OPEN-SOURCE CROSS-PLATFORM DATABASE LOG MONITORING SOLUTION

whoami

Major:

- System Security- MUST-SICT

Experience:

- Banking: 2+
- Telecommunication: 4+

Agenda

01 **Business needs**
Why this project

02 **TCO & ROSI**
Calculation

03 **System Overview**
Architecture & Components

04 **Automation**
Alert, Report, Check, Delete

05 **Conclusion**

Why this project?

Business & Security Drivers

Compliance:

- Mongolian Law on Personal Data Protection (2019) requires proper handling and retention of personal and sensitive data.
- Critical sectors to retain logs for audit purposes.

Security Requirements

- Centralized monitoring to detect threats.
- Supports forensic investigations in case of incidents.

Operational Benefits

- Standardized log collection across Oracle, MySQL, PostgreSQL
- Simplifies audit reporting and regulatory compliance
- Reduces operational risk and ensures accountability

What we need

Goal: Build a flexible, cost-effective, open-source log monitoring system

Scope: Oracle, MySQL, PostgreSQL across any OS or version

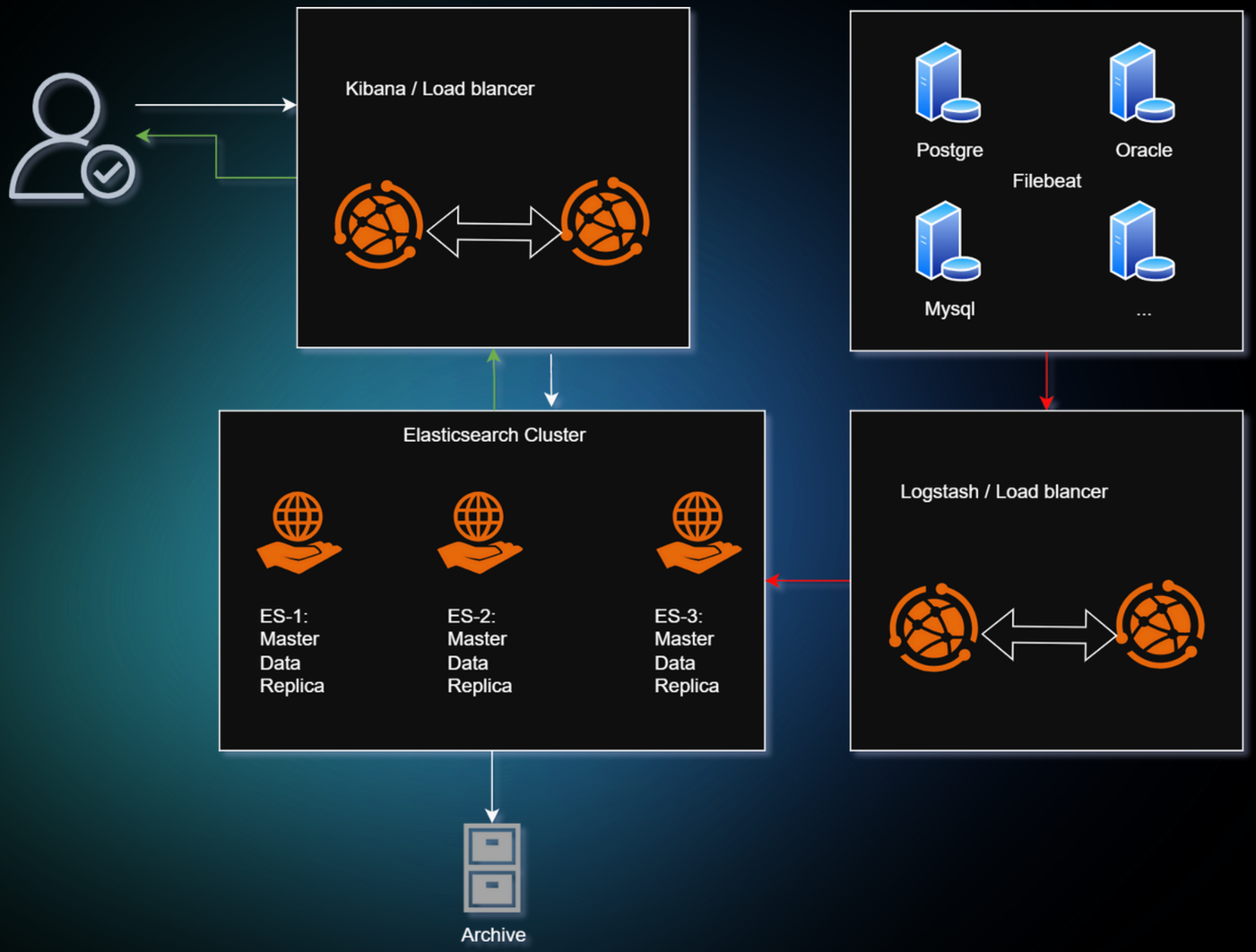
Components:

- Filebeat + Logstash: Collect database logs
- Elasticsearch: Store and index logs
- Kibana: Real-time visualization and dashboards
- Python Script: Custom alerting/reporting based on log patterns
- Automated Retention Job: Deletes/Archives logs older than 365 days, sends notifications

Why this project?

	DBF	OPEN-SOURCE SOLUTION
cost	High licensing fees	Free / low cost (only infrastructure)
flexibility	Limited by vendor	Fully customizable (Python, Filebeat, Logstash)
database support	**Depends on OS, kernel, DB version**	**Oracle, MySQL, PostgreSQL (any version/OS)**
scalability	May require premium plan	Easily scalable via Elasticsearch cluster
alerting & automation	Built-in, sometimes limited	Fully programmable via Python scripts
action	**Can block**	**No block feature**

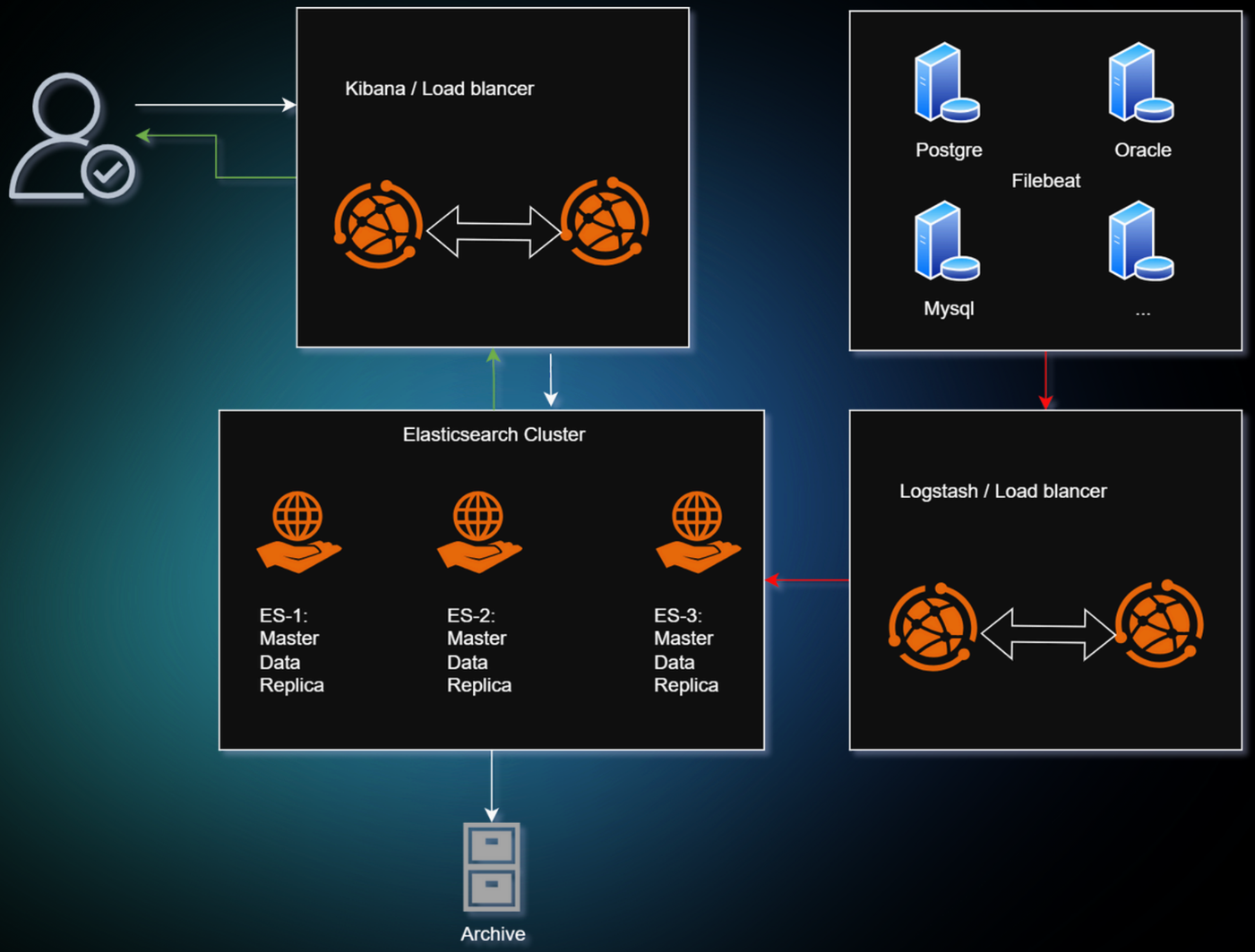
Architecture



- 7 database /5 table each/
- Elastic /3 server- HA /
- Logstash /2 server- HA/
- Kibana /2 server-HA/
- Archive server /1 server/

Figure 1: System architecture

Architecture



№	Hostname	RAM (GB)	vCPU	Storage (TB)	Support (Hardware and management)
1	Elastic node1	64	16	5.6	Yes
2	Elastic node2	64	16	5.6	Yes
3	Elastic node3	64	16	5.6	Yes
4	Logstash & Kibana cluster node1	32	8	0.25	Yes
5	Logstash & Kibana cluster node2	32	8	0.25	Yes

Cost Type	1 year (USD)	3 year (USD)	5 year (USD)
Support and maintenance	5,159	15,477	25,795
Management software	1,548	4,644	7,740
Facilities costs	348	1,044	1,740
Total	**$7,055**	**$21,165**	**$35,275**

Figure 1: System architecture

TCO (total cost of ownership)

7 database:

at least:

№	Hostname	RAM (GB)	vCPU	Storage (TB)	Support *(Hardware and management)*
1	Elastic node1	32	12	5.6	Yes
2	Elastic node2	32	12	5.6	Yes
3	Logstash & Kibana cluster node1	16	8	0.25	Yes

Cost Type	1 year (USD)	3 year (USD)	5 year (USD)
Support and maintenance	2,403	7,208	12,013
Management software	721	2,162	3,604
Facilities costs	162	487	813
Total	**$3,286**	**$9,858**	**$16,430**

recommended:

№	Hostname	RAM (GB)	vCPU	Storage (TB)	Support (Hardware and management)
1	Elastic node1	64	16	5.6	Yes
2	Elastic node2	64	16	5.6	Yes
3	Elastic node3	64	16	5.6	Yes
4	Logstash & Kibana cluster node1	32	8	0.25	Yes
5	Logstash & Kibana cluster node2	32	8	0.25	Yes

Cost Type	1 year (USD)	3 year (USD)	5 year (USD)
Support and maintenance	5,159	15,477	25,795
Management software	1,548	4,644	7,740
Facilities costs	348	1,044	1,740
Total	**$7,055**	**$21,165**	**$35,275**

TCO (total cost of ownership)

7 database:

Cost Type	1 year (USD)	3 year (USD)	5 year (USD)
Support and maintenance	5,159	15,477	25,795
Management software	1,548	4,644	7,740
Facilities costs	348	1,044	1,740
Total	**$7,055**	**$21,165**	**$35,275**

DBF solution:

Cost Component	1 year (USD)	3 year (USD)	5 year (USD)
License (yearly)	4,800	14,400	24,000
Support (yearly)	44,500	133,500	222,500
Appliance (one-time)			
Professional services (one-time)			
MX	8,100	24,300	40,500
Archive Server 1	165	495	825
Total	**$57,765**	**$173,295**	**$288,825**

Not included salary

ROSI (return of security investment)

	Global	Example
SLE= Customer data /PII/	$169	$50
Records		20,000
ARO/Likelihood/	Every 2-4 years	0.3
Effectiveness	80%	

$$\text{ROSI} = \frac{\text{Risk Reduction} - \text{Control Cost}}{\text{Control Cost}} = \frac{240,000 - 7,000}{7,000} = \frac{233,000}{7,000} \approx 33.29\,(3329\%)$$

$$\text{ROSI} = \frac{\text{Risk Reduction} - \text{Control Cost}}{\text{Control Cost}} = \frac{240,000 - 57,000}{57,000} = \frac{183,000}{57,000} \approx 3.21\,(321\%)$$

SLE = $50 x 20k = $1 m

ALE after = $1 m × 0.3 = $300,000/year

Risk Reduction = $300,000 x 0.8 = $240,000/year

Cost of Control (paid) = $57,000

Cost of Control (opensource) = $7,000

Effectiveness (%)	Risk Reduction ($)	ALE After ($)	ROSI (7,000$)	ROSI (57,000$)
0	0	300,000	-1.00 (-100%)	-1.00 (-100%)
10	30,000	270,000	3.29	-0.47 (-47%)
20	60,000	240,000	7.57	0.05 (5%)
30	90,000	210,000	11.86	0.58 (58%)
40	120,000	180,000	16.14	1.11 (111%)
50	150,000	150,000	20.43	1.63 (163%)
60	180,000	120,000	24.71	2.16 (216%)
70	210,000	90,000	28.99	2.68 (268%)
80	240,000	60,000	33.29	3.21 (321%)
90	270,000	30,000	37.57	3.74 (374%)
100	300,000	0	41.86	4.26 (426%)

effectiveness 10% → +

effectiveness 40% → +

ARO= Annualized Rate of Occurrence

ALE= Annualized Loss Expectancy

SLE= Single Loss Expectancy

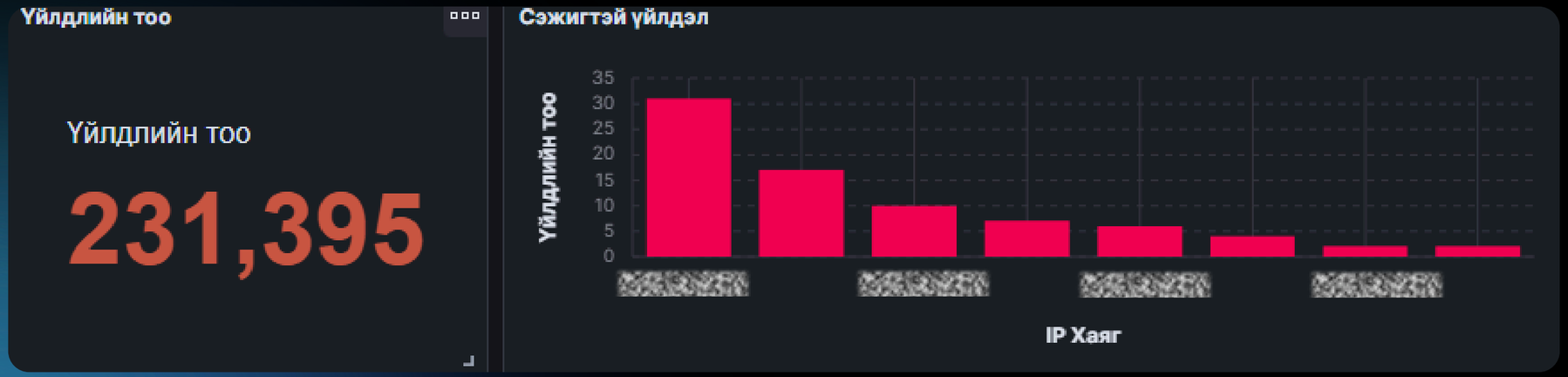
source: https://socradar.io/ibms-cost-of-data-breach-report-2024-cybersecurity/?utm_source=chatgpt.com

System Overview

1 week

DASHBOARD

This dashboard included a total count of events, suspicious actions. Showing covering tables name, action percentage by IP address, and by users.
Also can create a dashboard via requirements. For such as business needs, security requirements, and audit purpose.

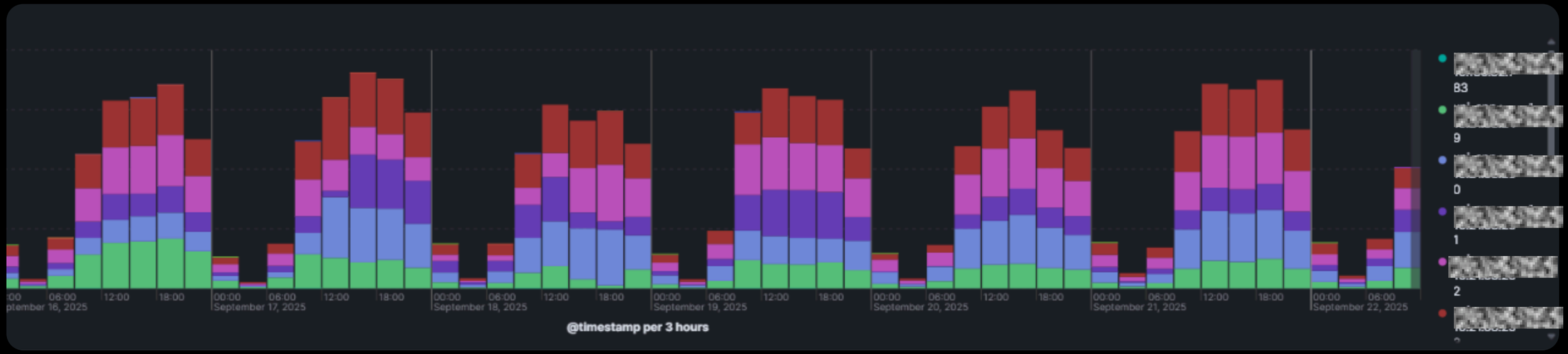


Dashboard

1 week

TOTAL EVENTS, SUSPICIOUS EVENTS BY IP ADDRESS.

The total events section counts the log coming through the index log.
The suspicious event section indicates that the log is coming with an unauthorized IP address.



Dashboard

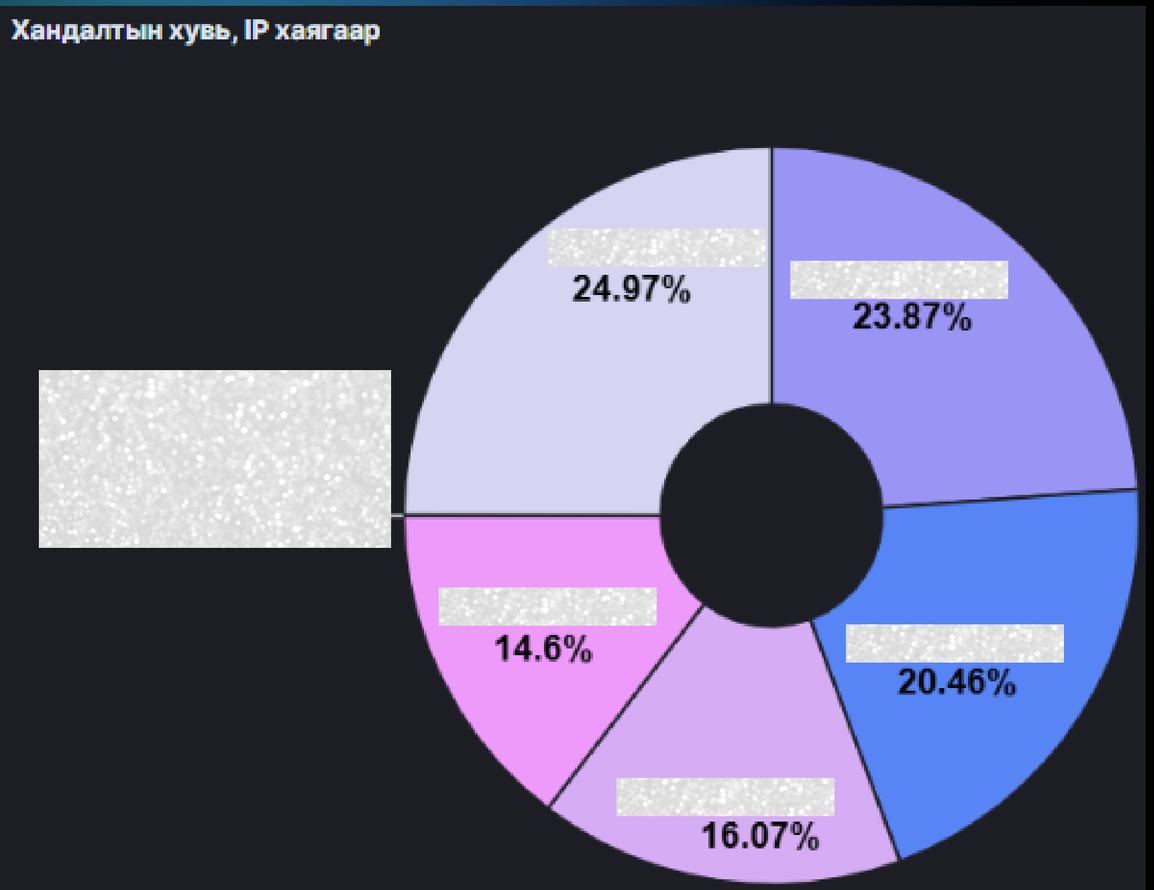
1 week

EVENT COUNT SHOWING USER WITHIN IP BY TIME

This bar shows overall users activities by time. It shows a lot of things, such as how many users are in there, which account is shared, how many activities they did.

Dashboard

1 week

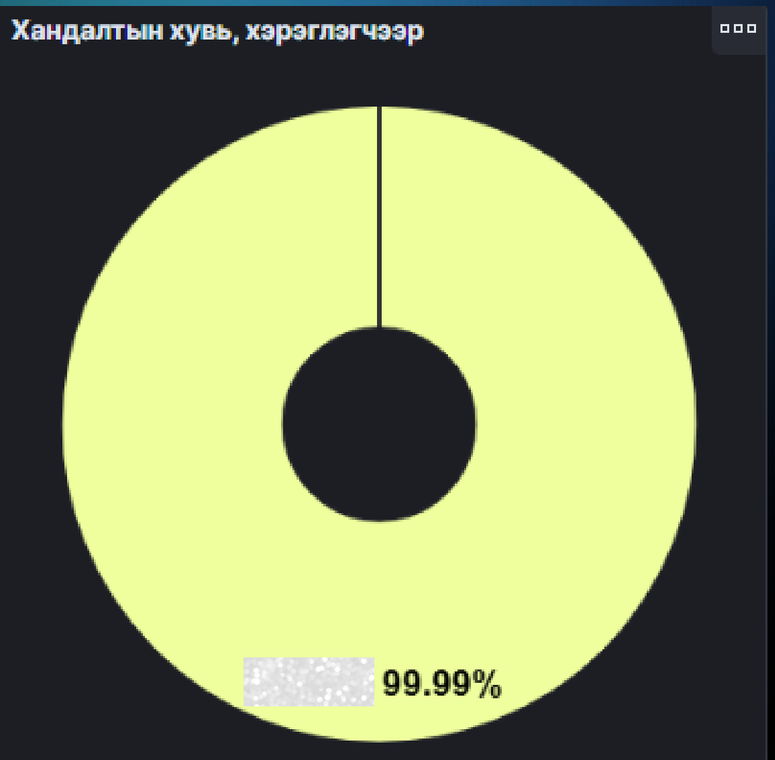


PERCENTAGE OF ACCESS BY IP

This Donut shows the access percentages by IP address. Means which IP address is accessing and performing actions in the monitoring tables of database.

Dashboard

1 week



PERCENTAGE OF ACCESS BY USER

This Donut shows the access percentages by USER. Means which user is accessing and performing actions in the monitoring tables of database.

System Overview

Logstash

1. Focus main area *(PII)*

Why filebeat ?

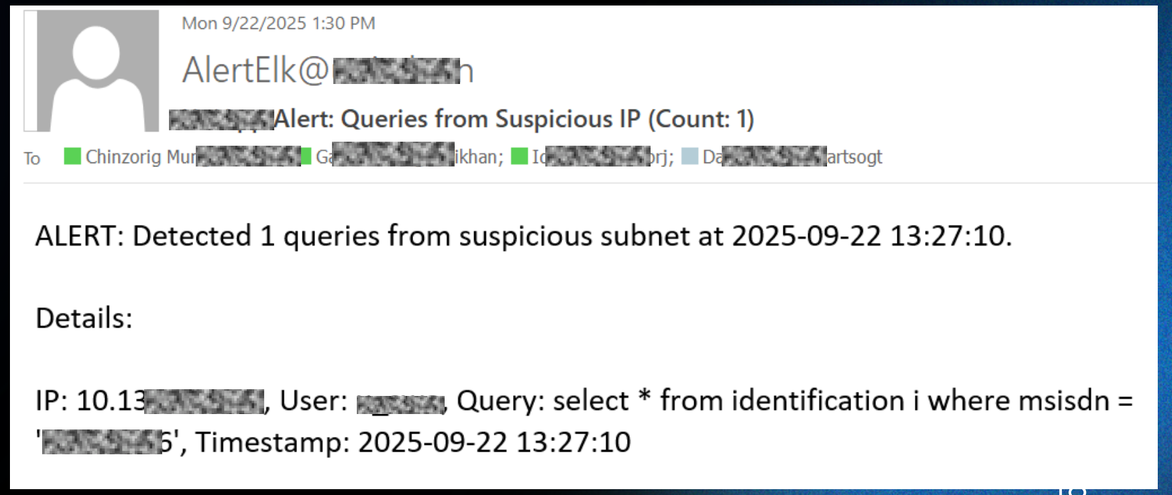
```
# [Service]
# CPUQuota=13%
# MemoryLimit=500M
```

```
filter {
  if "192.168.[redacted]app" in [tags] {
    grok {
      match => { "message" => "%{GREEDYDATA:log_content}" }
    }

    if [log_content] =~ "change_plan"
    or [log_content] =~ "identification"
    or [log_content] =~ "number_sales"
    or [log_content] =~ "change_admin_number" {
      mutate {
        add_field => { "[@metadata][index_name]" => "mysql-logs-192.168.[redacted]app" }
      }
    } else {
      drop { }
    }

    grok {
      match => { "message" => "user=%{WORD:user_field}" }
    }

    grok {
      match => { "message" => "client=%{IP:src_ip}" }
    }
```

System Overview

Alert

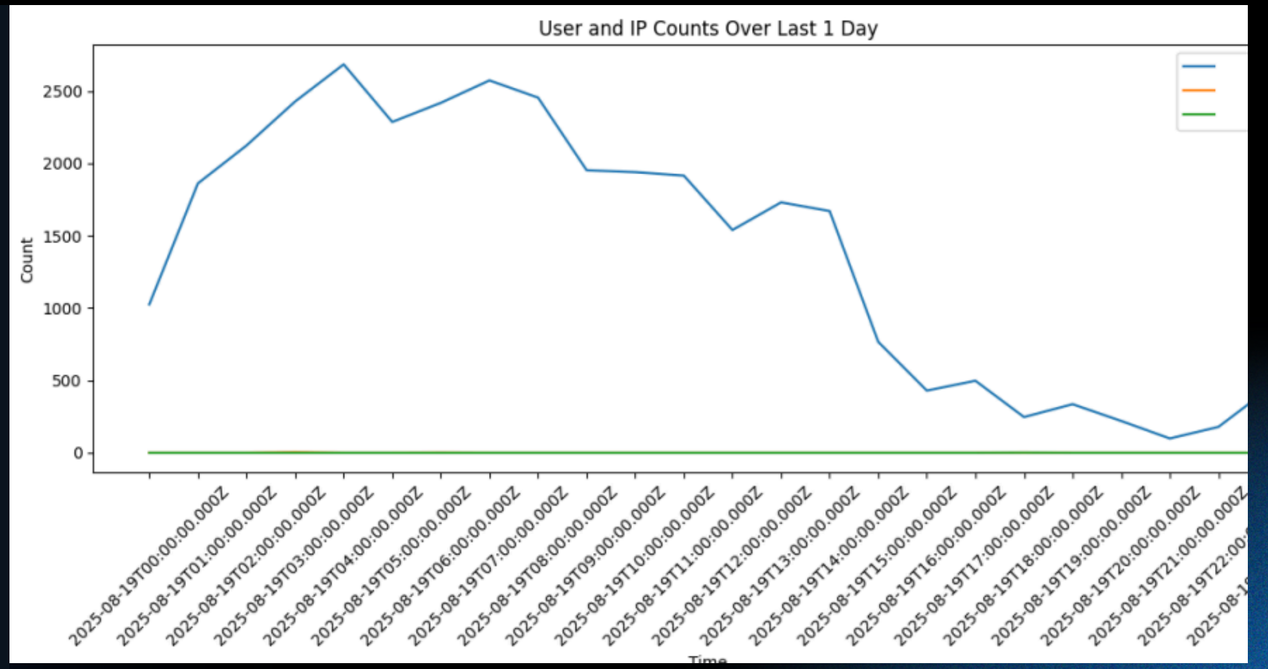
```
query = {
    "size": 10,
    "query": {
        "bool": {
            "must": [
                {"range": {"@timestamp": {"gte": "now-5m", "lte": "now"}}}
            ],
            "should": [
                {"wildcard": {"src_ip.keyword": "10.136.*"}},
                {"wildcard": {"src_ip.keyword": "172.16.*"}},
                {"wildcard": {"src_ip.keyword": "10.22.*"}},
                {"wildcard": {"src_ip.keyword": "10.23.*"}},
                {"wildcard": {"src_ip.keyword": "10.24.*"}},
                {"wildcard": {"src_ip.keyword": "10.25.*"}}
            ],
            "minimum_should_match": 1
        }
    },
    "_source": ["src_ip", "user_field", "message", "@timestamp"]  # Retrieve
}
```



System Overview

Report

```
# Plot the graph
plt.figure(figsize=(12, 6))
for user, data in users_data.items():
    counts = [data[timestamp] for timestamp in all_timestamps]
    plt.plot(all_timestamps, counts, label=user)

plt.xlabel("Time")
plt.ylabel("Count")
plt.title("User and IP Counts Over Last 1 Day")
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
graph_file = "user_ip_graph.png"
plt.savefig(graph_file)
plt.close()

# Generate PDF report
pdf = FPDF()
pdf.add_page()
pdf.set_font("Arial", size=12)
pdf.cell(200, 10, txt="Toki dpp master: User and IP Counts (Last 1 Day)", ln=True, align='C')
pdf.ln(10)

for user_bucket in response["aggregations"]["by_user"]["buckets"]:
    pdf.cell(0, 10, txt=f"User: {user_bucket['key']} (Count: {user_bucket['doc_count']})", ln=True)
    for ip_bucket in user_bucket["by_ip"]["buckets"]:
        pdf.cell(0, 10, txt=f"  - IP: {ip_bucket['key']} (Count: {ip_bucket['doc_count']})", ln=True)
    pdf.ln(5)

pdf.image(graph_file, x=10, y=pdf.get_y(), w=190)
pdf_file = "user_ip_report.pdf"
pdf.output(pdf_file)
```

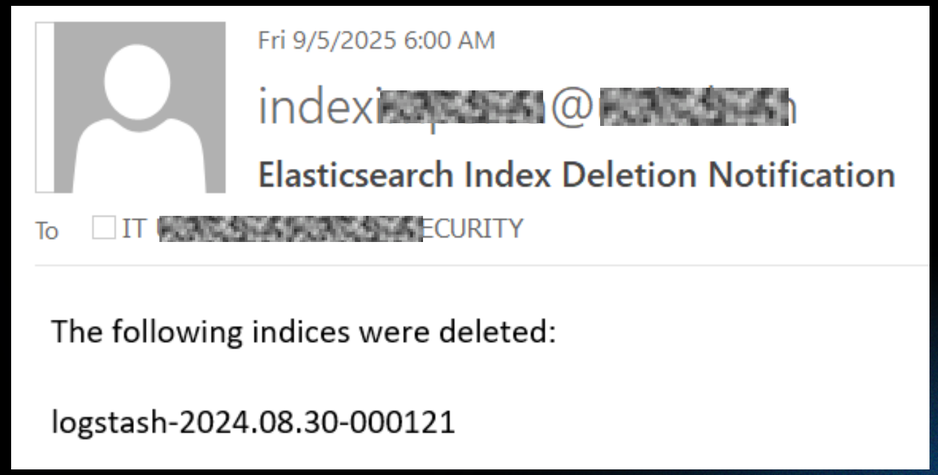


Automation

Storage

1. Delete old data

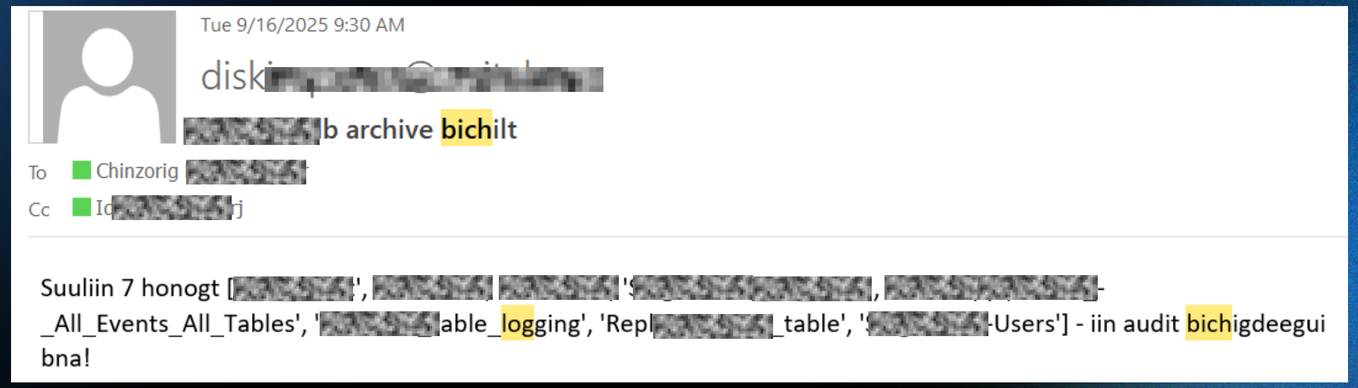
```
def delete_old_indices():
    url = f"{es_url}_cat/indices?format=json"
    response = requests.get(url, auth=(username, password))

    if response.status_code == 200:
        indices = response.json()
        deleted_indices = []

        for index in indices:
            index_name = index['index']
            match = re.match(r"logstash-(\d{4})\.(\d{2})\.(\d{2})-\d+$", index_name)

            if match:
                settings_url = f"{es_url}{index_name}/_settings"
                settings_response = requests.get(settings_url, auth=(username, password))

                if settings_response.status_code == 200:
                    settings_data = settings_response.json()
                    creation_date = settings_data[index_name]['settings']['index']['creation_date']
                    index_creation_datetime = datetime.utcfromtimestamp(int(creation_date) / 1000)

                    if index_creation_datetime < cutoff_date:
                        delete_url = f"{es_url}{index_name}"
                        delete_response = requests.delete(delete_url, auth=(username, password))

                        if delete_response.status_code == 200:
                            print(f"Deleted index: {index_name}")
                            deleted_indices.append(index_name)
                        else:
                            print(f"Error deleting index {index_name}: {delete_response.status_code}")

        if deleted_indices:
            body = f"The following indices were deleted:\n\n" + "\n".join(deleted_indices)
            send_email(subject, body)
            run_commands()
        else:
            body = "No indices older than 365 days were found for deletion."
            send_email(subject, body)

    else:
        print(f"Error fetching indices: {response.status_code}")
```



Automation

Log check

```
for line in stdout:
    #print line
    for _ in names:
        if _ in line:
            names.remove(_) # dor hayj 1 log bichigdsen tul listees gargav
#mime type-d audit bichigdeegui nersiig oruulna.
sender = 'disk[redacted].mn'
receivers = 'chinzorig.m@unitel.mn'
msg = MIMEText('Suuliin 7 honogt ' + str(names) + ' - iin audit bichigdeegui bna!')
print msg
```



Log integrity and availability are critical for compliance, auditing, and incident response.

CONCLUSION

- **Cost-Effective** – open-source stack, reduces license costs.
- **Reliable** – redundancy & replication prevent log/data loss.
- **Free to Scale** – easily expand nodes as data grows.
- **Compliance-Ready** – supports retention, integrity, and access control for audits.
- **Secure** – encryption, authentication, and file permission controls.
- **Flexible** – integrates with multiple log sources (databases, apps, systems).

Proper architecture design, redundancy planning, and security measures transform ELK into a reliable and compliant log management solution.

Thank you

Chinzoo.gl@gmail.com