

## **1.List some logging and visualization tools available in the market with the preferred scenario to use one over the other.**

Below are some of the logging and visualization tools and their features and scenario of their uses:

**Datadog:** In addition to logging, this platform also enables teams to look at analytics and other features. Users can use the service to find all of their logs in a single interface. Moreover, the platform can help DevOps teams deal with outages that may happen across the globe. Various data related to web and app traffic is also available.

**Dynatrace:** It offers team members a wide range of log visualization tools and data. The platform also has business analytics and allows DevOps managers to analyze their apps' performances. Digital experience monitoring is also possible.

**Logz.io:** The Logz.io platform has a range of log visualization tools, such as app insights and ELK stack-based log management solutions. Users can also set up alerts if they suffer system outages. Logz.io also has security and DevOps analytics solutions, plus monitoring tools for business apps. Users can index up to 1 GB of log data for free.

**Graylog:** Its primary offerings are a range of log analysis tools. Users have access to numerous features, including alerts and compliance. The tool serves multiple industries and job functions, such as government, compliance and audit, telecom, fintech, and education. DevOps teams can also use Graylog to look over their app performance, along with identifying security breaches. A wide range of data options is also available.

**Google Cloud Logging:** Features of Google Cloud Logging include importing custom log data, as well as real-time analysis. Teams can also use the platform to carry out audits and report problems that may arise. A wide range of third-party integrations is also available.

**Scalyr:** It offers a wide range of app architecture and log visualization tools. Users will discover a user-friendly interface, as well as features to monitor in case problems arise. Data is also available soon after collection. We will also find that the platform offers comprehensive data for all events that take place. Moreover, we can integrate various data tools – including Logstash.

**LOGalyze:** LOGalyze allows DevOps teams to uncover events as they happen and break down their searches within multiple categories. LOGalyze supports various systems, including Windows, Linux, and OS. Logging customization is also available for both native and custom apps. Users can also analyze their logging data by correlating trends, understanding patterns, and individual event management. You can also tag each event and create separate categories.

Logiq.ai: Using the platform, you can set up comprehensive monitoring for your cloud and app infrastructures. Moreover, you can create alerts that prompt you to take action whenever something has gone wrong or about to go wrong. Logiq.ai also comes with an integrated user interface, making log data management and monitoring easier and better-laid out. The platform also enables you to incorporate logs from Openstack, Kubernetes, and other IaaS services. On the Logiq.ai website, you'll also find a wide range of resources, such as blog posts, tutorials, and eBooks – to further expand your knowledge.

## **2. Mention 10 best practises when logging. Why is log formatting necessary?**

Below are the best practices while logging:

- Logging of only what is needed rather than logging everything
- using a different log level for different logs according to their characteristics and effect they have on the application or system  
(eg: FATAL, ERROR, INFO, WARN, TRACE, DEBUG, NOTICE may be different log levels)
- Writing meaningful log messages
- Adding context to log messages ( acknowledging the cause of the log message)
- Logging in machine parseable format.
- Logging for the purpose of auditing, profiling and maintaining statistics rather than just for troubleshooting purpose.
- Not logging sensitive information like passwords, credit card numbers, social security numbers, session identifiers, authorization tokens etc.
- Making the logs human readable (eg. using standard date and time format)
- Centralizing logs in cases where logs are generated from multiple servers so that accessing them is easier.
- Rotating the logs periodically to restrict the volume of the log data.

Normally logs are unformatted text data. In order to infer information from a lot of logs it is hard for us to just scan through the texts and find the information. If logs are formatted (eg. we may format the logs in JSON) then it becomes easier to query the desired information. Also, log formatting makes the logs more human readable as well as machine parsable at the same time.

**3.Create a file in your system. Whenever someone performs some action(read, write, execute) on that file, the event should be logged somewhere.**

We install auditd which is a service used to audit the changes in files and system.

```
(kali@kali:~)$ sudo apt install audiotd
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
baobab caribou cryptsetup-run folks-common gir1.2-caribou-1.0 gir1.2-clutter-1.0 gir1.2-cogl-1.0 gir1.2-coglpango-1.0 gir1.2-handry-0.0 gnome-accessibility-themes
gnome-characters gnome-contacts gnome-core gnome-font-viewer gnome-logs gnome-online-miners gnome-themes-extra gnome-themes-extra-data gnome-tweak-tool
gststreamer1.0-packagekit gtk2-engines-murrine libamt-5.0 libamt-5-common libamt-1.2-62 libcaribou-common libcaribou0 libdap27 libdapclient6v5 libdav1d4
libdataserver-1.2-25 libdataserverui-1.2-2 libepsilont1 libextutils-pkgconfig-perl libfolks-eds26 libfolks26 libgdal28 libgeos-3.9.0 libgfbgraph-0.2-0
libgupnp-1.2 libhandy-0.0-0 libidn1 libimusicbrainz5-2 libimusicbrainz25 libnetcdf18 libntfs-3g883 libomp-11-dev libomp5-11 libproj19 libquvi-0.9-0.9.3
libquvi-scripts-0.9 libtepl-5.0 libtracker-control-2.0-0 libtracker-miner-2.0-0 libtracker-sparql-2.0-0 liburcu libx265-192 libxface4util-bin libxface4util-common
libxface4util1 libxconf-0.3 libxmlb1 libyara4 libzapojit-0.0-0 linux-image-5.10.0-kali7-amd64 lua-bitop lua-expat lua-luaos lua-socket moonshine python3-editor
python3-exif python3-gevent python3-gevent-websocket python3-greenlet python3-ipython-genutils python3-jupyter-core python3-m2crypto python3-nbformat
python3-parameterized python3-plopy python3-pylnk python3-stem python3-zope.event xfcconf
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
libaaparse0
Suggested packages:
audispd-plugins
The following NEW packages will be installed:
audiotd libaaparse0
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 288 kB of archives.
After this operation, 948 kB of additional disk space will be used.
Do you want to continue? [y/n] y
```

## We enable the service auditd

```
(kalilinux@kali)-[~]
$ sudo systemctl enable auditd
Synchronizing state of auditd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable auditd
Created symlink /etc/systemd/system/multi-user.target.wants/auditd.service → /lib/systemd/system/auditd.service.

(kalilinux@kali)-[~]
```

After enabling the service, we start it

```
(kalilinux@kali)-[~]  
$ sudo systemctl start audit  
  
(kalilinux@kali)-[~]
```

We then verify that the service is properly running by viewing it's status

```
(kalilinux@kali)-[~]
└─$ sudo systemctl status auditd
● auditd.service - Security Auditing Service
   Loaded: loaded (/lib/systemd/system/auditd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2021-11-27 20:44:07 +0545; 2s ago
     Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
   Process: 26084 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
   Process: 26088 ExecStartPost=/sbin/auditd --load (code=exited, status=0/SUCCESS)
  Main PID: 26085 (auditd)
    Tasks: 2 (limit: 9328)
   Memory: 704.0K
      CPU: 79ms
   CGroup: /system.slice/auditd.service
           └─26085 /sbin/auditd

Nov 27 20:44:07 kali augenrules[26098]: backlog_wait_time_actual 0
Nov 27 20:44:07 kali augenrules[26098]: enabled 1
Nov 27 20:44:07 kali augenrules[26098]: failure 1
Nov 27 20:44:07 kali augenrules[26098]: pid 26085
Nov 27 20:44:07 kali augenrules[26098]: rate_limit 0
Nov 27 20:44:07 kali augenrules[26098]: backlog_limit 8192
Nov 27 20:44:07 kali augenrules[26098]: lost 0
Nov 27 20:44:07 kali augenrules[26098]: backlog 0
Nov 27 20:44:07 kali augenrules[26098]: backlog_wait_time 60000
Nov 27 20:44:07 kali augenrules[26098]: backlog_wait_time_actual 0
```

We check the rules and status of the audit system as follows(currently there are no rules)

```
(kalilinux@kali)-[~]
└─$ sudo auditctl -l
No rules

(kalilinux@kali)-[~]
└─$ sudo auditctl -s
enabled 1
failure 1
pid 26085
rate_limit 0
backlog_limit 8192
lost 0
backlog 0
backlog_wait_time 60000
backlog_wait_time_actual 0
loginuid_immutable 0 unlocked

(kalilinux@kali)-[~]
```

Now we create a file named samana.

```
(kalilinux㉿kali)-[~]  
$ touch samana  
  
(kalilinux㉿kali)-[~]
```

Now we set auditctl rule to set a watch for filesystem where it watches for read write execute and attribute change in the file (rwx)

We associate the rule with a key named `key1` to identify the rule with the key

```
(kalilinux@kali)-[~]  
$ sudo auditctl -w /home/kalilinux/samana -p rwx -k key1
```

Now we can see the rules listed as below:

```
(kalilinux㉿kali)-[~]
└─$ sudo auditctl -l
-w /home/kalilinux/sam -p rwx -k key
-w /home/kalilinux/samana -p rwx -k key1
(kalilinux㉿kali)-[~]
└─$
```

Now we write on that file named samana

```
(kalilinux㉿kali)-[~]  
$ nano samana
```

```
GNU nano 5.9 samana *
hello!!!!
```

Also we read the contents of that file

```
(kalilinux@kali)-[~]
$ cat samana
hello!!!!
(kalilinux@kali)-[~]
```

After performing read and write permissions on that file, we search for the audit logs using our key named “key1”.

Below we can see the logs that show what actions have been performed on our specified file. We can also see the mediums through which our file was read and written (namely cat and nano)

```
(kalilinux@kali)-[~]
$ sudo ausearch -k key1
-----
time-->Sat Nov 27 20:57:18 2021
type=PROCTITLE msg=audit(1638025938.904:73): proctitle=617564697463746C002D77002F686F6D652F68616C696C696E75782F73616D616E61002D700072777861002D68006B657931
type=PATH msg=audit(1638025938.904:73): item=0 name="/home/kalilinux/" inode=11935078 dev=08:01 mode=040755 ouid=1000 ogid=1000 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1638025938.904:73): cwd="/home/kalilinux"
type=SOCKADDR msg=audit(1638025938.904:73): saddr=10000000000000000000000000000000
type=SYSCALL msg=audit(1638025938.904:73): arch=c000003e syscall=44 success=yes exit=1084 a0=4 a1=7ffc065951d0 a2=43c a3=0 items=1 ppid=26588 pid=26589 auid=1000 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts3 ses=3 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1638025938.904:73): auid=1000 ses=3 subj=unconfined op=add_rule key="key1" list=4 res=1
-----
time-->Sat Nov 27 20:57:49 2021
type=PROCTITLE msg=audit(1638025969.820:76): proctitle=6E616E6F0073616D616E61
type=PATH msg=audit(1638025969.820:76): item=0 name="/home/kalilinux/samana" inode=11927833 dev=08:01 mode=0100644 ouid=1000 ogid=1000 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1638025969.820:76): cwd="/home/kalilinux"
type=SYSCALL msg=audit(1638025969.820:76): arch=c000003e syscall=257 success=yes exit=3 a0=ffffff9c a1=557c1db57b00 a2=0 a3=0 items=1 ppid=26215 pid=26617 auid=1000 uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts4 ses=3 comm="nano" exe="/usr/bin/nano" subj=unconfined key="key1"
-----
time-->Sat Nov 27 20:58:14 2021
type=PROCTITLE msg=audit(1638025994.976:77): proctitle=6E616E6F0073616D616E61
type=PATH msg=audit(1638025994.976:77): item=1 name="samana" inode=11927833 dev=08:01 mode=0100644 ouid=1000 ogid=1000 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1638025994.976:77): item=0 name="/home/kalilinux/" inode=11935078 dev=08:01 mode=040755 ouid=1000 ogid=1000 rdev=00:00 nametype=PARENT cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1638025994.976:77): cwd="/home/kalilinux"
type=SYSCALL msg=audit(1638025994.976:77): arch=c000003e syscall=257 success=yes exit=3 a0=ffffff9c a1=557c1db60c90 a2=241 a3=1b6 items=2 ppid=26215 pid=26617 auid=1000 uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts4 ses=3 comm="nano" exe="/usr/bin/nano" subj=unconfined key="key1"
-----
time-->Sat Nov 27 20:58:15 2021
type=PROCTITLE msg=audit(1638025995.252:78): proctitle="/usr/libexec/tracker-extract-3"
type=PATH msg=audit(1638025995.252:78): item=0 name="/home/kalilinux/samana" inode=11927833 dev=08:01 mode=0100644 ouid=1000 ogid=1000 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1638025995.252:78): cwd="/home/kalilinux"
type=SYSCALL msg=audit(1638025995.252:78): arch=c000003e syscall=257 success=yes exit=11 a0=ffffff9c a1=5640d8fa4da0 a2=40000 a3=0 items=1 ppid=1722 pid=26632 auid=1000 uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=(none) ses=3 comm="tracker-extract" exe="/usr/libexec/tracker-extract-3" subj=unconfined key="key1"
-----
time-->Sat Nov 27 20:58:31 2021
type=PROCTITLE msg=audit(1638026011.764:79): proctitle=6361740073616D616E61
type=PATH msg=audit(1638026011.764:79): item=0 name="samana" inode=11927833 dev=08:01 mode=0100644 ouid=1000 ogid=1000 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1638026011.764:79): cwd="/home/kalilinux"
type=SYSCALL msg=audit(1638026011.764:79): arch=c000003e syscall=257 success=yes exit=3 a0=ffffff9c a1=7fff4ca653a6 a2=0 a3=0 items=1 ppid=26215 pid=26676 auid=1000 uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts4 ses=3 comm="cat" exe="/usr/bin/cat" subj=unconfined key="key1"
(kalilinux@kali)-[~]
```

4. install logstash in your system. download a sample nginx log from [https://github.com/elastic/examples/blob/master/Common%20Data%20Formats/nginx\\_logs/nginx\\_logs](https://github.com/elastic/examples/blob/master/Common%20Data%20Formats/nginx_logs/nginx_logs) , parse the logs using logstash. The parsed output must contain the geographical information like country, state etc. that the request is originating from. save the parsed output to a file in your system.

Before installing logstash we check if java is installed in our system

```
(kalilinux@kali)-[~]
└─$ java -version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
openjdk version "11.0.13" 2021-10-19
OpenJDK Runtime Environment (build 11.0.13+8-post-Debian-1)
OpenJDK 64-Bit Server VM (build 11.0.13+8-post-Debian-1, mixed mode, sharing)

(kalilinux@kali)-[~]
```

We add the GPG key for elasticsearch

```
(kalilinux@kali)-[~]
└─$ wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
[sudo] password for kalilinux:
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK

(kalilinux@kali)-[~]
```

Then, We install apt-transport-https

```
(kalilinux@kali)-[~]
└─$ sudo apt-get install apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.3.11).
The following packages were automatically installed and are no longer required:
baobab caribou cryptsetup-run folks-common girl1.2-caribou-1.0 girl1.2-clutter-1.0 girl1.2-cogl-1.0 girl1.2-coglpango-1.0 girl1.2-handy-0.0 gnome-accessibility-themes
gnome-characters gnome-contacts gnome-core gnome-font-viewer gnome-logs gnome-online-miners gnome-themes-extra gnome-themes-extra-data gnome-tweak-tool
gststreamer1.0-packagekit gtk2-engines-murrine libamtk-5-0 libamtk-5-common libcamel-1.2-62 libcaribou-common libcaribou0 libdap27 libdapclient6v5 libdav1d4
libedataserver-1.2-25 libedataserverui-1.2-2 libepsilon1 libextutils-pkgconfig-perl libfolks-eds26 libfolks26 libgdal28 libgeos-3.9.0 libgfbgraph-0.2-0
libgupnp-1.2-0 libhandy-0.0-0 libidn11 libmusicbrainz5-2 libmusicbrainz5cc2v5 libnetcdf18 libntfs-3g883 libomp-11-dev libomp5-11 libproj19 libquvi-0.9-0.9.3
libquvi-scripts-0.9 libtepl-5-0 libtracker-control-2.0-0 libtracker-miner-2.0-0 libtracker-sparql-2.0-0 liburcu6 libx265-192 libxfce4util-bin libxfce4util-common
libxfce4util7 libxfceconf-0-3 libxmlb1 libyara4 libzapojit-0.0-0 linux-image-5.10.0-kali7-amd64 lua-bitop lua-expat lua-json lua-socket mousepad python3-editor
python3-exif python3-gevent python3-gevent-websocket python3-greenlet python3-ipython-genutils python3-jupyter-core python3-m2crypto python3-nbformat
python3-parameterized python3-plotly python3-pylnk python3-stem python3-zope.event xfcconf
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

(kalilinux@kali)-[~]
```



Then we add the elastic package repository to our repository list

```
(kalilinux@kali)-[~]
└─$ echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
(kalilinux@kali)-[~]
```

Now after performing `sudo apt-get update`, we install logstash as follows:

```
(kalilinux@kali)-[~]
└─$ sudo apt-get install logstash
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
logstash is already the newest version (1:7.15.2-1).
The following packages were automatically installed and are no longer required:
  baobab caribou cryptsetup-run folks-common gir1.2-caribou-1.0 gir1.2-clutter-1.0 gir1.2-cogl-1.0 gir1.2-coglpango-1.0 gir1.2-handy-0.0 gnome-accessibility-themes
  gnome-characters gnome-contacts gnome-core gnome-font-viewer gnome-logs gnome-online-miners gnome-themes-extra gnome-themes-extra-data gnome-tweak-tool
  gstreamer1.0-packagekit gtk2-engines-murrine libamtk-5-0 libamtk-5-common libcamel-1.2-62 libcaribou-common libcaribou0 libdap27 libdapclient6v5 libdav1d4
  libdataserver-1.2-25 libdataserverui-1.2-2 libepsilon1 libextutils-pkgconfig-perl libfolks-eds26 libfolks26 libgdal28 libgeos-3.9.0 libgfbgraph-0.2-0
  libgupnp-1.2-0 libhandy-0.0-0 libidn11 libmusicbrainz5-2 libmusicbrainz5cc2v5 libnetcdf18 libntfs-3g883 libomp-11-dev libomp5-11 libproj19 libquvi-0.9-0.9.3
  libquvi-scripts-0.9 libtepl-5-0 libtracker-control-2.0-0 libtracker-miner-2.0-0 libtracker-sparql-2.0-0 liburcu6 libx265-192 libxfce4util-bin libxfce4util-common
  libxfce4util7 libxfce4conf-0-3 libxmlb1 libyara4 libzapojit-0.0-0 linux-image-5.10.0-kali7-amd64 lua-bitop lua-expat lua-json lua-socket mousepad python3-editor
  python3-exif python3-gevent python3-gevent-websocket python3-greenlet python3-ipython-genutils python3-jupyter-core python3-m2crypto python3-nbformat
  python3-parameterized python3-plotly python3-pylnk python3-stem python3-zope.event xfcconf
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
(kalilinux@kali)-[~]
```

Now we download the sample nginx logs and copy it to the `/var/log/nginx` directory under the name `nginx_logs.log`

```
(kalilinux@kali)-[/var/log/nginx]
└─$ sudo cp /home/kalilinux/Downloads/nginx_logs nginx_logs.log
[sudo] password for kalilinux:
(kalilinux@kali)-[/var/log/nginx]
└─$ sudo nano nginx_logs.log
(kalilinux@kali)-[/var/log/nginx]
```



Here we can see the sample nginx logs as follows:

```
GNU nano 5.9 nginx_logs.log
93.180.71.3 - - [17/May/2015:08:05:32 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)"
93.180.71.3 - - [17/May/2015:08:05:23 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)"
80.91.33.133 - - [17/May/2015:08:05:24 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.17)"
217.168.17.5 - - [17/May/2015:08:05:34 +0000] "GET /downloads/product_1 HTTP/1.1" 200 490 "-" "Debian APT-HTTP/1.3 (0.8.10.3)"
217.168.17.5 - - [17/May/2015:08:05:09 +0000] "GET /downloads/product_2 HTTP/1.1" 200 490 "-" "Debian APT-HTTP/1.3 (0.8.10.3)"
93.180.71.3 - - [17/May/2015:08:05:57 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)"
217.168.17.5 - - [17/May/2015:08:05:02 +0000] "GET /downloads/product_2 HTTP/1.1" 404 337 "-" "Debian APT-HTTP/1.3 (0.8.10.3)"
217.168.17.5 - - [17/May/2015:08:05:42 +0000] "GET /downloads/product_1 HTTP/1.1" 404 332 "-" "Debian APT-HTTP/1.3 (0.8.10.3)"
80.91.33.133 - - [17/May/2015:08:05:01 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.17)"
93.180.71.3 - - [17/May/2015:08:05:27 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)"
217.168.17.5 - - [17/May/2015:08:05:12 +0000] "GET /downloads/product_2 HTTP/1.1" 200 3316 "-" "-"
188.138.60.101 - - [17/May/2015:08:05:49 +0000] "GET /downloads/product_2 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
80.91.33.133 - - [17/May/2015:08:05:14 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.16)"
46.4.66.76 - - [17/May/2015:08:05:45 +0000] "GET /downloads/product_1 HTTP/1.1" 404 318 "-" "Debian APT-HTTP/1.3 (1.0.1ubuntu2)"
93.180.71.3 - - [17/May/2015:08:05:26 +0000] "GET /downloads/product_1 HTTP/1.1" 404 324 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)"
91.234.194.89 - - [17/May/2015:08:05:22 +0000] "GET /downloads/product_2 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
80.91.33.133 - - [17/May/2015:08:05:07 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.17)"
37.26.93.214 - - [17/May/2015:08:05:38 +0000] "GET /downloads/product_2 HTTP/1.1" 404 319 "-" "Go 1.1 package http"
188.138.60.101 - - [17/May/2015:08:05:25 +0000] "GET /downloads/product_2 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
93.180.71.3 - - [17/May/2015:08:05:11 +0000] "GET /downloads/product_1 HTTP/1.1" 404 340 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)"
46.4.66.76 - - [17/May/2015:08:05:02 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (1.0.1ubuntu2)"
62.75.198.179 - - [17/May/2015:08:05:06 +0000] "GET /downloads/product_2 HTTP/1.1" 200 490 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
80.91.33.133 - - [17/May/2015:08:05:55 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.16)"
173.203.139.108 - - [17/May/2015:08:05:53 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
210.245.80.75 - - [17/May/2015:08:05:32 +0000] "GET /downloads/product_2 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (1.0.1ubuntu2)"
46.4.83.163 - - [17/May/2015:08:05:52 +0000] "GET /downloads/product_2 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
91.234.194.89 - - [17/May/2015:08:05:18 +0000] "GET /downloads/product_2 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.9.7.9)"
31.22.86.126 - - [17/May/2015:08:05:24 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.16)"
217.168.17.5 - - [17/May/2015:08:05:25 +0000] "GET /downloads/product_1 HTTP/1.1" 200 3301 "-" "-"
80.91.33.133 - - [17/May/2015:08:05:50 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.22)"
[ Read 51462 lines ]
```

Now we create a config file for logstash named nginx.conf(since we are parsing the nginx logs)

```
(kalilinux@kali)-[~]
└─$ sudo nano nginx.conf
└─(kalilinux@kali)-[~]
```

The logstash configuration file is created as follows:

```
GNU nano 5.9 nginx.conf
input {
  file {
    type => "nginx-log"
    path => "/var/log/nginx/nginx_logs.log"
    sinedb_path => "/dev/null"
    start_position => "beginning"
  }
}
filter {
  grok{
    match=>{
      "message"=>"%{IP:clientip} \- \- \["
    }
  }
  geoip {
    source => "clientip"
  }
}
output {
  file {
    path => "/var/log/logstash/geo_parsed_nginx.log"
  }
}
```

Now we run the created configuration file as follows:

```
(kalilinux@kali)-[/usr/share/logstash]
└─$ sudo bin/logstash --path.settings /etc/logstash --path.data sensor39 -f /home/kalilinux/nginx.conf
Using bundled JDK: /usr/share/logstash/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
```

```
(kalilinux@kali)-[/usr/share/logstash]
└─$ sudo bin/logstash --path.settings /etc/logstash --path.data sensor39 -f /home/kalilinux/nginx.conf
Using bundled JDK: /usr/share/logstash/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
[2021-11-27T20:02:13,145][INFO ][logstash.runner] Log4j configuration path used is: /etc/logstash/log4j2.properties
[2021-11-27T20:02:13,164][INFO ][logstash.runner] Starting Logstash {"logstash.version":"7.15.2", "jruby.version":"jruby 9.2.19.0 (2.5.8) 2021-06-15 55810c55
2b OpenJDK 64-Bit Server VM 11.0.12+7 on 11.0.12+7 +indy +jit [linux-x86_64]}
[2021-11-27T20:02:13,973][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line options are specified
[2021-11-27T20:02:16,894][INFO ][logstash.agent] Successfully started Logstash API endpoint {:port=>9600}
[2021-11-27T20:02:18,345][INFO ][org.reflections.Reflections] Reflections took 176 ms to scan 1 urls, producing 120 keys and 417 values
[2021-11-27T20:02:20,940][WARN ][org.logstash.instrument.metrics.gauge.LazyDelegatingGauge][main] A gauge metric of an unknown type (org.jruby.RubySymbol) has been creat
ed for key: status. This may result in invalid serialization. It is recommended to log an issue to the responsible developer/development team.
[2021-11-27T20:02:20,973][WARN ][org.logstash.instrument.metrics.gauge.LazyDelegatingGauge][main] A gauge metric of an unknown type (org.jruby.RubySymbol) has been creat
ed for key: status. This may result in invalid serialization. It is recommended to log an issue to the responsible developer/development team.
[2021-11-27T20:02:20,973][WARN ][org.logstash.instrument.metrics.gauge.LazyDelegatingGauge][main] A gauge metric of an unknown type (org.jruby.RubySymbol) has been creat
ed for key: status. This may result in invalid serialization. It is recommended to log an issue to the responsible developer/development team.
[2021-11-27T20:02:23,499][INFO ][logstash.filters.geopip.downloadmanager] new database version detected? false
[2021-11-27T20:02:23,769][INFO ][logstash.filters.geopip.database.manager][main] By not manually configuring a database path with 'database =>', you accepted and agreed Ma
xMind EULA. For more details please visit https://www.maxmind.com/en/geolite2/eula
[2021-11-27T20:02:23,772][INFO ][logstash.filters.geopip] [main] Using geopip database {:path=>sensor39/plugins/filters/geopip/1638021268/GeoLite2-City.mmdb"}
[2021-11-27T20:02:23,912][INFO ][logstash.javapipeline] [main] Starting pipeline {:pipeline_id=>"main", "pipeline.workers">4, "pipeline.batch.size">125, "pipeline.b
atch.delay">500, "pipeline.max.inflight">500, "pipeline.sources">[/home/kalilinux/nginx.conf], :thread=>#<Thread:0x7faeaf run>}
[2021-11-27T20:02:25,396][INFO ][logstash.javapipeline] [main] Pipeline Java execution initialization time {"seconds">1.48}
[2021-11-27T20:02:25,492][INFO ][logstash.javapipeline] [main] Pipeline started {"pipeline_id">"main"}
[2021-11-27T20:02:25,608][INFO ][logstash.agent] [main] Pipelines running {:count=>1, :running_pipelines=>[main], :non_running_pipelines=>[]}
[2021-11-27T20:02:25,617][INFO ][filewatch.observingtail] [main][dcce2e62ea9c843ce55d0655d2b3ace1cf2e6cb1c183a7d890a2aa99ee8db256] START, creating Discoverer, Watch wit
h file and sinedb collections
[2021-11-27T20:02:27,138][INFO ][logstash.outputs.file] [main][572e934efa8b2c585f89365dab120384ce2b50bda81852589819fa7b8bac2ccf] Opening file {:path=>"/var/log/logsta
sh/geo_parsed_nginx.log"}
```

Now we can see the parsed files in /var/logs/logstash as follows

```
(kali@kali)~[/var/log/logstash]
$ ls
geo_parsed_nginx.log  logstash-deprecation.log  logstash-json.log  logstash-plain.log  logstash-slowlog-json.log  logstash-slowlog-plain.log
(kali@kali)~[/var/log/logstash]
```

The output file in our configuration file was named as geo\_parsed\_nginx.log which now has the following contents(continent\_code,country\_name,country\_code,latitude,timezone,location etc)

```
(kali@kali)~[/var/log/logstash]
$ cat geo_parsed_nginx.log
{"@version":"1","host":"kali","@timestamp":"2021-11-27T14:08:28.108Z","path":"/var/log/nginx/nginx_logs.log","clientip":"93.180.71.3","geoip":{"continent_code":"EU","country_name":"Netherlands","country_code3":"NL","latitude":52.3824,"timezone":"Europe/Amsterdam","ip":"93.180.71.3","location":{"lat":52.3824,"lon":4.8995},"longitude":4.8995,"country_code2":"NL"},"type":"nginx-log","message":"93.180.71.3 - - [17/May/2015:08:05:32 +0000] \"GET /downloads/product_1 HTTP/1.1\" 304 0 \"-\" \"Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)\""}
{"@version":"1","host":"kali","@timestamp":"2021-11-27T14:14:39.616Z","path":"/var/log/nginx/nginx_logs.log","clientip":"80.91.33.133","geoip":{"continent_code":"EU","country_code3":"NO","region_name":"Oslo County","location":{"lat":59.955,"lon":10.859},"city_name":"Oslo","country_name":"Norway","region_code":"03","latitude":59.955,"postal_code":"0114","ip":"80.91.33.133","timezone":"Europe/Oslo","longitude":10.859,"country_code2":"NO"},"type":"nginx-log","message":"80.91.33.133 - - [17/May/2015:08:05:24 +0000] \"GET /downloads/product_1 HTTP/1.1\" 304 0 \"-\" \"Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.17)\""}
{"@version":"1","host":"kali","@timestamp":"2021-11-27T14:14:39.618Z","path":"/var/log/nginx/nginx_logs.log","clientip":"217.168.17.5","geoip":{"continent_code":"EU","country_code3":"GB","region_name":"Essex","location":{"lat":51.5645,"lon":0.4636},"city_name":"Basildon","country_name":"United Kingdom","region_code":"ESS","latitude":51.5645,"postal_code":"SS16","ip":"217.168.17.5","timezone":"Europe/London","longitude":0.4636,"country_code2":"GB"},"type":"nginx-log","message":"217.168.17.5 - - [17/May/2015:08:05:02 +0000] \"GET /downloads/product_2 HTTP/1.1\" 404 337 \"-\" \"Debian APT-HTTP/1.3 (0.8.10.3)\""}
{"@version":"1","host":"kali","@timestamp":"2021-11-27T14:14:39.620Z","path":"/var/log/nginx/nginx_logs.log","clientip":"217.168.17.5","geoip":{"continent_code":"EU","country_code3":"GB","region_name":"Essex","location":{"lat":51.5645,"lon":0.4636},"city_name":"Basildon","country_name":"United Kingdom","region_code":"ESS","latitude":51.5645,"postal_code":"SS16","ip":"217.168.17.5","timezone":"Europe/London","longitude":0.4636,"country_code2":"GB"},"type":"nginx-log","message":"217.168.17.5 - - [17/May/2015:08:05:12 +0000] \"GET /downloads/product_2 HTTP/1.1\" 200 3316 \"-\" \"-\""}
{"@version":"1","host":"kali","@timestamp":"2021-11-27T14:14:39.621Z","path":"/var/log/nginx/nginx_logs.log","clientip":"93.180.71.3","geoip":{"continent_code":"EU","country_name":"Netherlands","country_code3":"NL","latitude":52.3824,"timezone":"Europe/Amsterdam","ip":"93.180.71.3","location":{"lat":52.3824,"lon":4.8995},"longitude":4.8995,"country_code2":"NL"},"type":"nginx-log","message":"93.180.71.3 - - [17/May/2015:08:05:26 +0000] \"GET /downloads/product_1 HTTP/1.1\" 404 324 \"-\" \"Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.21)\""}
{"@version":"1","host":"kali","@timestamp":"2021-11-27T14:14:39.624Z","path":"/var/log/nginx/nginx_logs.log","clientip":"188.138.60.101","geoip":{"continent_code":"EU","country_code3":"FR","region_name":"Bas-Rhin","location":{"lat":48.6025,"lon":7.7844},"city_name":"Strasbourg","country_name":"France","region_code":"67","latitude":48.6025,"postal_code":"67000","ip":"188.138.60.101","timezone":"Europe/Paris","longitude":7.7844,"country_code2":"FR"},"type":"nginx-log","message":"188.138.60.101 - - [17/May/2015:08:05:25 +0000] \"GET /downloads/product_2 HTTP/1.1\" 304 0 \"-\" \"Debian APT-HTTP/1.3 (0.9.7.9)\""}
{"@version":"1","host":"kali","@timestamp":"2021-11-27T14:14:39.626Z","path":"/var/log/nginx/nginx_logs.log","clientip":"80.91.33.133","geoip":{"continent_code":"EU","country_code3":"NO","region_name":"Oslo County","location":{"lat":59.955,"lon":10.859},"city_name":"Oslo","country_name":"Norway","region_code":"03","latitude":59.955,"postal_code":"0114","ip":"80.91.33.133","timezone":"Europe/Oslo","longitude":10.859,"country_code2":"NO"},"type":"nginx-log","message":"80.91.33.133 - - [17/May/2015:08:05:55 +0000] \"GET /downloads/product_1 HTTP/1.1\" 304 0 \"-\" \"Debian APT-HTTP/1.3 (0.8.16-exp12ubuntu10.16)\""}

```