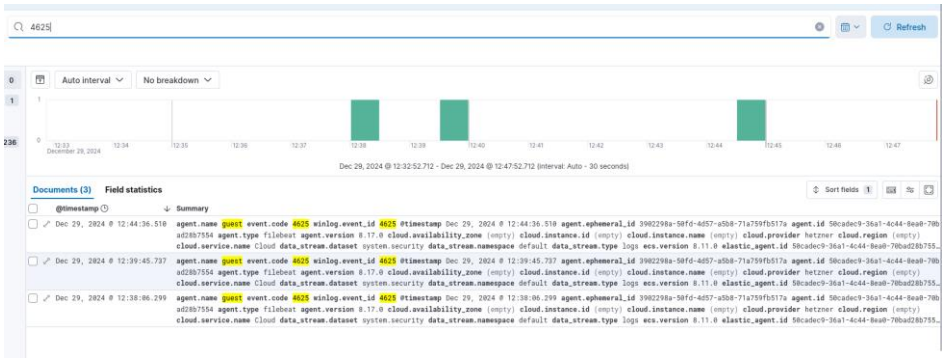


- **so monitor only the “guest” which is our windows server:**

- **and this is the event id which tells us that whoever try to login into the system.**



- **when search for this id “4625” in the filter tab:**



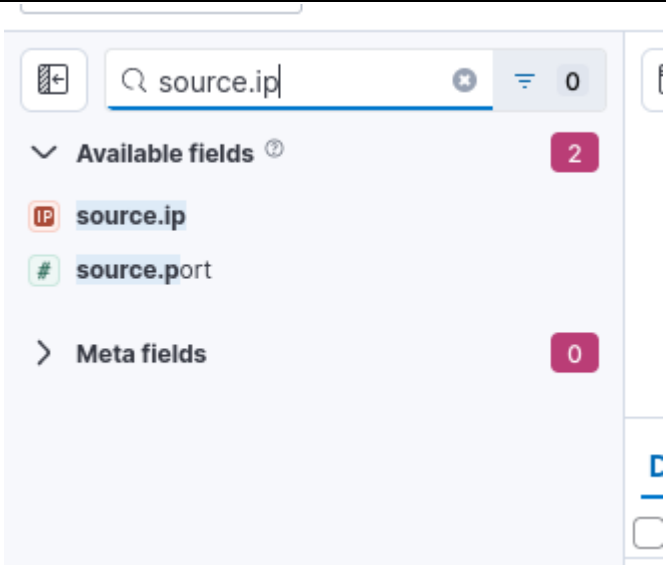
✧

➢ **RDP AUTHENTICATION DETECTION**

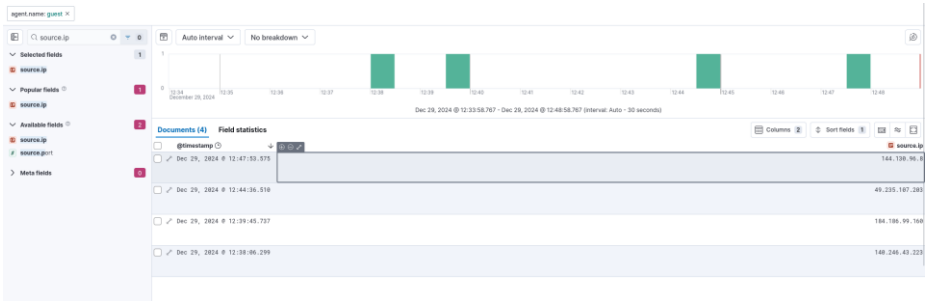
✧ **you can see some logs as well. and you can search like this as well:**

✧

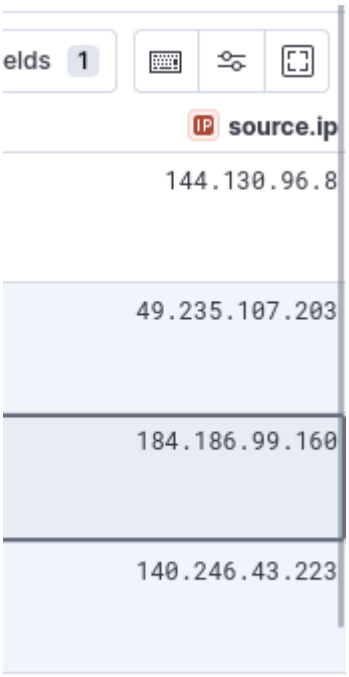
✧ **apply this filter to know what is the source ip:**



✧ **you can see the source ip:**

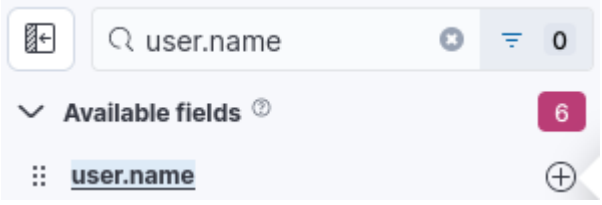


✧



✧

✧ **now we have username as well so filter that also:**

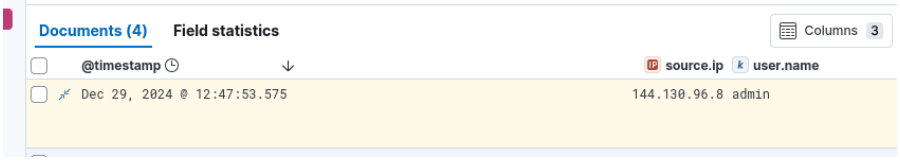

user.name
0
Available fields
6
user.name

✧ **so this is the some steps:**

source.ip	user.name
144.130.96.8	admin
49.235.107.203	Admin
184.186.99.160	admin
140.246.43.223	Admin

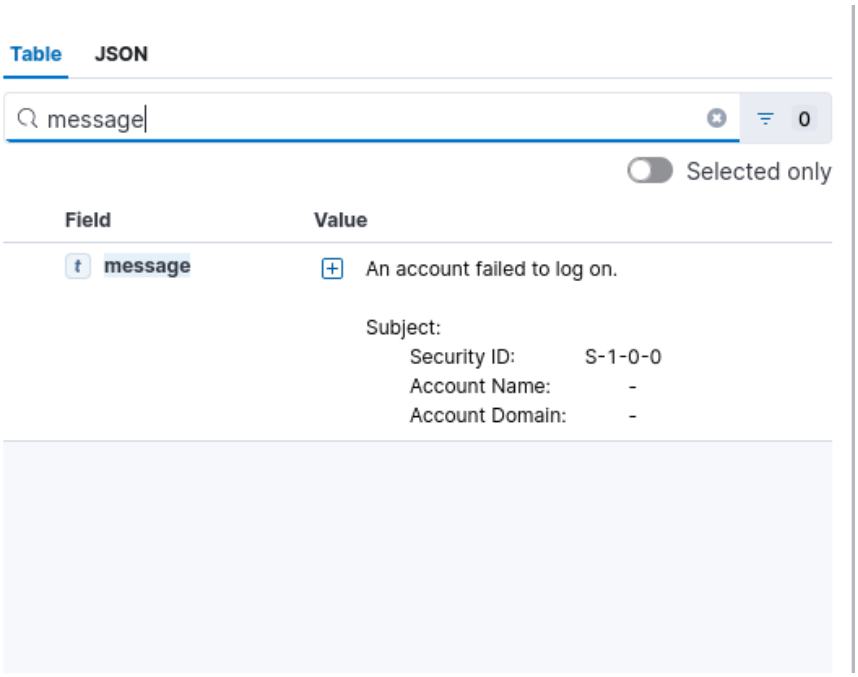
✧

✧ **when open any event:**



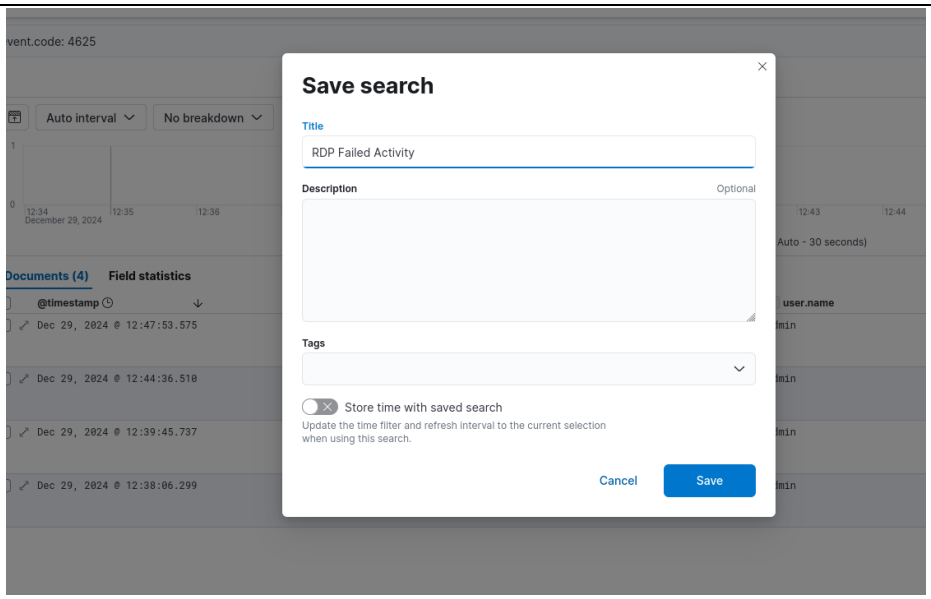
✧

✧ **see the message:**

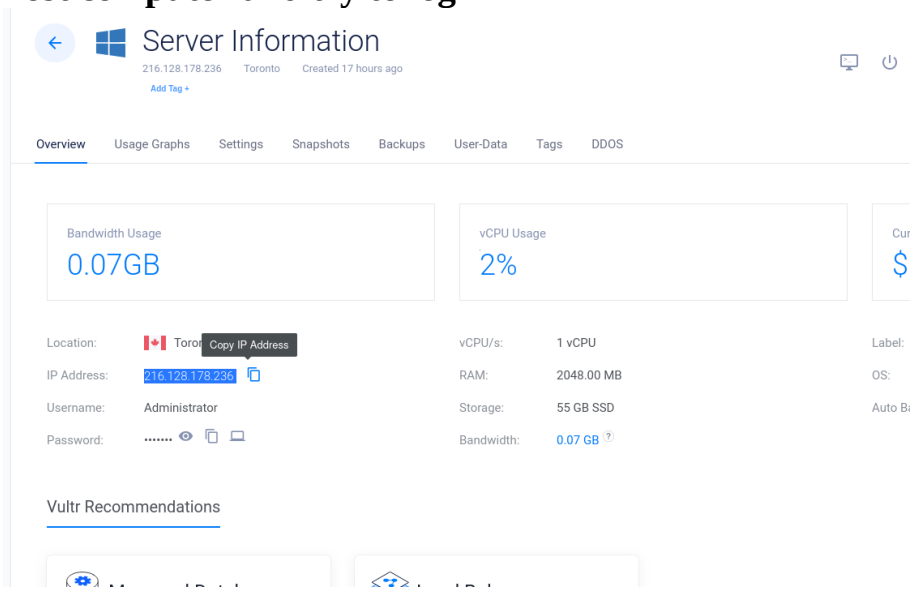


✧

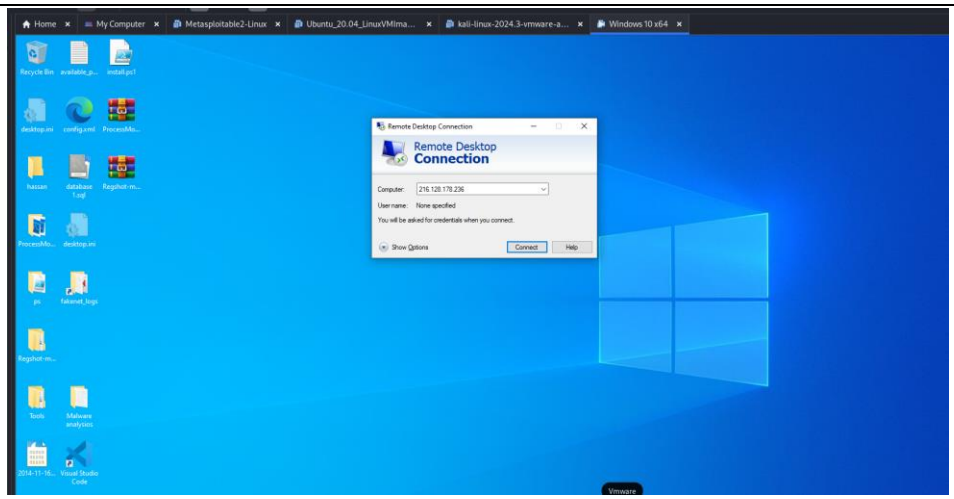
✧ **now save this activity as “RDP failed actity”:**



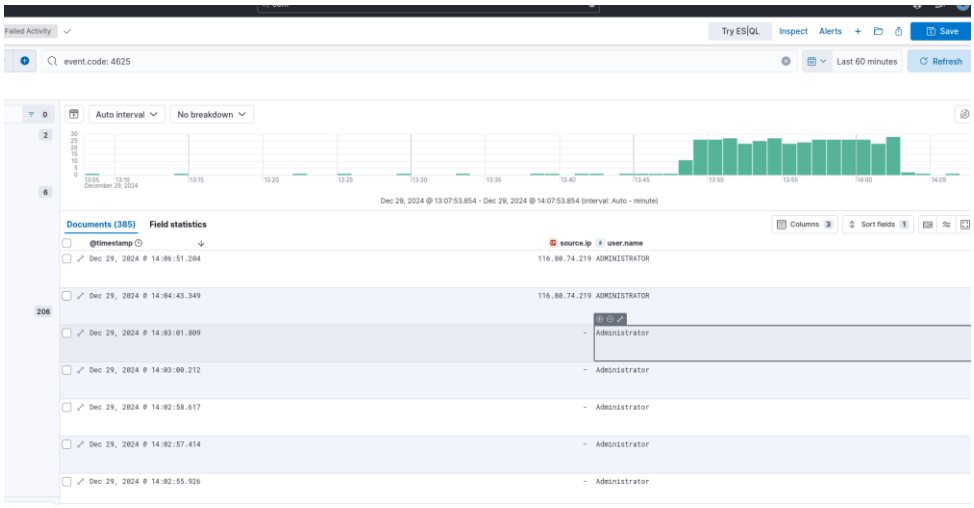
✧ **now perform an RDP login using the attacker laptop which is my host computer and try to login**

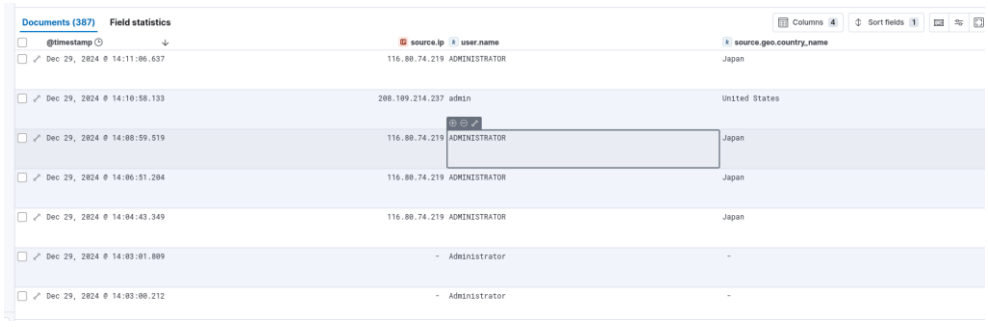


✧ **copy this ipaddress and try to login using the attacker laptop:**



- **logs are created like this:**

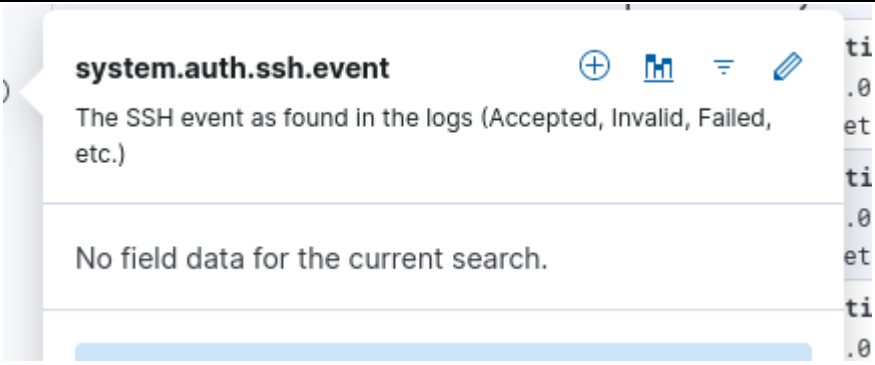




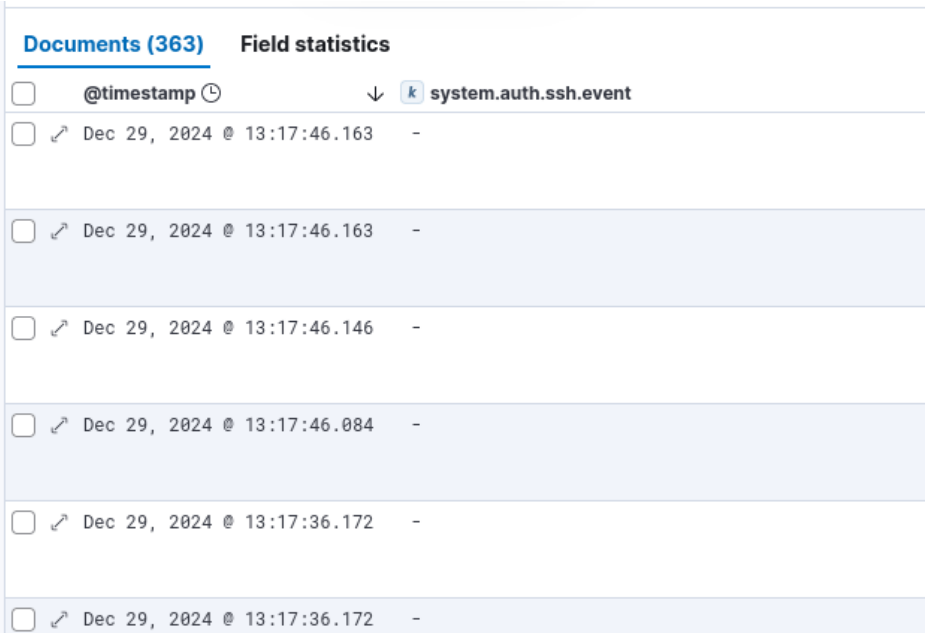
- **so these are the logs when trying RDP to acess the windows servrer**

✧ ssh brute force attack:

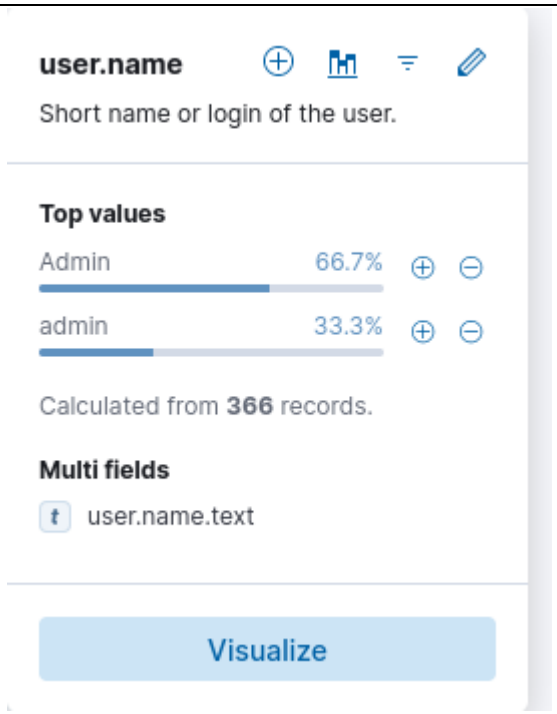
✧ **choose this:**



- **add this column**

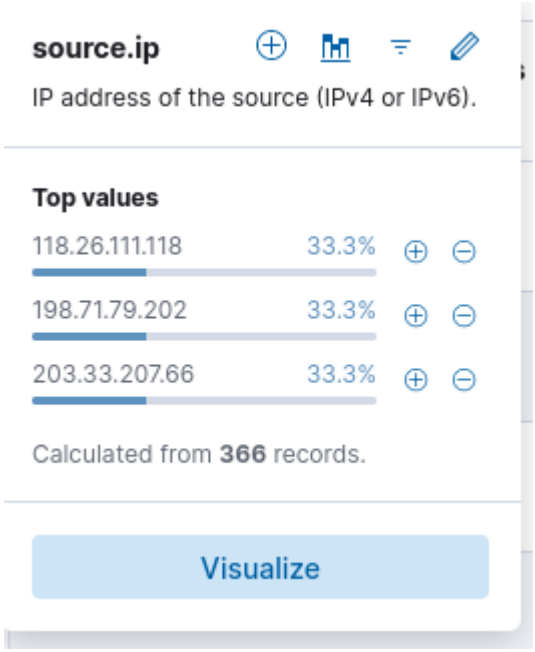


- **track the usernames as well:**

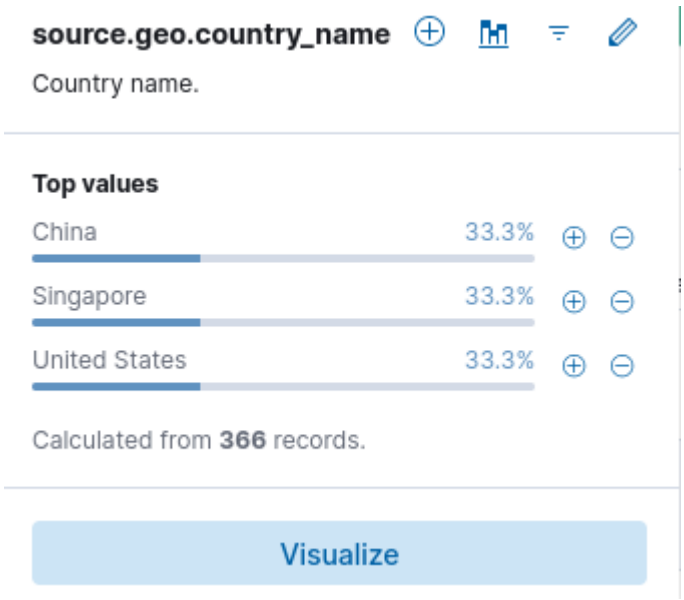


✧ **so add this into the column:**

✧ **also we have to see the source ip:**

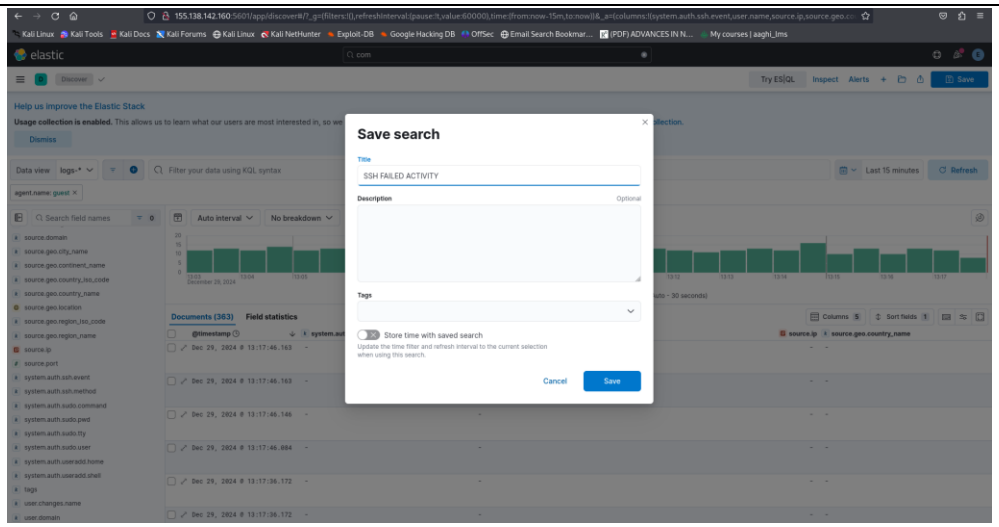


✧ **also add country name:**



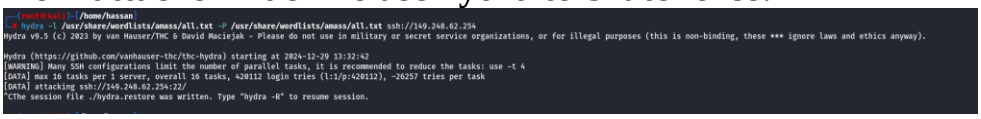
✧ **add this into the column and the final look like this:**

✧ **Save it as “SSH FAILED ACTIVITY”:**


Save search
Title
SSH FAILED ACTIVITY
Description
Optional
Tags
Store time with saved search
Update the time filter and refresh interval to the current selection when using this search.
Cancel
Save

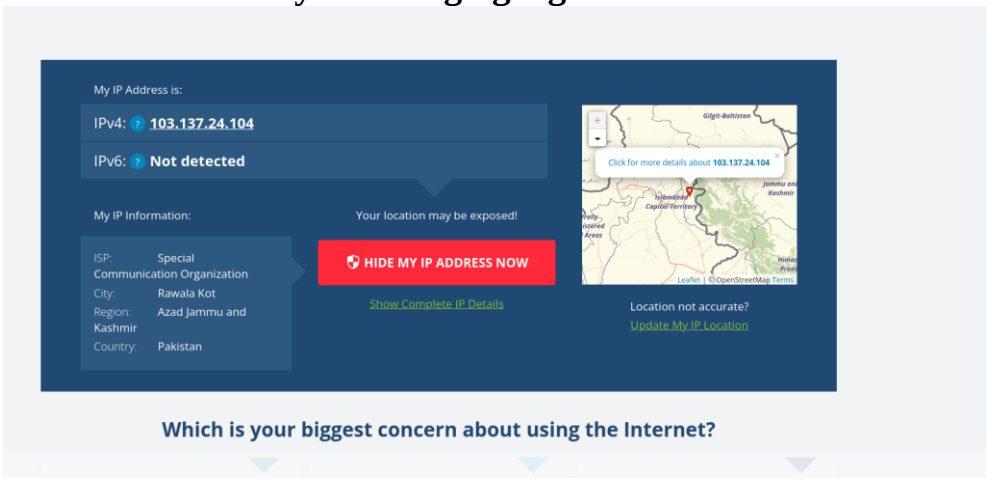
✧ **now see that:**

- **from attacker machine use hydra to brute force:**



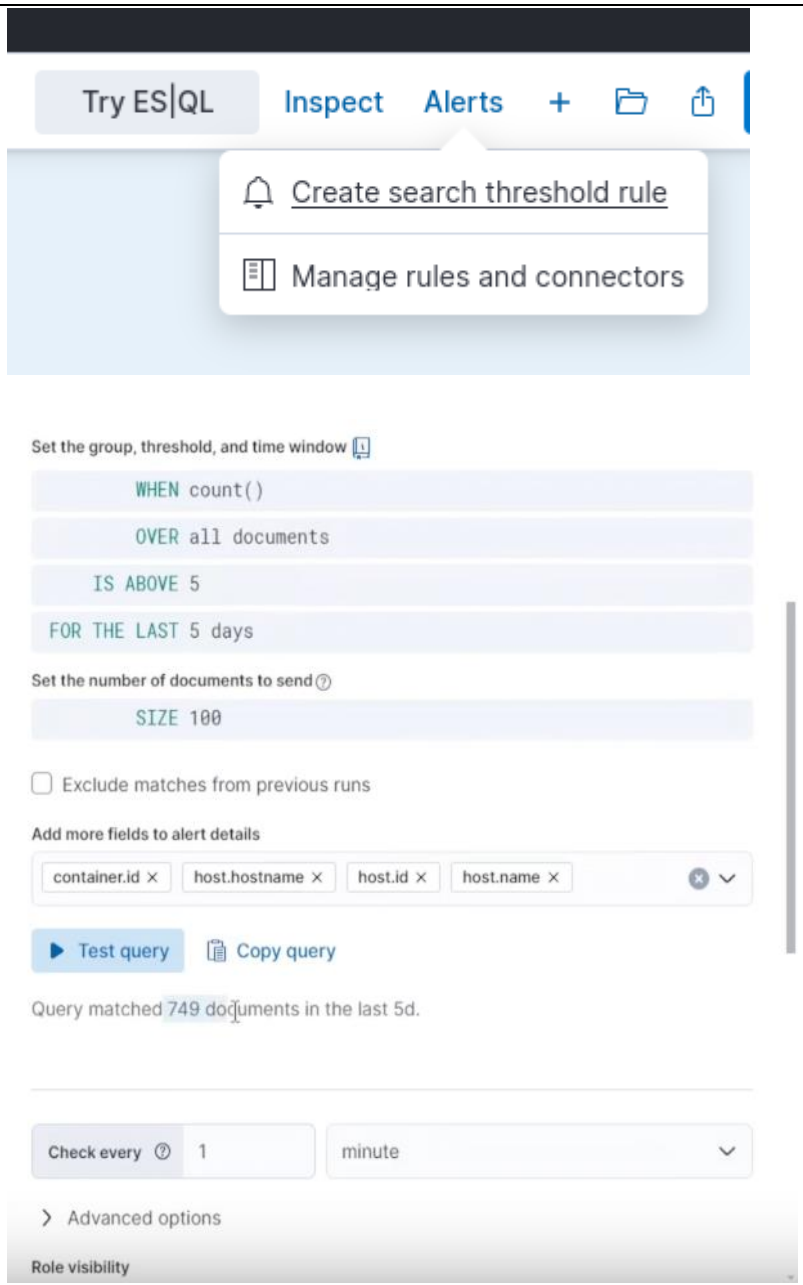
- **this is the logs:**

- **so you can see that the logs ipaddress is the attacker machine:**
- **and we can identify it through google its location as well:**

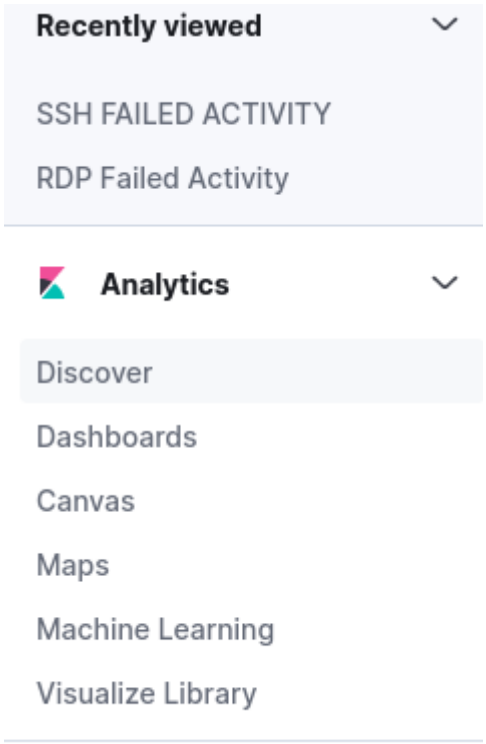


- **as i am using cloud so you can see someone trying from other countries as well:**

- **create an alert:**

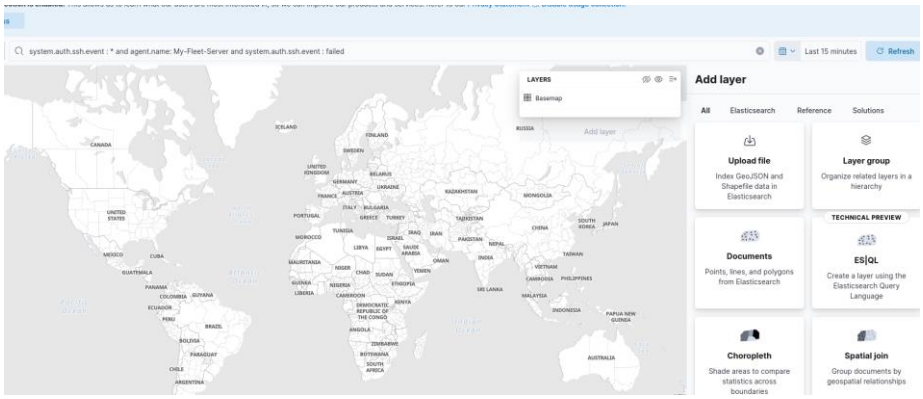

Try ES|QL
Inspect
Alerts
Create search threshold rule
Manage rules and connectors
Set the group, threshold, and time window
WHEN count()
OVER all documents
IS ABOVE 5
FOR THE LAST 5 days
Set the number of documents to send
SIZE 100
Exclude matches from previous runs
Add more fields to alert details
container.id
host.hostname
host.id
host.name
Test query
Copy query
Query matched 749 documents in the last 5d.
Check every
1
minute
Advanced options
Role visibility

✧ **now go to maps:**



✧

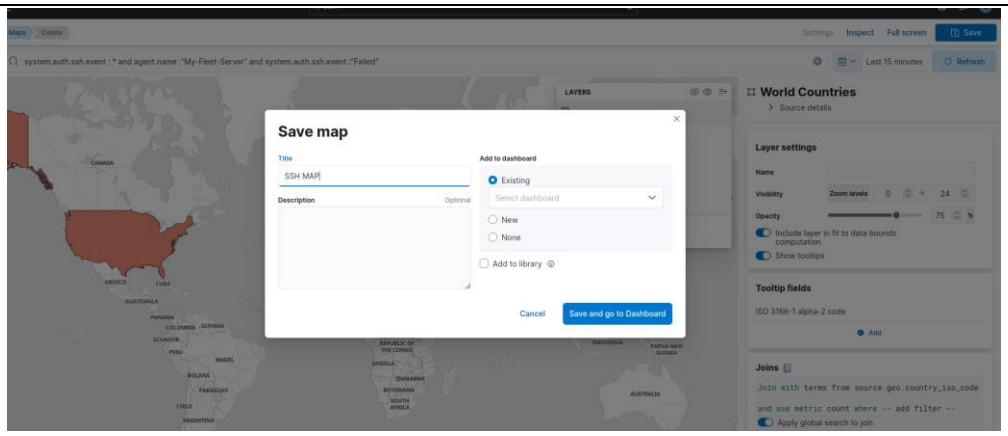
✧ **now add an layer:**

✧ 

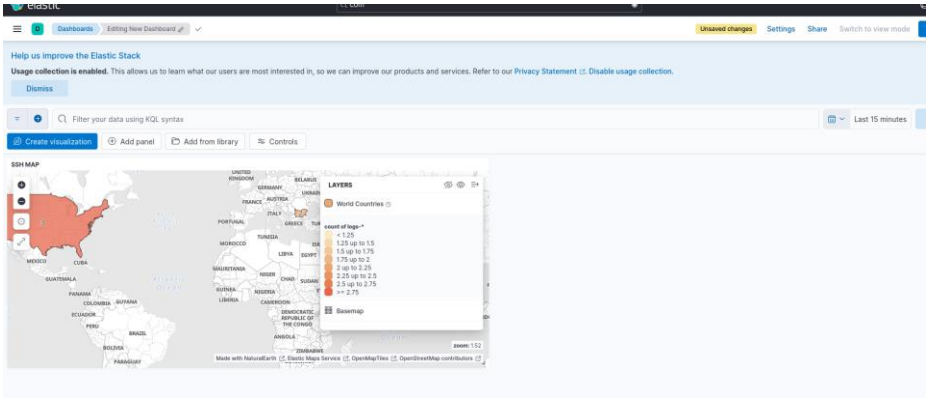
✧ **so the location like this:**

✧ 

✧ **NOW SAVE AS SSH MAP:**

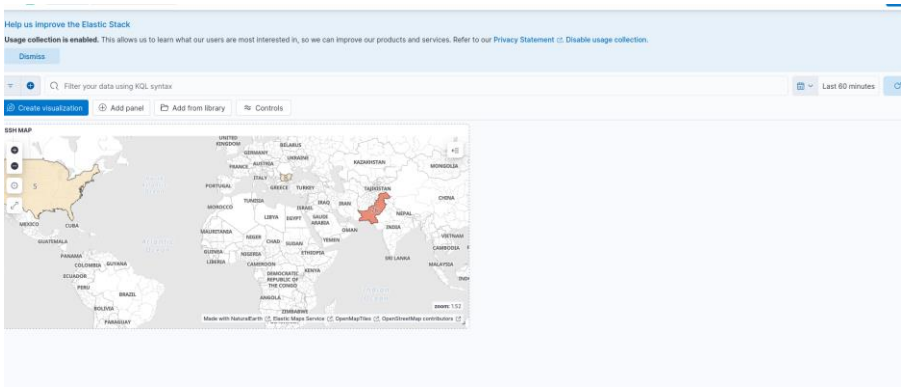

Maps
Create
Settings
Inspect
Full screen
Save
system.auth.ssh.event : * and agent.name :"My-Fleet-Server" and system.auth.ssh.event :"Failed"
Last 15 minutes
Refresh
LAYERS
World Countries
Source details
Layer settings
Name
Visibility
Zoom levels
0
24
Opacity
75
Include layer in fit to data bounds computation
Show tooltips
Tooltip fields
ISO 3166-1 alpha-2 code
Add
Joins
Join with terms from source.geo.country_iso_code
and use metric count where -- add filter --
Apply global search to join
Save map
Title
SSH MAP
Description
Optional
Add to dashboard
Existing
Select dashboard
New
None
Add to library
Cancel
Save and go to Dashboard
CANADA
MEXICO
CUBA
GUATEMALA
PANAMA
COLOMBIA
GUYANA
ECUADOR
PERU
BRAZIL
BOLIVIA
PARAGUAY
CHILE
ARGENTINA
REPUBLIC OF THE CONGO
ANGOLA
ZIMBABWE
BOTSWANA
SOUTH AFRICA
INDONESIA
PAPUA NEW GUINEA
AUSTRALIA

✧ **NOW THE DASHBOARD LOOKS LIKE THIS:**

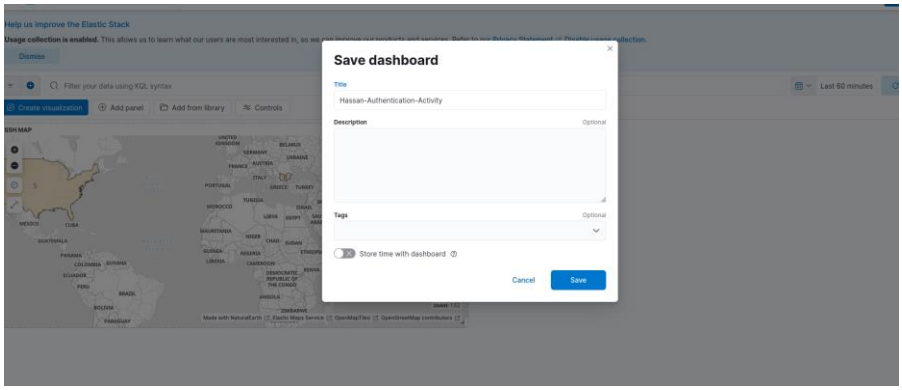


✧

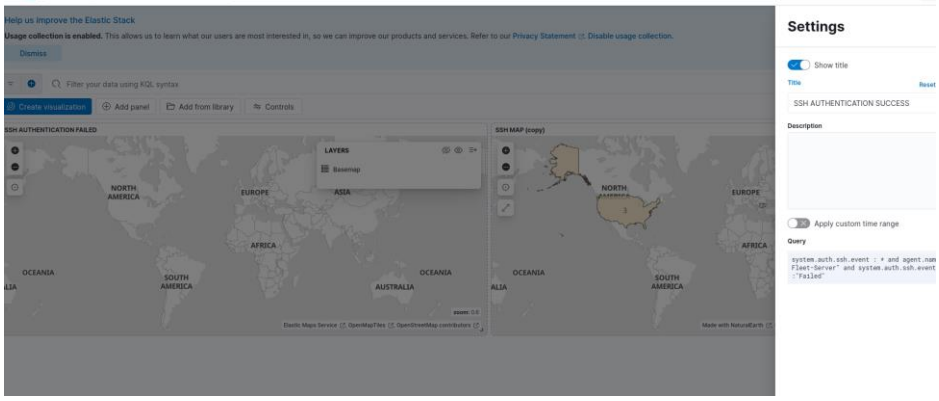
- **LAST 60 MINUTES YOU CAN SEE:**



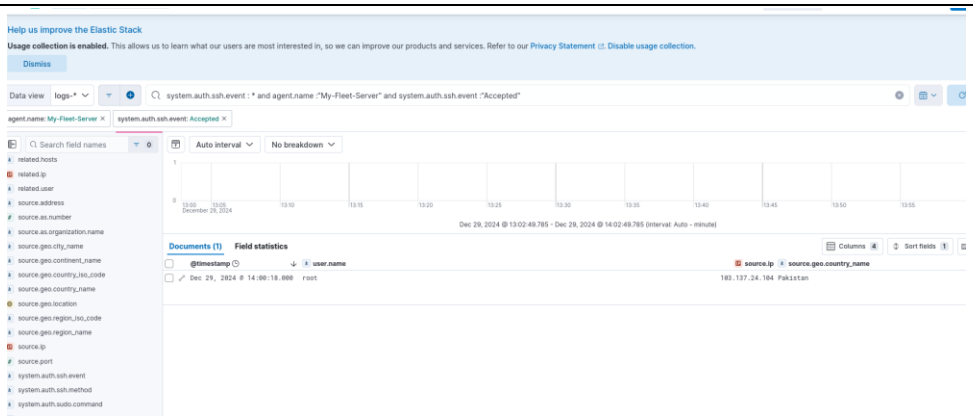
- **now save dashboard :**



- **NOW MAKE A COPY OF THE MAP AND CHANGE THE SETTING AS ONE IS FAILED AND OTHER IS SUCCES:**



- **as accepted is like this:**



- **SO this is our project to setup and ELK and ubuntu and windows server also using fleet server to detect the attacks, and uisng the VULTR cloud proivder to host the server in the internet.**