Centralized Logging & Monitoring using ELK Stack on GCP

This project demonstrates a cloud-based centralized logging architecture using the ELK stack. Two virtual machines were deployed on Google Cloud Platform: one for hosting the ELK stack and another running a web server that generates traffic logs. Logs are shipped in real time, parsed, indexed, and visualized through Kibana dashboards.

Architecture Design

The system consists of:

- **monitor-vm** → Elasticsearch, Logstash, Kibana
- **web-vm** → Nginx web server and Filebeat log shipper

Log Flow:

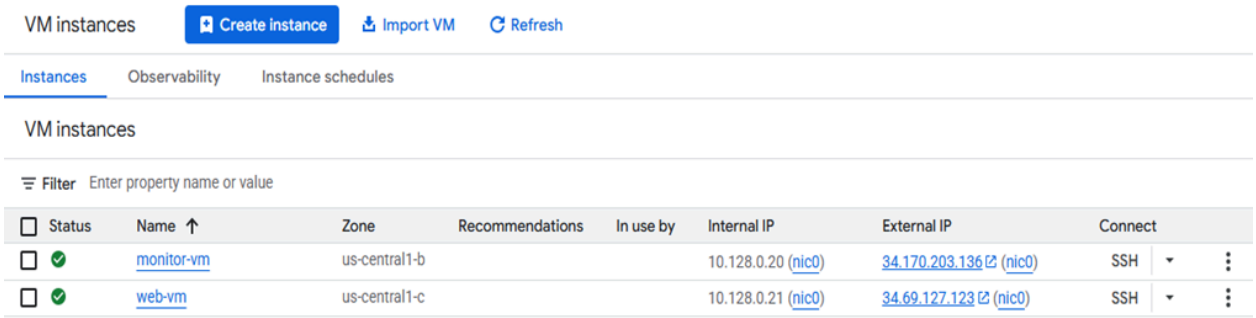
Nginx → Filebeat → Logstash → Elasticsearch → Kibana

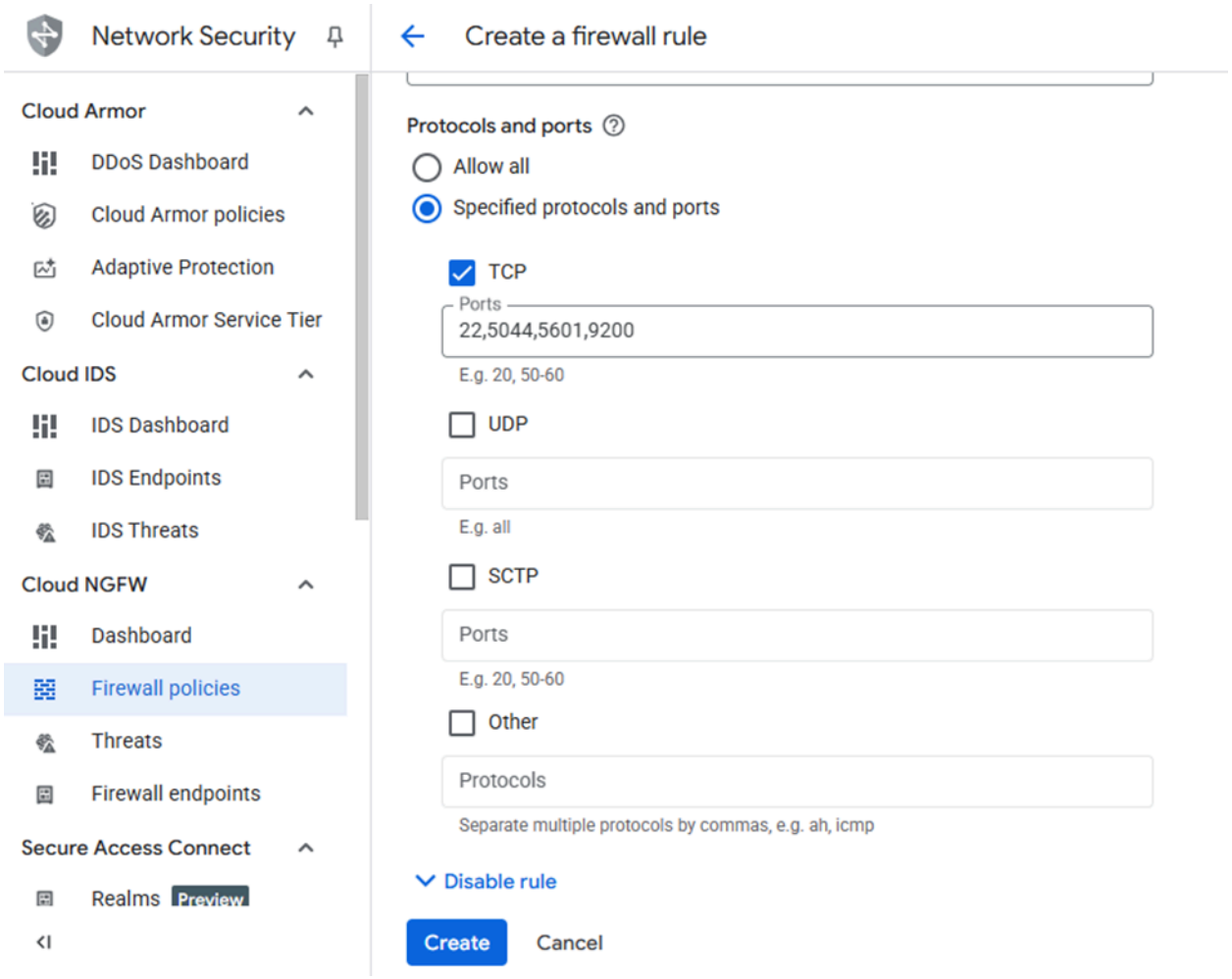
Step 1: Cloud Infrastructure Setup

Two Ubuntu 22.04 virtual machines were provisioned on Google Cloud Platform:

- monitor-vm to host Elasticsearch, Logstash, and Kibana
- web-vm to host Nginx and Filebeat

Firewall rules were configured to allow ports 22, 5044, 9200, and 5601 to enable secure communication between components.





Step 2: Elasticsearch Installation and Verification

Elasticsearch was installed on monitor-vm using the official Elastic repository.
The service was enabled and started using systemctl.

Cluster health was verified using:
curl http://localhost:9200/
Successful JSON response confirmed that Elasticsearch was running correctly.

```
mihirmashruwala@monitor-vm:~$ sudo systemctl daemon-reload
mihirmashruwala@monitor-vm:~$ sudo systemctl enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elast
mihirmashruwala@monitor-vm:~$ sudo systemctl start elasticsearch
mihirmashruwala@monitor-vm:~$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
     Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-11-16 18:43:10 UTC; 2s ago
       Docs: https://www.elastic.co
   Main PID: 2982 (java)
      Tasks: 96 (limit: 4687)
     Memory: 2.4G
        CPU: 1min 21.079s
     CGroup: /system.slice/elasticsearch.service
             ├─2982 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+UseSerialGC -Dcli.name=server -
             ├─3041 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddre
             └─3061 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Nov 16 18:42:28 monitor-vm systemd[1]: Starting Elasticsearch...
Nov 16 18:43:10 monitor-vm systemd[1]: Started Elasticsearch.
```

```
mihirmashruwala@monitor-vm:~$ sudo vim /etc/elasticsearch/elasticsearch.yml
mihirmashruwala@monitor-vm:~$ sudo systemctl restart elasticsearch
mihirmashruwala@monitor-vm:~$ curl http://localhost:9200/
{
  "name" : "monitor-vm",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "HQ2KB3vZSLO2yFjNQC4JwQ",
  "version" : {
    "number" : "8.19.7",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "198d86868932741b4e0d184425510217febc27d1",
    "build_date" : "2025-11-07T13:35:54.762042224Z",
    "build_snapshot" : false,
    "lucene_version" : "9.12.2",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Started kibana on monitor-vm

```
mihirmashruwala@monitor-vm:~$ sudo nano /etc/kibana/kibana.yml
mihirmashruwala@monitor-vm:~$ sudo systemctl restart kibana
sudo systemctl status kibana
● kibana.service - Kibana
     Loaded: loaded (/lib/systemd/system/kibana.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-11-16 19:15:51 UTC; 39ms ago
       Docs: https://www.elastic.co
   Main PID: 6407 (node)
      Tasks: 1 (limit: 4687)
     Memory: 1008.0K
        CPU: 27ms
     CGroup: /system.slice/kibana.service
             └─6407 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/kibana/bin/../src/cli/dist

Nov 16 19:15:51 monitor-vm systemd[1]: Started Kibana.
```

Step 3: Logstash Pipeline Configuration

A Logstash configuration file was created to define:

- Beats input on port 5044
- Grok filter to parse Nginx access logs
- Output to Elasticsearch

Logs were indexed using a time-based pattern:
shopez-logs-YYYY.MM.dd
This ensures structured and searchable log storage.

```
mihirmashruwala@monitor-vm:~$ sudo systemctl enable logstash
sudo systemctl start logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /
mihirmashruwala@monitor-vm:~$ sudo systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/lib/systemd/system/logstash.service; enabled; vendor prese
     Active: active (running) since Sun 2025-11-16 19:27:55 UTC; 4s ago
   Main PID: 7540 (java)
      Tasks: 20 (limit: 4687)
     Memory: 245.4M
        CPU: 7.503s
     CGroup: /system.slice/logstash.service
             └─7540 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.he

Nov 16 19:27:55 monitor-vm systemd[1]: Started logstash.
Nov 16 19:27:55 monitor-vm logstash[7540]: Using bundled JDK: /usr/share/logstas

mihirmashruwala@monitor-vm:~$ sudo nano /etc/logstash/conf.d/beat-pipeline.conf
```

```
  GNU nano 6.2
input {
  beats {
    port => 5044
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "filebeat-%{+YYYY.MM.dd}"
  }
}
```

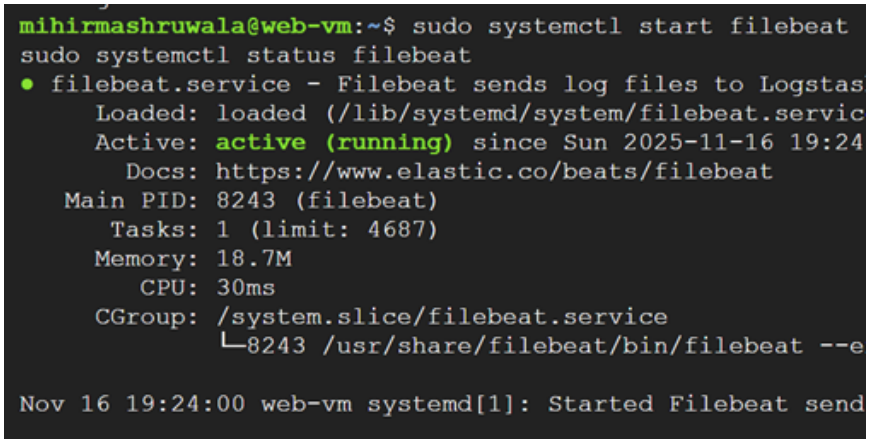
```
mihirmashruwala@monitor-vmm:~$ sudo nano /etc/logstash/conf.d/shopez.conf
mihirmashruwala@monitor-vmm:~$ sudo systemctl enable logstash
sudo systemctl start logstash
sudo systemctl status logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /lib/system
● logstash.service - logstash
     Loaded: loaded (/lib/systemd/system/logstash.service; enabled; vendor preset: enabled
     Active: active (running) since Sun 2025-11-16 20:13:59 UTC; 46ms ago
   Main PID: 5833 (logstash)
      Tasks: 2 (limit: 4687)
     Memory: 1.3M
        CPU: 25ms
     CGroup: /system.slice/logstash.service
             ├─5833 /bin/bash /usr/share/logstash/bin/logstash --path.settings /etc/logsta
             └─5845 /usr/share/logstash/jdk/bin/java -cp /usr/share/logstash/vendor/jruby/

Nov 16 20:13:59 monitor-vmm systemd[1]: Started logstash.
Nov 16 20:13:59 monitor-vmm logstash[5833]: Using bundled JDK: /usr/share/logstash/jdk
lines 1-13/13 (END)
```

```
mihirmashruwala@monitor-vm:~$ cat /etc/logstash/conf.d/shopez.conf
input {
  beats {
    port => 5044
  }
}

filter {
  if [source] =~ "access.log" {
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "shopez-logs-%{+YYYY.MM.dd}"
  }
}
```

Step 4: Filebeat Configuration and Log Shipping

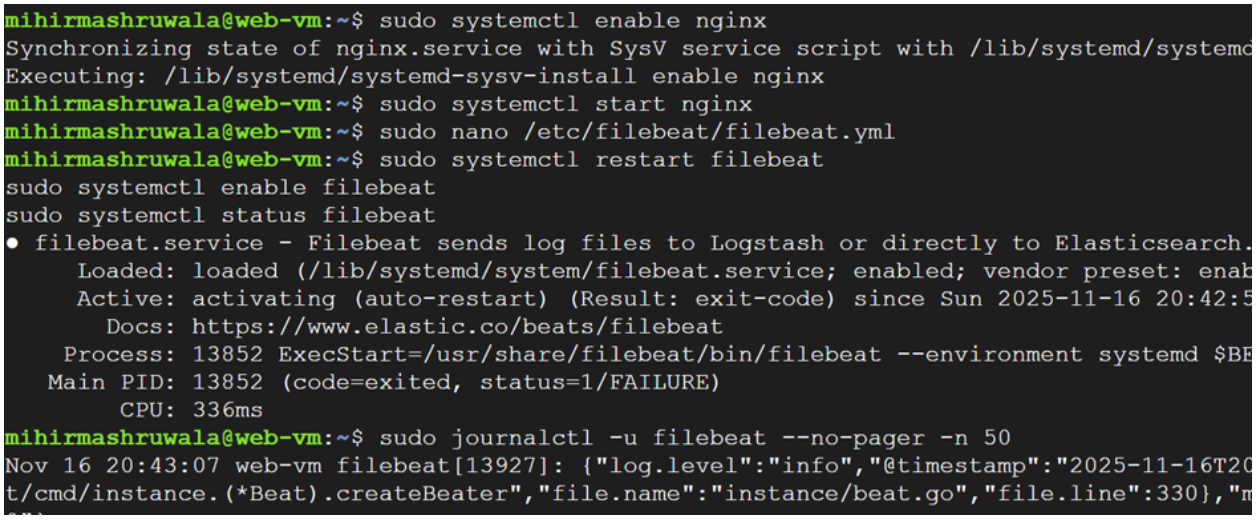
Filebeat was installed on web-vm and configured to monitor:
/var/log/nginx/access.log
Filebeat forwards logs in real time to Logstash on monitor-vm using the Beats protocol.
Configuration validation confirmed successful connectivity.

```
mihirmashruwala@web-vm:~$ sudo apt install filebeat -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 67.1 MB of archives.
After this operation, 258 MB of additional disk space will be u
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/mair
Fetched 67.1 MB in 1s (50.3 MB/s)
Selecting previously unselected package filebeat.
(Reading database ... 69439 files and directories currently ins
Preparing to unpack .../filebeat_8.19.7_amd64.deb ...
Unpacking filebeat (8.19.7) ...
Setting up filebeat (8.19.7) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries or
mihirmashruwala@web-vm:~$ sudo nano /etc/filebeat/filebeat.yml
mihirmashruwala@web-vm:~$ sudo filebeat modules enable system
Enabled system
mihirmashruwala@web-vm:~$ sudo filebeat test config
Config OK
```



Step 4: Filebeat Configuration and Log Shipping

Filebeat was installed on web-vm and configured to monitor:
/var/log/nginx/access.log
Filebeat forwards logs in real time to Logstash on monitor-vm using the Beats protocol.
Configuration validation confirmed successful connectivity.

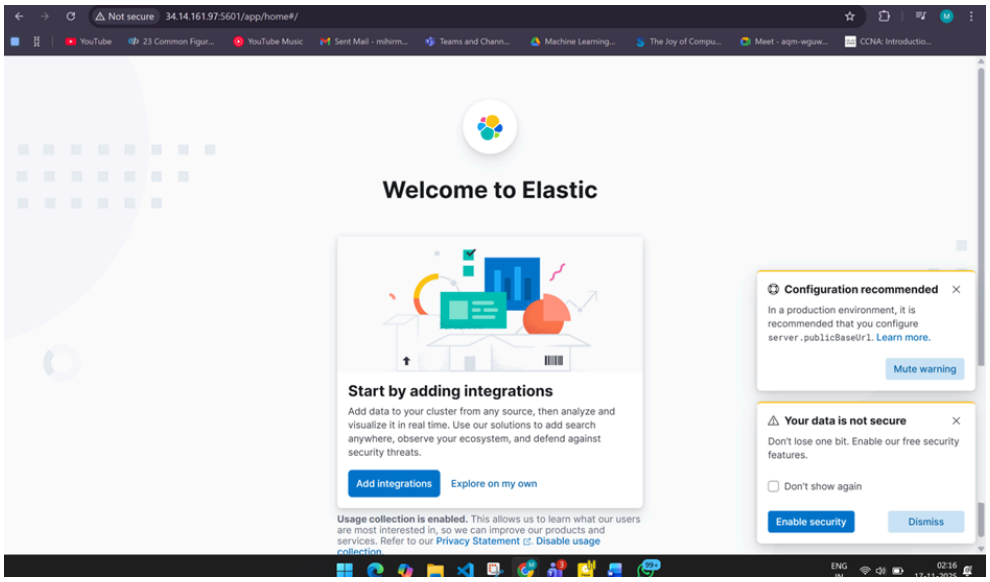


Step 5: Kibana Service Verification

After starting Kibana on monitor-vm, the service was accessed using the external IP address on port 5601.

Successful access to the “Welcome to Elastic” page confirmed that Kibana was running correctly and connected to Elasticsearch.

This validated that the visualization layer of the ELK stack was operational.

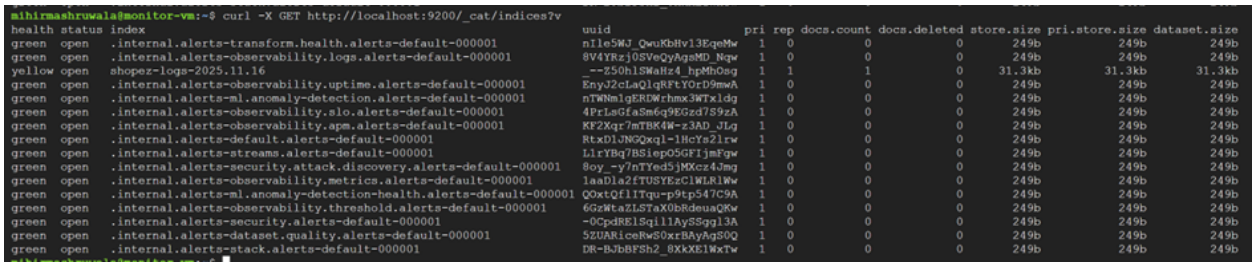


Step 6: Elasticsearch Index Verification

To verify that logs were successfully ingested, Elasticsearch indices were checked using:

curl -X GET http://localhost:9200/_cat/indices?v

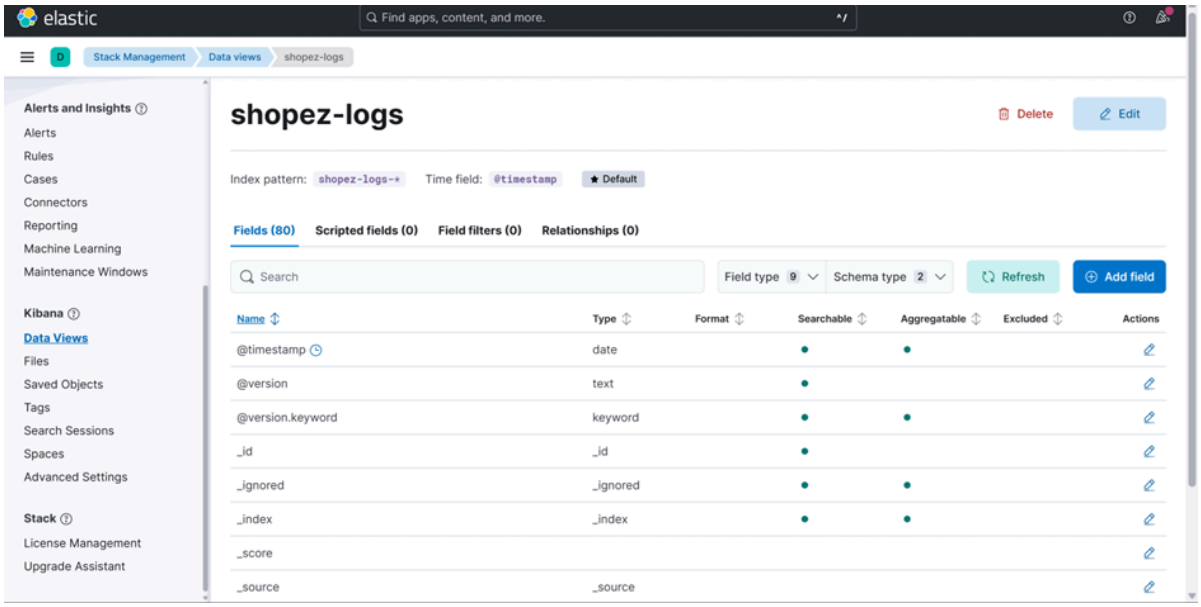
The output shows the creation of the index **`shopez-logs-*`** along with document count.
This confirms that Logstash successfully parsed logs and Elasticsearch indexed them.



```
mihirmashruwala@monitor-vm:~$ curl -X GET http://localhost:9200/_cat/indices?v
health status index                                                                uuid                   pri rep docs.count docs.deleted store.size pri.store.size dataset.size
green  open   .internal.alerts-transform.health.alerts-default-000001              n1le5WJ_QwuKbHv13EqeMw   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.logs.alerts-default-000001            8V4YRxjOSVeQyAgsMD_Nqw   1   0          0            0       249b           249b         249b
yellow open   shopez-logs-2025.11.16                                               __Z50hlSWaHz4_hpMhOsg   1   1          1            0      31.3kb         31.3kb       31.3kb
green  open   .internal.alerts-observability.uptime.alerts-default-000001          EnyJ2cLaQlqRFtYOrD9mwA   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-ml.anomaly-detection.alerts-default-000001          nTWNm1gERDWrhmx3WTx1dg   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.slo.alerts-default-000001             4PrLsGfaSm6q9EGzd7S9zA   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.apm.alerts-default-000001             KF2Xqr7mTBK4Wrz3AD_JLg   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-default.alerts-default-000001                       RtxD1JNGQxql-1HcYs2lrw   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-streams.alerts-default-000001                       LlrYBq7BSiepO5GFIjmFgw   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-security.attack.discovery.alerts-default-000001     8oy_-y7nTYed5jMXcz4Jmg   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.metrics.alerts-default-000001         1aaDla2fTUSYEzClWLRlWw   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-ml.anomaly-detection-health.alerts-default-000001 QOxtQflITqu-p9tp547C9A   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-observability.threshold.alerts-default-000001     6GzWtaZLSTaX0bRdeuaQKw   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-security.alerts-default-000001                     -0CpdRElSqil1AySSggl3A   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-dataset.quality.alerts-default-000001              5ZUARiceRwS0xrBAyAgSOQ   1   0          0            0       249b           249b         249b
green  open   .internal.alerts-stack.alerts-default-000001                        DR-BJbBFSh2_8XkXE1WxTw   1   0          0            0       249b           249b         249b
```

Step 7: Creating Index Pattern in Kibana

An index pattern **`shopez-logs-*`** was created in Kibana under Data Views.
The time field **`@timestamp`** was selected to enable time-based log analysis.
This step allows Kibana to recognize and visualize indexed log data from Elasticsearch.



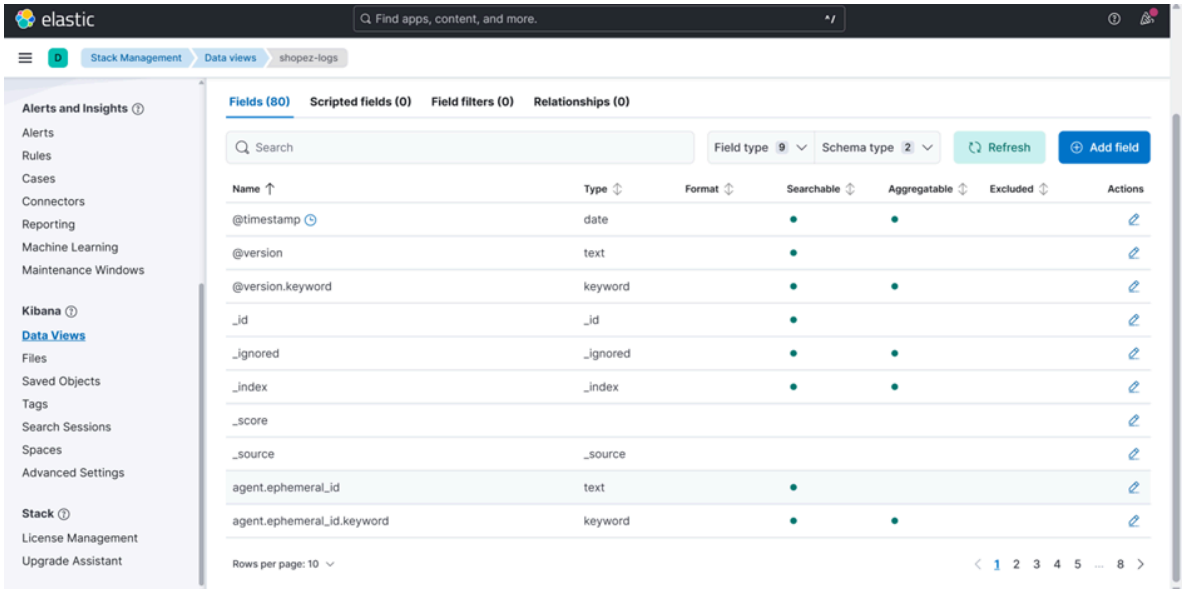
Step 8: Field Mapping Verification

After creating the index pattern, Kibana automatically detected available fields from indexed logs.

Fields such as:

- @timestamp
- version
- agent.id
- cloud.instance
- request metadata

were visible and searchable.
This confirms successful log parsing and structured indexing.



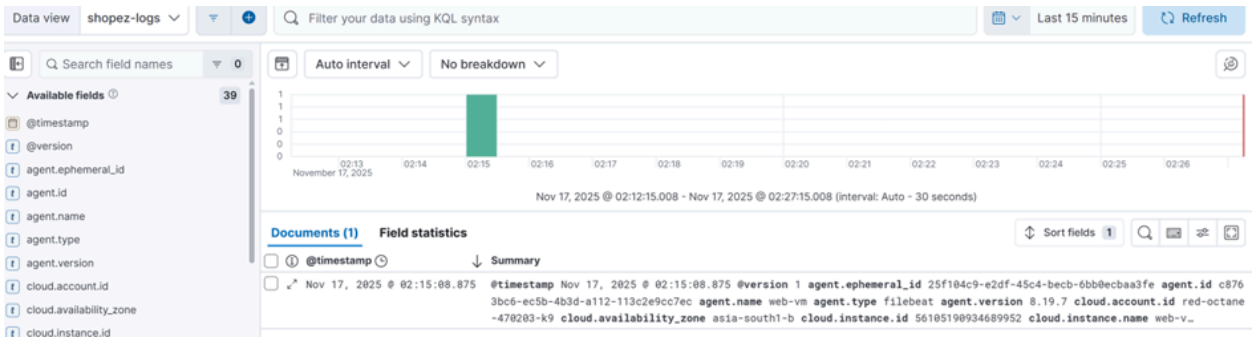
Step 9: Log Data Verification in Discover

Using the Discover tab, indexed logs were inspected in real time.

Each document contains:

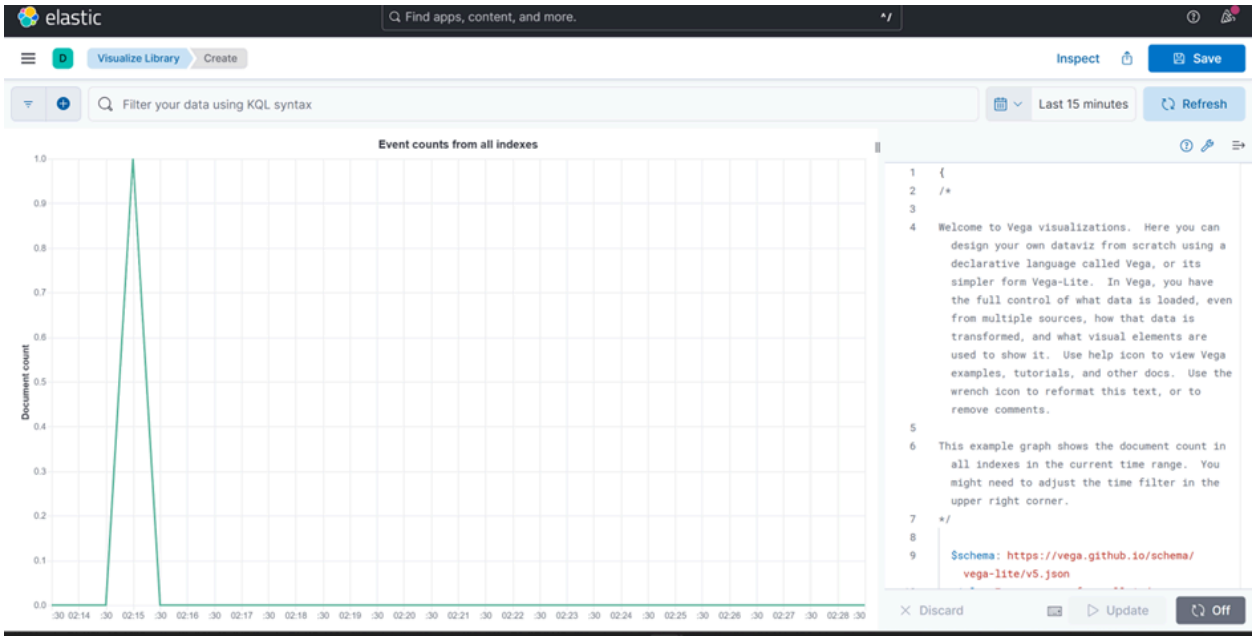
- Timestamp
- HTTP request details
- Cloud instance metadata
- Agent information

This confirms that logs from web-vm were successfully shipped, parsed, and stored.



Step 10: Visualization Setup

A visualization was created using Kibana's visualization tools to display event counts over time.
This graph represents document count indexed within selected time intervals.
Visualization enables monitoring of traffic trends and system behavior.

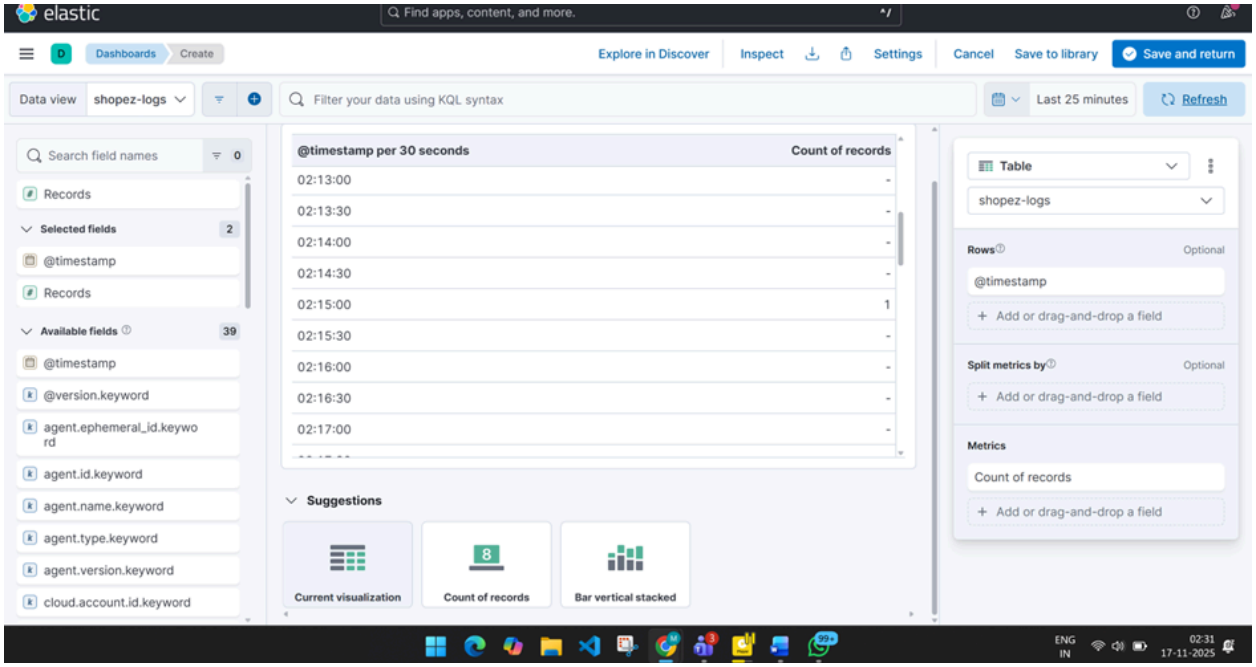


Step 11: Table-Based Log Analysis

A tabular visualization was configured to display:

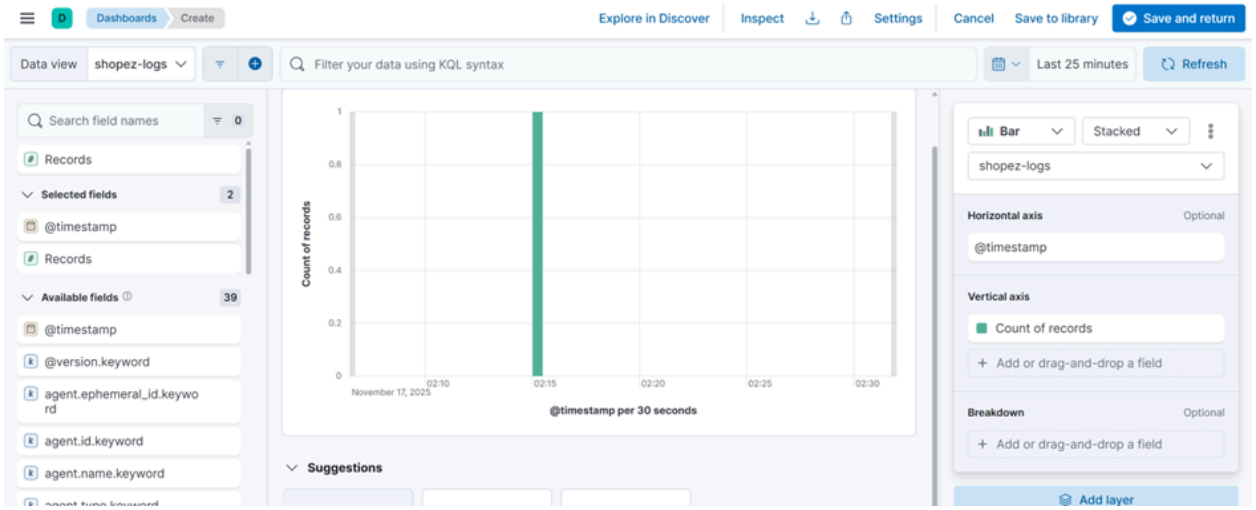
- @timestamp
- Count of records

This provides structured insight into log frequency over time.
Such visualizations help analyze traffic spikes and system activity.



Step 12: Time-Series Traffic Visualization

A bar chart visualization was created to represent log events per time interval.
This allows monitoring of request distribution and detection of anomalies in traffic patterns.
The visualization confirms that logs were continuously ingested and indexed.



Final Outcome

The ELK stack was successfully deployed on GCP using a distributed architecture.
The system demonstrates:

- Real-time log collection using Filebeat
- Log parsing with Logstash and Grok filters
- Time-based indexing in Elasticsearch
- Visualization and monitoring using Kibana
-

This setup replicates a production-style centralized logging pipeline in a cloud environment.