

# 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.

The screenshot shows the Google Cloud Platform's VM instances interface. At the top, there are buttons for 'Create instance', 'Import VM', and 'Refresh'. Below that is a navigation bar with tabs for 'Instances' (which is selected), 'Observability', and 'Instance schedules'. A search bar labeled 'Filter' with the placeholder 'Enter property name or value' is present. The main table lists two VM instances:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	<a href="#">monitor-vm</a>	us-central1-b			10.128.0.20 (nic0)	<a href="#">34.170.203.136 (nic0)</a>	SSH <input type="button" value="⋮"/>
<input checked="" type="checkbox"/>	<a href="#">web-vm</a>	us-central1-c			10.128.0.21 (nic0)	<a href="#">34.69.127.123 (nic0)</a>	SSH <input type="button" value="⋮"/>

Network Security

Cloud Armor

- DDoS Dashboard
- Cloud Armor policies
- Adaptive Protection
- Cloud Armor Service Tier

Cloud IDS

- IDS Dashboard
- IDS Endpoints
- IDS Threats

Cloud NGFW

- Dashboard
- Firewall policies**
- Threats
- Firewall endpoints

Secure Access Connect

- Realms Preview

Create a firewall rule

Protocols and ports [?](#)

Allow all

Specified protocols and ports

TCP

Ports  
22,5044,5601,9200

E.g. 20, 50-60

UDP

Ports

E.g. all

SCTP

Ports

E.g. 20, 50-60

Other

Protocols

Separate multiple protocols by commas, e.g. ah, icmp

Disable rule

**Create** **Cancel**

## 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/elasticsearch.service
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/guide/en/elasticsearch/reference/8.10/...
   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 -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=60 -Des.http.c...
                  ├─3041 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=60 -Des.http.c...
                  ├─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" : "HQ2KB3vZSL02yFjNQC4JwQ",
  "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 preset: enabled)
    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.headless=true -jar /usr/share/logstash/logstash.jar --config /etc/logstash/conf.d/beats.conf

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/logstash/jdk
```

```
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/systemd/system/logstash.service
● 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/logstash
                  ├─5845 /usr/share/logstash/jdk/bin/java -cp /usr/share/logstash/vendor/jruby/lib/ruby/gems/2.7.0/gems/jruby-openssl-1.0.0/lib/jni/libjopenssl.so /usr/share/logstash/jdk/bin/java -cp /usr/share/logstash/vendor/jruby/lib/ruby/gems/2.7.0/gems/jruby-openssl-1.0.0/lib/jni/libjopenssl.so -jar /usr/share/logstash/logstash.jar --config /etc/logstash/conf.d/shopez.conf

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 used.
Get:1 https://artifacts.elastic.co/packages/8.x/apt stable/main filebeat amd64 8.19.7+rev1-1
Fetched 67.1 MB in 1s (50.3 MB/s)
Selecting previously unselected package filebeat.
(Reading database ... 69439 files and directories currently installed)
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
```

```
mihirmashruwala@web-vm:~$ sudo systemctl start filebeat
sudo systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash
   Loaded: loaded (/lib/systemd/system/filebeat.service)
   Active: active (running) since Sun 2025-11-16 19:24
     Docs: https://www.elastic.co/beats/filebeat
     Main PID: 8243 (filebeat)
        Tasks: 1 (limit: 4687)
       Memory: 18.7M
          CPU: 30ms
        CGroup: /system.slice/filebeat.service
                  └─8243 /usr/share/filebeat/bin/filebeat --e

Nov 16 19:24:00 web-vm systemd[1]: Started Filebeat send
```

## 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.

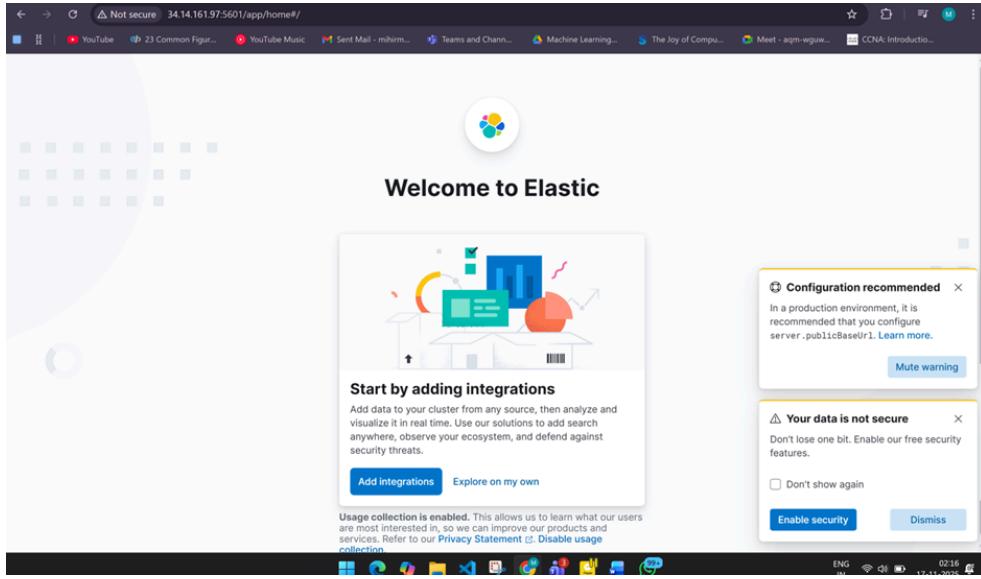
```
mihirmashruwala@web-vm:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemctl
Executing: /lib/systemd/systemd-sysv-install enable nginx
mihirmashruwala@web-vm:~$ sudo systemctl start nginx
mihirmashruwala@web-vm:~$ sudo nano /etc/filebeat/filebeat.yml
mihirmashruwala@web-vm:~$ sudo systemctl restart filebeat
sudo systemctl enable filebeat
sudo systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
   Active: activating (auto-restart) (Result: exit-code) since Sun 2025-11-16 20:42:50
     Docs: https://www.elastic.co/beats/filebeat
    Process: 13852 ExecStart=/usr/share/filebeat/bin/filebeat --environment systemd $BEAT_NAME
   Main PID: 13852 (code=exited, status=1/FAILURE)
      CPU: 336ms
mihirmashruwala@web-vm:~$ sudo journalctl -u filebeat --no-pager -n 50
Nov 16 20:43:07 web-vm filebeat[13927]: {"log.level":"info","@timestamp":"2025-11-16T20:43:07.000Z","@version":1,"@source":{"type":"cmd","instance":{},"(*Beat).createBeater"},"file.name":"instance/beat.go","file.line":330}, "m
```

## 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.

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_QwuKhBv13EqMw	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-observability.logs.alerts-default-000001	8V4YRzj0SVeQyAgsMD_Nqw	1	0	0	0	249b	249b	249b
yellow	open	shopez-logs-2025.11.16	_Z50hLSWaItz4_hpMh0sg	1	1	1	0	31.3kb	31.3kb	31.3kb
green	open	.internal.alerts-observability.uptime.alerts-default-000001	Enyu2cLaQLqFtYOrd9mwA	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-ml.anomaly-detection.alerts-default-000001	nTWnzbGRDWRchmx3WTx1.dg	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-observability.slo.alerts-default-000001	4PrLsGfaSm6g9E6Gz7dS9zA	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-observability.apm.alerts-default-000001	KF2Xqr_mTBqW-z3AD_JLq	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-default.alerts-default-000001	Rtx0lNGQxql-1HcYs21rw	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-streams.alerts-default-000001	L1ryBq7BSlepOSGFijmfqw	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	1aAdla2FTUSYEzC1WdRlw	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-ml.anomaly-detection.health.alerts-default-000001	Q0xtQ11fqu-p9tp547C9A	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-observability.threshold.alerts-default-000001	6GzWtaZLSraX0hRdeuaQKw	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-security.alerts-default-000001	-0Cp4REISgll1aySgg13A	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-dataset.quality.alerts-default-000001	52UAN1celNwSOxrBhyhgs00	1	0	0	0	249b	249b	249b
green	open	.internal.alerts-stack.alerts-default-000001	DR-BjhBFSh_8XX0E1WxTw	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.

The screenshot shows the Elasticsearch Stack Management interface. On the left, there's a sidebar with various management sections like Alerts, Rules, Cases, Connectors, Reporting, Machine Learning, Maintenance Windows, Kibana, Data Views, and Stack. The main area is titled 'shopez-logs' and shows an index pattern of 'shopez-logs-\*'. It has a time field of '@timestamp' and a default field type. Below this, there are tabs for 'Fields (80)', 'Scripted fields (0)', 'Field filters (0)', and 'Relationships (0)'. A search bar and filter buttons for 'Field type' (dropdown), 'Schema type' (dropdown), 'Refresh' button, and 'Add field' button are at the top of the table. The table lists 80 fields with columns for 'Name', 'Type', 'Format', 'Searchable', 'Aggregatable', 'Excluded', and 'Actions'.

## 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.

The screenshot shows the Elasticsearch Stack Management interface. On the left, there's a sidebar with various management sections like Alerts, Rules, Cases, Connectors, Reporting, Machine Learning, Maintenance Windows, Kibana, Data Views, and Stack. The main area is titled 'shopez-logs' and shows an index pattern of 'shopez-logs-\*'. It has a time field of '@timestamp' and a default field type. Below this, there are tabs for 'Fields (80)', 'Scripted fields (0)', 'Field filters (0)', and 'Relationships (0)'. A search bar and filter buttons for 'Field type' (dropdown), 'Schema type' (dropdown), 'Refresh' button, and 'Add field' button are at the top of the table. The table lists 80 fields with columns for 'Name', 'Type', 'Format', 'Searchable', 'Aggregatable', 'Excluded', and 'Actions'. At the bottom, there's a 'Rows per page' dropdown set to 10, and a page navigation bar with buttons for <, 1, 2, 3, 4, 5, ..., 8, >.

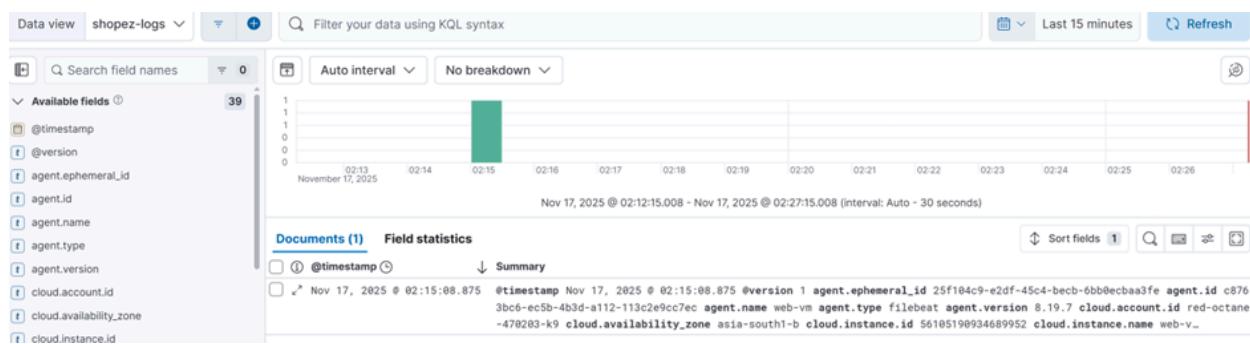
## Step 9: Log Data Verification in Discover

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

Each document contains:

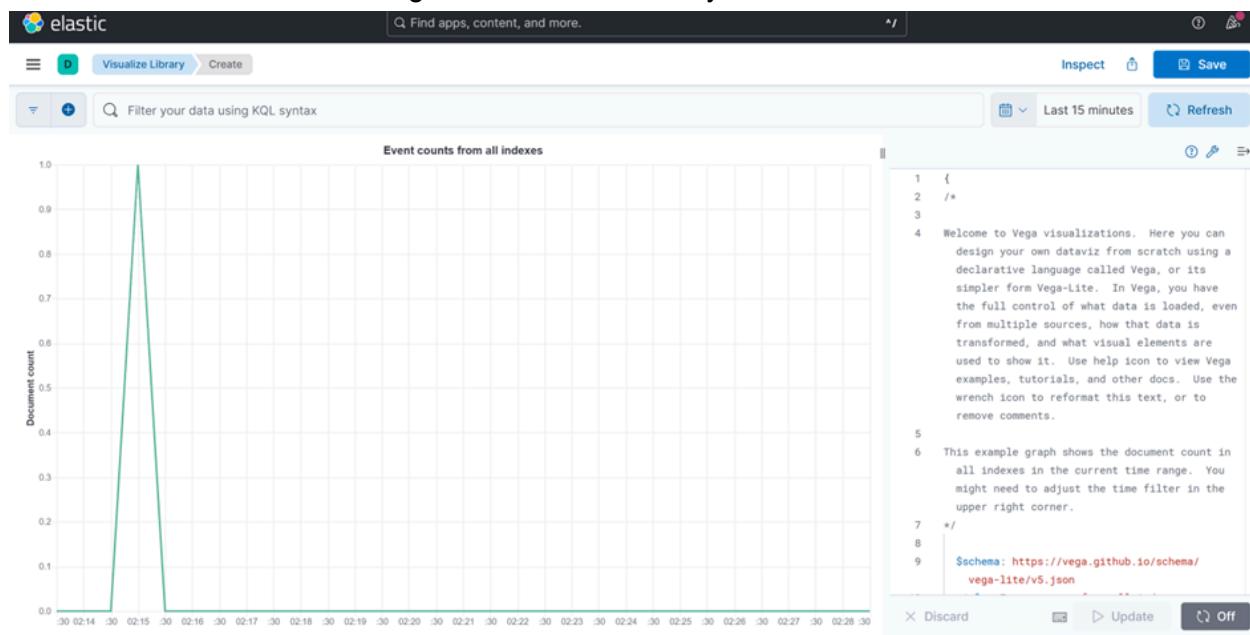
- Timestamp
- HTTP request details
- Cloud instance metadata
- Agent information

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.



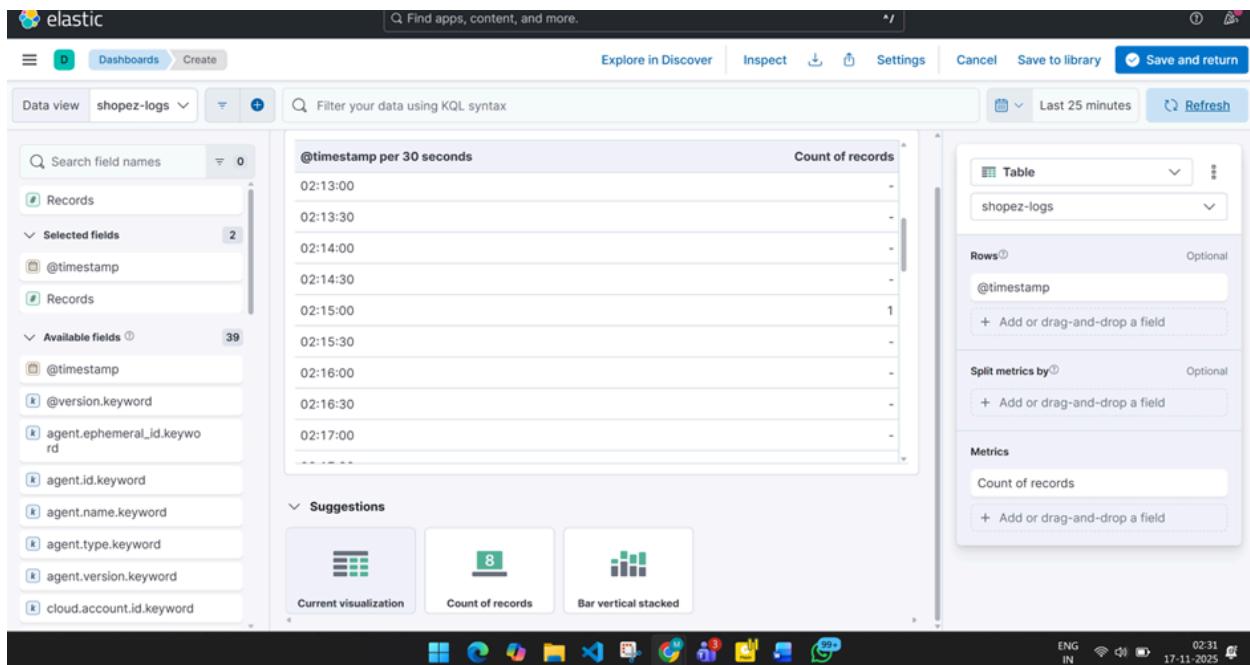
## Step 11: Table-Based Log Analysis

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.

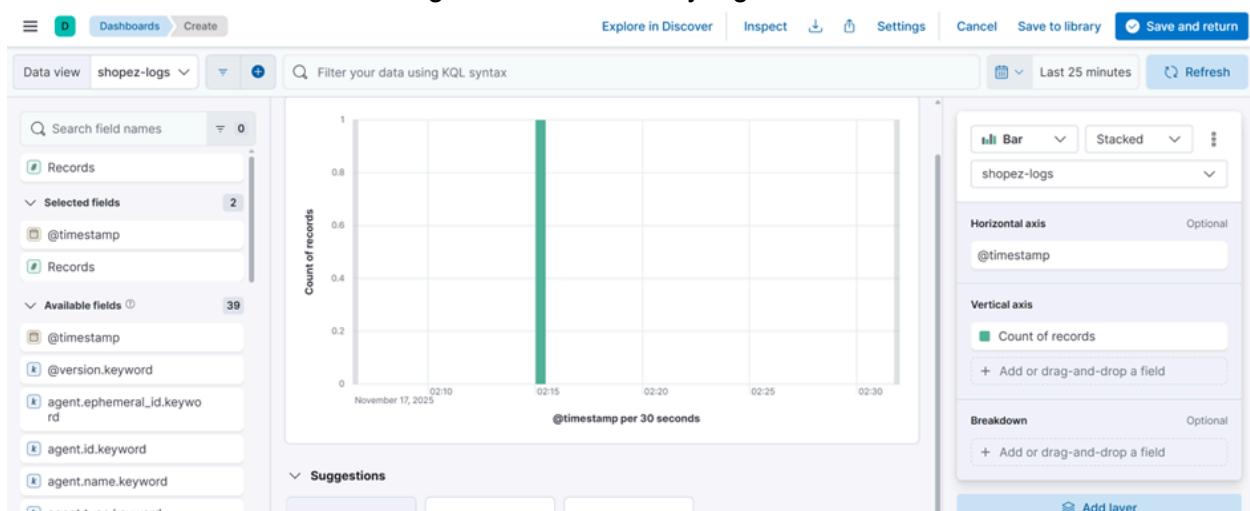


## Step 12: Time-Series Traffic Visualization

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.